

Guide d'installation et de configuration de Linux

Par Christian Casteyde

Date de publication : 1 janvier 2004

Dernière mise à jour : 16 janvier 2020

TOUT PUBLIC

L'objet de ce tutoriel est de donner les connaissances de base nécessaires à l'installation de Linux sur un ordinateur de particulier ou un petit serveur. Tout sera expliqué au fur et à mesure des besoins et, si nécessité est, des compléments d'information seront donnés pour permettre la compréhension des opérations à effectuer. Néanmoins, les notions qui seront abordées ici ne seront pas simples, et il est possible que la plupart des personnes qui n'ont pas une grande habitude de l'informatique aient quelques difficultés à les assimiler. Cela dit, à vaincre sans peine, on triomphe sans gloire, et l'installation de Linux vous procurera le plaisir d'apprendre.

N'hésitez pas à commenter cet article ! [Commentez](#)

I - Introduction.....	11
I-A - Matériel, logiciel et système d'exploitation.....	11
I-B - Qu'est-ce que Linux ?.....	11
I-C - Pourquoi Linux ?.....	12
I-D - Plan du document.....	13
II - GNU, Linux et les logiciels libres.....	14
II-A - Les logiciels libres et la Free Software Foundation.....	14
II-B - Droits d'auteurs et la licence GPL.....	14
II-C - Le projet GNU et Linux.....	15
II-D - Avantages et inconvénients des logiciels libres.....	16
II-E - La motivation des auteurs et le financement des logiciels libres.....	17
III - Installation du système.....	18
III-A - Choix de la distribution.....	18
III-B - Partitionnement et systèmes de fichiers.....	19
III-B-1 - Notion de partition.....	20
III-B-2 - Notion de système de fichiers.....	22
III-B-2-a - Définition.....	22
III-B-2-b - Nommage des fichiers.....	23
III-B-2-c - Montage des systèmes de fichiers.....	23
III-B-2-d - Identification des partitions.....	24
III-B-3 - Choix du plan de partitionnement.....	25
III-C - Prérequis.....	27
III-C-1 - Récupération des informations sur le matériel.....	27
III-C-2 - Sauvegarde des données.....	28
III-D - Procédure d'installation générique.....	29
III-D-1 - Premier démarrage.....	29
III-D-2 - Trouver de la place pour installer le système.....	30
III-D-2-a - Utilisation de parted.....	30
III-D-2-b - Utilisation de fips.....	32
III-D-3 - Partitionnement du disque.....	33
III-D-4 - Création des systèmes de fichiers.....	34
III-D-5 - Création de la partition de swap.....	36
III-D-6 - Installation des composants de base.....	37
III-D-7 - Configuration du gestionnaire d'amorçage.....	39
III-D-7-a - Réalisation d'un multiboot avec LILO.....	39
III-D-7-b - Réalisation d'un multiboot avec le GRUB.....	43
III-D-7-c - Réalisation d'un multiboot avec NTLDR.....	45
III-E - Installation de la Mandriva 2007.....	46
III-F - Installation de la Debian 4.0r1.....	53
III-G - Installation de la Slackware 12.0.....	60
IV - Présentation générale du système.....	69
IV-A - Architecture du système.....	70
IV-A-1 - Principe de modularité.....	70
IV-A-2 - Les différentes couches logicielles sous Linux.....	71
IV-A-3 - Résumé de l'architecture de Linux.....	72
IV-B - Sécurité et utilisateurs.....	73
IV-B-1 - Généralités.....	73
IV-B-2 - Notion d'utilisateur et d'administrateur.....	74
IV-B-3 - La sécurité au niveau du système de fichiers.....	75
IV-B-3-a - Les droits sur les fichiers.....	76
IV-B-3-b - Notion d'ACL.....	77
IV-B-3-c - Les attributs spéciaux des fichiers.....	77
IV-C - Fonctionnalités des systèmes de fichiers.....	79
IV-C-1 - Le système de fichiers virtuel.....	79
IV-C-2 - Liens symboliques et liens physiques.....	81
IV-C-3 - Autres fonctionnalités.....	82
IV-D - Structure du système de fichiers.....	83
V - Commandes Unix de base.....	88

V-A - Login et déconnexion.....	88
V-B - Arrêt et redémarrage du système.....	90
V-C - Pages de manuel.....	90
V-D - Opérations de base sur les répertoires.....	92
V-E - Notions sur les chemins Unix.....	93
V-F - Opérations de base sur les fichiers.....	95
V-G - Autres commandes utiles.....	96
V-G-1 - Gestion des liens.....	96
V-G-2 - Recherche de fichiers.....	97
V-G-3 - Recherche d'un texte dans un fichier.....	97
V-G-4 - Remplacement de texte dans les fichiers.....	97
V-G-5 - Compression et décompression des fichiers.....	98
V-G-6 - Archivage de fichiers.....	99
V-G-7 - Passage en mode superviseur.....	99
V-G-8 - Changement des droits des fichiers, du propriétaire et du groupe.....	100
V-G-8-a - Changement de propriétaire et de groupe.....	100
V-G-8-b - Modification des droits Unix sur les fichiers.....	100
V-G-8-c - Utilisation des ACL.....	101
V-H - vi, l'éditeur de fichiers de base.....	102
V-I - Utilisation du shell bash.....	104
V-I-1 - Contrôle des processus.....	105
V-I-1-a - Lancement d'un programme en arrière-plan.....	105
V-I-1-b - Listing des processus.....	106
V-I-1-c - Notion de signal.....	106
V-I-1-d - Arrêt d'un processus.....	107
V-I-1-e - Gel d'un processus.....	107
V-I-1-f - Relancement d'un processus.....	107
V-I-2 - Redirections.....	108
V-I-2-a - Principe de base.....	108
V-I-2-b - Redirections de données en entrée.....	108
V-I-2-c - Redirection de données en sortie.....	109
V-I-2-d - Insertion de documents.....	110
V-I-3 - Les tubes.....	111
V-I-3-a - Syntaxe des tubes.....	111
V-I-3-b - Les tubes nommés.....	113
V-I-3-c - La commande tee.....	113
V-I-3-d - La commande xargs.....	114
V-I-4 - Manipulation des variables d'environnement.....	114
V-I-5 - Caractère d'échappement et chaînes de caractères.....	118
V-I-6 - Les substitutions.....	120
V-I-6-a - Génération de chaînes de caractères selon un motif.....	120
V-I-6-b - Substitution du nom d'utilisateur.....	120
V-I-6-c - Remplacements de variables.....	120
V-I-6-d - Substitution du résultat d'une commande.....	123
V-I-6-e - Évaluation d'expressions arithmétiques.....	123
V-I-6-f - Substitution de commandes.....	124
V-I-6-g - Découpage en mots.....	125
V-I-6-h - Remplacement des caractères génériques.....	125
V-I-7 - Les expressions rationnelles.....	125
V-I-8 - Structures de contrôle.....	126
V-I-8-a - Les instructions composées.....	126
V-I-8-b - Les tests.....	128
V-I-8-c - Le branchement conditionnel.....	131
V-I-8-d - Les boucles.....	131
V-I-8-e - Les itérations.....	132
V-I-8-f - Les ruptures de séquence.....	132
V-I-8-g - Les fonctions.....	133
V-I-8-h - Les entrées / sorties de données.....	134

V-I-9 - Les alias.....	135
V-I-10 - Les scripts shell.....	135
VI - Administration du système de base.....	136
VI-A - Sauvegarde de la configuration d'installation.....	137
VI-B - Mise à l'heure du système.....	137
VI-C - Gestion des utilisateurs et de la sécurité.....	140
VI-C-1 - Mécanismes d'authentification des utilisateurs.....	140
VI-C-2 - Création et suppression des utilisateurs.....	142
VI-C-3 - Description de la bibliothèque PAM.....	144
VI-D - Gestion des paquetages.....	146
VI-D-1 - Le gestionnaire de paquetages rpm.....	147
VI-D-2 - Le gestionnaire de paquetages apt.....	147
VI-D-3 - Le gestionnaire de paquetages pkgtool.....	149
VI-E - Notion de niveau d'exécution et amorçage du système.....	150
VI-F - Maintenance des systèmes de fichiers.....	152
VI-F-1 - Création des systèmes de fichiers.....	152
VI-F-2 - Montage des systèmes de fichiers.....	152
VI-F-3 - Démontage des systèmes de fichiers.....	154
VI-F-4 - Vérification des systèmes de fichiers.....	155
VI-F-5 - Configuration du montage des systèmes de fichiers.....	157
VI-F-6 - Montage des systèmes de fichiers à la demande.....	160
VI-G - Gestion des volumes.....	162
VI-G-1 - Gestion des fichiers images.....	162
VI-G-2 - Agrégation de volumes.....	163
VI-G-3 - Chiffrement des systèmes de fichiers.....	164
VI-H - Configuration des terminaux virtuels.....	166
VI-I - Configuration de la console.....	167
VI-I-1 - Pages de codes et Unicode.....	168
VI-I-2 - Principe de fonctionnement du clavier.....	169
VI-I-3 - Principe de fonctionnement de l'écran de la console.....	171
VI-I-4 - Configuration du clavier.....	173
VI-I-4-a - Définition de scancodes.....	173
VI-I-4-b - Définition d'un plan de clavier.....	175
VI-I-4-c - Modification des paramètres du clavier.....	178
VI-I-5 - Choix de la police de caractères.....	179
VI-I-6 - Configuration des paramètres du terminal.....	181
VI-I-7 - Description des terminaux.....	181
VI-I-8 - Paramétrage des applications.....	184
VI-I-8-a - Configuration du clavier pour la bibliothèque readline.....	185
VI-I-8-b - Configuration du clavier pour vi.....	185
VI-I-8-c - Configuration du clavier pour less.....	188
VI-I-9 - Configuration de la souris.....	189
VI-J - Configuration de l'imprimante.....	190
VI-J-1 - Concepts de base de l'impression sous Unix.....	190
VI-J-2 - Le système d'impression LPRng.....	191
VI-J-2-a - Le mécanisme des filtres APSFILTER.....	191
VI-J-2-b - Installation des filtres et configuration des files d'impression.....	192
VI-J-2-c - Commandes d'impression.....	193
VI-J-2-d - Description du fichier /etc/printcap.....	194
VI-J-3 - Le système d'impression CUPS.....	195
VI-J-3-a - Le mécanisme des filtres de CUPS.....	195
VI-J-3-b - Configuration d'une imprimante CUPS.....	196
VI-J-3-c - Les fichiers de configuration de CUPS.....	196
VI-K - Configuration du lancement automatique des tâches.....	199
VI-L - Gestion de l'énergie.....	200
VI-L-1 - Généralités sur la gestion de l'énergie.....	200
VI-L-2 - Configuration de la gestion de l'énergie.....	201
VI-L-3 - Le démon ACPI.....	202

VII - Notions de compilation et configuration du noyau.....	204
VII-A - Notions de base.....	204
VII-A-1 - Définition des termes.....	204
VII-A-2 - Processus de génération.....	207
VII-B - Compilation de GCC.....	209
VII-B-1 - Prérequis.....	210
VII-B-2 - Installation des sources.....	210
VII-B-3 - Configuration.....	210
VII-B-4 - Compilation.....	211
VII-B-5 - Installation de GCC.....	212
VII-C - Compilation du noyau Linux.....	212
VII-C-1 - Installation des sources de Linux.....	213
VII-C-2 - Choix des options de configuration du noyau.....	213
VII-C-3 - Compilation et installation du noyau.....	214
VII-C-4 - Compilation et installation des modules.....	216
VIII - Configuration du matériel et des périphériques.....	216
VIII-A - Généralités sur le support matériel sous Linux.....	216
VIII-A-1 - Modules du noyau.....	217
VIII-A-1-a - Chargement et déchargement des modules.....	217
VIII-A-1-b - Options des modules.....	218
VIII-A-1-c - Options des périphériques intégrés au noyau.....	219
VIII-A-2 - Les fichiers spéciaux de périphériques.....	220
VIII-A-2-a - Généralités.....	220
VIII-A-2-b - Le système de fichiers virtuels udev.....	220
VIII-A-2-b-i - Présentation.....	220
VIII-A-2-b-ii - Principe de fonctionnements de udev.....	221
VIII-A-2-b-iii - Identification et chargement des pilotes de périphérique.....	223
VIII-A-2-b-iv - Chargement des firmwares.....	224
VIII-A-2-b-v - Création des fichiers spéciaux de périphériques.....	225
VIII-A-2-b-vi - Persistance des fichiers spéciaux de périphériques.....	226
VIII-A-2-b-vii - Notification des applications utilisateurs.....	228
VIII-A-2-b-viii - Initialisation du système.....	229
VIII-A-2-c - Gestion statique des fichiers spéciaux de périphériques.....	229
VIII-B - Configuration des périphériques de masse.....	231
VIII-B-1 - Configuration des périphériques SCSI.....	231
VIII-B-2 - Configuration des disques durs IDE.....	232
VIII-B-3 - Installation d'un graveur de CD/DVD.....	234
VIII-B-3-a - Notions de base sur le gravage sous Linux.....	234
VIII-B-3-b - Configuration du noyau.....	234
VIII-B-3-c - Configuration des modules du noyau.....	236
VIII-B-3-d - Installation des logiciels de gravage.....	236
VIII-B-3-e - Utilisation des logiciels de gravage.....	236
VIII-C - Configuration des cartes filles.....	239
VIII-C-1 - Généralités sur les cartes ISA, Plug And Play et PCI.....	240
VIII-C-2 - Configuration des cartes son.....	242
VIII-C-2-a - Fonctionnalités du matériel.....	242
VIII-C-2-b - Configuration du noyau.....	243
VIII-C-2-c - Configuration des modules du noyau.....	244
VIII-C-2-d - Ajustage des paramètres audio avec ALSA.....	245
VIII-C-2-e - Fichiers MIDI et synthétiseurs logiciels.....	247
VIII-C-3 - Installation d'une carte graphique 3D.....	248
VIII-C-4 - Installation d'une carte d'acquisition vidéo.....	250
VIII-C-5 - Configuration des cartes réseau.....	252
VIII-C-6 - Configuration des adaptateurs Wi-Fi.....	252
VIII-D - Configuration des ports de communication.....	256
VIII-D-1 - Prise en charge des périphériques ISA standards.....	257
VIII-D-1-a - Configuration du port parallèle.....	257
VIII-D-1-b - Configuration des ports série.....	258

VIII-D-2 - Installation des périphériques USB.....	259
VIII-D-3 - Installation des périphériques IEEE1394.....	260
VIII-D-3-a - Configuration du noyau.....	261
VIII-D-3-b - Installation des bibliothèques complémentaires.....	261
IX - Configuration du réseau.....	262
IX-A - Notions de réseau TCP/IP.....	263
IX-A-1 - Généralités sur les réseaux.....	263
IX-A-2 - Le protocole IP.....	264
IX-A-3 - Le protocole TCP.....	270
IX-A-4 - Les protocoles de haut niveau.....	272
IX-B - Configuration du réseau sous Linux.....	272
IX-B-1 - Configuration statique des interfaces réseau.....	272
IX-B-2 - Définition des règles de routage.....	274
IX-B-3 - Définition du nom de la machine.....	275
IX-B-4 - Résolution des noms de domaine.....	276
IX-B-5 - Utilisation des protocoles DHCP et BOOTP.....	277
IX-B-5-a - Autoconfiguration des clients DHCP et BOOTP.....	277
IX-B-5-b - Configuration d'un client DHCP au niveau utilisateur.....	278
IX-B-6 - Définition des protocoles de haut niveau.....	279
IX-B-7 - Les super-démons inetd et xinetd.....	280
IX-B-7-a - Le super-démon inetd.....	281
IX-B-7-b - Le super-démon xinetd.....	281
IX-C - Configuration de la connexion à Internet.....	286
IX-C-1 - Le protocole PPP.....	286
IX-C-2 - Création d'une connexion à Internet.....	287
IX-C-3 - Connexion à l'ADSL.....	291
IX-C-4 - Les autres outils de connexion.....	293
IX-C-5 - Configuration d'un cache de DNS.....	293
IX-C-6 - Installation d'un proxy HTTP.....	295
IX-D - Pare-feu et partages de connexion à Internet.....	298
IX-D-1 - Mécanismes de filtrage du noyau.....	298
IX-D-2 - Translations d'adresses et masquering.....	300
IX-D-3 - Trajet des paquets dans le code de Netfilter.....	301
IX-D-4 - Configuration du noyau et installation des outils.....	302
IX-D-5 - Utilisation d'iptables.....	303
IX-D-5-a - Manipulation des chaînes.....	303
IX-D-5-b - Manipulation des règles.....	304
IX-D-6 - Exemple de règles.....	305
IX-D-6-a - Exemple de règles de filtrage.....	306
IX-D-6-b - Exemple de partage de connexion à Internet.....	308
IX-D-7 - Configuration des clients.....	309
IX-E - Configuration de la sécurité du réseau.....	310
IX-E-1 - Limitation des services et des accès.....	311
IX-E-1-a - Réduction du nombre des services.....	311
IX-E-1-b - Définition de règles de contrôle d'accès.....	311
IX-E-1-b-i - Restrictions d'accès avec tcpd.....	312
IX-E-1-b-ii - Restrictions d'accès avec xinetd.....	313
IX-E-1-c - Contrôle des utilisateurs au niveau des services.....	314
IX-E-2 - Chiffrement des communications.....	314
IX-E-2-a - Principes de base de cryptographie.....	315
IX-E-2-b - Utilisation de SSH.....	317
IX-E-2-b-i - Principes de base de l'authentification SSH.....	318
IX-E-2-b-ii - Compilation et installation d'OpenSSH.....	319
IX-E-2-b-iii - Configuration d'OpenSSH côté serveur.....	320
IX-E-2-b-iv - Utilisation d'OpenSSH côté client.....	322
IX-E-2-b-v - Création d'un tunnel SSH.....	323
IX-E-2-c - Utilisation d'IPSec.....	324
IX-E-2-c-i - Fonctionnement d'IPSec.....	324

IX-E-2-c-ii - Configuration manuelle d'IPSec en mode transport.....	325
IX-E-2-c-iii - Configuration manuelle d'IPSec en mode tunnel.....	327
IX-E-2-c-iv - Autoconfiguration avec ISAKMP.....	328
IX-F - Configuration des fonctions serveur.....	332
IX-F-1 - Paramétrage des connexions extérieures.....	332
IX-F-2 - Configuration des liaisons PPP.....	333
IX-F-3 - Liaison de deux ordinateurs par un câble série.....	337
IX-F-4 - Configuration d'un serveur DHCP.....	337
IX-G - Systèmes de fichiers en réseau.....	339
IX-G-1 - Installation d'un serveur de fichiers NFS.....	339
IX-G-2 - Configuration d'un client NFS.....	342
IX-G-3 - Installation d'un serveur de fichiers SMB.....	343
IX-G-4 - Configuration d'un client SMB.....	349
X - Installation de XWindow.....	352
X-A - Généralités sur XWindow.....	352
X-B - Installation de X.org.....	354
X-C - Configuration de X.org.....	355
X-C-1 - Génération automatique du fichier xorg.conf.....	356
X-C-2 - Utilisation de xorgconfig.....	356
X-C-3 - Utilisation de xorgcfg.....	359
X-C-3-a - Configuration en mode graphique.....	359
X-C-3-b - Configuration en mode texte.....	362
X-C-4 - Description du fichier xorg.conf.....	363
X-C-4-a - Structure générale du fichier xorg.conf.....	363
X-C-4-b - Section « Files ».....	365
X-C-4-c - Section « ServerFlags ».....	366
X-C-4-d - Section « Module ».....	366
X-C-4-e - Section « InputDevice ».....	367
X-C-4-f - Sections « Device ».....	368
X-C-4-g - Sections « Monitor ».....	369
X-C-4-h - Sections « Modes ».....	377
X-C-4-i - Sections « Screen ».....	377
X-C-4-j - Sections « ServerLayout ».....	378
X-C-5 - Informations utilisées lors du démarrage de X.org.....	379
X-C-6 - Utilisation de xvidtune.....	380
X-D - Utilisation du pilote frame buffer du noyau.....	380
X-D-1 - Configuration du noyau et installation du pilote.....	381
X-D-2 - Configuration du serveur X.....	382
X-E - Configuration des terminaux X.....	383
X-E-1 - Principe de fonctionnement de xdm.....	383
X-E-2 - Configuration de xdm.....	383
X-E-2-a - Serveurs X locaux.....	384
X-E-2-b - Serveurs X utilisant XDMCP.....	385
X-E-2-c - Paramétrage du serveur X pour utiliser le protocole XDMCP.....	387
X-E-2-d - Fichiers d'initialisation de sessions.....	387
X-E-3 - Paramétrage des terminaux X.....	388
X-E-3-a - La commande xset.....	389
X-E-3-b - Configuration de la disposition du clavier.....	390
X-F - Paramétrage des applications et ressources X.....	392
X-G - Gestion de la sécurité sous XWindow.....	394
X-G-1 - La commande xhost.....	395
X-G-2 - La commande xauth.....	395
X-H - Gestion des polices de caractères.....	396
X-H-1 - Gestion des polices de caractères sous XWindow.....	396
X-H-2 - Installation des polices Truetype.....	398
X-H-2-a - Configuration du serveur X.....	398
X-H-2-b - Configuration des polices Truetype pour l'impression.....	399
X-H-2-b-i - Conversion des polices Truetype en polices Adobe de Type 42.....	400

X-H-2-b-ii - Installation des polices TrueType pour GhostScript.....	400
X-H-3 - Configuration d'un serveur de polices.....	401
X-I - Problèmes classiques rencontrés.....	403
XI - Conclusion.....	404
XII - Annexe A. Options de configuration du noyau.....	404
XII-A - Menu « General setup ».....	404
XII-A-1 - Sous-menu « Configure standard kernel features (for small systems) ».....	406
XII-B - Menu « Loadable module support ».....	408
XII-C - Menu « Block layer ».....	408
XII-C-1 - Sous-menu « IO Schedulers ».....	409
XII-D - Menu « Processor type and features ».....	409
XII-D-1 - Sous-menu « Firmware Drivers ».....	414
XII-E - Menu « Power management options (ACPI, APM) ».....	415
XII-F - Menu « Bus options (PCI, PCMCIA, EISA, MCA, ISA) ».....	416
XII-F-1 - Sous-menu « PCCARD (PCMCIA/CardBus) support ».....	417
XII-F-2 - Sous-menu « PCI Hotplug Support ».....	418
XII-G - Menu « Executable file formats / Emulations ».....	418
XII-H - Menu « Networking ».....	418
XII-H-1 - Menu « Networking options ».....	418
XII-H-1-a - Sous-menu « IP: Virtual Server Configuration ».....	423
XII-H-1-b - Sous-menu « Network packet filtering (replace ipchains) ».....	424
XII-H-1-b-i - Sous-menu « Core Netfilter Configuration ».....	424
XII-H-1-b-ii - Sous-menu « IP: Netfilter Configuration ».....	428
XII-H-1-b-iii - Sous-menu « IPv6: Netfilter Configuration ».....	430
XII-H-1-b-iv - Sous-menu « DECnet: Netfilter Configuration ».....	431
XII-H-1-b-v - Sous-menu « Bridge: Netfilter Configuration ».....	431
XII-H-1-c - Sous-menu « DCCP Configuration (EXPERIMENTAL) ».....	431
XII-H-1-d - Sous-menu « SCTP Configuration (EXPERIMENTAL) ».....	431
XII-H-1-e - Sous-menu « TIPC Configuration (EXPERIMENTAL) ».....	431
XII-H-1-f - Sous-menu « QoS and/or fair queueing ».....	432
XII-H-1-g - Sous-menu « Network testing ».....	432
XII-H-2 - Menu « Amateur Radio support ».....	432
XII-H-2-a - Sous-menu « AX.25 network device drivers ».....	432
XII-H-3 - Menu « IrDA (infrared) subsystem support ».....	432
XII-H-3-a - Sous-menu « Infrared-port device drivers ».....	432
XII-H-4 - Menu « Bluetooth subsystem support ».....	432
XII-H-4-a - Sous-menu « Bluetooth device drivers ».....	433
XII-H-5 - Option « RxRPC session socket ».....	434
XII-H-6 - Menu « Wireless ».....	434
XII-H-7 - Option « RF switch subsystem support ».....	435
XII-H-8 - Option « Plan 9 Resource Sharing Support (9P2000) (Experimental) ».....	435
XII-I - Device Drivers.....	436
XII-I-1 - Menu « Generic Driver Options ».....	436
XII-I-2 - Menu « Connector - unified userspace <-> kernelspace linker ».....	436
XII-I-3 - Menu « Memory Technology Devices (MTD) ».....	436
XII-I-4 - Menu « Parallel port support ».....	437
XII-I-5 - Menu « Plug and Play support ».....	437
XII-I-6 - Menu « Block devices ».....	438
XII-I-7 - Sous-menu « Misc devices ».....	440
XII-I-8 - Menu « ATA/ATAPI/MFM/RLL support ».....	441
XII-I-9 - Menu « SCSI device support ».....	443
XII-I-9-a - Sous-menu « SCSI Transport ».....	444
XII-I-9-b - Sous-menu « SCSI low-level drivers ».....	444
XII-I-9-c - Sous-menu « PCMCIA SCSI adapter support ».....	445
XII-I-10 - Menu « Serial ATA (prod) and Parallel ATA (experimental) drivers ».....	445
XII-I-11 - Menu « Multidevice support (RAID and LVM).....	445
XII-I-12 - Menu « Fusion MPT device support ».....	447
XII-I-13 - Menu « IEEE 1394 (FireWire) support (EXPERIMENTAL) ».....	447

XII-I-14 - Menu « I2O device support ».....	448
XII-I-15 - Configuration des interfaces réseau.....	449
XII-I-15-a - Sous-menu « ARCnet devices ».....	451
XII-I-15-b - Sous-menu « PHY device support ».....	451
XII-I-15-c - Sous-menu « Ethernet (10 or 100Mbit) ».....	452
XII-I-15-d - Sous-menu « Ethernet (1000 Mbit) ».....	452
XII-I-15-e - Sous-menu « Ethernet (10000 Mbit) ».....	452
XII-I-15-f - Sous-menu « Token Ring devices (depends on LLC=y) ».....	452
XII-I-15-g - Sous-menu « Wireless LAN ».....	452
XII-I-15-h - Sous-menu « USB Network Adapters ».....	452
XII-I-15-i - Sous-menu « PCMCIA network device support ».....	453
XII-I-15-j - Sous-menu « Wan interfaces ».....	453
XII-I-15-k - Sous-menu « ATM drivers ».....	453
XII-I-16 - Menu « ISDN subsystem ».....	454
XII-I-16-a - Ancienne interface ISDN4Linux.....	454
XII-I-16-a-i - Gestionnaires de périphériques ISDN4Linux.....	455
XII-I-16-b - Interface CAPI.....	455
XII-I-17 - Menu « Telephony Support ».....	455
XII-I-18 - Menu « Input device support ».....	456
XII-I-18-a - Sous-menu « Hardware I/O ports ».....	457
XII-I-19 - Menu « Character devices ».....	457
XII-I-19-a - Sous-menu « Serial drivers ».....	460
XII-I-19-b - Sous-menu « IPMI ».....	461
XII-I-19-c - Sous-menu « Watchdog cards ».....	461
XII-I-19-d - Sous-menu « PCMCIA character device support ».....	461
XII-I-19-e - Sous-menu « TPM Hardware support ».....	462
XII-I-20 - Sous-menu « I2C support ».....	462
XII-I-21 - Sous-menu « SPI support ».....	462
XII-I-22 - Sous-menu « Dallas's 1-wire bus ».....	462
XII-I-23 - Sous-menu « Power supply class support ».....	463
XII-I-24 - Sous-menu « Hardware Monitoring support ».....	463
XII-I-25 - Sous-menu « Multifunction device drivers ».....	463
XII-I-26 - Menu « Multimedia devices ».....	463
XII-I-26-a - Sous-menu « Video Capture Adapters ».....	463
XII-I-26-b - Sous-menu « Radio Adapters ».....	464
XII-I-27 - Menu « Graphics support ».....	464
XII-I-28 - Menu « Sound ».....	465
XII-I-29 - Menu « HID Devices ».....	466
XII-I-30 - Menu « USB support ».....	466
XII-I-31 - Menu « MMC/SD Card support ».....	468
XII-I-32 - Menu « LED devices ».....	468
XII-I-33 - Menu « InfiniBand support ».....	468
XII-I-34 - Menu « EDAC - error detection and reporting (RAS) (EXPERIMENTAL) ».....	468
XII-I-35 - Menu « Real Time Clock ».....	468
XII-I-36 - Menu « DMA Engine support ».....	469
XII-I-37 - Menu « Auxiliary Display support ».....	469
XII-I-38 - Menu « Virtualization ».....	469
XII-I-39 - Menu « Userspace I/O ».....	469
XII-I-40 - Menu « Linux hypervisor example code (NEW) ».....	469
XII-J - Menu « File systems ».....	469
XII-J-1 - Sous-menu « CDROM/DVD Filesystems ».....	471
XII-J-2 - Sous-menu « DOS/FAT/NT Filesystems ».....	471
XII-J-3 - Sous-menu « Pseudo filesystems ».....	472
XII-J-4 - Sous-menu « Miscellaneous filesystems ».....	472
XII-J-5 - Sous-menu « Network File Systems ».....	472
XII-J-6 - Sous-menu « Partition Types ».....	474
XII-J-7 - Sous-menu « Native Language Support ».....	474
XII-K - Menu « Instrumentation Support ».....	474

XII-L - Menu « Kernel hacking ».....	475
XII-M - Menu « Security options ».....	478
XII-N - Menu « Cryptographic options ».....	479
XII-O - Menu « Library routines ».....	479
XIII - Annexe B. Compilation et mise à jour des principaux composants du système.....	479
XIII-A - Compilation de make 3.81.....	479
XIII-B - Compilation des binutils 2.17.0.....	480
XIII-C - Compilation de la bibliothèque C 2.5.0.....	480
XIII-D - Compilation de OpenSSL.....	483
XIII-E - Compilation de MESA 6.5.2.....	484
XIII-F - Compilation de X.org 7.3.0.....	484
XIII-G - Compilation de Lesstif 0.95.0.....	487
XIII-H - Compilation de KDE 3.5.7.....	487
XIII-I - Compilation de Gnome 2.20.0.....	490
XIII-I-1 - Récupération des sources.....	490
XIII-I-2 - Compilation des bibliothèques générales.....	491
XIII-I-3 - Compilation des modules de base d'interfaçage avec les langages.....	493
XIII-I-4 - Compilation des bibliothèques de Gnome.....	494
XIII-I-5 - Compilation de Gnome.....	495
XIII-I-6 - Compilation des bindings de Gnome.....	496
XIII-I-7 - Compilation des outils de développements complémentaires.....	497
XIII-J - Compilation de Samba 3.0.23.....	497
XIV - Annexe C. Formulaire pour la création des lignes de mode de X.org.....	498
XV - Annexe D. GNU Free Documentation License.....	501
XVI - Annexe E. Licence de documentation libre GNU.....	506
XVII - Remarques de l'auteur.....	511
XVII-A - Au sujet de ce document.....	511
XVII-B - Copyright.....	511
XVII-C - Commentez.....	511

I - Introduction

Ce livre est un guide d'installation et d'administration du système d'exploitation GNU/Linux pour ordinateur de type PC.

L'objet de ce document est donc de donner les connaissances de base nécessaires à l'installation de Linux sur un ordinateur de particulier ou un petit serveur. Il est supposé que l'utilisateur a déjà utilisé un autre système d'exploitation, par exemple MS Windows. Cependant, aucune notion avancée d'informatique n'est nécessaire. Tout sera expliqué au fur et à mesure des besoins et, si nécessité est, des compléments d'information seront donnés pour permettre la compréhension des opérations à effectuer. Néanmoins, les notions qui seront abordées ici ne seront pas simples, et il est possible que la plupart des personnes qui n'ont pas une grande habitude de l'informatique aient quelques difficultés à les assimiler. Cela dit, à vaincre sans peine, on triomphe sans gloire, et l'installation de Linux vous procurera le plaisir d'apprendre.

I-A - Matériel, logiciel et système d'exploitation

Dans les grandes lignes, un ordinateur est généralement constitué :

- d'un ou plusieurs processeurs, qui effectuent tous les traitements de calcul ;
- de mémoire vive, où sont stockées les instructions que le ou les processeurs doivent exécuter, et les données qu'ils doivent manipuler ;
- d'unités de stockage de masse telles que les disques durs, permettant de stocker les données et les programmes de manière permanente (c'est-à-dire même après extinction de l'ordinateur) ;
- et d'une carte mère, sur laquelle on connecte tous ces éléments.

À ces composants de base s'ajoutent des composants annexes, que l'on appelle « périphériques ». Le clavier, la souris, les cartes graphiques, son et réseau en sont des exemples.

Ce matériel n'est toutefois généralement pas exploitable tel quel. Il faut pour cela utiliser des programmes dédiés, qui permettent de l'utiliser. L'ensemble de ces programmes constituent ce que l'on appelle le « système d'exploitation ». C'est au-dessus de ce système que viennent se placer les programmes utilisateurs, que l'on appelle également les « applications », ou « logiciels ».

La fonction première d'un système d'exploitation est donc de gérer l'ordinateur, et de permettre aux logiciels de s'exécuter. Toutefois, les systèmes d'exploitation fournissent généralement une abstraction du matériel pour les programmes, afin que ceux-ci n'aient pas à se soucier des spécificités du matériel sur lequel ils sont installés. Par exemple, il est possible de simuler certaines fonctions absentes, telles que les fonctions 3D des cartes graphiques modernes, ou encore une interface réseau sur un ordinateur qui n'en dispose pas. Ainsi, les programmes qui utilisent ces fonctionnalités peuvent s'exécuter correctement, mais avec des restrictions et des performances moindres. L'essentiel est donc, dans ce cas, de fournir une interface de programmation uniforme, quel que soit le matériel sous-jacent. Le système d'exploitation apparaît donc comme une machine virtuelle de haut niveau pour les programmes.

Il est évident que le système doit savoir se montrer le plus discret possible, car le but final n'est pas de gérer l'ordinateur, mais bel et bien de s'en servir. Toutefois, la mode actuelle est de faire des systèmes de plus en plus complexes, et de plus en plus visibles, et par conséquent de moins en moins discrets (suivez mon regard...). De nos jours, la notion de système d'exploitation englobe donc de plus en plus l'ensemble des logiciels permettant d'utiliser le matériel et les logiciels d'application courants. Mais bien entendu, il ne s'agit là que d'un abus de langage, dont le grand public n'a pas forcément pris conscience.

I-B - Qu'est-ce que Linux ?

Linux est le noyau d'un système d'exploitation libre de type Unix, écrit initialement par Linus Torvalds en 1991 et auquel un grand nombre de programmeurs ont contribué par Internet depuis. Les origines de tous les systèmes Unix remontent à la première version d'un système d'exploitation expérimental développé par Dennis Ritchie et Ken Thompson dans les laboratoires AT&T's Bell Laboratories en 1969. Ce système a avant tout été développé par des

programmeurs, pour des programmeurs, et reprenait un certain nombre de concepts qui avaient été développés auparavant pour le système d'exploitation Multics (abréviation de « Multiplexed Information and Computing Service »), dont le rôle était de fournir des services informatiques centralisés à un grand nombre de personnes (un peu comme le Minitel a tenté de le faire par la suite). Multics n'a jamais réellement vu le jour, en revanche, le système Unix initial a engendré un grand nombre d'autres systèmes plus ou moins compatibles. Récemment, les différents fournisseurs de systèmes Unix se sont accordés pour définir l'ensemble des fonctionnalités que tous les systèmes Unix doivent supporter, afin de résoudre les problèmes engendrés par les incompatibilités existantes entre ces différents systèmes. Le terme Unix est donc un terme générique pour représenter l'ensemble de tous ces systèmes, dont Linux fait partie. Pour l'anecdote, la dénomination Unix provient de la contraction de « Unics » (abréviation de « Uniplexed Information and Computing Service »), terme forgé ironiquement pour bien signaler qu'Unix était une version allégée de ce que Multics devait être.

Bien que compatible avec les dernières spécifications Unix, Linux ne contient pas une ligne du code source du système Unix original, ce qui en fait ce que l'on appelle un « clone ». Cela dit, il s'agit réellement d'un système Unix à part entière. En tant que tel, il dispose des fonctionnalités fournies par les systèmes Unix : il est multitâche, multiutilisateur et relativement orienté réseau. Vous aurez donc, avec Linux, un système fiable, fonctionnel et performant.

Comme nous l'avons dit, Linux n'est que le noyau d'un système d'exploitation. Ce n'est donc que le composant de base qui prend en charge toute la gestion du matériel. Mais quel est donc ce système d'exploitation ? Il s'agit du système GNU/Linux, comprenant donc, outre le noyau Linux, plusieurs autres couches logicielles développées par la Free Software Foundation et d'autres organisations. Ces couches prennent en charge différentes fonctionnalités, telles que l'utilisation de l'ordinateur en ligne de commande, l'affichage graphique, et la gestion complète de l'environnement utilisateur en mode graphique. Parler de Linux en tant que système d'exploitation est donc, encore une fois, un abus de langage. Cela étant dit, nous nous autoriserons à le faire dans la suite de ce document, par souci de simplicité.

Contrairement aux idées reçues, il existe un grand nombre d'applications pour Linux. La plupart de ces applications peuvent être installées avec le système GNU/Linux, ce qui fait qu'en pratique ce système forme un ensemble complet et parfaitement utilisable pour la plupart des tâches courantes.

Une installation correcte de Linux vous permettra donc de réaliser les opérations les plus classiques, comme effectuer un travail bureautique, naviguer sur Internet, réaliser l'acquisition, la capture et le retraitement d'images, réaliser des animations 3D ou encore programmer. En revanche, autant vous prévenir tout de suite : nombre de jeux ne sont tout simplement pas disponibles sous Linux, bien que les principaux titres soient régulièrement portés. De même, vous ne trouverez pas forcément tous les logiciels spécialisés qui existent sous Windows, et certains logiciels se contenteront des fonctionnalités fondamentales. Vous ne pourrez donc pas réaliser ce que vous faisiez avec ces applications Windows dont il n'existe pas encore d'équivalent sous Linux. C'est par exemple le cas pour les applications de gestion et de paie utilisées par nombre de professionnels indépendants ou par des PME.

Les systèmes Linux se présentent généralement sous la forme de « distributions », que l'on peut acheter dans le commerce ou télécharger sur Internet (de manière tout à fait légale, comme nous le verrons dans le chapitre suivant). Une distribution n'est rien d'autre que le regroupement de l'ensemble des programmes qui constituent le système d'exploitation et des logiciels les plus utiles et les plus connus pour Linux. Une distribution Linux est donc réellement bien plus qu'un système d'exploitation : c'est un tout qui vous permettra réellement d'utiliser complètement votre ordinateur, généralement sans même à avoir à installer de logiciels complémentaires ! De ce point de vue, Linux est beaucoup plus fonctionnel que les autres systèmes d'exploitation propriétaires, qui sont en pratique livrés « nus ».

I-C - Pourquoi Linux ?

Que les choses soient claires : l'installation de Linux peut être une opération relativement compliquée, et l'usage d'un système Unix en général n'est pas à la portée de tout le monde. Même si la qualité des distributions actuellement disponibles s'est grandement accrue ces derniers temps, au point que n'importe qui peut installer un système Linux viable sans trop de problèmes, la configuration du système pour obtenir un fonctionnement correct exige un travail assez important. En particulier, les distributions actuelles éprouvent encore quelques difficultés pour optimiser les périphériques exotiques, et souvent seules les fonctionnalités de base sont correctement configurées après une

installation classique. Par ailleurs, la plupart des applications sont développées par des groupes de programmeurs indépendants, et bien que ce soit justement le rôle des distributions de réaliser l'intégration de tous ces composants dans un environnement homogène, celle-ci n'est pas forcément parfaite. Les outils de configuration des distributions vous permettront sans doute de configurer votre système de base simplement, mais pour aller au-delà, il faudra sans doute intervenir manuellement.

Néanmoins, il faut reconnaître que celui qui installe Linux à partir d'une distribution sur un ordinateur assez vieux (c'est-à-dire un ordinateur qui ne dispose pas des derniers périphériques et cartes graphiques à la mode), ou dont les composants sont de marque courante, obtient rapidement un système fonctionnel et capable de réaliser la plupart des opérations qu'il désire. En particulier, celui qui utilise son ordinateur pour travailler (j'entends par là écrire des lettres, les imprimer, naviguer sur Internet pour récupérer des informations, ou programmer) peut parfaitement se contenter de l'installation par défaut. Ce type de situation ne convient pas à tout le monde : la plupart des gens disposent de cartes graphiques récentes (surtout depuis l'avènement des jeux 3D) ou de périphériques spécifiques. Tout le monde ne se place pas uniquement dans le cadre d'une utilisation professionnelle, et il est absurde de disposer d'une carte son et de ne pas pouvoir l'utiliser. Et c'est là que le bât blesse ! Si l'on désire que Linux reconnaisse ces matériels exotiques, il va falloir mettre les mains dans le cambouis et avoir une bonne dose de patience. Ce problème de configuration apparaît malheureusement principalement pour les particuliers, qui souvent disposent de machines hétéroclites et absolument non standards. Dans le cadre d'une entreprise, il existe des personnes qualifiées pour résoudre ce type de problème, mais ce sont des informaticiens et, de plus, les machines sont souvent homogènes, ce qui permet d'apporter des solutions génériques.

Il faut donc être informaticien ou amateur très éclairé pour installer Linux sur une machine de particulier et pour le configurer de manière optimale. La situation est d'autant plus grave que la plupart des gens ne connaissent pas Linux, et qu'il est toujours difficile d'apprendre et de prendre de nouvelles habitudes. Je veux dire par là que même une tâche très simple à réaliser peut prendre un certain temps, car tout simplement on ne l'a jamais faite. Celui qui a installé trois fois MS Windows sait parfaitement le faire à présent, et il pense que c'est relativement facile. Et pourtant, il réalise souvent des tâches d'une complexité qui dépasse, là aussi, le commun des mortels.

Heureusement, et c'est là la force de Linux, ces opérations ne doivent être effectuées qu'une seule fois. On n'a absolument pas besoin de changer la configuration à chaque instant, comme c'est le cas sous MS Windows, parce que le système est globalement beaucoup plus stable. Il ne plante quasiment jamais, les applications ne peuvent pas le corrompre, et sa qualité supprime le besoin permanent de mettre à jour une partie du système. En clair, quand on en a un qui fonctionne, on le garde, non pas parce que c'est un enfer à installer et à configurer, mais tout simplement parce que ce n'est pas nécessaire de le changer.

En résumé, on peut affirmer que :

- Linux est un système simple à installer sur des machines standards ;
- sa configuration sur une machine plus exotique requiert parfois une intervention manuelle ;
- dans la plupart des cas, cette intervention n'est pas très difficile à réaliser ;
- cependant, elle peut dérouter les personnes qui ne l'ont jamais effectuée ;
- mais le jeu en vaut la chandelle, parce que le système est réellement solide.

I-D - Plan du document

Ce document est structuré en neuf parties distinctes, qui correspondent essentiellement aux grandes étapes que vous suivrez pour installer et utiliser Linux.

Le **chapitre 2** a pour but de clarifier un peu les notions ayant trait aux logiciels libres. Elle tente d'expliquer pourquoi ces logiciels existent, et pourquoi ils font partie des meilleurs logiciels actuels.

Le **chapitre 3** décrit les concepts de base de la plupart des systèmes d'exploitation, et présente l'installation de trois distributions Linux représentatives de ce qui existe actuellement. À l'issue de cette partie, vous devez disposer d'un système fonctionnel, utilisable, mais non optimisé et ne permettant pas forcément d'utiliser tous vos périphériques.

Le [chapitre 4](#) présente un peu le système et vous aidera à faire vos premiers pas avec Linux, et vous donnera les notions de base qui vous permettront de mieux comprendre la philosophie du système.

Le [chapitre 5](#) constitue un petit cours d'Unix pour les nouveaux utilisateurs de ce type de système. Sa lecture en est fortement recommandée, du moins si vous voulez avoir un aperçu des possibilités de Linux et de l'ensemble de ses fonctionnalités.

Le [chapitre 6](#) traite des opérations d'administration et de maintenance de base des systèmes Unix. Cette partie est en quelque sorte la suite de la quatrième partie, où les notions plus spécifiques à l'administration de la machine sont décrites.

Le [chapitre 7](#) donne les notions de base sur les mécanismes de compilation et décrit la manière de faire pour compiler la dernière version de GCC, le compilateur C/C++ du projet GNU. Elle décrit également la technique à utiliser pour compiler et installer un nouveau noyau dans le système, opération indispensable pour obtenir un noyau optimisé qui « colle » à la machine.

Le [chapitre 8](#) présente la manière dont la configuration du matériel est réalisée sous Linux, ainsi que la manière dont le système fonctionne en interne. Elle vous permettra de configurer et d'optimiser votre système pour votre matériel.

Le [chapitre 9](#) traite de la configuration du réseau sous Linux. Le réseau est réellement l'un des aspects les plus importants de Linux, et nécessite donc une attention toute particulière.

Enfin, le [chapitre 10](#) et dernier chapitre vous décrit le fonctionnement et la procédure d'installation de XWindow, l'environnement graphique de Linux. Cet environnement est particulièrement riche et fonctionnel, et mérite donc que l'on s'y attarde également.

II - GNU, Linux et les logiciels libres

Vous entendrez souvent parler de la licence « GPL », du projet « GNU », de la « Free Software Foundation » et des logiciels libres dans le monde de Linux. Pour bien comprendre ce que sont les logiciels libres et la Free Software Foundation, et ce que signifie la licence GPL, il est nécessaire d'en faire une brève présentation.

II-A - Les logiciels libres et la Free Software Foundation

La Free Software Foundation est une organisation dont le but est de développer des logiciels libres. Le terme de « libre » signifie clairement que chacun peut faire ce qu'il veut du logiciel, y compris le modifier. La vente n'est absolument pas interdite, et il faut donc faire la distinction entre libre et gratuit. Cela étant dit, les logiciels libres sont souvent de facto gratuits, car ils sont librement redistribuables par quiconque en possède une copie.

La liberté de modifier les logiciels libres implique naturellement que leur code source, c'est-à-dire le texte de leur programme tel qu'il a été saisi par ses auteurs, soit librement accessible et modifiable. Les logiciels libres sont donc qualifiés de logiciels « Open Source », ce qui signifie en anglais que les sources du programme sont disponibles. Attention cependant, tous les logiciels Open Source ne sont pas forcément libres, car il n'est pas toujours possible de modifier ce code source et de le redistribuer librement (éventuellement gratuitement). Ainsi, nombre d'éditeurs de logiciels propriétaires publient leur code source sans pour autant donner de droits supplémentaires à ceux qui les lisent. Certains d'entre eux jouent d'ailleurs explicitement sur cette confusion. De plus, la plupart des journalistes anglo-saxons font cette confusion et, de ce fait, occultent tous les avantages des logiciels libres. Vous trouverez de plus amples informations sur la notion de code source dans le [Chapitre 7](#).

II-B - Droits d'auteurs et la licence GPL

Il faut bien comprendre que le fait de diffuser un logiciel sous une licence libre ne prive absolument pas son auteur de ses droits. Il en reste l'auteur et, en tant que tel, conserve les droits d'auteurs sur son travail. Il ne fait que concéder la

liberté d'exploiter ce travail aux autres. C'est en cela que les logiciels libres se démarquent du domaine public, dont les logiciels ne sont plus soumis à aucun droit.

Afin de protéger les logiciels libres et leurs auteurs, la Free Software Foundation a rédigé la licence GPL (abréviation de l'anglais « General Public License »). Cette licence stipule que le logiciel libre peut être redistribué, utilisé, modifié librement, pourvu que celui qui en bénéficie accorde les mêmes droits à ceux à qui il fournit les copies du logiciel, qu'il l'ait modifié ou non. En d'autres termes, elle garantit que la liberté des uns s'arrête là où commence celle des autres.

Cette licence empêche donc l'aliénation du logiciel et sa transformation en logiciel propriétaire, de quelque manière que ce soit. Cela implique que tout logiciel libre sous licence GPL modifié par une autre personne que son auteur reste libre, et le restera à jamais. Ainsi, il est impossible qu'une société commerciale puisse un jour s'approprier un logiciel libre, même si elle l'améliore. Si vous désirez lire la licence GPL, vous pouvez en trouver une copie dans le fichier `/usr/src/linux/COPYING` une fois que vous aurez installé Linux.

La FSF a également rédigé d'autres licences plus adaptées aux bibliothèques de programmes et aux documentations libres. Ainsi, la licence LGPL (« Lesser General Public License ») permet d'utiliser les bibliothèques de programmes dans des programmes propriétaires, et la licence FDL (« Free Documentation License ») permet de diffuser des documentations libres. À titre d'exemple, ce guide est distribué sous licence FDL, dont vous trouverez une traduction française [en annexe](#).

Précisons que la licence GPL n'est pas la seule licence permettant de distribuer des logiciels libres. Il existe d'autres licences, dont les termes sont à peu près similaires. Par exemple, la licence BSD (un autre système Unix libre) exige également la distribution des sources, mais permet l'appropriation des sources par des sociétés commerciales. De même, la licence X, sous laquelle est diffusé l'environnement graphique X11 qu'utilise Linux, est une licence libre. Quelques outils fournis avec Linux sont distribués avec d'autres licences plus rares.

II-C - Le projet GNU et Linux

La licence GPL a été écrite initialement pour le projet GNU de la Free Software Foundation, dont le but est de réaliser un système Unix libre et indépendant des Unix commerciaux. Précisons ici que le terme « Unix » caractérise un ensemble de systèmes d'exploitation, qui disposent tous à peu près des mêmes fonctionnalités et proposent d'y accéder de la même manière.

Le projet GNU est toujours en cours, puisque la Free Software Foundation a déjà écrit la plupart des utilitaires Unix, mais que le cœur du système (ce que l'on appelle le noyau) est toujours en cours de réalisation. Pour information, ce noyau se nomme « Hurd ».

Cependant, d'autres noyaux sont disponibles, avec lesquels les commandes GNU peuvent être utilisées. Parmi ces noyaux, il existe bien entendu Linux, qui a été écrit par le Finlandais Linus Torvalds lorsqu'il était étudiant, et amélioré par des programmeurs du monde entier sur Internet. Linux est un noyau parmi tant d'autres, à ceci près qu'il est, lui aussi, distribué sous la licence GPL, bien que n'ayant rien avoir avec la Free Software Foundation.

Cela signifie qu'il serait en fait plus exact de parler du système « GNU/Linux » que de « Linux » tout court. Sous cette dénomination, il est clair que ce système est constitué des outils GNU fonctionnant sur le noyau Linux. C'est donc un ensemble de logiciels libres provenant de plusieurs sources distinctes. Cependant, il est très courant d'entendre parler de « Linux » tout court, par abus de langage et par souci de simplicité. Bien entendu, cette dénomination est proscrite sur les sites Internet de la Free Software Foundation, qui devient très susceptible à ce sujet.

Pour information, le terme « GNU » est l'abréviation de l'anglais « GNU's Not Unix ». Cette curieuse phrase rappelle que le projet GNU est de réaliser un système Unix différent des autres. Vous remarquerez que cette définition est récursive, c'est-à-dire qu'elle utilise le mot « GNU » elle-même. Cela doit être attribué au goût des développeurs de la Free Software Foundation pour ce genre de définition infiniment récursive. Vous ne saurez sans doute jamais les raisons qui les ont poussés à choisir la lettre 'G' dans leur définition. Cela étant, « GNU » se prononce « gnou » en anglais, et vous trouverez donc souvent la représentation d'un gnou sur les sites Internet de GNU.

II-D - Avantages et inconvénients des logiciels libres

Les logiciels libres disposent d'avantages indéniables par rapport aux logiciels « propriétaires » ou « fermés ». Je vais tenter de donner ici une liste non exhaustive de ces avantages :

- Les programmes distribués sous licence libre ont souvent été écrits par des passionnés du domaine applicatif auquel ils appartiennent. Les logiciels libres disposent donc souvent des dernières fonctionnalités à la mode et sont donc généralement extrêmement compétitifs sur ce plan.
- Du fait du grand nombre possible d'intervenants sur les sources des logiciels libres, un grand nombre de possibilités techniques peuvent être explorées, et c'est souvent la meilleure qui est sélectionnée. C'est une forme de sélection naturelle de la meilleure solution. Ainsi, sur le long terme, les logiciels libres sont les plus efficaces en terme de performances.
- Toujours du fait du grand nombre d'intervenants, et surtout de par la possibilité de consulter et de modifier librement le code source, le cycle de détection/identification/correction des bogues est très court. Les logiciels libres sont donc parmi les plus fiables qui se font. On peut considérer qu'un logiciel libre utilisé par un grand nombre de personnes est virtuellement « sans bogue » connu, puisque si tel était le cas il serait immédiatement corrigé.
- La possibilité de repartir d'une base de source existante permet de réaliser des développements beaucoup plus rapidement que dans un modèle fermé. Les logiciels libres sont donc également ceux qui se développent le plus rapidement à coût fixe, et sont certainement les plus rentables en terme de coût global pour la collectivité.
- Afin de garantir l'interopérabilité entre les différents intervenants du monde du logiciel libre, chacun s'efforce à respecter les standards. Les logiciels libres sont donc les plus ouverts, non seulement en terme de code source, mais également au niveau des formats de fichiers et des protocoles de communication. Cela garantit une interopérabilité optimale et l'absence de mauvaise surprise.
- Professionnellement parlant, la disponibilité du code source fournit une garantie de fonctionnement que l'on ne peut pas retrouver ailleurs. En cas de problème, il est toujours possible de s'en sortir, éventuellement en recourant à des compétences externes pour adapter le logiciel à ses propres besoins.
- Enfin, la disponibilité du code source garantit une pérennité absolue du logiciel, ce qu'aucune société commerciale vendant des logiciels propriétaires ne peut ou ne veut faire.

Pour être honnête, il faut admettre que les logiciels libres ont également des inconvénients. La liste suivante en présente quelques-uns :

- La diversité des logiciels libres a également un revers. L'utilisateur peut avoir à choisir entre plusieurs logiciels, ce qui ne simplifie pas forcément l'apprentissage ou la communication entre les différents utilisateurs de logiciels libres. Prenez par exemple ce guide : il fait la présentation de l'installation de trois distributions Linux, qu'il a fallu choisir parmi les centaines de distributions existantes... De plus, ce n'est pas le seul document traitant du sujet de l'installation et de la configuration de Linux (mais c'est sans doute le meilleur, n'est-ce pas ?). Cela ne simplifie pas les choses pour l'utilisateur.
- La diversité des bibliothèques et des outils, ainsi que le nombre d'applications susceptibles de communiquer entre elles, implique une complexité accrue dans le travail d'intégration de tous ces logiciels. Les distributions s'assurent que les logiciels qu'elles fournissent fonctionnent bien ensemble, mais la redondance existe malgré tout et a un coût non négligeable finalement, aussi bien pour les distributions que les programmeurs et les éditeurs de logiciels. La description de l'installation de trois distributions dans ce document a également un coût pour son auteur (pfff !). Enfin, même l'utilisation simultanée de plusieurs logiciels peut amener à charger en mémoire de nombreuses bibliothèques ayant pourtant la même fonction, alourdissant le système inutilement.
- Certaines fonctions des logiciels ne seront pas forcément implémentées, si ses auteurs n'y voient pas d'intérêt. Si le logiciel est développé par une seule personne ou une petite équipe, ils peuvent ne pas en avoir les moyens financiers ou temporels. Toutefois, si un nombre suffisant d'utilisateurs la réclament, il est probable qu'une personne ayant les compétences nécessaires pour ajouter la fonctionnalité se manifeste. Mais il est également possible qu'un autre projet soit démarré, ajoutant encore une fois un élément à la complexité de l'écosystème des logiciels libres. Il n'est donc pas rare d'avoir plusieurs logiciels réalisant la même chose, mais qu'aucun ne soit complet !
- Du fait que les logiciels libres sont justement développés par des passionnés ou par des sociétés qui ne peuvent pas en tirer un bénéfice direct, certains domaines ne sont pas couverts par les logiciels libres.

Ainsi, vous ne trouverez pas toujours une alternative libre à un logiciel propriétaire, soit parce que le sujet n'intéresse personne, soit parce que les coûts de développement dépassent les capacités des organisations de développeurs de logiciels libres.

- Plus spécifiquement, le marché monopolistique de Windows est beaucoup plus grand que celui de Linux. De ce fait, même les éditeurs de logiciels propriétaires rechignent à faire l'effort du portage de leurs logiciels pour Unix/Linux. Ainsi, la logithèque pour les systèmes libres s'en trouve d'autant plus réduite. Cela est particulièrement vrai pour les jeux et les logiciels professionnels, et malheureusement également pour les pilotes de périphériques de certains constructeurs de matériel.
- Quand bien même les éditeurs de logiciels voudraient publier leurs logiciels sous licence libre, ils n'en ont pas toujours le droit, en raison d'accords de licence avec des tiers ou de brevets qu'ils utilisent, et parfois même en raison de la réglementation locale de certains pays (comme la France, depuis la loi DADVSI, pour tous les logiciels de lecture de films protégés par exemple). De même, les vendeurs de matériel ne peuvent pas toujours fournir de pilotes libres, ni même d'informations ou de spécifications sur le matériel.

Hélas, ces inconvénients sont parfois incontournables, car inhérents à la nature même des logiciels libres. Des reproches injustifiés sont parfois réalisés envers les auteurs des logiciels libres, alors qu'ils ne sont pas en mesure d'éviter ces problèmes. L'utilisateur doit donc également se faire une raison : après tout, il reçoit déjà beaucoup de la communauté des logiciels libres.

Mais, et c'est là l'essentiel, l'aspect le plus important des logiciels libres est le fait qu'ils garantissent la liberté des utilisateurs par rapport aux éditeurs de logiciels. Le respect des standards, l'ouverture des formats de documents et des protocoles de communication garantissent une interopérabilité absolue, qui permet ainsi à chacun de rester libre de ses choix pour sa solution informatique. Il n'est que trop courant de voir les éditeurs de logiciels enfermer leurs clients dans une dépendance vis-à-vis d'eux, simplement en leur faisant utiliser des produits fermés et inutilisables sans leur concours. Le pire est sans doute que cette dépendance est transitive (le fait pour un auteur d'utiliser un produit implique que ses lecteurs le possèdent également), durable (on ne peut faire de mise à jour que chez le même éditeur de logiciel), voire imposée par la loi (interdiction en France d'analyser les formats de documents protégés/chiffrés depuis la loi DADVSI par exemple).

II-E - La motivation des auteurs et le financement des logiciels libres

Bien que cela ne se situe pas au même niveau philosophique, la question de la motivation des auteurs de logiciel libres et, pour les entreprises, de leur financement, se pose également de manière récurrente. Il n'est en effet pas évident, en première analyse, de déterminer les raisons qui poussent un auteur ou une entreprise à rendre ainsi public son savoir-faire, au risque de se le faire tout simplement voler.

Pour ce qui est des auteurs bénévoles, la motivation provient généralement de l'amusement qu'ils ont à développer ces logiciels et à les partager avec la communauté du logiciel libre. Cette communauté, constituée de l'ensemble des auteurs et des utilisateurs des logiciels libres, leur apporte en général la reconnaissance, des rapports de bogues ou des idées d'amélioration de leur logiciel, des conseils ou même de l'aide sur des sujets qu'ils ne maîtrisent pas (traduction, nouvelles fonctionnalités, etc.).

De plus, en s'intégrant à la communauté du logiciel libre, les programmeurs amateurs peuvent également récupérer des bibliothèques de programmes ou des morceaux complets d'autres programmes, et ainsi voir leur logiciel progresser plus vite. Certains peuvent même se distinguer des autres et se voir approcher par une entreprise pour un emploi sur le logiciel qui leur servait à l'origine de passe-temps ! Il est toujours plus agréable de travailler sur quelque chose qui nous plaît...

Les sociétés quant à elles peuvent financer le développement des logiciels libres qu'elles éditent ou auxquels elles contribuent de plusieurs manières. Quelques sociétés vivent de manière dérivée des logiciels libres qu'elles développent (par vente de produits dérivés, de contrat de support, ou de services complémentaires). C'est notamment le cas des sociétés qui éditent des distributions Linux et des sociétés de service en logiciel libre. D'autres sociétés proposent une double licence, constituée d'une licence libre telle que la GPL pour bénéficier des avantages des logiciels libres, et d'une licence propriétaire, permettant d'obtenir une rémunération de la part des clients qui ne veulent pas se plier aux exigences de la GPL (obligation de redistribution des sources en particulier).

Enfin, de nombreuses sociétés contribuent à des logiciels libres tout simplement parce qu'elles en ont besoin. Il est en effet souvent préférable d'adapter un logiciel libre qui satisfait ses besoins à 80 % et de développer les 20 % restants, quitte à redistribuer les modifications et améliorations qui y sont apportées, que de redévelopper l'intégralité d'un logiciel équivalent ou d'en financer le développement par une société tiers. Le coût total de développement d'une solution complètement propriétaire est en effet généralement beaucoup plus élevé. On voit bien que dans ce cas, les développeurs de logiciels libres sont avant tout leurs propres utilisateurs...

Cela dit, il faut être clair sur ce sujet : le logiciel libre rapporte moins que le logiciel propriétaire, tout simplement parce qu'il n'est pas question de monopole ici et qu'on ne peut pas pressurer le client aussi facilement qu'avec une offre logicielle fermée.

Comme on le voit, le logiciel libre est très loin d'être l'apanage de quelques fanatiques, et les mauvaises langues qui considèrent ce modèle économique comme du communisme n'ont assurément rien compris. Bien au contraire, il s'agit là de capitaliser les développements et de réduire les coûts, deux objectifs fondamentaux dans une économie de marché et très souvent mal appliqués à l'informatique ! Quant à ceux qui prétendent que les logiciels libres constituent une forme de dumping dans le domaine informatique, ils devraient plutôt analyser la pertinence de ce modèle économique et voir si, finalement, ce n'est pas tout simplement un modèle plus rentable sur le long terme que le modèle propriétaire. Il suffit de considérer le temps consacré, et le coût probablement pharaonique, du développement de Windows Vista, pour constater que le rapport qualité/prix est loin d'être reluisant, même par rapport son aîné Windows XP...

III - Installation du système

Maintenant que nous avons fait connaissance avec Linux et les logiciels libres, nous allons pouvoir passer au vif du sujet, à savoir l'installation du système.

La configuration minimale supportée par Linux est un 386 avec au minimum 12 Mo de mémoire vive. Il va de soi qu'avec un tel foudre de guerre, vous ne pourrez pas aller bien loin, et une configuration minimale plus raisonnable serait plutôt un 486DX2 66MHz avec au moins 32Mo de mémoire pour une utilisation en mode texte. La quantité de mémoire est de loin le facteur le plus important, et si vous voulez utiliser l'environnement graphique X11, il faudra au minimum compter sur 64 Mo de mémoire vive. Si, de plus, vous désirez utiliser les environnements graphiques modernes tels que KDE ou Gnome, 128 Mo seront nécessaires. N'importe quelle machine moderne peut donc parfaitement faire fonctionner Linux de nos jours, et vous n'aurez donc aucun problème à l'installer. Vous pourrez même transformer un vieux 486 ou pentium en routeur ou en pare-feu avec partage de connexion à Internet par exemple.

Même si les distributions les plus modernes permettent une installation quasiment transparente pour l'utilisateur, il est bon de savoir ce qui se passe en arrière-plan. En effet, sauf dans le cas où l'on réalise l'installation sur un disque dur dédié, l'installation de Linux requiert des manipulations susceptibles de détruire les données du système déjà en place, qui est généralement Microsoft Windows. De plus, la bonne compréhension des opérations nécessite une connaissance minimale des notions de base relatives à Linux et aux systèmes d'exploitation en général.

Ce chapitre a donc pour but de présenter les concepts fondamentaux utilisés pendant l'installation du système d'exploitation, et notamment la notion de partitionnement de disque. Les prérequis à l'installation seront ensuite abordés, puis l'installation elle-même sera réalisée. Mais avant tout, il va falloir vous décider pour le choix d'une distribution parmi les nombreuses distributions Linux existantes...

III-A - Choix de la distribution

Comme vous le savez peut-être déjà, il existe un grand nombre de distributions Linux. Si l'une d'entre elles était réellement meilleure que toutes les autres, la question du choix ne se poserait pas, et il n'y aurait aucun problème. Mais en réalité, chaque distribution a son public, et le choix doit se faire en fonction de ses besoins et de ses désirs. Chose qui, évidemment, n'est pas forcément évidente quand on n'a jamais installé Linux !

Pour faire concis, et sans vouloir froisser les partisans de chaque distribution, on peut en première approche distinguer les distributions selon leur procédure d'installation. Si vous êtes un débutant total, il est probable que vous cherchiez à installer une distribution « grand public », telle que la Mandriva de la société éponyme, la Fedora Core soutenue par Redhat, ou la SuSE de Novell. D'autres préféreront des distributions « moins commerciales », plus simples et plus fiables, qui ne s'appuient pas forcément sur les toutes dernières avancées technologiques. Ce sera le cas avec les distributions Debian, Ubuntu ou Slackware par exemple. Enfin, les intégristes de la pire espèce s'orienteront vers des distributions « orientées source », dont le principe est essentiellement de reconstruire tout le système lors de l'installation. Ceux-ci seront sans doute ravis avec la distribution Gentoo ou, s'ils veulent réellement tout refaire à la main, avec la distribution LFS. Je pense que vous êtes le plus à même de choisir la distribution qui vous conviendra le mieux, en fonction de vos compétences ou de vos appréhensions...

En revanche, dans une catégorie donnée, il n'est a priori pas évident de faire ce choix. Heureusement, il existe souvent une solution simple et rapide pour faire son choix : c'est tout simplement d'essayer chacune de ces distributions ! Non, je ne suis pas fou, je ne vous demande pas d'installer cinq versions de Linux les unes après les autres, juste pour que vous puissiez choisir celle que vous garderez... Je parle ici de la possibilité d'utiliser les versions « Live » des distributions. Ces versions sont simplement des versions allégées et conçues pour tenir sur un CD amorçable. Il est donc possible de démarrer son ordinateur sur ces CD et, sans faire la moindre installation sur disque dur, d'évaluer une distribution. La plupart des distributions fournissent un CD Live (parfois même plusieurs, selon le niveau de fonctionnalité proposé). Vous pouvez télécharger sur Internet leurs fichiers ISO (format des images de CD et de DVD), à partir desquels tout bon logiciel de gravage peut graver un CD ou un DVD. Vous n'avez donc plus qu'à faire vos emplettes et tester...

En ce qui nous concerne, ce document a pour vocation de faire une description la plus générale possible de l'installation de Linux, et si possible la plus indépendante des différentes distributions existantes. Être exhaustif serait une perte de temps, et de toute façon la qualité de la documentation obtenue serait sans doute moindre que celle des sites dédiés à chaque distribution. Par conséquent, seules trois distributions seront présentées, ce qui n'est déjà pas mal.

Le choix de ces distributions ne préjuge bien sûr en rien de leur valeur, il a fallu tout simplement faire un choix. En pratique toutefois, ce choix a été orienté sur les différents systèmes d'installation existants. En effet, de nombreuses distributions sont dérivées d'autres distributions plus connues ou originelles, ce qui fait qu'il n'y a finalement pas autant de variantes de systèmes d'installation qu'il y a de distributions sur le marché.

Les trois distributions présentées dans ce document permettent donc de voir les systèmes d'installation les plus classiques, et il sera aisé d'adapter les procédures d'installation de ce document aux autres distributions de même nature. Ces distributions sont les suivantes :

- la distribution Mandriva, basée sur le système d'installation RPM, originellement développé par Redhat et également utilisé par SuSE ;
- la distribution Debian, basée sur le système d'installation apt, que de nombreuses autres distributions utilisent, par exemple les distributions Ubuntu et Knoppix ;
- et enfin la distribution Slackware, basée sur le système d'installation pkg, qui est moins populaire, mais qui reste d'une simplicité exemplaire.

III-B - Partitionnement et systèmes de fichiers

L'installation d'un système d'exploitation tel que Linux sur un PC touche aux structures de données fondamentales du disque dur et est de ce fait une opération très sensible. Il est donc nécessaire de connaître certaines notions de base afin de savoir ce que l'on est en train de faire. Cette section a donc pour but de vous présenter ce qu'est une partition et un système de fichiers, comment choisir un plan de partitionnement, et comment l'accès aux partitions se fait dans un système Linux.

III-B-1 - Notion de partition

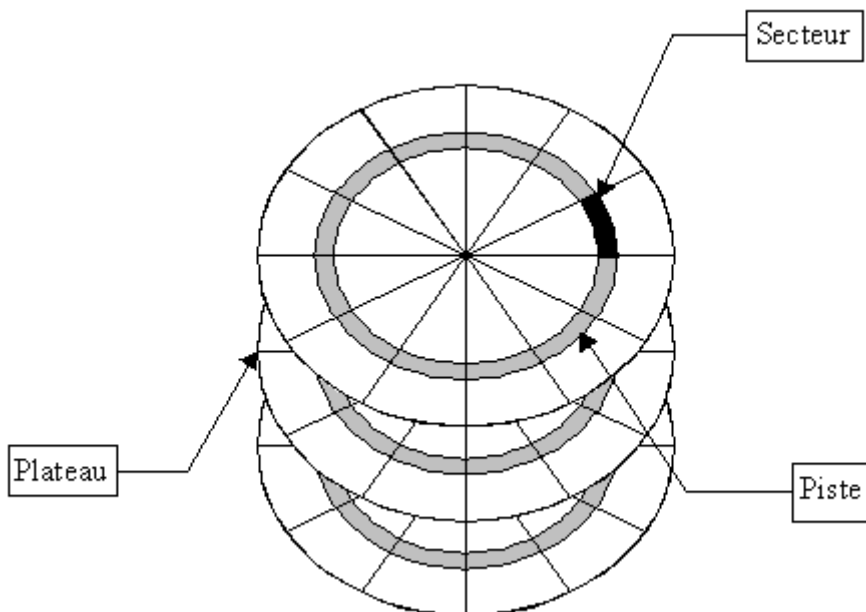
Une « partition » est, comme son nom l'indique, une partie d'un disque dur. Les partitions permettent de diviser l'espace de stockage des disques durs en zones indépendantes de taille restreinte. La notion de partition permet de réserver certaines portions du disque dur à un usage particulier, et de bien séparer les données qui se trouvent dans chaque partition. L'opération de « partitionnement » est l'opération de création des différentes partitions d'un disque dur.

L'installation de plusieurs systèmes d'exploitation nécessite souvent d'allouer une partition à chaque système, car les systèmes d'exploitation ne comprennent généralement pas le format des partitions des autres systèmes. Il est également parfois nécessaire, pour un même système, de définir plusieurs partitions, qui seront utilisées à des fins spécifiques. Par exemple, Linux fonctionne nettement mieux si on lui attribue une partition de « swap » (dite aussi « partition d'échange ») pour stocker des données peu utilisées qui se trouvent en mémoire, lorsqu'il a besoin de plus de mémoire qu'il n'en est physiquement installé sur la machine. De même, il est possible de créer plusieurs partitions pour séparer les données utilisateurs des programmes, ce qui permet de faciliter les mécanismes de sauvegarde d'une part, et d'assurer une plus grande sécurité des données lors des opérations de maintenance du système d'autre part.

Sur les machines de type PC, chaque disque dur peut être découpé en quatre partitions dites « primaires ». La position, la taille et le type de ces partitions sont enregistrés dans le premier secteur du disque dur, que l'on appelle souvent le « Master Boot Record » (« MBR » en abrégé). Le MBR ne contient que quatre entrées pour la définition des partitions, d'où la limite de quatre partitions primaires. Le type des partitions est un code numérique qui indique le système d'exploitation capable de l'utiliser et sa nature (partition de swap ou système de fichiers par exemple). À titre d'exemple, Linux utilise principalement deux types de partitions : les partitions de swap (numéro 82) et les partitions pour les systèmes de fichiers (type 83).

La définition des partitions se fait donc en donnant leur point de départ, leur taille et leur type. Le point de départ et la longueur des partitions sont exprimés en secteurs. Un « secteur » est l'unité de base pour les données des disques durs, qui correspond à un bloc de 512 octets utiles (auxquels s'ajoutent bien entendu d'éventuels octets de contrôle d'erreur, mais qui ne sont manipulés que par le disque dur lui-même et par son contrôleur, et que l'on ne peut donc pas utiliser pour y stocker des données). Cela dit, certains systèmes (ceux de Microsoft) ne permettent pas une telle finesse dans la définition des partitions et nécessitent de travailler au niveau du cylindre.

Note : Pour comprendre ce qu'est un cylindre, il faut savoir que les données des disques durs sont stockées sur les faces magnétiques de plateaux en rotation, au-dessus (et en dessous) desquelles les têtes de lecture/écriture du disque se déplacent radialement. Les données sont donc écrites en cercles concentriques sur les différents plateaux en raison de leur rotation sous les têtes de lecture (contrairement aux microsillons et aux CD, il s'agit bien ici de cercles et non d'une spirale, car les têtes de lecture/écriture restent à une position fixe pendant la rotation des plateaux). On appelle ces cercles des « pistes » (« track » en anglais). Chaque tête accède donc à une piste et une seule à un instant donné, sur laquelle les secteurs sont enregistrés. Comme toutes les têtes sont solidaires (elles se déplacent ensemble lorsque l'une d'entre elles doit changer de piste), les différentes pistes des différents plateaux sont accédées simultanément. L'ensemble de ces pistes, situées à un rayon donné pour tous les plateaux, constitue ce que l'on appelle un « cylindre ». Les paramètres des disques durs sont donc exprimés en termes de nombre de têtes, de cylindres et de secteurs par piste.



Pistes et secteurs d'un disque dur

Vous remarquerez souvent que le nombre de têtes est impair, alors qu'en général, un plateau a deux faces... Cela est dû au fait que les fabricants de disques durs conservent toujours une face d'un plateau pour y écrire des données de contrôle permettant aux têtes de lecture/écriture de se positionner ou pour y placer des pistes complémentaires en cas de zones défectueuses sur la surface de l'un des autres plateaux.

Bien entendu, la limitation à quatre partitions seulement est extrêmement contraignante, aussi la notion de partition étendue a-t-elle été introduite. Une « partition étendue » est une partition primaire spéciale, dans laquelle il est possible de définir jusqu'à 64 sous-partitions. Ces sous-partitions sont appelées des « partitions logiques ». Les données ne sont jamais stockées dans la partition étendue elle-même, mais dans ses partitions logiques. On ne peut définir qu'une seule partition étendue sur un disque donné, mais cela n'empêche pas d'avoir des partitions primaires normales à côté de celle-ci. Il est donc recommandé, lorsque l'on crée la quatrième partition, de créer une partition étendue et non une partition primaire, afin de se réserver la possibilité de créer de nouvelles partitions ultérieurement. Il faut toutefois savoir que certains systèmes ne peuvent pas être installés sur des partitions logiques (notamment DOS et Windows 9x/Millennium), bien qu'ils soient capables d'y accéder une fois qu'ils ont démarré.

TP	Partition primaire 1	Partition primaire 2	Partitions logiques		
			TP	Partition logique 2	Partition logique 2
					etc.

Partitions primaires et partitions logiques

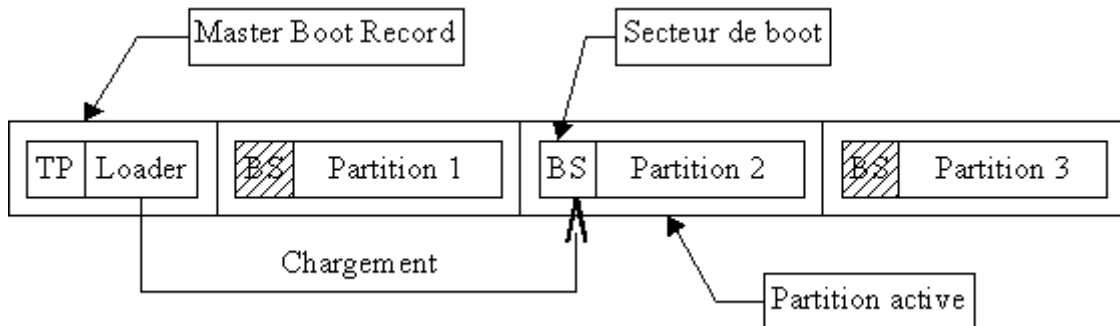
Outre la table des partitions primaires, le MBR contient un petit programme appelé le « bootstrap loader » qui permet de charger le premier secteur d'une des partitions primaires. Ce secteur est communément appelé le « secteur de boot », parce qu'il contient le programme capable de charger le système d'exploitation. La partition dont le secteur de boot est chargé par le bootstrap loader est appelée la « partition active ». Il ne peut y avoir qu'une seule partition active à chaque instant : celle du système d'exploitation principal.

Généralement, le programme stocké sur le secteur de boot d'une partition a pour but de charger le système d'exploitation qui y est installé. Cependant, pour certains systèmes d'exploitation, ce programme est très évolué et permet de lancer d'autres systèmes d'exploitation, éventuellement installés sur d'autres partitions ou d'autres disques durs. Ces programmes sont alors appelés des « gestionnaires d'amorçage ».

Linux dispose de deux gestionnaires d'amorçage très puissants : le GRUB et LILO. Windows NT, 2000, XP disposent également d'un gestionnaire d'amorçage capable de lancer d'autres systèmes d'exploitation : NTLDR. Windows Vista fournit son propre gestionnaire d'amorçage, bcdedit, qui est relativement difficile à utiliser.

Pour résumer, lors du démarrage d'un PC, le programme d'amorçage de la machine (communément appelé le « BIOS ») charge le MBR du premier disque dur en mémoire et exécute le bootstrap loader. Celui-ci cherche ensuite à charger le secteur de boot de la partition active, et exécute le gestionnaire d'amorçage qui s'y trouve. Ce gestionnaire peut donner accès aux différents systèmes d'exploitation, qu'ils soient situés sur d'autres partitions ou même d'autres disques durs.

La manière dont chaque système est lancé dépend ensuite du système. Il faut donc, en général, lancer chaque système d'exploitation avec son propre chargeur. Cependant, on peut toujours utiliser un gestionnaire d'amorçage d'un autre système en rusant quelque peu : il suffit d'indiquer à ce gestionnaire de charger le secteur de boot de la partition d'installation du système que l'on désire lancer si celui-ci n'est pas pris en charge directement. Dans ce cas, le gestionnaire d'amorçage ne fait que passer la main au chargeur de l'autre système.



Master boot record et secteurs de boot

III-B-2 - Notion de système de fichiers

Les systèmes d'exploitation utilisent généralement les partitions pour y stocker ce que l'on appelle des « systèmes de fichiers ». Nous allons voir à présent ce qu'est un système de fichiers, et comment ils sont manipulés sous Unix.

III-B-2-a - Définition

Un système de fichiers est une structure de données sur disque contenant des fichiers, dans lesquels applications peuvent y stocker leurs données. Les fichiers eux-mêmes sont des entités capables de contenir des données au sens large, ce peut être des documents (texte, image, film, son), des programmes, des données utilisées par le système ou tout autre type de données imaginable.

La plupart des systèmes de fichiers sont structurés hiérarchiquement, et regroupent les fichiers dans des répertoires. Cette organisation permet de classer les fichiers et de pouvoir les retrouver plus facilement. Il existe donc un répertoire racine, qui contient tous les fichiers soit directement, soit indirectement dans ses sous-répertoires. En fait, les répertoires sont eux-mêmes des fichiers spéciaux, interprétés par le système différemment des autres fichiers. Les données stockées dans les répertoires sont simplement les entrées de répertoires, qui caractérisent et permettent d'avoir accès aux autres fichiers et aux autres répertoires.

Il existe de nombreux types de systèmes de fichiers, pour chaque système d'exploitation. Les plus connus dans le monde Windows sont la FAT (système de fichier originel du DOS) et la FAT32, ainsi que NTFS (système de fichiers de Windows NT4, 2000, XP et Vista) et ISO9660 (système de fichiers des CD-ROM et DVD). Sous Linux, on retrouve, outre les systèmes de fichiers Windows, les systèmes Ext2 et Ext3, ReiserFS v3, XFS, JFS, etc. Chacun de ces systèmes de fichiers a ses avantages et ses inconvénients, aussi bien en termes de performances et de limites que de fiabilité face aux défaillances matérielles. En pratique, les systèmes de fichiers les plus utilisés sous Linux sont Ext3 et ReiserFS v3.

III-B-2-b - Nommage des fichiers

Afin de retrouver les fichiers, un nom leur est attribué. Il est possible, à partir du nom des différents sous-répertoires et du nom du fichier, de construire un « chemin d'accès », qui permet de référencer le fichier en question de manière bien précise et de pouvoir le retrouver facilement. Bien entendu, c'est du répertoire racine que débutent tous les chemins possibles dans le système de fichiers. De ce fait, le répertoire racine est le seul répertoire qui n'a pas de nom.

Dans les chemins d'accès aux fichiers, les noms de répertoires et de fichiers sont séparés par un caractère spécial. Ce caractère est traditionnellement, sous Unix, la barre oblique de division (nommée « slash » en anglais) : '/'. Les utilisateurs du DOS et de Windows prendront garde ici au fait que Microsoft a préféré la barre oblique inverse (nommée « backslash » en anglais) '\', rendant ainsi tous ses systèmes incompatibles avec les systèmes Unix, et générant ainsi beaucoup de problèmes supplémentaires là où il n'était pas nécessaire d'en avoir (sincèrement, le coût de cette ânerie, ainsi que celle des marqueurs de fin de ligne différents dans les fichiers textes, doit atteindre des sommes astronomiques dans tous les projets de portage ou de développement d'applications portables). Comme le répertoire racine n'a pas de nom, il peut être accédé directement avec un simple slash :

```
/
```

L'écriture d'un chemin d'accès se fait en écrivant les noms des différents répertoires et le nom du fichier, tous séparés par le séparateur. L'exemple suivant vous montre l'allure d'un chemin d'accès typique sous Unix :

```
/home/dupond.jean/lettres/professionnelles/marketing/ventes2005.odt
```

Le chemin de cet exemple référence le fichier « ventes2005.odt » de l'utilisateur « Jean Dupond ».

Comme on peut le voir dans cet exemple, le nom de fichier contient un suffixe « odt », séparé du corps du nom par un point. En effet, il est d'usage d'ajouter un tel suffixe aux noms des fichiers, afin de pouvoir le caractériser plus facilement. Ce suffixe est appelé « l'extension ». Dans le cas présent, l'extension « odt » indique qu'il s'agit d'un fichier « OpenOffice Document Text », donc qu'il s'agit d'un document texte de la suite bureautique OpenOffice.

Les utilisateurs de DOS et Windows devront faire attention aux points suivants lorsqu'ils utiliseront des fichiers sous Linux :

- Contrairement à DOS et à Windows, les systèmes Unix n'utilisent pas l'extension pour identifier la nature d'un fichier. Ainsi, il est tout à fait possible de stocker une image JPEG dans un document d'extension « .doc » : cela n'en fera pas pour autant un document Microsoft Word...
- L'extension n'ayant aucun rôle particulier, un nom de fichier peut parfaitement contenir un ou plusieurs points, et une extension peut être arbitrairement longue. En revanche, par convention, les fichiers dont le nom commence par un point sont des fichiers cachés. Dans ce cas, ils ne seront visibles que si on le demande explicitement.
- Les systèmes de fichiers Unix font la distinction entre les majuscules et les minuscules. Il faut donc prendre garde à la manière dont on écrit les noms de fichiers et de répertoires. Cependant, la plupart des gens n'utilisent que des minuscules pour leurs noms de fichiers.

Les systèmes de fichiers Unix n'ont pas les mêmes limitations sur les noms que les systèmes de fichiers FAT, FAT32 et ISO9660. Par exemple, les noms des fichiers et des répertoires peuvent être très longs (jusqu'à 256 caractères par nom), et la longueur maximale d'un chemin d'accès est de 4 ko dans le système de fichiers EXT2.

III-B-2-c - Montage des systèmes de fichiers

Contrairement à l'usage sous Windows, les chemins d'accès Unix ne comportent pas de spécification de lecteur. Les systèmes de fichiers Unix sont dits monotêtes, ce qui signifie qu'ils n'ont qu'un seul point de départ : le répertoire racine (alors que les systèmes Microsoft sont multitêtes, puisqu'ils ont un point de départ par lecteur et par partition). Le fait de n'avoir qu'un seul point de départ est beaucoup plus simple et permet, encore une fois, d'écrire les programmes plus simplement et donc avec moins de bogues potentiels.

Je sens tout de suite venir la question de la part des habitués du DOS : « Mais alors, comment spécifie-t-on le lecteur que l'on veut utiliser ? » Cette question a deux réponses. Premièrement, sous Unix, on n'accède pas aux lecteurs, mais aux systèmes de fichiers. Il faut donc bien comprendre qu'un lecteur représente un périphérique physique, et qu'il est possible qu'il contienne plusieurs systèmes de fichiers dans différentes partitions. De même, un système de fichiers n'est pas nécessairement stocké sur un lecteur ou dans une partition : il peut être également stocké dans un fichier (c'est le cas par exemple pour les images disques de CD-ROM), accessible par le réseau (c'est le cas des systèmes de fichiers réseau, « Network File System » en anglais), ou encore généré par un composant du système (c'est le cas des systèmes de fichiers virtuels du noyau).

Cependant, le problème de l'accès aux systèmes de fichiers se pose malgré tout. La réponse à ce problème-ci est cette fois la suivante : pour accéder à un système de fichiers, il faut réaliser une opération que l'on nomme le « montage ». Cette opération associe le répertoire racine de ce système de fichiers à l'un des répertoires de l'arborescence existante. Ce répertoire est couramment appelé « point de montage ». Les fichiers qui se trouvent dans le point de montage sont masqués par ceux du système de fichiers montés et ne sont donc plus accessibles, mais ils ne sont bien entendu pas détruits.

Par exemple, il est courant de monter le lecteur de disquette dans le répertoire `/floppy/`. Ainsi, si la disquette contient le fichier `ventes1999.odt`, ce fichier sera accessible grâce au chemin suivant :

```
/floppy/ventes1999.odt
```

Cette solution permet d'accéder à tous les systèmes de fichiers de la même manière, à partir d'un seul répertoire racine, que ces systèmes de fichiers soient EXT2, FAT, ISO9660, NTFS ou Amiga... En pratique, c'est nettement plus simple.

Note : L'opération de montage peut réaliser bien plus d'opérations qu'une simple association de la racine d'un système de fichiers à un point de montage. En effet, elle peut générer les opérations suivantes de manière implicite :

- chargement des pilotes pour ce système de fichiers ;
- allocation de tampons mémoire pour optimiser les entrées/sorties en mémoire.

On fera donc très attention à toujours démonter les systèmes de fichiers pour les lecteurs amovibles. En effet, Linux utilise en effet les tampons mémoire (« buffers » en anglais) pour y stocker les données des systèmes de fichiers montés, et il n'écrit ces données sur le support de stockage que lorsque c'est nécessaire. Ce mécanisme permet d'accélérer les lectures et les écritures sur les disques, mais a l'inconvénient de nécessiter une requête de vidange des tampons (opération que l'on appelle « sync ») avant de retirer le lecteur ou avant d'éteindre le système. Si on ne le fait pas, des données seront certainement perdues. Pour les systèmes de fichiers fixes, le système effectue ce sync lorsqu'on l'arrête proprement, mais il ne le fait pas si on coupe le courant brutalement. De manière similaire, Linux empêche l'éjection des CD-ROM tant qu'ils sont montés, mais il ne peut rien faire pour les lecteurs amovibles tels que les disquettes et les clefs ou disques externes USB. C'est donc à l'utilisateur de prendre garde à bien démonter les systèmes de fichiers situés sur ces périphériques avant de les débrancher.

III-B-2-d - Identification des partitions

Comme nous le verrons plus tard, sous Linux, la plupart des périphériques sont considérés comme des fichiers (la principale exception étant les interfaces réseau). Cela signifie qu'ils peuvent être manipulés via des fichiers spéciaux, sur lesquels les opérations classiques des fichiers peuvent être réalisées (et notamment la lecture et l'écriture). Cette règle s'appliquant aux disques durs, et même à leurs partitions, il est utile de préciser les conventions de nommage utilisées par Linux pour représenter ces entités.

Les fichiers spéciaux de périphériques sont par conventions tous situés dans le répertoire `/dev/`. C'est donc dans ce répertoire que l'on trouve les fichiers spéciaux de périphériques pour les disques et autres périphériques de stockage de masse. Malheureusement, leur nom dépend de leur interface de connexion, ce qui fait que l'on ne peut pas écrire de procédure générique pour leur manipulation.

Historiquement, les périphériques IDE sont accessibles via des fichiers spéciaux nommés avec des noms de la forme `hdX`, où 'X' est une lettre identifiant le disque sur le bus IDE. Par exemple, le lecteur maître du premier contrôleur IDE est accessible via le fichier spécial de périphérique `/dev/hda` (qu'il s'agisse d'un disque dur ou d'un lecteur de CD/DVD), tandis que le lecteur esclave de ce même contrôleur sera accessible via le fichier spécial de périphérique `/dev/hdb`. Les lecteurs maître et esclave du deuxième contrôleur IDE seront quant à eux accessibles via les fichiers spéciaux de périphérique `/dev/hdc` et `/dev/hdd`, et ainsi de suite.

Inversement, les périphériques SCSI sont accessibles via des fichiers spéciaux de périphériques dont le nom est de la forme `sdX`. On pourra donc accéder aux divers disques SCSI via les fichiers spéciaux de périphérique `/dev/sda`, `/dev/sdb`, etc.

Note : Afin d'uniformiser la manière dont les périphériques sont accédés, la tendance est de considérer les nouveaux périphériques comme des périphériques SCSI. De ce fait, la terminologie `sdX` est de plus en plus employée. Ainsi, les lecteurs de connectique Serial ATA, USB ou Firewire utilisent tous cette terminologie. De plus, un nouveau jeu de pilotes pour les périphériques IDE classiques est en cours de développement, et il utilise déjà cette terminologie, même si étymologiquement 's' signifie « SCSI ». Il n'y aura donc plus lieu de faire la distinction entre les interfaces utilisées sous peu.

Les fichiers spéciaux de périphérique des disques peuvent être utilisés directement, même si ce n'est pas l'usage de le faire. Par exemple, il est possible de lire la totalité du contenu d'un disque en lisant son fichier spécial de périphérique. En pratique toutefois, ces fichiers spéciaux de périphériques ne sont utilisés que pour manipuler la table des partitions. Ils ne sont généralement utilisés que par les programmes de partitionnement, donc que lors de l'installation du système ou lors de l'installation d'un nouveau disque. Par exemple, l'édition de la table des partitions du disque `/dev/hda` se fera avec la commande suivante sous Linux :

```
fdisk /dev/hda
```

Les partitions d'un disque quant à elle sont accessibles sous la terminologie `hdXn` ou `sdXn`, où 'X' est toujours la lettre du lecteur, et 'n' est le numéro de la partition considérée (la numérotation des partitions commence à partir de 1). Ainsi, la première partition du disque dur maître du premier contrôleur IDE (ouf !) est accessible via le fichier spécial de périphérique `/dev/hda1`, et ainsi de suite.

Ces fichiers spéciaux de périphériques peuvent également être utilisés directement, par exemple lorsque l'on veut copier une partition. Toutefois, en pratique, ces fichiers spéciaux sont surtout utilisés par les programmes permettant de créer les systèmes de fichiers. Par exemple, la création d'un système de fichiers FAT32 sur la partition `/dev/sda1` se fera avec la commande suivante sous Linux :

```
mkdosfs -F32 /dev/sda1
```

Note : Comme il l'a déjà été dit ci-dessus, il est tout à fait possible de créer un système de fichiers dans un fichier... Il suffit en effet de donner le nom de ce fichier en paramètre à la commande de création du système de fichiers, en lieu et place du nom du fichier spécial de périphérique de la partition ou du disque !

Certains utilitaires non spécifiques à Linux utilisent d'autres conventions de nommage pour identifier les disques et les partitions. C'est notamment le cas, comme on le verra plus loin, pour le gestionnaire d'amorçage GRUB. En effet, ce gestionnaire ne fait pas d'hypothèse sur le système utilisé, et ne peut donc pas reprendre la terminologie Linux.

III-B-3 - Choix du plan de partitionnement

Le partitionnement du disque peut généralement être réalisé automatiquement par les programmes d'installation des systèmes d'exploitation. Cependant, les choix faits par ces programmes ne sont pas toujours très judicieux, et peuvent ne pas convenir si votre disque contient des partitions déjà existantes. Nous verrons donc comment réaliser un partitionnement manuellement lors de l'installation de Linux.

Toutefois, avant de se lancer dans cette opération, il faut établir un plan de partitionnement. Cela consiste tout simplement à déterminer la taille et la nature de chaque partition dans le système. Il est normal, sur un système où seul Linux sera utilisé, de disposer d'au moins trois partitions :

- Une partition d'échange, que Linux utilisera pour y stocker temporairement des données lorsqu'il aura besoin de récupérer un peu de place en mémoire. Il est recommandé de placer cette partition au début du disque (c'est-à-dire au plus près du cylindre 0, là où le taux de transfert est le plus rapide). La taille de cette partition peut empiriquement être définie comme étant égale à deux fois la quantité de mémoire vive installée sur la machine, sans toutefois dépasser 512Mo. Cette limite part du principe que si le système a besoin de plus de mémoire virtuelle (somme de la quantité de mémoire vive et de la taille des zones d'échange), c'est que de toute façon il n'est pas dimensionné pour faire ce qu'on lui demande. Sachez qu'on peut rajouter des fichiers d'échange a posteriori très facilement sous Linux.
- Une partition pour le système de fichiers racine dans laquelle se trouvera l'ensemble des fichiers du système. Selon l'âge de la machine et la version du BIOS, il peut y avoir des limitations en ce qui concerne le début de cette partition. Il est recommandé qu'elle se trouve dans les 1024 premiers cylindres pour que le BIOS puisse y accéder et charger le gestionnaire d'amorçage du système. Sa taille devra être de 4 à 8Go, afin de pouvoir installer le système d'exploitation, les environnements graphiques et la plupart des applications sans problème. Il est inutile de dépasser les 8Go : si cette partition vient à se remplir, c'est que les fichiers temporaires du système, situés dans les répertoires `/tmp/` et `/var/`, prennent trop de place, auquel cas l'activité de la machine justifie l'utilisation d'une partition dédiée pour ces données.
- Une partition devant contenir les données des utilisateurs (qui se trouveront dans le répertoire `/home/`). Elle pourra prendre le reste de l'espace disponible sur le disque moins, bien entendu, l'espace nécessaire aux partitions des éventuels autres systèmes d'exploitation et celui d'une éventuelle partition de sauvegarde ou d'échange de données.

L'avantage d'avoir une partition séparée pour toutes les données des utilisateurs est considérable, puisque dans ce cas on peut mettre à jour le système ou le réinstaller complètement sans avoir à faire de sauvegarde de ses données. De plus, les fichiers de configuration importants peuvent être sauvegardés sur cette partition avant la réinstallation, ce qui est extrêmement pratique. Ce type de partitionnement est donc à prendre sérieusement en considération, surtout pour les machines de particuliers, sur lesquelles un grand nombre de programmes peuvent être installés simplement pour les tester. Il n'est donc pas rare, dans ces conditions, d'avoir à refaire une installation complète pour « nettoyer » rapidement le système.

Toutefois, si l'on ne dispose pas de beaucoup de place sur le disque, il est possible de regrouper la partition racine et la partition contenant les données utilisateurs. Un mauvais dimensionnement de ces partitions aura dans ce cas de moins lourdes conséquences. En effet, lorsqu'une partition est pleine, on ne peut pas facilement utiliser l'espace restant sur les autres partitions pour l'agrandir, car il faut déplacer toutes les données au préalable.

Si votre machine est destinée à accueillir plusieurs systèmes d'exploitation, il est peut-être intéressant de créer au moins une partition FAT ou FAT32. Cette partition permettra en effet d'échanger des données entre Linux et les autres systèmes d'exploitation, car les systèmes de fichiers FAT sont reconnus par tous les systèmes d'exploitation courants. Notez que si Windows NT4 doit être installé, vous devrez créer une partition FAT plutôt qu'une partition FAT32, car Windows NT ne reconnaît pas, sans programmes additionnels, les partitions FAT32. Ce problème ne se pose plus pour Windows 2000 et les suivants. Notez également que bien que Linux sache parfaitement lire les partitions NTFS (utilisées par Windows NT4, 2000, XP et Vista), l'écriture sur ces partitions n'est pas complètement implémentée (il n'est possible d'écrire que dans un fichier existant). Inversement, aucun système Windows ne sait et ne saura lire les systèmes de fichiers Linux, quels qu'ils soient. De même, les systèmes Windows ne savent pas accéder à un système de fichiers stocké dans un autre fichier, sans installer d'outils complémentaires (pour ceux qui disposent de Windows XP, l'outil [filedisk](#) sera sans doute relativement utile). Avoir une partition FAT est donc souvent la solution la plus simple pour échanger des informations entre les deux systèmes.

Note : Sachez cependant que la taille maximum des fichiers sur les partitions FAT32 est limitée à 4Go. Cette limitation est en dessous de la taille d'un DVD, cela pourra donc être assez gênant à l'occasion pour transférer de gros fichiers tels que des fichiers images de DVD ou des fichiers vidéo non compressés. De plus, Microsoft ne recommande pas que des systèmes de fichiers FAT de plus de 32 Go soient utilisés, même s'il est possible de créer de tels systèmes de fichiers avec des utilitaires spécifiques. Les systèmes de fichiers Linux n'ont pas ce genre de limites.

Dans ce cas également, vous devrez prévoir une ou deux partitions pour le deuxième système d'exploitation, afin de séparer les données de ce système et les données des utilisateurs. Vous aurez alors certainement à créer une partition étendue et des partitions logiques, pour éviter d'être limité aux quatre partitions primaires. On constate ici que le fait que les données des utilisateurs ne puissent pas être partagées entre les systèmes est très pénalisant...

Rien ne vous empêche de créer d'autres partitions si vous le désirez. Par exemple, si la machine doit être d'une fiabilité absolue ou si vous êtes soumis à des contraintes d'exploitation fortes, vous pouvez opter pour des solutions radicales qui consistent à séparer les données d'exploitation (normalement situées dans le répertoire `/var/`) des fichiers des programmes, et de les placer dans une partition dédiée. Vous pourrez alors ne monter que cette partition en lecture/écriture. Ainsi, en cas de crash système, seule la partition contenant les données d'exploitation devra être réparée, ou à l'extrême réinitialisée complètement par les scripts de démarrage.

Vous voyez que définir un plan de partitionnement n'est pas une chose facile, et il n'existe pas de solution générique qui convienne à tous les usages. Cela est d'autant plus vrai qu'il est impératif de bien déterminer ses besoins en espace disque, aussi bien pour les programmes que pour les données et le swap, puisque les partitions ne peuvent pas facilement être redimensionnées. Quoi qu'il en soit, dans tous vos choix, gardez à l'esprit que ce qui est le plus important pour un particulier, ce sont ses données.

III-C - Prérequis

Maintenant que les principes de base ont été vus, on peut passer à l'action. Mais avant de réaliser l'installation de Linux proprement dite, il faut s'attarder sur quelques prérequis importants : le recensement du matériel installé et bien entendu la sauvegarde des données existantes !

III-C-1 - Récupération des informations sur le matériel

Avant de commencer quoi que ce soit, vous devriez récupérer le maximum de données concernant votre matériel. En effet, ces informations peuvent être particulièrement utiles pour diagnostiquer la cause des éventuels problèmes de configuration du matériel que vous pourriez rencontrer. Vous aurez aussi peut-être à spécifier certains paramètres matériels lors de l'installation et, dans le pire des cas, lors du premier démarrage de Linux. Les informations dont vous aurez certainement besoin sont les suivantes :

- type de processeur, avec sa marque et éventuellement sa vitesse ;
- type de carte mère, avec sa marque et impérativement son chipset ;
- type de branchement de la souris (série, PS/2, USB, interface bus propriétaire) ;
- nombre de ports série, ainsi que leurs paramètres (ports d'entrée/sortie, lignes d'interruption) ;
- nombre de ports parallèles, ainsi que leurs paramètres (ports d'entrée/sortie, lignes d'interruption) ;
- le type de port USB (vitesse, compatible UHCI ou OHCI) ;
- si vous disposez d'un contrôleur SCSI, marque et modèle de ce contrôleur (modèle du chipset de la carte SCSI) ;
- sinon, nom du contrôleur de disque IDE (regardez sur le chipset de la carte mère !) ;
- types de disques durs (IDE, IDE UltraDMA 33, 66 ou 100, SCSI), ainsi que leur taille et leur position dans le système (contrôleurs auxquels ils sont connectés, numéros d'unités logiques SCSI) ;
- types de lecteurs ou de graveurs de CD-ROM ou de DVD (IDE, ATAPI, SCSI, connecté sur carte son) et le type de leur branchement s'ils sont externes (port parallèle, SCSI ou USB) ;
- nom, modèle et marque de la carte graphique, type de chipset utilisé par cette carte et taille de sa mémoire vidéo ;
- nom, modèle et marque de l'écran, avec, s'il s'agit d'un écran à tube cathodique, ses fréquences de balayage horizontal et vertical maximal, sa bande passante et les fréquences recommandées par le constructeur pour les différentes résolutions ;
- type de carte son (ISA, PCI, PnP) ainsi que le nom du chipset utilisé par cette carte ;
- type de carte réseau (ISA, PCI, PnP) ainsi que le nom de son chipset ;
- type de chipset pour les réseaux sans fils Wi-Fi ;
- type de modem (interne, externe) et, si le modem est interne, sa nature (modem complet ou émulation) ;
- type de chipset pour le contrôleur FireWire (IEEE 1394) si vous en disposez d'un ;

- type de chipset et nom, marque, modèle pour les périphériques annexes (contrôleurs infrarouges, lecteurs de cartes mémoire intégrés, etc.).

Vous pouvez obtenir ces informations en consultant la documentation fournie avec votre ordinateur, les sites Web des constructeurs des différents composants, les informations fournies par le programme de configuration du BIOS ou affichées directement par le BIOS au démarrage de la machine, ou tout simplement en ouvrant le boîtier de la machine et en regardant ce qu'il y a à l'intérieur. Vous pouvez également récupérer les informations indiquées dans la configuration de MS Windows si celui-ci est installé.

Bien entendu, il se peut que vous n'ayez pas besoin de ces informations et que l'installation se passe sans problème. Cependant, si d'aventure Linux exige que vous les connaissiez, mieux vaut pouvoir lui répondre.

Note : Bien que cela soit assez rare, certains composants ou périphériques additionnels peuvent ne pas être gérés par Linux, faute d'avoir un gestionnaire de périphériques adéquat (cet état de fait est en général dû à une rétention d'informations de la part des fabricants, qui empêchent ainsi les développeurs de Linux d'écrire les gestionnaires de périphériques dans de bonnes conditions). C'est en particulier le cas de nombre de périphériques bas de gamme conçus pour ne fonctionner qu'avec MS Windows, ou de quelques périphériques exotiques. Les Winmodems ont une triste réputation à ce sujet (il s'agit de modems incomplets, dont une partie du traitement de signal est reporté dans un logiciel spécifique à Windows), ainsi que des imprimantes dites « GDI », qui ne comprennent que les commandes graphiques de Windows. Vous pourrez également avoir des problèmes avec des cartes de numérisation vidéo analogiques ainsi que certaines cartes de décompression MPEG. Certaines marques se distinguent par le fait qu'elles refusent obstinément de développer des gestionnaires de périphériques pour leur matériel ou pire encore, de fournir les informations nécessaires aux développeurs de Linux pour écrire ces gestionnaires sous licence libre. Il est donc recommandé, comme toujours, de bien se renseigner auprès des vendeurs et des groupes de discussion avant tout achat de matériel... La [Liste de compatibilité Linux](#) vous permettra également d'effectuer une recherche sur n'importe quel type de matériel et sur n'importe quel modèle très facilement. Il existe d'autres sites réalisant périodiquement des tests sur le matériel récent, par exemple [LinuxHardware](#).

III-C-2 - Sauvegarde des données

L'installation d'un système d'exploitation est une opération délicate, surtout si l'on modifie les partitions, puisqu'il faut travailler au niveau le plus bas. La moindre erreur peut provoquer une catastrophe, ce qui peut au mieux vous obliger à recommencer complètement l'installation, et au pire à perdre l'ensemble de vos données, y compris celles des autres systèmes installés ! Par conséquent :

IL FAUT DONC FAIRE UNE SAUVEGARDE DE VOS DONNÉES !

Toutefois, si Linux est le premier système que vous installez sur votre ordinateur, ou si vous n'avez pas encore utilisé ou pas l'intention d'utiliser les autres systèmes d'exploitation installés dessus, vous pouvez tenter votre chance directement.

La manière de procéder pour réaliser une telle sauvegarde dépend bien entendu des systèmes d'exploitation déjà installés. Généralement, il s'agit de Microsoft Windows. Si vous disposez d'un lecteur de bande, le problème de la sauvegarde est très simple : vous n'avez qu'à faire une sauvegarde complète. Dans le cas contraire, vous allez sans doute devoir trier vos fichiers afin d'identifier ceux auxquels vous tenez réellement, et les recopier sur un support physique fiable (évités absolument les disquettes, ils sont lents et n'ont jamais été fiables). Vous pouvez par exemple faire un DVD de sauvegarde si vous disposez d'un graveur de DVD.

Dans ce cas, l'idéal est tout simplement de disposer d'un autre disque dur, que l'on pourra consacrer complètement à l'installation de Linux. Ainsi, vous pourrez expérimenter tout à loisir, sans craindre de perdre vos précieuses données. De plus, l'installation de Linux sur un disque vierge est nettement plus facile que sur un disque où Windows est déjà installé, parce que vous n'avez pas à dégager de la place pour son installation. Prenez toutefois garde au fait que Linux restera capable d'accéder aux données de votre premier disque dur, et que toute erreur de manipulation peut détruire les données qui s'y trouvent. Par exemple, si vous ne spécifiez pas le bon disque dur lors de la création des systèmes de fichiers, vous perdrez à nouveau toutes vos données. L'achat d'un tiroir peut être la vraie solution,

car vous n'aurez alors qu'un seul disque à un instant donné dans votre ordinateur. Toutefois, cette technique a l'inconvénient de ne pas vous permettre d'accéder aux données de votre disque Windows à partir de Linux.

III-D - Procédure d'installation générique

Si vous avez fait la sauvegarde de vos données, que vous savez comment partitionner vos disques et où installer Linux, vous pouvez vous lancer effectivement dans l'installation de Linux.

Cette section vous présentera de manière générique les étapes d'une installation de Linux. Si vous désirez installer une distribution Mandriva, Debian ou Slackware, vous pouvez passer directement aux sections suivantes, qui décrivent plus spécifiquement l'installation de Linux avec ces distributions.

De manière générale, une installation de Linux se déroule selon les étapes suivantes :

- amorçage du système ;
- création ou redimensionnement des partitions du disque dur ;
- création des systèmes de fichiers et de la partition d'échange ;
- installation du système proprement dite ;
- installation du gestionnaire d'amorçage ;
- configuration du système.

Nous allons donc voir ces différentes étapes, sauf la configuration du système, qui sera présentée dans toute la suite de ce document.

III-D-1 - Premier démarrage

Pour commencer votre installation, vous devez pouvoir démarrer votre ordinateur sous Linux. La méthode la plus simple est certainement de mettre le CD ou le DVD de la distribution que vous avez choisie dans votre lecteur et de redémarrer l'ordinateur.

Toutefois, il faut que votre BIOS soit configuré pour amorcer le système sur le lecteur de CD pour que cette solution fonctionne. Tous les BIOS modernes permettent de le faire, il suffit de rentrer dans le programme SETUP (généralement, cela se fait en appuyant sur la touche F2 ou la touche Suppr/Del de votre clavier juste au moment où l'ordinateur s'initialise).

Par ailleurs, certaines cartes mères et BIOS disposent d'une fonction de protection contre les virus de secteurs de boot. Les virus de ce type sont généralement assez anciens, et s'installent dans le MBR pour être exécutés à chaque démarrage de la machine et avant le système d'exploitation. Je prédis toutefois une recrudescence de ce type de virus depuis l'apparition des processeurs modernes qui permettent de virtualiser complètement le système d'exploitation. Quoi qu'il en soit, la modification des partitions et l'installation du gestionnaire d'amorçage de Linux peuvent apparaître comme une infection virale, et échouer en raison d'un blocage par le BIOS. Il est donc nécessaire de désactiver cette fonctionnalité pendant l'installation du système, et de la réactiver après. Cela se fait également dans le programme SETUP du BIOS.

Le noyau de Linux qui est fourni avec votre distribution est un noyau qui a été spécialement conçu pour démarrer correctement sur le plus grand nombre de machines possible. Il contient donc les pilotes (« drivers » en anglais) pour votre disque dur, ainsi que pour d'autres types de disques. Lorsqu'il se lance, Linux tente de détecter le matériel de votre ordinateur. Seuls les pilotes correspondant à votre matériel s'activent, les autres sont tout simplement ignorés.

Il se peut cependant que Linux ne détecte pas vos disques durs. Cela peut arriver surtout pour les disques SCSI. Dans ce cas, il est probable que votre distribution fournisse des noyaux alternatifs, pour les configurations plus exotiques. Chacun de ces noyaux permet de démarrer sur un certain type d'ordinateur : vous devrez donc trouver le noyau qui correspond à votre matériel et réessayer.

Une autre solution est de fournir, lors du démarrage, des options au noyau, pour qu'il utilise un pilote plutôt qu'un autre, ou qu'il ne tente pas de détecter un matériel qui n'est pas présent sur votre machine, si ce teste le bloque au démarrage. Pour trouver la liste des options et la manière de les fournir au noyau, vous devez lire les informations du gestionnaire d'amorçage de votre distribution.

Si votre machine est très vieille, il se peut que votre BIOS ne vous permette malheureusement pas de booter sur un lecteur de CD. Dans ce cas, vous pouvez tenter de le mettre à jour, mais s'il n'en existe pas de nouvelle version, il ne vous restera plus qu'à démarrer sur une disquette. Pour cela, vous devrez copier une image de disquette d'amorçage sur une disquette. Vous pourrez alors utiliser cette disquette pour démarrer l'installation. En général, les distributions fournissent un petit utilitaire DOS nommé RAWRITE.EXE. Cet utilitaire s'utilise sous DOS avec la ligne de commande suivante :

```
rawrite image lecteur:
```

où `image` est l'image d'une des disquettes d'amorçage (que vous trouverez sur le CD d'amorçage de votre distribution), et `lecteur` est le lecteur de disquette utilisé (en général, « A: »). Vous pouvez aussi consulter la documentation de votre distribution pour savoir comment indiquer au noyau les modules qu'il doit charger, ou comment créer une disquette de démarrage pour un autre noyau. Dans le pire des cas, la distribution utilisée ne supporte plus le démarrage sur disquette. Il ne reste plus alors qu'à trouver une version plus ancienne de cette distribution, l'installer, puis faire une mise à jour.

Avec la plupart des distributions, les outils nécessaires à l'installation sont placés sur un disque virtuel en mémoire après le démarrage de Linux. C'est en particulier le cas du programme d'installation, et des outils de manipulation des partitions et des systèmes de fichiers. Vous devez toutefois être en mesure de les utiliser : il est donc impératif que Linux ait détecté vos disques durs. Si ce n'est pas le cas, soit les fichiers spéciaux de périphériques de ces disques ne seront pas présent dans le répertoire `/dev/`, soit les outils vous signaleront une erreur lorsque vous tenterez de les utiliser.

III-D-2 - Trouver de la place pour installer le système

L'un des problèmes les plus courants lors d'une nouvelle installation est l'absence de place disponible sur le disque dur pour créer une partition Linux. Ce problème n'a hélas pas de solution : il faut supprimer une autre partition appartenant à un autre système, ou en réduire une.

Souvent, réduire une partition revient à sauvegarder les données, la supprimer, la recréer avec une taille inférieure et à restaurer les données. Certains outils commerciaux permettent de réduire des partitions directement, mais ils se vendent souvent pour le prix d'un disque dur neuf (je ne peux donc réellement pas conseiller de les acheter, sauf si vous avez plusieurs disques durs dont les partitions doivent être redimensionnées ou déplacées !). De plus, Linux dispose également d'un outil similaire qui, bien qu'il soit encore en phase de développement, permet déjà de réaliser la plupart des opérations que l'on peut avoir à faire sur les partitions FAT et les partitions EXT2 de Linux. Il s'agit du programme `parted`, que nous décrirons ci-après. De même, il existe un programme DOS permettant de réduire la taille d'une partition FAT, moyennant quelques précautions complémentaires. Bien entendu, il va de soi que si l'on doit déplacer ou modifier une partition, il faut faire au préalable des sauvegardes de toutes les données de cette partition... Mais ne l'ai-je pas déjà dit ?

Il est possible que votre distribution dispose d'un outil graphique permettant de manipuler les partitions existantes pendant la phase d'installation. Ce type d'outil pourra être utilisé pour dégager de la place pour Linux, éventuellement en réduisant une partition existante. Nous ne décrirons cependant pas ces outils ici, car ils sont spécifiques à chaque distribution. En revanche, nous présenterons les deux principales techniques permettant de redimensionner une partition, la première fonctionnant sous Linux et la deuxième sous DOS.

III-D-2-a - Utilisation de `parted`

L'utilitaire GNU `parted` est le standard en ce qui concerne les manipulations de partitions sous Linux. Cet outil s'utilise en ligne de commande, et peut donc être utilisé à partir d'un terminal en mode texte pendant la phase d'installation, si

vosre distribution l'inclut avec ses outils d'installation standards. Si ce n'est pas le cas, vous pourrez récupérer l'image d'une disquette de boot Linux contenant cet outil à l'adresse <ftp://ftp.gnu.org/gnu/parted/bootdisk/partboot.img> [<ftp://ftp.gnu.org/gnu/parted/bootdisk/partboot.img>]. Cette image pourra être copiée sur une disquette à l'aide de l'utilitaire DOS RAWRITE.EXE, de la même manière que les disquettes de boot de votre distribution peuvent être créées. La manière de procéder a été décrite dans la [Section 3.4.1](#).

Une fois que vous aurez démarré Linux et obtenu un terminal fonctionnel, vous pourrez lancer parted avec la simple commande suivante :

```
parted disque
```

où `disque` est l'identifiant du disque dur sur lequel la partition à modifier se trouve. Par exemple, la modification des partitions du premier disque dur IDE se fera donc à l'aide de la commande suivante :

```
parted /dev/hda
```

parted dispose de plusieurs commandes permettant de modifier les partitions. À l'heure actuelle, il ne permet réellement de travailler que sur les partitions FAT et les partitions contenant un système de fichiers EXT2, EXT3 ou ReiserFS. Pour les partitions contenant un système de fichiers NTFS, il vous faudra utiliser l'outil ntfsresize. Les exemples de la suite supposeront que le système de fichiers manipulés est un système de fichiers de type FAT.

La première commande indispensable est la commande print, qui permet d'afficher la table des partitions du disque courant. Les informations affichées par cette commande se présentent de la manière suivante :

```
Disk geometry for /dev/hda: 0.000-32634.492 megabytes
Disk label type: msdos
Minor  Start      End      Type      Filesystem  Flags
1       0.031  32632.031  primary   FAT         lba
```

Comme vous pouvez le constater, les partitions sont décrites à raison d'une ligne par partition. Cette ligne contient le numéro de la partition (les quatre premiers numéros étant affectés aux partitions primaires), les points de départ et de fin de ces partitions, exprimés en mégaoctets, le type de la partition, le système de fichiers de cette partition, et des indications complémentaires.

Note : Comme il l'a déjà été expliqué ci-dessus, les partitions sont généralement décrites en terme de cylindres. parted préfère utiliser le mégaoctet comme unité, ce qui est généralement plus clair. Il prend complètement en charge la traduction des informations de taille en termes plus physiques, en tenant compte des éventuels problèmes d'alignement aux limites de cylindre. Son utilisation est donc relativement directe.

La commande qui nous intéresse le plus ensuite est la commande resize, qui permet de redimensionner une partition. Cette commande utilise la syntaxe suivante :

```
resize partition début fin
```

où `partition` est le numéro de la partition tel qu'il est présenté dans la première colonne des informations de la commande print, `début` est la nouvelle position où la partition commencera et `fin` est la nouvelle limite de cette partition. Comme vous pouvez le constater, il est possible de réduire une partition aussi bien par son début que par sa fin ! La seule contrainte est, bien entendu, que cette partition reste suffisamment grande pour contenir l'ensemble de ses données. Par exemple, pour réduire de 8 Go la première partition du disque dur de l'exemple précédent afin d'y placer la partition de swap et les systèmes de fichiers, on utiliserait la commande suivante :

```
resize 1 8192 32632.031
```

Note : L'opération de redimensionnement peut prendre un certain temps. En effet, parted doit déplacer les données qui se trouveraient en dehors de la partition après son redimensionnement si aucune mesure spéciale n'était prise,

```
et il doit reconstruire complètement la FAT de cette partition pour qu'elle soit  
cohérente avec le nouvel emplacement de ces données.
```

Linux peut définir des limites sur les ressources utilisables par les utilisateurs afin de protéger le système. En particulier, il est possible qu'une limite soit définie concernant la taille maximale des fichiers manipulés. Il est bien entendu impératif de désactiver cette limite avant d'effectuer la moindre opération sur les partitions, étant donné que ce sont généralement des fichiers de très grande taille. Pour cela, il faut taper la commande suivante avant de lancer parted :

```
ulimit -f unlimited
```

Si vous désirez déplacer une partition plutôt que de la redimensionner, vous pouvez utiliser la commande move :

```
move partition début
```

où `partition` est le numéro de la partition et `début` est son nouvel emplacement de départ, exprimé en mégaoctets. De même, si vous désirez copier une partition, la commande `cp` devra être utilisée. Cette commande suit la syntaxe suivante :

```
cp [disque] source destination
```

où `disque` est le disque dur où se trouve la partition source, `source` est le numéro de cette partition, et `destination` est le numéro de la partition destination. La partition destination est toujours située sur le disque dur courant (c'est-à-dire celui qui a été indiqué en ligne de commande lors du lancement de parted ou celui spécifié par la commande `select` de parted).

Enfin, une fois que toutes les manipulations sur les partitions auront été effectuées, vous pourrez quitter parted avec la simple commande `quit`. Vous pourrez obtenir la liste des autres commandes acceptées par parted en tapant la commande `help`.

III-D-2-b - Utilisation de fips

fips (abréviation de l'anglais « First Interactive Partition Splitter ») est un utilitaire similaire à parted, à ceci près qu'il fonctionne sous DOS et qu'il ne permet que de réduire la limite supérieure d'une partition FAT. De plus, cet utilitaire est incapable de réorganiser le système de fichiers à l'issue de la réduction de la taille de la partition. Il est donc nécessaire de défragmenter le système de fichiers avant d'utiliser ce programme, afin de placer toutes ses données au début de la partition.

Note : Vérifiez bien les options du défragmenteur de système de fichiers que vous utilisez : quelques outils consolident bien l'espace libre, mais placent certains fichiers à la fin de la partition FAT pour laisser plus de place aux fichiers les plus utilisés au début du disque, afin d'optimiser le débit de données sur ces fichiers (c'est notamment le cas avec le défragmenteur de Norton). Il faut impérativement désactiver ce type d'option avant de réduire la partition, faute de quoi vous perdriez définitivement les fichiers qui se trouvent à la fin de la partition et votre FAT serait dans un état incohérent.

Une fois la défragmentation réalisée, fips peut être utilisé pour réduire la taille de la partition FAT. La plupart des distributions de Linux fournissent des utilitaires DOS sur leur CD/DVD d'installation, généralement dans le répertoire `dosutils`. C'est là que vous pourrez sans doute trouver fips. Attention, vous devrez impérativement utiliser la version 2.0 de fips pour manipuler les FAT32 et FAT32X.

La réduction de la taille d'une partition se fait en modifiant la variable qui contient la taille de la partition dans la table des partitions. Pour cela, vous devrez simplement lancer fips. Celui-ci vous présentera alors la liste des disques durs installés sur votre système, et vous devrez lui indiquer le disque sur lequel la partition à réduire se trouve. Il vous demandera ensuite la partition que vous désirez réduire, puis le cylindre auquel cette partition devra se terminer. Lorsque vous aurez déterminé la nouvelle taille de cette partition, vous devrez presser la touche 'c' pour poursuivre.

fips vous demandera alors confirmation avant d'écrire sur disque les nouvelles informations de partition. Si vous êtes sûr de vous, vous pouvez répondre par l'affirmative en pressant la touche 'y'.

Note : Contrairement à parted, fips ne reconstruit pas la table d'allocation des fichiers (la « FAT ») après avoir réduit la taille de la partition, ce qui fait que cette dernière est trop grosse pour cette nouvelle taille après réduction. Cela n'est pas gênant, seuls quelques mégaoctets seront perdus sur la partition FAT dans la FAT elle-même. Cette technique a en revanche l'avantage d'être extrêmement rapide.

III-D-3 - Partitionnement du disque

Le partitionnement en soi peut se faire soit directement à l'aide du fdisk de Linux, soit par l'intermédiaire du programme d'installation de la distribution correspondante. Il est recommandé d'utiliser ce programme d'installation, qui vous guidera et vous indiquera comment réaliser cette opération. Si toutefois vous désirez utiliser fdisk, il vaut mieux faire attention à ce que vous faites. Pour lancer fdisk, il suffit de taper la commande suivante en ligne de commande :

```
fdisk disque
```

où `disque` est le fichier spécial de périphérique représentant le disque que vous désirez partitionner. Si vous voulez partitionner le disque maître du premier contrôleur IDE, vous devrez donc taper :

```
fdisk /dev/hda
```

Si vous ne spécifiez aucun disque en paramètre à fdisk, il prendra par défaut le disque `/dev/sda`, ou `/dev/hda` si aucun disque SCSI n'est installé.

fdisk est un programme très peu interactif. Il attend que vous lui communiquiez les commandes à exécuter en tapant sur une lettre. Les différentes commandes possibles peuvent être affichées avec la commande 'm'.

Lorsque vous créez une partition, vous devez utiliser la commande 'n', puis indiquer son type avec les commandes 'p' (pour « primary ») pour une partition primaire ou 'e' (pour « extended ») pour une partition étendue. Vous devrez ensuite donner son numéro dans la table des partitions, puis indiquer le début et la fin de la partition. Par défaut, l'unité utilisée par fdisk est le cylindre. Il est recommandé de conserver cette unité, surtout si l'on utilise un système qui ne sait manipuler que les cylindres. Toutefois, on peut changer cette unité grâce à la commande 'u' et utiliser le secteur comme unité.

Si vous avez créé une partition étendue, celle-ci sera utilisée pour y stocker des partitions logiques. Pour pouvoir les créer, il faut encore utiliser la commande 'n', et choisir le type de partition logique avec la commande 'l' (pour « logical »). Les partitions logiques sont numérotées avec les nombres 5 et suivants. La création des partitions logiques se fait exactement de la même manière que les partitions primaires, en spécifiant leur début et leur fin, soit en cylindres, soit en secteurs selon l'unité courante.

Une fois les partitions créées, vous pouvez spécifier leur type à l'aide de la commande 't' (pour « type »). Cette commande demande successivement le numéro de la partition à modifier et la valeur de son identificateur en hexadécimal. Rappelons que les identificateurs à utiliser pour Linux sont 83 pour les partitions de systèmes de fichiers Linux, et 82 pour les partitions de swap. La liste des valeurs admissibles peut être obtenue à l'aide de la commande 'l'. Par défaut, le fdisk de Linux crée des partitions Linux natives, de code 83.

Lorsque vous aurez complètement défini vos partitions, il ne vous restera plus qu'à activer la partition qui contiendra le gestionnaire d'amorçage. La sélection de la partition active se fait avec la commande 'a' de fdisk. C'est donc sur cette partition que le chargeur du MBR ira chercher le gestionnaire d'amorçage du système à lancer.

Note : Théoriquement, il est tout à fait possible d'installer le gestionnaire d'amorçage d'un système directement sur le MBR, mais procéder de cette manière est très déconseillé. En effet, certains systèmes d'exploitation (notamment tous les systèmes de Microsoft) écrasent systématiquement le MBR lorsqu'ils s'installent, détruisant ainsi le chargeur d'un autre système qui y serait éventuellement installé. Cela implique que si l'on désire installer un gestionnaire

d'amorçage autre que celui des systèmes Microsoft sur le MBR, il faut le faire après l'installation de ces systèmes. En pratique, cela veut dire que dans ce cas, on doit installer Linux après Windows ou le DOS.

Notez qu'il n'est toutefois pas toujours faisable d'installer le gestionnaire d'amorçage sur le secteur de boot de la partition de son système, en particulier si cette partition ne se trouve pas sur le premier disque dur de la machine. En effet, certains des BIOS sont incapables d'utiliser les MBR des autres disques durs. Dans ce cas, on peut soit créer une partition de démarrage de petite taille (quelques mégaoctets, un cylindre au maximum) au début du disque et sur laquelle on installera le gestionnaire d'amorçage et éventuellement quelques outils de réparation en cas de coup dur, soit installer le gestionnaire d'amorçage directement sur le MBR du premier disque dur. Dans ce cas, on devra faire particulièrement attention à l'ordre d'installation des systèmes d'exploitation. De manière générale, il faut toujours installer les systèmes Microsoft en premier (respectivement dans l'ordre suivant si l'on veut éviter les problèmes : DOS, Windows 9x/Millennium et Windows NT4/2000/XP). Nous verrons plus loin comment installer le gestionnaire d'amorçage de Linux et faire une configuration multiboot avec les principaux autres systèmes d'exploitation existants.

III-D-4 - Création des systèmes de fichiers

Une fois le disque correctement partitionné, il faut créer les systèmes de fichiers. Cette opération n'est pas nécessaire pour les partitions de swap, cependant il faut le faire pour les autres partitions.

Note : Attention ! Dans le monde du DOS et de Windows, l'opération de création d'un système de fichiers est appelée le « formatage » d'une partition. C'est une erreur de langage relativement grave, car il n'y a strictement rien à voir entre le formatage d'un disque, qui est l'opération consistant à enregistrer des marques sur le support magnétique pour définir les pistes et les secteurs du disque, et la création du système de fichiers d'une partition, qui consiste à enregistrer les structures de données permettant de retrouver les fichiers dans cette partition. Le formatage est normalement effectué par le fabricant du disque, et a lieu avant le partitionnement. En effet, partitionner un disque suppose qu'il existe déjà des pistes et des secteurs sur le disque. C'est pour cette raison que l'on ne parlera ici que de création de systèmes de fichiers.

La création des systèmes de fichiers EXT2 et EXT3 se fait avec une même commande, à savoir la commande `mke2fs`. La création des autres systèmes de fichiers se fait généralement de la même manière que pour ces systèmes, mais avec des outils différents. Par exemple, la création d'un système de fichiers ReiserFS se fait à l'aide de l'outil `mkreiserfs`. ReiserFS constitue une alternative possible à EXT3, car, tout comme EXT3, il prend en charge les mécanismes de journalisation qui assurent la cohérence du système de fichiers même en cas d'arrêt intempestif de l'ordinateur. Cependant, ce système de fichiers est encore jeune et n'a certainement pas été débogué autant que EXT3, qui s'appuie sur la base de code du vieux système de fichiers EXT2. Sachez également que ReiserFS, contrairement à EXT2 et à EXT3, ne se comporte pas encore très bien si le disque dur a des secteurs physiques défectueux. La suite de cette section ne traitera que des systèmes de fichiers EXT2 et EXT3, consultez la documentation de votre distribution et les pages de manuel des outils des autres systèmes de fichiers pour plus de détails sur la méthode à utiliser pour les créer.

Afin de pouvoir utiliser `mke2fs` correctement, il est nécessaire de définir quelques termes, et d'expliquer à quelles notions d'EXT2 et d'EXT3 ils se réfèrent.

Premièrement, le système de fichiers EXT2 travaille, comme la plupart des systèmes de fichiers (ce n'est pas le cas des systèmes de fichiers Reiser par exemple), avec des blocs de taille fixe (« clusters » en anglais). Cela signifie que l'allocation de l'espace disque se fait par multiples de la taille de ces blocs : il est impossible de demander seulement une partie d'un bloc. Cette technique présente des avantages et des inconvénients. Essentiellement, l'avantage est la rapidité engendrée par la simplification des mécanismes d'allocation et de libération d'espace disque. L'inconvénient majeur est évidemment qu'on perd de la place pour tous les fichiers qui ne tiennent pas dans un nombre entier de blocs, puisqu'il faut allouer un bloc supplémentaire qui sera partiellement utilisé. En moyenne, on perd la moitié d'un bloc par fichier, ce qui ne peut être réduit qu'en limitant la taille des blocs à une valeur relativement faible.

Note : Les systèmes de fichiers Unix gèrent l'allocation et la libération des blocs de manière à toujours trouver le meilleur bloc à allouer pour créer un fichier. Ainsi, ils limitent la fragmentation des fichiers à son strict minimum, ce qui rend inutiles les programmes de défragmentation de systèmes de fichiers. De ce fait, la question de la

défragmentation, souvent posée par les nouveaux utilisateurs de Linux, ne se pose pas : en pratique, les systèmes de fichiers Unix ne se fragmentent pas (si, si, c'est vrai).

Deuxièmement, les systèmes de fichiers Unix utilisent des structures de données appelées « inodes » pour définir les fichiers. Un inode contient la plupart des informations d'un fichier, à savoir :

- son propriétaire et le groupe d'utilisateurs de son propriétaire ;
- ses droits d'accès ;
- ses dates de création, modification, accès ;
- les blocs qu'il utilise ;
- d'autres informations utilisées en interne.

Dans le cas d'EXT2 et EXT3, ces inodes sont stockés dans une table du système de fichiers, ce qui permet d'accéder très rapidement à toutes ces informations et de retrouver également très simplement le ou les blocs contenant les données du fichier. Le problème est ici que cette table a un nombre d'entrées limité, ce qui implique un nombre limité de fichiers dans le système de fichiers. Plus cette table est grande, plus le nombre de fichiers que l'on pourra créer sera grand, et inversement. Il faut donc trouver un compromis entre la taille de cette table et le nombre de fichiers que l'on est susceptible de créer. Il va de soi qu'en général, les grandes partitions contiennent plus de fichiers, mais que la table d'inodes peut également avoir une taille supérieure sans que cela soit déroutant. Par conséquent, il est relativement courant de définir le taux d'inode par bloc ou, autrement dit, la proportion d'inodes dans la partition par rapport à sa taille.

Toutes ces informations (blocs libres et inodes) sont sauvegardées à plusieurs endroits dans la partition, ce qui permet de disposer en permanence de copies de la structure du système de fichiers. De cette manière, il est relativement simple de réparer un système de fichiers endommagé, même si les données sont détruites en raison d'une erreur matérielle (secteurs défectueux sur le disque dur par exemple). Chacune de ces copies s'appelle un groupe de blocs. Chaque groupe de blocs contient un bloc particulier, le « super bloc », qui contient la description de son groupe.

Lors de la création d'un système de fichiers EXT2 ou EXT3, il est nécessaire d'indiquer la taille d'un bloc en octets. Cette taille doit impérativement être un multiple de la taille d'un secteur du support physique de données, parce que les blocs ne peuvent contenir qu'un nombre entier de secteurs. Pour un disque dur, la taille des secteurs est fixée à 512 octets, ce qui fait que la taille d'un bloc est au moins de 512 octets. De même, il faut spécifier le nombre d'inodes de la partition. Il est possible de spécifier ce nombre soit directement, ou d'indiquer seulement le nombre d'octets de la partition par inode. Le nombre total d'inodes utilisé sera alors calculé à partir de ce nombre d'octets et de la taille de la partition. Bien entendu, le nombre maximal d'inodes possibles est le nombre total de blocs, puisque tout fichier non vide requiert au moins un bloc et que chaque inode caractérise un fichier. Si vous ne savez pas quelles valeurs prendre, vous pouvez utiliser des blocs de 1024 octets (2 secteurs), et un rapport de 4096 octets par inode (donc de 4 blocs de 1 Ko par inode).

La plupart des systèmes de fichiers Unix, comme EXT3 (mais pas EXT2), gèrent à présent, en plus de tout ce que nous avons vu précédemment, un journal contenant les opérations à réaliser de manière transactionnelle sur le disque dur. Les opérations du journal sont exécutées de telle sorte que la structure du système de fichiers reste cohérente en toutes circonstances. Cela implique que le système de fichiers est toujours valide, même si une panne de courant se produit pendant une opération disque.

Voyons à présent la syntaxe de la commande `mke2fs` :

```
mke2fs [-j] fichier
```

`fichier` est le nom du fichier spécial de périphérique représentant la partition sur laquelle le système de fichiers doit être créé. Ce nom est le nom du disque dur, suffixé du numéro de la partition. Les numéros de partitions commencent à 0, si bien que la première partition du premier disque dur IDE sera référencée par le chemin `/dev/hda0`. L'option `-j` quant à elle est facultative. Lorsqu'elle est utilisée, le système de fichiers créé est un système de fichiers EXT3. Il est recommandé d'utiliser cette option pour tous les systèmes de fichiers dont on voudra garantir la cohérence, par exemple pour les systèmes de fichiers devant contenir des documents ou des informations importantes. Sachez cependant que la journalisation peut dégrader sensiblement les performances de votre ordinateur, aussi pouvez-

vous vous en passer sur les partitions pour lesquelles un débit élevé est nécessaire (par exemple pour les partitions devant servir à manipuler des fichiers vidéo).

Note : Le système de fichiers EXT3 utilise les mêmes structures de données que le système de fichiers EXT2. Il est donc parfaitement compatible avec ce dernier, et un système de fichiers EXT3 peut être utilisé comme un système de fichiers EXT2. En réalité, les mécanismes de journalisation peuvent être activés ou non lors de l'opération de montage du système de fichiers, en fonction du type de système de fichiers indiqué à la commande de montage. Nous détaillerons la manière de monter les systèmes de fichiers dans la [Section 6.6.2](#) et dans la [Section 6.6.5](#).

Pour les mêmes raisons, il est possible de convertir un système de fichiers EXT2 en système de fichiers EXT3 a posteriori, à l'aide de l'option `-j` de la commande `tune2fs`. Cette commande permet d'activer et de désactiver des fonctionnalités complémentaires pour ces systèmes de fichiers, dont la journalisation fait partie.

Invoquée sans autres options, la commande `mke2fs` prend des valeurs par défaut pour tous les paramètres du système de fichiers créé, mais vous pouvez également spécifier d'autres valeurs. La taille des blocs peut être indiquée en octets avec l'option suivante :

```
mke2fs [-j] -b taille fichier
```

où `taille` représente la taille d'un bloc en octets. De même, le nombre d'octets par inode peut être précisé avec l'une des options `-i` :

```
mke2fs [-j] -i octets fichier
```

où `octets` est le rapport de la taille de la partition en octets par le nombre d'inodes à créer. Il est possible d'indiquer directement ce nombre avec la commande suivante :

```
mke2fs [-j] -N nombre fichier
```

Enfin, sachez que l'option `-c` permet de demander à `mke2fs` d'effectuer une vérification des secteurs défectueux du disque dur avant de créer le système de fichiers. Il est fortement recommandé d'utiliser cette option lors de la première création d'un système de fichiers.

III-D-5 - Création de la partition de swap

Comme on l'a vu précédemment, Linux peut utiliser une partie du disque dur pour y stocker les données temporairement inutilisées afin de libérer de l'espace mémoire lorsqu'il manque de mémoire vive. Cette opération permet de continuer à travailler, même si la machine ne dispose pas de suffisamment de mémoire vive pour exécuter tous les processus dont elle a la charge.

Évidemment, l'inconvénient de cette méthode est la dégradation des performances, mais c'est un bon compromis si l'on considère le prix du mégaoctet de mémoire par rapport à celui des disques durs d'une part, et le fait qu'il vaut mieux parvenir à faire son travail, même lentement, que de ne pas le faire du tout.

L'espace disque consacré par Linux pour ce stockage temporaire est appelé « swap », du terme anglais « to swap » qui fait référence à l'échange des données de la mémoire vers le disque dur (et inversement, lorsqu'elles sont rechargées en mémoire). Linux est capable de gérer plusieurs formes de swap. Il est capable d'utiliser des fichiers d'échange, qui sont stockés dans un système de fichiers, ou les partitions de swap. Ces dernières ont l'avantage d'être bien plus rapides, puisque le noyau n'a pas à se préoccuper de la structure du système de fichiers lors des opérations de swap (qui, rappelons-le, constituent déjà un ralentissement notable de la machine). En revanche, elles ont l'inconvénient majeur d'être très contraignantes, puisqu'elles nécessitent de réserver une partition pour le swap de manière permanente. Cependant, il est tout à fait acceptable de consacrer 128 ou 256 Mo de disque dur pour une partition de swap de nos jours. Linux est capable de gérer jusqu'à 8 partitions de swap dont la taille peut aller jusqu'à 2 Go chacune, plus les fichiers d'échange que l'on peut rajouter ultérieurement. Nous ne décrivons que la

manière de créer une partition de swap, car les fichiers d'échange ne constituent plus le meilleur compromis avec les tailles de disques que l'on rencontre de nos jours.

Bien entendu, le programme d'installation de votre distribution prend certainement déjà en charge la création des partitions de swap. Il est donc recommandé, encore une fois, d'utiliser ce programme, même si la description qui suit vous permettra de comprendre ce dont il s'agit.

Les partitions de swap peuvent être créées, comme toutes les partitions, à l'aide du programme fdisk. En fait, la seule distinction entre une partition de swap et une partition réservée à un système de fichiers est tout simplement son identificateur. Comme on l'a déjà vu lors du partitionnement du disque, l'identificateur utilisé pour les partitions de systèmes de fichiers Linux est 83, et celui pour les partitions de swap est 82. Vous devrez donc affecter cet identificateur à votre partition de swap lorsque vous partitionnerez votre disque dur. Il est recommandé de placer la partition de swap au début du disque dur, car c'est à cet emplacement que le taux de transfert est le plus élevé (et donc c'est à cet emplacement qu'on obtiendra les meilleures performances en cas de manque de mémoire vive).

Une fois créée, la partition de swap peut être préparée pour que le noyau puisse l'utiliser. Cette préparation revient à peu près à formater un système de fichiers, à ceci près que les structures de données écrites dans la partition de swap sont beaucoup plus simples, car il ne s'agit plus ici de stocker une arborescence complète de fichiers. La commande mkswap permet de préparer les partitions pour être utilisées en tant que partition de swap. Elle s'utilise selon la syntaxe suivante :

```
mkswap -c partition
```

où `partition` est la partition à préparer pour le swap. Notez que, en réalité, mkswap peut tout aussi bien travailler sur un fichier que sur une partition.

Lorsque la partition aura été préparée pour le swap, il est possible de demander à Linux de l'utiliser avec la commande suivante :

```
swapon partition
```

où `partition` est la partition de swap à utiliser. Cette zone de swap est alors automatiquement prise en compte par le système. La commande suivante permet d'arrêter le swapping sur une partition :

```
swapoff partition
```

Normalement, vous n'aurez jamais à utiliser ces commandes manuellement. Le programme d'installation de votre distribution configure le swap, et fait en sorte que les partitions de swap sont chargées automatiquement lors du démarrage de la machine. Notez cependant que cette méthode de configuration dynamique permet d'ajouter temporairement un fichier d'échange si les besoins s'en font sentir, sans avoir à redémarrer la machine.

III-D-6 - Installation des composants de base

Si vous êtes arrivé jusqu'ici, vous avez fini les opérations les plus risquées et sans doute les plus difficiles. Dans les premiers jours de Linux, il fallait installer à la main les différents composants du système, voire les recompiler soi-même. Heureusement, toutes les distributions actuelles disposent aujourd'hui de programmes d'installation évolués, qui simplifient beaucoup le travail. Les opérations que vous allez réaliser à présent sont donc plus simples, et certainement moins dangereuses.

L'ordre logique est évidemment de commencer par installer les couches les plus basses du système, donc en particulier le noyau. Cependant, comme on le verra plus tard, le noyau ne permet pas réellement d'utiliser le système, et il vous faudra installer d'autres couches logicielles. Les principales sont les suivantes :

- le shell (environnement utilisateur en mode texte) ;
- les outils et applications systèmes en ligne de commande (utilisables via le shell) ;

- l'environnement graphique X Window, constitué d'un serveur d'affichage et d'applications graphiques simples ;
- un environnement de bureau (environnement utilisateur complet en mode graphique) ;
- et enfin, les applications de haut niveau (celles que vous utiliserez au jour le jour).

Les programmes d'installation cherchent généralement à installer en premier ce qu'on appelle le « système de base » (ensemble constitué du noyau, des bibliothèques systèmes, du shell et des outils d'administration fondamentaux). Je vous suggère de vous assurer que ce système de base est correctement configuré, avant de vous lancer dans l'installation des couches supérieures telles que XWindow.

C'est évidemment l'installation et la configuration du système de base qui est la partie la plus technique. Une fois ces opérations réalisées, la suite ne pose généralement pas de problème particulier. De nos jours, l'installation de XWindow se fait de manière relativement simple. Lorsque vous y serez parvenu, vous pourrez enfin installer les applications. Là encore, lorsque le système est bien configuré, l'installation des applications est une tâche relativement facile et se fait rapidement.

Ce document ne traitera pas en détail de l'installation des applications, car il y en a trop pour que l'on puisse donner des informations valides pour toutes les applications. Quelques règles générales seront malgré tout données, car elles peuvent s'appliquer pour certaines applications.

Toutes les distributions organisent les différents composants logiciels qu'elles fournissent en paquetages (« package » en anglais). Ainsi, l'installation du système se fait par groupes homogènes de fichiers, et le regroupement dans un paquetage est généralement une dépendance forte (en pratique, ce sont les fichiers d'une même application). En installant un paquetage, on installe finalement un logiciel particulier. Cependant, certains paquetages dépendent d'autres paquetages, par exemple, les paquetages contenant le système de base sont évidemment utilisés par tous les autres paquetages. Les programmes d'installation gèrent relativement bien les dépendances et les conflits entre paquetages, si bien que l'installation peut maintenant se faire sans trop de problèmes.

Afin d'organiser un peu tous ces paquetages, les distributions les trient souvent par « séries ». Une série n'est rien d'autre qu'un ensemble de paquetages regroupés par domaine fonctionnel. Cela signifie que l'on peut facilement retrouver un paquetage donné, en allant le chercher dans la série contenant tous les paquetages fonctionnellement proches. Le regroupement des paquetages en séries ne signifie absolument pas que tous les paquetages de la série doivent être installés pour obtenir une fonctionnalité donnée, mais que les logiciels qui s'y trouvent ont plus ou moins trait à cette fonctionnalité. En fait, il peut même y avoir redondance ou conflit entre deux paquetages d'une même série. Dans ce cas, il faut choisir l'un ou l'autre, selon ses besoins personnels.

Certains paquetages sont indispensables pour le système, d'autres sont purement optionnels. Mais la plupart des paquetages sont simplement les paquetages des applications, et vous devrez faire le tri et choisir ceux qui vous intéressent parce qu'il est impensable d'installer tous les paquetages (une distribution de Linux peut faire 5 ou 6 CD-ROM, en tenant compte du système, des applications et des sources). Les seuls paquetages qu'il faut impérativement installer sont les paquetages de la série de base. En général, cette série porte le nom A, ou AAA, ou quelque chose de similaire, afin qu'elle puisse toujours être en tête de liste dans les programmes d'installation. Cette série comprend au moins les paquetages des commandes Unix de base, du shell, du programme d'installation et de tous les fichiers nécessaires au fonctionnement du système (fichiers de configuration, scripts et bibliothèques partagées). Tant que les programmes de cette série sont intacts et fonctionnels, le système est utilisable. S'il en manque, les problèmes peuvent survenir à tout moment : de l'absence ou l'indisponibilité de la documentation à l'impossibilité complète de démarrer le système.

Le choix des paquetages à installer est crucial, mais non définitif. En effet, si vous avez installé un paquetage dont vous n'avez pas ou plus besoin, rien ne vous empêche de le supprimer par la suite. De même, si vous vous apercevez que vous avez oublié d'installer un paquetage dont vous avez besoin, vous pouvez l'installer ultérieurement.

Il n'est pas possible de donner ici la liste des paquetages que vous devez installer, car cette liste dépend beaucoup trop de la distribution que vous possédez. Il est conseillé de lire le manuel de cette distribution ou de bien lire les écrans d'aides du programme d'installation. Cependant, les paquetages dont vous aurez certainement besoin pour poursuivre l'installation sont sans doute les suivants :

- paquetage du système de base ;
- paquetage du compilateur GCC ;
- paquetage du réseau ;
- paquetage de XWindow ;
- paquetage de documentation ;
- paquetages susceptibles de faire fonctionner votre matériel (carte son, serveur XWindow approprié à votre carte graphique...).

Gardez à l'esprit que dans le monde du logiciel libre, les programmes sont souvent distribués sous la forme de fichiers sources et que vous aurez sans doute besoin des outils de développement pour les compiler et les installer. Veillez donc à inclure d'office tous ces outils, même si vous ne désirez pas programmer personnellement. Bien entendu, vous pourrez revenir ultérieurement dans le programme d'installation et réinstaller un paquetage si vous l'avez oublié pendant la phase d'installation.

III-D-7 - Configuration du gestionnaire d'amorçage

Lorsque vous aurez installé votre système de base, vous devrez faire en sorte qu'il puisse démarrer. Pour cela, il existe plusieurs possibilités, les principales étant les deux suivantes :

- soit vous démarrez à partir d'une disquette ou d'un CD d'amorçage ;
- soit vous utiliser un gestionnaire d'amorçage.

Il va de soi que c'est la deuxième solution qui est recommandée. Cependant, la première solution pourra être utile si d'aventure votre MBR se trouvait être écrasé ou endommagé. Seule l'utilisation des gestionnaires d'amorçage de Linux et de Windows NT/2000/XP sera décrite ici. Le gestionnaire d'amorçage de Windows Vista ne sera pas décrit.

Le gestionnaire d'amorçage le plus utilisé sous Linux se nomme « LILO » (pour « Linux LOader »). LILO permet de démarrer un grand nombre de systèmes, dont DOS, Windows 95/98/Millennium, Windows NT/2000/XP/Vista et OS/2. Linux dispose également d'un autre gestionnaire d'amorçage également très performant : le « GRUB » (abréviation de l'anglais « GRand Unified Bootloader »). Vous êtes libre de choisir celui que vous voulez. Le GRUB est quelque peu plus moderne que LILO, mais le choix du gestionnaire d'amorçage est ici une question de goût. Windows NT, Windows 2000 et Windows XP disposent également d'un gestionnaire d'amorçage nommé « NTLDR », capable de démarrer d'autres systèmes d'exploitation. Vous aurez donc également la possibilité d'utiliser ce gestionnaire à la place de LILO ou du GRUB si un de ces systèmes est installé sur votre machine. En revanche, Windows Vista utilise un nouveau gestionnaire d'amorçage différent de NTLDR, et qui est techniquement beaucoup plus difficile à utiliser que les autres gestionnaires d'amorçage. Ce gestionnaire ne sera donc pas décrit dans ce document, référez-vous à la documentation Microsoft si vous voulez installer Linux en marge de Vista, ou utilisez tout simplement l'un des gestionnaires d'amorçage de Linux.

Quel que soit le gestionnaire d'amorçage que vous désirez utiliser, il vous faudra activer la partition sur laquelle il est installé. Cette opération peut être réalisée à l'aide de l'utilitaire fdisk et a déjà été décrite dans la [Section 3.2.1](#). Cela permettra au bootstrap loader de sélectionner le secteur de boot de cette partition et de lancer le gestionnaire d'amorçage qui y est installé. Bien entendu, cela suppose que ce gestionnaire soit installé sur cette partition, ce qui est normalement le cas (rappelons qu'il n'est en général pas conseillé d'installer le gestionnaire d'amorçage directement sur le MBR du disque dur).

III-D-7-a - Réalisation d'un multiboot avec LILO

LILO est un gestionnaire d'amorçage extrêmement performant, puisqu'il permet de démarrer Linux comme tout autre système d'exploitation très simplement, en donnant le nom du système à lancer lors de l'amorçage. Il est bien entendu possible de lancer un système par défaut, et de donner un temps d'attente avant de choisir ce système si l'utilisateur n'intervient pas.

LILO est constitué de deux parties. La première partie peut s'installer sur le secteur d'amorçage principal du disque dur ou sur le secteur de boot de n'importe quelle partition. Comme on l'a déjà indiqué plus haut, il est fortement

recommandé d'installer cette partie de LILO sur le secteur de boot de la partition racine de Linux, afin d'éviter qu'elle ne soit écrasée par le DOS ou par une quelconque version de Windows installée ultérieurement. La deuxième partie de LILO est enregistrée directement dans la partition Linux. Elle contient les informations nécessaires pour pouvoir charger les différents systèmes d'exploitation gérés. Bien entendu, la première partie est capable de retrouver directement la deuxième sur le disque dur de manière autonome, car, lors de l'amorçage, les systèmes de fichiers de Linux ne sont pas encore chargés.

LILO utilise le fichier de configuration `/etc/lilo.conf` pour y retrouver tous ses paramètres de configuration. Ce fichier contient la description des différents systèmes d'exploitation que LILO doit proposer au démarrage. Vous pourrez consulter ce fichier avec un des nombreux éditeurs de fichiers texte présents sur toute installation de Linux. Toutefois, si vous installez Linux pour la première fois, il est possible que vous n'en connaissiez aucun et que vous soyez un peu perdu. Cela est normal, et dans ce cas je vous recommande de vous familiariser un peu avec le système et l'environnement utilisateur avant de vous lancer dans l'édition de ce fichier. Il existe bon nombre d'éditeurs graphiques ou en mode texte et il est hors de question de tous les décrire ici. Toutefois, toutes les distributions Linux installent un éditeur historique, j'ai nommé l'affreux « vi ». Cet éditeur n'est pas du tout convivial pour les nouveaux utilisateurs, mais il dépannera toujours quand tous les autres seront inutilisables ou inaccessibles. En fait, on finit même par l'apprécier à l'usage...

La manière d'utiliser vi sera décrite ultérieurement, dans le chapitre donnant les notions de base d'Unix à la [Section 5.8](#). Vous devriez donc jeter un coup d'œil à cette section si vous désirez modifier immédiatement le fichier `/etc/lilo.conf`, ou revenir ultérieurement à la présente section une fois que vous vous serez familiarisé avec un autre éditeur.

Quoi qu'il en soit, les options les plus importantes du fichier `/etc/lilo.conf` sont les suivantes :

- l'option `boot`, qui permet d'indiquer sur quel secteur d'amorçage LILO doit s'installer. Cette option suit la syntaxe suivante :

```
boot = destination
```

- l'option `read-only` permet de demander au noyau de monter le système de fichiers racine en lecture seule lors du démarrage. Cette option est nécessaire pour que les scripts de démarrage du système puissent effectuer les vérifications du système de fichiers de cette partition si nécessaire. La partition sera remontée en lecture et en écriture une fois ces vérifications réalisées ;
- l'option `prompt`, qui permet à LILO de demander le système à lancer à chaque démarrage. Cette option force donc l'apparition du message d'invite de LILO au démarrage : « LILO boot: » auquel on pourra répondre en tapant le nom de la configuration à lancer ;
- l'option `timeout`, qui permet de fixer un délai au-delà duquel LILO lancera la première configuration définie dans le fichier `lilo.conf`. La syntaxe de cette option est la suivante :

```
timeout = dixièmes
```

- l'option `keytable`, qui donne la possibilité de spécifier un fichier de traduction des codes de caractère envoyés par le BIOS (qui suppose généralement que le clavier utilise la disposition d'un clavier américain) en les codes de caractère qui seraient envoyés par un BIOS localisé. Cette option permet donc de redéfinir la disposition des touches du clavier pour prendre en compte les claviers non américains. La syntaxe de cette option est la suivante :

```
keytable = fichier
```

```
keytab-lilo.pl us local > fichier
```

```
keytab-lil.pl us fr-latin1 > /boot/fr-latin1.klt
```

La suite du fichier `lilo.conf` décrit les différentes configurations que LILO peut lancer. Les sections de configuration permettant de charger Linux ont le format suivant :

```
image = noyau
root = root_device
```

```
label = nom
```

où `noyau` est le chemin complet sur le noyau de Linux à charger, `root_device` est le nom complet du fichier spécial de périphérique contenant le système de fichier racine et `nom` est le nom de la configuration tel qu'il devra être saisi à l'invite de LILO. L'exemple donné ci-dessous permet de charger le noyau `/boot/vmlinuz` en utilisant la partition `/dev/hda2` comme partition racine :

```
image = /boot/vmlinuz
root = /dev/hda2
label = linux
```

Si vous désirez créer une section de configuration permettant de lancer un autre système d'exploitation que Linux (DOS ou Windows par exemple), vous pouvez utiliser la possibilité de passer le relais au chargeur de ces systèmes, qu'il s'agisse d'un simple secteur de boot ou d'un gestionnaire d'amorçage complet. Cela se fait avec la syntaxe suivante :

```
other = partition
table = disque
loader = relais
label = nom
```

où `partition` est la partition sur laquelle le secteur de boot de l'autre système est installé, `disque` est le disque dur contenant la table des partitions utilisée par ce système, `relais` est le nom d'un chargeur spécial permettant de simplement passer la main au chargeur du système, et `nom` est le nom de la configuration. Le chargeur à utiliser pour demander à LILO de passer le relais au chargeur de l'autre système d'exploitation est le chargeur contenu dans le fichier `chain.b` de LILO. Ce fichier se trouve généralement dans le répertoire `/boot/`, aussi doit-on spécifier `/boot/chain.b` pour le champ `relais`.

Note : Prenez garde au fait que Windows NT/Windows 2000/XP installent NTLDR dans la première partition à laquelle il savent accéder en général. Donc, si un DOS ou Windows 95, Windows 98 ou Millenium est installé en premier, ils installeront NTLDR dans la partition de ces systèmes. Dans ce cas, la configuration permettant de lancer le DOS ou le Windows 95, Windows 98 ou Millenium qui se trouve sur cette partition risque fort de lancer NTLDR qui proposera, à son tour, de lancer les différents systèmes d'exploitation Microsoft installés.

Cela peut être relativement gênant et peut être corrigé en déplaçant NTLDR sur la partition de Windows NT/Windows 2000/XP et en reconstruisant les secteurs de boot des différentes partitions pour que leurs chargeurs s'occupent de leurs systèmes respectifs, mais il s'agit là d'une opération extrêmement technique d'une part, et qui ne concerne absolument pas Linux d'autre part. Cela ne sera donc pas décrit dans ce document. Il existe toutefois des documents sur Internet qui décrivent la manière de procéder et je vous invite à vous y référer (avec une prudence extrême cependant).

L'exemple donné ci-dessous permet de donner la possibilité de charger Linux ou Windows NT, en lançant Linux par défaut au bout de 10 secondes. Windows NT est installé sur la troisième partition, et Linux utilise la deuxième et la quatrième partition respectivement pour y stocker sa partition racine et la partition des répertoires personnels des utilisateurs. LILO est ici installé sur la partition racine de Linux :

```
# Exemple de fichier de configuration /etc/lilo.conf :

# Options générales :
boot = /dev/hda2
read-only
prompt
timeout=100
keytable = /boot/fr-latin1.klt

# Première configuration (Linux) :
image = /boot/vmlinuz
root = /dev/hda2
label = linux

# Deuxième configuration (NT) :
```

```
other = /dev/hda3
table = /dev/hda
loader = /boot/chain.b
label = NT
```

L'installation de LILO est très simple une fois que l'on a écrit le fichier `lilo.conf`. En effet, il suffit de taper la commande suivante :

```
lilo [-L]
```

L'option `-L` permet de demander à LILO d'utiliser le mode d'adressage LBA pour accéder au disque dur pendant la phase d'amorçage. Cette option est nécessaire si vous disposez d'un grand disque dur et que certaines partitions disposant de systèmes à lancer sont situées au-delà du cylindre 1024. Il est recommandé de l'utiliser systématiquement étant donné les tailles des disques durs actuels.

Note : Comprenez bien que si votre BIOS est incapable d'utiliser le mode LBA ou le si bootstrap loader est incapable d'utiliser ce mode, cette option ne vous sera d'aucune utilité. En effet, dans ce cas, le bootstrap loader ne parviendrait même pas à charger le secteur de boot de la partition Linux. C'est pour cette raison qu'il a été recommandé de placer la partition du système principal en deçà de cette limite des 1024 cylindres. Cette limitation est donc bien une limitation du BIOS, mais vous ne devriez plus rencontrer ce genre de problème que sur de vieilles machines sur lesquelles un nouveau disque dur de grande capacité a été installé.

Si lilo signale une erreur, il vaut mieux ne pas insister et corriger le fichier `lilo.conf`.

Une fois LILO installé correctement, il affichera l'invite de démarrage lors du démarrage de la machine : LILO boot:

Il attend ici que vous indiquiez le nom du système que vous désirez démarrer. Vous devez ici taper le nom du système à charger et valider : LILO boot:linux

Si vous ne tapez rien, et que vous avez donné un délai d'attente dans le fichier de configuration de LILO, la première configuration sera lancée automatiquement après ce délai.

LILO permet de spécifier des paramètres de démarrage complémentaires pour Linux à la suite du nom de la configuration qui permet de le lancer. Ces paramètres servent principalement à renseigner le noyau sur la configuration matérielle (en particulier les ports d'entrée/sortie et les lignes d'interruption des périphériques non Plug and Play), pour le cas où il ne parviendrait pas à les déterminer automatiquement. L'un des paramètres les plus intéressants est sans doute `mem`, qui permet d'indiquer au noyau la taille de la mémoire vive dont dispose l'ordinateur. Ce paramètre peut être nécessaire si vous disposez de plus de 64 Mo de mémoire, parce que les fonctions du BIOS ne permettent pas d'indiquer les tailles de mémoire plus grandes (la plupart des BIOS récents n'ont toutefois plus ce problème). Par exemple, si votre ordinateur dispose de 256 Mo de mémoire, vous devrez taper la ligne de paramètres suivante au démarrage : LILO boot:linux mem=256M

Bien entendu, il est possible d'enregistrer ces paramètres dans le fichier de configuration de LILO afin de ne pas avoir à les saisir à chaque démarrage. Pour cela, il suffit d'indiquer le paramètre de démarrage du noyau dans une ligne `append` de la section de configuration de Linux :

```
append="paramètre"
```

Ainsi, la section de configuration de Linux du fichier `lilo.conf` exemple donné ci-dessus pourrait être remplacée par celle-ci sur une machine disposant de 256 Mo de mémoire :

```
# Première configuration (Linux) :
image = /boot/vmlinuz
root = /dev/hda2
label = linux
append="mem=256M"
```

La liste des paramètres que l'on peut fournir au noyau est relativement grande et ne sera pas décrite ici. Les plus utiles seront présentés en temps et en heure, notamment dans le chapitre décrivant la configuration du système.

III-D-7-b - Réalisation d'un multiboot avec le GRUB

Le « GRUB » (abréviation de l'anglais « GRand Unified Bootloader ») est le gestionnaire d'amorçage développé par la Free Software Foundation pour amorcer le noyau Hurd du projet GNU. Il est capable de faire démarrer tous les systèmes utilisant un protocole de démarrage standard initialement défini pour le Hurd. Bien entendu, il sait aussi amorcer les systèmes Linux, qui n'utilisent pas ce protocole, ainsi que la plupart des autres systèmes d'exploitation en passant le relais à leurs propres gestionnaires d'amorçage.

En fait, le GRUB fournit la possibilité de contrôler totalement l'amorçage de son système par l'intermédiaire d'un interpréteur de commandes simplifié. Il est possible, par l'intermédiaire de cet interpréteur de commandes, d'effectuer nombre d'opérations dans le but de charger un noyau de système d'exploitation et de l'amorcer. Bien entendu, ces commandes peuvent être écrites dans un fichier de configuration afin d'automatiser le processus de chargement.

Le GRUB est normalement installé dans le répertoire `/boot/grub/`. Ce répertoire contient, outre les fichiers binaires du GRUB lui-même, son fichier de configuration. Ce fichier se nomme normalement `menu.lst`, en raison du fait qu'il permet de définir les différentes configurations correspondantes aux systèmes d'exploitation à charger et qui apparaîtront dans le menu de démarrage lors de l'amorçage de la machine.

Contrairement à LILO, qui enregistre l'emplacement des fichiers des différents noyaux à charger dans une liste de blocs du disque dur, le GRUB sait interpréter les systèmes de fichiers classiques de Linux. En particulier, il est capable de retrouver son fichier de configuration et les fichiers images des noyaux Linux dans les systèmes de fichiers FAT, EXT2/EXT3 et ReiserFS. Cette particularité fait qu'il n'est pas nécessaire, lorsqu'on modifie le fichier de configuration `menu.lst`, de réinstaller le GRUB.

Tout comme le fichier de configuration de LILO, le fichier `menu.lst` se compose d'une partie contenant les options globales et d'une ou plusieurs parties contenant la description des différents systèmes d'exploitation à proposer au démarrage. Les options générales les plus utiles sont les suivantes :

- l'option `default`, qui permet de spécifier la configuration par défaut à charger. Cette option doit être suivie du numéro de cette configuration. Les configurations sont numérotées à partir de 0, dans leur ordre d'apparition dans le fichier de configuration ;
- l'option `timeout`, qui permet de spécifier le délai d'attente avant que la configuration par défaut spécifiée par l'option `default` ne soit lancée.

Les configurations spécifiques aux systèmes d'exploitation suivent la syntaxe suivante :

```
title titre
root partition
kernel noyau options
```

où `titre` est le titre de la configuration tel qu'il doit apparaître dans le menu de démarrage du GRUB, `partition` est la partition dans laquelle se trouve le noyau à charger, et `noyau` est le chemin sur le fichier image de ce noyau dans cette partition. Attention, ce chemin est défini dans la partition elle-même et peut donc être différent du chemin utilisé sous Linux. En effet, il faut définir ce chemin par rapport au point de montage de la partition, faute de quoi le GRUB ne retrouverait pas le fichier image du noyau à charger.

Comme le GRUB n'est pas un chargeur spécifique à Linux, mais a été écrit au contraire avec comme principal objectif une généricité absolue, la manière de spécifier la partition dans laquelle le noyau se trouve utilise une syntaxe différente de celle utilisée sous Linux. Cette syntaxe, propre au GRUB donc, est la suivante :

```
(hdc,m)
```

où *n* est le numéro du disque dans l'ordre énuméré par le BIOS de la machine et *m* est le numéro de la partition. Ce dernier numéro est facultatif (ainsi que la virgule qui le précède), ce qui permet de référencer un disque complet et non une partition. La numérotation des disques et des partitions commence toujours à 0 dans le GRUB, ce qui fait que la première partition du premier disque est référencée par (hd0,0), la deuxième partition du premier disque par (hd0,1), la première partition du deuxième disque par (hd1,0), etc.

Tout comme avec LILO, il est possible de spécifier des options de démarrage qui devront être fournies au noyau. Ces options devront être spécifiées immédiatement après le nom de l'image du noyau. Comme vous pouvez le constater, la définition d'une configuration de démarrage pour un système Linux est très simple, puisqu'il suffit quasiment de donner la ligne de commande pour lancer ce noyau !

Par exemple, pour charger le noyau `/boot/vmlinuz` d'un système situé sur la deuxième partition du premier disque, la configuration suivante doit être définie :

```
title Linux
root (hd0,1)
kernel /boot/vmlinuz mem=256M
```

Cet exemple présente également comment spécifier la taille de la mémoire disponible dans la machine (cela n'est normalement pas nécessaire avec les BIOS récents et avec le GRUB). Il suffit simplement de fournir des options en ligne de commande au noyau pour cela. Beaucoup d'autres options peuvent être fournies de cette manière au noyau. En particulier, des options peuvent être fournies aux pilotes de périphériques si nécessaire (par exemple s'il y a des conflits d'interruptions ou de ressources entre plusieurs périphériques). Ces options seront présentées au fur et à mesure dans la suite de ce document.

Bien entendu, le GRUB est capable de charger le secteur de boot d'une partition afin de passer le relais au gestionnaire d'amorçage d'un autre système d'exploitation. Pour cela, il faut utiliser la commande `chainloader`, plutôt que la commande `kernel`, dans la description de la configuration de démarrage de ce système. La forme générale d'une configuration de ce type est donc la suivante :

```
title titre
root partition
chainloader +1
```

Le `+1` qui suit la commande `chainloader` indique au GRUB de charger le premier secteur de la partition indiquée par la commande `root` et d'exécuter le gestionnaire d'amorçage normalement stocké dans ce secteur. Comme pour les configurations Linux, la syntaxe utilisée pour spécifier la partition où ce secteur est situé est la syntaxe du GRUB et non celle utilisée sous Linux.

Le fichier de configuration d'exemple suivant correspond au fichier de configuration de LILO vu dans la section précédente. Il permet de démarrer un Linux installé sur la deuxième partition ou un Windows NT installé sur la troisième partition du premier disque dur de la machine :

```
# Exemple de fichier de configuration /boot/grub/menu.lst :

# Options générales :
default 0
timeout 10

# Première configuration (Linux) :
title Linux
root (hd0,1)
kernel /boot/vmlinuz root=/dev/hda2 mem=256M

# Deuxième configuration (NT) :
title NT
root (hd0,2)
chainloader +1
```


L'installation du GRUB sur une nouvelle machine ne pose quant à elle pas de problème particulier. Il suffit de s'assurer que le fichier de configuration `menu.lst` se situe bien dans le répertoire `/boot/grub/`, de même que les fichiers binaires du GRUB. Ces fichiers sont respectivement les fichiers `stage1`, `stage2` et tous les fichiers `*_stage1_5`. S'ils ne s'y trouvent pas, vous pourrez les copier à partir du répertoire `/usr/share/grub/i386-pc/`, dans lequel le programme d'installation du GRUB les place par défaut.

Lorsque tous les fichiers sont en place, il n'y a plus qu'à lancer le GRUB en mode interactif avec la commande suivante :

```
grub
```

et à définir le secteur où il doit installer son fichier d'amorçage principal `stage1` (c'est-à-dire dans le secteur de boot d'une partition ou directement sur le MBR du premier disque dur). Pour cela, vous devrez utiliser les deux commandes suivantes :

```
root source
setup destination
```

`source` est ici la partition où est installé le GRUB (il s'agit donc de la partition où se trouvent le répertoire `/boot/grub/`), et `destination` est le disque dur ou la partition dont le premier secteur doit recevoir le code d'amorçage du GRUB. Ces deux informations doivent suivre la syntaxe utilisée par le GRUB pour spécifier les disques durs et les partitions. Par exemple, pour installer le GRUB sur le secteur de boot de la deuxième partition du premier disque dur, on utilisera les deux commandes suivantes :

```
root (hd0,1)
setup (hd0,1)
```

Cet exemple suppose que le GRUB est également installé dans cette partition. Si ce n'est pas le cas pour vous, vous devrez modifier la partition spécifiée dans la commande `root`. Vous pourrez quitter le grub avec la commande quit une fois l'installation terminée.

III-D-7-c - Réalisation d'un multiboot avec NTLDR

Le gestionnaire d'amorçage de Windows NT, Windows 2000 et XP se nomme « NTLDR ». Ce gestionnaire d'amorçage permet de démarrer ces systèmes, bien entendu, mais également les autres systèmes d'exploitation les plus courants, dont Linux. Cette section ne traitera bien entendu que de la manière d'utiliser NTLDR pour démarrer Linux, pour de plus amples informations sur la manière d'ajouter les autres systèmes d'exploitation, veuillez consulter la documentation de Microsoft.

Tout comme LILO et le GRUB, NTLDR dispose d'un fichier de configuration qui permet de décrire les différentes options de son menu de démarrage. Il s'agit du fichier `boot.ini`, qui est normalement placé à la racine du disque où NTLDR est installé. Il s'agit donc généralement du répertoire racine du disque `C:\`.

Le fichier `boot.ini` contient la définition du système d'exploitation à lancer par défaut et du délai d'attente à attendre avant de le sélectionner. Ces deux informations sont écrites dans la section `[boot loader]` du fichier et sont introduites respectivement par les options `default` et `timeout`. La valeur de l'option `default` doit être l'une des entrées de la section `[operating systems]`, qui contient la définition de tous les systèmes d'exploitation que NTLDR devra proposer au démarrage.

La syntaxe des entrées pour les systèmes d'exploitation dans la section `[operating systems]` est la suivante :

```
emplacement = "Nom"
```

où `emplacement` est la description de l'emplacement où se trouve le système d'exploitation, et `Nom` est le nom avec lequel ce système devra apparaître dans le menu de démarrage de NTLDR. Cette syntaxe est très simple, mais pour les systèmes NT4, Windows 2000 et XP, la définition de l'emplacement du système est assez compliquée. En effet,

elle nécessite de définir complètement le disque physique et sa partition, ainsi que le répertoire du système. Pour les autres systèmes d'exploitation, le plus simple est de spécifier un fichier contenant l'image du secteur de boot de leur partition, et de laisser leur chargeur prendre en charge leur amorçage.

Pour cela, il faut bien entendu disposer d'un tel fichier. Il faut donc copier le premier secteur de la partition du système dans un fichier accessible à NTLDR, donc, en pratique, situé dans le répertoire C:\. Vous pourrez extraire le contenu du secteur de boot de votre partition Linux avec la commande suivante sous Linux :

```
dd if=/dev/hda3 of=linux.sec bs=512 count=1
```

Cette commande suppose que LILO ait été installé sur la partition /dev/hda3. Elle permet de lire un bloc de 512 octets de la troisième partition du premier disque dur et de l'enregistrer dans le fichier linux.sec. Vous pourrez ensuite transférer ce fichier dans le répertoire C:\ de Windows, soit en passant par une partition FAT, soit tout simplement à l'aide d'une disquette (rappelons que le système de fichiers NTFS n'est utilisable qu'en lecture seule sous Linux).

Une fois ce fichier obtenu, vous pourrez simplement ajouter la ligne suivante dans votre fichier boot.ini :

```
C:\linux.sec="Linux"
Le contenu de votre fichier boot.ini devrait alors ressembler à ceci :
[boot loader]
timeout=10
default=multi(0)disk(0)rdisk(0)partition(2)\WINNT

[operating systems]
multi(0)disk(0)rdisk(0)partition(2)\WINNT="Microsoft Windows 2000 Professionnel" /fastdetect
C:\linux.sec="Linux"
```

Vous devriez alors pouvoir démarrer Linux directement à partir du menu de démarrage de NTLDR.

Note : Prenez soin à utiliser un nom court pour le fichier contenant le secteur de Linux (c'est-à-dire un nom ne comprenant pas plus de huit caractères et une extension d'au plus trois caractères). En effet, Windows, même dans ses versions les plus récentes, a toujours du mal à prendre en charge les noms longs et il se peut que le gestionnaire d'amorçage NTLDR ne puisse pas localiser le fichier si vous ne respectez pas cette règle.

III-E - Installation de la Mandriva 2007

Nous allons présenter dans cette section la procédure d'installation de la distribution Mandriva 2007. Comme nous allons le voir, cette distribution simplifie réellement la procédure d'installation et vise à être particulièrement conviviale.

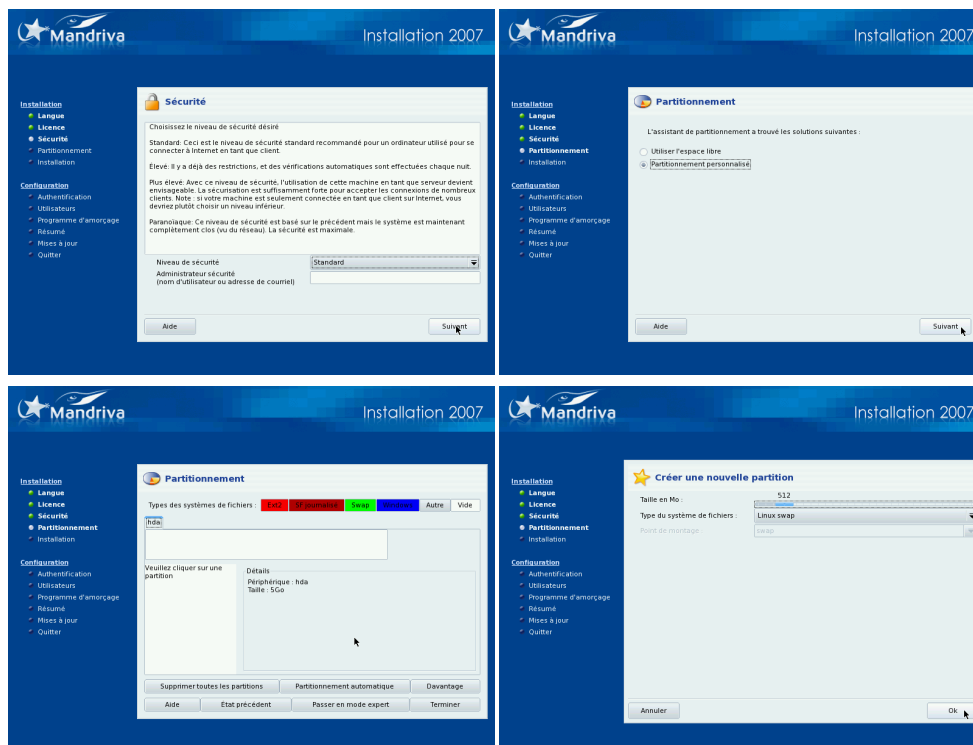
L'installation peut être débutée simplement en plaçant le DVD d'installation dans le lecteur de DVD de l'ordinateur et en configurant le BIOS pour qu'il démarre sur ce DVD. Les premiers écrans du programme d'installation graphique apparaissent alors :



Le premier écran vous permettra de lancer l'installation proprement dite. Vous pouvez changer la langue du système d'installation à ce niveau en appuyant sur la touche « F2 », ainsi que des options complémentaires de démarrage avec la touche « F3 ». Une fois ceci fait, vous pouvez choisir l'option « Installation » pour poursuivre.

Le deuxième écran vous permettra cette fois de choisir la langue du système d'exploitation (et non seulement celle du programme d'installation). Les deux écrans qui suivent vous demandent respectivement d'accepter la licence d'utilisation et de choisir le type de clavier que vous voudrez utiliser.

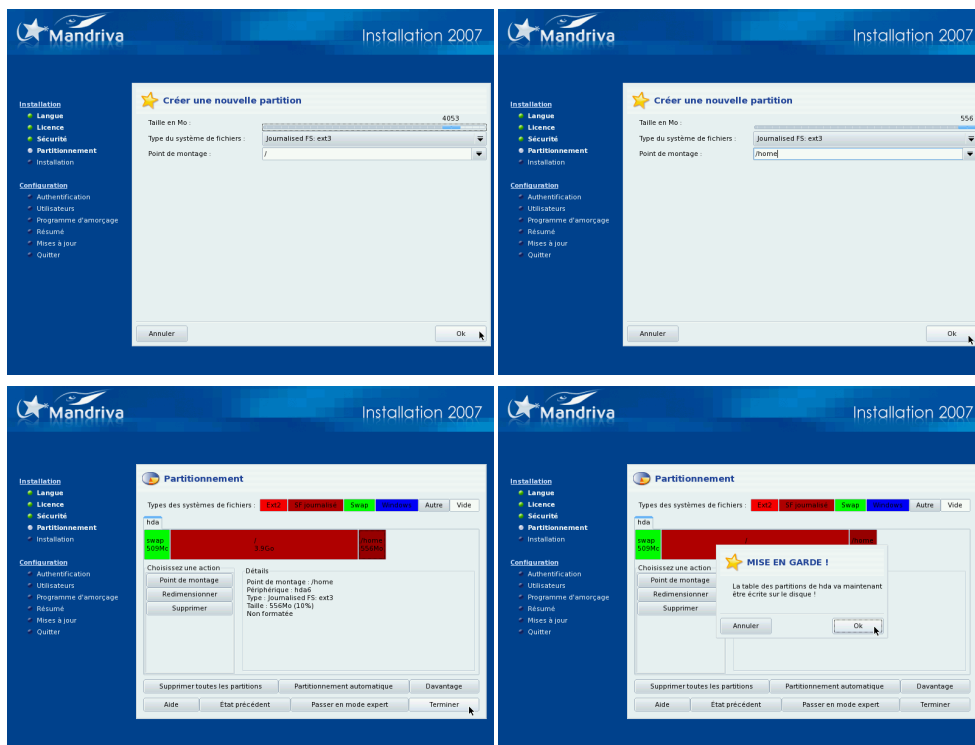
Le programme d'installation vous invite ensuite à choisir un profil de sécurité pour votre système Linux, et à créer les partitions à l'aide de son outil de partitionnement interactif :



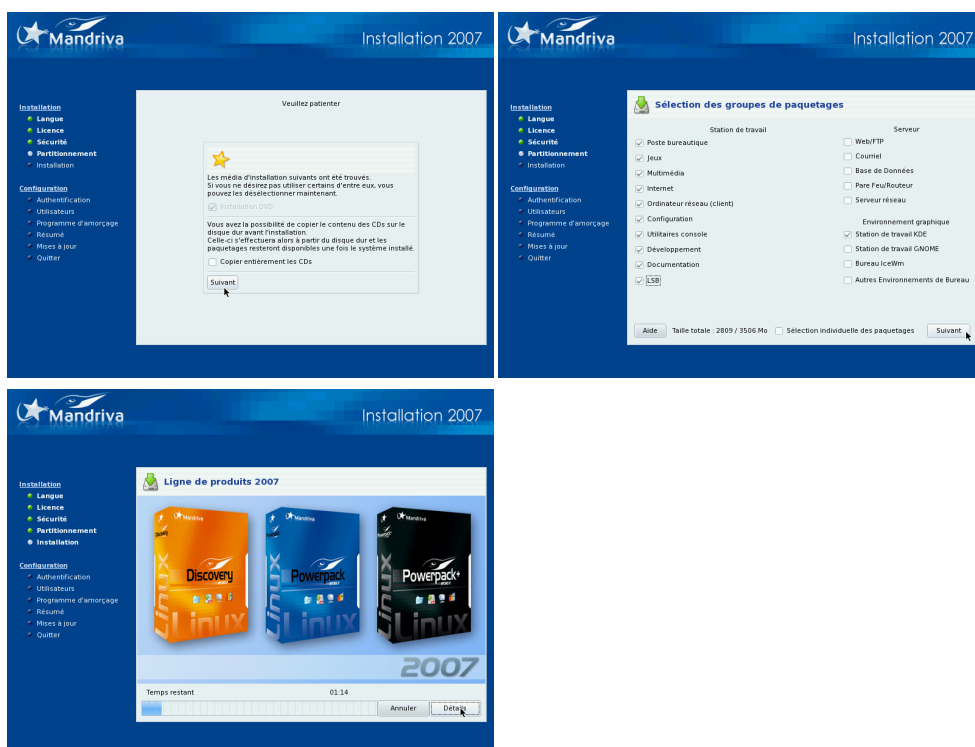
Le niveau de sécurité standard convient pour la plupart des utilisateurs. Les niveaux de sécurité supérieurs peuvent nécessiter des compétences d'administration plus élevées, et ne sont a priori pas nécessaires pour un particulier.

Le partitionnement peut être réalisé manuellement ou automatiquement. Il est recommandé de choisir la méthode manuelle, quitte à accepter les valeurs proposées par défaut, ne serait-ce que pour savoir ce que fait l'outil d'installation. Dans la suite de ce texte, le partitionnement a été réalisé manuellement. On voit sur le troisième écran l'outil de partitionnement. La création d'une partition se fait tout simplement, en cliquant d'abord sur l'espace libre devant la contenir, puis sur le type de la partition. Les options de la partition peuvent ensuite être modifiées. Le quatrième écran présente la création de la partition d'échange (swap).

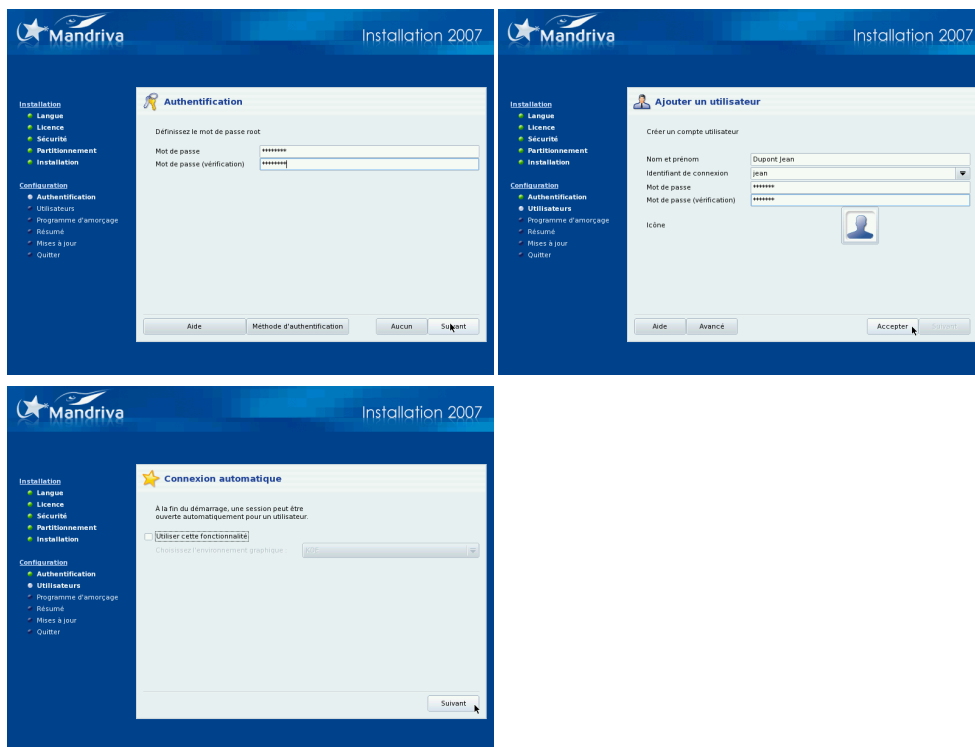
Les deux écrans suivants présentent la manière de créer les partitions racine et /home/. Le troisième écran présente le plan de partitionnement final, et le quatrième écran la demande de confirmation finale après laquelle la table des partitions sera écrite sur le disque dur :



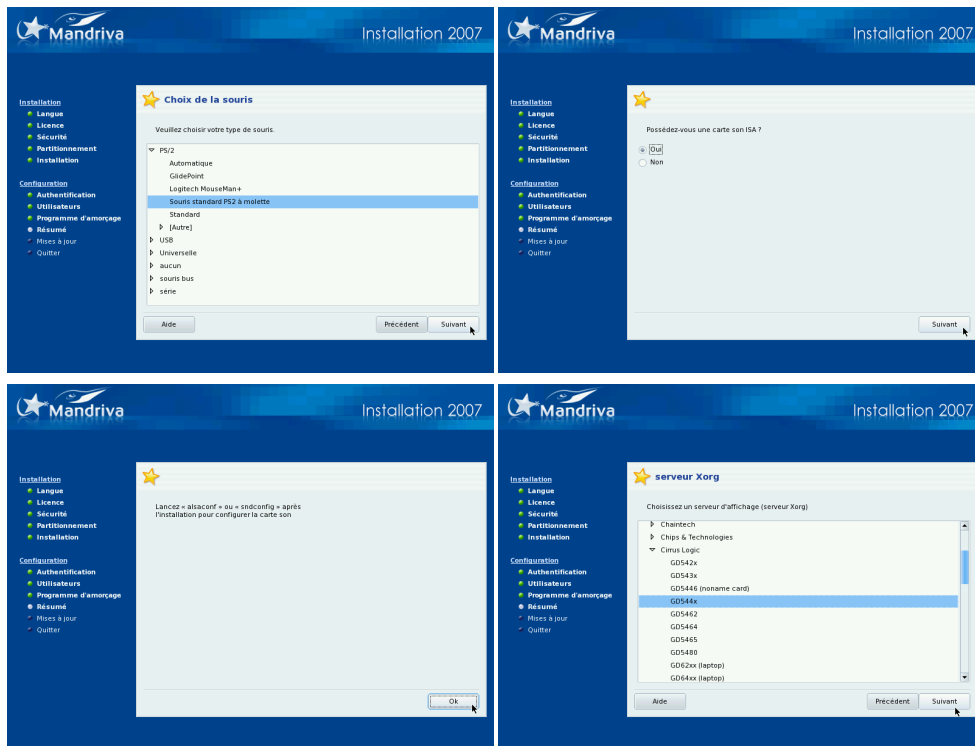
Il faut ensuite indiquer le support d'installation à utiliser (dans notre cas, c'est le DVD d'installation), et choisir les logiciels à installer. La copie des fichiers a ensuite lieu, cette opération peut prendre un certain temps :



Une fois l'installation terminée, le système vous demande un mot de passe pour l'administrateur. Vous pourrez ensuite créer les utilisateurs standards (cette étape est impérative). En revanche, vous pourrez activer la fonctionnalité de connexion automatique au démarrage de la machine si vous n'utilisez votre machine qu'à titre personnel :

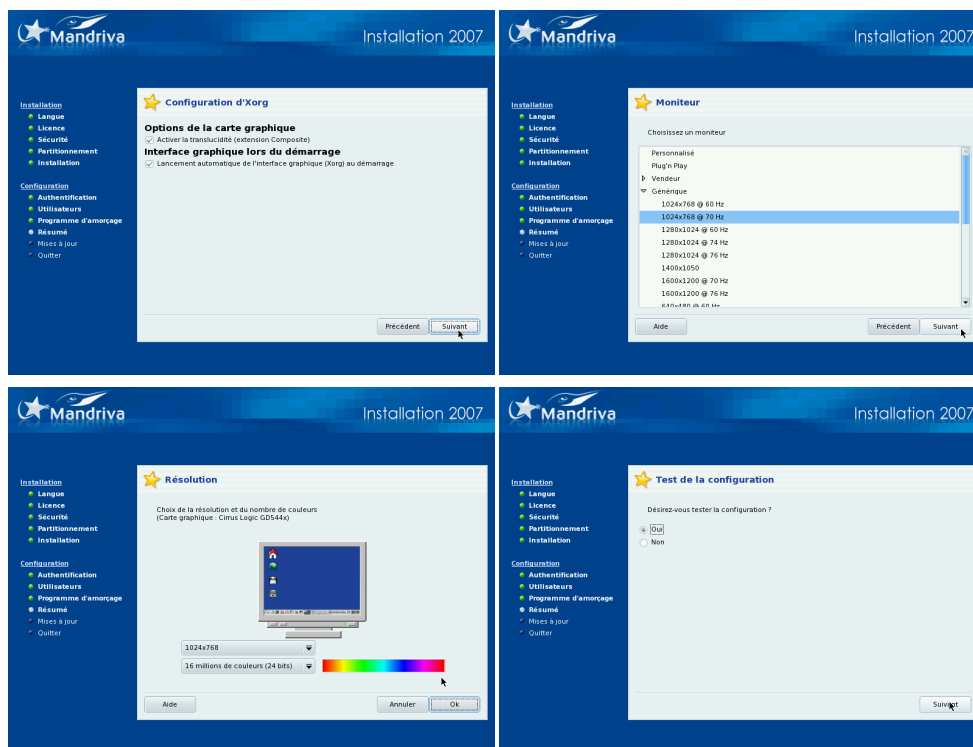


Vient ensuite la configuration du matériel. Vous devez définir le type de souris, indiquer si vous utilisez une carte son ISA ou non (le troisième écran indique que la configuration des périphériques audio n'est pas encore totalement intégrée dans le programme d'installation et doit être terminée après l'installation avec un outil spécifique aux pilotes de son), et le type de carte graphique utilisée pour l'affichage graphique :



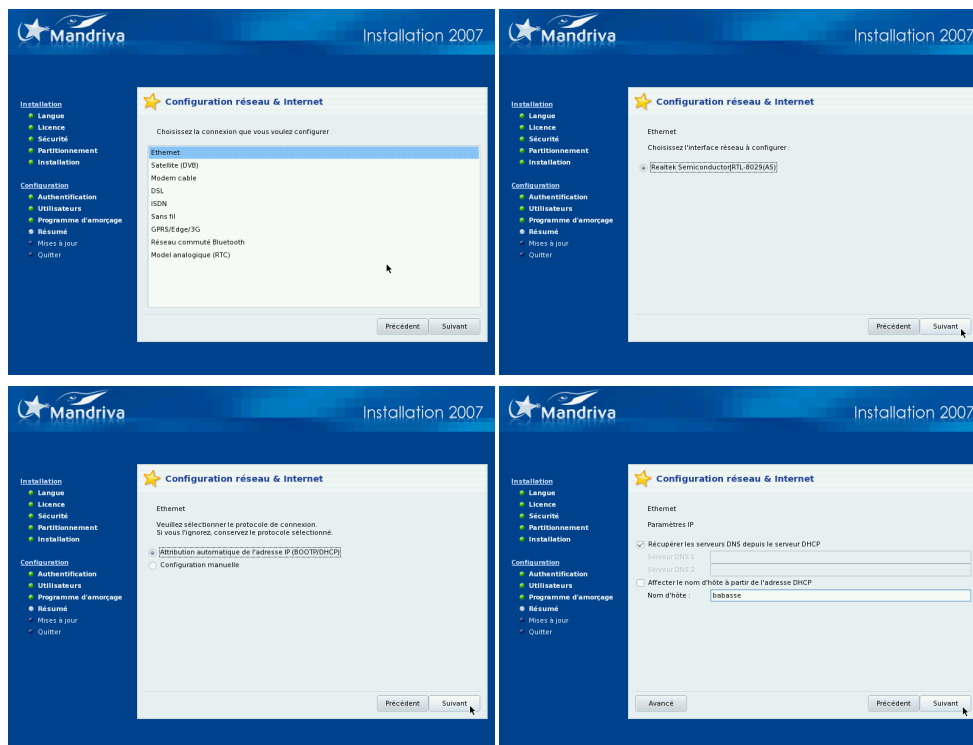
Le programme d'installation exige des informations complémentaires pour l'interface graphique. Vous devez indiquer si vous voulez qu'elle soit activée au démarrage ou non et si vous voulez que les fonctionnalités de transparence soient activées (premier écran). Vous devez également indiquer l'écran que vous utilisez et la résolution graphique

utilisée (deuxième et troisième écran). Le dernier écran vous propose de tester la configuration graphique, ce qui est recommandé :

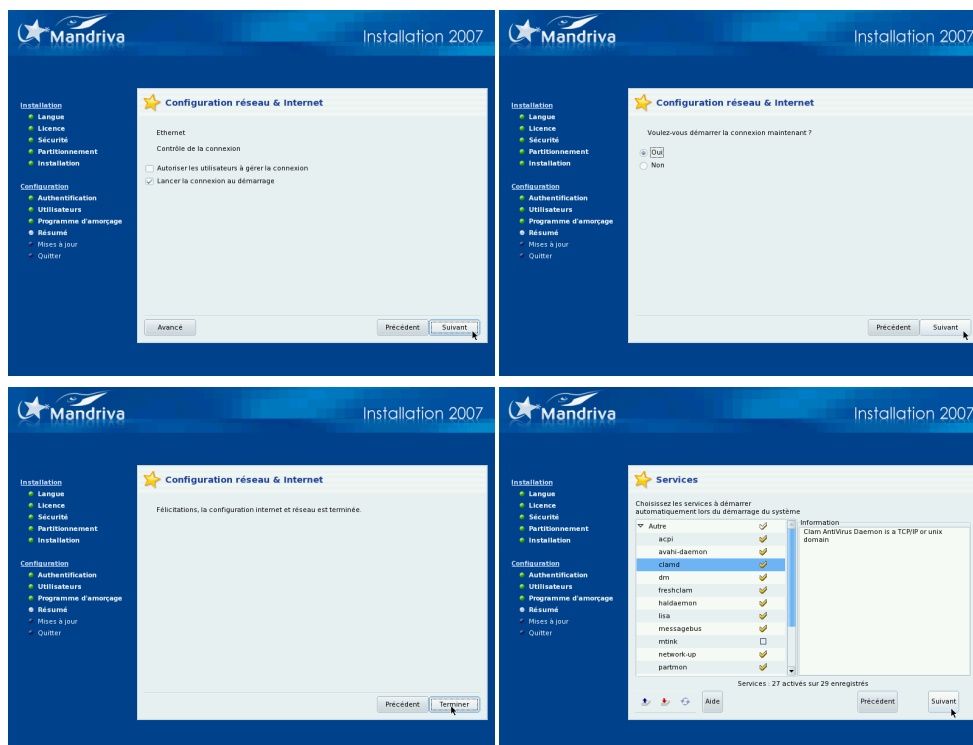


Vient ensuite la configuration du réseau. La solution la plus simple pour mettre en réseau un ordinateur de nos jours est d'utiliser une carte Ethernet ou le port Ethernet intégré à la carte mère (toutes les cartes mères récentes en disposent d'un). Cela permet de réaliser un petit réseau familial, et de pouvoir utiliser un modem/routeur pour se connecter à Internet de manière quasi automatique et sécurisée.

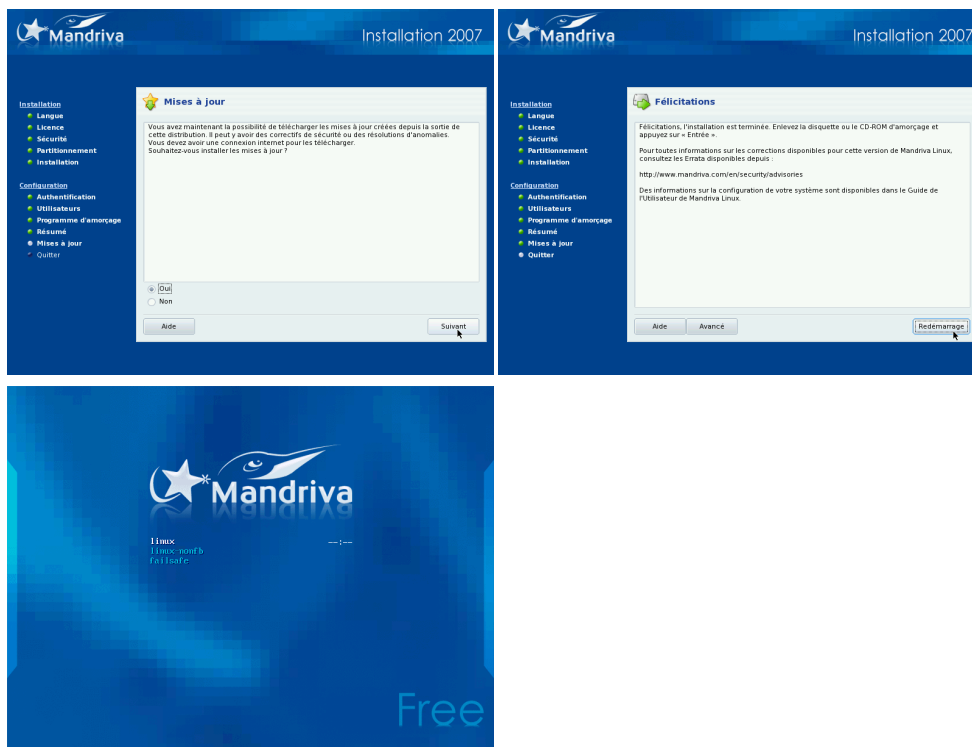
C'est cette configuration qui est décrite dans la suite : le premier et le deuxième écran permettent de choisir le type et le modèle de l'adaptateur réseau utilisé, et le troisième écran permet d'activer le protocole de configuration réseau DHCP. Le quatrième écran vous permet quant à lui de définir le nom de votre machine sur le réseau et d'utiliser le protocole DHCP pour récupérer les adresses des serveurs de noms de votre fournisseur d'accès :



Les écrans suivants vous demandent la confirmation du démarrage automatique des fonctionnalités réseau au démarrage de la machine et du démarrage du réseau pour la suite de l'installation (recommandé). Le quatrième écran vous présente la liste des services système à activer. Soyez assez strict sur ce sujet, en général un particulier n'a besoin d'aucun de ces services :



L'installation se termine ensuite par une mise à jour du système via le réseau (recommandé). Vous devrez ensuite redémarrer votre ordinateur, qui vous proposera alors la liste des systèmes d'exploitation installés et les modes dégradés pour le démarrage en cas de réparation :



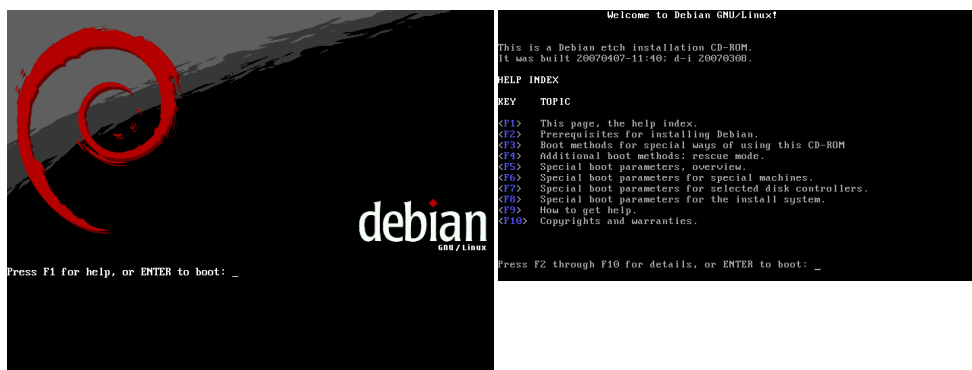
Vous aurez droit à remplir un petit questionnaire lors du premier démarrage de votre système Linux. Faites-en ce que vous voulez. Si le son ne fonctionne pas automatiquement après le premier démarrage, connectez-vous en tant qu'administrateur et exécutez le programme alsacnf.

III-F - Installation de la Debian 4.0r1

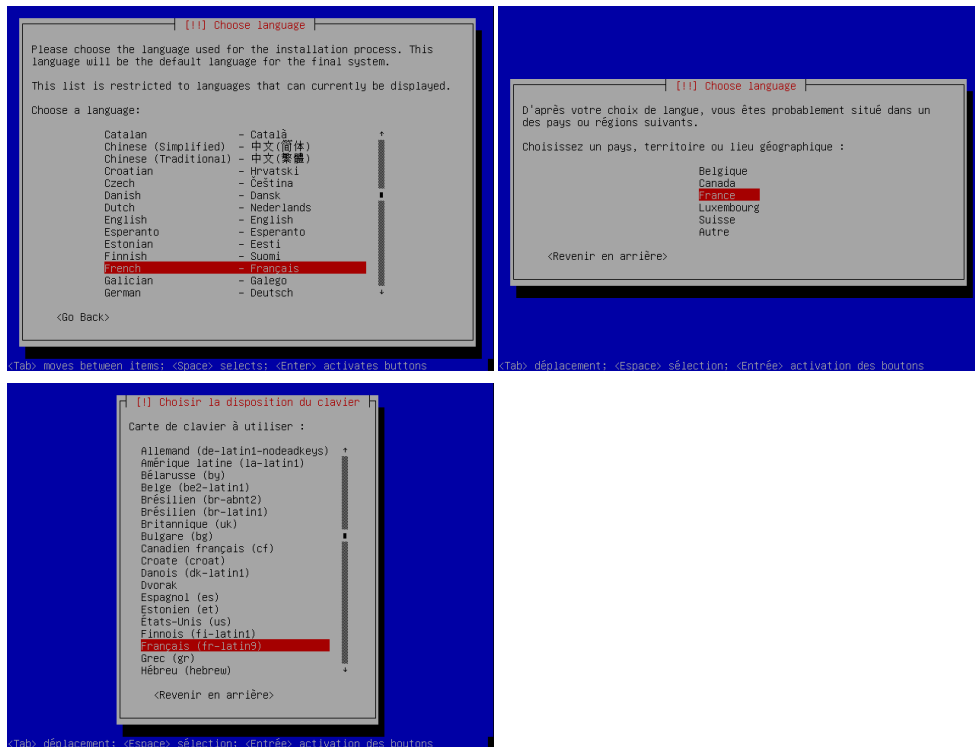
Nous allons présenter dans cette section la procédure d'installation de la distribution Debian Etch 4.0r1. La distribution Debian a une réputation justifiée de très grande stabilité, en contrepartie de laquelle les sorties des versions majeures sont assez espacées dans le temps.

L'installation de la Debian est relativement rustique, mais elle fournit des outils de configuration permettant aux utilisateurs une installation sans avoir à connaître les arcanes du système et de la configuration des logiciels installés.

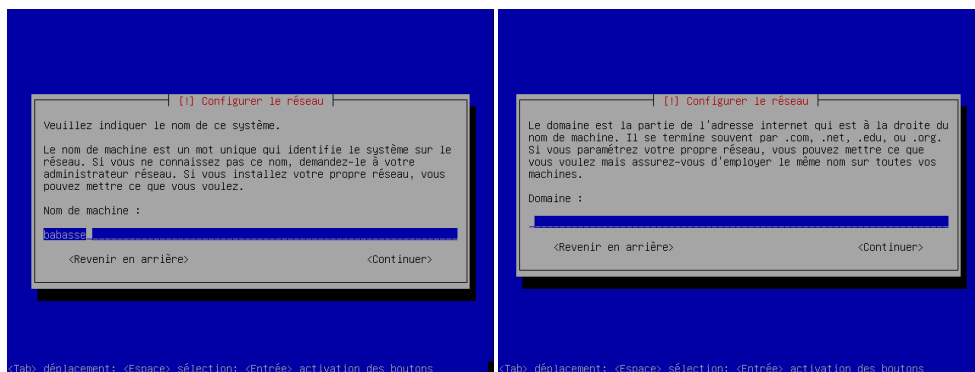
L'installation de la Debian commence par un écran d'accueil vous proposant de démarrer le système. L'appui sur la touche « F1 » vous donnera accès à un écran d'aide permettant d'obtenir tous les paramètres de démarrage du système. Ces paramètres pourront être utiles pour démarrer Linux sur certaines machines exotiques. Vous pouvez essayer dans un premier temps le démarrage en utilisant le noyau par défaut en appuyant sur « Entrée » :



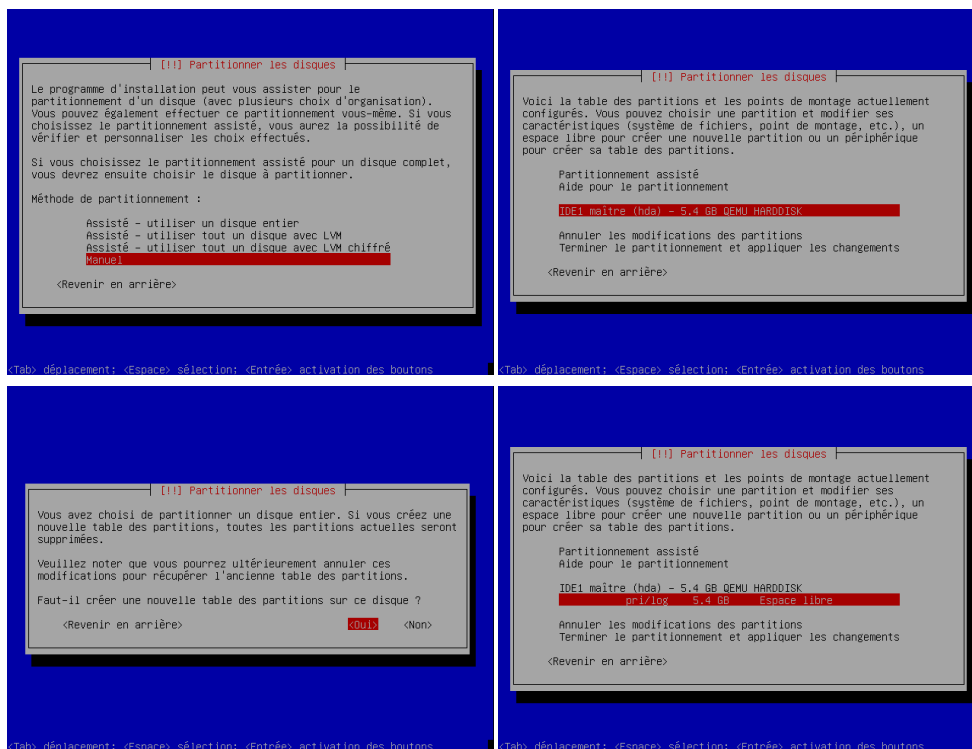
Le programme d'installation commence alors par vous demander la langue du système, le pays où vous vous trouvez, et le type de clavier utilisé dans ce pays :



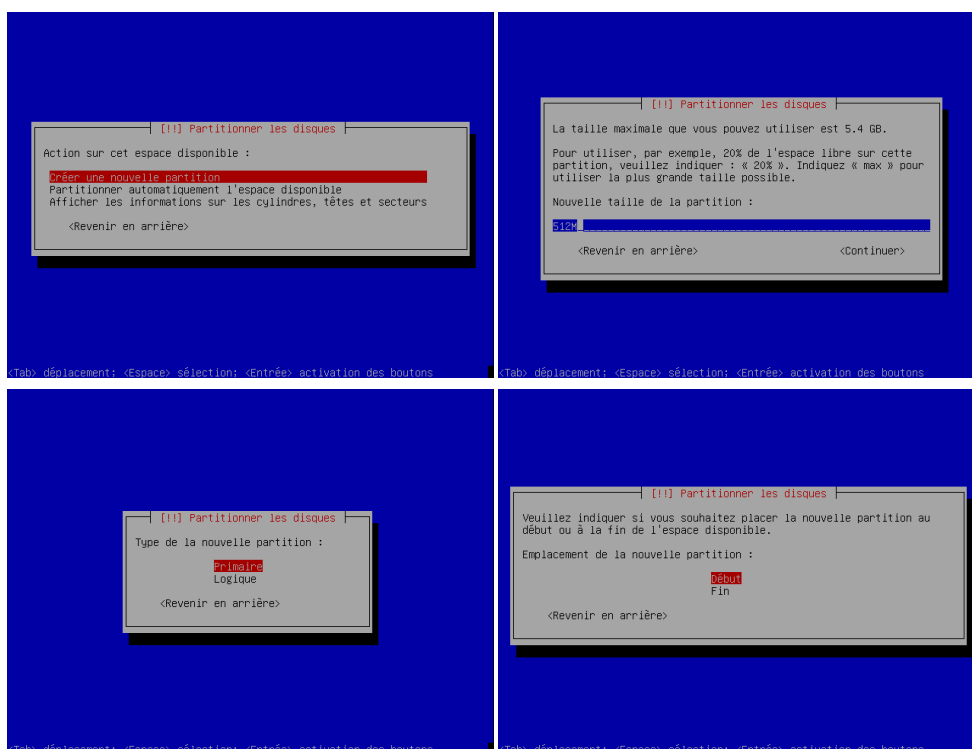
Il vous faut ensuite indiquer le nom de votre machine et, éventuellement, le nom du domaine auquel elle appartient :



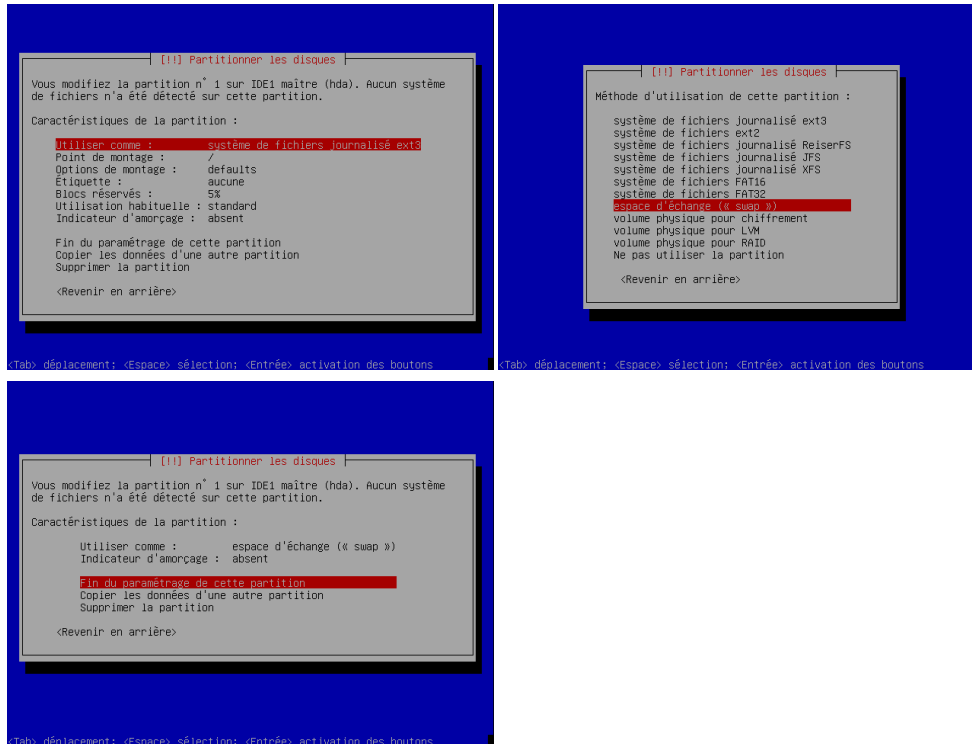
Vient ensuite la phase de partitionnement du disque. Ce document décrit la procédure de partitionnement manuelle, que l'on peut activer en choisissant la méthode de partitionnement manuelle. Cette méthode vous permettra de choisir le disque à partitionner, et d'éventuellement y créer une nouvelle table des partitions. Vous pourrez alors sélectionner l'espace libre ainsi rendu disponible afin d'y créer vos partitions :



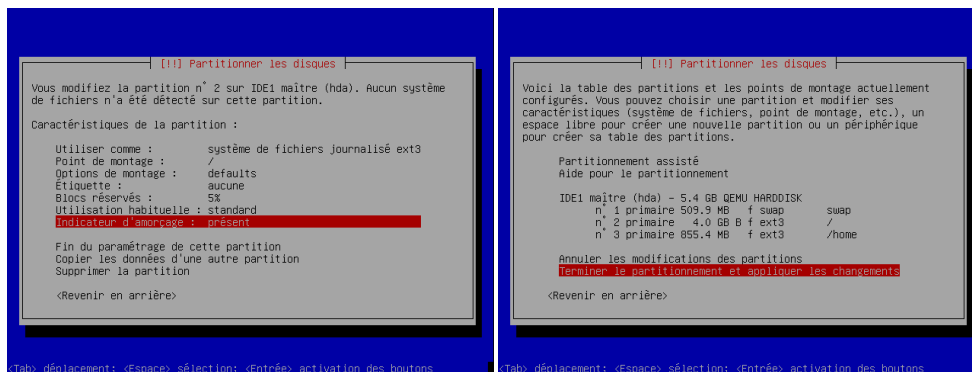
La création d'une partition en soi se fait simplement en choisissant l'option « Créer une nouvelle partition ». Vous pourrez alors indiquer la taille de cette partition, son type, et son emplacement dans la table des partitions :



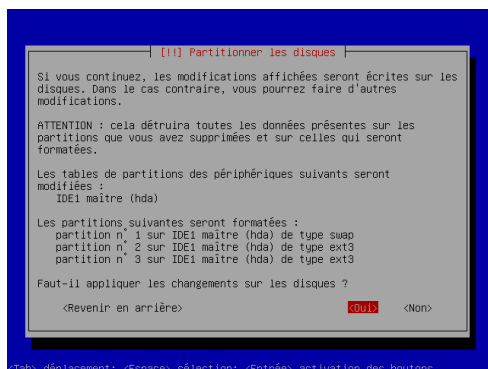
Pour chaque partition ainsi définie, vous devez indiquer sa nature et le système de fichiers utilisé. Les écrans suivants montrent comment définir une partition d'échange :



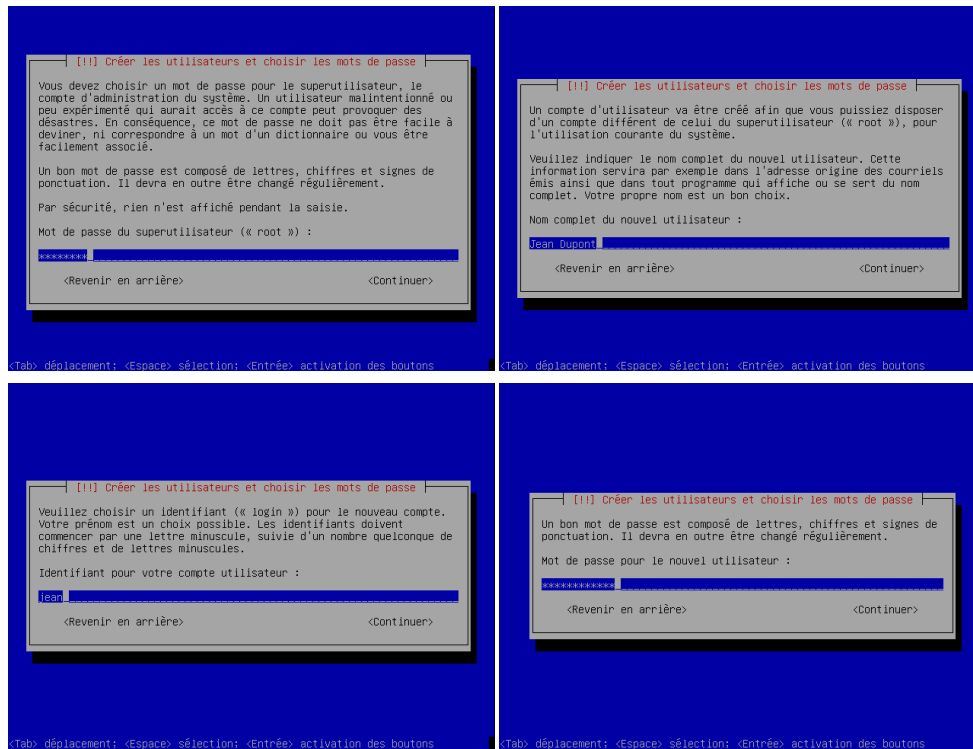
Vous procéderez de même pour définir la partition racine de votre système, ainsi que les éventuelles autres partitions pour les différents points de montage. Vous devrez indiquer ensuite que la partition racine est amorçable en fixant son indicateur d'amorçage :



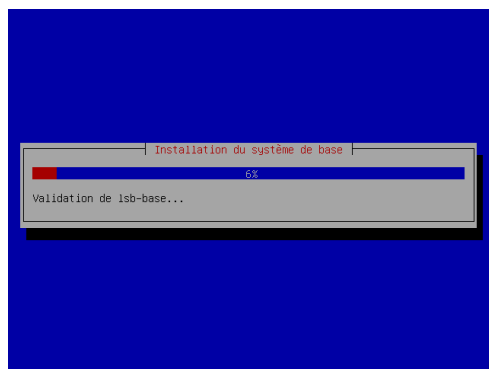
Vous pouvez alors valider les modifications effectuées et écrire la table de partition sur le disque dur :



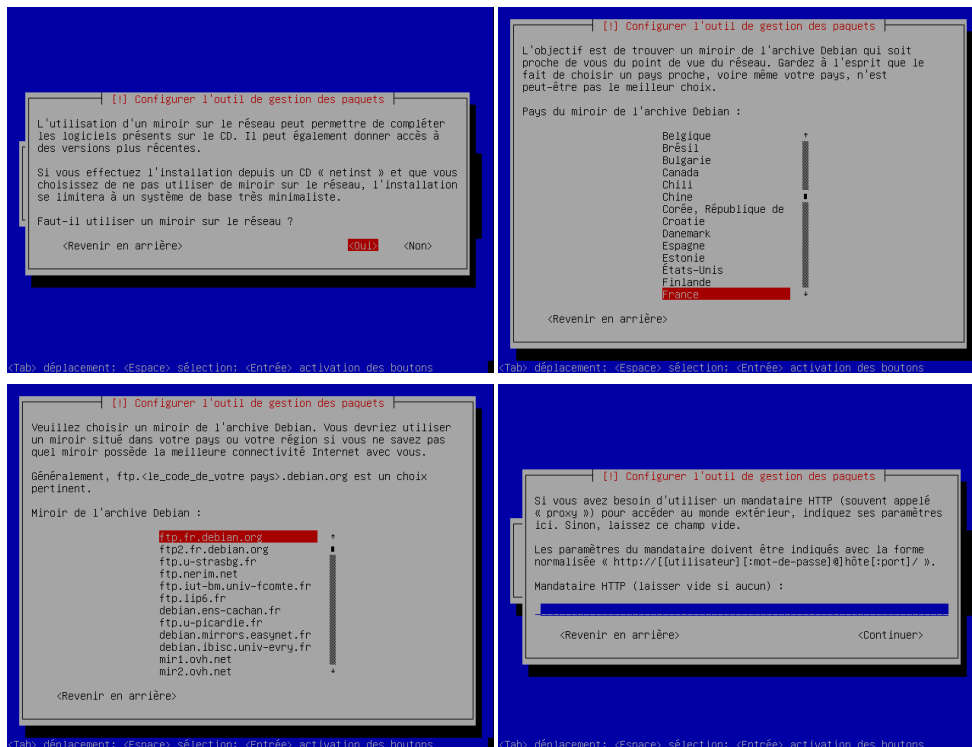
Le programme d'installation vous demandera ensuite de définir le mot de passe de l'administrateur, et de confirmer ce mot de passe. Vous pourrez ensuite ajouter un premier utilisateur, en donnant son nom complet, puis le login et le mot de passe qu'il devra utiliser pour se connecter :



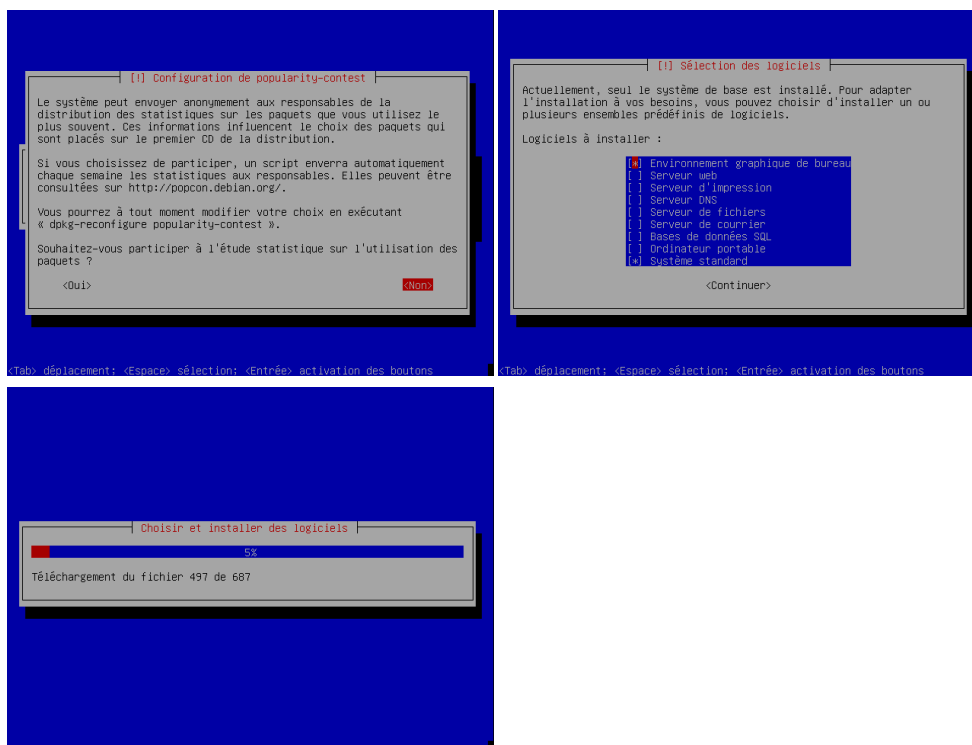
Le programme d'installation poursuivra en copiant les fichiers du système de base sur le disque dur :



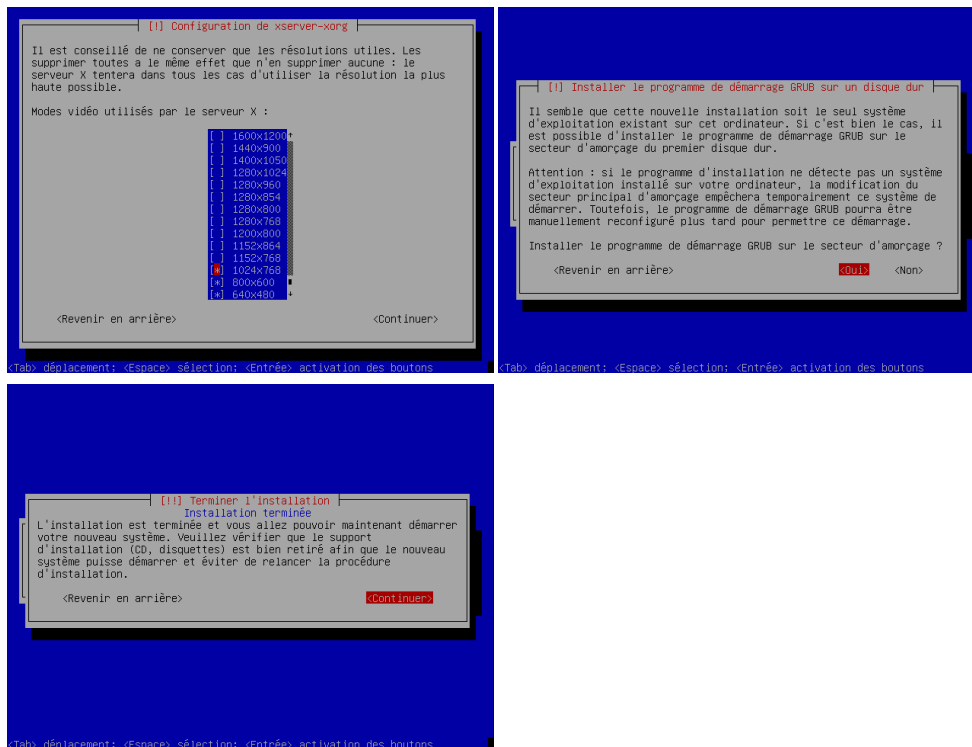
Vous pourrez ensuite choisir le miroir le plus proche de vous sur Internet pour la récupération des mises à jour et des paquetages manquants :



Le programme d'installation va ensuite vous demander de choisir les paquetages que vous désirez installer, puis les copiera sur le disque dur. Afin de déterminer quels sont les paquetages les plus importants et d'améliorer la facilité d'installation de Debian il vous proposera au préalable de participer à un sondage de popularité des paquetages :



Une fois le système installé, le programme d'installation passe à la configuration du serveur X et du gestionnaire d'amorçage GRUB avant de terminer l'installation. Pour le serveur X, vous devrez indiquer les résolutions que vous comptez utiliser. En ce qui concerne le gestionnaire d'amorçage, vous pouvez indiquer si vous désirez l'installer sur le secteur d'amorçage ou non :

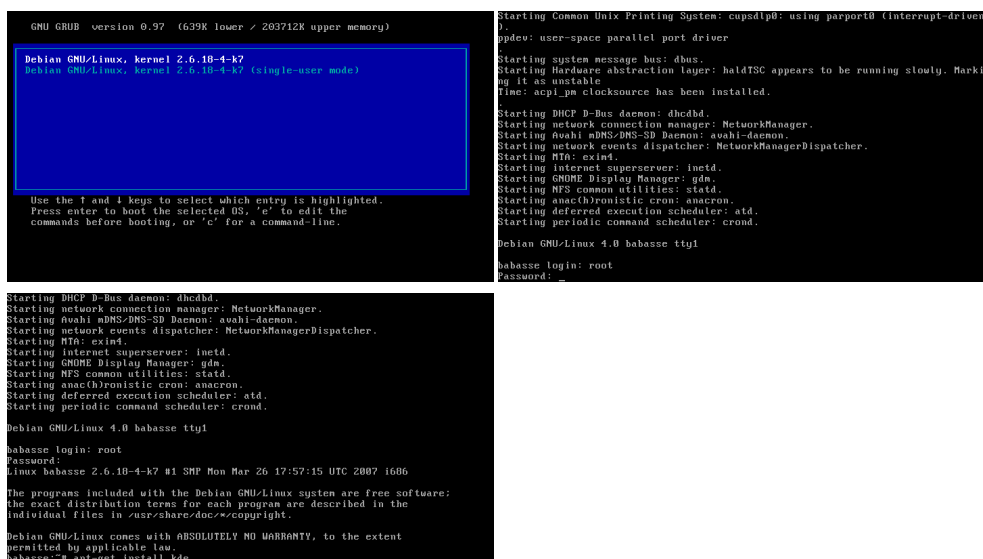


L'installation telle que présentée ici n'installe que l'environnement graphique Gnome par défaut. Si vous désirez installer l'environnement graphique, il vous faudra vous connecter sous le compte administrateur et compléter l'installation manuellement. Cela se fait en ligne de commande, à l'aide du gestionnaire de paquetage de la Debian, APT (abréviation de « A Package Tool »).

Pour cela, redémarrez, basculez sur le premier terminal virtuel avec les trois touches CTRL+Alt+F1, loguez-vous en tant qu'administrateur, et exécutez la commande suivante : en utilisant le gestionnaire de Ainsi, vous devrez exécuter la commande suivante :

```
apt-get install kde
```

Les écrans suivants illustrent cette opération :



À la suite de ces opérations, vous pouvez redémarrer votre machine. Vous devriez alors avoir un système Debian utilisable et opérationnel.

III-G - Installation de la Slackware 12.0

Nous allons présenter dans cette section la procédure d'installation de la distribution Slackware 12.0. Tout comme Debian, la Slackware vise une grande stabilité et est par conséquent également assez conservatrice dans ses choix de logiciels. Toutefois, les versions officielles sortent à une fréquence supérieure à celle de la Debian, car le temps d'intégration et de validation est plus court que pour la Debian. Cela est dû au fait que la Slackware a pour vocation d'être une distribution à taille humaine, et se restreint donc aux logiciels les plus courants et fournit un nombre de logiciels nettement moindre. De ce fait, les versions des logiciels fournis sont souvent plus récentes, ce qui constitue un bon compromis.

Contrairement aux autres distributions en revanche, la Slackware ne fournit pas d'outil de configuration spécialisé. Comme nous allons le voir, elle requiert donc une connaissance plus approfondie de Linux et des principaux logiciels de la part de l'utilisateur.

L'installation de la Slackware commence par un écran d'accueil vous proposant de démarrer le système. Vous pouvez choisir plusieurs noyaux en fonction de votre configuration matérielle, la liste des noyaux disponibles étant accessible en appuyant sur la touche « F3 ». Il est recommandé de choisir le noyau proposé par défaut, à savoir le noyau « hugesmp.s », et d'appuyer sur la touche « Entrée » pour démarrer Linux. Attention, le clavier étant en configuration anglaise au démarrage, vous trouverez le point sur la touche « : ».

```

boot master: QEMU CD-ROM ATAPI-4 CD-Rom-BUD-Rom
Booting from CD-Rom...
ISOLINUX 3.36 2007-02-10 Copyright (C) 1994-2007 H. Peter Anvin
Welcome to Slackware version 12.0 (Linux kernel 2.6.21.5)!

If you need to pass extra parameters to the kernel, enter them at the prompt
below after the name of the kernel to boot (huges.s etc). NOTE: If your machine
is not at least a Pentium-Pro, you must boot and install with the huges.s
kernel, not the hugesmp.s kernel! For older machines, use "huges.s" at the
boot prompt.

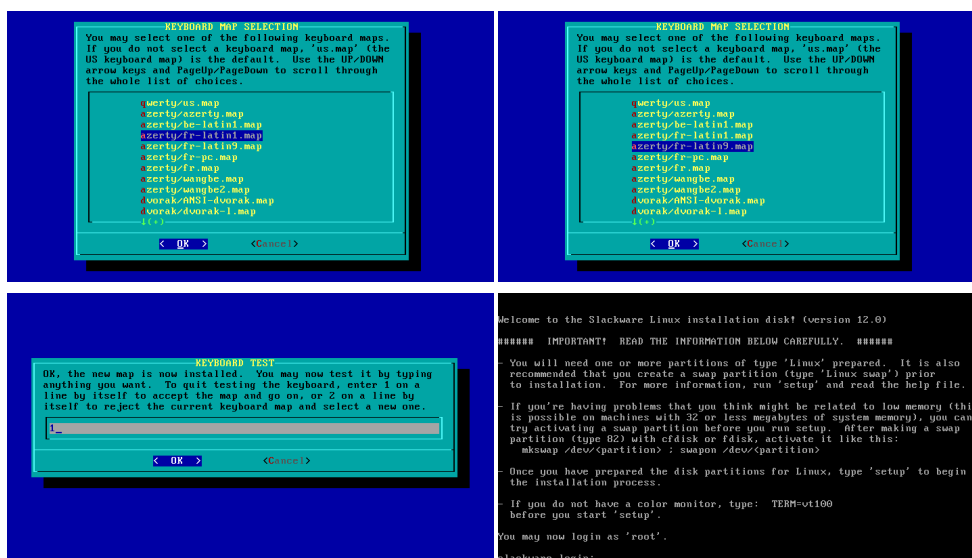
In a pinch, you can boot your system from here with a command like:
boot: hugesmp.s root=/dev/hda1 rdinit= ro

In the example above, /dev/hda1 is the / Linux partition.

This prompt is just for entering extra parameters. If you don't need to enter
any parameters, hit ENTER to boot the default kernel "hugesmp.s" or press [F2]
for a listing of more kernel choices.

boot:
  
```

Le système propose de configurer le clavier dès le premier démarrage. Lorsque vous aurez choisi votre clavier, vous pourrez faire un test de celui-ci, comme le montre le troisième écran. Si la configuration du clavier vous convient, vous pouvez l'accepter en saisissant '1' sur une ligne vide et en validant. Vous aurez alors accès à l'écran de login de la Slackware :



Le système ne possédant pas encore de mot de passe à ce stade de l'installation, vous pouvez vous connecter simplement en appuyant sur « Entrée ». L'étape suivante est de partitionner le disque, en utilisant la commande `fdisk`. La manière d'utiliser cette commande ayant déjà été décrite dans la section décrivant une installation générique, nous n'entrerons pas plus dans les détails ici :

```

- If you do not have a color monitor, type: TERM=vt100
  before you start 'setup'.

You may now login as 'root'.

Slackware login: root

Linux 2.6.21.5-smp.

If you're upgrading an existing Slackware system, you might want to
remove old packages before you run 'setup' to install the new ones. If
you don't, your system will still work but there might be some old files
left laying around on your drive.

Just mount your Linux partitions under /mnt and type 'pkgtool'. If you
don't know how to mount your partitions, type 'pkgtool' and it will tell
you how it's done.

To partition your hard drive(s), use 'cfdisk' or 'fdisk'.
To activate FCPCI/Carbus devices needed for installation, type 'pcmcia'.
To activate network devices needed for installation, type 'network'.
To start the main installation, type 'setup'.

root@slackware:~# fdisk /dev/hda

Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that, of course, the previous
content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help): n
Command action
  e extended
  p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-652, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-652, default 652): +512M
Command (m for help): p

Disk /dev/hda: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1            1           63       506016   03  Linux

Command (m for help):

```

Pour chaque partition, il vous faut indiquer son type et, si vous ne comptez pas installer le gestionnaire d'amorçage sur le secteur principal du disque dur, la rendre amorçable. Vous pourrez ensuite écrire la nouvelle table de partitions sur le disque :

```

content won't be recoverable.
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help): n
Command action
  e extended
  p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-652, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-652, default 652): +512M
Command (m for help): p

Disk /dev/hda: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1            1           63       506016   03  Linux

Command (m for help): t
Selected partition 1
Hex code (type l to list codes): 82
Partition number (1-4): 2
Command (m for help): n
Command action
  e extended
  p primary partition (1-4)
p
Partition number (1-4): 3
First cylinder (551-652, default 551):
Using default value 551
Last cylinder or +size or +sizeM or +sizeK (551-652, default 652):
Using default value 652
Command (m for help): p

Disk /dev/hda: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

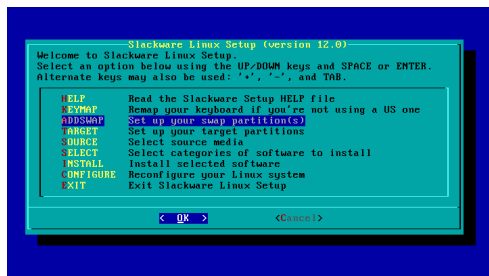
   Device Boot      Start         End      Blocks   Id  System
/dev/hda1            1           63       506016   82  Linux swap
/dev/hda2            64          550      3911827*  03  Linux
/dev/hda3           551          652       819315   03  Linux

Command (m for help): w

```

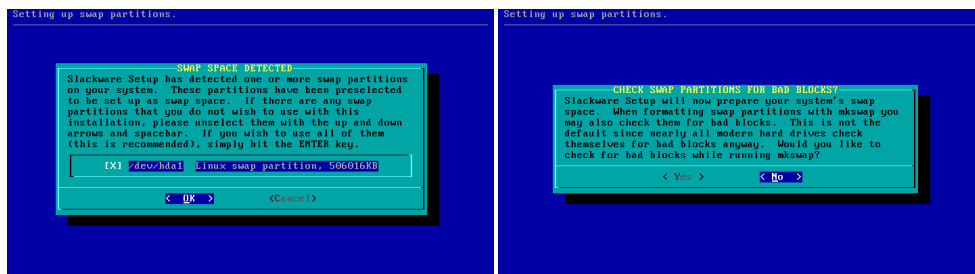
Note : Pour rappel, la création d'une nouvelle partition se fait avec la commande 'n' de `fdisk`, et la visualisation des partitions existantes se fait avec la commande 'p'. Pour changer le type d'une partition, il faut utiliser la commande 't', et pour la rendre amorçable, il faut utiliser la commande 'a'. Enfin, pour écrire la table des partitions sur le disque, il faut utiliser la commande 'w'.

Le programme d'installation de la Slackware peut ensuite être lancé en exécutant la commande `setup`. Vous accéderez alors à l'écran suivant, qui vous permettra d'exécuter les différentes phases de l'installation :

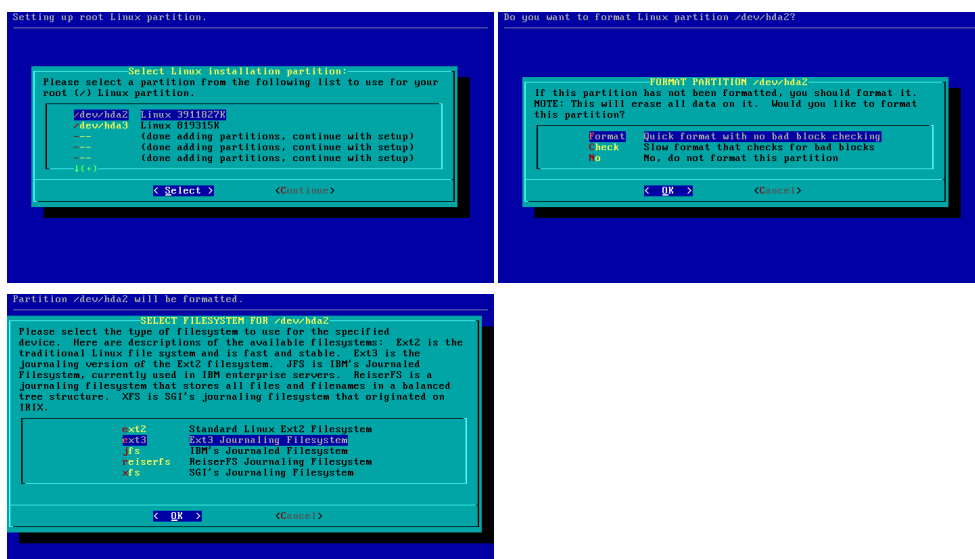


L'option « HELP » affiche un écran d'aide. L'option « KEYMAP » permet de choisir la disposition du clavier à utiliser, mais cette opération étant déjà faite, il est inutile de la refaire. Les options suivantes permettent de définir la partition d'échange du système, de choisir les partitions cibles et les supports d'installation, de choisir et d'installer les logiciels, et enfin de configurer le système. Toutes ces options sont exécutées automatiquement les unes après les autres par le système d'installation, aussi suffit-il de commencer par l'option « ADDSWAP ».

Le programme d'installation utilise les deux écrans suivants pour configurer la partition d'échange. La vérification des blocs de la partition d'échange est fortement recommandée, une défaillance matérielle à ce niveau pouvant planter le système ultérieurement et détruire l'intégralité de ses données :

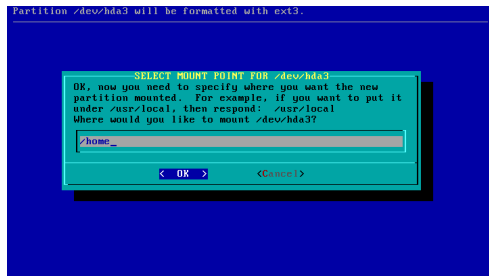


Le programme d'installation va alors relire la table des partitions et vous demander comment utiliser chacune d'elles. La première partition à définir est la partition racine du système, vers laquelle l'installation doit se faire. Pour chaque partition ainsi définie, il faut créer un système de fichiers, en indiquant son type et le nombre d'inodes dans le système de fichiers en fonction de sa taille (le quatrième écran vous demande donc d'indiquer la densité d'inodes du système de fichiers par rapport à sa taille) :

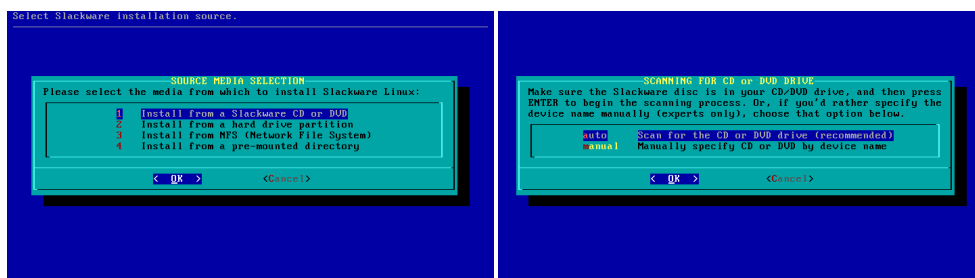


Cette procédure est à répéter pour chaque partition. Toutefois, les autres partitions que la partition racine nécessitent un point de montage, qui doit donc être spécifié. L'écran suivant montre comment définir le point de montage d'une

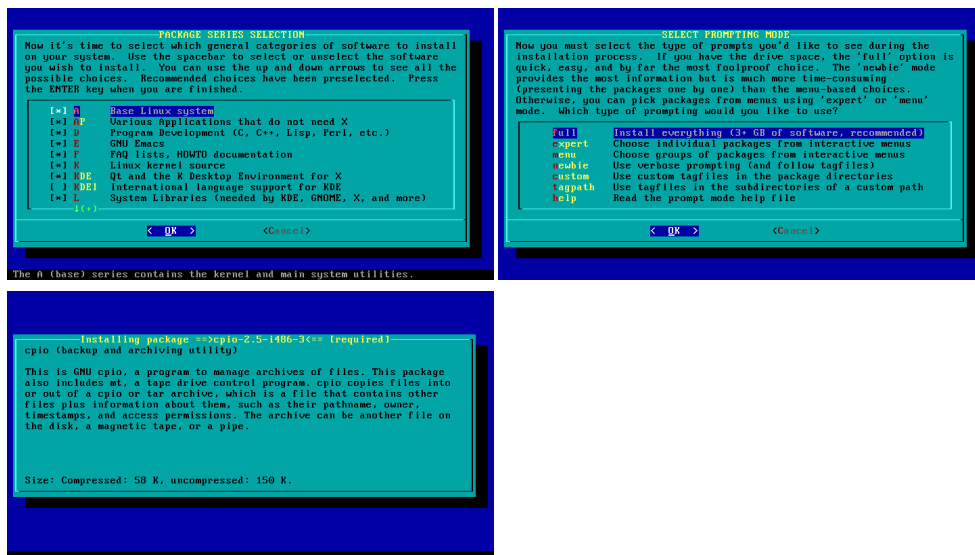
partition. Vous devez y indiquer le chemin du point de montage de cette partition. Par exemple, si vous dédiez une partition pour les données des utilisateurs (ce qui est recommandé), vous devez saisir le chemin `/home/` :



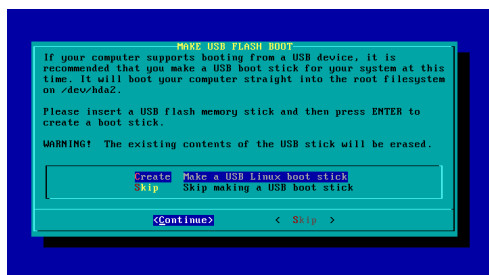
Le programme d'installation demande ensuite où il peut trouver le support d'installation. Dans notre cas, l'installation se fait à partir d'un CD-ROM ou d'un DVD-ROM. Le programme d'installation détectera automatiquement le lecteur dans lequel le média d'installation se trouve :



La suite de l'installation se fait alors classiquement : il suffit de sélectionner les différentes catégories de logiciels à installer, et le mode de choix des paquetages. Pendant la copie des fichiers, des informations sont affichées sur le paquetage en cours d'installation :



À la fin de l'installation, le programme d'installation demande s'il faut créer une clé USB de démarrage de secours :



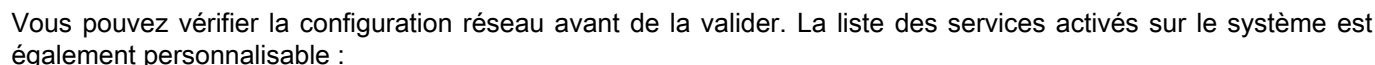
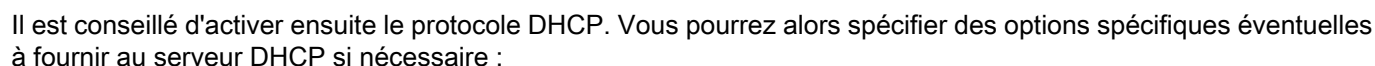
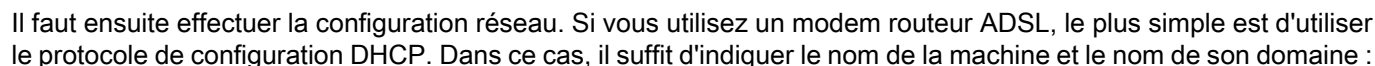
La configuration du modem est ensuite proposée. Il s'agit là des modems analogiques classiques, et non des modems ADSL ou câble par exemple :



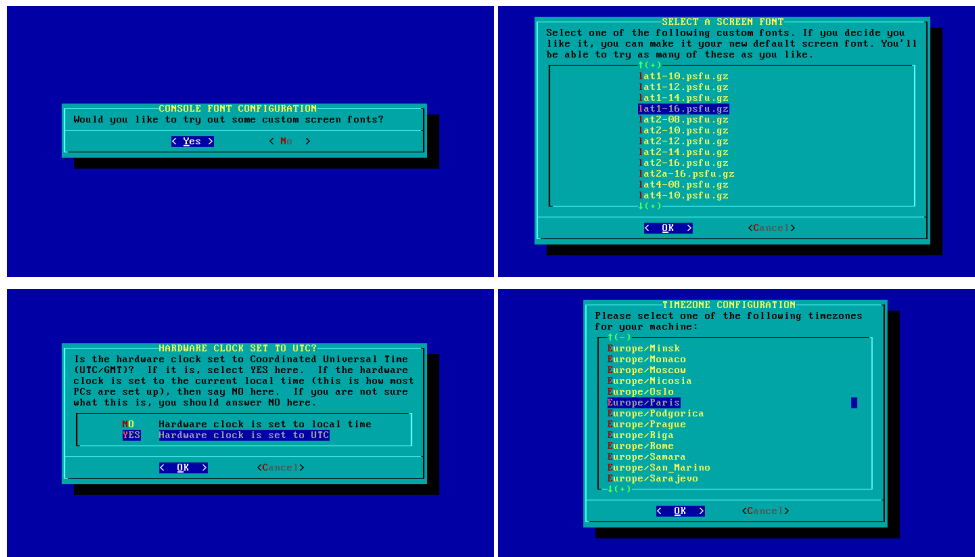
L'étape suivante consiste à installer le gestionnaire d'amorçage du système. Par défaut, la Slackware utilise LILO. La configuration automatique fonctionne généralement bien, aussi pouvez-vous choisir l'option correspondante. Le programme d'installation demande ensuite la résolution de l'écran au démarrage, les options complémentaires qui doivent être passées au noyau, et la zone du disque où LILO doit être installé :



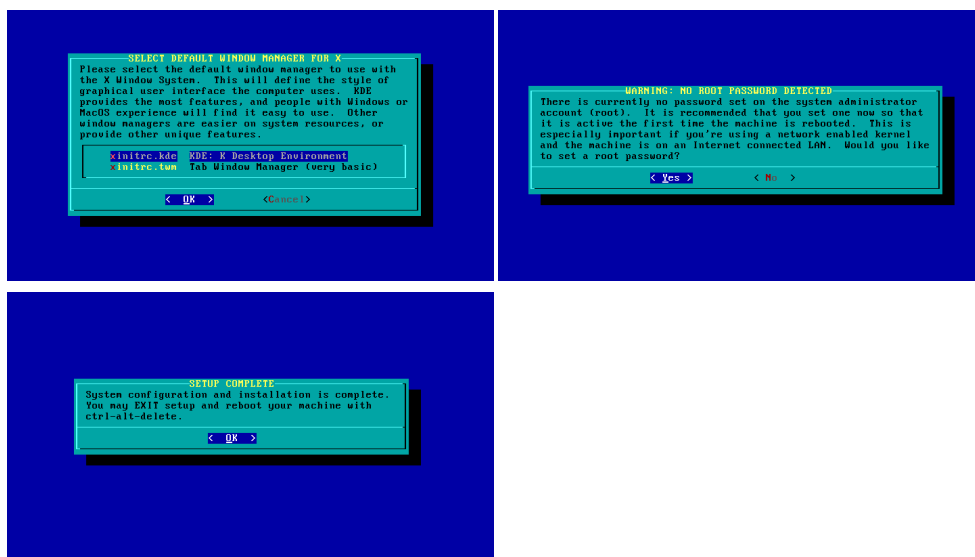
La configuration de la souris est ensuite proposée. Vous devez indiquer le type de souris dont vous disposez, et si vous voulez lancer le gestionnaire de souris pour le mode texte GPM :



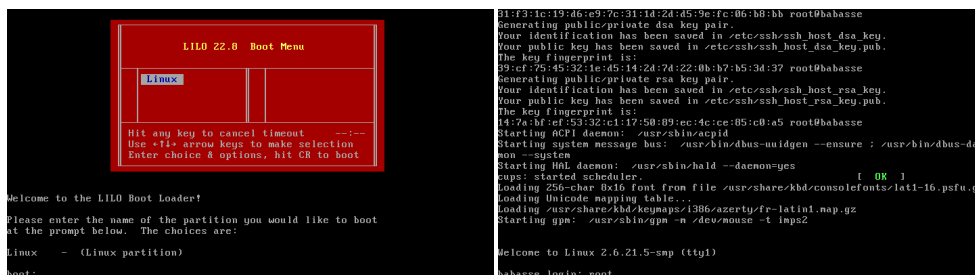
La configuration du système se termine par le choix de la police de caractères pour la console et par le réglage des paramètres horaires (horloge système en temps local ou en temps universel, et fuseau horaire de la machine) :



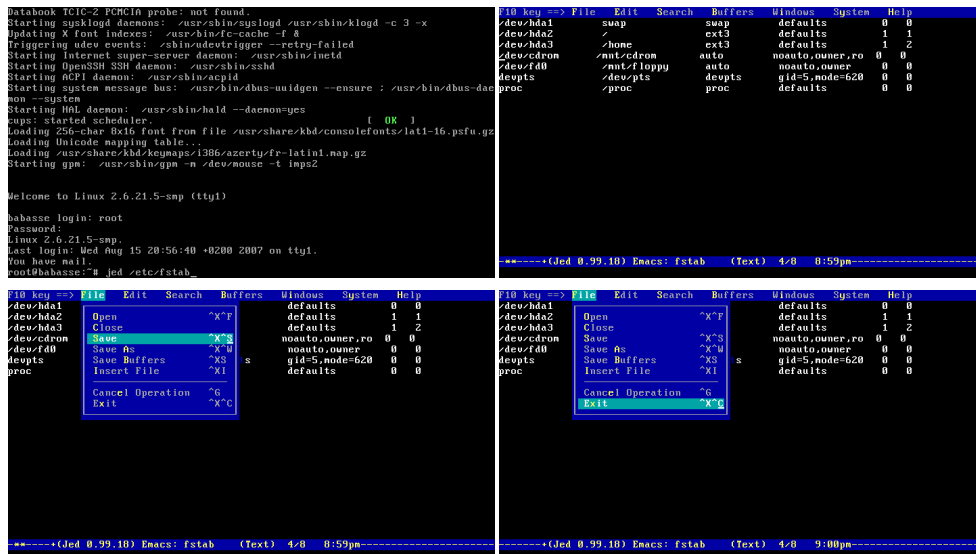
Le programme d'installation demande enfin le gestionnaire de connexion graphique à utiliser et le mot de passe de l'administrateur du système avant de redémarrer la machine :



L'installation doit toutefois être complétée par quelques opérations manuelles après le redémarrage. Pour cela, il faut d'abord se connecter sur la console en tant qu'administrateur :



Le programme d'installation laisse commentées les entrées permettant de monter les lecteurs de CD-ROM et de DVD-ROM dans le fichier `/etc/fstab`. Il faut donc décommenter ces entrées et éventuellement les corriger pour pouvoir accéder au lecteur de CD-ROM et terminer l'installation. Pour cela, vous pouvez utiliser par exemple l'éditeur de fichier `jed` et supprimer la marque de commentaire devant la ligne correspondante au lecteur de CD-ROM :



```

Databook TCIC-2 PCMCIA probe: not found.
Starting sysklogd daemons: /usr/sbin/syslogd /usr/sbin/klogd -c 3 -x
Updating X font indexes: /usr/bin/X-csache -f 1
Triggering udev events: /sbin/udevtrigger --retry-failed
Starting Internet super-server daemon: /usr/sbin/inetd
Starting OpenSSH SSH daemon: /usr/sbin/sshd
Starting ACPI daemon: /usr/sbin/acpid
Starting system message bus: /usr/bin/dbus-uidgen --ensure : /usr/bin/dbus-daemon --system
Starting HAL daemon: /usr/sbin/halld --daemon=yes
 cups: started scheduler.
Loading Z56-char 8x16 font from file /usr/share/kbd/consolefonts/lat1-16.psfu.gz
Loading Unicode mapping table...
Loading /usr/share/kbd/keymaps/i386/azerty/fr-variant1.map.gz
Starting gpm: /usr/sbin/gpm -n /dev/mouse -t imps2

Welcome to Linux 2.6.21.5-smp (tty1)
babasse login: root
Password:
Linux 2.6.21.5-smp.
Last login: Wed Aug 15 20:56:40 +0200 2007 on tty1.
You have mail.
root@babasse:~# jed /etc/fstab

F10 key=> File Edit Search Buffers Windows System Help
/dev/hda1 swap swap defaults 0 0
/dev/hda2 Open "X" F defaults 1 1
/dev/hda3 Close "X" F defaults 1 2
/dev/cdrom Save As "X" S noauto,owner,ro 0 0
/dev/fd0 Save As "X" W noauto,owner 0 0
devpts Save Buffers "X" S gid=5,mode=620 0 0
proc Insert File "X" I defaults 0 0
Cancel Operation ^G
Exit ^X C

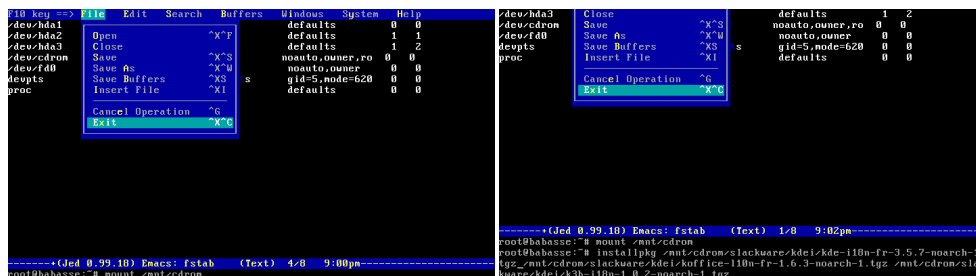
F10 key=> File Edit Search Buffers Windows System Help
/dev/hda1 swap swap defaults 0 0
/dev/hda2 Open "X" F defaults 1 1
/dev/hda3 Close "X" F defaults 1 2
/dev/cdrom Save As "X" S noauto,owner,ro 0 0
/dev/fd0 Save As "X" W noauto,owner 0 0
devpts Save Buffers "X" S gid=5,mode=620 0 0
proc Insert File "X" I defaults 0 0
Cancel Operation ^G
Exit ^X C

F10 key=> File Edit Search Buffers Windows System Help
/dev/hda1 swap swap defaults 0 0
/dev/hda2 Open "X" F defaults 1 1
/dev/hda3 Close "X" F defaults 1 2
/dev/cdrom Save As "X" S noauto,owner,ro 0 0
/dev/fd0 Save As "X" W noauto,owner 0 0
devpts Save Buffers "X" S gid=5,mode=620 0 0
proc Insert File "X" I defaults 0 0
Cancel Operation ^G
Exit ^X C

F10 key=> File Edit Search Buffers Windows System Help
/dev/hda1 swap swap defaults 0 0
/dev/hda2 Open "X" F defaults 1 1
/dev/hda3 Close "X" F defaults 1 2
/dev/cdrom Save As "X" S noauto,owner,ro 0 0
/dev/fd0 Save As "X" W noauto,owner 0 0
devpts Save Buffers "X" S gid=5,mode=620 0 0
proc Insert File "X" I defaults 0 0
Cancel Operation ^G
Exit ^X C

root@babasse:~# mount /mnt/cdrom
root@babasse:~# installpkg /mnt/cdrom/slackware/kde1/kde-110n-fr-3.5.7-noarch-1.
tgz /mnt/cdrom/slackware/kde1/koffice-110n-fr-1.6.3-noarch-1.tgz /mnt/cdrom/sla
ckware/kde1/k3b-110n-1.0.2-noarch-1.tgz
  
```

Il est alors possible de monter le disque d'installation dans le répertoire `/mnt/cdrom/` et d'installer les fichiers de prise en charge de la langue française pour l'environnement KDE :



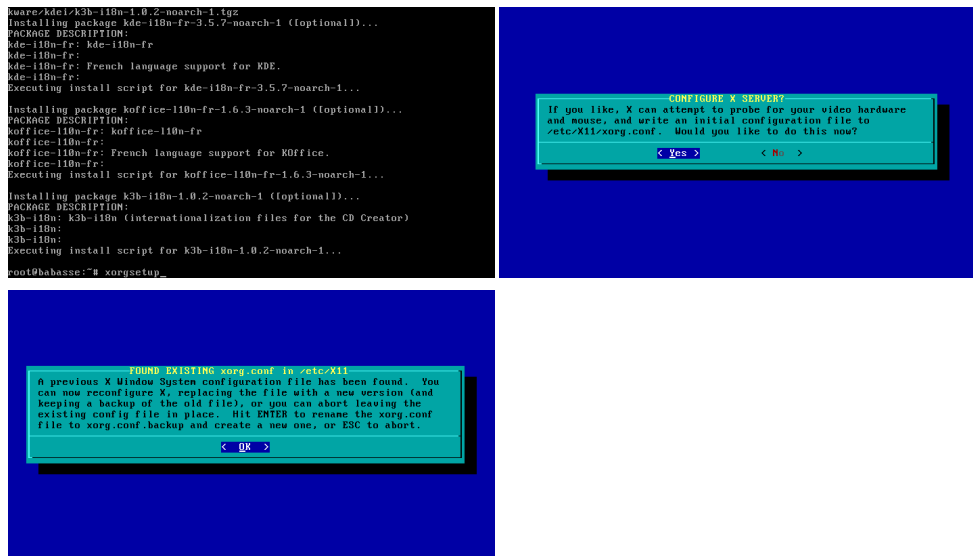
```

F10 key=> File Edit Search Buffers Windows System Help
/dev/hda1 swap swap defaults 0 0
/dev/hda2 Open "X" F defaults 1 1
/dev/hda3 Close "X" F defaults 1 2
/dev/cdrom Save As "X" S noauto,owner,ro 0 0
/dev/fd0 Save As "X" W noauto,owner 0 0
devpts Save Buffers "X" S gid=5,mode=620 0 0
proc Insert File "X" I defaults 0 0
Cancel Operation ^G
Exit ^X C

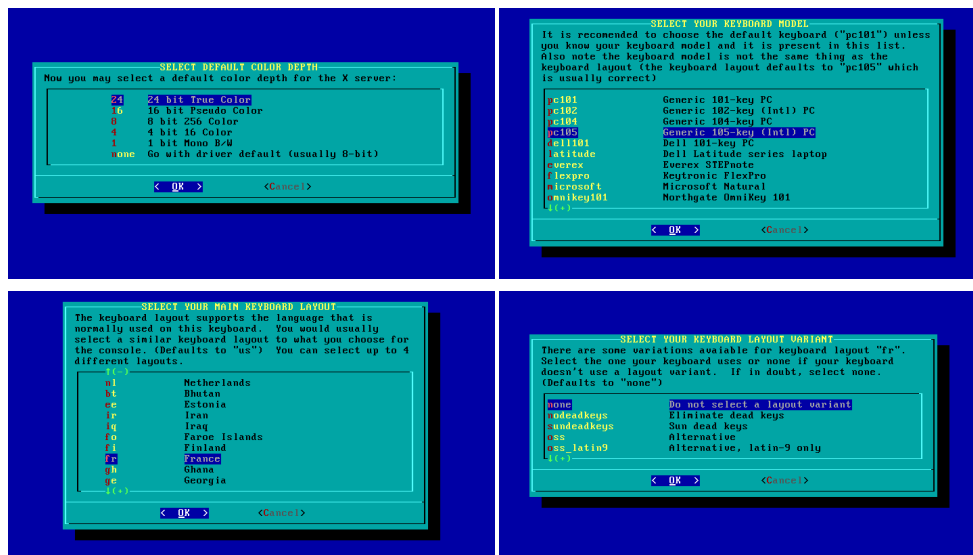
F10 key=> File Edit Search Buffers Windows System Help
/dev/hda1 swap swap defaults 0 0
/dev/hda2 Open "X" F defaults 1 1
/dev/hda3 Close "X" F defaults 1 2
/dev/cdrom Save As "X" S noauto,owner,ro 0 0
/dev/fd0 Save As "X" W noauto,owner 0 0
devpts Save Buffers "X" S gid=5,mode=620 0 0
proc Insert File "X" I defaults 0 0
Cancel Operation ^G
Exit ^X C

root@babasse:~# mount /mnt/cdrom
root@babasse:~# installpkg /mnt/cdrom/slackware/kde1/kde-110n-fr-3.5.7-noarch-1.
tgz /mnt/cdrom/slackware/kde1/koffice-110n-fr-1.6.3-noarch-1.tgz /mnt/cdrom/sla
ckware/kde1/k3b-110n-1.0.2-noarch-1.tgz
  
```

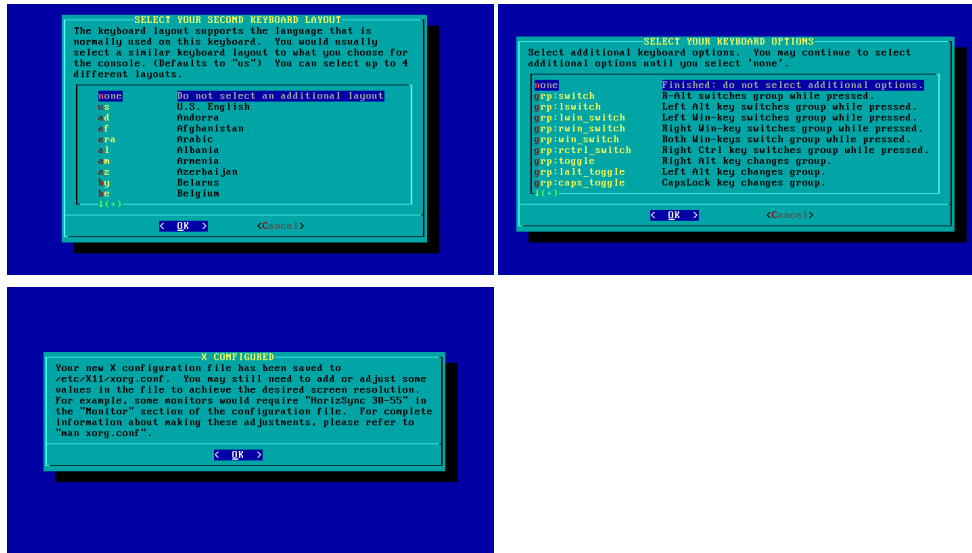
La configuration de l'environnement graphique X11 pour la Slackware se fait avec la commande `xorgsetup`. Cette commande vous demandera confirmation que vous voulez bien configurer X11, et affichera un avertissement pour vous indiquer que la configuration existante sera écrasée :



Le programme détectera ensuite la configuration matérielle et vous proposera ensuite diverses options pour la configuration du serveur X (nombre de couleurs de l'affichage par défaut et configuration principale du clavier) :



xorgsetup vous permet également de définir une configuration alternative pour le clavier, ainsi que de choisir des options sur le comportement des différentes touches. Une fois la configuration terminée, il vous signale que l'environnement X11 est prêt à être utilisé :



Comme il est plus confortable de démarrer l'environnement graphique automatiquement au démarrage de la machine, on devra modifier le niveau d'exécution par défaut du système (voir plus loin pour plus de détails à ce sujet). Pour cela, il suffit d'éditer le fichier `/etc/inittab`, par exemple avec l'éditeur `jed`. Il faut y remplacer la valeur '3' précédant le mot-clé « `initdefault` » par '4' :

```

#10 key => File Edit Search Buffers Windows System Help
# Default runlevel. (Do not set to 0 or 6)
id:4:~initdefault:

# System initialization (runs when system boots).
s1:S:sysinit:/etc/rc.d/rc.S

# Script to run when going single user (runlevel 1).
su:1S:wait:/etc/rc.d/rc.K

# Script to run when going multi user.
rc:2345:wait:/etc/rc.d/rc.M

# What to do at the "Three Finger Salute".
ca::ctrlaltdel:/sbin/shutdown -t5 -p now

# Runlevel 0 halts the system.
0:0:wait:/etc/rc.d/rc.0

# Runlevel 6 reboots the system.
6:6:wait:/etc/rc.d/rc.6

---(Jed 0.99.10) Emacs: inittab (Text) 24/75 8:13pm

```

La dernière étape est de définir un compte d'utilisateur pour le travail quotidien. Cela se fait avec la commande `adduser` :

```

# Insert Clocks lines here if appropriate
EndSection

# Screen sections
# Any number of screen sections may be present. Each describes
# the configuration of a single screen. 0 single specific screen section
---(Jed 0.99.10) Emacs: xorg.conf (SH) 399/403 8:27pm
root@basse:~# adduser

Login name for new user []: jean

User ID ('UID') [ defaults to next available ]:

Initial group [ users ]:

Additional groups (comma separated) []:

Home directory [ /home/jean ]:

Shell [ /bin/bash ]:

Expiry date (YYYY-MM-DD) []:

Initial group....: users
Additional groups: [ None ]
Home directory...: /home/jean
Shell.....: /bin/bash
Expiry date.....: [ Never ]

This is it... If you want to bail out, hit Control-C. Otherwise, press
ENTER to go ahead and make the account.

Creating new account...

Changing the user information for Jean
Enter the new value, or press ENTER for the default
Full Name []: Jean Dupond
Room Number []:
Work Phone []:
Home Phone []:
Other []:

Changing password for Jean
Enter the new password (minimum of 5, maximum of 127 characters)
Please use a combination of upper and lower case letters and numbers.
New password:
Re-enter new password:

```

Votre système Slackware est à présent installé, et vous pouvez le redémarrer pour commencer à l'utiliser. Vous pouvez à tout moment modifier la configuration du système à l'aide du programme `pkgtool`, ou celle de `X11` à l'aide de `xorgsetup`.

IV - Présentation générale du système

Si vous êtes parvenu jusqu'ici, vous disposez d'un système Linux fonctionnel. Félicitations ! Un monde nouveau s'ouvre à vous, et vous allez pouvoir explorer toutes les fonctionnalités de ce nouveau système.

Toutefois, découvrir un nouveau système d'exploitation peut s'avérer moins facile qu'on ne le pense. En effet, chaque système a ses spécificités, et la manière de réaliser une tâche n'est pas forcément celle à laquelle on peut s'attendre. Ce chapitre a donc pour but de vous présenter les notions générales sur lesquelles le système se base, afin de vous mettre le pied à l'étrier et de vous donner un cadre auquel vous pourrez vous raccrocher.

Si vous avez déjà utilisé un système Unix, vous n'aurez sans doute pas de difficulté à vous adapter, et la plupart des notions qui seront présentées dans ce chapitre ne vous seront donc pas étrangères. De même, si Linux est le premier système d'exploitation que vous utilisez, vous pourrez apprendre les procédures sans trop de difficultés. En revanche, si vous êtes d'abord un utilisateur de Microsoft Windows, la transition sera sans doute difficile dans un premier temps. Les habitudes ayant la vie dure, vous devrez en effet commencer par vous débarrasser d'une bonne partie d'elles avant de pouvoir utiliser correctement Linux.

Nous allons présenter l'architecture générale du système dans un premier temps. La notion d'utilisateur et les principes de base de sécurité des systèmes Unix seront ensuite abordés. Enfin, les notions relatives à la gestion des fichiers sous Linux, ainsi que les principaux systèmes de fichiers disponibles et leurs fonctionnalités seront vues. Ce chapitre se terminera par une présentation de l'arborescence standard du système de fichiers de Linux, qui vous sera sans doute très utile pour savoir le rôle de chacun des répertoires systèmes de Linux et l'emplacement des différents composants du système.

IV-A - Architecture du système

IV-A-1 - Principe de modularité

Comme tout logiciel d'une certaine taille, un système d'exploitation est d'une grande complexité. Tous les systèmes d'exploitation récents sont en effets constitués d'un grand ensemble de composants qui interagissent entre eux et qui doivent fonctionner de concert afin d'assurer la bonne marche du système et répondre aux besoins de l'utilisateur.

Cette complexité implique un grand nombre d'erreurs, d'anomalies et de dysfonctionnements possibles, ainsi qu'une grande difficulté à en comprendre l'origine. En effet, pour qu'un système informatique fonctionne correctement, il faut tout prévoir pour donner une action appropriée à tous les événements possibles, et trouver la cause d'un problème est d'autant plus difficile qu'il y a de composants qui interviennent dans la fonction défaillante. Cela n'est pas humainement réalisable quand le système devient trop complexe si l'on n'est pas organisé et qu'on ne connaît pas les grandes lignes de l'architecture du système.

Pour résoudre ce problème, il est courant de subdiviser les systèmes complexes en composants indépendants, dont le mauvais fonctionnement potentiel ne peut perturber que partiellement les autres parties du système. Par ailleurs, ces composants sont regroupés en couches logicielles successives, les couches supérieures s'appuyant sur les services des couches inférieures. Ainsi, des points de synchronisation à partir desquels le système peut reprendre un fonctionnement normal après une défaillance peuvent être définis. Ces points de synchronisation permettent donc d'assurer la viabilité du système, ou du moins des couches inférieures, même en cas d'erreur inopinée d'un composant dans une couche de plus haut niveau.

Il va de soi que, lorsqu'un composant se plante, ceux qui l'utilisent risquent de se retrouver dans un état d'erreur assez difficile à gérer. Cela peut souvent provoquer leur propre perte. Par conséquent, plus un composant est utilisé, plus il doit être fiable. Autrement dit, les composants des couches inférieures doivent nécessairement être beaucoup plus fiables que ceux des couches de plus haut niveau, car ils sont utilisés par tout le reste du système.

Pour quelques systèmes, le découpage fonctionnel est trop grossier, voire inexistant. Il arrive également que les interactions entre les composants soient trop importantes et impliquent une dépendance entre eux qui annihile le bénéfice de leur isolation. Dans ces systèmes, une défaillance d'un composant peut donc entraîner un dysfonctionnement global sévère, et en pratique peut nécessiter jusqu'à un redémarrage de l'ordinateur. Les systèmes monolithiques et les systèmes dont les composants sont trop interdépendants sont donc sujets à de nombreux redémarrages, parfois même pour des raisons mineures (modification de la configuration du système, ajout d'un logiciel, ou même simple erreur d'un logiciel).

Il est d'usage de considérer que Linux est un système monolithique. En effet, toute la gestion du matériel, ainsi que la gestion de certaines fonctionnalités de haut niveau, est prise en charge au niveau du noyau (« kernel » en anglais). On pourrait donc craindre une faible fiabilité de ce système d'exploitation. Toutefois, cette vision des choses n'est pas tout à fait juste, car, comme on l'a déjà dit, le noyau n'est qu'une partie du système d'exploitation, et en réalité un système Linux est effectivement constitué de différentes couches fonctionnelles. De plus, le noyau reste d'une taille raisonnable. Pour l'essentiel, il est constitué que d'une base relativement petite et très bien testée, à laquelle beaucoup de pilotes de périphériques ont été ajoutés. La complexité du noyau reste donc limitée, car les pilotes de périphériques ne sont bien entendu pas tous utilisés en même temps sur une même machine.

Par ailleurs, le noyau Linux est d'une très, très grande fiabilité, et il n'est pas rare de voir un système Linux fonctionner plusieurs mois ou années sur des serveurs. Cette grande fiabilité provient du modèle de développement de Linux, qui est ouvert à tout le monde (chacun peut récupérer, lire, modifier, et compléter ou corriger le noyau à condition de savoir bien programmer). Ainsi, à partir d'une taille critique en terme de nombre d'utilisateurs, taille que Linux a atteinte, il existe suffisamment de développeurs pour détecter et corriger les erreurs. En pratique, dès qu'une erreur est détectée, elle est souvent corrigée dans les jours qui suivent si la fonctionnalité concernée est effectivement utilisée, ce qui assure finalement une très grande qualité.

IV-A-2 - Les différentes couches logicielles sous Linux

Ainsi, un système Linux est constitué, outre du noyau, d'un certain nombre d'autres couches logicielles, qui s'appuient les unes sur les autres. Nous allons donc présenter brièvement ces couches ici.

Le noyau se trouve bien évidemment à la base de toutes les autres couches, puisqu'il gère quasiment tout le matériel (mémoire, disques, systèmes de fichiers, réseau, clavier, etc.). Mais comme il n'est pas exploitable tel quel (par exemple, il n'est pas capable d'offrir une interface utilisateur permettant de lui donner les commandes que l'ordinateur doit exécuter) il faut au moins une interface utilisateur.

Cette interface utilisateur, c'est ce qu'on appelle le « shell » (ce qui signifie grosso modo « environnement utilisateur »). Le shell est capable de lire des commandes saisies au clavier, de les exécuter et d'afficher leurs résultats à l'écran. En général, les programmes capables de réaliser ces opérations sont appelés des interpréteurs de commandes. Mais le shell est bien plus que cela, car il peut être programmé, et il peut gérer les processus (en arrêter un, en lancer un autre, etc.). Bien entendu, il existe plusieurs shells, mais le plus utilisé sous Linux est le shell bash, qui est développé par la Free Software Foundation et distribué sous licence GPL.

En fait, les commandes que le shell peut exécuter sont en nombre très réduit. La plupart des commandes sont donc tout simplement prises en charge par d'autres programmes. Ces programmes, que l'on peut utiliser directement dans le shell, sont ce que l'on appelle des programmes « en ligne de commande », parce qu'ils sont propres à être utilisés dans les lignes de commandes que l'on saisit au clavier dans le shell. Ces programmes sont, encore une fois, développés soit par la Free Software Foundation, soit par des bénévoles, toujours sous la licence GPL. Toutes ces commandes sont des commandes compatibles Unix, et nous apprendrons à nous en servir dans le [Chapitre 5](#).

Bien entendu, ces commandes sont absolument essentielles pour pouvoir utiliser le système, mais elles sont assez rébarbatives et peu d'utilisateurs acceptent de s'en contenter. C'est pour cela qu'une couche graphique a été développée, pour introduire une interface graphique plus conviviale : XWindow. Encore une fois, cette couche logicielle est constituée de plusieurs composants, dont la base est le serveur X. Le serveur X est un programme capable de fournir les services graphiques (d'où le nom de serveur) aux autres applications. Plusieurs implémentations concurrentes existent. L'une d'elles est particulièrement utilisée sous Linux, puisqu'elle est libre : X.org. À vrai dire, un serveur X ne fait pas grand-chose d'autre que de réaliser des affichages sous les ordres d'autres programmes. D'autres composants permettent donc d'obtenir des fonctionnalités de plus haut niveau.

Le gestionnaire de fenêtres (« Window Manager » en anglais) est un de ces composants. Il se place juste au-dessus du serveur X et prend en charge, comme son nom l'indique, la gestion des fenêtres des applications graphiques sous X. C'est le gestionnaire de fenêtres qui prend en charge la gestion des décorations des fenêtres de premier niveau (c'est-à-dire des fenêtres principales des programmes). Par exemple, il s'occupe d'afficher les bords, la barre de titre, les boutons de réduction et de restauration, etc. des fenêtres. C'est également lui qui s'occupe du positionnement des fenêtres, et qui donc permet à l'utilisateur de déplacer et de réduire les fenêtres des applications

graphiques. L'utilisateur est libre d'utiliser le gestionnaire de fenêtres qu'il désire, selon ses propres goûts et ses désirs, la différence est souvent une pure question de style.

Il existe également des environnements graphiques complets qui, en plus de fournir un gestionnaire de fenêtres, fournissent la plupart des outils classiques que l'on est en droit d'attendre d'un système graphique moderne. Ainsi, ces environnements comprennent des éditeurs, des outils de configuration, des navigateurs Internet, des logiciels multimédias... En plus de ces applications, ils fournissent un cadre standard pour les applications graphiques qui savent communiquer avec eux. Ce cadre permet d'améliorer l'intégration des diverses applications entre elles, et c'est la raison pour laquelle on appelle souvent ces environnements des gestionnaires de bureau. KDE et Gnome sont des exemples de tels environnements de travail.

Enfin, au-dessus de toutes ces couches logicielles, on trouve les applications X, qui sont aussi diverses que variées (traitement de texte, tableurs, logiciels de dessin...). Quelques-unes de ces applications sont simplement des « front-end » d'applications en ligne de commande, c'est-à-dire des interfaces graphiques pour des programmes non graphiques. Ce concept permet d'avoir un composant unique réalisant une action, utilisable en ligne de commande et donc scriptable, et une ou plusieurs interfaces graphiques pour ce composant. De plus, ce modèle permet de séparer clairement l'interface graphique du traitement qu'elle permet de réaliser. La stabilité en est d'autant plus accrue.

Bon nombre d'applications pour XWindow sont libres, ou utilisables librement à des fins non commerciales (dans ce cas, on a le droit de les utiliser tant que ce n'est pas pour réaliser un travail qu'on revendra). On peut donc considérer qu'il est actuellement possible, avec Linux, d'avoir un environnement logiciel complet, fiable et performant... pour un prix de revient minime.

Note : Il n'est pas évident d'établir un parallèle entre l'architecture d'un système comme Linux avec celle de MS Windows. Cependant, on peut considérer que le noyau Linux correspond aux modules KERNEL de Windows et de tous les services de type pilote de périphérique, que le shell correspond à l'interpréteur de commandes CMD.EXE, que les programmes en ligne de commande correspondent aux programmes en mode console classique (xcopy, fdisk, format...), que le serveur X correspond au couple (pilote de carte graphique, GDI), que le gestionnaire de fenêtres correspond au module USER, et que le gestionnaire de bureau correspond à l'explorateur, Internet Explorer, aux fonctionnalités OLE permettant la communication entre applications, et aux programmes fournis avec Windows lui-même.

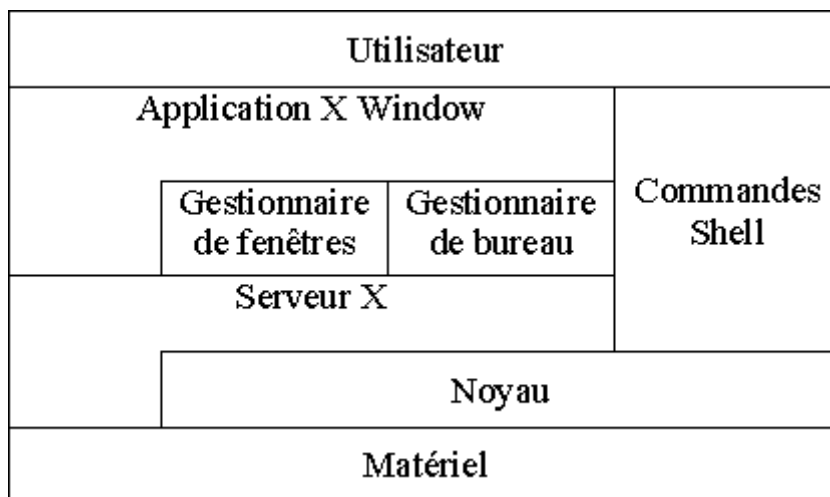
La différence essentielle vient du fait que le shell est à peine programmable sous Windows, que le pilote de carte graphique, la GDI et le module USER sont tous intégrés dans le système au lieu d'en être séparés (ce qui multiplie les chances de crash du système complet), et que la plupart des applications Windows ne peuvent fonctionner que dans l'environnement graphique. Elles sont donc entraînées par le système lorsque les modules graphiques de Windows plantent (je n'ai d'ailleurs jamais vu un processus DOS survivre à un crash de l'interface graphique de Windows).

IV-A-3 - Résumé de l'architecture de Linux

En résumé, un système GNU/Linux est structuré de la manière suivante :

- le noyau Linux ;
- les programmes en ligne de commande et le shell ;
- le serveur XWindow ;
- le gestionnaire de fenêtres et le gestionnaire de bureau ;
- les applications XWindow.

La figure suivante vous présente comment ces différentes couches logicielles s'agencent les unes par rapport aux autres.



Architecture des systèmes GNU/Linux

Cette architecture est, comme on peut le voir, très avantageuse :

- les systèmes Unix, donc Linux, sont très structurés, donc plus simples à utiliser, à configurer, à maintenir et à développer ;
- ils sont très stables, car les composants de haut niveau n'interfèrent pas sur les composants de bas niveau ;
- ils sont faciles à personnaliser, puisque l'utilisateur a le choix des outils à chaque niveau ;
- Linux a de plus la particularité d'être parfaitement modifiable, puisque si l'on sait programmer, on peut personnaliser les composants du système ou en rajouter à tous les niveaux ;
- et il n'est pas propriétaire, c'est-à-dire que l'on n'est pas dépendant d'un éditeur de logiciel pour résoudre un problème donné.

En bref, c'est la voie de la vérité.

IV-B - Sécurité et utilisateurs

À l'heure où la sécurité devient un enjeu majeur en raison de l'importance des menaces virales et autres attaques contre les systèmes informatiques et leurs utilisateurs, les systèmes Unix tels que Linux, bien que de conception très ancienne, apparaissent comme avant-gardistes et parfaitement en phase avec le marché.

La sécurité est à présent considérée comme une fonctionnalité essentielle du système d'exploitation, et devient de plus en plus incontournable. De ce fait, il est nécessaire de présenter les concepts de sécurité utilisés par les systèmes Unix et Linux, afin que vous puissiez appréhender les risques et maîtriser toutes les principales fonctions de sécurité du système.

Nous verrons donc ces concepts de base dans un premier temps. Puis, nous nous intéresserons à la notion d'utilisateur, qui est à la base de la sécurité sous Unix. Enfin, nous verrons comment cette sécurité est assurée dans le système, en particulier au niveau du système de fichiers.

IV-B-1 - Généralités

Les systèmes Unix, depuis leurs origines, ont toujours pris en compte les problèmes de sécurité. En particulier, les développeurs d'application Unix ont toujours fait attention à ce que leurs logiciels se comportent relativement bien en termes de sécurité. En pratique, bien que la sécurité des programmes soit loin d'être parfaite, surtout pour les plus anciens, des principes de base ont toujours été respectés, ce qui fait que les applications Unix ne font pas, en règle générale, n'importe quoi avec le système.

Ainsi, il est beaucoup plus facile de faire fonctionner une application Unix dans un contexte de sécurité restreint qu'une application Windows, car les développeurs de ces dernières ont longtemps été habitués à une très grande

liberté et ont donc réalisé des applications qui ne peuvent pas être utilisées si elles n'ont pas les pleins pouvoirs. Les applications Unix qui se comportent mal sont donc très rares, ce qui fait que les systèmes Unix peuvent faire peu de cas d'elles et n'ont pas à maintenir une compatibilité ascendante avec elles. Les systèmes Unix ont donc un modèle de sécurité beaucoup plus simple et strict, et n'ont pas à s'embarrasser avec des techniques complexes et des circonvolutions insensées telles que la virtualisation de l'ensemble du système pour les applications afin d'assurer un semblant de sécurité tout en permettant la compatibilité ascendante.

Par ailleurs, la sécurité permet de protéger non seulement le système, mais aussi l'utilisateur de ses propres erreurs. En effet, le système est capable de se protéger contre les fausses manipulations de l'utilisateur. Cela peut être extrêmement utile avec les débutants, puisqu'ils peuvent expérimenter à loisir tout en étant sûrs qu'ils ne feront pas de bêtise irrémédiable. Ainsi, ils seront plus sereins et n'auront pas « peur de faire une bêtise ». Mais cela est très utile même pour les utilisateurs chevronnés, car tout le monde peut effectivement faire une erreur !

La sécurité permet également de protéger les données de chaque utilisateur vis-à-vis des autres. En effet, par défaut, les données d'un utilisateur ne peuvent pas être modifiées par un autre utilisateur. Vous n'aurez donc aucune crainte que vos chères têtes blondes puissent effacer ou détruire par mégarde votre correspondance professionnelle ou l'ensemble des photos de famille...

Enfin, ayant été prise en compte dès le départ, la sécurité sait se rendre très discrète sur les systèmes Unix, et bien que parfaitement fonctionnelle, elle n'ennuie pas l'utilisateur et reste facilement configurable. Ces deux aspects sont fondamentaux, car du coup, l'utilisateur n'a pas la tentation - ni le besoin - de tout désactiver pour faire fonctionner ses applications !

En résumé, la sécurité sous Linux, c'est :

- une fonctionnalité à part entière, qui est parfaitement intégrée dans le système ;
- un moyen de se protéger contre ses propres erreurs ainsi que celles des autres utilisateurs ;
- une facilité d'utilisation et une transparence qui la rendent utilisable en pratique sans perturber le travail de l'utilisateur.

IV-B-2 - Notion d'utilisateur et d'administrateur

Linux dispose de plusieurs modèles de sécurité, mais, historiquement, la sécurité se base sur la notion d'utilisateur. Nous allons donc nous intéresser ici à la notion d'utilisateur, et voir comment cette notion peut être utilisée pour mettre en pratique les contrôles de sécurité du système.

Comme vous le savez sans doute déjà, Linux est un système multi-utilisateur. Cela signifie que plusieurs personnes peuvent utiliser l'ordinateur simultanément (et pas uniquement les unes à la suite des autres), et que le système se charge de faire respecter la loi entre elles. Le système protège les données de chaque utilisateur des actions que peuvent effectuer les autres utilisateurs. Il veille également à partager équitablement les ressources de la machine entre les utilisateurs, tant au niveau de la puissance de calcul qu'au niveau de la mémoire, du disque, des imprimantes, etc.

Note : Si vous vous demandez comment plusieurs personnes peuvent se partager le clavier et l'écran et ainsi utiliser l'ordinateur en même temps, la réponse est simple : ils ne le peuvent pas. Par contre, il est possible que plusieurs personnes soient connectées à l'ordinateur via le réseau, utilisant d'autres claviers et d'autres écrans (on appelle un couple clavier/écran un « terminal »). Il est de plus théoriquement possible de connecter plusieurs claviers, souris et écrans à une même unité centrale, même si ces configurations sont encore très rares et non standards. Les utilisateurs peuvent également lancer des programmes en arrière-plan, et laisser ces programmes tourner même lorsqu'ils ne sont plus connectés à l'ordinateur. Évidemment, cela nécessite que les programmes ainsi lancés ne soient pas interactifs. Autrement dit, ils doivent être capables de fonctionner sans intervention de celui qui les a lancés.

Ne croyez pas que ce dernier cas de situation est extrêmement rare. En effet, le système utilise des tâches qui tournent en arrière-plan dans des comptes dédiés et différents du vôtre, afin de prendre en charge certaines fonctionnalités ou de réaliser des opérations de maintenance.

En interne, le système affecte un numéro unique à chaque utilisateur afin de pouvoir l'identifier. Ce numéro d'utilisateur est défini lors de la création de son compte dans le système, et ne peut être modifié par la suite. Ainsi, toute action effectuée dans le système peut être rattachée à un utilisateur particulier. L'utilisation du système requiert donc que chaque utilisateur s'identifie au niveau du système avant toute autre opération. Généralement, cela se fait en indiquant son nom d'utilisateur, que l'on appelle classiquement le « login ». Bien entendu, afin d'éviter qu'un utilisateur puisse usurper l'identité d'un autre utilisateur, le système exige que l'utilisateur fournisse un mot de passe après s'être identifié pour s'authentifier.

La sécurité du système se base alors, lorsqu'une opération est exécutée, sur l'identité de l'utilisateur qui réalise cette commande et sur les droits dont cet utilisateur dispose sur les objets que cette commande doit manipuler. Par exemple, la suppression d'un fichier système ne peut pas être réalisée par un utilisateur lambda, parce que cet utilisateur n'a pas les droits de suppression (ni de modification d'ailleurs, et parfois même de lecture) sur ce fichier.

L'administration du système lui-même revient à la charge d'un utilisateur particulier. Cet utilisateur, classiquement nommé « root », a virtuellement tous les droits sur le système. C'est ce qu'on appelle l'administrateur du système, ou encore le super utilisateur. En interne, le super utilisateur est identifié par le numéro 0, qui lui est réservé. Ainsi, seul l'utilisateur dont le numéro est 0 peut modifier les fichiers de configuration du système.

Note : Bien entendu, travailler sous l'identité du super utilisateur est extrêmement dangereux, car dès lors tous les mécanismes de sécurité du système sont désactivés. Une erreur de manipulation avec cette identité peut donc être fatale et détruire complètement le système ! C'est pour cette raison que la plupart des distributions imposent de définir un compte utilisateur standard en plus du compte root, et que vous ne devriez travailler que sous ce compte. Ce n'est pas une restriction, c'est tout simplement pour votre bien. Ne vous inquiétez pas, comme il l'a déjà été dit plus haut, vous n'avez quasiment jamais besoin des droits administrateurs pour travailler sous Linux, parce que les applications sont en général conçues dès le départ pour être utilisables dans des comptes restreints. Alors n'ayez aucune crainte, et respectez les bonnes manières : ne travaillez jamais sous le compte root.

Les nouveaux utilisateurs de Linux qui ont fait leurs premiers pas en informatique sous Microsoft Windows ont souvent tendance à considérer cette recommandation comme un « faites ce que je dis, pas ce que je fais » et croient donc que les utilisateurs chevronnés ne la respectent pas. Qu'ils se détrompent, la plupart des utilisateurs de Linux respectent cette règle en permanence ! Ce n'est pas parce que ce n'est pas votre habitude que cette recommandation est un « on-dit ». D'ailleurs, elle est tellement importante que certaines distributions désactivent tout bonnement le compte administrateur. Il est donc tout simplement impossible de se connecter sous ce compte, même si l'on est la personne qui a installé le système ! Bien entendu, dans ce cas, les opérations d'administration sont toutes identifiées et nécessitent de fournir le mot de passe administrateur pour être exécutées.

En résumé, les mécanismes de sécurité basés sur la notion d'utilisateur fonctionnent selon les principes suivants :

- l'utilisateur doit s'identifier (généralement, en fournissant son identifiant d'utilisateur, ou « login ») et s'authentifier (classiquement en fournissant son mot de passe, ou « password ») lorsqu'il se connecte à l'ordinateur ;
- le noyau conserve pour chaque processus que l'utilisateur lance l'identité de cet utilisateur ;
- le noyau contrôle les droits de l'utilisateur à chaque accès d'une ressource de chaque programme de cet utilisateur (en particulier, comme nous allons le voir plus loin, au niveau des fichiers) ;
- le noyau assure une indépendance totale des processus, tant au niveau des zones de mémoire utilisées qu'au niveau de l'accès aux ressources de la machine. Un processus ne peut donc, par défaut, ni lire ni écrire dans la mémoire d'un autre processus, et les ressources de calcul ou d'entrée/sortie sont partagées entre les différents processus par les mécanismes du multitâche.

IV-B-3 - La sécurité au niveau du système de fichiers

Nous avons vu que le noyau contrôlait les droits de l'utilisateur vis-à-vis des ressources qu'il manipulait pour chaque processus qu'il exécute. En particulier, l'accès à chaque fichier est contrôlé lors de l'ouverture de ce fichier par les processus de l'utilisateur. Ainsi, les données de chaque utilisateur sont protégées par le système, ainsi que les fichiers du système lui-même. De plus, comme l'accès aux périphériques se fait généralement par l'intermédiaire de fichiers

spéciaux, les accès aux ressources matérielles sont également contrôlés via les mécanismes de sécurité du système de fichiers. Il est donc important de comprendre comment les systèmes de fichiers gèrent la sécurité.

IV-B-3-a - Les droits sur les fichiers

Les systèmes de fichiers définissent, pour chaque fichier, les droits auxquels un utilisateur peut prétendre. Ces droits comprennent la possibilité de lire ou écrire un fichier, d'accéder ou non à une ressource ou d'exécuter un programme. En pratique, chaque utilisateur peut définir les droits qu'il veut se donner sur ses propres fichiers, et surtout les droits qu'il veut donner aux autres utilisateurs du système sur ses fichiers. Par exemple, il peut protéger un fichier auquel il tient particulièrement en lecture seule, afin d'éviter de le détruire par inadvertance.

L'administrateur peut également créer un ou plusieurs « groupes » d'utilisateurs, afin que les utilisateurs puissent donner des droits spécifiques aux utilisateurs de ces groupes sur leurs fichiers. En pratique, cela se fait en fixant le groupe propriétaire du fichier, et en fixant les droits spécifiques à ce groupe sur le fichier.

Il est donc possible de définir les droits sur un fichier à trois niveaux différents :

- au niveau du propriétaire du fichier (par défaut, l'utilisateur qui l'a créé) ;
- au niveau du groupe propriétaire du fichier (par défaut, les fichiers créés par les utilisateurs appartiennent au groupe « users ») ;
- au niveau de tous les autres utilisateurs (c'est-à-dire les utilisateurs qui ne sont ni le propriétaire du fichier ni les utilisateurs du groupe du fichier).

En pratique, les droits des fichiers sont représentés par une série de lettres indiquant les droits de chacun des niveaux que l'on vient de voir. Le droit de lecture est représenté par la lettre 'r' (pour « Read only »), le droit d'écriture par la lettre 'w' (pour « Writeable »), et le droit d'exécution par la lettre 'x' (pour « eXecutable »). Le droit de lecture correspond à la possibilité d'ouvrir et de consulter un fichier, ou de lister le contenu d'un répertoire. Le droit d'écriture correspond à la possibilité de modifier un fichier, ou de créer ou supprimer un fichier d'un répertoire. Enfin, le droit d'exécution correspond à la possibilité d'exécuter un fichier contenant un programme, ou d'entrer dans un répertoire.

Par exemple, la liste des droits sur un fichier lisezmoi.txt appartenant à un hypothétique utilisateur Jean Dupont (dont le login serait jean) se présente comme suit dans la sortie de la commande `ls -l lisezmoi.txt` (qui, comme on le verra plus loin, permet d'afficher les informations de ce fichier) :

```
-rw-r--r-- 1 jean users 444 2006-12-10 08:42 lisezmoi.txt
```

Ce fichier appartient bien à l'utilisateur `jean` et au groupe `users` et a une taille de 444 octets. Les droits sont affichés au début de la ligne. Le premier groupe de droit (`rw-`) indique que l'utilisateur peut lire et écrire le fichier, mais pas l'exécuter. En effet, le tiret (`-`) indique que le droit d'exécution n'est pas donné. Les deuxième et troisième groupes de droits (`r--` tous les deux) indiquent respectivement que les utilisateurs du groupe `users` et tous les autres utilisateurs n'ont que le droit de lecture. Le premier tiret au début de la ligne signifie que le fichier n'est pas un répertoire, dans le cas contraire il serait remplacé par la lettre 'd' (pour « Directory »).

Note : On notera que le fait d'avoir le droit d'écriture sur un fichier ne donne pas le droit de le supprimer (cependant, on peut le vider !). Pour cela, il faut avoir le droit d'écriture sur le répertoire contenant ce fichier. De même, il est possible d'obtenir la liste des fichiers d'un répertoire sans pouvoir s'y déplacer, ou encore de modifier un fichier sans pouvoir le lire. Comme on le voit, les droits d'accès aux fichiers et aux répertoires sont très souples.

Par défaut, lors de la création d'un fichier par un utilisateur, les droits de lecture et d'écriture sont donnés à cet utilisateur, et seuls les droits de lecture sont donnés au groupe du fichier et aux autres utilisateurs. Ainsi, tout le monde peut lire le fichier, mais seul son propriétaire peut le modifier. Cela correspond à l'usage classique sur un système multiutilisateur où les utilisateurs se connaissent et n'ont pas de fichiers confidentiels. Dans le cas contraire, l'utilisateur devra retirer les droits de lecture pour les autres utilisateurs (nous verrons comment faire cela plus loin).

Note : Comme vous pouvez le constater, les droits d'exécution ne sont pas donnés par défaut, même pour les fichiers binaires. Ainsi, le système ne se base pas sur le contenu du fichier, et, comme on l'a vu, encore moins sur son extension, pour déterminer si un fichier est exécutable ou non. Pour qu'un fichier soit exécutable, il faut lui donner explicitement les droits d'exécution !

Cela n'a l'air de rien, mais cette contrainte est un facteur de sécurité extrêmement important de nos jours. En effet, la plupart des virus de mail ne sont en fait rien d'autre que des chevaux de Troie transférés en pièce jointe, et qui ne peuvent s'exécuter que si l'utilisateur les lance explicitement. Sous Windows, le simple fait de double-cliquer sur une pièce jointe dont l'extension est exécutable lance effectivement le programme (c'est pour cela que les pièces jointes de ce type sont souvent filtrées maintenant). Sous Unix et Linux, cela ne fait strictement rien, même si c'est un programme, car pour l'exécuter, il faut d'abord l'enregistrer, modifier ses droits d'exécution explicitement, et l'exécuter. Ainsi, un utilisateur non averti n'a quasiment aucune chance de lancer un tel cheval de Troie par mégarde...

IV-B-3-b - Notion d'ACL

La notion de groupe d'utilisateurs que l'on vient de voir est généralement utilisée pour définir des classes d'utilisateurs, et attribuer un peu plus de privilèges que les utilisateurs normaux n'en disposent aux utilisateurs de ces classes, selon les nécessités.

Par exemple, l'administrateur peut n'autoriser l'accès au lecteur de CD-ROM qu'à quelques utilisateurs, en modifiant les droits sur le fichier spécial de périphérique permettant d'accéder au lecteur de CD-ROM. Premièrement, il crée un groupe d'utilisateurs « CD-ROM » et affecte le fichier spécial de périphérique du CD-ROM à ce groupe. Ensuite, il restreint les droits d'accès à ce fichier pour que seuls les membres de ce groupe puissent y accéder. Enfin, il place les utilisateurs autorisés à accéder au CD-ROM dans le groupe en question.

Toutefois, cette technique est relativement contraignante pour les utilisateurs, parce que la création d'un nouveau groupe d'utilisateurs est une opération privilégiée que seul l'administrateur du système peut réaliser d'une part, et parce qu'un utilisateur ne peut changer le groupe propriétaire d'un fichier que vers un groupe dont il fait lui-même partie, d'autre part. Enfin, un même fichier ne peut appartenir qu'à un seul groupe, et créer un groupe pour rassembler tous les utilisateurs de deux groupes n'est pas une solution très élégante. C'est pour cette raison que les systèmes de fichiers modernes intègrent la notion d'ACL (abréviation de l'anglais « Access Control List », c'est-à-dire les listes de contrôle d'accès).

Les ACL sont des listes d'informations décrivant les droits d'accès aux fichiers. Avec les ACLs, les droits ne sont pas définis pour chaque catégorie d'utilisateurs (propriétaire, groupe et autres), mais de manière plus fine, fichier par fichier, pour des utilisateurs et des groupes variés. De plus, les ACLs peuvent être manipulées par le propriétaire du fichier, donc sans avoir recours aux droits administrateur, ce qui résout le problème des groupes.

Note : La contrepartie de cette facilité est qu'il n'est pas toujours évident de contrôler la sécurité de manière globale avec les ACLs, puisqu'on ne peut plus retirer les droits d'un utilisateur sur tout un jeu de fichiers appartenant à un même groupe simplement en le supprimant de la liste des utilisateurs de ce groupe. Il n'est donc pas recommandé d'utiliser les ACLs pour définir les droits sur les fichiers systèmes, et de ne conserver cette fonctionnalité que pour que les utilisateurs puissent définir des droits plus fins sur leurs propres fichiers.

IV-B-3-c - Les attributs spéciaux des fichiers

Nous avons vu dans les sections précédentes que la notion d'utilisateur, couplée avec les droits sur les fichiers, permettait au noyau de fournir une sécurité sur les fichiers des utilisateurs et du système. En particulier, les processus lancés par l'utilisateur s'exécutent en son nom, et sont soumis aux mêmes restrictions que l'utilisateur qui les a lancés.

Il existe toutefois quelques programmes qui doivent faire exceptions à cette règle. Il s'agit généralement de programmes qui ont besoin des droits administrateurs pour réaliser une opération de reconfiguration sur le compte de l'utilisateur. C'est en particulier le cas de quelques commandes systèmes (comme `passwd`, qui permet de changer de mot de passe), qui peuvent être lancées par les utilisateurs, mais qui doivent s'exécuter au nom du système (dans le compte `root`).

Ces programmes sont bien entendu réalisés avec un grand soin, de telle sorte que l'on peut être sûr que l'utilisateur ne peut pas les utiliser dans le but de réaliser une opération à laquelle il n'a pas droit dans le compte administrateur. De ce fait, il est impossible à un utilisateur de violer les règles de sécurité du système.

Pour parvenir à ce comportement, les systèmes de fichiers utilisent des attributs spéciaux sur les fichiers exécutables de ces programmes. Ces attributs viennent en complément des droits d'accès que l'on a présentés dans les sections précédentes.

Le premier de ces attributs est le bit « setuid » (qui est l'abréviation de l'anglais « SET User IDentifier »). Il ne peut être placé qu'au niveau des droits du propriétaire sur le fichier. Il permet d'indiquer que le fichier est exécutable, et que lorsque le programme qu'il contient est lancé par un utilisateur, le processus correspondant s'exécute avec les droits du propriétaire du fichier et non pas avec ceux de l'utilisateur qui l'a lancé. Cependant, le système conserve tout de même le numéro de l'utilisateur réel qui a lancé le processus en interne, ce qui fait que le programme peut savoir par qui il a été lancé et au nom de qui il s'exécute effectivement.

Finalement, un processus dispose donc toujours de deux numéros d'utilisateur :

- le numéro de l'utilisateur réel (« real user id » en anglais), qui est le numéro de l'utilisateur qui a lancé le programme ;
- le numéro de l'utilisateur effectif (« effective user id » en anglais), qui est le numéro de l'utilisateur avec les droits duquel le processus fonctionne.

Le bit setuid permet donc simplement d'affecter le numéro du propriétaire du fichier au numéro d'utilisateur effectif du processus lorsqu'il est lancé. Le fait de conserver le numéro de l'utilisateur réel permet au programme de réaliser des vérifications de sécurité additionnelles. Par exemple, la commande `passwd`, qui permet de changer le mot de passe d'un utilisateur, a besoin des droits de l'utilisateur root pour enregistrer le nouveau mot de passe. Il dispose donc du bit setuid pour que tous les utilisateurs puissent l'utiliser. Cependant, même s'il s'exécute au nom de l'utilisateur root, il ne doit pas permettre à n'importe qui de changer le mot de passe des autres utilisateurs : seul l'utilisateur root a le droit de faire cette opération. Il utilise donc le numéro de l'utilisateur réel qui a lancé la commande pour savoir si c'est bien l'utilisateur root qui l'a lancé.

Le bit setuid est l'attribut le plus couramment utilisé, essentiellement pour certaines commandes systèmes. Il est représenté par la lettre 's' (comme « Setuid »), et il remplace le droit d'exécution ('x') des fichiers pour le propriétaire des fichiers (rappelons que le bit setuid implique que le fichier est exécutable). Il n'a aucune signification pour les répertoires.

Le deuxième attribut spécial est le bit « setgid » (qui est l'abréviation de l'anglais « SET Group IDentifier »). Ce bit fonctionne un peu de la même manière que le bit setuid, à ceci près qu'il fixe le numéro de groupe effectif du processus lancé à celui de son fichier exécutable. Cet attribut est également représenté par la lettre 's', et remplace le droit d'exécution ('x') pour les utilisateurs du groupe auquel appartient le fichier exécutable. De plus, et contrairement au bit setuid, ce bit a une signification pour les répertoires. Un répertoire disposant du bit setgid permet de faire en sorte que tous les fichiers qui sont créés dans ce répertoire se voient automatiquement attribués le même groupe que le répertoire. Ce bit est relativement peu utilisé.

Enfin, il existe un troisième attribut, le bit « sticky ». Cet attribut remplace l'attribut exécutable pour les autres utilisateurs que le propriétaire du fichier ou du répertoire et les membres du groupe auquel il appartient. Contrairement aux bits setuid et setgid, il est représenté par la lettre 't' (pour « sTicky »). Sa signification est assez spéciale : elle permet de faire en sorte que les programmes restent chargés en mémoire après leur terminaison, ce qui permet de les relancer plus rapidement. Afin de ne pas consommer la mémoire de manière permanente, le code du programme est placé automatiquement dans le swap s'il n'est toujours pas relancé après un certain temps, mais même dans ce cas, tous les calculs de chargement sont déjà effectués. Le lancement des programmes marqués de ce bit sera donc toujours accéléré. Sachez cependant ne pas abuser du bit sticky car la mémoire (même virtuelle) est encore une ressource rare. Pour les répertoires, sa signification est totalement différente : elle permet de restreindre les droits des utilisateurs sur les répertoires ayant ce bit positionné. Ce bit fait en sorte que même si un utilisateur dispose des droits d'écriture sur le répertoire, il ne peut pas supprimer tous les fichiers de ce répertoire. Les seuls fichiers

qu'il est autorisé à supprimer sont ses propres fichiers. Bien entendu, il est toujours possible d'ajouter des fichiers dans le répertoire en question.

Les questions qui se posent évidemment sont les suivantes. Est-ce qu'un particulier a besoin de tout cela ? Ces fonctionnalités ne sont-elles pas réservées aux serveurs ? Est-ce qu'on ne risque pas de perdre beaucoup de temps pour définir les droits pour chaque utilisateur et pour chaque ressource du système ? La gestion de la sécurité ne consomme-t-elle pas trop de ressources ? Ces questions sont légitimes. Heureusement, les réponses sont simples. Rappelons pour commencer qu'il est très intéressant, même pour un particulier, de disposer de ces fonctionnalités. En effet, la sécurité permet tout simplement de protéger le système contre les erreurs des utilisateurs, voire l'utilisateur lui-même de ses propres erreurs. Ainsi, avec Linux, on peut faire n'importe quoi, on est certain que le système restera intact. Cette sécurité est telle que, finalement, Linux est justement le système d'exploitation idéal pour apprendre l'informatique à quelqu'un : savoir que le système protège tout ce qui est important permet aux débutants de prendre des initiatives sans crainte. Quant aux éventuels revers de médaille, ils sont absents : la gestion de la sécurité ne consomme quasiment aucune ressource, et sa configuration est élémentaire. Toutes les distributions s'installent de telle sorte que le système se protège des utilisateurs, et que ceux-ci soient indépendants les uns des autres. En pratique, les utilisateurs n'ont tout simplement pas à se soucier de tous ces attributs de fichiers. Même l'administrateur laisse souvent les attributs par défaut définis par les distributions, car ils correspondent à la majorité des besoins de sécurité.

IV-C - Fonctionnalités des systèmes de fichiers

Les systèmes de fichiers Unix sont des systèmes de fichiers extrêmement évolués, qui fournissent à la fois d'excellentes performances, une grande sécurité, et des fonctionnalités puissantes. Peu d'utilisateurs savent exactement quels sont les services qu'ils peuvent fournir en général, et beaucoup croient que les systèmes de fichiers savent juste stocker des données dans des fichiers organisés dans une hiérarchie de répertoires. Mais nous allons voir qu'ils permettent de faire beaucoup mieux que cela !

IV-C-1 - Le système de fichiers virtuel

Pour commencer, il faut préciser que Linux ne travaille pas directement avec les systèmes de fichiers physiques. En effet, il interpose systématiquement un système de fichiers intermédiaire, nommé « Virtual File System » (« VFS » en abrégé), qui permet aux applications d'accéder à différents systèmes de fichiers de manière indépendante de leur nature et de leur structure interne. Le système de fichiers virtuel ne fait pas grand-chose en soi : il se contente de transférer les requêtes des applications vers les systèmes de fichiers réels. Il fournit donc une interface bien définie pour les applications, que celles-ci doivent utiliser. Les systèmes de fichiers réels, quant à eux, doivent simplement fournir les services dont le système de fichiers virtuel a besoin. Tous les systèmes de fichiers réels ne disposent toutefois pas forcément de toutes les fonctionnalités demandées par le système de fichiers virtuel. Dans ce cas de configuration, la requête de l'application désirant effectuer l'opération manquante échouera tout simplement.

Comme on peut le constater, cette architecture est modulaire. Et comme on l'a vu pour l'architecture du système en général, cela apporte beaucoup de bénéfices. Les plus évidents sont indiqués ci-dessous :

- Linux est capable de gérer plusieurs systèmes de fichiers réels. La seule condition est qu'ils doivent tous fournir les services de base exigés par le système de fichiers virtuel ;
- les applications peuvent utiliser plusieurs de ces systèmes de fichiers réels de manière uniforme, puisqu'elles n'utilisent que le système de fichiers virtuel. Cela simplifie leur programmation, et permet d'éviter autant de bogues potentiels ;
- chaque système de fichiers réel étant indépendant des autres, il ne perturbe pas leur fonctionnement. En particulier, un système de fichiers corrompu ne corrompt pas les autres.

Applications					
Système de fichiers virtuel					
EXT2/3	ReiserFS	NFS	DOS	ISO9660	NTFS
Pilotes E/S					
Périphériques					

Le système de fichiers virtuel

Avec cette architecture, un grand nombre de systèmes de fichiers ont été développés pour Linux. Parmi ces systèmes de fichiers, on retrouve les plus connus, à savoir :

- le système de fichiers EXT2, qui est le système de fichiers natif de Linux ;
- le système de fichiers EXT3, qui est une évolution du système de fichiers EXT2 capable de prendre en charge également les mécanismes de journalisation (voir la note ci-dessous pour savoir ce qu'est la journalisation d'un système de fichiers) ainsi que les blocs défectueux sur le support physique ;
- le système de fichiers ReiserFS, qui supprime la notion de bloc disque et qui est également journalisé ;
- les systèmes de fichiers FAT, FAT32 et FAT32X (utilisés par les systèmes DOS et Windows, ainsi que par les clefs USB et les cartes mémoire) ;
- le système de fichiers NTFS (utilisé par Windows NT, Windows 2000 et XP), en lecture et partiellement en écriture (seul l'écrasement des données existantes dans un fichier, sans redimensionnement, est autorisé à l'heure actuelle) ;
- le système de fichiers ISO9660, qui est utilisé par tous les CD-ROM. Les extensions permettant de gérer les noms longs sont également gérées. Ces extensions comprennent en particulier le système de fichiers Joliet (extensions de Microsoft pour Windows 95) et Rock Ridge (extensions utilisées par tous les systèmes Unix) ;
- le système de fichiers NFS (utilisé pour distribuer sur un réseau un système de fichiers).

Note : La journalisation consiste à écrire sur le disque toutes les opérations en cours à chaque instant. Ainsi, lorsqu'un redémarrage intempestif se produit, le système peut ramener rapidement la structure de données du système de fichiers dans un état cohérent. La journalisation accroît donc encore à la fiabilité du système de fichiers.

Linux gère également d'autres systèmes de fichiers natifs ou utilisés par d'autres systèmes d'exploitation (Unix ou non). Il permet même d'intégrer des pseudo systèmes de fichiers générés par le noyau. Ces systèmes de fichiers sont complètement fictifs : leur structure et leurs fichiers sont générés dynamiquement par le noyau lorsqu'une application y accède. Ils sont principalement utilisés pour fournir aux applications des informations que le noyau met à leur disposition, pour réaliser des systèmes de fichiers en mémoire, et pour gérer l'ensemble des fichiers spéciaux de périphériques installés sur l'ordinateur.

Les systèmes de fichiers natifs de Linux sont de loin les systèmes de fichiers les plus fonctionnels, les plus fiables et les plus courants. Le choix du système de fichiers dépend donc généralement de l'usage que l'on en fera, certains systèmes de fichiers étant plus appropriés pour certains types d'utilisation. Si l'on recherche essentiellement la stabilité, je recommande le système de fichiers EXT3, car c'est pour l'instant le seul système de fichiers capable de prendre en compte les blocs défectueux sur le support physique.

Quoi qu'il en soit, on ne pourra installer Linux que sur un système de fichiers de type Unix. Ces systèmes de fichiers sont tous plus fonctionnels et plus performants que les systèmes de fichiers FAT. Leurs principales fonctionnalités sont les suivantes :

- les accès aux fichiers sont rapides, même plus rapides que les systèmes de fichiers basés sur la FAT sous Windows, qui pourtant ne gèrent pas les droits des utilisateurs ni les autres fonctionnalités avancées des systèmes de fichiers Unix ;

- la fragmentation des fichiers est quasiment inexistante. En fait, la fragmentation des fichiers existe effectivement, mais les algorithmes utilisés par les systèmes de fichiers pour allouer les blocs du disque dur lors de l'écriture dans un fichier sont suffisamment évolués pour éviter qu'elle ne se produise de manière trop importante. De plus, les algorithmes de lecture anticipée des blocs et de mise en mémoire cache des données lues du sous-système d'entrée/sortie masquent complètement les latences dues à la fragmentation. De ce fait, en pratique, la fragmentation n'a pas d'impact sur les performances du système, et peut être tout simplement ignorée. Pour donner un ordre de grandeur, après installation, suppression, manipulation d'un grand nombre d'applications et de petits fichiers sur une partition EXT2 de 800 Mo, le tout réalisé par plusieurs processus fonctionnant en même temps, la fragmentation reste inférieure à 1% sur 57571 fichiers (sur un total de 249856 fichiers que le système de fichiers pourrait contenir) ;
- quant à la fiabilité, elle est gérée grâce à un stockage redondant des principales structures de données internes. Ainsi, si une erreur apparaît dans le système de fichiers, les parties défectueuses peuvent être reconstituées à partir des informations sauvegardées. Cette réparation est réalisée automatiquement à chaque redémarrage de la machine si nécessaire.

IV-C-2 - Liens symboliques et liens physiques

Une fonctionnalité intéressante fournie par les systèmes de fichiers Unix est la possibilité de réaliser des liens sur des fichiers ou des répertoires.

Un lien est une référence à un fichier ou un répertoire existant, qui peut être manipulé exactement comme sa cible. Il existe deux sortes de liens : les liens physiques, qui sont réellement une référence sur les données du fichier au niveau de la structure même du système de fichiers, et les liens symboliques, qui ne sont rien d'autre qu'un fichier additionnel contenant les informations nécessaires pour retrouver la cible.

Les liens physiques présentent les inconvénients de ne pas pouvoir référencer des répertoires, et de ne pouvoir référencer que des objets du même système de fichiers que celui dans lequel ils sont créés. La limitation sur les répertoires permet d'éviter de construire des cycles dans la structure du système de fichiers. Quant à la limitation à la frontière des systèmes de fichiers, elle est obligatoire puisque les liens physiques sont gérés directement au niveau de la structure du système de fichiers. En revanche, ils présentent des avantages certains :

- le déplacement des cibles ne les perturbe pas si celles-ci restent dans le même système de fichiers, parce que dans ce cas les données ne sont pas déplacées sur le disque ;
- la suppression de la cible ne détruit pas le lien physique. Tous les liens physiques sur un fichier partagent la même structure de données du système de fichiers, et celle-ci n'est réellement détruite qu'à la destruction du dernier lien physique.

En fait, toute entrée de répertoire est un lien physique sur le contenu du fichier. Le fait d'avoir plusieurs liens physiques sur les mêmes données correspond à disposer de plusieurs entrées de répertoire donnant accès aux mêmes données dans le système de fichiers. Il serait possible de créer des liens physiques dans un système de fichiers FAT, mais ils seraient interprétés comme des références croisées par les outils de vérification de disque. Le système de fichiers FAT de Linux interdit donc la création des liens physiques, tout comme le font DOS et Windows.

Les liens symboliques, quant à eux, permettent de référencer des fichiers ou des répertoires se trouvant dans d'autres systèmes de fichiers que le leur. C'est pour cette raison qu'ils sont très couramment utilisés (en fait, les liens physiques ne sont quasiment pas utilisés, parce qu'il est très courant de faire un lien sur un répertoire, ce que seuls les liens symboliques savent faire). En revanche, ils sont extrêmement dépendants de leur cible : si elle est supprimée ou déplacée, tous les liens symboliques qui s'y réfèrent deviennent invalides.

La référence sur le fichier ou le répertoire cible contenue dans les liens symboliques peut être soit relative à l'emplacement de leur cible, soit absolue dans le système de fichiers. Chacune de ces méthodes a ses avantages et ses inconvénients : les liens symboliques qui contiennent des références relatives ne sont pas brisés lors d'un déplacement de la cible, pourvu qu'ils soient déplacés également et restent à la même position relative par rapport à celle-ci dans la hiérarchie du système de fichiers. En revanche, ils sont brisés s'ils sont déplacés et que la cible ne l'est pas. Les liens symboliques utilisant des références absolues sont systématiquement brisés lorsque la cible est déplacée, mais ils restent valides lorsqu'ils sont eux-mêmes déplacés. Comme en général c'est le comportement

que l'on recherche, les liens symboliques sont toujours créés avec des références absolues, mais vous êtes libre de faire autrement si vous en ressentez le besoin. Sachez cependant que déplacer une source de données n'est jamais une bonne idée. Le tableau suivant récapitule les avantages et les inconvénients des différents types de liens :

Tableau 4-1. Caractéristiques des liens physiques et symboliques

		Référence relative	Référence absolue
Fonctionnalité	Liens physiques	Liens symboliques	
Peuvent être déplacés	Oui	Avec la cible	Oui
Suivent la cible	Oui	Si déplacés avec elle	Non
Gèrent la suppression de la cible	Oui	Non	Non
Peuvent référencer des cibles sur un autre système de fichiers	Non	Oui	Oui
Peuvent référencer des répertoires	Non	Oui	Oui

IV-C-3 - Autres fonctionnalités

Les systèmes de fichiers de Linux fournissent également des fonctionnalités qui peuvent être intéressantes, mais que l'on utilise généralement moins souvent.

La première de ces fonctionnalités est les quotas. Ceux-ci permettent de définir la proportion d'espace disque que chaque utilisateur peut consommer dans le système de fichiers. Ainsi, il est possible de restreindre les utilisateurs et d'éviter qu'un seul d'entre eux ne consomme l'ensemble des ressources disque. En pratique, cette fonctionnalité n'est pas d'une grande utilité pour un particulier, qui souvent est le seul utilisateur de sa machine.

Note : Sachez toutefois que les systèmes de fichiers réservent généralement une petite partie (environ 5%) de l'espace disque pour l'administrateur du système, afin de s'assurer que celui-ci pourra toujours effectuer les opérations de maintenance en cas de saturation du système de fichiers. Cet espace libre est également utilisé pour les mécanismes de défragmentation automatique des systèmes de fichiers.

Les systèmes de fichiers permettent également d'enregistrer des informations complémentaires sur les fichiers, afin par exemple de pouvoir les marquer pour un traitement spécifique. Ces informations complémentaires sont stockées sous la forme « d'attributs », pouvant contenir virtuellement n'importe quel type d'information. Certains systèmes de fichiers utilisent cette fonctionnalité pour implémenter les mécanismes de sécurité basés sur les ACL.

Enfin, les systèmes de fichiers Unix sont également capables de prendre en charge ce que l'on appelle les fichiers « troués » (« sparse files » en anglais). Ces fichiers sont des fichiers contenant des données séparées par de grands espaces vides. On peut donc dire qu'ils sont « presque vides ». Pour ces fichiers, il est évident qu'il est inutile de stocker les espaces vides sur le disque. Les systèmes de fichiers mémorisent donc tout simplement qu'ils contiennent des trous, et ils ne stockent que les données réelles et la position des trous avec leurs tailles. Cela constitue une économie de place non négligeable. Les applications classiques des fichiers presque vides sont les bases de données, qui utilisent souvent des fichiers structurés contenant relativement peu de données effectives, et les images disque pour installer des systèmes d'exploitation dans des machines virtuelles.

IV-D - Structure du système de fichiers

Nous allons à présent nous intéresser à l'organisation du système de fichiers de Linux. Ce système de fichiers contient un certain nombre de répertoires et de fichiers standards, qui ont chacun une fonction bien définie. Cette structure standard permet de gérer tous les systèmes Linux de manière homogène : les programmes savent où trouver ce dont ils ont besoin. Ils sont donc portables d'un système à un autre, et les utilisateurs ne sont pas dépayés lorsqu'ils changent de machine (en fait, cette structure est à peu près la même pour tous les systèmes Unix, ce qui donne encore plus de poids à l'argument précédent).

Cette organisation standard du système de fichiers a été conçue de telle manière que les données et les programmes sont placés chacun à leur place, et qu'ils puissent être facilement intégrés dans un réseau. L'intégration dans un réseau sous-entend que les fichiers des programmes peuvent être partagés par les différentes machines de ce réseau, ce qui permet d'économiser beaucoup de place. De plus, il est possible d'accéder à la plupart des ressources du système grâce à une interface uniforme. En particulier, il est possible d'accéder à tous les périphériques installés sur l'ordinateur par l'intermédiaire de fichiers spéciaux, et on peut récupérer des informations sur le système mises à disposition par le noyau simplement en lisant des fichiers générés par un pseudo système de fichiers. Le tableau suivant décrit les principaux éléments de l'arborescence du système de fichiers de Linux.

Tableau 4-2. Hiérarchie standard du système de fichiers

Répertoire	Signification
/	Répertoire racine. Point de départ de toute la hiérarchie du système de fichiers. Le système de fichiers contenant ce répertoire est monté automatiquement par le noyau pendant l'amorçage du système. Ce système de fichiers est appelé système de fichiers racine (« root » en anglais).
/boot/	Répertoire contenant le noyau de Linux et ses informations de symboles. Ce répertoire est parfois le point de montage d'un système de fichiers de très petite taille, dédié au noyau. Dans ce cas, il est recommandé que le système de fichiers correspondant soit monté en lecture seule. On notera que sur certains systèmes, le noyau reste placé dans le répertoire racine. Cette technique n'est pas recommandée, car on ne peut pas monter en lecture seule la partition racine en utilisation normale du système.
/boot/vmlinuz	Noyau compressé de Linux. Les noyaux compressés se décompressent automatiquement lors de l'amorçage du système. Sur certains systèmes, le noyau est encore placé dans le répertoire racine du système de fichiers.
/boot/System.map	Fichier système contenant la liste des symboles du noyau. Ce fichier est utilisé par certains programmes donnant des renseignements sur le système. En particulier, il est utilisé par le programme « top » (programme qui indique la liste des principaux processus actifs) afin de donner le nom de la fonction du noyau dans lequel un processus se trouve bloqué lorsqu'il est en attente de la fin d'une opération que le noyau doit exécuter pour lui.
/dev/	Répertoire contenant tous les fichiers spéciaux permettant d'accéder aux périphériques. Comme on l'a vu, hormis les interfaces réseau, les périphériques sont généralement accessibles au travers de fichiers spéciaux. Ces fichiers permettent l'envoi et la réception des données vers les périphériques associés, et ce de manière uniforme. Tous ces fichiers spéciaux sont tous placés dans le répertoire /dev/. Généralement, le répertoire /dev/ est un système de fichiers virtuel (c'est-à-dire un système de fichiers qui

	est complètement géré en mémoire par le noyau), qui est rempli dynamiquement par le système en fonction du matériel installé. Le répertoire <code>/dev/</code> ne contient donc que les fichiers spéciaux des périphériques pour lesquels le noyau dispose effectivement d'un pilote de périphérique chargé. Toutefois, il est également possible que ce répertoire soit effectivement prérempli avec les fichiers spéciaux de la plupart des périphériques existants sur le marché, même ceux qui ne sont pas physiquement présents dans la machine. Dans ce cas, les opérations sur les fichiers spéciaux des périphériques non installés seront tout simplement refusées par le noyau. Ce type de configuration se rencontre encore assez souvent sur des installations assez anciennes. Quoi qu'il en soit, quelle que soit la technique utilisée, ce répertoire doit être impérativement placé dans le système de fichiers racine.
<code>/sbin/</code>	Répertoire contenant les commandes systèmes nécessaires à l'amorçage et réservées à l'administrateur. Ce répertoire doit être impérativement placé dans le système de fichiers racine. En général, seul l'administrateur utilise ces commandes.
<code>/bin/</code>	Répertoire contenant les commandes systèmes générales nécessaires à l'amorçage. Ce répertoire doit être impérativement placé dans le système de fichiers racine. Tous les utilisateurs peuvent utiliser les commandes de ce répertoire.
<code>/lib/</code>	Répertoire contenant les bibliothèques partagées (« DLL » en anglais, pour « Dynamic Link Library ») utilisées par les commandes du système des répertoires <code>/bin/</code> et <code>/sbin/</code> . Ce répertoire doit être impérativement placé dans le système de fichiers racine.
<code>/lib64/</code>	Sur les systèmes 64 bits basés sur une architecture x86_64 (processeurs AMD Athlon64 ou ultérieurs, ou processeurs Intel avec jeu d'instruction EMT64), répertoire équivalent au répertoire <code>/lib/</code> pour les bibliothèques 64 bits natives. Ces machines étant capables d'exécuter des programmes 32 bits en plus des programmes natifs 64 bits, il est nécessaire de fournir deux jeux de bibliothèques pour les deux modes de fonctionnement. Normalement, le répertoire <code>/lib/</code> devrait être utilisé pour les bibliothèques natives, mais il est d'usage de continuer à y placer les bibliothèques 32 bits pour des raisons de compatibilité ascendante, les programmes 32 bits cherchant naturellement leurs bibliothèques dans ce répertoire.
<code>/lib32/</code>	Sur les systèmes 64 bits natifs basés sur une architecture x86_64 (processeurs AMD Athlon64 ou ultérieurs, ou processeurs Intel avec jeu d'instruction EMT64), répertoire des bibliothèques 32 bits si les bibliothèques 64 bits natives sont placées dans le répertoire <code>/lib/</code> . Attention, sur les systèmes qui utilisent ce type d'organisation, les paquetages 32 bits natifs ne peuvent pas être installés sans modification, car ils placeraient leurs bibliothèques 32 bits dans le répertoire <code>/lib/</code> , qui contient dans ce cas des bibliothèques 64 bits. Réaliser une installation de paquetages non modifiés dans ces conditions peut écraser les bibliothèques 64 bits natives et donc provoquer de graves dysfonctionnements.
<code>/lib/modules/</code>	Ce répertoire contient les modules additionnels du noyau. Les modules sont des composants logiciels du noyau.

	mais qui ne sont pas chargés immédiatement pendant la phase d'amorçage du système. Ils peuvent en revanche être chargés et déchargés dynamiquement, lorsque le système est en fonctionnement. Il est fortement recommandé que ce répertoire soit placé dans le système de fichiers racine.
<code>/etc/</code>	Répertoire contenant tous les fichiers de configuration du système. Ce répertoire doit être impérativement placé dans le système de fichiers racine.
<code>/etc/X11/</code>	Répertoire contenant les fichiers de configuration de l'environnement graphique XWindow.
<code>/etc/rc.d/</code>	Répertoire contenant les scripts de démarrage du système. Ces scripts sont exécutés lorsque le système démarre ou s'arrête, ainsi que lorsqu'il change de mode de fonctionnement. Il est également possible de les exécuter pour démarrer ou arrêter un service particulier. Dans certaines distributions, ces fichiers sont placés dans le répertoire <code>/sbin/init.d/</code> .
<code>/etc/opt/</code>	Répertoire contenant les fichiers de configuration des applications.
<code>/tmp/</code>	Répertoire permettant de stocker des données temporaires. En général, <code>/tmp/</code> ne contient que des données très éphémères. Il est préférable d'utiliser le répertoire <code>/var/tmp/</code> . En effet, le répertoire <code>/tmp/</code> ne dispose pas nécessairement de beaucoup de place disponible.
<code>/usr/</code>	Répertoire contenant les fichiers du système partageables en réseau et en lecture seule.
<code>/usr/bin/</code>	Répertoire contenant la plupart des commandes des utilisateurs et des logiciels installés.
<code>/usr/sbin/</code>	Répertoire contenant les commandes systèmes non nécessaires à l'amorçage. Ces commandes ne sont normalement utilisées que par l'administrateur système.
<code>/usr/lib/</code>	Répertoire contenant les bibliothèques partagées de tous les programmes de <code>/usr/bin/</code> et <code>/usr/sbin/</code> et les bibliothèques statiques pour la création de programmes.
<code>/usr/lib64/</code>	Sur les systèmes 64 bits basés sur une architecture x86_64 (processeurs AMD Athlon64 ou ultérieurs, ou processeurs Intel avec jeu d'instruction EMT64), répertoire équivalent au répertoire <code>/usr/lib/</code> pour les bibliothèques 64 bits natives. La raison pour laquelle les bibliothèques 64 bits natives de ces systèmes n'est pas placée dans le répertoire <code>/usr/lib/</code> est la même que pour les bibliothèques du répertoire <code>/lib64/</code> .
<code>/usr/lib32/</code>	Sur les systèmes 64 bits natifs basés sur une architecture x86_64 et utilisant le répertoire <code>/usr/lib/</code> pour les bibliothèques 64 bits natives, répertoire contenant les bibliothèques 32 bits. Les mêmes précautions d'usage que celles indiquées pour le répertoire <code>/lib32/</code> s'appliquent ici.
<code>/usr/include/</code>	Répertoire contenant les fichiers d'en-têtes du système pour le compilateur C/C++. Les fichiers de ce répertoire sont utilisés pour réaliser des programmes dans les langages de programmation C et C++.
<code>/usr/X11/</code>	Répertoire historique qui contenait l'environnement graphique X11 et ses applications sur les anciennes distributions. Dans les distributions récentes (environnement X11 basé sur X.org de version 7.0 ou plus), X11 est à présent directement installé dans le répertoire <code>/usr/</code> , et ce répertoire n'est plus conservé qu'à titre de compatibilité sous la forme de liens

	symboliques. Ce répertoire contient des sous-répertoires <code>bin/</code> , <code>lib/</code> et <code>include/</code> , où se trouvent les exécutables de XWindow, les bibliothèques et les fichiers d'en-têtes pour créer des programmes pour XWindow en C et C++.
<code>/usr/share/</code>	Répertoire contenant l'ensemble des données et des fichiers partagés par les applications du système. C'est ici que sont par exemple stockées les polices de caractères, les sons et les images utilisés par les environnements de bureau, les fonds d'écran, les fichiers d'aide ou encore les associations de fichiers avec leurs extensions.
<code>/usr/src/</code>	Répertoire contenant les fichiers sources du noyau et des applications de la distribution. Normalement, ce répertoire ne doit contenir que le code source des applications dépendantes de la distribution que vous utilisez.
<code>/usr/src/linux/</code>	Sources du noyau de Linux. Il est vivement recommandé de conserver les sources du noyau de Linux sur son disque, afin de pouvoir changer la configuration du système à tout moment.
<code>/usr/local/</code>	Répertoire contenant les programmes d'extension du système indépendants de la distribution. Ce n'est toutefois pas le répertoire d'installation des applications, que l'on installera en général dans le répertoire <code>/opt/</code> . « local » ne signifie pas ici que les programmes qui se trouvent dans ce répertoire ne peuvent pas être partagés sur le réseau, mais plutôt que ce sont des extensions du système qu'on ne trouve donc que localement sur un site donné. Ce sont donc les extensions qui ne font pas partie de la distribution de Linux utilisée, et qui doivent être conservées lors des mises à jour ultérieures de cette distribution. Ce répertoire contient les sous-répertoires <code>bin/</code> , <code>lib/</code> , <code>include/</code> et <code>src/</code> , qui ont la même signification que les répertoires du même nom de <code>/usr/</code> , à ceci près qu'ils ne concernent que les extensions locales du système, donc indépendantes de la distribution.
<code>/var/</code>	Répertoire contenant toutes les données variables du système. Ce répertoire contient les données variables qui ne pouvaient pas être placées dans le répertoire <code>/usr/</code> , puisque celui-ci est normalement accessible en lecture seule.
<code>/var/tmp/</code>	Répertoire contenant les fichiers temporaires. Il est préférable d'utiliser ce répertoire plutôt que le répertoire <code>/tmp/</code> .
<code>/var/opt/</code>	Répertoire contenant les données variables des applications.
<code>/var/log/</code>	Répertoire contenant les traces de tous les messages système. C'est dans ce répertoire que l'on peut consulter les messages d'erreurs du système et des applications.
<code>/var/spool/</code>	Répertoire contenant les données en attente de traitement. Les travaux d'impression en cours, les mails et les fax en attente d'émission, les travaux programmés en attente d'exécution sont tous stockés dans ce répertoire.
<code>/var/locks/</code>	Répertoire contenant les verrous sur les ressources système. Certaines ressources ne peuvent être utilisées que par une seule application (par exemple, un modem). Les applications

	qui utilisent de telles ressources le signalent en créant un fichier de verrou dans ce répertoire.
<code>/var/cache/</code>	Répertoire contenant les données de résultats intermédiaires des applications. Les applications qui doivent stocker des résultats intermédiaires doivent les placer dans ce répertoire.
<code>/opt/</code>	Répertoire historique contenant les applications qui ne font pas réellement partie du système d'exploitation. En particulier, sur les anciennes distributions, le gestionnaire de bureau KDE était installé dans le sous-répertoire <code>/opt/kde/</code> , mais à présent il est considéré comme partie intégrante du système et est donc installé directement dans le répertoire <code>/usr/</code> sur les distributions récentes.
<code>/home/</code>	Répertoire contenant les répertoires personnels des utilisateurs. Il est fortement recommandé de placer ce répertoire dans un système de fichiers indépendant de ceux utilisés par le système. Cela permet de faire des sauvegardes plus facilement, et de faire les mises à jour du système de manière sûre, sans craindre de perdre les données des utilisateurs.
<code>/root/</code>	Répertoire contenant le répertoire personnel de l'administrateur. Il est donc recommandé que le répertoire personnel de l'administrateur soit placé en dehors de <code>/home/</code> pour éviter qu'un problème sur le système de fichiers des utilisateurs ne l'empêche de travailler. Toutefois, il est important que l'administrateur puisse travailler même si les répertoires <code>/root/</code> et <code>/home/root/</code> ne sont pas présents. Dans ce cas, son répertoire personnel sera le répertoire racine.
<code>/media/</code>	Répertoire réservé au montage des systèmes de fichiers sur périphériques amovibles (CD-ROM, disquettes, etc.). Ce répertoire peut contenir plusieurs sous-répertoires pour chaque périphérique amovible, afin de permettre d'en monter plusieurs simultanément. Notez qu'il est assez courant de disposer de liens symboliques dans la racine référençant les principaux systèmes de fichiers, afin d'en simplifier l'accès. Par exemple, il est courant d'avoir un répertoire <code>/floppy/</code> référençant le lecteur de disquette et un répertoire <code>/cdrom/</code> référençant le lecteur de CD-ROM.
<code>/mnt/</code>	Répertoire réservé à l'administrateur pour le montage temporaire des systèmes de fichiers lors de leur maintenance (disques d'installation, disques durs externes qui ne doivent pas être manipulables par les utilisateurs, etc.). Ce répertoire peut contenir plusieurs sous-répertoires pour chaque périphérique hébergeant les systèmes de fichiers en question, afin de permettre d'en monter plusieurs simultanément. Notez que ce répertoire contenait historiquement les mêmes entrées que le répertoire <code>/media/</code> , qui a été introduit récemment afin de séparer les opérations de montage réalisées dans un but purement administratif des opérations de montage réalisées dans un contexte d'utilisation normale des systèmes de fichiers sur périphériques amovibles.
<code>/lost+found/</code>	Répertoire contenant les données récupérées lors de la réparation d'un système de fichiers endommagé. Ces données sont écrites par les utilitaires de vérification et de réparation des systèmes de fichiers lorsqu'ils trouvent des informations qui ne peuvent être rattachées à aucun fichier.

	existant, ainsi, il est possible de récupérer ces informations si elles ne sont pas dans un état de détérioration trop avancé.
<code>/proc/</code>	Répertoire contenant le pseudo système de fichiers du noyau. Ce pseudo système de fichiers contient des fichiers permettant d'accéder aux informations sur le matériel, la configuration du noyau et sur les processus en cours d'exécution.
<code>/sys/</code>	Répertoire contenant le pseudo système de fichiers des gestionnaires de périphériques. Ce pseudo système de fichiers contient des fichiers permettant d'obtenir des informations sur l'ensemble des objets du noyau, en particulier sur l'ensemble des périphériques de l'ordinateur. Ce système de fichiers est appelé à recevoir une bonne partie des informations exposées par le système de fichiers <code>/proc/</code> , dont le rôle se restreindra sans doute à fournir des informations plus générales et à réaliser la configuration des fonctions générales du système.

Note : Les informations données ici peuvent ne pas être correctes pour votre distribution. En effet, certaines distributions utilisent une structure légèrement différente. Les informations données ici sont conformes à la norme de hiérarchie de systèmes de fichiers version 2.3 (« FHS » en anglais). Vous pouvez consulter ce document pour une description exhaustive du système de fichiers de Linux.

Vous avez pu constater que les répertoires `bin/`, `lib/`, `include/` et `src/` apparaissent régulièrement dans la hiérarchie du système de fichiers. Cela est normal : les répertoires sont classés par catégorie d'applications et par importance. Les répertoires `bin/` contiennent en général les programmes, et les répertoires `lib/` les bibliothèques partagées par ces binaires. Cependant, les répertoires `lib/` peuvent aussi contenir des bibliothèques statiques, qui sont utilisées lors de la création de programmes. En général, tous les systèmes Unix fournissent en standard un compilateur pour générer ces programmes. Dans le cas de Linux, ce compilateur est « gcc » (pour « GNU C Compiler »). La création d'un programme nécessite que l'on dispose des fichiers source, qui contiennent le programme écrit dans un langage de programmation, des fichiers d'en-têtes, qui contiennent les déclarations de toutes les fonctions utilisables, et des fichiers de bibliothèques statiques, contenant ces fonctions. Ces différents fichiers sont stockés respectivement dans les répertoires `src/`, `include/` et `lib/`. Les notions de sources et de compilation seront décrites en détail dans le [Chapitre 7](#).

V - Commandes Unix de base

Mais pour cela, il va falloir que vous appreniez les commandes Unix de base et la manière d'utiliser un système Linux en ligne de commande (c'est-à-dire en mode texte, sans XWindow).

V-A - Login et déconnexion

Comme il l'a déjà été dit, Linux est un système multiutilisateur. Il faut donc que chacun s'identifie pour que le système puisse fonctionner correctement. Cela est réalisé lors de l'opération dite de login (du verbe anglais « to log in », qui signifie « s'enregistrer » dans le système). Le login consiste essentiellement à taper son nom d'utilisateur, valider, et répondre éventuellement à la demande de mot de passe de la part du système.

Le login doit être la première opération à effectuer. Il est impossible d'accéder au système d'une autre manière, et la vérification du mot de passe fournit l'authenticité de l'utilisateur qui se logue. Ainsi, le système sait en permanence au nom de quelle personne il effectue les opérations demandées. Cette opération est à la base des mécanismes de sécurité et de personnalisation du système pour chaque utilisateur.

Il existe deux types de login. Le plus courant est le login en mode texte, qui peut être fait directement sur le poste local ou à travers un réseau. Le système vous invite à vous identifier avec la ligne suivante :

login:

D'autres informations peuvent être affichées avant le mot login, qui peuvent vous renseigner sur la nature du système. Quoi qu'il en soit, vous devez taper votre nom d'utilisateur (que l'on appelle simplement « login »), ou « root » si vous désirez vous connecter en tant qu'administrateur. Le système vous demande alors le mot de passe avec la ligne suivante :

password:

Bien entendu, vous ne devrez jamais oublier votre mot de passe administrateur. Si toutefois cela vous arrive, vous n'aurez plus qu'une seule solution : démarrer l'ordinateur à partir d'une disquette système, et faire le ménage dans le fichier de mots de passe. Cette opération n'est jamais très agréable à réaliser. Conclusion : n'oubliez jamais votre mot de passe.

Le deuxième type de login est le login graphique, sous X11. Ce type de login a en général lieu sur un terminal X (c'est-à-dire un terminal graphique). La procédure peut varier selon l'environnement utilisé, mais le principe reste toujours le même : il faut fournir son nom d'utilisateur et son mot de passe.

Si, comme la plupart des gens, vous ne cherchez pas à utiliser votre ordinateur à distance à travers un réseau, vous vous connecterez quasiment toujours en local. Linux fournit, pour l'utilisateur local, plusieurs terminaux virtuels. Cela signifie qu'il est possible de se connecter plusieurs fois dans le système dans des terminaux différents. Pour passer d'un terminal virtuel à un autre, il suffit de taper les combinaisons de touches ALT+DROITE ou ALT+GAUCHE, où DROITE et GAUCHE sont respectivement les flèches du curseur droite et gauche. Il est également possible d'accéder à un terminal donné à l'aide de la combinaison de touches ALT+Fn, où Fn est l'une des touches de fonction F1, F2, F3, etc. La plupart des distributions utilisent au moins quatre terminaux virtuels, plus un terminal X. Le terminal X est le terminal graphique, qui fonctionne sous XWindow. Vous noterez sans doute que lorsqu'on est sous XWindow, les combinaisons ALT+Fn ont une autre signification. Elles ne peuvent donc pas être utilisées pour basculer vers les terminaux en mode texte. Pour remédier à ce problème, une autre combinaison de touches a été définie, spécialement pour XWindow : CTRL+ALT+Fn. Il suffit donc simplement d'utiliser la touche CTRL en plus de la touche ALT.

L'utilisation des terminaux virtuels est très pratique, même pour un seul utilisateur. En effet, ceux-ci permettent de lancer plusieurs programmes simplement, à raison d'un par terminal virtuel, et de s'y retrouver ainsi plus facilement. Pour ceux qui ne connaissent pas les systèmes Unix, il est recommandé de jouer un peu avec les terminaux virtuels afin de simuler la présence de plusieurs utilisateurs. Ils auront ainsi un aperçu de la puissance de ces systèmes.

Lorsqu'on a fini de travailler, il faut se déconnecter. Cette opération est très simple pour les terminaux non graphiques, puisqu'il suffit de taper la commande suivante :

```
logout
```

Si d'aventure cette commande ne fonctionnait pas, vous pourrez utiliser la commande exit ou la combinaison de touches CTRL+d, qui terminent le shell courant (y compris le shell de login).

Pour les terminaux X, le processus de déconnexion dépend de l'environnement utilisé. Il faut tâtonner un peu, et normalement on trouve une option de menu du style « logout » ou « déconnexion ». Vous pouvez par exemple cliquer sur le bouton droit de la souris sur le bureau de l'environnement de travail, afin d'appeler un menu contextuel. Dans bien des cas, ce menu contient une option de déconnexion.

Il est très important de se déconnecter et de ne jamais laisser une session ouverte. En effet, cette négligence peut vous coûter cher, car une personne mal intentionnée pourrait très bien utiliser ce terminal à vos dépens. Il aurait tous vos droits, et effectuerait ses opérations en votre nom. La sécurité du système garantissant que vous seul pouvez vous connecter sous ce nom, grâce au mot de passe, vous seriez donc responsable des agissements de l'intrus. Bien entendu, ce genre de considération n'a pas autant d'importance pour un particulier que dans une entreprise ou une collectivité quelconque.

V-B - Arrêt et redémarrage du système

Il faut bien comprendre que Linux, tout comme la plupart des systèmes d'exploitation modernes, ne peut pas être arrêté en éteignant directement l'ordinateur, comme on le faisait autrefois avec le DOS. En effet, la plupart des systèmes d'exploitation utilisent une partie de la mémoire de l'ordinateur pour y stocker temporairement les données qui ont été lues à partir du disque et celles qui doivent y être écrites. Cette zone de mémoire constitue ce qu'on appelle un tampon (« buffer » en anglais), et elle sert à accélérer les accès aux périphériques plus lents, que sont les disques durs et lecteurs de CD-ROM. Il va de soi qu'une requête de lecture sur des données déjà situées en mémoire est infiniment plus rapide que si elles ne s'y trouvaient pas. Il est en revanche plus difficile de comprendre pourquoi les requêtes d'écriture doivent être différées. La raison est la suivante : le système préfère différer l'écriture physique sur le disque parce qu'une autre requête d'écriture dans la même zone du disque peut très bien avoir lieu ultérieurement. Si les données qui n'ont pas été écrites sont ainsi modifiées par une requête ultérieure, il n'est plus nécessaire de les écrire, et ainsi le système peut économiser un temps précieux en ne le faisant pas. Si les données à écrire sont contiguës à celles d'une requête précédente, le système peut les écrire en bloc, ce qui est toujours plus rapide que de faire plusieurs écritures partielles (notamment parce que les têtes de lecture du disque n'ont pas à être déplacées). Enfin, si les données qui doivent être écrites font l'objet d'une requête de lecture, il va de soi qu'elles sont immédiatement accessibles. On voit que cette stratégie permet de travailler beaucoup plus vite. De facto, Linux utilise toute la mémoire vive libre pour ses tampons d'entrées / sorties, ce qui en fait un système extrêmement performant. Le gain en performances peut facilement atteindre un facteur 3 ou 4.

Le problème majeur est évidemment que si l'on éteint l'ordinateur brutalement, les données dont l'écriture a été différée sont perdues. Pire, parmi ces données, il est probable qu'il y ait des informations vitales pour le système de fichiers, ce qui fait qu'il risque fort d'être endommagé. Les systèmes de fichiers journalisés comme EXT3 et ReiserFS sont à l'abri de ce type d'erreur en raison de l'accès transactionnel aux structures de données du système de fichiers qu'ils utilisent, et le système parvient généralement à réparer les autres systèmes de fichiers lors de la vérification qui est lancée au redémarrage suivant de la machine, mais il est inutile de prendre des risques. Tout cela signifie qu'il est impératif de prévenir le système avant de l'arrêter, pour qu'il puisse écrire les données situées dans ses tampons.

L'arrêt du système est une opération qui est du ressort de l'administrateur. On ne peut donc le réaliser que sous le compte root. Plusieurs commandes sont disponibles, les plus simples sont données ci-dessous :

- halt, qui permet d'arrêter le système.
- reboot, qui permet de le redémarrer.

Ces commandes sont en fait des scripts permettant d'effectuer les opérations d'arrêt et de redémarrage du système rapidement. Si elles ne sont pas disponibles sur votre distribution, vous devrez sans doute utiliser la commande générique suivante :

```
shutdown [-r] now
```

où l'option -r permet de demander un redémarrage et non un arrêt simple.

Il est également possible que votre gestionnaire de bureau vous donne le moyen d'arrêter l'ordinateur par l'intermédiaire de l'interface graphique de X11. La technique à utiliser dépend évidemment de l'environnement que vous aurez installé, et elle ne sera pas décrite ici. Consultez la documentation de votre distribution pour plus de détails à ce sujet. De plus, la plupart des distributions provoquent un redémarrage de la machine lorsqu'on appuie sur les touches **CTRL+ALT+SUPPR** simultanément dans un terminal virtuel.

V-C - Pages de manuel

Maintenant que vous savez l'essentiel pour conserver votre système en bon état, nous allons traiter des autres commandes Unix. Parmi elles, il en est qui sont certainement fondamentales : ce sont les commandes qui permettent d'obtenir de l'aide !

Chaque commande Unix a une page de manuel qui la décrit. Ces pages sont très souvent écrites en anglais, mais elles sont très précises et fournissent toutes les informations dont on peut avoir besoin. Pour afficher la page de manuel d'une commande, il suffit d'utiliser la commande suivante :

```
man page
```

où `page` est la page de manuel de la commande sur laquelle on cherche des informations. En général, le nom de la page de manuel est le même que celui de la commande. Par exemple, pour afficher l'aide sur la commande `cp`, il suffit de taper :

```
man cp
```

Lorsqu'une page de man est affichée, il est possible de faire défiler son texte à l'aide des touches du curseur. Pour quitter l'aide, il suffit d'appuyer sur la touche `q`.

Les pages de man sont classées en groupes de pages thématiques, chaque groupe étant identifié généralement par un numéro ou une lettre. Si la page de man affichée ne correspond pas à celle désirée, c'est qu'une page homonyme d'un autre groupe a été utilisée. Dans ce cas, il faut préciser l'identificateur du groupe de pages de manuel avant le nom de la page à afficher :

```
man groupe page
```

où `groupe` est l'identificateur du groupe auquel la page de manuel appartient. Les principaux groupes sont les suivants :

Tableau 5-1. Groupes de pages de man

Identificateur	Type de pages de manuel
1	Commandes utilisateur
2	Appels systèmes (programmation en C)
3	Fonctions de la bibliothèque C
4	Description des fichiers spéciaux
5	Description des fichiers de configuration
6	Jeux et programmes divers
7	Programmes systèmes divers
8	Administration système

Si vous ne savez pas dans quel groupe se trouve une page de manuel, vous pouvez utiliser l'option `-k`, qui permet d'afficher l'ensemble des pages disponibles portant ce nom :

```
man -k commande
```

L'identificateur du groupe est en général précisé entre parenthèses, à la suite du nom de la page de manuel.

Il se peut également que vous recherchiez de l'aide sur un sujet donné, mais que vous ne connaissiez pas le nom exact de la page de manuel qui en parle. Pour ce genre de recherche, vous pourrez utiliser le programme `apropos`, qui recherchera toutes les pages de manuel qui contiennent un mot clé particulier. Ce programme s'utilise avec la syntaxe suivante :

```
apropos mot
```

où `mot` est le mot clé à rechercher dans toutes les pages de manuel.

La commande `man` est la commande d'aide standard sur tous les systèmes Unix. Cependant, Linux utilise un grand nombre de commandes écrites sous la licence GNU, et qui utilisent un format d'aide spécifique à GNU. L'aide pour ces commandes peut être obtenue par la commande suivante :

```
info commande
```

Il se peut que les deux méthodes fonctionnent. Dans ce cas, la page de `man` sera certainement moins récente que la page d'`info`, car la commande que vous utilisez est sans aucun doute une commande GNU, qui a été fournie avec sa page d'information. Il est donc recommandé de lire plutôt la page d'information GNU.

Le format d'aide GNU est plus riche que celui de `man`, puisqu'il permet de naviguer dans le système d'aide à l'aide de liens hypertextes. Ces liens sont organisés hiérarchiquement, avec des chapitres et des sous-chapitres. Chaque chapitre dispose d'une forme de table des matières constituée de menus, qui permettent d'accéder aux sous-chapitres. Les menus se distinguent du texte normal par un astérisque (« * ») en début de ligne dans la table des matières. Les commandes clavier suivantes pourront vous être utiles pour naviguer dans la hiérarchie du système d'aide de GNU :

- la touche de tabulation permet de passer au lien hypertexte suivant ;
- la touche `n` (pour « Next ») permet de passer au chapitre suivant ;
- la touche `p` (pour « Previous ») permet de revenir au chapitre précédent ;
- la touche `u` (pour « Up ») permet de remonter d'un niveau dans le système d'aide et d'atteindre la table des matières référençant le chapitre courant.

Enfin, la commande `q` permet de quitter le système d'aide.

V-D - Opérations de base sur les répertoires

Ce paragraphe va vous décrire les opérations de base qu'il faut savoir faire pour manipuler les répertoires du système de fichiers.

La première commande est évidemment celle qui permet de lister le contenu d'un répertoire. Elle dispose d'un grand nombre d'options :

```
ls [options] [fichier]
```

où `fichier` est le nom d'un fichier ou d'un répertoire que l'on désire lister. Si ce paramètre est absent, `ls` affichera tous les fichiers du répertoire courant. Les principales options sont `-l`, qui permet d'afficher des informations étendues (notamment les propriétaires, les groupes, les droits, la taille et éventuellement les liens), et `-a`, qui permet d'afficher tous les fichiers, y compris les fichiers cachés (ceux dont le nom commence par un point).

La deuxième commande est celle qui permet de changer de répertoire courant. Sa syntaxe est très simple :

```
cd [chemin]
```

où `chemin` est un chemin de répertoire Unix valide. Ce chemin est constitué des noms des répertoires et sous-répertoires successifs, séparés par des barres obliques « / ». Si aucun chemin n'est spécifié, cette commande change le répertoire courant pour le répertoire personnel de l'utilisateur. Par exemple, pour aller dans le répertoire d'installation de XWindow, il faut taper la commande suivante :

```
cd /usr/X11
```

La notion de chemin sera détaillée dans le paragraphe suivant. « `cd` » est l'abréviation de l'anglais « Change Directory ».

La troisième commande permet de créer un répertoire :

```
mkdir chemin
```

où `chemin` est le chemin spécifiant le répertoire à créer. Si le chemin ne contient que le nom du répertoire à créer, celui-ci est créé dans le répertoire courant et devient donc un sous-répertoire. « `mkdir` » est l'abréviation de l'anglais « **Ma**Ke **DI**Rectory »).

La commande pour supprimer un répertoire est la suivante :

```
rmdir chemin
```

Pour supprimer un répertoire, il faut qu'il soit vide (c'est-à-dire qu'il ne contienne ni fichier ni répertoire). « `rmdir` » est l'abréviation de l'anglais « **Re**Mov**e** **DI**Rectory ».

Enfin, voici une commande dont vous ne vous servirez normalement que très peu, voire pas du tout. Elle permet d'afficher le répertoire courant :

```
pwd
```

Cette commande n'est a priori pas très utile, car le shell affiche toujours le répertoire courant sur la plupart des distributions. Cependant, le chemin affiché par le shell étant relatif au répertoire personnel de l'utilisateur lorsqu'on se trouve dans un sous-répertoire de celui-ci, la commande `pwd` peut être utile lorsqu'on désire obtenir un chemin absolu sur le répertoire courant. « `pwd` » est l'abréviation de l'anglais « **P**rint **W**orking **D**irectory ». Cette commande est également utilisée par les scripts pour déterminer le répertoire à partir duquel ils sont exécutés.

V-E - Notions sur les chemins Unix

Les chemins Unix permettent de qualifier complètement un répertoire ou un fichier sur le disque. Comme on l'a vu, ils utilisent pour cela les noms de ces répertoires et de ces fichiers, et ils les combinent pour indiquer comment atteindre la cible dans le système de fichiers. Sous Unix, le séparateur utilisé pour séparer les différents constituants du nom est la barre oblique (« `/` »). Le répertoire racine n'a pas de nom, et peut donc être référencé par une barre oblique seule.

Les chemins peuvent être absolus (c'est-à-dire qu'ils peuvent partir du répertoire racine) ou relatifs (c'est-à-dire qu'ils peuvent partir du répertoire courant). Si l'on utilise un chemin relatif, il faut savoir que le répertoire courant est désigné par un point (« `.` »), et que le répertoire parent du répertoire courant est désigné par deux points successifs (« `..` »). Ainsi, si l'on est dans le répertoire `/usr/local/bin`, on peut accéder au répertoire `/usr/X11/bin` avec les deux chemins suivants :

```
/usr/X11/bin
```

ou :

```
../../X11/bin
```

Le premier chemin est absolu, parce qu'il part directement du répertoire racine. Le deuxième chemin est relatif, car il part du répertoire courant.

Note : Il va de soi que les chemins relatifs ne sont valides, sauf coup de chance, que dans le répertoire dans lequel ils sont écrits, alors que les chemins absolus sont toujours valables. En revanche, si des répertoires sont déplacés ensemble, les chemins relatifs entre ces répertoires restent valides, mais les chemins absolus deviennent faux. Toutefois, ces considérations ne concernent pas un utilisateur de base.

La plupart des shells sont capables d'effectuer ce que l'on appelle la complétion automatique des commandes. La complétion automatique permet de n'écrire qu'une partie des noms de fichiers ou de répertoires et de demander au shell de compléter ces noms. Cela peut se faire de deux manières. La première solution, qui est aussi la plus simple, consiste à taper le début du nom, puis d'utiliser une touche spéciale qui permet de demander au shell de le

compléter. Si vous utilisez le shell bash (bash est le shell de prédilection sur les systèmes Linux), cette touche est la touche de tabulation. Ainsi, si vous tapez :

```
cd /ho
```

et que vous appuyez sur la touche de tabulation, bash complétera cette ligne de la manière suivante :

```
cd /home/
```

Pour cela, il regarde la liste des fichiers et des répertoires qui commencent par « `ho` » dans le répertoire racine. Normalement, il ne s'y trouve que le répertoire `/home/`, et c'est ce nom que bash utilise. Il va de soi qu'il ne faut pas qu'il y ait ambiguïté sur un nom partiel. Par exemple, si vous tapez la commande suivante :

```
cd /usr/l
```

et que vous demandiez au shell de compléter le nom, il ne pourra pas choisir quel répertoire utiliser entre `/usr/lib/` et `/usr/local/`. Dans ce cas, il émettra un petit bip signalant l'erreur. En appuyant une fois de plus sur la touche de tabulation, bash affiche la liste des choix possibles et vous propose de terminer la ligne de commande en saisissant des caractères supplémentaires afin de résoudre l'ambiguïté.

La deuxième solution est d'utiliser les caractères génériques du shell. Ces caractères permettent de désigner n'importe quel caractère, ou n'importe quelle séquence de caractères. Ils sont désignés respectivement par un point d'interrogation (« `?` ») et par un astérisque (« `*` »). Ainsi, si l'on tape la commande suivante :

```
cd /ho*
```

le shell ira directement dans le répertoire `/home/`, car le caractère générique « `*` » peut être remplacé par la séquence de caractères « `me` ». Il est également possible d'écrire :

```
cd /?ome
```

et dans ce cas le caractère générique « `?` » sera remplacé par « `h` ». Encore une fois, il ne faut pas qu'il y ait ambiguïté. Dans le cas contraire, le comportement varie selon le shell. En général, il essaie de résoudre l'ambiguïté au mieux en analysant la suite du chemin, et s'il ne peut pas, il affiche un message d'erreur.

Note : Ces caractères génériques sont interprétés directement par le shell et non par la commande qui les reçoit en paramètres. Tout nom de fichier contenant un caractère générique est remplacé par la liste des fichiers qui correspondent au motif donné. S'il n'existe qu'un seul fichier dans cette liste, il est possible d'utiliser les commandes comme `cd`, qui ne prennent qu'un seul paramètre. Mais il est possible d'utiliser les commandes acceptant plusieurs paramètres, même s'il y a plusieurs fichiers dans cette liste. Ainsi, la commande suivante :

```
ls *txt
```

permet de lister tous les fichiers dont le nom se termine par « `txt` ». Il ne peut évidemment pas y avoir d'ambiguïté dans ce cas.

Si on doit passer un paramètre comprenant l'un des caractères génériques interprétés par le shell à une commande particulière, on devra préfixer les caractères génériques d'un caractère d'échappement pour signaler au shell qu'il ne doit pas l'interpréter. Ce caractère d'échappement est la barre oblique inverse (« `\` »). Il est également possible de passer les paramètres entre guillemets « `"` », car le shell n'interprète pas les caractères génériques dans les chaînes de caractères. Par exemple, pour créer un répertoire?, on utilisera la commande suivante :

```
mkdir \?
```


Les noms de fichiers commençant par un tiret (caractère '-') posent également des problèmes avec la plupart des commandes, car ce caractère est utilisé pour spécifier des options. Ce n'est pas le shell qui interprète ce caractère dans ce cas, mais le problème est le même. La plupart des commandes utilisent l'option - (elle-même introduite par un tiret, ce qui fait donc deux tirets accolés) pour signaler que ce qui suit dans leur ligne de commande ne contient plus d'options, et ne doit donc plus être interprété. Il suffit donc de faire précéder le nom du fichier par deux tirets pour arrêter l'interprétation des options. Par exemple, pour afficher les informations relatives à un fichier nommé « -l », il faudrait utiliser la commande suivante :

```
ls -l -- -l
```

Enfin, sachez que le caractère '~' représente le répertoire personnel de l'utilisateur courant. Il est donc très facile d'accéder aux fichiers et aux répertoires de son répertoire personnel (situé, rappelons-le, dans le répertoire /home/). Par exemple, pour lister le contenu de son répertoire personnel, on utilisera la commande suivante :

```
ls -l ~
```

V-F - Opérations de base sur les fichiers

Vous aurez sans doute à afficher le contenu d'un fichier. Pour cela, la commande la plus appropriée est certainement la commande less :

```
less fichier
```

Cette commande affiche le contenu du fichier et vous permet de le faire défiler avec les flèches du curseur. Lorsque vous désirez terminer la visualisation, il suffit de taper la touche q (pour « quitter » less). Pour information, le nom de la commande less provient d'un trait d'humour sur une commande Unix plus classique, la commande more. Cette commande effectue à peu près le même travail que less, mais elle n'affiche le texte que page par page. Pour passer à la page suivante, il faut appuyer sur la barre d'espace. Quant à l'origine du nom de la commande more, c'est qu'elle affiche le mot « more » au bas de l'écran pour indiquer qu'il y a encore du texte à visualiser, et qu'il faut appuyer sur la barre d'espace pour lire la suite.

La commande less permet également d'effectuer une recherche dans le fichier en cours d'édition. Pour cela, il suffit de taper une commande de recherche de less. Cette commande commence par une barre oblique, suivie du texte à chercher. Par exemple, pour rechercher la chaîne de caractères « local » dans un fichier en cours de visualisation avec less, il suffit de taper :

```
/local
```

Lorsque vous voudrez rechercher l'occurrence suivante du motif de recherche, vous pourrez appuyer sur la touche n (pour « Next » en anglais). Pour rechercher l'occurrence précédente, il suffit de taper la touche N (en majuscule, cette fois).

Il est encore plus probable que vous aurez à éditer un fichier. Cette opération peut se faire relativement facilement grâce à un éditeur simplifié, vi. Cet éditeur n'est pas franchement ce qui se fait de plus convivial, cependant, il existe sur toutes les plates-formes Unix d'une part, et il est suffisamment léger pour pouvoir fonctionner sur un système minimal. Il est donc recommandé de savoir se servir de vi, ne serait-ce que dans le cas où votre système ne serait pas complètement fonctionnel. En clair, quand tout va mal, on peut compter sur vi ! vi sera décrit plus loin dans la [Section 5.8](#), car il dispose d'un grand nombre de commandes et il ne serait pas opportun de les décrire ici.

En général, la création d'un fichier se fait avec `vi`, bien que d'autres commandes puissent créer des fichiers. En revanche, pour supprimer un fichier, il n'existe qu'une seule commande :

```
rm chemin
```

où `chemin` est le chemin complet permettant d'accéder au fichier à supprimer. Il est possible de spécifier plusieurs fichiers à la commande `rm`. Dans ce cas, ils seront tous supprimés. `rm` est également capable de supprimer tous les fichiers d'un répertoire, ainsi que ses sous-répertoires. Dans ce cas, elle détruit toute une branche de l'arborescence du système de fichiers. Pour cela, il suffit d'utiliser l'option `-r` (pour « récursif ») avant le chemin du répertoire à supprimer.

Attention ! : La commande `rm` ne demande aucune confirmation avant de supprimer les fichiers ! D'autre part, les fichiers supprimés sont irrémédiablement perdus (il n'y a pas de commande « undelete » ou autre commande similaire). Vérifiez donc bien ce que vous avez tapé avant de valider une commande `rm` (surtout si vous êtes sous le compte `root`). Il peut être judicieux de forcer la commande `rm` à demander confirmation avant la suppression des fichiers, à l'aide de son option `-i`. On pourra pour cela définir un alias « `rm -i` » pour la commande `rm` dans le fichier d'initialisation du shell (c'est-à-dire le fichier `.bashrc` pour le shell `bash`). La notion d'alias sera décrite dans la [Section 5.9.9](#).

La copie d'un fichier se fait avec la commande `cp`, dont la syntaxe est donnée ci-dessous :

```
cp fichiers repertoire
```

où `fichiers` est la liste des fichiers à copier, et `repertoire` est le répertoire destination dans lequel ces fichiers doivent être copiés.

Enfin, le déplacement des fichiers se fait avec la commande `mv`, comme indiqué ci-dessous :

```
mv source destination
```

où `source` est le nom du fichier source et `destination` est le nom du répertoire destination. Notez que `mv` est une commande très puissante, puisqu'elle permet également de déplacer des répertoires et de renommer des fichiers et des répertoires. Pour renommer un fichier ou un répertoire, il suffit d'indiquer le nouveau nom de ce fichier ou de ce répertoire à la place de destination.

V-G - Autres commandes utiles

Pour terminer ce petit cours d'Unix, nous allons décrire quelques-unes des autres commandes d'Unix parmi les plus utiles. Elles sont utilisées moins souvent que les commandes vues précédemment, mais vous apprendrez certainement très vite à vous en servir, car elles sont très pratiques.

V-G-1 - Gestion des liens

La commande pour créer un lien est `ln`. Cette commande utilise la syntaxe suivante :

```
ln [-s] source lien
```

où `source` est le nom du fichier ou du répertoire source auquel le lien doit se référer, et `lien` est le nom du lien. L'option `-s` permet de créer un lien symbolique. Par défaut, ce sont des liens physiques qui sont créés. Rappelons qu'il est impossible de créer des liens physiques sur des répertoires.

Lorsqu'on liste des fichiers, on peut demander l'affichage d'informations complémentaires sur les liens. Pour cela, il suffit d'utiliser l'option `-l` de la commande `ls`. Ainsi, la commande suivante :

```
ls -l lien
```

permet d'afficher les informations sur le lien `lien`, et en particulier le fichier ou le répertoire cible de ce lien.

La suppression des liens se fait exactement comme celle d'un fichier. La destination n'est pas affectée en général, sauf si le lien est un lien physique et constitue la dernière référence au fichier pointé par le lien.

Les liens symboliques n'ont pas de droits d'accès ni de propriétaires, les informations de sécurité de la cible sont utilisées lorsqu'on accède au lien.

V-G-2 - Recherche de fichiers

Il vous sera sans doute nécessaire de rechercher des fichiers selon un critère donné dans toute une arborescence de répertoires. Pour cela, vous utiliserez la commande `find`. Cette commande est très puissante, mais dispose d'une syntaxe assez compliquée :

```
find repertoire -name nom -print
```

où `repertoire` est le répertoire à partir duquel la recherche doit commencer et `nom` est le nom du fichier à rechercher. Ce nom peut contenir des caractères génériques du shell, mais dans ce cas doit être placé entre guillemets afin d'éviter que ce dernier ne les interprète.

`find` accepte d'autres options de recherche que le nom (partie « `-name` » de la ligne de commande), et peut effectuer d'autres actions que l'affichage du chemin des fichiers trouvés (partie « `-print` »). Consultez les pages de manuel pour plus d'informations à ce sujet.

V-G-3 - Recherche d'un texte dans un fichier

La recherche d'une chaîne de caractères dans un ou plusieurs fichiers peut se faire à l'aide de la commande `grep`. Cette commande prend en premier paramètre le texte à rechercher, puis la liste des fichiers dans lequel ce texte doit être trouvé :

```
grep texte fichiers
```

Le texte peut être placé entre guillemets si nécessaire (en particulier, s'il contient des espaces ou des caractères interprétés par le shell, comme `*` et `?`). `grep` accepte un grand nombre d'options, qui ne seront pas décrites ici. Consulter les pages de manuel pour plus d'information à ce sujet.

V-G-4 - Remplacement de texte dans les fichiers

Le remplacement de texte dans un fichier peut être effectué de manière automatique, c'est-à-dire sans avoir à ouvrir le fichier dans un éditeur, grâce à la commande `sed` (abréviation de l'anglais « Stream Editor »). Cette commande est en fait un utilitaire de manipulation de flux de données, qui permet d'effectuer des traitements plus généraux que le simple remplacement de texte, mais c'est malgré tout pour cette opération qu'elle reste la plus utilisée.

`sed` peut travailler à la volée sur un flux de données textuelles, que ce flux provienne de l'entrée standard ou d'un fichier. Par défaut, il écrit le résultat de son travail sur le flux de sortie standard. Les opérations qu'il doit effectuer sur le flux de données peuvent être spécifiées de différentes manières, soit en fournissant un fichier script à l'aide de l'option `-f`, soit directement sur la ligne de commande, avec l'option `-e`. La syntaxe utilisée pour appeler `sed` est donc typiquement la suivante :

```
sed -e "commandes" fichier > resultat
```

ou :

```
sed -f script fichier > resultat
```

où `fichier` est le fichier sur lequel `sed` doit travailler, et `resultat` est le fichier devant recevoir le flux de données modifiées. Notez que cette commande utilise une redirection du flux de sortie standard dans un fichier. Ce type de redirection sera décrit en détail dans la [Section 5.9.2](#).

`sed` peut effectuer un grand nombre de commandes différentes et est réellement un outil très puissant. Cependant, nous ne verrons ici que la commande qui permet d'effectuer un remplacement de texte. Cette commande utilise la syntaxe suivante :

```
s/texte/remplacement/options
```

où `texte` est le texte à rechercher, `remplacement` est le texte de remplacement, et `options` est un jeu d'options exprimant la manière dont le remplacement doit être fait. Les options sont spécifiées à l'aide de simples caractères, les plus utiles étant sans doute `g`, qui permet d'effectuer un remplacement global (au lieu de ne remplacer que la première occurrence du texte rencontrée dans chaque ligne), et `l`, qui permet d'effectuer une recherche sans tenir compte de la casse des caractères.

Par exemple, la ligne de commande suivante :

```
sed -e "s/bonjour/bonsoir/g" test.txt > modif.txt
```

permet de remplacer toutes les occurrences de la chaîne de caractères « `bonjour` » par la chaîne de caractères « `bonsoir` » dans le texte du fichier `test.txt`, et d'enregistrer le résultat dans le fichier `modif.txt`.

Note : Il ne faut pas utiliser le même nom de fichier pour le fichier source et le fichier de résultat. En effet, `sed` lit le fichier source à la volée, et effectuer une redirection sur ce fichier pendant son traitement provoquerait la perte irrémédiable de son contenu. Pour résoudre ce problème, on pourra utiliser un nom de fichier temporaire, et écraser le fichier original par ce fichier une fois la commande `sed` exécutée.

V-G-5 - Compression et décompression des fichiers

Linux fournit un grand nombre de programmes de compression de fichiers. Le meilleur est sans doute `bzip2`, et le plus compatible sans doute `compress`. Cependant, le plus utilisé et le plus courant, surtout pour la distribution des sources, reste incontestablement `gzip`. Nous allons décrire brièvement comment compresser et décompresser des fichiers avec `gzip` et `bzip2` dans ce paragraphe.

La compression d'un fichier se fait de manière élémentaire :

```
gzip fichier
```

où `fichier` est le fichier à compresser. Après avoir effectué son travail, `gzip` renomme le fichier compressé en « `fichier.gz` ». La compression d'un fichier avec `bzip2` utilise exactement la même syntaxe, à ceci près qu'il faut remplacer `gzip` par `bzip2`. De plus, le nom du fichier compressé porte l'extension `.bz2` au lieu de `.gz`. Le fichier obtenu est donc nommé « `fichier.bz2` ».

La décompression d'un fichier se fait à l'aide de la commande suivante :

```
gunzip fichier.gz
```

ou

```
bunzip2 fichier.bz2
```

selon qu'il a été compressé avec gzip ou bzip2. Après décompression, l'extension complémentaire .gz ou .bz2 est supprimée du nom de fichier.

V-G-6 - Archivage de fichiers

L'archivage de fichiers se fait classiquement sous Unix avec le programme tar (abréviation de l'anglais « Tape ARchiver »). Ce programme permet simplement de regrouper tous les fichiers qu'il doit archiver dans un seul fichier structuré en blocs. Il a été initialement écrit pour permettre des archivages sur bandes ou sur tout autre périphérique de stockage de masse, mais il est également utilisé pour créer des fichiers archives contenant toute une arborescence.

La syntaxe de tar est très simple :

```
tar options archive [fichiers]
```

où **options** sont les options qui indiquent l'opération à effectuer et comment elle doit être réalisée, **archive** est le nom de l'archive qui doit être créée ou le nom du fichier de périphérique du périphérique d'archivage, et **fichiers** est la liste des fichiers à archiver.

Les options de tar que vous utiliserez le plus souvent sont les suivantes :

- **cvf** pour créer une archive ;
- **tvf** pour lister le contenu d'une archive ;
- **xvf** pour restaurer le contenu d'une archive.

Par exemple, pour archiver le contenu du répertoire courant dans le fichier **archive.tar**, vous utiliserez la ligne de commande suivante :

```
tar cvf archive.tar *
```

De plus, pour extraire le contenu de l'archive **archive.tar**, vous utiliserez la commande suivante :

```
tar xvf archive.tar
```

Note : L'option **z** permet d'effectuer une compression des données archivées ou une décompression des données restaurées à la volée. tar utilise gzip et gunzip pour la compression et la décompression. De même, l'option **j** permet de compresser l'archive à la volée avec bzip2.

Si l'on utilise un signe négatif ('-') à la place du nom de l'archive, tar enverra le résultat de la compression vers la sortie standard. Cela peut être utilisé pour des opérations avancées. Un exemple sera donné dans la [Section 5.9.2](#).

V-G-7 - Passage en mode superviseur

Si vous êtes prudent, vous avez sans doute créé un compte utilisateur juste après avoir installé votre système de base, et vous ne travaillez plus que dans ce compte. Cette technique est prudente, cependant, elle pose un problème évident : vous ne pouvez pas y faire votre travail d'administrateur. C'est pour cela que la commande **su** a été créée. Cette commande permet de changer son identité dans le système :

```
su [utilisateur]
```

où **utilisateur** est l'utilisateur dont on veut prendre l'identité. Par défaut, si aucun utilisateur n'est spécifié, le changement d'identité se fait vers l'utilisateur root. Bien entendu, il va de soi que la commande **su** demande le mot de passe avant d'obtempérer...

V-G-8 - Changement des droits des fichiers, du propriétaire et du groupe

Les mécanismes de droits d'accès ont déjà été décrits en détail ci-dessus dans le chapitre concernant les [notions générales sur Unix](#). Il peut être judicieux de relire ce chapitre afin de comprendre les effets des commandes présentées dans cette section.

V-G-8-a - Changement de propriétaire et de groupe

Le changement de propriétaire d'un fichier ne peut être réalisé que par l'administrateur du système. Cette opération se fait à l'aide de la commande suivante :

```
chown utilisateur fichier
```

où `utilisateur` est le nom de l'utilisateur qui doit devenir propriétaire du fichier, et `fichier` est le fichier devant changer de propriétaire.

Le changement de groupe peut être réalisé par n'importe quel utilisateur, mais on ne peut donner un fichier qu'à l'un des groupes dont on est membre. Cette opération se fait à l'aide de la commande suivante :

```
chgrp groupe fichier
```

où `groupe` est le nom du groupe qui doit être affecté au fichier, et `fichier` est le fichier devant changer de groupe. Bien entendu, l'administrateur peut affecter un fichier à n'importe quel groupe d'utilisateur.

V-G-8-b - Modification des droits Unix sur les fichiers

La commande permettant de changer les droits d'un fichier ou d'un répertoire est la suivante :

```
chmod droits fichier
```

où `fichier` est le fichier ou le répertoire dont on désire changer les droits, et `droits` est une chaîne de caractères permettant de spécifier les nouveaux droits. Cette chaîne commence par une lettre indiquant le groupe d'utilisateurs auquel le droit doit être appliqué, d'un caractère `+` ou `-` indiquant si le droit doit être ajouté ou supprimé, et d'une lettre indiquant le droit que l'on est en train de manipuler. La première lettre peut prendre les valeurs suivantes :

- `u` pour le champ « utilisateur », c'est-à-dire le propriétaire du fichier ;
- `g` pour le champ « groupe », c'est-à-dire tous les utilisateurs faisant partie du groupe du fichier ;
- `o` pour le champ « other », c'est-à-dire pour tous les utilisateurs qui ne sont ni propriétaires ni membres du groupe du fichier ;
- `a` pour tous les champs sans distinction, donc pour tous les utilisateurs.

Les droits sont identifiés par l'un des caractères suivants :

- `r` pour le droit de lecture ;
- `w` pour le droit d'écriture ;
- `x` pour le droit d'exécution ;
- `s` pour les bits `setuid` et `setgid` ;
- `t` pour le bit `sticky`.

Ainsi, la commande suivante :

```
chmod g+w exemple
```


permet de donner le droit d'écriture sur le fichier `exemple` à tous les membres du groupe auquel ce fichier appartient.

V-G-8-c - Utilisation des ACL

Si l'on veut donner des droits à un utilisateur ou un groupe particulier, on pourra définir une ACL (« Access Control List ») sur le fichier ou le répertoire, et affecter les droits unitairement. Ceci se fait simplement avec la commande `setfacl`, de la manière suivante :

```
setfacl -m ACL fichier
```

où `ACL` est l'ACL à affecter au fichier ou au répertoire `fichier`. Les ACLs sont constitués d'une liste d'entrées nommées des ACE (« Access Control Entries »). Les ACE sont séparées par des virgules et définissent chacune un droit. Chacun de ces droits doit être spécifié de manière complète, en précisant la classe d'utilisateur sur lequel il porte avec un mot-clé (`user` pour l'utilisateur propriétaire, `group` pour les utilisateurs du groupe propriétaire, ou `other` pour les autres utilisateurs), le nom de l'utilisateur ou du groupe si nécessaire, et les droits affectés à cet utilisateur ou à ce groupe. Tous ces paramètres doivent être séparés par deux points (caractère `:`). Par exemple, pour ajouter les droits d'écriture à l'utilisateur `alfred` sur le fichier `exemple`, vous pourrez utiliser la commande suivante :

```
setfacl -m user:alfred:w exemple
```

Si l'on ne spécifie aucun utilisateur avec la classe d'utilisateurs `user` dans une entrée, cette entrée se réfère automatiquement à l'utilisateur propriétaire du fichier ou du répertoire. De même, si l'on ne spécifie aucun groupe avec la classe d'utilisateurs `group`, l'entrée se réfère au groupe auquel le fichier appartient. Fixer ces entrées d'ACL sur un fichier avec ces deux syntaxes revient donc exactement à utiliser `chmod` avec les droits Unix classiques (à un détail près que l'on verra ci-dessous pour le groupe).

Les droits complets d'un fichier ou d'un répertoire peuvent être consultés avec la commande `getfacl`. Cette commande affiche en commentaire les informations sur l'objet auquel elle s'applique, à savoir son nom, son propriétaire et son groupe, à la suite d'un dièse (caractère `#`). Suivent toutes les ACE affectées à cet objet. Les droits Unix classiques sont lisibles directement avec les entrées `user::`, `group::` et `other::` respectivement pour l'utilisateur, les utilisateurs du groupe et les autres utilisateurs. Par exemple, si l'on affiche les ACL associées au fichier `exemple` après la commande précédente, on obtient ceci :

```
# file: exemple
# owner: batman
# group: users
user::rw-
user:alfred:-w-
group::r--
mask::rw-
other::r--
```

Dès qu'une ACL nominative a été attribuée à un fichier, une ACL spéciale est créée automatiquement. Il s'agit de l'ACL `mask`, qui, comme son nom l'indique, définit un masque de droits complémentaires pour tous les utilisateurs et les groupes ajoutés nominativement à l'ACL. Pour ces utilisateurs, les droits effectivement accordés sont leurs droits respectifs, restreints aux droits présents dans le masque. Le masque permet donc de restreindre les droits de tous les utilisateurs de la classe `group`, au sens large. Par exemple, si le masque contient les droits de lecture et d'écriture, et que l'utilisateur `alfred` dispose des droits de lecture et d'exécution sur le fichier, ses droits effectifs seront uniquement la lecture. Les droits effectifs sont indiqués en commentaire par `getfacl` pour les utilisateurs et les groupes s'ils ne sont pas égaux aux droits indiqués dans leurs entrées respectives.

Dès qu'il est défini, le masque remplace l'entrée du groupe du fichier pour les commandes classiques. Ainsi, dès qu'un masque est défini dans l'ACL d'un fichier, les changements de droits sur le groupe effectués avec la commande `chmod` ne modifient plus que le champ de masque. Cela peut surprendre dans certaines situations. Par exemple, si l'entrée de l'ACL décrivant les droits du groupe du fichier ne donne aucun droit, les utilisateurs de ce groupe n'auront toujours aucun droit même après un `chmod g+rw` sur le fichier. En effet, cette dernière commande ne modifie que le

masque. Il est donc nécessaire d'ajouter les droits sur le groupe du fichier explicitement avec setfacl, de la manière suivante :

```
chmod g+rw exemple  
setfacl -m group::rw exemple
```

La première commande modifie les droits sur le masque, et la deuxième ajoute les droits pour tous les utilisateurs du groupe.

Notez bien que le masque ne s'applique pas pour la détermination des droits du propriétaire et des utilisateurs de la classe other. Par ailleurs, lorsque l'on affiche les droits étendus d'un fichier avec l'option -l de la commande ls, les droits affichés pour le groupe du fichier sont les droits définis dans le masque lui-même (même si aucune entrée d'ACL ne donne complètement ces droits à un utilisateur ou à un groupe). Cela permet donc de voir directement les droits les plus forts qui pourraient être attribués sur ce fichier, ce qui est cohérent avec ce qu'attendent les outils Unix classiques.

Il est possible de supprimer toutes les entrées de l'ACL d'un fichier avec l'option -b de setfacl. Pour supprimer une entrée spécifique de l'ACL, vous devrez utiliser l'option -x. Cette dernière option ne permet pas de supprimer les entrées génériques pour le propriétaire et le groupe du fichier, ni celles des autres utilisateurs. Par exemple, pour retirer toutes les entrées définies précédemment sur le fichier `exemple`, on utilisera la commande suivante :

```
setfacl -b exemple
```

Vous pourrez toutefois constater que getfacl continue d'afficher les entrées génériques « user:: », « group:: » et « other:: ».

Les répertoires disposent, en plus de leur ACL normale, d'une ACL par défaut. L'ACL par défaut d'un répertoire est celle qui est appliquée à tout nouveau fichier ou répertoire créés dans ce répertoire. De plus, les sous-répertoires héritent de l'ACL par défaut de leur parent. Les ACLs par défaut sont modifiables avec l'option -d de setfacl et sont affichées avec le préfixe `default:` par getfacl.

Note : Notez bien que de nombreux outils Unix classiques ne gèrent pas correctement la notion d'ACL (en particulier les gestionnaires de fichiers graphiques). La copie ou la modification d'un fichier peut donc provoquer la perte de son ACL, modifiant ainsi les droits des utilisateurs sur ce fichier. Les ACLs doivent donc être utilisées avec parcimonie et leur manipulation entourée du plus grand soin.

V-H - vi, l'éditeur de fichiers de base

Vous serez obligé, lorsque vous effectuerez la configuration de votre système, d'éditer les fichiers de configuration (classiquement, ces fichiers sont placés dans le répertoire `/etc/`). Ces modifications peuvent être réalisées avec n'importe quel éditeur a priori, et il est même conseillé d'utiliser votre éditeur favori. Cependant, il faut savoir se servir de vi, parce que c'est le seul éditeur qui sera toujours installé, et qui fonctionnera en toutes circonstances. Le prix à payer pour cette fiabilité est un nombre restreint de fonctionnalités. En fait, vi est très puissant, mais il ne s'embarrasse pas de superflu, ce qui en fait certainement l'éditeur le moins convivial du monde. Ce paragraphe vous donnera la liste des principales commandes de vi. Cette liste ne sera toutefois pas exhaustive, car vous n'utiliserez certainement pas vi dans la vie courante.

Pour éditer un fichier avec vi, il suffit de passer le nom de ce fichier en ligne de commande :

```
vi fichier
```

Il est possible de passer plusieurs fichiers dans la ligne de commande, et vi les éditera les uns après les autres. Cependant, il faut savoir que vi ne permet de travailler que sur deux fichiers à la fois, et qu'il n'est pas facile de passer de l'un à l'autre. Par conséquent, il est conseillé de n'éditer qu'un seul fichier à la fois.

vi est un éditeur qui fonctionne dans plusieurs modes différents : le mode d'édition, dans lequel le texte peut être modifié, le mode de commande, dans lequel des commandes particulières peuvent être données, et le mode de visualisation, dans lequel le fichier ne peut être que visualisé. Par défaut, vi est en mode de visualisation, et il faut utiliser une commande d'édition pour passer en mode d'édition. Quand on est en mode d'édition, on peut revenir au mode de visualisation en appuyant sur la touche Echap (ou Esc, selon votre clavier). Cette touche a aussi une signification dans le mode de commande : elle annule la saisie de la commande en cours. Par conséquent, lorsqu'on est perdu et que l'on ne sait plus dans quel mode on se trouve (ce qui arrive fatalement à un moment donné), il suffit d'appuyer sur cette touche. On sait alors qu'on se trouve en mode de visualisation.

Le déplacement du curseur en mode de visualisation se fait avec les touches du curseur. Cependant, si votre clavier n'est pas bien configuré, ces touches peuvent ne pas fonctionner. C'est pour cette raison que vi fournit un jeu de touches alternatif :

- la touche **h** permet de déplacer le curseur vers la gauche ;
- la touche **l** permet de déplacer le curseur vers la droite ;
- la touche **j** permet de déplacer le curseur vers le bas ;
- la touche **k** permet de déplacer le curseur vers le haut.

Le curseur est bien entendu déplacé automatiquement lors de la saisie du texte en mode d'édition.

Le passage en mode d'édition peut se faire avec l'une des commandes suivantes :

- la touche **i** permet de passer en mode d'insertion (le texte saisi s'insère avant le caractère sur lequel le curseur est positionné) ;
- la touche **a** permet de passer en mode d'ajout de caractères (le texte saisi s'insère après le caractère sur lequel le curseur est positionné) ;
- la touche **A** permet de placer le curseur en fin de ligne et de passer en mode d'ajout de caractères ;
- la touche **o** permet de créer une nouvelle ligne après la ligne où se trouve le curseur et de passer en mode d'édition sur cette nouvelle ligne ;
- la touche **O** permet de créer une nouvelle ligne avant la ligne où se trouve le curseur et de passer en mode d'édition sur cette nouvelle ligne.

La création d'une nouvelle ligne peut donc être faite avec les commandes **o** et **O**, mais il est possible de couper une ligne en deux, ou de passer à la ligne simplement en tapant sur la touche Entrée en mode d'édition. Inversement, la commande **J** permet de supprimer un saut de ligne en fin de ligne et de placer ainsi le texte de la ligne suivante à la suite du texte de la ligne courante.

La suppression d'un caractère se fait avec la touche Suppr (ou Del, selon le clavier) ou la touche de retour arrière (dite touche Backspace). Cependant, encore une fois, vi fournit un jeu de touches alternatif permettant de travailler avec un clavier mal configuré :

- la commande **x** permet d'effacer le caractère situé sous le curseur ;
- la commande **dd** permet d'effacer la ligne où se trouve le curseur ;
- la commande **dw** permet d'effacer le mot où se trouve le curseur.

Le texte qui a été supprimé est placé dans ce que l'on appelle un buffer. Le contenu du buffer peut être inséré à n'importe quel endroit du fichier grâce à la commande **p**. Ainsi, il est possible de faire un couper/coller en effaçant la ligne désirée et en appuyant sur la touche **p** à l'emplacement destination.

La commande **u** permet d'annuler la dernière opération effectuée, et la commande **U** permet de la réexécuter.

La commande **yy** permet de copier la ligne courante dans le buffer. Cette commande est donc utilisée pour effectuer des copier/coller, en combinaison avec la commande **p**.

Les commandes de vi peuvent être répétées un certain nombre de fois, en spécifiant ce nombre avant de les écrire. Ainsi, pour supprimer 3 lignes, il suffira de taper la commande suivante :

3dd

Dans ce cas, ces trois lignes sont également placées dans le buffer. La même technique peut être utilisée pour copier/coller plusieurs lignes en une seule opération.

Enfin, vi accepte un certain nombre de commandes générales lorsqu'il est en mode de commande. Ce mode est activé dès que l'on appuie sur la touche deux points (':') dans le mode de visualisation. Les commandes générales les plus utiles sont décrites ci-dessous :

- la commande `:q` permet de quitter vi. Si le fichier en cours d'édition a été modifié, vi refusera de se terminer sans l'enregistrer. Si l'on veut malgré tout sortir sans l'enregistrer, il faudra utiliser la commande `:q!` ;
- la commande `:w` permet d'enregistrer le fichier courant. Pour enregistrer ce fichier et quitter vi, la commande `:wq` peut être utilisée ;
- la commande `:help sujet` permet d'obtenir de l'aide sur le sujet « sujet » ;
- la commande `!:commande` permet d'exécuter la commande du shell « commande ». Cela peut être pratique pour effectuer une opération dans le shell sans avoir à quitter vi. Cela dit, il sera sans doute plus efficace d'utiliser un autre terminal virtuel.

Comme vous l'avez constaté, vi est réellement une horreur à utiliser. Malgré tout, il permet de faire tout ce dont on a besoin pour éditer un fichier. Il dispose même de puissantes fonctionnalités que même les traitements de texte évolués ne sont pas capables de faire. Elles ne seront cependant pas décrites ici, car cela dépasserait le cadre de la simple installation de Linux. Vous pourrez toujours consulter l'aide de vi pour de plus amples informations.

V-I - Utilisation du shell bash

Le shell est l'environnement utilisateur en mode texte sous Linux. C'est le programme qui se charge de lire et d'exécuter les commandes que l'utilisateur saisit. Classiquement, le shell est utilisé de manière interactive, c'est-à-dire que l'utilisateur dialogue avec le système par l'intermédiaire du shell. Il saisit les commandes, et le shell les exécute et affiche les résultats. Le shell le plus couramment utilisé sous Linux est sans aucun doute bash. En tout cas, c'est le shell par défaut que la plupart des distributions utilisent. Il est donc conseillé de connaître un petit peu ce que ce shell est capable de réaliser, et comment. Le shell bash est une évolution du shell sh, utilisé par quasiment tous les systèmes Unix. Son nom provient de l'abréviation de l'anglais « Bourne Again SHell », ce qui signifie qu'il s'agit effectivement d'une nouvelle variante du shell sh.

Au temps des interfaces graphiques complexes et sophistiquées, il peut paraître archaïque de vouloir encore utiliser des lignes de commandes pour utiliser un ordinateur. C'est en partie vrai, mais il faut savoir que les shells Unix sont extrêmement puissants et que les interfaces graphiques ne permettent toujours pas, même à l'heure actuelle, de réaliser toutes les tâches faisables avec un shell. D'autre part, il est souvent plus efficace de taper une simple commande dans un shell que de rechercher un outil graphique et de parcourir les divers menus, puis de choisir les options de la commande désirée avant de valider. Des ergonomes ont démontré, et des graphistes du monde entier le confirmeront, que la souris n'est pas le périphérique d'entrée le plus précis et le plus facile à utiliser pour manipuler les objets de l'environnement utilisateur. La plupart des programmeurs utilisent encore bon nombre de ce qu'on appelle des « raccourcis clavier » pour exécuter des commandes, même dans les environnements utilisateurs graphiques.

Quoi qu'il en soit, le shell est bien plus qu'un interpréteur de commande. Il s'agit réellement d'un environnement de programmation, permettant de définir des variables, des fonctions, des instructions complexes et des programmes complets, que l'on appelle des scripts shell. Les sections suivantes ont pour objectif de vous montrer les principales caractéristiques du shell, sans pour autant prétendre vous apprendre la programmation des scripts shell. La lecture de cette section pourra donc être différée dans un premier temps. Toutefois, elle pourra être bénéfique à ceux qui désirent comprendre les scripts de configuration utilisés par leur distribution, ou tout simplement à ceux qui sont curieux de nature.

V-I-1 - Contrôle des processus

Un des avantages des lignes de commandes par rapport aux environnements graphiques est la facilité avec laquelle elles permettent de contrôler les processus. Ce paragraphe décrit les principales méthodes pour lancer et arrêter un processus, ainsi que pour lui fournir les données sur lesquelles il doit travailler et récupérer ses résultats.

V-I-1-a - Lancement d'un programme en arrière-plan

Le lancement normal d'un programme se fait en tapant sa ligne de commande et en appuyant sur la touche de validation. Le shell ne rendra pas la main et ne permettra pas de lancer un autre programme tant que le processus en cours ne sera pas terminé. Cependant, vous pouvez fort bien désirer lancer en arrière-plan une commande dont la durée d'exécution peut être très longue et continuer à travailler. Après tout, Linux est multitâche... Eh bien, rien de plus facile !

Pour lancer une commande en arrière-plan, il faut :

- s'assurer que la commande aura toutes les informations nécessaires pour travailler sans intervention de l'utilisateur (ou, autrement dit, que la commande n'est pas interactive) ;
- ajouter une esperluette (caractère « & ») à la fin de la ligne de commande.

Par exemple, la commande suivante :

```
cp /cdrom/kernel/linux-2.6.6.tar.gz . &
```

copiera l'archive du noyau 2.6.6 du CD-ROM vers le répertoire courant, et s'exécutera en arrière-plan.

Lorsqu'une commande est lancée en arrière-plan, le shell affiche deux nombres qui permettront de l'identifier par la suite. Le premier nombre, indiqué entre crochets, est le numéro de « job » du shell. Ce numéro sert à identifier les commandes du shell de manière unique. Un job est donc en réalité une commande du shell, simple ou complexe. Le deuxième numéro est le numéro de processus (« PID », pour « Process IDentifier ») dans le système du processus maître du job. Le PID est un numéro unique dans le système, qui permet d'identifier de manière unique les processus en cours. Ces deux nombres permettront de manipuler les processus, avec les commandes que l'on verra plus tard.

Il ne faut pas confondre les numéros de job avec les numéros de processus. Premièrement, un numéro de job n'est unique que dans un shell donné, et n'a aucune signification au niveau du système complet, alors que le numéro de processus est attribué par le système à chaque programme en cours d'exécution. Ensuite, une même commande du shell peut lancer plusieurs processus conjointement. Dans ce cas, il y a bien évidemment plusieurs numéros de processus, mais un seul et unique job. Ce genre de situation se produit par exemple lors de l'utilisation d'une redirection du flux de sortie standard d'un processus vers le flux d'entrée standard d'un autre processus.

Le numéro de processus affiché par le shell lors du lancement d'une ligne de commande complexe représente le PID du processus maître de la commande, c'est-à-dire, en pratique, le dernier processus d'une série de redirections ou le processus du shell exécutant les commandes complexes. Cela signifie que dans tous les cas de configuration, ce PID est celui du processus qui contrôle l'ensemble des opérations effectuées par la ligne de commande. C'est donc par ce processus que l'on peut manipuler la commande complète, par exemple pour l'interrompre.

Il est possible de retrouver le PID du processus maître d'une commande à partir du numéro de job correspondant du shell. Cela se fait simplement, en utilisant l'expression suivante :

```
%job
```

où job est le numéro du job dont on cherche le PID. Ainsi, dans toutes les commandes décrites ci-dessous, le PID peut être utilisé directement, ou être remplacé par le numéro du job préfixé du caractère de pourcentage.

V-I-1-b - Listing des processus

Il n'est pas nécessaire de retenir tous les numéros de jobs et tous les PID des processus en cours d'exécution. Il existe en effet des commandes permettant d'obtenir la liste des processus et des jobs. La plus simple à utiliser est bien évidemment la commande du shell pour obtenir la liste des jobs avec leurs lignes de commandes. Pour obtenir cette liste, il suffit de taper la commande suivante :

```
jobs
```

qui affiche, dans l'ordre, le numéro de job, l'état du processus correspondant, et la ligne de commande.

Une autre commande, plus bas niveau, permet d'obtenir des informations plus complètes directement à partir du système. Il s'agit de la commande `ps`, dont la syntaxe est donnée ci-dessous :

```
ps [options]
```

Les options les plus utiles sont sans doute `x`, qui permet de demander l'affichage de toutes les commandes en cours d'exécution et non pas seulement les processus en cours d'exécution dans le shell où la commande `ps` est exécutée, et `a`, qui permet d'obtenir l'affichage de toutes les commandes, pour tous les utilisateurs connectés. Ces deux options peuvent être cumulées, et la commande suivante :

```
ps ax
```

affiche donc toutes les commandes en cours d'exécution sur le système.

Les informations les plus intéressantes affichées par `ps` sont le PID du processus, qui est donné par le premier nombre affiché, et la ligne de commande, qui est la dernière information affichée. Pour plus de détails sur la commande `ps`, veuillez consulter la page de manuel correspondante.

V-I-1-c - Notion de signal

Dans un système Unix, tous les processus peuvent recevoir des messages, envoyés soit par l'utilisateur, soit par un autre processus, soit par le système. Ces messages sont appelés signaux. La plupart des signaux sont envoyés par le système pour indiquer au processus qu'il a fait une faute et qu'il va être terminé. Cependant, ce n'est pas toujours le cas : certains signaux sont envoyés uniquement dans le cadre de la communication entre les processus, et certains autres ne peuvent même pas être captés par le processus et sont traités directement par le système. Nous n'entrerons pas en détail dans la gestion des signaux ici, car cela nous emmènerait trop loin. Cependant, la manière d'envoyer un signal à un processus à partir du shell sera décrite.

L'envoi d'un signal se fait avec la commande `kill`, avec la syntaxe suivante :

```
kill [-signal] PID
```

où `signal` est une option qui permet de préciser le signal qui doit être envoyé, et `PID` est le numéro du processus qui doit le recevoir. Les numéros de signaux les plus importants sont décrits dans le tableau ci-dessous :

Tableau 5-2. Principaux signaux Unix

Numéro de signal	Signification
15	Signal de terminaison de processus.
9	Signal de destruction inconditionnelle de processus.
19	Signal de suspension de processus.
18	Signal de reprise d'exécution d'un processus suspendu.

Lorsqu'aucun signal n'est spécifié, le signal 15 de terminaison est utilisé par défaut. Ce signal demande au processus en cours d'exécution de se terminer immédiatement. Il peut être capté par le processus, pour lui donner une chance d'enregistrer les données sur lesquelles il travaillait et de libérer les ressources qu'il utilisait. Pour certains processus, cela ne fonctionne pas, et il faut utiliser le signal de destruction du processus à l'aide de la commande suivante :

```
kill -9 PID
```

Attention cependant à cette commande : le processus est immédiatement détruit, sans autre forme de procès. Il peut donc s'ensuivre une perte de données, n'en abusez donc pas trop.

V-I-1-d - Arrêt d'un processus

Tout processus lancé en ligne de commande peut être arrêté immédiatement sous Linux. Pour cela, deux méthodes sont disponibles. La première consiste à taper la combinaison de touches CTRL+C lorsque le processus est en cours d'exécution interactive (c'est-à-dire lorsqu'il n'a pas été lancé en arrière-plan). S'il a été lancé en arrière-plan, on peut soit le ramener en avant-plan (avec la commande fg, que l'on verra plus loin) avant d'utiliser CTRL+C, soit lui envoyer le signal de terminaison à l'aide de la commande kill vue précédemment. Dans le cas d'une ligne de commande, le signal de terminaison est transmis au processus maître de la ligne de commande, et est ensuite propagé à l'ensemble des processus fils de ce processus. Cela signifie que tous les processus invoqués dans le cadre de cette commande sont également arrêtés.

V-I-1-e - Gel d'un processus

Il est possible de « geler » un processus en cours d'exécution, c'est-à-dire de le suspendre, sans pour autant l'arrêter définitivement. Cela peut être utilisé pour libérer un peu les capacités de calcul, lorsque ce processus consomme trop de ressources par exemple. Pour cela, deux méthodes sont possibles :

- soit on utilise la combinaison de touches CTRL+Z, lorsque le processus est en avant-plan ;
- soit on envoie le signal 19 au processus (signal « STOP ») à l'aide de la commande kill.

La première méthode est recommandée pour les processus lancés par une ligne de commande complexe, car le signal STOP est envoyé au processus maître de la commande, et est propagé à l'ensemble des processus fils de ce processus. Cela signifie que tous les processus invoqués dans le cadre de cette commande sont également gelés. La deuxième méthode est plus bas niveau, et permet de geler n'importe quel processus que l'on a lancé.

V-I-1-f - Relancement d'un processus

Un processus suspendu peut être relancé soit en avant-plan, soit en arrière-plan. Pour relancer un processus en avant-plan, il faut utiliser la commande suivante :

```
fg [PID]
```

où PID est le PID du processus à relancer en avant-plan. Si ce paramètre n'est pas spécifié, le dernier processus stoppé sera relancé en arrière-plan. fg est l'abréviation de l'anglais « foreground », ce qui signifie « avant-plan ». Il faut attendre que ce processus se termine pour entrer de nouvelles commandes. Par conséquent, on ne peut lancer en avant-plan qu'un seul processus.

De même, pour lancer un processus en arrière-plan, il faut utiliser la commande bg, qui est l'abréviation de l'anglais « background ». Cette commande s'utilise de la même manière que la commande fg :

```
bg [PID]
```

Le relancement d'un processus suspendu peut également se faire en lui envoyant le signal 18 à l'aide de la commande kill.

V-I-2 - Redirections

Pour pouvoir lancer un programme en arrière-plan, il est nécessaire qu'il n'ait pas besoin de demander des données à l'utilisateur. En effet, lorsqu'il est en arrière-plan, la saisie de ces données ne peut pas se faire, puisque le shell les interpréterait comme une nouvelle commande. De plus, tout affichage en provenance d'une commande en arrière-plan apparaît sur la console tel quel, et risque de se mélanger avec l'affichage des autres programmes ou même avec la commande en cours d'édition. C'est pour résoudre ces problèmes que le mécanisme des redirections a été introduit.

V-I-2-a - Principe de base

Le mécanisme des redirections a pour but de transférer les données provenant d'un flux vers les données d'un autre flux. Il se base sur la notion de descripteur de fichier, utilisée par la plupart des systèmes Unix. Un descripteur de fichier est un numéro utilisé par les programmes pour identifier les fichiers ouverts. Les descripteurs 0, 1 et 2 sont respectivement affectés d'office au flux d'entrée standard (nommé « stdin »), au flux de sortie standard (« stdout ») et au flux d'erreur standard (« stderr »), qui en général apparaît également sur l'écran.

Les descripteurs de fichiers d'un processus sont généralement hérités par tous ses processus fils. Cela signifie que, lors de leur lancement, ces processus peuvent utiliser tous les descripteurs de fichiers mis à leur disposition par leur père. Dans le cas des lignes de commande, les processus fils sont les processus lancés par le shell, et les descripteurs de fichiers hérités sont donc les descripteurs de fichiers du shell. C'est de cette manière que le shell peut manipuler les descripteurs de fichiers des processus qu'il lance : il effectue d'abord les redirections sur ses propres descripteurs de fichiers, puis il lance le processus fils avec ces redirections actives. Le mécanisme est donc complètement transparent pour les processus fils.

Le mécanisme des redirections permet en fait d'injecter dans un descripteur de fichier des données provenant d'un autre descripteur ou d'un fichier identifié par son nom, et d'envoyer les données provenant d'un descripteur de fichier dans un autre descripteur ou dans un fichier identifié par son nom. Si l'on utilise les descripteurs de fichiers des flux d'entrée / sortie standards, on peut exécuter n'importe quelle commande interactive en arrière-plan.

V-I-2-b - Redirections de données en entrée

Pour injecter des données provenant d'un fichier dans le descripteur de fichier `n` d'un processus, il suffit d'ajouter la ligne suivante à la fin de la commande permettant de lancer ce processus :

```
n<fichier
```

où `fichier` est le nom du fichier dont les données doivent être injectées dans le descripteur `n`. Dans cette syntaxe, le descripteur peut ne pas être précisé. Dans ce cas, le shell utilisera le descripteur 0, et les données du fichier seront donc envoyées dans le flux d'entrée standard du processus. Par exemple, supposons que l'on désire utiliser une commande nommée « `search` », et que cette commande demande un certain nombre d'informations lors de son exécution. Si l'on sait à l'avance les réponses aux questions qui vont être posées, on peut créer un fichier de réponse (nommé par exemple « `answer.txt` ») et alimenter la commande « `search` » avec ce fichier. Pour cela, on utilisera la ligne de commande suivante :

```
search < answer.txt
```

Il est également possible d'injecter des données provenant d'un autre descripteur de fichier dans un descripteur de fichier. On utilisera pour cela la syntaxe suivante :

```
n<&s
```

où `n` est toujours le descripteur de fichier du processus à exécuter dans lequel les données doivent être injectées, et `s` est un descripteur de fichier contenant les données sources à injecter. Par défaut, si `n` n'est pas précisé, le flux d'entrée standard du processus sera utilisé.

V-I-2-c - Redirection de données en sortie

Inversement, il est possible d'enregistrer les données écrites par un processus dans un de ses descripteurs de fichier dans un fichier. Pour cela, on utilisera l'opérateur `>` avec la syntaxe suivante :

```
n>fichier
```

où `n` est le numéro du descripteur de fichier du processus à enregistrer, et `fichier` est le nom du fichier dans lequel les données doivent être stockées. Par défaut, si `n` n'est pas spécifié, le descripteur du flux de sortie standard sera utilisé (descripteur 1). Par exemple, si la commande précédente affiche des résultats et que l'on désire les stocker dans le fichier « `result.txt` », on utilisera la ligne de commande suivante :

```
search < answer.txt >result.txt
```

Notez que cette commande détruira systématiquement le contenu du fichier « `result.txt` » et le remplacera par les informations provenant du flux de sortie standard du processus « `search` ». Il est possible de ne pas vider le fichier « `result.txt` » et d'ajouter les informations en fin de fichier, en utilisant l'opérateur `>>` à la place de l'opérateur `>`. Ainsi, la commande suivante :

```
search < answer.txt >>result.txt
```

aura pour effet d'ajouter à la fin du fichier « `result.txt` » les informations affichées par le processus « `search` ».

Le flux d'erreur standard, qui correspond normalement à l'écran et qui permet d'afficher les messages d'erreur, peut être redirigé avec l'opérateur `2>`, de la même manière que l'opérateur `>` est utilisé pour le flux de sortie standard (puisque c'est le descripteur de fichier utilisé par défaut par l'opérateur `>`). Par exemple, si l'on veut envoyer les messages d'erreurs éventuels de la commande précédente vers le périphérique nul (c'est-à-dire le périphérique qui n'en fait rien) pour ignorer ces messages, on utilisera la ligne de commande suivante :

```
search <answer.txt >result.txt 2> /dev/null
```

Une telle ligne de commande est complètement autonome, et peut être lancée en arrière-plan, sans aucune intervention de l'utilisateur :

```
search <answer.txt >result.txt 2> /dev/null &
```

Il est également possible d'effectuer une redirection des données provenant d'un descripteur de fichier du processus vers un autre descripteur de fichier de ce processus. On utilisera pour cela la syntaxe suivante :

```
n>&d
```

où `n` est le descripteur de fichier dont les données doivent être redirigées, et `d` le descripteur de fichier destination. Cette syntaxe est souvent utilisée pour rediriger le flux d'erreur standard vers le flux d'entrée standard, lorsqu'on veut récupérer les erreurs et les messages d'exécution normale dans un même fichier. Par exemple, si l'on veut rediriger le flux de sortie et le flux d'erreurs de la commande « `search` » dans un même fichier, on utilisera la ligne de commande suivante :

```
search <answer.txt >result.txt 2>&1
```

Cette ligne de commande utilise deux redirections successives pour les données affichées par la commande « `search` » : la première redirige le flux de sortie standard vers un fichier, et la deuxième le flux d'erreur standard vers le flux

de sortie standard. Notez que l'ordre des redirections est important. Elles sont appliquées de gauche à droite. Ainsi, dans la commande précédente, le flux de sortie standard est redirigé vers le fichier « result.txt », puis le flux d'erreur standard est injecté dans le flux de sortie standard ainsi redirigé.

Note : Il est également possible d'utiliser un autre descripteur de fichier que les descripteurs des flux standards. Cependant, il est nécessaire, dans ce cas, d'ouvrir ce descripteur dans le shell avant de lancer la commande. Cela peut se faire à l'aide de la syntaxe suivante :

```
n<>fichier
```

où n est un numéro de descripteur de fichier non encore utilisé, et fichier est un nom de fichier. Ce nouveau descripteur de fichier pourra être utilisé dans les commandes précédentes, afin de faire manipuler le fichier fichier par les processus fils de manière transparente.

Les descripteurs de fichiers ouverts de cette manière le restent d'une commande sur l'autre dans le shell. Cela implique que toutes les données écrites dans ces descripteurs de fichiers sont ajoutées automatiquement à la fin des fichiers manipulés par ces descripteurs. Ce comportement est différent de la redirection vers un fichier effectuée par l'opérateur '>', qui ouvre à chaque fois le fichier en écriture à son début et qui supprime donc toutes les données déjà existantes. Il n'y a donc pas d'opérateur '>>&' pour ajouter des données à un descripteur de fichier, car cela n'a pas de sens.

Les descripteurs de fichiers peuvent également être manipulés directement, par l'intermédiaire de fichiers virtuels du répertoire /dev/fd/. À chaque descripteur de fichier (y compris les descripteurs pour les flux d'entrée/sortie standards !) y correspond un fichier dont le nom est le numéro du descripteur. Par exemple, le fichier /dev/fd/2 correspond au flux d'erreur standard.

En fait, le répertoire /dev/fd/ est un lien symbolique vers le répertoire /proc/self/fd/ du système de fichiers virtuel /proc/. Ce système de fichiers est géré par le noyau directement, et permet d'accéder aux informations sur le système et les processus. Il contient en particulier un sous-répertoire portant le nom du PID de chaque processus existant dans le système, et chacun de ces répertoires contient lui-même un sous-répertoire fd/ où sont représentés les descripteurs de fichiers ouverts par le processus correspondant. Le système de fichiers /proc/ contient également un lien symbolique self/ pointant sur le sous-répertoire du processus qui cherche à l'ouvrir. Ainsi, /proc/self/fd/ est un chemin permettant à chaque processus d'accéder à ses propres descripteurs de fichiers.

En pratique, la manipulation directe des descripteurs de fichiers n'est réellement intéressante que pour les flux standards, dont les numéros de descripteurs sont fixes et connus de tous les programmes. Pour les autres descripteurs, cette technique est souvent inutilisable ou inutile, sauf lorsqu'on utilise des programmes sachant manipuler des descripteurs de numéros bien déterminés.

V-I-2-d - Insertion de documents

Il existe un dernier opérateur de redirection, qui n'est utilisé en pratique que dans les scripts shell. Cet opérateur permet d'insérer directement un texte complet dans le flux d'entrée standard, sans avoir à placer ce document dans un fichier à part. Cette technique permet donc de stocker des données avec le code des scripts shell, et de n'avoir ainsi qu'un seul fichier contenant à la fois le script et ses données.

Cet opérateur est l'opérateur '<<', il s'utilise selon la syntaxe suivante :

```
<<EOF
texte
.
.
.
EOF
```

où `texte` est le contenu du texte à insérer, et `EOF` est un marqueur quelconque qui sera utilisé seul sur une ligne afin de signaler la fin du texte.

Par exemple, il est possible de créer un fichier `test.txt` de la manière suivante :

```
cat <<fin >test.txt
Ceci est un fichier texte saisi directement dans le shell.
On peut écrire tout ce que l'on veut, et utiliser les fonctions d'éditations
de ligne du shell si l'on veut.
Pour terminer le fichier, il faut taper le mot "fin" tout seul, au début
d'une ligne vide.
fin
```

V-I-3 - Les tubes

Les redirections sont très pratiques lorsqu'il s'agit d'injecter un fichier dans le flux d'entrée standard d'un processus, ou inversement de rediriger le flux standard d'une commande vers un fichier, mais elles ont justement le défaut de devoir utiliser des fichiers. Il est des situations où l'on désirerait injecter le résultat d'une commande dans le flux d'entrée standard d'une autre commande, sans passer par un fichier intermédiaire. Cela est heureusement réalisable, grâce à ce que l'on appelle les « tubes ».

V-I-3-a - Syntaxe des tubes

Pour rediriger le résultat d'une commande dans le flux d'entrée d'une autre commande, il faut utiliser l'opérateur `|`. Cet opérateur représente un tuyau canalisant les données issues d'une commande vers le flux d'entrée standard de la commande suivante, d'où le nom de « pipe » en anglais (ce qui signifie « tuyau » ou « tube »). L'opérateur tube s'utilise de la manière suivante :

- on écrit la première commande, qui doit fournir les données à la deuxième commande ;
- on écrit l'opérateur tube ;
- on écrit la deuxième commande, qui doit lire les données provenant de la première.

La commande se trouvant à la gauche de l'opérateur tube doit être complète, avec ses autres redirections éventuelles. La redirection dans un tube s'effectue après les autres types de redirections vues précédemment.

Le système contrôle l'exécution des processus qui se trouvent aux deux bouts d'un tube, de telle sorte que le transfert de données puisse toujours se faire. Si le processus source a trop de données, il est figé par le système d'exploitation en attendant que le processus consommateur ait fini de traiter les données déjà présentes. Inversement, si le processus source est trop lent, c'est le processus consommateur qui attendra patiemment que les données soient disponibles.

Les tubes sont utilisés très couramment, ne serait-ce que pour afficher page par page le contenu d'un répertoire. La commande suivante effectue un tel travail :

```
ls | less
```

Ici, le résultat de la commande `ls` est redirigé vers la commande `less`, qui permet d'afficher page par page (et de revenir en arrière dans ces pages) la liste des fichiers du répertoire courant.

Note : ATTENTION ! Il se peut que vous obteniez parfois un message d'erreur de la part de l'interpréteur de commande vous signalant « `bash: less: command not found` », sans que cette erreur ne soit systématique. Il s'agit d'un des gags les plus comiques que j'ai rencontré dans le monde linuxien : en réalité, il s'agit d'une faute de frappe de votre part. En effet, certaines distributions utilisent le plan de clavier `fr-latin1`, et d'autres utilisent plan de clavier `fr-latin9`. Or, sur le plan de clavier `fr-latin9`, la combinaison de touches `AltGr+Espace` génère une espace insécable qui, bien qu'elle soit totalement indifférenciable visuellement de l'espace normale, n'est pas représentée par le même caractère. Or il est très facile de maintenir la touche `AltGr` enfoncée lorsqu'on tape une espace après un tube ! De ce fait, la ligne de

commande est erronée, puisque la commande située à droite du tube est précédée d'une espace insécable, et que ce caractère n'est pas compris par le shell... Par conséquent, si vous rencontrez souvent ce problème, vous pouvez changer de plan de clavier, cesser de mettre des espaces après les tubes, ou apprendre à taper !

Prenons un exemple un peu plus complexe. Supposons que l'on veuille archiver et compresser un répertoire. Il est possible d'archiver ce répertoire avec la commande tar, puis de compresser le fichier archive résultant :

```
tar cvf archive.tar *
gzip archive.tar
```

Cette méthode est correcte, mais souffre d'un défaut : elle utilise un fichier intermédiaire, qui peut prendre beaucoup de place disque. Une méthode plus économe consiste à lancer tar et gzip en parallèle, et à rediriger la sortie standard de l'un dans le flux d'entrée de l'autre. Ainsi, il n'y a plus de fichier temporaire, et la place consommée sur le disque est minimale :

```
tar cv * | gzip > archive.tar.gz
```

La première commande demande à tar d'archiver tous les fichiers du répertoire et d'envoyer le résultat dans le flux standard de sortie. Le pipe redirige ce flux standard vers le flux d'entrée standard de gzip. Celui-ci compresse les données et les émet vers son flux standard de sortie, qui est lui-même redirigé vers le fichier `archive.tar.gz`. Aucun fichier temporaire n'a été utilisé, et on a ainsi économisé l'espace disque de l'archive complète non compressée, c'est-à-dire environ la taille complète du répertoire à archiver. Ce genre de considération peut être très important lorsque le disque dur commence à être plein...

Note : En fait, la commande tar de GNU permet de compresser à la volée les données à archiver, permettant d'éviter de se prendre la tête comme on vient de le faire. Pour cela, il suffit d'utiliser l'option `z` dans la ligne de commande de tar. Ainsi, la ligne de commande suivante fournit le même résultat :

```
tar cvfz archive.tar.gz *
```

Mais cette solution ne fonctionne pas avec les versions non GNU de tar, qui ne supportent pas cette option.

Un autre exemple pratique est le déplacement de toute une arborescence de fichiers d'un système de fichiers à un autre. Vous ne pourrez pas y parvenir à l'aide de la commande `mv`, car celle-ci ne fait que modifier la structure du système de fichiers pour déplacer les fichiers et les répertoires, elle ne peut donc pas fonctionner avec deux systèmes de fichiers. Vous ne pouvez pas non plus utiliser la commande `cp`, car celle-ci ne prendra pas en compte les dates des fichiers, leur propriétaire et leur groupe, ainsi que les liens symboliques et physiques. Il faut donc impérativement utiliser un programme d'archivage. La méthode à suivre est donc de créer une archive temporaire, puis de se déplacer dans le répertoire destination, et enfin d'extraire l'arborescence de l'archive :

```
cd source
tar cvf archive.tar *
cd destination
tar xvf source/archive.tar
rm source/archive.tar
```

Malheureusement, cette technique nécessite beaucoup de place disque, puisque l'archive temporaire est stockée directement sur disque. De plus, elle est assez lente, car toutes les données à copier sont recopiées sur le disque dur, et relues ensuite, pour finalement être détruites... La vraie solution est de réaliser un tube entre les deux processus tar invoqués. Dans ce cas, le transfert se fait simplement via la mémoire vive :

```
cd source
tar cv * | (cd destination ; tar xvf -)
```

La commande à utiliser est cette fois un peu plus compliquée, car la commande d'extraction des fichiers nécessite un changement de répertoire. Il faut donc utiliser une commande multiple du shell. Ces commandes sont constituées

de plusieurs autres commandes séparées par des points virgules. La première commande effectuée ici est le changement de répertoire, et la deuxième est l'extraction par tar de l'archive qui lui est transférée par le flux d'entrée standard (représenté ici par '-'). Ces deux commandes sont mises entre parenthèses, car l'opérateur '|' du tube est prioritaire sur l'opérateur ';' de concaténation des commandes du shell. Si vous trouvez que cela est un peu compliqué, je vous l'accorde. Cependant, la commande qui utilise le tube consomme deux fois moins d'espace disque et est deux fois plus rapide que la commande qui n'en utilise pas. Je vous invite à mesurer le gain de temps sur un répertoire contenant un grand nombre de données (utilisez la commande time !).

V-I-3-b - Les tubes nommés

Les tubes créés par l'opérateur '|' constituent ce que l'on appelle des tubes anonymes, car ils sont créés directement par le shell pour une commande donnée. Il est possible de créer manuellement des tubes en leur donnant un nom, et de les utiliser a posteriori dans plusieurs commandes. Ces tubes constituent ce que l'on appelle des tubes nommés.

En fait, les tubes nommés sont des fichiers spéciaux, que l'on crée dans un système de fichiers capable de les gérer. Les seules opérations réalisables sont l'écriture et la lecture, sachant que les données écrites en premier seront forcément les premières données lues. C'est ce comportement qui a donné leur nom à ces fichiers, que l'on appelle des « FIFO » (abréviation de l'anglais « First In First Out »). De plus, la quantité de données en transit dans ces fichiers est souvent très réduite, ce qui fait que ces données sont toujours placées dans la mémoire cache du système. Ainsi, bien qu'il s'agisse de fichiers, aucune écriture ou lecture sur disque n'a lieu lors de l'utilisation d'un pipe.

Les tubes nommés sont créés par la commande mkfifo, dont la syntaxe est la suivante :

```
mkfifo nom
```

où `nom` est le nom du tube nommé. Notez que cette commande échouera sur les systèmes de fichiers incapables de gérer les tubes nommés.

Une fois créé, le fichier de tube peut être utilisé comme n'importe quel fichier dans les redirections que l'on a vues dans la section précédente. Par exemple, la redirection suivante :

```
ls | less
```

peut être réécrite pour utiliser un tube nommé temporaire de la manière suivante :

```
mkfifo /tmp/tempfifo  
ls > /tmp/tempfifo  
less < /tmp/tempfifo
```

La destruction d'un tube nommé se fait comme n'importe quel fichier, à l'aide de la commande rm.

V-I-3-c - La commande tee

La commande tee est un petit programme permettant d'enregistrer les données qu'il reçoit dans son flux d'entrée standard dans un fichier et de les renvoyer simultanément vers son flux de sortie standard. Elle est couramment utilisée, en conjonction avec les tubes, pour dupliquer un flux de données. Sa syntaxe est la suivante :

```
tee fichier
```

où `fichier` est le nom du fichier dans lequel le flux d'entrée standard doit être enregistré.

Supposons par exemple que l'on désire rediriger tous les messages (d'erreur ou non) de la commande `ls /proc/1/*` dans un fichier `result.txt`, tout en continuant à les visualiser sur l'écran. Pour cela, on utilisera la commande suivante :

```
ls -l /proc/1 2>&1 | tee result.txt
```

À l'issue de cette commande, le fichier `result.txt` contiendra une copie des données qui ont été émises par la commande `ls -l /proc/1 2>&1`.

V-I-3-d - La commande xargs

La commande `xargs` permet d'appeler une autre commande, en passant en paramètre les données qu'elle reçoit dans le flux d'entrée standard. Sa syntaxe est la suivante :

```
xargs commande
```

où `commande` est la commande que `xargs` doit exécuter. `xargs` construira une ligne de commande complète pour cette commande, en utilisant comme paramètres les données issues du flux d'entrée standard. Une fois cette ligne de commande construite, `xargs` l'exécutera. Par exemple, la commande suivante :

```
ls -l
```

peut être exécutée également de la manière suivante :

```
xargs ls
```

et en tapant la chaîne de caractères « `-l` » suivie du caractère de fin de fichier `CTRL+D`.

La commande `xargs` est une commande extrêmement utile lorsqu'elle est utilisée conjointement avec les tubes, parce qu'elle permet d'utiliser le résultat d'une commande en tant que paramètre pour une autre commande. Ce mécanisme est donc complémentaire de celui des pipes, puisque ceux-ci permettaient d'utiliser le résultat d'une commande pour alimenter le flux d'entrée standard d'une autre commande.

Un exemple plus utile que le précédent permettra de mieux comprendre comment on utilise la commande `xargs`. Supposons que l'on désire trouver tous les fichiers d'une arborescence complète dont l'extension est `.txt` et contenant la chaîne de caractères « `test` ». La liste des fichiers de l'arborescence peut être déterminée simplement à l'aide de la commande `find`, et la recherche du texte dans les fichiers se fait naturellement à l'aide de la commande `grep`. On utilisera `xargs` pour construire la ligne de commande pour `grep`, à partir du résultat fourni par la commande `find` :

```
find -name "*.txt" | xargs grep -l "test"
```

Cette commande est plus simple et plus efficace que la commande équivalente :

```
find -name "*.txt" -exec grep -l "test" {} \;
```

parce que `grep` n'est exécuté qu'une seule fois (alors que l'option `-exec` de la commande `find` l'exécute pour chaque fichier trouvé).

V-I-4 - Manipulation des variables d'environnement

Les systèmes Unix permettent de définir un environnement d'exécution pour chaque programme en cours d'exécution. L'environnement est un ensemble de paramètres, que l'on appelle les variables d'environnement, qui permettent de modifier le comportement du programme. Ces variables contiennent une valeur de type chaîne de caractères, dont la signification est propre à chaque variable. Il est d'usage que les noms des variables d'environnement soient écrits complètement en majuscules, mais ce n'est pas une obligation.

Chaque programme est susceptible de reconnaître un certain nombre de variables d'environnement qui lui sont propres, mais il existe également des variables standards que tous les programmes utilisent. C'est notamment le cas de la variable d'environnement `PATH`, qui contient la liste des répertoires dans lesquels le système doit rechercher les

programmes à exécuter. Cette variable permet donc de lancer les programmes en tapant simplement leur nom, et de laisser le système rechercher le fichier de ce programme dans chacun des répertoires indiqués dans cette variable.

Par défaut, les programmes sont lancés avec l'environnement du programme qui les lance, c'est-à-dire dans la plupart des cas l'environnement d'exécution du shell. Les programmes peuvent également définir de nouvelles variables d'environnement, qui seront ainsi accessibles par les programmes qu'ils lanceront eux-mêmes.

Comme tout programme, le shell dispose d'un environnement, qu'il utilise pour stocker ses propres variables. En effet, comme nous l'avons déjà signalé plus haut, le shell est bien plus qu'un interpréteur de commande : il est complètement programmable. Et en tant qu'interpréteur d'un langage de programmation, il fournit la possibilité de définir des variables de ce langage. Les variables du shell sont donc également des variables d'environnement, mais le shell ne les communique pas par défaut aux programmes qu'il lance. Pour être plus précis, le shell utilise deux environnements différents :

- son propre environnement, qui contient les variables d'environnement locales à la session du shell en cours ;
- l'environnement d'exécution, dont les variables d'environnement sont transmises aux programmes que le shell lance.

Il est très facile de définir une variable du shell. Pour cela, il suffit de lui affecter une valeur, à l'aide de la syntaxe suivante :

```
variable=valeur
```

où `variable` est le nom de la variable à définir, et `valeur` est la valeur que l'on désire lui affecter. Notez qu'il n'est pas nécessaire de fournir une valeur. Dans ce cas, la variable ainsi définie sera vide.

Par exemple, la ligne suivante :

```
BONJOUR="Bonjour tout le monde \!"
```

permet de définir la variable `BONJOUR`. Notez que la valeur est encadrée entre guillemets, car elle contient des espaces. Notez également que le caractère point d'exclamation (!) est précédé d'un caractère d'échappement antislash (\), car il a une signification particulière pour le shell. Ce caractère d'échappement permet simplement de signaler au shell qu'il ne doit pas interpréter le caractère qui le suit, et fait donc en sorte que le point d'exclamation fasse partie de la chaîne de caractères à affecter à la variable. Bien entendu, le caractère antislash étant lui-même un caractère spécial pour le shell, il doit lui-même être préfixé d'un autre antislash si l'on désire l'utiliser dans une chaîne de caractères.

La valeur d'une variable peut être récupérée simplement en préfixant le nom de la variable du symbole dollar (\$). Ainsi, la commande suivante permet d'afficher le contenu de la variable `BONJOUR` :

```
echo $BONJOUR
```

Les variables ainsi définies ne font partie que de l'environnement du shell, elles ne sont donc pas accessibles aux programmes que le shell lance. Donc, si l'on relance un nouveau shell avec la commande suivante :

```
bash
```

et que l'on essaie de lire le contenu de la variable `BONJOUR` avec la commande `echo`, on obtient une chaîne vide. Cela est normal, puisque le deuxième shell (c'est-à-dire celui qui est en cours d'exécution) n'utilise pas le même environnement que le premier shell. Vous pouvez quitter le nouveau shell avec la commande suivante :

```
exit
```

Dès lors, vous serez à nouveau dans le shell initial, et la variable `BONJOUR` sera à nouveau accessible.

Pour rendre une variable du shell accessible aux programmes que celui-ci peut lancer, il faut l'exporter dans l'environnement d'exécution. Cela peut être réalisé avec la commande `export` :

```
export variable
```

où `variable` est le nom de la variable du shell à exporter dans l'environnement d'exécution.

La syntaxe précédente exporte de manière permanente les variables du shell. Mais il existe également une autre syntaxe, qui permet de ne définir des variables d'environnement que pour l'environnement d'exécution d'une seule commande. Cette syntaxe consiste simplement à préfixer la commande à exécuter par la définition de ladite variable. Par exemple, la commande suivante :

```
BONSOIR="Bonsoir tout le monde \!" bash
```

permet de lancer un shell et de lui communiquer la variable d'environnement `BONSOIR`. Cette variable ne sera définie que pour ce programme, si l'on quitte ce shell avec un `exit`, la variable `BONSOIR` ne sera plus définie.

Une variable peut être détruite à tout instant à l'aide de la commande `unset`. Cette commande prend en paramètre le nom de la variable à supprimer. Par exemple, la commande suivante supprime notre variable :

```
unset BONJOUR
```

Vous pouvez à tout moment visualiser l'ensemble des variables définies avec la commande `set`. Le tableau donné ci-dessous vous présentera les variables d'environnement les plus utilisées, que la plupart des programmes utilisent pour permettre à l'utilisateur de modifier leur comportement :

Tableau 5-3. Variables d'environnements courantes

Nom	Signification
HOME	Chemin du répertoire personnel de l'utilisateur.
USER	Nom de login de l'utilisateur. Cette information est également disponible au travers de la variable d'environnement <code>LOGNAME</code> .
TERM	Type de terminal utilisé. La valeur de cette variable sert aux applications pour déterminer les caractéristiques du terminal et ses fonctionnalités afin d'optimiser leur affichage. La valeur de cette variable est souvent <code>linux</code> sur les consoles Linux, et <code>xterm</code> dans les émulateurs de terminal graphiques sous X11. Nous verrons l'utilité de cette variable plus en détail dans la Section 6.9.7 .
SHELL	Chemin sur le fichier de programme du shell actuellement utilisé. Sous Linux, il s'agit souvent du shell <code>bash</code> .
PATH	Liste des répertoires dans lesquels les programmes à exécuter seront recherchés. Cette liste ne doit pas contenir le répertoire courant (<code>.</code>) pour des raisons de sécurité de base (il suffit de placer un cheval de Troie portant le nom d'une commande classique dans un répertoire pour que l'utilisateur le lance sans s'en rendre compte).
LD_LIBRARY_PATH	Liste des répertoires dans lesquels les bibliothèques dynamiques seront recherchées si elles ne sont pas trouvables dans les répertoires classiques des bibliothèques de programme du système.
C_INCLUDE_PATH	Liste des répertoires dans lesquels le compilateur C recherchera les fichiers d'en-tête lors de la compilation des fichiers sources C. Cette liste doit contenir les

	répertoires additionnels, qui ne sont pas déjà pris en compte automatiquement par le compilateur C.
CPLUS_INCLUDE_PATH	Liste des répertoires dans lesquels le compilateur C++ recherchera les fichiers d'en-tête lors de la compilation des fichiers sources C/C++. Cette liste doit contenir les répertoires additionnels, qui ne sont pas déjà pris en compte automatiquement par le compilateur C++.
LIBRARY_PATH	Liste des répertoires dans lesquels les bibliothèques à utiliser lors de l'édition de liens des programmes doivent être recherchées. Cette variable n'est utilisée que par les outils de développement lors de la compilation de fichiers sources et elle ne doit pas être confondue avec la variable d'environnement LD_LIBRARY_PATH, qui indique la liste des répertoires dans lequel l'éditeur de liens dynamiques recherchera les bibliothèques dynamiques utilisées par les programmes lors de leur chargement. Les notions de fichiers sources et de compilation seront détaillées dans le Chapitre 7 .
TMPDIR	Répertoire des fichiers temporaires. Par défaut, le répertoire des fichiers temporaires est le répertoire /tmp/, mais il est possible d'en changer grâce à cette variable d'environnement.
TZ	Définition de la zone horaire de l'utilisateur. Le système travaillant exclusivement en temps universel, chaque utilisateur peut définir sa propre zone horaire pour obtenir l'affichage des dates et des heures dans son temps local. Le format de cette variable d'environnement est assez complexe. Il est constitué de plusieurs champs séparés par des espaces, représentant successivement le nom du fuseau horaire (au moins trois caractères), le décalage à ajouter à l'heure universelle pour obtenir l'heure locale, le nom du fuseau horaire pour l'heure d'été, le décalage pour l'heure d'été, et les dates de début et de fin de l'heure d'été. Les décalages horaires doivent être exprimés avec un '+' pour les fuseaux horaires placés à l'ouest de Greenwich, '-' pour ceux situés à l'est. Les dates de début et de fin de la période d'heure d'été peuvent être exprimées de deux manières différentes. La première méthode est d'indiquer le numéro du jour dans l'année après la lettre 'J'. Ce numéro ne doit pas tenir compte du 29 février, même pour les années bissextiles. La deuxième méthode est d'indiquer le mois de l'année, la semaine du mois et le jour de la semaine, séparés par des '.', et après la lettre 'M'. Les mois sont comptés de 1 à 12, les semaines de 1 à 5 et les jours de 0 à 6, 0 étant le dimanche. Seul le premier champ est obligatoire, et il est possible d'utiliser les noms de fuseaux horaires définis par la bibliothèque C. En France, on utilise normalement le fuseau CES (temps d'Europe centrale).
LANG	Nom de la locale à utiliser par défaut pour les paramètres d'internationalisation des applications. Cette valeur sera utilisée pour les paramètres qui n'en définissent pas une explicitement. Elle doit être composée de deux codes à deux caractères, le premier indiquant la langue, et le deuxième le pays (car plusieurs pays peuvent parler la même langue, et un pays peut avoir plusieurs langues nationales). Pour la France, on utilise normalement la

	valeur « fr_FR ». Cette valeur peut être redéfinie par l'une des variables d'environnement décrites ci-dessous.
LC_MESSAGES	Nom de la locale à utiliser pour déterminer la langue des messages. La valeur par défaut est spécifiée par la variable d'environnement LANG.
LC_TYPE	Nom de la locale à utiliser pour déterminer les règles de classification des caractères. La classification des caractères permet de dire si un caractère est un chiffre ou non, s'il est en majuscule ou en minuscule, etc. La valeur par défaut est spécifiée par la variable d'environnement LANG.
LC_COLLATE	Nom de la locale à utiliser pour déterminer les règles de comparaison des caractères. La comparaison des caractères est utilisée pour les tris lexicographiques (tri par ordre alphabétique par exemple). La valeur par défaut est spécifiée par la variable d'environnement LANG.
LC_MONETARY	Nom de la locale à utiliser pour déterminer l'emplacement et le caractère représentant le symbole monétaire du pays. La valeur par défaut est spécifiée par la variable d'environnement LANG.
LC_NUMERIC	Nom de la locale à utiliser pour déterminer les conventions locales d'écriture des nombres (séparateur décimal, format de la virgule, etc.). La valeur par défaut est spécifiée par la variable d'environnement LANG.

En résumé, le shell utilise les variables d'environnement du système pour gérer ses propres variables, et permet de les exporter vers l'environnement d'exécution qu'il communique aux commandes qu'il lance. Un grand nombre de variables d'environnement classiques sont reconnues par les programmes. Elles servent à paramétrer leur comportement. Nous reverrons ultérieurement quelques-unes de ces variables lors de la configuration du système de base.

V-I-5 - Caractère d'échappement et chaînes de caractères

Un certain nombre de caractères sont interprétés par le shell d'une manière spéciale. Nous en avons déjà vu quelques-uns pour les redirections et les tubes, mais il en existe d'autres. Par conséquent, il faut utiliser une syntaxe particulière lorsqu'on désire utiliser un de ces caractères dans une commande du shell sans qu'il soit interprété par le shell. Pour cela, il suffit de faire précéder ces caractères du caractère d'échappement antislash (caractère de la barre oblique inverse, '\'). Ce caractère permet d'indiquer au shell que le caractère suivant doit être traité tel quel et ne doit pas être interprété avec son sens habituel. Par exemple, pour créer un répertoire nommé <, on utilisera la commande suivante :

```
mkdir \<
```

Bien entendu, le caractère antislash peut lui-même être précédé d'un autre antislash, lorsqu'on veut l'utiliser en tant que caractère normal.

Le caractère d'échappement antislash permet également, lorsqu'il est placé en fin de ligne, de supprimer le saut de ligne qui le suit. Cela signifie qu'il permet de répartir une commande trop longue sur plusieurs lignes, à des fins de lisibilité. Vous trouverez quelques exemples de cette notation plus loin dans ce document, pour présenter des commandes trop longues pour tenir sur une page A4.

Il peut être relativement fastidieux de devoir taper des antislashs dans les chaînes de caractères qui contiennent beaucoup de caractères interprétables par le shell. C'est pour cela que le shell permet de définir des chaînes de caractères dont il ignore le contenu lors de l'analyse syntaxique. Ces chaînes de caractères sont simplement données entre guillemets simples (caractère '). Par exemple, la commande suivante :


```
MESSAGE='La syntaxe est A | B'
```

permet d'affecter la chaîne de caractères `La syntaxe est A | B`, contenant des espaces et le caractère `|` normalement utilisé par le shell pour les tubes, dans la variable d'environnement `MESSAGE`.

Note : Une chaîne de caractères commence par un guillemet et se termine par un guillemet. Les chaînes de caractères ne peuvent donc pas contenir de guillemet, même précédé d'un caractère d'échappement.

On veillera à ne surtout pas confondre les guillemets simples (caractère `'`) avec les guillemets inverses (caractère ```). Ces deux caractères se ressemblent en effet énormément dans certaines polices de caractères, mais ont néanmoins une signification très différente. Le premier sert à définir des chaînes de caractères, et le deuxième à exécuter une commande et à en inclure le résultat dans une autre commande. Nous verrons plus loin comment utiliser ce type de guillemets.

Les guillemets simples sont donc très pratiques pour écrire simplement une chaîne de caractères, mais ne permettent pas de bénéficier des fonctionnalités de substitutions du shell, par exemple le remplacement d'une variable par sa valeur dans la chaîne de caractères. De plus, elles ne peuvent pas contenir de guillemets simples, puisque c'est leur caractère de terminaison. C'est pour ces raisons que le shell donne la possibilité de définir des chaînes de caractères plus souples, à l'aide des guillemets doubles (caractère `"`). Dans ces chaînes de caractères, la plupart des caractères normalement interprétés par le shell ne le sont plus, comme pour les chaînes de caractères utilisant les guillemets simples. Cependant, les caractères spéciaux `$`, ``` et `\` conservent leur signification initiale. Il est donc possible, par exemple, d'utiliser des variables d'environnement dans les chaînes de caractères de ce type :

```
echo "Mon nom est $USER"
```

Le caractère d'échappement antislash peut toujours être utilisé, en particulier pour insérer un caractère de guillemets doubles dans une chaîne de caractères. En effet, ce caractère marquerait la fin de la chaîne de caractères s'il n'était pas précédé d'un antislash.

Note : Remarquez que les guillemets et les caractères d'échappement ne sont utilisés que pour l'analyse de la ligne de commande. Une fois toutes les chaînes de caractères et toutes les substitutions traitées, les guillemets et les caractères d'échappement inutiles sont supprimés. En pratique, ce sont tous les caractères d'échappement et les guillemets qui restent après traitement de la ligne de commande et qui ne font pas partie du résultat d'une des substitutions. Ainsi, la commande suivante :

```
echo "Bonjour tout le monde"
```

a pour but de passer la chaîne de caractères `Bonjour tout le monde` en tant que premier (et unique) paramètre de la commande `echo`, puis de l'exécuter. Les guillemets ne font pas partie de la chaîne de caractères, ils ont été supprimés par le shell et seul le contenu de la chaîne sera effectivement affiché.

Notez que la commande précédente est très différente de celle-ci :

```
echo Bonjour tout le monde
```

même si le résultat est le même. En effet, cette dernière commande passe les chaînes de caractères `Bonjour`, `tout`, `le` et `monde` en tant que paramètres (4 au total) à la commande `echo`, alors que l'utilisation des guillemets permet de passer toute la phrase en un seul paramètre. On peut voir la différence en utilisant plus d'un espace entre chaque mot : les espaces superflus ne sont conservés que dans la première commande.

V-I-6 - Les substitutions

L'une des fonctionnalités les plus puissantes du shell est sans doute sa capacité à effectuer des substitutions d'expressions par leur valeur. L'une des substitutions les plus courantes est sans doute le remplacement d'une variable par sa valeur, mais le shell peut faire beaucoup plus que cela. Les lignes de commandes peuvent être écrites en utilisant différents types d'expressions spéciales, qui seront remplacées par leur valeur par le shell avant l'exécution de la commande. Ces expressions permettent de spécifier des motifs de chaîne de caractères, d'exprimer des chemins partiels sur des fichiers ou des répertoires, de récupérer la valeur des variables du shell, et de calculer des expressions mathématiques, voire d'inclure le résultat d'une autre commande dans la ligne de commande en cours.

Les mécanismes des substitutions décrits ci-dessous sont présentés par ordre de priorité décroissante. Cela signifie que si une expression substituable contient elle-même une autre expression substituable de priorité inférieure, cette expression sera remplacée après la substitution de l'expression contenante.

V-I-6-a - Génération de chaînes de caractères selon un motif

Il est possible de demander au shell de générer une série de chaînes de caractères selon un motif simple. Ce motif est toujours constitué d'un préfixe, suivi d'une partie variable, suivie d'un suffixe. La partie variable du motif est celle qui subira les substitutions pour générer une liste de chaînes de caractères commençant par le préfixe suivi du résultat de la substitution et se terminant par le suffixe. Cette partie variable doit être spécifiée entre accolades, et prend la forme d'une liste de valeurs possibles pour chaque substitution, séparées par des virgules. Par exemple, la commande suivante :

```
ls test{0,1,2,3,4}
```

sera transformée par le shell en la commande suivante :

```
ls test0 test1 test2 test3 test4
```

Note : Ceux qui se souviennent un peu de leurs mathématiques se diront qu'il s'agit là d'une factorisation. C'est rigoureusement exact.

V-I-6-b - Substitution du nom d'utilisateur

Le caractère tilde ('~') est remplacé par le nom de l'utilisateur courant ou, à défaut de nom, par le chemin sur le répertoire personnel de cet utilisateur. Il est possible de spécifier un autre utilisateur en donnant le nom de login de cet autre utilisateur immédiatement après le caractère tilde. Par exemple, la commande suivante :

```
cp *.txt ~jean
```

permet de copier tous les fichiers d'extension .txt dans le répertoire personnel de l'utilisateur jean.

V-I-6-c - Remplacements de variables

Comme il l'a déjà été indiqué [plus haut](#), la valeur des variables du shell et des variables d'environnement peut être récupérée en préfixant le nom de la variable par le caractère dollar ('\$'). En fait, cette écriture est l'une des formes les plus simples que peuvent prendre les substitutions de paramètres. En effet, il est possible de remplacer l'expression par une partie seulement de la valeur de la variable, ou une par une autre valeur calculée à partir de celle de la variable.

En pratique, les expressions utilisées par les substitutions de variables peuvent être relativement compliquées, et il peut être nécessaire de les isoler du reste de la ligne de commande à l'aide d'accolades. La syntaxe exacte complète de ce type de substitution est donc la suivante :

```
${expression}
```

où `expression` est l'expression qui définit la chaîne de remplacement à utiliser.

Si cette expression est un nom de variable, ce sera le contenu de cette variable qui sera utilisé pour la substitution. Il est possible de fournir une valeur par défaut pour le cas où cette variable ne contient rien ou n'est pas définie. Pour cela, on utilisera la syntaxe suivante :

```
${variable:-valeur}
```

où `valeur` est la valeur par défaut à utiliser dans ce cas. Notez que la variable reste indéfinie après la substitution. Pour fixer la valeur de la variable à cette valeur par défaut en plus d'effectuer la substitution, on utilisera plutôt la syntaxe suivante :

```
${variable:=valeur}
```

`valeur` a toujours la même signification dans cette syntaxe.

Il est parfois préférable d'afficher un message d'erreur plutôt que de donner une valeur par défaut lorsqu'une variable n'est pas définie. Cela peut se faire avec la syntaxe suivante :

```
${variable:?message}
```

où `message` est le message à afficher dans le cas où la variable `variable` est non définie ou de valeur nulle.

Si l'on veut tester si une variable est non définie et renvoyer une valeur spécifique si elle est définie, on utilisera la syntaxe suivante :

```
${variable:+valeur}
```

où `valeur` est la valeur à renvoyer si la variable est définie. Si la variable n'est pas définie, la substitution sera faite avec la chaîne de caractères vide (l'expression complète sera donc supprimée).

Le shell permet également de faire la substitution avec une sous-chaîne de la valeur de la variable, à partir d'une position donnée et d'une longueur. La syntaxe à utiliser est donnée ci-dessous :

```
${variable:position:longueur}
```

où `position` est la position à laquelle commence la sous-chaîne à extraire, et `longueur` est le nombre de caractères à extraire. Ce dernier champ est facultatif (on ne mettra pas non plus les deux-points précédents si on décide de ne pas spécifier de longueur). Si on ne le précise pas, la sous-chaîne extraite sera constituée du reste de la valeur de la variable à partir de la position indiquée. La position quant à elle doit être positive ou nulle. Une valeur négative indique un point de départ correspondant au nombre de caractères correspondant à partir de la droite de la valeur de la variable. Si l'on veut obtenir la longueur d'une chaîne de caractères contenue dans une variable, on utilisera cette syntaxe :

```
${#variable}
```

où `variable` est toujours le nom de la variable.

Il est également possible de considérer que la valeur d'une variable est une chaîne de caractère préfixée d'une autre chaîne de caractères particulière. Le shell permet d'extraire la chaîne de caractères principale, en supprimant ce préfixe. Pour réaliser cette opération, on utilisera l'une des syntaxes suivantes :

```
${variable#préfixe}
```

ou :

```
${variable##préfixe}
```

où `variable` est la variable contenant la chaîne de caractères à traiter, et `préfixe` est le préfixe à supprimer.

En fait, le préfixe peut être spécifié à l'aide d'un motif de caractères. Ce motif peut correspondre à une partie plus ou moins grande de la valeur de la variable. Dans ce cas, il y a plusieurs manières d'interpréter ce motif, et donc plusieurs choix de préfixes possibles à supprimer. La première syntaxe devra être utilisée lorsqu'on désire supprimer le plus petit préfixe possible correspondant au motif. La deuxième syntaxe, quant à elle, permettra de supprimer le préfixe le plus long. Par exemple, si la variable `VAR` contient la chaîne de caractères `abbbc`, la commande suivante :

```
echo ${VAR#a*b}
```

affichera la chaîne de caractères `bbc`, car le plus petit préfixe correspondant au motif `a*b` est `ab`. Inversement, la commande :

```
echo ${VAR##a*b}
```

utilisera le préfixe le plus long, à savoir `abbb`. Le résultat de cette substitution sera donc la chaîne de caractères `c`. La syntaxe des motifs de caractères utilisés ici sera précisée dans la [Section 5.9.7](#).

Le shell fournit une syntaxe similaire pour extraire des suffixes de la valeur des variables. Cette syntaxe utilise simplement le caractère `%` au lieu du caractère `#`. Comme pour les préfixes, le fait de doubler ce caractère implique que le suffixe le plus long correspondant au motif sera utilisé, alors que l'utilisation d'un seul `%` permet de choisir le suffixe le plus court. Ainsi, la commande :

```
echo ${VAR%b*c}
```

affichera la chaîne de caractères `abb`, alors que la commande :

```
echo ${VAR%%b*c}
```

n'affichera que `a`.

Pour terminer ce tour d'horizon des remplacements de variables, nous allons voir les possibilités de recherche et de remplacement du shell dans les chaînes de caractères contenues dans des variables. La syntaxe suivante :

```
${variable/motif/remplacement}
```

permet de rechercher la plus grande sous-chaîne de caractères correspondant au motif `motif` dans la chaîne contenue dans la variable `variable`, et de remplacer cette sous-chaîne par la chaîne de caractères `remplacement`. Par exemple, si la variable `VAR` contient la chaîne de caractères `abab`, la commande suivante :

```
echo ${VAR/b/d}
```

affichera la chaîne de caractères `adab`.

Ce remplacement n'est donc effectué qu'une seule fois. Si l'on veut que toutes les occurrences du motif soient remplacées par la chaîne de remplacement, il suffit de doubler le premier / :

```
${variable//motif/remplacement}
```

Dans les deux syntaxes, la présence du champ remplacement est facultative. Cela permet de supprimer purement et simplement les sous-chaînes de caractères qui correspondent au motif.

La syntaxe des motifs sera détaillée dans la [Section 5.9.7](#). Cependant, une précision doit être signalée : si le motif commence par le caractère #, il sera obligatoirement recherché au début de la chaîne de caractères contenue dans la variable. De même, si le motif commence par le caractère %, il sera obligatoirement recherché à la fin de cette chaîne. Ces deux notations permettent d'obtenir le même effet que les suppressions de préfixes et de suffixes présentées plus haut.

V-I-6-d - Substitution du résultat d'une commande

Le shell peut évaluer une commande apparaissant dans une expression afin de la remplacer par son résultat dans la commande appelante. Il existe deux syntaxes pour réaliser ce type de substitutions. La première, et la plus classique (voire historique), utilise des guillemets inverses :

```
`commande`
```

où `commande` est la commande devant être remplacée par son résultat (c'est-à-dire ce qu'elle enverra ce résultat sur le flux standard de sortie). Pour donner un exemple, la commande suivante :

```
kill `cat /var/pid/p.pid`
```

a pour résultat de lancer un signal SIGTERM au processus dont le PID est stocké dans le fichier `/var/pid/p.pid`. La commande `cat` est utilisée pour afficher le contenu de ce fichier, et elle est substituée par ce contenu. En fin de compte, la commande `kill` est appliquée au PID affiché par `cat`.

La deuxième syntaxe utilisable est la suivante :

```
$(commande)
```

où `commande` est toujours la commande à exécuter et à substituer. La différence entre ces deux syntaxes est que, dans le premier cas, les caractères \$, ` et \ sont toujours interprétés par le shell et doivent être précédés d'un antislash s'ils doivent apparaître tels quels dans la commande à substituer, alors que, dans le deuxième cas, on peut utiliser tous les caractères sans protection particulière (sauf, bien entendu, la parenthèse fermante, puisqu'elle marque la fin de la commande).

V-I-6-e - Évaluation d'expressions arithmétiques

En général, le shell ne manipule que des chaînes de caractères. Cependant, il est capable d'évaluer des expressions mathématiques simples faisant intervenir des entiers. Pour cela, il faut utiliser la syntaxe suivante :

```
$( (expression) )
```

où `expression` est l'expression à évaluer.

Les expressions mathématiques peuvent contenir tous les opérateurs classiques du langage C : addition, soustraction, multiplication et division. Il existe en plus un opérateur d'élévation à la puissance, représenté par une double étoile (**). Les opérateurs de décalage binaires vers la gauche (<<) et la droite (>>) sont également utilisables,

ainsi que les opérateurs de manipulation de bits & (« ET binaire »), | (« OU binaire »), ^ (« OU binaire exclusif ») et ~ (« négation binaire »).

Comme en C, les comparaisons logiques peuvent également être évaluées, elles ont la valeur 1 lorsque l'expression qui les utilise est vraie, et 0 dans le cas contraire. Les opérateurs disponibles sont ==, <, <=, >, >= et !=. Les tests peuvent être composés à l'aide des opérateurs && (« ET logique ») et || (« OU logique »).

Les divers opérateurs d'affectation du langage C +=, -=, etc. sont également disponibles.

V-I-6-f - Substitution de commandes

Nous avons déjà vu qu'il était possible de récupérer le résultat d'une commande et de l'injecter dans le flux d'entrée standard d'un processus par l'intermédiaire d'un pipe (cf. [Section 5.9.3.2](#)). Mais le shell fournit une généralisation de cette fonctionnalité sous la forme des substitutions de commandes. Ces substitutions permettent de lancer une commande et de remplacer son expression par un pipe nommé, grâce auquel on peut communiquer avec le processus qui exécute la commande.

La syntaxe utilisée par les substitutions de commande est similaire à celle des redirections classiques :

```
< (command)
```

ou :

```
> (command)
```

où `command` est la commande à substituer.

Note : Attention à ne pas mettre d'espace entre le caractère de redirection et la parenthèse ouvrante, faute de quoi le shell signalera une erreur.

La première syntaxe permet de lancer une commande en arrière-plan en redirigeant son flux standard de sortie vers un descripteur de fichiers du shell. Le résultat de cette substitution est le nom du fichier `/dev/fd/n`, permettant de lire les données écrites par la commande dans ce descripteur de fichier. En pratique, on utilise donc cette substitution en lieu et place d'un fichier d'entrée pour une commande normale. La deuxième commande permet de lancer également une commande en arrière-plan, mais en redirigeant le flux d'entrée standard de cette commande cette fois. Il est alors possible de fournir les données nécessaires à cette commande en écrivant dans le fichier `/dev/fd/n` dont le nom est fourni par le résultat de la substitution.

Ces deux commandes permettent donc de simplifier l'usage des pipes nommés, en évitant d'avoir à créer un fichier de tube nommé manuellement et d'avoir à lancer les deux commandes devant se servir de ce tube pour communiquer. Ainsi, la commande suivante :

```
cat <(ls)
```

est fonctionnellement équivalente à la série de commandes suivante :

```
mkfifo /tmp/lsfifo
ls > /tmp/lsfifo
cat /tmp/lsfifo
rm /tmp/lsfifo
```

Les substitutions de commandes sont donc nettement plus pratiques et plus sûres, car elles n'imposent pas la création d'un fichier de pipe nommé dont le nom peut être choisi arbitrairement.

V-I-6-g - Découpage en mots

Les résultats provenant des substitutions vues précédemment sont systématiquement décomposés en série de mots par le shell avant de poursuivre le traitement de la ligne de commande. Cela signifie que les résultats de substitutions sont analysés pour identifier les mots qu'ils contiennent, en se basant sur la notion de séparateur. Par défaut, les séparateurs utilisés sont l'espace, le caractère de tabulation et le retour de ligne, mais il est possible de spécifier des séparateurs différents à l'aide de la variable d'environnement IFS (abréviation de l'anglais « Internal Field Separator »).

Par exemple, le résultat de la commande `ls` dans la commande suivante :

```
echo `ls`
```

est une chaîne de caractères contenant la liste des fichiers du répertoire courant, chacun étant séparé du suivant par un caractère de saut de ligne. La substitution du résultat de cette commande est donc soumise au découpage en mots, et chaque caractère de retour à la ligne est interprété comme un séparateur. Par conséquent, cette chaîne de caractères est transformée en une liste de mots, chacun de ces mots étant un des noms de fichiers renvoyés par la commande `ls`. Finalement, la commande `echo` est appelée, avec comme paramètres ces noms de fichiers, à raison d'un par paramètre. Les noms de fichiers sont donc affichés sur une seule ligne.

Note : Ce découpage en mot est effectué automatiquement par le shell à la suite des substitutions vues précédemment. Cela signifie en particulier que s'il n'y a pas de substitution, il n'y a pas de découpage en mots non plus.

V-I-6-h - Remplacement des caractères génériques

Si, après avoir appliqué toutes les formes de substitutions précédentes, le shell trouve des caractères génériques `*` et `?` dans l'expression en cours de traitement, il interprétera la partie de l'expression contenant ces caractères comme un motif représentant des chemins de fichier Unix. Les caractères `*` et `?` auront donc le comportement que l'on a déjà décrit dans la [Section 5.5](#). Ce motif sera donc remplacé par autant de chemins Unix lui correspondant que possible. Rappelons que le caractère générique `*` représente 0 ou plusieurs caractères quelconques, et que le caractère générique `?` représente un caractère et un seul. Les chemins générés sont classés par ordre alphabétique.

Il est possible également de restreindre le jeu de caractères utilisé par le shell pour rechercher les noms de fichiers correspondants au motif. Pour cela, il faut lui indiquer un ensemble de caractères ou de plages de caractères utilisables, séparés par des virgules, et entre crochets. Les plages de caractères sont spécifiées en indiquant le premier et le dernier caractère, séparés par un tiret. Par exemple, la commande suivante :

```
ls [a-c,m-t]*.txt
```

permet d'afficher tous les fichiers dont le nom commence par les lettres `a`, `b`, `c` et les lettres allant de `m` à `t`, et dont l'extension est `.txt`. Vous trouverez de plus amples renseignements sur la syntaxe de ces motifs dans la [Section 5.9.7](#).

Sauf paramétrage pour indiquer explicitement de faire le contraire, le shell ignore systématiquement les répertoires `.` et `..` dans les substitutions. Cela est très important. En effet, une commande utilisant le caractère générique `*` ne s'appliquera pas, par défaut, sur le répertoire courant et le répertoire parent. Paramétrer `bash` pour qu'il prenne en compte ces répertoires peut être extrêmement dangereux, surtout avec une commande telle que `rm -f *`, qui dans ce cas effacerait également les répertoires parents en plus du contenu du répertoire courant !

V-I-7 - Les expressions rationnelles

Les substitutions de variables et de noms de fichiers utilisent des motifs pour identifier des chaînes de caractères. Ces motifs peuvent être reconnus dans plusieurs chaînes de caractères différentes, car ils contiennent une ou plusieurs parties variables qui pourront représenter chacune une sous-chaîne des chaînes qui vérifient ce motif. Par exemple,

le motif `a*b` représente toute chaîne de caractères commençant par un `a` et se terminant par un `b`. La sous-chaîne située entre ces deux caractères peut être quelconque, et constitue la partie variable du motif.

La syntaxe utilisée pour définir les motifs de chaînes de caractères dans le shell `bash` est un sous-ensemble d'un langage plus complexe permettant de décrire ce que l'on appelle les expressions rationnelles (l'usage dit également « expressions régulières »). Le langage des expressions rationnelles est relativement compliqué, mais extrêmement puissant. Ce langage permet d'identifier avec précision des sous-chaînes de caractères dans une chaîne de caractères à l'aide des parties variables des expressions rationnelles, et permet éventuellement de remplacer ces sous-chaînes par des chaînes de substitutions. Malheureusement, la description des expressions rationnelles pourrait prendre plusieurs pages, aussi ne verrons-nous ici que les expressions utilisables dans les substitutions du shell `bash`.

Comme vous l'avez sans doute déjà deviné au travers des exemples précédents, le caractère `*` permet d'identifier une quelconque chaîne de caractères, y compris la chaîne vide. Utilisé dans les expressions rationnelles, il constitue la partie variable principale de ces expressions. De la même manière, le caractère `?` représente un et un seul caractère quelconque. Ce caractère sera donc utilisé quand on désirera contrôler la taille de la partie variable d'une expression rationnelle, éventuellement en le répétant un certain nombre de fois.

Les deux caractères de substitutions précédents peuvent contenir n'importe quel caractère, ce qui peut parfois ne pas être assez restrictif dans la définition d'un motif. Le shell fournit donc une syntaxe plus évoluée, permettant de définir précisément le jeu de caractères auquel un caractère du motif doit appartenir. Cette syntaxe consiste simplement à donner la liste des caractères du jeu de caractères entre crochets :

```
[&#8230;]
```

Les points de suspension représentent ici l'ensemble des caractères qui peuvent apparaître dans le motif ainsi défini. Notez que dans le cas d'une suite de caractères, il suffit de spécifier le premier et le dernier caractère, et de les séparer par un trait d'union (caractère `-`). Ainsi, le motif suivant :

```
[a-egt]
```

représente n'importe lequel des caractères de `'a'` à `'e'`, plus les caractères `'g'` et `'t'`.

Note : Pour spécifier le caractère `-` lui-même, il suffit de le placer tout seul au début ou à la fin de la liste de caractères spécifiée entre les crochets. De même, pour spécifier le caractère `]` lui-même (normalement utilisé pour marquer la fin du jeu de caractères), il faut le placer au début de la liste, juste après le crochet ouvrant.

Pour finir, sachez que le shell `bash` est également capable de prendre en charge des expressions rationnelles plus complexes que celles présentées ici. Cependant, ces expressions ne sont pas actives par défaut, et ne sont donc accessibles qu'en activant une option complémentaire du shell. Ces extensions ne seront pas décrites ici, mais vous pouvez consulter la page de manuel de `bash` si vous désirez en savoir plus à ce sujet.

V-I-8 - Structures de contrôle

Tout langage de programmation digne de ce nom dispose de structures de contrôles évoluées permettant de contrôler l'exécution du programme, de réaliser des boucles et de structurer l'ensemble d'un programme. Le shell n'échappe pas à la règle, et fournit la plupart des constructions classiques. Cette section a pour but d'exposer leurs syntaxes.

V-I-8-a - Les instructions composées

Dans le langage du shell, une instruction se termine soit par un retour à la ligne (non précédé d'un antislash), soit par un point-virgule. Les instructions peuvent être pourtant très complexes, car elles peuvent contenir des tubes et des redirections. En fait, une instruction peut à peu près être définie comme étant une ligne de commande normale du shell.

Le shell permet bien entendu de réaliser des instructions composées, afin de regrouper plusieurs traitements dans un même bloc d'instructions. La méthode la plus simple pour réaliser un bloc d'instructions est tout simplement de les regrouper sur plusieurs lignes, ou de les séparer par des points-virgules, entre accolades. Par exemple, les instructions suivantes constituent un bloc d'instructions :

```
{  
    cd /tmp  
    rm *.bak  
}
```

Notez que l'accolade fermante est considérée comme une instruction à part entière. Cela signifie que si l'on ne met pas l'accolade fermante sur une ligne indépendante, il faut faire précéder l'instruction précédente d'un point-virgule. De même, il faut le faire suivre d'un autre point-virgule s'il ne se trouve pas à la fin d'une ligne.

Les instructions des instructions composées créées à l'aide des accolades sont exécutées au sein du shell courant. Les variables qu'elles définissent, ainsi que les changements de répertoires, sont donc toujours valides à l'issue de l'exécution de ces instructions. Si cela n'est pas désirable, on pourra créer des instructions composées à l'aide de parenthèses. Les instructions seront alors exécutées dans un autre shell, lancé pour l'occasion, et elles n'auront donc pas d'effet de bord imprévu dans le shell appelant. Par exemple, le répertoire courant à l'issue de l'instruction composée précédente est le répertoire `/tmp/`, alors que l'instruction composée suivante :

```
(  
    cd /tmp  
    rm *.bak  
)
```

ne change pas le répertoire courant.

Note : On ne confondra pas les instructions composées utilisant des parenthèses et les substitutions de résultat de commande. Les instructions composées renvoient le code d'erreur de la dernière instruction exécutée, alors que le résultat des substitutions est ce que la commande a écrit sur son flux de sortie standard.

Le shell permet également de réaliser des instructions composées conditionnelles, où l'exécution de chaque instruction de l'instruction composée est conditionnée par le résultat de l'instruction précédente. Ces instructions composées sont définies à l'aide des opérateurs `||` et `&&`. La syntaxe de ces opérateurs est la même :

```
command1 || command2  
command1 && command2
```

où `command1` et `command2` sont deux commandes du shell (composées ou non). Avec l'opérateur `||`, la commande `command2` n'est exécutée que si le code de retour de la commande `command1` est non nul, ou, autrement dit, si cette commande ne s'est pas exécutée correctement. Inversement, avec l'opérateur `&&`, la commande `command2` n'est exécutée que si la première commande s'est exécutée correctement (et renvoie donc un code de retour nul). Par exemple, la commande suivante :

```
rm *.txt 2> /dev/null || echo "Aucun fichier à supprimer"
```

permet d'effacer tous les fichiers d'extension `.txt`, ou d'afficher le message d'erreur « Aucun fichier à supprimer » s'il n'existe pas de fichier ayant une telle extension.

Les instructions composées peuvent être utilisées comme n'importe quelle commande normale. En particulier, elles peuvent être utilisées dans des commandes plus complexes, par exemple comme destination d'un tube. C'est ce que faisait l'exemple de déplacement de toute une arborescence dans la [Section 5.9.3.1](#).

V-I-8-b - Les tests

Sous Unix, chaque processus reçoit plusieurs valeurs en paramètres et renvoie un code de retour. La plupart des paramètres sont passés en ligne de commande, et sont récupérés directement par le processus, mais d'autres paramètres peuvent être fournis par le processus appelant par l'intermédiaire de variables d'environnement et de descripteurs de fichiers. Le code de retour, quant à lui, est un entier signalant si l'exécution du processus s'est terminée correctement ou si des erreurs ont eu lieu. Si les codes d'erreurs varient grandement d'un programme à un autre, la valeur 0 signifie toujours, et ce quel que soit le programme, que l'exécution s'est déroulée correctement.

Il est possible de tester le code de retour d'une commande avec l'instruction `if`. La syntaxe la plus simple pour un test est la suivante :

```
if commande ; then
    action
fi
```

où `commande` est la commande dont on désire tester le code de retour, et `action` est la commande à exécuter si ce code vaut 0 (c'est-à-dire, si la commande `commande` s'est exécutée correctement).

Il peut paraître réducteur de ne pouvoir tester que le code de retour d'une commande. Mais en fait, c'est une fonctionnalité très puissante du shell, car elle permet de réaliser tous les types de tests imaginables. En effet, il existe une commande spéciale, `[`, qui permet de réaliser divers types de tests sur les paramètres qu'on lui passe, et d'ajuster son code d'erreur en conséquence. Par exemple, pour tester l'égalité d'une variable d'environnement avec une chaîne de caractères, on utilisera la syntaxe suivante :

```
if [ $variable == valeur ] ; then
    action
fi
```

Notez que dans cette syntaxe, le test effectué est une commande complète. Cela implique qu'il faut mettre une espace entre chaque paramètre, et en particulier entre le nom de la commande (`[`), le premier opérande (`$variable`), l'opérateur utilisé (`==`), le deuxième opérande (`valeur`) et le caractère de marque de fin de test (`]`).

La commande `[` est capable d'effectuer tous les tests standards. Par défaut, elle considère que les deux opérandes du test sont des chaînes de caractères, et elle utilise l'ordre lexicographique pour les comparer. Les tests d'égalité et d'inégalité sont effectués respectivement avec les opérateurs `==` et `!=`. Les opérateurs d'antériorité dans l'ordre lexicographique sont `<` et `<=`, et les opérateurs de postériorité sont `>` et `>=`. Notez que l'utilisation de ces opérateurs peut être relativement pénible, parce que les caractères `<` et `>` sont interprétés par le shell en tant que redirections. Par conséquent, il faut souvent les précéder du caractère d'échappement antislash.

L'ordre lexicographique convient dans la plupart des cas, mais il n'est pas très approprié pour la comparaison de valeurs numériques. Par exemple, le test suivant :

```
if [ -1 \< -2 ] ; then
    echo "-1 est plus petit que -2"
fi
```

est vérifié, car le caractère `1` précède le caractère `2` dans l'ordre lexicographique. La commande `[` fournit donc la possibilité d'utiliser une autre syntaxe pour comparer les entiers. Cette syntaxe utilise les options `lt` et `gt` respectivement pour les tests d'infériorité stricte et de supériorité stricte, et les options `le` et `ge` respectivement pour les tests d'infériorité et de supériorité ou d'égalité. Ainsi, le test :

```
if [ $i -gt 3 ] ; then
    echo "$i est supérieur à 3"
fi
```

permet de comparer la valeur entière de la variable `i` avec le nombre 3.

Nous avons vu dans la [Section 5.9.8.1](#) que les opérateurs `||` et `&&` permettent de tester le code de retour d'une commande, et qu'en fonction de la valeur de ce code de retour, d'exécuter ou non la commande suivante. La syntaxe de ces opérateurs provient en fait de la possibilité de les employer pour effectuer des tests complexes avec l'instruction `if`. Par exemple, pour effectuer un ET logique entre deux tests, on utilisera la syntaxe suivante :

```
if [ $i == "A" ] && [ $j -lt 3 ] ; then
    echo "i contient la lettre \"A\" et j contient un nombre inférieur à 3"
fi
```

Notez que la deuxième commande `[` n'est exécutée que si le premier test est vérifié. L'utilisation de l'opérateur `||` se fait selon le même principe. Il est bien entendu possible de regrouper plusieurs commandes de test ensemble, à l'aide de parenthèses.

Comme dans la plupart des langages informatiques, l'instruction `if` peut prendre une forme plus complexe pour traiter les cas où le test n'est pas vérifié. Ainsi, pour exécuter une action spécifique pour le cas où le test serait faux, on peut utiliser la syntaxe suivante :

```
if commande ; then
    action1
else
    action2
fi
```

où `commande` est toujours la commande dont le code de retour sera testé, `action1` est l'action qui doit être réalisée si cette commande a renvoyé le code de retour 0, et `action2` la commande à exécuter dans le cas contraire. De même, si l'on veut enchaîner des tests, on utilisera le mot clé `elif`. La syntaxe générale du test est donc la suivante :

```
if commande1 ; then
    action1
elif commande2 ; then
    action2
elif commande3 ; then
    &#8230;
else
    actionn
fi
```

Note : Pour des raisons d'optimisation, le shell peut simuler le comportement du programme `[`, et éviter ainsi de le lancer à chaque fois qu'il a à faire un test. Cependant, le principe originel était bien celui décrit ci-dessus. Cette description, bien que n'étant plus tout à fait exacte, permet de mieux comprendre la syntaxe du shell.

Il est possible de récupérer la valeur du code de retour de la dernière commande exécutée grâce à la variable spéciale `$?`. Cependant, il est très rare d'avoir à manipuler cette valeur directement, car les structures de contrôle du shell telles que `if` permettent d'effectuer les actions qui s'imposent sans avoir à la connaître.

Pour ceux qui savent programmer en C, sachez que le code de retour est la valeur renvoyée par la fonction C `exit` ou par l'instruction `return` de la fonction principale `main`. Les paramètres de la ligne de commande, quant à eux, sont récupérables par l'intermédiaire des paramètres de la fonction principale `main`.

Il ne faut pas oublier que la fonction première du shell est de permettre les manipulations de fichiers. Il n'est donc pas étonnant que la commande `[` permette également de réaliser tous les tests imaginables sur les fichiers. Ces tests vont de l'existence d'un fichier au test de sa nature et de ses attributs, en passant par les tests sur l'identité de son propriétaire et de son groupe. La syntaxe générale de ces tests est la suivante :

```
if [ option fichier ] ; then
    .
    .
    .
```

fi

où option est une option de la commande [décrivant la propriété testée, et fichier est le nom du fichier sur lequel le test doit porter.

Les principales options utilisables dans les tests sur les fichiers sont récapitulées dans le tableau ci-dessous :

Tableau 5-4. Tests sur les fichiers

Option	Signification
-e	Test d'existence d'un fichier ou d'un répertoire.
-d	Test d'existence d'un répertoire.
-f	Test d'existence d'un fichier normal.
-s	Test d'existence d'un fichier et vérification que sa taille est non nulle.
-L	Test d'existence d'un lien symbolique.
-b	Test d'existence d'un fichier spécial de périphérique de type bloc (disque dur, CD-ROM, lecteur de cassettes, etc.).
-c	Test d'existence d'un fichier spécial de périphérique de type caractère (port série, port parallèle, carte son...).
-p	Test d'existence d'un tube.
-r	Test d'existence du fichier et d'accessibilité en lecture de ce fichier.
-w	Test d'existence du fichier et d'accessibilité en écriture de ce fichier
-x	Test d'existence du fichier et de possibilité d'exécution de ce fichier.
-g	Test d'existence du fichier et de présence du bit setgid sur ce fichier.
-u	Test d'existence du fichier et de présence du bit setuid sur ce fichier
-k	Test d'existence du fichier et de présence du bit sticky sur ce fichier.
-O	Test d'existence du fichier et d'appartenance de ce fichier à l'utilisateur effectif courant.
-G	Test d'existence du fichier et d'appartenance de ce fichier au groupe effectif courant.
-N	Test d'existence du fichier et de modification de ce fichier depuis la dernière fois qu'il a été lu.

Note : Ce tableau n'est pas exhaustif, mais les options les plus importantes et les plus utilisées s'y trouvent.

Vous pourrez vous rafraîchir la mémoire sur les notions de bit setuid, setgid et sticky, ainsi que sur les notions d'utilisateur et de groupe effectifs en relisant la [Section 4.2](#).

La commande [accepte également les options -nt et -ot, qui permettent respectivement de tester si un fichier est plus récent ou plus vieux qu'un autre, en se basant sur les dates de dernière modification de ces fichiers. Ces deux opérateurs s'utilisent avec la syntaxe suivante :

```
if [ fichier1 option fichier2 ] ; then
    .
    .
    .
fi
```

où `fichier1` et `fichier2` sont les deux fichiers sur lesquels la comparaison doit porter, et `option` est l'une des options `-nt` ou `-ot`.

V-I-8-c - Le branchement conditionnel

Lorsqu'on veut effectuer différentes opérations selon la valeur d'une variable, l'instruction `if` peut devenir très lourde à utiliser. En effet, si le nombre de valeurs différentes est grand, elle peut conduire à écrire un grand nombre de tests. Le shell fournit donc une instruction de branchement conditionnel, qui permet de spécifier quelle action doit être prise pour chaque valeur de la variable.

Le branchement conditionnel s'utilise de la manière suivante :

```
case valeur in
    ( motif1 | motif2 | &#8230; ) commande1 ;;
    ( motifn | motifn+1 | &#8230; ) commande2 ;;
    .
    .
    .
esac
```

où `motif1`, `motif2...` `motifn+1` sont des motifs spécifiant les valeurs possibles pour la valeur `valeur`, et `commande1`, `commande2`, etc. sont les commandes à exécuter pour les valeurs de ces motifs.

La commande exécutée est la première commande pour laquelle la variable correspond à l'un de ses motifs correspondants. Une fois exécutée, la recherche se termine, et l'exécution reprend à la suite du branchement conditionnel. Par exemple ce branchement conditionnel :

```
case $i in
    ( *.txt ) echo "$i est un fichier texte" ;;
    ( *.gz ) echo "$i est compressé avec gzip" ;;
    ( *.tar ) echo "$i est une archive" ;;
esac
```

affiche la nature du fichier dont le nom est stocké dans la variable `i` à partir de son extension.

Le code de retour du branchement conditionnel est 0 si la variable ne correspond à aucun des motifs, ou le code de retour de la commande exécutée sinon.

V-I-8-d - Les boucles

Il existe deux types de boucles : le `while` et le `until`. La syntaxe des boucles `while` est la suivante :

```
while commande ; do
    action
done
```

où `commande` est une commande dont le code de retour est utilisé comme critère de la fin de la boucle, et `action` est l'instruction (composée ou non) exécutée à chaque itération de la boucle. Comme on le voit, le shell utilise le même principe pour les boucles que pour les tests pour évaluer une condition. Tant que la commande `commande` renvoie un code de retour égal à 0, l'instruction `action` est exécutée.

L'instruction `until` utilise la même syntaxe que l'instruction `while` :


```
until commande ; do
    action
done
```

à ceci près que l'instruction `action` est exécutée tant que la commande `commande` renvoie un code de retour non nul. L'instruction `until` utilise donc simplement le test inverse de celui de l'instruction `while`.

Bien entendu, il est possible d'utiliser la commande `[` pour effectuer des tests plus complexes que le simple test du code de retour d'une commande. Par exemple, la boucle suivante calcule la somme des dix premiers entiers :

```
result=0
i=0
while [ $i -le 10 ] ; do
    result=$((result + $i))
    i=$((i + 1))
done
echo $result
```

V-I-8-e - Les itérations

Les itérations sont des boucles qui s'exécutent pour chaque élément d'un ensemble donné. Le shell gère les itérations par l'intermédiaire de l'instruction `for`. La syntaxe de cette instruction est la suivante :

```
for variable [ in ensemble ] ; do
    action
done
```

où `variable` est un nom de la variable utilisée pour l'itération, `ensemble` est l'ensemble des valeurs que peut prendre cette variable, et `action` est la commande (simple ou composée) à exécuter pour chaque valeur de cette variable.

Le principe des itérations est très simple. Pour chaque valeur indiquée dans l'ensemble des valeurs, la commande est exécutée, avec la valeur en question accessible dans la variable utilisée pour l'itération. Par exemple, la commande suivante :

```
for i in *.txt ; do
    mv $i ${i%.txt/.doc}
done
```

permet de renommer tous les fichiers portant l'extension `.txt` en fichier du même nom, mais avec l'extension `.doc`.

Il n'est pas nécessaire de préciser l'ensemble des valeurs que peut prendre la variable. Dans ce cas, l'ensemble utilisé sera celui de tous les paramètres du script ou de la fonction. Nous verrons plus loin comment réaliser des fonctions et des scripts, ainsi que la manière de récupérer leurs paramètres.

V-I-8-f - Les ruptures de séquence

Il est parfois nécessaire de modifier l'ordre d'exécution dans les boucles et les itérations du shell. Par exemple, il est souvent nécessaire de sortir de la boucle courante, soit parce qu'on ne peut plus la continuer dans de bonnes conditions, soit parce que le traitement est terminé. C'est notamment le cas lorsqu'une erreur se produit, ou lorsqu'on recherche une valeur spécifique en itérant sur les valeurs possibles d'un ensemble.

Le shell fournit donc les instructions `break` et `continue`, qui permettent respectivement de sortir de la boucle courante et de passer directement à l'itération suivante. Ces deux commandes peuvent être utilisées aussi bien à l'intérieur des boucles `while` et `until` que dans les itérations écrites avec l'instruction `for`. Par exemple, le calcul de la somme des dix premiers entiers aurait pu être écrit de la manière suivante :

```
result=0
```

```
i=0
while true ; do
    result=$((result + $i))
    i=$((i + 1))
    if [ $i == 11 ] ; then break ; fi
done
echo $result
```

Les instructions `break` et `continue` peuvent prendre un paramètre entier indiquant le niveau d'imbrication de la boucle sur laquelle elles s'appliquent. Ce paramètre doit impérativement être supérieur à sa valeur par défaut, c'est-à-dire 1. Ainsi, pour sortir directement d'une double boucle lorsqu'on est dans le corps de la boucle la plus imbriquée, on utilisera la commande suivante :

```
break 2
```

V-I-8-g - Les fonctions

Le langage du shell est un langage procédural. Cela signifie que l'on peut créer des fonctions pour regrouper des séries d'instructions couramment exécutées. La syntaxe permettant d'écrire de telles fonctions est la suivante :

```
function nom () {
    instructions
}
```

où `nom` est le nom de la fonction, et `instructions` est la liste des commandes à exécuter dans cette fonction.

Vous constaterez qu'il n'y a pas de déclaration des paramètres de cette fonction. C'est normal : les paramètres des fonctions sont passés implicitement dans les variables d'environnement `$1`, `$2`, `$3`, etc. En fait, comme nous le verrons [plus loin](#), cette syntaxe est également celle utilisée pour récupérer les paramètres de la ligne de commande des scripts shell. Cela signifie que les paramètres du script ne sont pas accessibles dans le corps d'une fonction, puisqu'ils sont masqués par les paramètres de la fonction.

Les autres variables utilisées dans les fonctions sont des variables globales. Celles qui sont déclarées dans une fonction sont donc également globales, et restent accessibles même après l'exécution de cette fonction. Si l'on veut définir des variables locales, on précédera la définition de la variable du mot clé `local` :

```
local variable=valeur
```

où `variable` est le nom de la variable locale, et `valeur` est sa valeur.

Les fonctions peuvent retourner une valeur numérique en code de retour. Cette valeur peut être indiquée à l'aide de l'instruction `return`. Par exemple, la fonction suivante calcule la somme des entiers de 0 à la valeur de l'entier qu'elle reçoit en paramètre :

```
function somme () {
    local result=0
    local i=0
    while [ $i -le $1 ] ; do
        result=$((result + $i))
        i=$((i + 1))
    done
    return $result
}
```

Ce code d'erreur pourra être récupéré par l'appelant dans la variable d'environnement `$?` :

```
somme 10
echo $?
```

V-I-8-h - Les entrées / sorties de données

Tout langage de programmation qui se respecte dispose de possibilités d'entrée / sortie pour permettre la communication avec l'utilisateur de manière interactive, et le shell n'échappe pas à la règle.

Nous avons déjà vu la commande `echo` dans bon nombre des exemples qui précédaient, et vous avez sans doute deviné qu'il s'agissait là de la commande qui permet d'afficher du texte à l'écran. Son utilisation est des plus simples, puisqu'elle se contente d'envoyer sur le flux de sortie standard une chaîne de caractères contenant tous les paramètres qu'elle reçoit, séparés par des espaces. Nous ne nous attarderons donc pas sur cette commande, qui n'a pas dû vous poser de problèmes jusqu'à présent.

Il ne nous reste donc plus qu'à voir la manière de demander à l'utilisateur de saisir une valeur. Avec `bash`, la demande de saisie des données se fait classiquement à l'aide de la commande `read`. Cette commande lit une ligne sur le flux d'entrée standard, la découpe en une ou plusieurs données et place les résultats dans les variables d'environnement qu'elle reçoit en paramètre. La syntaxe de `read` est donc la suivante :

```
read variable1 variable2 &#8230; variablen
```

où `variable1`, `variable2`, etc. sont les noms des variables d'environnement dans lesquelles les résultats de la saisie doivent être placés.

La commande `read` utilise les séparateurs indiqués dans la variable d'environnement `IFS` pour découper la ligne lue dans le flux d'entrée standard. Si le nombre de variables spécifié est inférieur au nombre de mots de cette ligne après découpage, les premières variables d'environnement reçoivent les premiers mots, et la dernière reçoit le reste de la commande. Par exemple, la commande suivante :

```
read MOT RESTE
```

permet de lire le premier mot d'une ligne dans la variable d'environnement `MOT` et de placer le reste dans la variable `RESTE`.

La commande `read` dispose d'une syntaxe simplifiée, qui ne prend aucun paramètre. Dans ce cas, la ligne lue dans le flux d'entrée standard est placée telle quelle dans la variable d'environnement `REPLY`. Il est à la charge du programmeur d'analyser son contenu.

Le shell dispose également d'une instruction évoluée permettant de réaliser des menus simplifiés : l'instruction `select`. Cette instruction construit un menu à partir d'un certain nombre de choix, chaque choix étant précédé par un numéro, et demande à l'utilisateur de taper le numéro de son choix. Elle affecte alors la valeur du choix correspondant à une variable d'environnement, et exécute une commande pour le traitement du choix. La syntaxe générale de l'instruction `select` est donnée ci-dessous :

```
select variable in liste ; do
    action
done
```

où `variable` est le nom de la variable devant recevoir la valeur choisie par l'utilisateur, `liste` est la liste des valeurs que cette variable peut prendre, et `action` est la liste des instructions à exécuter pour chaque choix effectué.

Si le choix de l'utilisateur est incorrect, la variable de contrôle reçoit la valeur nulle. Le programmeur peut récupérer la valeur saisie par l'utilisateur dans la variable d'environnement `REPLY` et effectuer un traitement d'erreur approprié.

L'instruction `select` est une boucle. Le menu est reproposé après chaque exécution de l'action `action`. La sortie de cette boucle ne peut se faire que si un caractère de fin de fichier (CTRL + D) est lu sur le flux d'entrée standard, ou si une option de menu spécifique est proposée pour quitter cette boucle. Vous trouverez un exemple de menu simplifié ci-dessous :

```
select LU in A B C D Sortir; do
  case $LU in
    ("A") echo "Vous avez choisi A" ;;
    ("B") echo "Vous avez choisi B" ;;
    ("C") echo "Vous avez choisi C" ;;
    ("D") echo "Vous avez choisi D" ;;
    ("Sortir") break ;;
  esac
done
```

V-I-9 - Les alias

Il est incontestable que certaines commandes peuvent avoir une grande complexité, et il peut être fastidieux de les retaper complètement à chaque fois que l'on en a besoin. D'autre part, la saisie d'une longue ligne de commande multiplie les risques de faire une faute de frappe et d'avoir à corriger la commande. Cela peut au mieux faire perdre son temps à l'utilisateur, et au pire l'énerver.

Le shell fournit donc un mécanisme pour donner un nom simplifié aux commandes complexes : le mécanisme des alias. Les alias représentent en fait des chaînes de caractères complexes, et sont remplacés automatiquement par le shell lorsqu'il analyse les lignes de commandes. C'est un mécanisme plus souple que celui des variables d'environnement, et qui permet de définir des macro-commandes plus facilement qu'avec les fonctions du shell.

Pour créer un alias, vous devrez utiliser la syntaxe suivante :

```
alias nom=chaîne
```

où `nom` est le nom de l'alias, et `chaîne` est la chaîne de caractères représentée par cet alias. Par exemple, pour faire un alias nommé `beep` permettant de faire un bip sonore, on pourra utiliser la commande suivante :

```
alias beep="echo $'\a'"
```

Cet alias pourra être utilisé simplement en tapant `beep` en ligne de commande.

Vous pouvez visualiser la liste des alias existant simplement à l'aide de la commande `alias`, appelée sans paramètres. Je vous recommande de consulter cette liste, pour vous donner une idée des alias courants, qui se révèlent généralement très utiles.

La suppression des alias se fait à l'aide de la commande `unalias`. Sa syntaxe est la suivante :

```
unalias nom
```

où `nom` est le nom de l'alias à supprimer.

Note : Les alias ne sont remplacés par la chaîne de caractères qu'ils représentent que lorsque le shell analyse la ligne de commande. Cela signifie que les définitions d'alias ne sont valides qu'après validation de cette ligne. On évitera donc de définir des alias dans la déclaration d'une instruction composée, car cet alias ne sera pas disponible à l'intérieur de son propre bloc d'instructions.

Par défaut, les alias ne sont disponibles que dans les shells interactifs. Ils ne peuvent donc pas être utilisés dans les scripts shell. La notion de script shell est détaillée dans la [Section 5.9.10](#).

V-I-10 - Les scripts shell

Pour l'instant, toutes les fonctionnalités de `bash`, aussi puissantes soient-elles, ne constituent que l'interface d'un interpréteur de commandes puissant. Mais nous avons dit que le shell était véritablement un langage de

programmation. Cela signifie qu'il est possible d'écrire des programmes complexes en langage shell, simplement en stockant plusieurs commandes dans un fichier. On appelle ces fichiers des scripts shell.

L'écriture d'un script shell n'est pas plus compliquée que de taper les commandes du programme les unes à la suite des autres dans un shell interactif. La seule différence est que les scripts shell peuvent être rejoués plusieurs fois, recevoir des paramètres en ligne de commande et renvoyer un code de retour.

Tout script shell est en fait un fichier texte sur lequel on a mis les droits d'exécution. Il contient les différentes commandes qu'il doit exécuter. Sa première ligne est très importante, elle permet d'indiquer au shell exécutant quelle est la nature du fichier. La syntaxe de cette ligne est la suivante :

```
#!/shell
```

où shell est le chemin absolu sur le shell ou l'interpréteur de commande capable d'exécuter ce script. En pratique, pour bash, on utilisera toujours la ligne suivante :

```
#!/bin/bash
```

Les paramètres des scripts shell sont accessibles exactement comme des paramètres de fonction. On récupérera donc le premier paramètre avec l'expression \$1, le deuxième avec l'expression \$2, le troisième avec l'expression \$3, etc.

Le code de retour d'un shell pourra être fixé à l'aide de la commande exit. Par exemple :

```
exit 0
```

Ce code de retour pourra être récupéré par l'appelant à l'aide de l'expression \$?.

Nous n'irons pas plus loin dans la description du shell bash, car ce n'est pas le but de ce document. Vous pouvez vous référer à un bon livre d'Unix ou aux pages de manuel si vous désirez approfondir le sujet. Comme vous avez dû vous en rendre compte dans cette section, les shells Unix sont des véritables langages de programmation, qui dépassent de très loin les interpréteurs de commandes du type DOS. De plus, il existe plusieurs autres langages dont nous n'avons pas parlé ici, chacun étant conçu souvent pour réaliser un certain type de tâche (administration système, manipulation de fichiers textes, création de pages Web dynamiques, création d'interfaces utilisateur en mode fenêtré, pilotage d'applications, etc.). Si vous vous y intéressez, vous verrez que le sujet est réellement vaste et passionnant.

VI - Administration du système de base

Un certain nombre d'opérations que l'on peut faire avec un système Unix ne rentrent pas dans le cadre d'une utilisation quotidienne, mais sont destinées plutôt à l'administration du système lui-même. Ces opérations peuvent être réalisées à l'aide de commandes Unix spéciales, généralement réservées à l'administrateur du système, ou peuvent être réalisées en modifiant les fichiers de configuration du système.

Il est très probable que le programme d'installation ou le programme de configuration de votre distribution vous permette d'effectuer ces tâches de manière relativement aisée ou conviviale. L'utilisation de ces programmes est très simple, puisqu'en général il suffit de répondre à quelques questions et les modifications sont effectuées automatiquement pour vous. Il est fortement recommandé de toujours essayer les programmes de ce type en premier lieu, car eux seuls connaissent les spécificités de chaque distribution. Cela dit, ces programmes ne peuvent pas tout prévoir, parce que Linux est un système capable d'effectuer un grand nombre de tâches très diversifiées d'une part, et parce que ce que vous voulez en faire personnellement ne correspond pas forcément à un standard prédéterminé d'autre part.

Cette partie décrira donc les commandes d'administration et de maintenance les plus importantes et le mécanisme général d'amorçage des systèmes Linux. Les principaux fichiers de configuration permettant de modifier le comportement du système seront également décrits afin de permettre un usage courant de Linux dans de bonnes

conditions. Les notions les plus avancées concernant l'administration système ne seront en revanche pas abordées, car cela dépasserait le cadre de ce document. Les lecteurs les plus intéressés pourront toujours se référer à un guide d'administration Unix.

L'administration du système est un peu moins sensible que son installation. En effet, les seuls risques que l'on encourt sont de détruire les fichiers de configuration du système, et donc de devoir les recréer manuellement. Il n'y a pas de manipulation de partitions ou de système de fichiers à créer, aussi le risque de perdre des données est-il nettement plus faible. Cependant, les opérations d'administration se feront sous le compte root, ce qui implique une prudence extrême. C'est pour cette raison que nous allons commencer par sauvegarder l'ensemble des fichiers de configuration, afin de pouvoir revenir à l'état initial après installation, sans repasser par la case départ.

VI-A - Sauvegarde de la configuration d'installation

La sauvegarde de la configuration du système est une opération facile à réaliser. En effet, tous les fichiers de configuration sont placés dans le répertoire `/etc/`. Par conséquent, il suffit de faire une archive des fichiers de ce répertoire et de ses sous-répertoires. Cette opération peut être réalisée avec la commande suivante :

```
tar cvfz /root/install.conf.tar.gz /etc/*
```

Cette commande créera une archive nommée `install.conf.tar.gz` dans le répertoire personnel de l'administrateur système. On notera que, pour certaines distributions, quelques fichiers de configuration sont placés dans le répertoire `/sbin/init.d/`. Pour ces distributions, on utilisera donc plutôt la commande suivante :

```
tar cvfz /root/install.conf.tar.gz /etc/* /sbin/init.d/*
```

De cette manière, si l'on a un gros problème avec la configuration de la machine, on peut revenir simplement à la configuration utilisée juste après l'installation du système avec la simple commande suivante :

```
tar xvfz /root/install.conf.tar.gz
```

commande que l'on exécutera dans la racine du système de fichiers.

Cette commande écrasera tous les fichiers existants par ceux de la sauvegarde. Les fichiers qui ont été ajoutés depuis cette sauvegarde seront bien entendu conservés.

Il est également recommandé de faire une sauvegarde identique à celle-ci une fois que l'on aura réussi à configurer le système correctement et que, théoriquement, il n'y aura plus à toucher aux fichiers de configuration. Cette sauvegarde devra être placée sur une disquette ou un support amovible que l'on conservera en lieu sûr.

VI-B - Mise à l'heure du système

Les systèmes d'exploitation utilisent l'heure pour un certain nombre de tâches. En particulier, les fichiers disposent de plusieurs dates (date de création, date d'accès et date de dernière modification), qui sont utilisées par différents programmes. Les programmes de sauvegarde en font évidemment partie, parce qu'ils se basent sur les dates de modification des fichiers pour déterminer quels sont les fichiers qui doivent être sauvegardés depuis la dernière sauvegarde (cas des sauvegardes dites « incrémentales »). Les programmes de maintenance sont également lancés à des dates précises, et les applications normales des utilisateurs peuvent utiliser la date système pour l'intégrer dans leurs documents. En clair, il est important que votre système soit à l'heure.

En fait, il existe deux horloges dans votre système. La première horloge, qui est l'horloge de référence pour toutes les opérations effectuées dans le système, est l'horloge dite « système ». Cette horloge est maintenue par le noyau grâce à un compteur qui est incrémenté régulièrement, sur la base d'une interruption matérielle. La précision de ce compteur est a priori la même que celle de l'interruption du timer matériel. Sur les PC, cette interruption a lieu, par défaut, 18,6 fois par seconde, ce qui donne pour la plus petite unité de temps mesurable environ 1/20 de seconde (cependant, la plupart des systèmes reprogramment la fréquence de cette interruption à 100 Hz environ). La deuxième horloge est

l'horloge matérielle, qui est l'horloge qui maintient l'heure de votre ordinateur pendant qu'il est éteint. Cette horloge est couramment appelée l'horloge CMOS, parce qu'elle est gérée par un composant CMOS qui stocke toutes les informations permanentes du BIOS.

Pour répondre immédiatement à une question (désormais sans objet), précisons que Linux n'a aucun problème vis-à-vis des dates critiques du changement de millénaire. En effet, les systèmes Unix n'utilisent qu'un seul format de date au niveau application : le nombre de secondes écoulées depuis le 01/01/1970 à 0 heure. Ce compteur est stocké sur 32 chiffres binaires sur la plupart des machines et passe donc allègrement le cap de l'an 2000. En fait, le débordement de ce compteur est prévu pour 2038, mais n'aura jamais lieu, car l'apparition des processeurs 64 bits va porter, d'ici là, la taille de ce compteur à 64 bits. Cela étant, il est possible que certaines applications mal écrites n'utilisent pas ce format de date, et ne soient donc pas compatibles. Heureusement, ce cas de figure est très rare sous Unix. Bien entendu, le problème reste entier si l'horloge matérielle de votre PC n'est pas compatible. Dans ce cas, la solution la plus simple est de régler l'heure système à chaque démarrage, manuellement ou à l'aide de scripts de correction de la date renvoyée par l'horloge matérielle.

La valeur du compteur de l'horloge système est toujours interprétée en temps universel (« UTC » en anglais, abréviation de « Universal Time Coordinated »), c'est-à-dire le temps de référence valide dans le monde entier. Ce temps ne comprend pas les fuseaux horaires ni les réglementations concernant les heures d'hiver et d'été. Cette convention est utilisée partout dans le système, ce qui est la condition sine qua non pour que tous les ordinateurs du monde utilisent la même date et la même heure. Ainsi, deux ordinateurs connectés à Internet peuvent communiquer sans se poser de questions quant à leurs localisations respectives, ce qui simplifie beaucoup les choses. Notez également que le fait de compter le temps en secondes permet de s'affranchir des conventions de découpage du temps et des calendriers utilisés dans chaque pays.

Bien entendu, les dates présentées à l'utilisateur doivent être traduites en temps local, corrigé des écarts pour l'heure d'été et l'heure d'hiver. Cela est réalisé par tous les programmes qui doivent afficher ces dates (par exemple, les simples commandes `ls` et `date`). Cette conversion est effectuée par le système en fonction du fuseau horaire et des plages de validité des horaires d'été et d'hiver.

La solution la plus simple pour régler la date et l'heure de votre machine est donc de régler l'horloge matérielle sur le temps universel, et de définir le fuseau horaire dans lequel elle se trouve, pour que le système puisse calculer l'heure locale. Malheureusement, les systèmes d'exploitation de Microsoft ne voient pas la chose de la même manière. Ils attendent que l'horloge matérielle soit réglée à l'heure locale. Par conséquent, si Linux est installé sur un ordinateur disposant déjà de Windows, vous devrez régler l'heure de votre ordinateur en temps local. A priori, cela ne fait aucune différence, le système étant également capable de calculer le temps universel à partir de l'heure locale et de la zone horaire. Cependant, cela a un inconvénient : il est nécessaire de mettre à l'heure l'horloge système en cas de déplacement de la machine, et à chaque changement d'horaire d'été ou d'hiver. Bien sûr, Windows est supposé être capable de mettre à jour l'heure matérielle en « observation avec l'heure d'été / d'hiver ». Mais il utilise pour cela des règles qui sont fixées définitivement dans le système et qui ne peuvent pas être mises à jour avec les réglementations locales (par exemple, la règle de changement d'heure a été modifiée en 1996, si bien que Windows 95 n'a jamais pu fonctionner correctement sur ce point...).

Quoi qu'il en soit, la mise à l'heure d'un système Linux requiert la définition de la zone horaire, la mise à l'heure du système et la mise à l'heure de l'horloge matérielle. La définition de la zone horaire est primordiale et doit avoir lieu avant toute autre opération, car le réglage des horloges dépend évidemment de cette zone.

Les zones horaires sont définies par un ensemble de règles, qui comprennent chacune la période de validité de la règle (en général avec une date de départ et une date de fin) et la différence entre le temps universel et le temps local lorsque cette règle s'applique (gestion des horaires d'été et d'hiver compris). Toutes ces règles portent le nom de la zone géographique dans laquelle elles sont valides. Vous pourrez trouver des exemples de définitions de règles (ainsi que l'historique des conventions concernant le temps) dans le répertoire « timezone » des sources de la bibliothèque C GNU.

Les fichiers de règles des zones horaires doivent être compilés avec le programme `zic` et installés dans le répertoire `/usr/share/zoneinfo`. Normalement, votre système dispose de la totalité des règles, déjà compilées, des différentes zones horaires du monde. Le programme `zic` permet également de définir la zone horaire active. Cette opération se fait dans les fichiers de démarrage de votre système, avec une commande similaire à la suivante :


```
zic -l zone
```

où zone est le chemin relatif du fichier de définition des règles de la zone horaire locale, par rapport au répertoire de base /usr/share/zoneinfo. Pour les systèmes situés en France métropolitaine, la commande utilisée est donc celle-ci :

```
zic -l Europe/Paris
```

Une fois la zone horaire fixée, il est possible de régler l'horloge système. Il existe deux solutions pour cela. La première solution est d'utiliser la commande système date. Cette commande, appelée sans paramètres, permet d'obtenir la date système, exprimée en temps local. Mais elle permet également de modifier la date et l'heure système avec l'option -s. La syntaxe complète utilisée est donnée ci-dessous :

```
date -s "MM/JJ/AAAA HH:MM:SS"
```

Il n'est pas nécessaire de préciser l'année si celle-ci ne doit pas être changée. De même, vous pouvez ne donner que l'heure, si la date du jour est correcte. En revanche, vous devez obligatoirement préciser l'heure si vous changez la date. Notez que l'heure doit être donnée en temps local, à moins que l'option -u ne soit précisée. Le système réglera son horloge en temps universel automatiquement, selon les règles de zones horaires en vigueur qui ont été indiquées par zic. Vous pouvez obtenir l'heure exacte en appelant le 3699.

La deuxième solution est celle qui est utilisée au démarrage du système. Elle consiste à initialiser l'horloge système à partir de l'horloge matérielle. Cette opération se fait normalement à l'aide de la commande clock (qui en fait est un lien symbolique vers hwclock, mais la commande Unix traditionnelle est clock). La syntaxe de cette commande est la suivante :

```
clock [-u] -s | -w | -a
```

L'option -s permet d'initialiser l'horloge système à partir de la date et de l'heure stockées dans l'horloge matérielle. C'est typiquement cette commande qui est utilisée dans les scripts de démarrage du système. L'option -w permet de réaliser l'opération inverse, c'est-à-dire sauvegarder la date et l'heure de l'horloge système dans l'horloge matérielle. Elle n'est en général utilisée qu'après avoir remis à l'heure l'horloge système. L'option -a permet, quant à elle, de corriger l'avance ou le retard que l'horloge matérielle peut prendre.

Ce dernier point mérite quelques explications complémentaires. En fait, l'horloge matérielle n'est pas extrêmement précise, et peut se décaler petit à petit de l'heure réelle. Heureusement, ce décalage est constant, ce qui fait qu'il est possible de le mesurer et de le prendre en compte. Le programme clock utilise le fichier /etc/adjtime pour enregistrer de combien est ce décalage afin de pouvoir effectuer les corrections. Le principe de fonctionnement est le suivant :

- lors du premier réglage de l'horloge matérielle (avec l'option -w), il enregistre l'instant de ce réglage dans le fichier /etc/adjtime ;
- lors des réglages suivants, il calcule le temps qui s'est écoulé depuis le réglage précédent, et le décalage entre l'heure de l'horloge matérielle et l'heure à laquelle celle-ci aurait dû se trouver. Il enregistre ce décalage et met à jour la date de mise à l'heure (pour pouvoir refaire ce calcul ultérieurement) ;
- lorsqu'on l'appelle avec l'option -a, clock ajuste l'horloge matérielle. Pour cela, il regarde la date courante, calcule le temps écoulé depuis la dernière mise à l'heure ou le dernier ajustement, en déduit l'avance ou le retard de l'horloge matérielle, et la remet à l'heure en conséquence. Il enregistre également la date de cet ajustement comme nouvelle date de mise à l'heure, afin de ne pas faire deux fois l'ajustement pour cette période la prochaine fois.

De cette manière, il est possible de maintenir l'horloge système à une valeur proche de la réalité (sans ce genre de mécanisme, il est courant de prendre 5 minutes d'écart en trois ou quatre mois, ce qui est déjà considérable).

Les scripts d'initialisation de votre système doivent donc certainement contenir au moins les deux lignes suivantes après le réglage de la zone horaire :

```
# Ajuste l'horloge matérielle :
```

```
clock -a
# Initialise l'horloge système :
clock -s
```

Dans tous les cas, l'option -u permet d'indiquer que l'horloge matérielle est réglée en temps universel. Si votre machine ne dispose pas d'autre système que Linux, il est recommandé de procéder ainsi et d'utiliser systématiquement cette option.

Note : Il est important de définir la zone horaire avec zic avant d'utiliser clock. En effet, si l'horloge matérielle est réglée en temps local, clock ne pourra pas déterminer l'heure en temps universel. D'autre part, clock initialise la structure de zone horaire interne noyau, que celui-ci utilise notamment pour l'écriture des dates en temps local sur les systèmes de fichiers FAT (Eh oui, les dates des fichiers des systèmes de fichiers FAT sont enregistrées en temps local...).

Sachez également que l'horloge système peut également se décaler sensiblement sur de longues périodes. Évidemment, ce phénomène ne peut se détecter que si le système reste actif suffisamment longtemps, ce qui en pratique ne se produit que dans les serveurs (n'oubliez pas que Linux peut fonctionner des mois sans interruption...). Si vous êtes intéressé par la manière de resynchroniser l'horloge système pour de telles configurations, vous devriez vous intéresser à la diffusion du temps sur le réseau Internet avec le protocole NTP (« Network Time Protocol »). En général, la resynchronisation de l'heure système doit se faire progressivement afin de ne pas perturber la ligne du temps pour les applications. Cela peut être fait avec le programme adjtimex.

VI-C - Gestion des utilisateurs et de la sécurité

La règle de sécurité numéro un sous Unix est de ne jamais travailler dans le compte root. En effet, ce compte dispose de tous les droits, et la moindre erreur de la part de l'utilisateur dans ce compte peut endommager non seulement ses propres données, mais également l'ensemble du système d'exploitation. De plus, le fait de ne pas travailler sous le compte root restreint à un seul utilisateur les dégâts que pourraient faire un éventuel virus ou programme défectueux.

L'une des premières étapes dans l'installation d'un système est donc de créer un compte utilisateur normal, qui devra être utilisé pour le travail quotidien. Le compte root ne doit donc être réservé qu'aux tâches d'administration, et toute opération réalisée sous cette identité doit être contrôlée deux fois avant d'être effectivement lancée. Les programmes d'installation des distributions demandent donc toujours un mot de passe pour protéger le compte root et le nom et le mot de passe pour au moins un compte utilisateur standard après une nouvelle installation. Le mot de passe root doit être choisi avec un grand soin, surtout si l'ordinateur est susceptible d'être connecté à Internet. En effet, la moindre erreur de configuration au niveau des services fournis par l'ordinateur, couplée avec un mot de passe faible, risque de laisser votre ordinateur à la merci de pirates mal intentionnés.

Ces mêmes programmes d'installation peuvent être utilisés par la suite pour ajouter de nouveaux utilisateurs dans le système. Il est d'ailleurs recommandé de les utiliser dès qu'une telle opération doit être effectuée. Cela dit, il est bon de connaître la manière dont les utilisateurs sont gérés dans les systèmes Unix, aussi une approche plus bas niveau sera-t-elle adoptée dans cette section.

VI-C-1 - Mécanismes d'authentification des utilisateurs

La sécurité des systèmes Unix repose fondamentalement sur les mécanismes d'authentification des utilisateurs. Ces mécanismes visent à s'assurer que chacun est bien celui qu'il prétend être, afin de donner à chacun les droits d'accès aux différents services du système en fonction de ses privilèges.

L'accès aux services du système repose donc sur deux opérations essentielles : l'identification et l'authentification. L'opération d'identification consiste à annoncer qui l'on est, afin de permettre au système de déterminer les droits auxquels on a droit, et l'opération d'authentification consiste à « prouver » qu'on est bien celui qu'on prétend être. Le système refuse ses services à tout utilisateur inconnu (c'est-à-dire qui s'est identifié sous un nom inconnu) ou qui n'a pas passé avec succès la phase d'authentification.

En interne, les systèmes Unix identifient les utilisateurs par un numéro qui est propre à chacun, son « UID » (abréviation de l'anglais « User IDentifier »), mais il existe une correspondance entre cet UID et un nom d'utilisateur plus humainement lisible. Ce nom est classiquement appelé le « login », en raison du fait que c'est la première chose que le système demande lorsqu'on cherche à accéder à ses services, pendant l'opération dite de login.

L'authentification des utilisateurs se fait classiquement par mot de passe, bien que d'autres mécanismes soient possibles en théorie. L'accès au système se passe donc toujours de la manière suivante :

- le système demande à l'utilisateur son nom (c'est-à-dire son login) ;
- il demande ensuite son mot de passe ;
- il vérifie la validité du couple (login / mot de passe) pour déterminer si l'utilisateur a le droit de l'utiliser ;
- et, si l'utilisateur est connu et s'est correctement authentifié, le programme qui a réalisé l'authentification prend l'identité et les privilèges de l'utilisateur, fixe son environnement et ses préférences personnelles, puis lui donne accès au système.

L'exemple classique de ces opérations est tout simplement l'opération de login sur une console : le programme `getty` de gestion de la console demande le nom de l'utilisateur, puis passe ce nom au programme `login` qui l'authentifie en lui demandant son mot de passe. Si l'authentification a réussi, il prend l'identité de cet utilisateur et lance son shell préféré. La suite des opérations dépend du shell. S'il s'agit de `bash`, le fichier de configuration `/etc/profile` est exécuté (il s'agit donc du fichier de configuration dans lequel toutes les options communes à tous les utilisateurs pourront être placées par l'administrateur), puis les fichiers de configuration `~/.bash_profile` et `~/.bashrc` sont exécutés. Ces deux fichiers sont spécifiques à chaque utilisateur et permettent à chacun d'entre eux de spécifier leurs préférences personnelles. Le fichier `~/.bash_profile` n'est exécuté que lors d'un nouveau login, alors que le fichier `~/.bashrc` est exécuté à chaque nouveau lancement de `bash`.

Bien entendu, ces opérations nécessitent que des informations relatives aux utilisateurs soient stockées dans le système. Historiquement, elles étaient effectivement stockées dans le fichier de configuration `/etc/passwd`. Cela n'est, en général, plus le cas. En effet, cette technique se révèle peu pratique lorsque plusieurs ordinateurs en réseau sont accédés par des utilisateurs itinérants. Dans ce genre de configuration, il est courant de recourir à un service réseau permettant de récupérer les informations concernant les utilisateurs à partir d'un serveur centralisé. Plusieurs solutions existent actuellement (NIS, LDAP, etc.), mais elles fonctionnent toutes plus ou moins selon le même principe. De plus, même pour des machines isolées, le fichier `/etc/passwd` ne contient plus que les informations « publiques » sur les utilisateurs. Les informations utilisées pour l'authentification sont désormais stockées dans un autre fichier de configuration, qui n'est lisible que pour l'utilisateur `root` : le fichier `/etc/shadow`. La raison première de procéder ainsi est d'éviter que des utilisateurs malicieux puissent « casser » les mots de passe.

Note : Les mots de passe n'ont jamais été stockés en clair dans le fichier `passwd`. Le mécanisme d'authentification repose en effet sur une fonction à sens unique générant une empreinte. Les fonctions de ce type ne disposent pas de fonction inverse, il n'est donc virtuellement pas possible de retrouver un mot de passe à partir de son empreinte. Lorsqu'un utilisateur saisit son mot de passe, l'empreinte est recalculée de la même manière que lorsqu'il l'a défini initialement, et c'est cette empreinte qui est comparée avec celle qui se trouve dans le fichier `/etc/passwd` ou le fichier `/etc/shadow`. Ainsi, il est nécessaire de connaître le mot de passe en clair pour authentifier l'utilisateur, mais à aucun moment ce mot de passe n'est stocké sur le disque dur.

Cela dit, même si la récupération des mots de passe est quasiment impossible, la connaissance des empreintes des mots de passe peut être d'une aide précieuse. Un intrus potentiel peut essayer de calculer les empreintes de tous les mots de passe possibles et imaginables à l'aide d'un dictionnaire ou de mots de passe probablement choisis par les utilisateurs peu inventifs, et comparer le résultat avec ce qui se trouve dans le fichier de mot de passe. S'il y a une correspondance, l'intrus pourra pénétrer le système et tenter d'utiliser d'autres failles pour acquérir les droits de l'utilisateur `root`. Cette technique, dite attaque du dictionnaire, est tout à fait réalisable, d'une part parce que la puissance des machines actuelles permet de calculer un nombre considérable d'empreintes à la seconde, et d'autre part parce que bon nombre de personnes utilisent des mots de passe triviaux (leur date de naissance, le nom de leur chien, etc.) facilement devinables et testables.

Les systèmes Unix utilisent deux techniques pour pallier ces problèmes. La première est de ne pas calculer l'empreinte du mot de passe tel qu'il est fourni par l'utilisateur, mais de la calculer avec une donnée générée aléatoirement et stockée dans le fichier de mot de passe. Cette donnée, que l'on appelle classiquement « sel », jour

le rôle de vecteur d'initialisation de l'algorithme de génération d'empreinte. Le sel n'est jamais le même pour deux utilisateurs, ce qui fait que deux utilisateurs qui auraient le même mot de passe n'auraient toutefois pas la même empreinte. Cela complique la tâche des attaquants qui utilisent la force brute, car ils doivent ainsi recalculer les empreintes pour chaque utilisateur.

La deuxième technique utilisée est tout simplement de ne pas laisser le fichier des empreintes accessible à tout le monde. C'est pour cela que le fichier de configuration `/etc/shadow` a été introduit. Étant lisible uniquement par l'utilisateur `root`, un pirate ne peut pas récupérer les empreintes de mots de passe pour les comparer avec des empreintes de mots de passe précalculées. Il doit donc essayer les mots de passe un à un, ce qui peut en décourager plus d'un, surtout si le système se bloque pendant un certain temps après quelques tentatives infructueuses...

La technique d'authentification par mot de passe peut paraître relativement primitive, à l'heure où les cartes à puce sont légion et où l'on commence à voir apparaître des scanners d'empreintes digitales. En fait, c'est une bonne solution, mais qui ne saurait en aucun cas être exhaustive en raison des problèmes mentionnés ci-dessus. L'idéal est donc d'utiliser plusieurs systèmes d'authentification en série, ce qui laisse libre cours à un grand nombre de possibilités.

Il est évident que la technique des mots de passe traditionnellement utilisée sur les systèmes Unix n'évoluera pas aisément vers de nouveaux mécanismes d'authentification. C'est pour cela que les programmes devant réaliser une opération d'authentification ou de gestion des utilisateurs de manière générale ont été modifiés pour utiliser la bibliothèque PAM (abréviation de l'anglais « Pluggable Authentication Modules »). Cette bibliothèque permet de réaliser les opérations d'identification et d'authentification de manière externe aux programmes qui l'utilisent, et se base pour ces opérations des fichiers de configuration et des modules dynamiquement chargeables. Ainsi, grâce à la bibliothèque PAM, l'administrateur peut définir avec précision les différentes opérations réalisées pour identifier et authentifier les utilisateurs, sans avoir à modifier les programmes qui ont besoin de ces fonctionnalités. De plus, il est possible d'ajouter de nouveaux modules au fur et à mesure que les besoins évoluent, et de les intégrer simplement en modifiant les fichiers de configuration.

De nos jours, la plupart des distributions utilisent la bibliothèque PAM. Bien entendu, il existe des modules qui permettent de réaliser les opérations d'identification et d'authentification Unix classiques, et ce sont ces modules qui sont utilisés par défaut. Nous verrons le format des fichiers de configuration de la bibliothèque PAM plus en détail dans les sections suivantes.

Note : Les mécanismes décrits ici ne sont sûrs que dans le cadre d'une connexion sur un terminal local. Cela dit, il faut bien prendre conscience que la plupart des applications réseau sont de véritables passoires ! En effet, ils utilisent des protocoles qui transmettent les mots de passe en clair sur le réseau, ce qui implique que n'importe quel pirate peut les capter en moins de temps qu'il n'en faut pour le dire. Les protocoles applicatifs suivants sont réputés pour être non sûrs et ne devront donc JAMAIS être utilisés sur un réseau non sûr (et donc, à plus forte raison, sur Internet) :

- TELNET, qui permet d'effectuer des connexions à distance ;
- FTP, qui permet de transférer et de récupérer des fichiers sur une machine distante ;
- POP3, qui permet de consulter son mail ;
- SMTP, qui permet d'envoyer des mails à un serveur de messagerie ;
- X, qui permet aux applications graphiques d'afficher leurs fenêtres sur un terminal X.

Cette liste n'est pas exhaustive, mais regroupe déjà les protocoles réseau des applications les plus utilisées.

La sécurisation de ces protocoles ne peut se faire qu'en les encapsulant dans un autre protocole utilisant un canal de communication chiffré. L'un des outils les plus courants pour cela est sans doute `ssh`. Cet outil sera décrit dans la [Section 9.5.2.2](#) du chapitre traitant du réseau.

VI-C-2 - Création et suppression des utilisateurs

La création d'un nouvel utilisateur est une opération extrêmement facile. Il suffit simplement de lui créer un répertoire personnel dans le répertoire `/home/` et de le définir dans le fichier de configuration `/etc/passwd`. Si votre système utilise

les shadow passwords, ce qui est probable, il faut également définir cet utilisateur dans le fichier de configuration `/etc/shadow`. De plus, il faut ajouter cet utilisateur dans au moins un groupe d'utilisateurs dans le fichier `/etc/group`.

Le fichier de configuration `/etc/passwd` est constitué de plusieurs lignes, à raison d'une ligne par utilisateur. Chaque ligne est constituée de plusieurs champs, séparés par deux points (caractère ':'). Ces champs contiennent respectivement le login de l'utilisateur, l'empreinte de son mot de passe, son identifiant numérique, l'identifiant numérique de son groupe principal, son nom complet ou un commentaire, le chemin de son répertoire personnel et le chemin sur son interpréteur de commandes favori. Si le champ du mot de passe contient un astérisque, le compte est désactivé. S'il est vide, le mot de passe est stocké dans le fichier `/etc/shadow`.

Le fichier de configuration `/etc/shadow` a une syntaxe similaire à celle de `/etc/passwd`, mais contient les champs suivants : le login de l'utilisateur, l'empreinte de son mot de passe, le nombre de jours depuis que le mot de passe a été défini (comptés à partir du premier janvier 1970), le nombre de jours après cette date à attendre avant que le mot de passe puisse être changé, le nombre de jours au-delà duquel le mot de passe doit obligatoirement être changé, le nombre de jours avant la date d'expiration de son mot de passe pendant lesquels l'utilisateur doit être averti que son mot de passe va expirer, le nombre de jours à attendre avant de désactiver le compte après l'expiration du mot de passe, et le nombre de jours depuis que le compte est désactivé, compté depuis le premier janvier 1970. Il est possible de supprimer l'obligation pour les utilisateurs de changer régulièrement de mot de passe en donnant un nombre de jours minimum supérieur au nombre de jours maximum avant le changement de mot de passe.

Enfin, le fichier de configuration `/etc/group`, dans lequel les groupes d'utilisateurs sont définis, ne dispose que des champs suivants : le nom du groupe, son mot de passe (cette fonctionnalité n'est plus utilisée), son identifiant numérique, et la liste des utilisateurs qui y appartiennent, séparés par des virgules.

Bien entendu, tous ces champs ne doivent pas être modifiés à la main. La commande `useradd` permet de définir un nouvel utilisateur simplement. Cette commande suit la syntaxe suivante :

```
useradd [-c commentaire] [-d répertoire] [-e expiration] [-f inactivité] \
  [-g groupe] [-G groupes] [-m [-k modèle]] [-p passe]
  [-s shell] [-u uid [-o]] login
```

Comme vous pouvez le constater, cette commande prend en paramètre le login de l'utilisateur, c'est-à-dire le nom qu'il devra utiliser pour s'identifier sur le système, et un certain nombre d'options complémentaires. Ces options sont récapitulées dans le tableau suivant :

Option	Signification
-c	Permet de définir le champ commentaire du fichier de mot de passe.
-d	Permet de fixer le répertoire personnel de l'utilisateur.
-e	Permet de fixer la date d'expiration du compte. Cette date doit être spécifiée au format AAAA-MM-JJ.
-f	Permet de définir le nombre de jours avant que le compte ne soit désactivé une fois que le mot de passe est expiré. La valeur -1 permet de ne jamais désactiver le compte.
-g	Permet de définir le groupe principal auquel l'utilisateur appartient. Il s'agit souvent du groupe users.
-G	Permet de donner la liste des autres groupes auxquels l'utilisateur appartient. Cette liste est constituée des noms de chacun des groupes, séparés par des virgules.
-m	Permet de forcer la création du répertoire personnel de l'utilisateur. Les fichiers du modèle de répertoire personnel stockés dans le répertoire <code>/etc/skel/</code> sont automatiquement copiés dans le nouveau répertoire. Si ces fichiers doivent être copiés à partir d'un autre répertoire, il faut spécifier celui-ci à l'aide de l'option -k
-p	Permet de donner le mot de passe initial du compte. Cette option ne doit jamais être utilisée, car le mot de passe apparaît dans la ligne de commande du processus et peut être lue par un utilisateur

	mal intentionné. On fixera donc toujours le mot de passe initial de l'utilisateur à l'aide de la commande <code>passwd</code> .
<code>-s</code>	Permet de spécifier le shell par défaut utilisé par l'utilisateur.
<code>-u</code>	Permet de spécifier l'UID de l'utilisateur. Cette valeur doit être unique en général, cependant, il est possible de forcer l'utilisation d'un même UID pour plusieurs utilisateurs à l'aide de l'option <code>-o</code> . Cela permet de créer un deuxième compte pour un utilisateur déjà existant.

L'ajout d'un groupe se fait avec la commande `groupadd`, qui suit la syntaxe suivante, beaucoup plus simple que celle de `useradd` :

```
groupadd [-g GID [-o]] nom
```

où `nom` est le nom du groupe et `GID` son numéro. Il n'est normalement pas possible de définir un groupe avec un identifiant numérique déjà attribué à un autre groupe, sauf si l'on utilise l'option `-o`.

De la même manière, la suppression d'un utilisateur peut se faire manuellement en effaçant son répertoire personnel et en supprimant les lignes qui le concernent dans les fichiers `/etc/passwd`, `/etc/shadow` et `/etc/group`. Il est également possible d'utiliser la commande `userdel`. Cette commande utilise la syntaxe suivante :

```
userdel [-r] login
```

où `login` est le nom de l'utilisateur. L'option `-r` permet de demander à `userdel` d'effacer récursivement le répertoire personnel de l'utilisateur, ce qu'elle ne fait pas par défaut. Il existe également une commande `groupdel` pour supprimer un groupe d'utilisateurs (cette commande supprime le groupe seulement, pas les utilisateurs qui y appartiennent !).

Note : Comme pour la plupart des autres opérations d'administration système, il est fortement probable que l'outil de configuration fourni avec votre distribution dispose de toutes les fonctionnalités nécessaires à l'ajout et à la suppression des utilisateurs. Il est recommandé d'utiliser cet outil, car il peut effectuer des opérations d'administration complémentaires que les outils standards n'effectuent pas forcément, comme la définition des comptes mail locaux par exemple.

VI-C-3 - Description de la bibliothèque PAM

La bibliothèque PAM permet de centraliser toutes les opérations relatives à l'identification et à l'authentification des utilisateurs. Ces tâches sont en effet déportées dans des modules spécialisés qui peuvent être chargés dynamiquement dans les programmes qui en ont besoin, en fonction de paramètres définis dans des fichiers de configuration. Le comportement des applications peut donc être parfaitement défini simplement en éditant ces fichiers.

La bibliothèque PAM permet une très grande souplesse dans l'administration des programmes ayant trait à la sécurité du système et devant réaliser des opérations privilégiées. Il existe déjà un grand nombre de modules, capables de réaliser une multitude de tâches diverses et variées, et que l'on peut combiner à loisir pour définir le comportement de toutes les applications utilisant la bibliothèque PAM. Il est hors de question de décrire chacun de ces modules ici, ni même de donner la configuration des programmes qui utilisent PAM. Cependant, nous allons voir les principes généraux permettant de comprendre comment les modules de la bibliothèque sont utilisés.

Initialement, toute la configuration de PAM se faisait dans le fichier de configuration `/etc/pam.conf`. Ce fichier contenait donc la définition du comportement de chaque programme utilisant la bibliothèque PAM, ce qui n'était pas très pratique pour l'administration et pour les mises à jour. Les informations de chaque programme ont donc été séparées en plusieurs fichiers distincts, à raison d'un fichier par application, tous stockés dans le répertoire `/etc/pam.d/`. Il est fort probable que votre distribution utilise cette solution, aussi le format du fichier `/etc/pam.conf` ne sera-t-il pas décrit.

Certains modules utilisent des fichiers de configuration pour déterminer la manière dont ils doivent se comporter. Ces fichiers de configuration sont tous stockés dans le répertoire `/etc/security/`. Les modules de PAM eux-mêmes sont, quant à eux, stockés dans le répertoire `/lib/security/`.

Le principe de fonctionnement est le suivant. Lorsqu'un programme désire réaliser une opération relative à l'authentification d'un utilisateur, il s'adresse à la bibliothèque PAM pour effectuer cette opération. La bibliothèque recherche dans le répertoire `/etc/pam.d/` le fichier de configuration correspondant à cette application (il porte généralement le nom de l'application elle-même), puis détermine les modules qui doivent être chargés dynamiquement dans l'application. Si le fichier de configuration d'une application ne peut pas être trouvé, le fichier de configuration `/etc/pam.d/other` est utilisé, et la politique de sécurité par défaut qui y est définie est utilisée. Quel que soit le fichier de configuration utilisé, chaque module est utilisé en fonction des paramètres qui y sont stockés. Les modules peuvent, s'ils en ont besoin, utiliser leurs propres fichiers de configuration, qui se trouvent dans le répertoire `/etc/security/`. Ces fichiers portent généralement le nom du module avec l'extension `.conf`. Par exemple, le fichier de configuration du module `limits`, qui prend en charge les limites d'utilisation des ressources système pour chaque utilisateur, est le fichier `/etc/security/limits.conf`.

Les fichiers de configuration des applications sont constitués de lignes définissant les différents modules qui doivent être chargés, le contexte dans lequel ils sont chargés, et comment doit se comporter l'application en fonction du résultat de l'exécution des opérations réalisées par ces modules. L'ordre des lignes est important, puisqu'elles sont analysées les unes après les autres. Chaque ligne est constituée de trois colonnes. La première colonne indique le cadre d'utilisation du module. La deuxième colonne indique le comportement que doit adopter la bibliothèque PAM en fonction du résultat renvoyé par le module après son exécution. Enfin, la troisième colonne donne le chemin d'accès complet au module, éventuellement suivi des options qui doivent lui être communiquées pour son exécution.

Les modules peuvent être utilisés dans l'un des contextes suivants :

- `auth`, qui est le contexte utilisé par les programmes qui demandent l'authentification de l'identité de l'utilisateur ;
- `account`, qui est le contexte utilisé par les programmes qui désirent obtenir des informations sur l'utilisateur (répertoire personnel, `shell`, etc.)
- `password`, qui est le contexte utilisé par les applications qui cherchent à revalider l'authentification de l'utilisateur. Les programmes comme `passwd` par exemple, qui demandent le mot de passe de l'utilisateur avant d'en fixer un nouveau, sont susceptibles d'utiliser ce contexte ;
- `session`, qui est le contexte utilisé par les applications lorsqu'elles effectuent les opérations de gestion d'ouverture et de fermeture de session. Ce contexte peut être utilisé pour réaliser des tâches administratives, comme l'enregistrement de l'utilisateur dans la liste des utilisateurs connectés ou le chargement des préférences personnelles de l'utilisateur par exemple.

Une même application peut définir plusieurs jeux de règles pour plusieurs contextes différents, et certains modules peuvent être utilisés dans plusieurs contextes différents également. Cependant, lorsqu'une application réalise une demande à la bibliothèque PAM, cette demande n'est exécutée que dans le cadre d'un contexte bien défini. Attention cependant, une même application peut effectuer plusieurs opérations successivement dans des contextes différents. Par exemple, le programme `login` peut utiliser le contexte `auth` pour valider l'identité de l'utilisateur, puis le contexte `account` pour déterminer le répertoire personnel de l'utilisateur, et enfin le contexte `session` pour enregistrer l'utilisateur dans le journal des utilisateurs connectés.

Le comportement de la bibliothèque PAM en fonction du résultat de l'exécution des modules dépend de ce qui est spécifié dans la deuxième colonne des lignes de ces modules. Les options qui peuvent y être utilisées sont les suivantes :

- `required`, qui permet d'indiquer que le succès de l'opération effectuée par le module est nécessaire pour que l'opération effectuée dans le contexte spécifié dans la première colonne de la ligne réussisse. Un échec sur cette ligne ne provoque pas l'arrêt de l'analyse du fichier de configuration, ce qui permet d'appeler d'autres modules, par exemple pour générer des traces dans les fichiers de traces du système. Cependant, quel que soit le comportement des modules suivants, l'opération demandée par le programme appelant échouera ;
- `requisite`, qui permet d'indiquer que le succès de l'opération effectuée par le module est nécessaire, faute de quoi l'opération réalisée par le programme appelant échoue immédiatement. Les autres modules ne sont donc pas chargés en cas d'échec, contrairement à ce qui se passe avec l'option `required` ;
- `sufficient`, qui permet d'indiquer que le succès de l'exécution du module garantira le succès de l'opération demandée par le programme appelant. Les modules suivants seront chargés malgré tout après l'appel de ce module, sauf s'il s'agit de modules de type `required` ;

- optional, qui permet d'indiquer que le résultat du module ne doit être pris en compte que si aucun autre module ne peut déterminer si l'opération est valide ou non. Dans le cas contraire, le module est chargé, mais son résultat est ignoré.

À titre d'exemple, nous pouvons présenter deux implémentations possibles du fichier de configuration par défaut `/etc/pam.d/other`. Une configuration extrêmement sûre interdira l'accès à toute fonctionnalité fournie par un programme n'ayant pas de fichier de configuration. Dans ce cas de configuration, on utilisera un fichier comme celui-ci :

```
auth      required /lib/security/pam_warn.so
auth      required /lib/security/pam_deny.so
account   required /lib/security/pam_deny.so
password  required /lib/security/pam_warn.so
password  required /lib/security/pam_deny.so
session   required /lib/security/pam_deny.so
```

Nous voyons que toutes les opérations de type authentification ou de revalidation de l'identité sont d'abord tracées dans les fichiers de traces du système, puis déclarées comme interdites. Une telle configuration peut être un peu trop restrictive, car, en cas d'erreur dans un fichier de configuration d'une application, l'accès à cette application peut être interdit systématiquement. Une autre configuration, plus permissive, cherchera à utiliser les mécanismes d'identification et d'authentification Unix classiques. Cela se fait avec les modules `pam_unix_auth`, `pam_unix_acct`, `pam_unix_passwd` et `pam_unix_session` :

```
auth      required /lib/security/pam_unix_auth.so
account   required /lib/security/pam_unix_acct.so
password  required /lib/security/pam_unix_passwd.so
session   required /lib/security/pam_unix_session.so
```

Les autres fichiers de configuration ne seront pas décrits ici, car ils dépendent de chaque application et de chaque distribution. Vous pouvez consulter ceux qui sont installés sur votre système si vous désirez en savoir plus.

Nous ne décrivons pas non plus la syntaxe des fichiers de configuration des différents modules, car cela dépasserait largement le cadre de ce document. Cela dit, la plupart de ces fichiers sont parfaitement commentés et leur modification ne devrait pas poser de problème particulier. À titre d'exemple, on peut présenter le cas du module de gestion des limites des ressources consommées par les utilisateurs. Si l'on désire restreindre le nombre de processus que les utilisateurs peuvent lancer, on pourra ajouter la ligne suivante dans le fichier `/etc/security/limits.conf` :

```
@users hard nproc 256
```

Il faudra également demander le chargement de ce module dans les fichiers de configuration des applications fournissant un accès au système. Pour cela, on ajoutera une ligne telle que celle-ci à la fin de leur fichier de configuration :

```
session required /lib/security/pam_limits.so
```

Il est également possible de limiter le nombre de logins de chaque utilisateur, la quantité de mémoire qu'il peut consommer, la durée de connexion autorisée, la taille maximum des fichiers qu'il peut manipuler, etc. Vous trouverez de plus amples renseignements sur la configuration des modules de PAM dans le guide d'administration de PAM pour Linux, que l'on trouvera avec les [sources de PAM](#).

VI-D - Gestion des paquetages

Il existe plusieurs systèmes de gestion de paquetages, chaque distribution fournissant ses propres outils. Toutefois, les fonctionnalités de ces outils sont toujours de maintenir la liste des paquetages installés, et de permettre l'installation, la mise à jour et la suppression de ces paquetages. Nous ne présenterons ici que les systèmes de paquetages des distributions les plus utilisées. Si les outils de votre distribution ne sont pas décrits dans cette section, ne vous alarmez pas, leur documentation vous permettra sans doute de trouver rapidement la manière de réaliser les opérations élémentaires équivalentes.

VI-D-1 - Le gestionnaire de paquetages rpm

La plupart des distributions actuelles utilisent le format de fichier « rpm » (« Redhat Package Manager ») pour leurs paquetages. Ce format de fichier a été introduit par la distribution Redhat, mais a été licencié sous la licence GNU, ce qui a permis aux autres distributions de l'utiliser. Ces fichiers encapsulent tous les fichiers des paquetages, ainsi que des informations permettant de gérer les dépendances entre les paquetages, leurs versions, la manière de les installer dans le système, de les supprimer ou de les mettre à jour facilement.

Les fichiers rpm peuvent être manipulés à l'aide du programme rpm. Il est probable que le programme d'installation de votre distribution vous évite d'avoir à manipuler cet outil vous-même. Cependant, les principales commandes de rpm seront décrites ici, afin que vous puissiez l'utiliser en cas de besoin.

Le programme rpm utilise une syntaxe très classique :

```
rpm options [paquetage]
```

Les options indiquent les opérations à effectuer. La première option est bien entendu l'option -i, qui permet l'installation d'un paquetage :

```
rpm -i paquetage
```

La mise à jour d'un paquetage déjà installé se fait à l'aide de l'option -U :

```
rpm -U paquetage
```

La suppression d'un paquetage se fait à l'aide de l'option -e :

```
rpm -e paquetage
```

La commande permettant d'obtenir les informations (auteur, description, version) sur un paquetage contenu dans un fichier rpm est la suivante :

```
rpm -qi -p paquetage
```

Enfin, la commande pour lister tous les fichiers d'un paquetage contenu dans un fichier rpm est la suivante :

```
rpm -ql -p paquetage
```

Cette commande affiche les chemins complets, ce qui permet de savoir dans quel répertoire chaque fichier sera installé.

Il existe beaucoup d'autres options disponibles. Cependant, leur description dépasserait le cadre de ce document. Vous pouvez toujours consulter la page de manuel rpm si vous désirez plus d'informations.

VI-D-2 - Le gestionnaire de paquetages apt

Les distributions basées sur la distribution Debian utilisent le gestionnaire de paquetages apt-get, qui est sans doute l'un des plus performants qui soient. Ce gestionnaire de paquetages permet d'installer, mettre à jour et supprimer des paquetages en tenant compte des dépendances entre ceux-ci, et ce à partir de n'importe quelle source. Ainsi, si l'on est connecté en réseau, il est possible d'installer ou de mettre à jour n'importe quel paquetage via le réseau. Dans le cas contraire, il faut référencer le répertoire contenant les paquetages pour que apt-get puisse les utiliser, ce après quoi les opérations sont exactement les mêmes.

La liste des localisations des paquetages est stockée dans le fichier de configuration `/etc/apt/sources.list`. Ce fichier contient une ligne par référentiel disponible. Le format général de ces lignes est le suivant :

```
type site distribution sections
```

où `type` est le type du paquetage, `site` est l'emplacement du référentiel, `distribution` est le nom de la distribution, et `sections` une liste des sections dans lesquelles les paquetages pourront être trouvés. Le type de paquetage le plus courant est `deb`, qui est utilisé pour les paquetages binaires pour la distribution Debian. Par exemple, la ligne suivante permet de référencer le site principal de Debian aux États-Unis :

```
deb http://http.us.debian.org/debian stable main contrib non-free
```

Par défaut, le fichier `/etc/apt/sources.list` contient les références sur les principaux sites Internet de Debian. Si vous voulez installer des paquetages que vous avez téléchargés, il va vous falloir y rajouter une ligne telle que celle-ci :

```
deb file:/base répertoire/
```

où `base` est un répertoire dans lequel vous devez placer un sous-répertoire contenant vos paquetages, et `répertoire` est le nom de ce sous-répertoire. Vous pouvez bien entendu classer vos paquetages dans différents sous-répertoires, du moment que vous ajoutez les lignes adéquates dans le fichier `sources.list`.

`apt-get` recherchera alors, dans chacun de ces sous-répertoires, un fichier d'index contenant la liste des paquetages. Ce fichier doit être nommé `Packages.gz`, et peut être créé simplement en compressant le résultat de la commande `dpkg-scanpackages`. La commande à utiliser pour créer un tel fichier est la suivante :

```
dpkg-scanpackages répertoire /dev/null | gzip > répertoire/Packages.gz
```

où `répertoire` est le sous-répertoire du répertoire de base dans lequel vous avez placé vos paquetages binaires. Cette ligne de commande suppose que le répertoire courant soit le répertoire base spécifié dans la ligne que vous avez ajoutée dans le fichier `sources.list`.

Dans le cas des CD-ROM, la procédure est plus simple. En effet, l'utilitaire `apt-cdrom` permet de prendre en compte un CD-ROM de manière automatique, avec une simple commande telle que celle-ci :

```
apt-cdrom -d répertoire add
```

où `répertoire` est le répertoire servant de point de montage de votre CD-ROM. Cette commande ajoute la référence du CD-ROM dans le fichier `/var/lib/apt/cdrom.list`. Il n'existe pas de commande pour supprimer un CD-ROM de cette liste.

Une fois les sources de paquetages définies, leur manipulation est élémentaire. Avant toute chose, il est nécessaire de mettre à jour la liste des paquetages dans le système de paquetages. Cette opération doit être réalisée de manière assez régulière en général, car cette liste évolue assez rapidement. Pour effectuer cette opération, il suffit d'exécuter la commande `update` d'`apt-get` :

```
apt-get update
```

L'installation d'un paquetage se fait avec la commande `install` d'`apt-get` :

```
apt-get install paquetage
```

`paquetage` est ici le nom du paquetage à installer. Si ce paquetage est déjà installé et que l'on désire le réinstaller, par exemple pour le réparer ou pour le mettre à jour ajoutera l'option `--reinstall` :

```
apt-get --reinstall install paquetage
```

La suppression d'un paquetage est toute aussi simple. Elle se fait évidemment avec la commande `remove` :

```
apt-get remove paquetage
```

Cette commande supprime le paquetage, mais ne détruit pas les fichiers de configuration qu'il a installés. Cela permet d'éviter de perdre les éventuelles modifications que l'on pourrait y avoir apportées. Si l'on désire supprimer ces fichiers également, il faut ajouter l'option `--purge` :

```
apt-get --purge remove paquetage
```

La mise à jour de tous les paquetages existants se fait simplement avec la commande `upgrade` d'`apt-get` :

```
apt-get upgrade
```

Cette commande ne permet pas toujours de résoudre les nouvelles dépendances sur les paquetages. C'est pour cela que la commande `dist-upgrade` a été définie. Elle permet de mettre à jour le système complet (mais est évidemment bien plus longue).

Enfin, si vous désirez obtenir des informations sur un paquetage, vous devez utiliser le programme `apt-cache`. Ce programme permet de rechercher un paquetage à l'aide de mots-clefs, et d'afficher les informations complètes sur le paquetage. Par exemple, pour obtenir des informations sur un paquetage, il faut utiliser la commande suivante :

```
apt-cache show paquetage
```

où `paquetage` est le nom du paquetage. De même, pour obtenir la liste des paquetages qui se rapporte à un mot-clef particulier, il faut utiliser la commande `search` d'`apt-cache` :

```
apt-cache search mot-clé
```

où `mot-clé` est le mot-clé que l'on doit rechercher dans les paquetages.

VI-D-3 - Le gestionnaire de paquetages `pkgtool`

Les distributions basées sur la Slackware utilisent le gestionnaire de paquetages `pkgtool`. Celui-ci est beaucoup plus rudimentaire que les gestionnaires présentés précédemment, car il ne prend pas en charge les dépendances entre les paquetages, et ne s'occupe pas de localiser les paquetages sur le réseau. Bien entendu, la manipulation des paquetages est très simplifiée, et il est du ressort de l'utilisateur de savoir ce qu'il fait.

L'installation d'un paquetage se fait avec la commande `installpkg` :

```
installpkg paquetage
```

où `paquetage` est le nom du paquetage à installer.

La suppression d'un paquetage se fait avec la commande `removepkg` :

```
removepkg paquetage
```

Enfin, la mise à jour d'un paquetage se fait avec la commande `upgradepkg` :

```
upgradepkg paquetage
```

Cette commande supprime toutes les anciennes versions du paquetage après avoir installé la nouvelle. La commande `upgradepkg` peut également accepter en paramètre l'option `--install-new`, qui permet d'installer le paquetage s'il n'est pas encore installé, et l'option `--reinstall`, qui permet de réinstaller le paquetage s'il est déjà installé.

Enfin, la Slackware fournit l'outil `pkgtool`, qui est un peu plus convivial à utiliser que les outils en ligne de commande précédents. Cet outil fournit une interface utilisateur en mode texte permettant d'installer, de supprimer, d'obtenir des informations sur les paquetages, ainsi que de reconfigurer le système.

VI-E - Notion de niveau d'exécution et amorçage du système

La plupart des systèmes Unix disposent de plusieurs modes de fonctionnement, que l'on appelle des niveaux d'exécution. Dans chaque niveau d'exécution, un certain nombre de services sont accessibles. Ainsi, le lancement et l'arrêt des services des systèmes Unix peut se faire de manière groupée simplement en changeant de niveau d'exécution. En général, il existe 7 niveaux d'exécution, dont seulement trois fournissent des services bien définis pour quasiment toutes les distributions de Linux.

Le niveau 0 correspond à l'arrêt du système, et aucun service n'est disponible (à part le redémarrage de la machine bien entendu...). Le fait de passer dans le niveau d'exécution 0 correspond donc à arrêter le système. Le niveau 6 correspond au redémarrage de la machine. Le fait de passer dans le niveau d'exécution 6 revient donc à arrêter et à redémarrer la machine. Le niveau d'exécution 1 correspond au mode de fonctionnement mono utilisateur (encore appelé mode de maintenance). Ce niveau d'exécution fournit les services de base pour un seul utilisateur (normalement l'administrateur du système). Dans ce niveau d'exécution, l'administrateur peut changer la configuration et effectuer les tâches de maintenance les plus critiques (par exemple, vérifier le système de fichiers racine). La signification des autres niveaux d'exécution dépend de la distribution que vous utilisez, mais en général le niveau d'exécution 2 correspond au mode multiutilisateur avec réseau, mais sans XWindow, et les niveaux d'exécution 3 et 4 correspondent au mode multiutilisateur avec login graphique sous XWindow. Les autres niveaux restent à votre disposition.

Le programme en charge de gérer les niveaux d'exécution est le programme `init`. Ce programme est le premier programme lancé par le noyau après qu'il a été chargé et démarré par le gestionnaire d'amorçage (à savoir, en général, LILO ou le GRUB). Ce programme ne peut pas être détruit ou arrêté, et c'est réellement le processus père de tous les autres dans le système. Le rôle fondamental d'`init` est de gérer les changements de niveau d'exécution et de lancer et arrêter les services du système en fonction de ces niveaux. Toutefois, il s'occupe également de tâches de base concernant la gestion des autres processus. En particulier, il permet de supprimer les processus zombies.

Note : Un processus « zombie » est un processus qui vient de se terminer et dont aucun processus n'a lu le code de retour. De tels processus apparaissent généralement lorsque leur processus père se termine avant eux, car, généralement, c'est toujours le processus père qui lit le code de retour de ses processus fils.

Il suffit d'utiliser la syntaxe suivante pour forcer le changement de niveau d'exécution :

```
init niveau
```

où `niveau` est le niveau d'exécution à atteindre. Cela dit, cette manière de faire est assez rare, car en général on n'a pas besoin de changer de niveau d'exécution, sauf pour arrêter et redémarrer la machine. Mais pour ces opérations, les commandes `shutdown`, `halt` et `reboot` sont déjà disponibles.

Le niveau d'exécution dans lequel le système doit se placer lors de son démarrage peut également être précisé en paramètre du noyau lors du démarrage. Vous devrez donc utiliser une commande semblable à celle-ci : `LILO boot:linux niveau` si vous utilisez LILO, ou `kernel noyau niveau` si vous utilisez le GRUB pour démarrer le noyau Linux dans le niveau d'exécution `niveau`. Ainsi, pour passer en mode monutilisateur (c'est-à-dire le mode de maintenance), il suffit de taper la commande suivante à l'amorçage de LILO :

`LILO boot:linux 1`

ou la commande suivante sur la ligne de commande interactive du GRUB :

```
kernel /boot/vmlinuz 1
```

Note : Il est également possible d'utiliser le paramètre `single`, qui est synonyme du niveau d'exécution 1.

Le comportement d'init est défini dans le fichier de configuration `/etc/inittab`. Ce fichier contient la description des niveaux d'exécution, le niveau par défaut dans lequel le système se place au démarrage, et les actions qu'init doit effectuer lorsque certains événements arrivent. En particulier, il est indiqué quels sont les scripts qui doivent être exécutés lors du changement de niveau d'exécution. Il est fortement, mais alors très fortement déconseillé de toucher au fichier `/etc/inittab` pour des raisons bien évidentes. Vous trouverez de plus amples renseignements dans les pages de manuel d'init et d'inittab.

Lorsqu'on change de niveau d'exécution, ainsi qu'au démarrage du système, init appelle des scripts de configuration pour effectuer les opérations de configuration du système et de lancement et d'arrêt des différents services. Comme on l'a vu, ces scripts sont spécifiés dans le fichier `/etc/inittab`. En général, ils sont tous placés dans le répertoire `/etc/rc.d/` (ou `/sbin/init.d/`, selon votre distribution). Ce répertoire contient donc :

- le script exécuté lors du démarrage du système ;
- les scripts exécutés lors de la sortie d'un niveau d'exécution ;
- les scripts exécutés lors de l'entrée dans un niveau d'exécution.

Le script appelé lors du démarrage du système est en général spécifié directement dans `/etc/inittab`. Vous pouvez y ajouter les commandes spécifiques à votre système, par exemple les commandes de configuration du matériel. Ce fichier n'est exécuté qu'une seule fois et est placé directement dans `/etc/rc.d/` (ou dans `/sbin/init.d/`).

En revanche, les scripts appelés lors du changement de niveau d'exécution sont souvent placés dans des sous-répertoires du répertoire `rc.d` ou `init.d`. Ils sont classés à raison d'un répertoire par niveau d'exécution. Ces sous-répertoires portent le nom de `rc0.d`, `rc1.d`, `rc2.d`, etc. pour les différents niveaux d'exécution. En fait, un seul script est exécuté par init lorsqu'on change de niveau d'exécution, et ce script se charge d'exécuter les bons scripts dans les sous-répertoires de `rc.d` ou `init.d`. Classiquement, ce script principal est appelé avec le numéro du niveau d'exécution en paramètre, et il commence par appeler les scripts de sortie du niveau d'exécution courant, puis les scripts d'entrée dans le nouveau niveau d'exécution.

La distinction entre les scripts d'entrée et de sortie dans chaque répertoire `rc?.d` se fait par la première lettre du script. Sur certaines distributions, la lettre 'K' correspond aux scripts de sortie et la lettre 'S' au script d'entrée (ces deux lettres correspondent respectivement aux mots anglais « Kill » et « Start »). De plus, l'ordre dans lequel ces scripts doivent être exécutés est indiqué par le nombre suivant cette lettre dans le nom du script. Cela dit, ces conventions peuvent varier selon votre distribution. Consultez votre documentation pour plus de détails à ce sujet.

Il est assez courant que les répertoires `rc?.d` ne contiennent que des liens symboliques vers les fichiers de scripts, et que ceux-ci soient tous placés directement dans le répertoire `/etc/rc.d/` (ou `/sbin/init.d/`). La raison en est qu'un même script peut être utilisé pour différents niveaux d'exécution, et qu'il n'a donc pas de raison d'être dans le répertoire d'un niveau plutôt que celui d'un autre.

De même, il est assez courant que chacun de ces scripts gère à la fois l'entrée et la sortie du niveau d'exécution, selon le paramètre qu'il reçoit lors de son appel. Parmi les paramètres les plus courants, on retrouve les suivants :

- `start`, pour le démarrage du service correspondant ;
- `stop`, pour son arrêt.

Ce sont les deux paramètres que le script de contrôle de changement de niveau d'exécution (celui appelé par init et enregistré dans `/etc/inittab`) utilisera lors de l'entrée et de la sortie du niveau d'exécution. Il existe d'autres paramètres, par exemple `restart` permet de redémarrer le service correspondant.

De cette manière, vous pouvez ajouter ou supprimer des services simplement dans chaque niveau d'exécution. Il suffit d'écrire un fichier script capable de prendre en paramètre l'action à réaliser sur le service (start ou stop), de le placer dans `/etc/rc.d/` (ou `/sbin/init.d/`) et de créer les liens dans les sous-répertoires `/etc/rc.d/rc?.d/` (ou `/sbin/init.d/rc?.d/`). Dès lors, votre service sera arrêté ou redémarré selon le niveau d'exécution dans lequel passera le système.

La rédaction des scripts shells de configuration dépasse largement le cadre de ce document, de même que la configuration du comportement du système à chaque changement de niveau d'exécution. La description qui était donnée ici permet simplement d'avoir une idée plus claire de la manière dont le système se comporte au démarrage et à l'arrêt. Consultez la documentation de votre distribution pour plus de détails à ce sujet.

VI-F - Maintenance des systèmes de fichiers

Cette section vous présentera les opérations de base sur les systèmes de fichiers, telle que leur création, leur montage et démontage, et leur vérification. Vous y trouverez également la manière de monter automatiquement les systèmes de fichiers les plus utilisés au démarrage du système, ainsi que la manière de monter « à la demande » les systèmes de fichiers amovibles. Vous verrez enfin comment réaliser des agrégats de volumes et comment chiffrer vos systèmes de fichiers. En revanche, la description des systèmes de fichiers réseau sera donnée dans le chapitre traitant de la [configuration réseau](#).

VI-F-1 - Création des systèmes de fichiers

Nous avons vu lors de l'installation comment créer un nouveau système de fichiers EXT2 sur une partition à l'aide de la commande `mke2fs`. Cependant, Linux peut gérer de nombreux autres systèmes de fichiers, et leur création peut se faire de manière tout à fait semblable à la création d'un système de fichiers EXT2.

Pour chaque système de fichiers, une commande spécifique est fournie afin de le créer. Toutefois, une commande générique de création de systèmes de fichiers permet d'uniformiser la manière de créer les systèmes de fichiers, rendant ainsi inutile la connaissance de ces commandes spécifiques.

Cette commande générique est la commande `mkfs`. Elle prend en paramètre le type de système de fichiers à créer, que l'on peut spécifier à l'aide de l'option `-t`. En fait, `mkfs` appelle systématiquement la commande de création du système de fichiers spécifié en paramètre. Pour que cela fonctionne, il est nécessaire que cette commande ait un nom de la forme `mkfs.type`, où `type` est le nom du système de fichiers. Ainsi, pour créer un système de fichiers JFS (système de fichiers journalisé créé par IBM) sur la première partition du premier disque SCSI, il suffit d'exécuter la commande suivante :

```
mkfs -t jfs /dev/sda1
```

En général, les commandes `mkfs.type` ne sont rien d'autre que des liens vers les programmes de création spécifiques des systèmes de fichiers.

Les commandes de création des systèmes de fichiers peuvent prendre des options particulières, qu'il faut donc pouvoir leur fournir via `mkfs`. `mkfs` transfère donc toutes les options qu'il trouve après la spécification du type de système de fichiers telles qu'elles aux programmes de création spécifique des systèmes de fichiers. La liste des options effectivement disponibles peut être consultée dans les pages de manuel respectives de ces programmes.

VI-F-2 - Montage des systèmes de fichiers

Comme il l'a été vu dans le chapitre expliquant les généralités sur Unix, les systèmes de fichiers ne sont donc pas accessibles directement. Ils doivent en effet subir une opération que l'on nomme le montage avant de pouvoir être utilisés. Le montage est donc l'opération qui consiste à associer un répertoire au point d'entrée d'un système de fichiers. Une fois monté, les données d'un système de fichiers sont accessibles à partir de ce répertoire. L'opération de montage permet ainsi de réaliser une abstraction du support des systèmes de fichiers, qui peuvent se trouver aussi bien sur disque qu'en mémoire ou que sur un réseau.

L'opération permettant de monter un disque suit la syntaxe suivante :

```
mount [-t type] fichier base
```

où `fichier` est le fichier contenant le système de fichiers à monter (en général, il s'agit d'un fichier spécial de périphérique, mais ce peut également être une image disque), et `base` est le point de montage, c'est-à-dire le répertoire à partir duquel le système de fichiers doit être accédé. L'option `-t` permet d'indiquer le type du système de fichiers. Notez qu'en général il n'est pas nécessaire de le préciser, car le noyau sait reconnaître la plupart des systèmes de fichiers automatiquement.

Pour information, les types de systèmes de fichiers les plus utilisés sont les suivants :

- `ext2`, pour les systèmes de fichiers EXT2 ;
- `ext3`, pour les systèmes de fichiers EXT3. Il est nécessaire d'avoir créé le journal du système de fichiers au préalable, avec l'option `-j` de la commande `mke2fs` ou de la commande `tune2fs` ;
- `reiserfs`, pour les systèmes de fichiers ReiserFS ;
- `iso9660`, pour les CD-ROM (qu'ils soient avec extensions Joliet ou Rock Ridge ou en mode ISO 9660 pur) ;
- `ntfs`, pour les systèmes de fichiers NTFS ;
- `msdos`, pour les systèmes de fichiers FAT normaux ;
- `vfat`, pour les systèmes de fichiers FAT32.

Si le répertoire de montage n'est pas vide, les fichiers qui s'y trouvent sont masqués par le système de fichiers monté. Il est donc recommandé de ne monter les systèmes de fichiers que dans des répertoires vides.

Pour les supports de système de fichiers amovibles, il arrive parfois que Linux ne puisse pas déterminer la géométrie ou la table de partition du support de données. Par exemple, lorsque l'on insère une carte mémoire dans un lecteur de carte, Linux considère que le périphérique n'a pas changé (puisque le lecteur de carte est toujours branché) et ne relit donc pas les informations de la carte mémoire. Il peut donc être nécessaire de demander explicitement au système de relire la table de partition du périphérique. Cela peut être réalisé avec l'option `--rereadpt` de la commande `blockdev` :

```
blockdev --rereadpt périphérique
```

où `périphérique` est le fichier spécial de périphérique dont la table de partition doit être relue.

La commande `mount` peut prendre diverses options pour le montage des systèmes de fichiers. Par exemple, elle permet de monter des systèmes de fichiers en lecture seule, ou de monter des systèmes de fichiers placés dans des images disques. Ces options sont introduites par l'option de ligne de commande `-o`, et doivent être séparées les unes des autres par des virgules.

Par exemple, pour monter un système de fichiers ISO9660 en lecture seule, on utilisera la ligne de commande suivante :

```
mount -t iso9660 -o ro fichier base
```

Une autre option utile pour le montage des CD-ROM est sans doute l'option `session`, qui permet d'indiquer le numéro de la session à monter dans le cas des CD-ROM multisessions. Par exemple, pour monter la deuxième session d'un CD-ROM multisession, on utilisera une ligne de commande telle que celle-ci :

```
mount -t iso9660 -o ro,session=2 fichier base
```

Le système de fichiers EXT3 prend également des options supplémentaires par rapport au système de fichiers EXT2. Ces options permettent de contrôler la manière dont la journalisation des opérations sur disque est réalisée. Avec EXT3, le journal peut être utilisé pour stocker toutes les opérations concernant la structure de données même du système de fichiers (c'est-à-dire ce que l'on appelle les « métadonnées » du système de fichiers) et les opérations concernant les données des fichiers elles-mêmes. La différence est importante et il faut bien la comprendre. Si l'on choisit de ne journaliser que les métadonnées du système de fichiers, les informations concernant les répertoires,

les fichiers et les droits d'accès seront toujours dans un état cohérent. En revanche, les données stockées dans les fichiers eux-mêmes peuvent être a priori fausses à la suite d'un redémarrage imprévu. Si, en revanche, on décide de stocker également les informations concernant les données des fichiers dans le journal, le contenu des fichiers sera également garanti, au détriment d'une perte de performances notable. Le mode de fonctionnement à utiliser est spécifié à l'aide de l'option `data` du système de fichiers, qui doit donc être fixée lors de l'opération de montage. Cette option peut prendre l'une des trois valeurs suivantes :

- `journal`, qui permet d'effectuer une journalisation complète des métadonnées et des données des fichiers. Il est donc garanti que le contenu des fichiers est toujours cohérent, tout comme l'est le système de fichiers. Cette option procure le maximum de sécurité, mais c'est également celle qui pénalise le plus les performances du système (les données sont écrites deux fois, une fois dans le journal, et une fois sur disque) ;
- `ordered`, qui est l'option par défaut et qui permet de ne journaliser que les métadonnées, mais qui garantit également que les tampons d'écriture sont vidés avant chaque journalisation d'une opération disque. Tout comme avec l'option `journal`, il est garanti que le système de fichiers est dans un état cohérent. Les données des fichiers seront également cohérentes avec les structures de données du système de fichiers. Cependant, rien ne garantit que le contenu des fichiers sera lui aussi cohérent en interne, car même si les données sont écrites avant toute modification du système de fichiers, aucun contrôle n'est effectué pour que les écritures soient réalisées dans l'ordre dans lequel les applications les ont effectuées. Cela dit, les performances sont meilleures qu'avec l'option `journal`, tout en garantissant une sécurité quasi totale des fichiers ;
- `writeback`, qui permet de ne journaliser que les métadonnées du système de fichiers. Le contenu des fichiers peut donc être incorrect, voire même contenir des données aléatoires à la suite d'un arrêt brutal du système, mais le système de fichiers est toujours dans un état correct. Cette option permet donc simplement de rendre facultatives la vérification et la réparation des systèmes de fichiers EXT2 au redémarrage. Les performances sont quasiment aussi bonnes que pour le système de fichiers EXT2.

Enfin, il est possible de monter un système de fichiers plusieurs fois, éventuellement avec des options différentes, dans différents points de montage. De même, il est possible de monter une partie d'un système de fichiers seulement dans un autre répertoire, par exemple pour réaliser un raccourci vers un sous ensemble du système de fichiers hôte. Vous pouvez consulter la page de manuel de la commande `mount` pour obtenir plus de détail à ce sujet.

Vous pourrez trouver la liste des autres options acceptées par `mount` et par les systèmes de fichiers dans la page de manuel `mount`.

VI-F-3 - Démontage des systèmes de fichiers

Les systèmes de fichiers sont gérés de manière très efficace par les systèmes d'exploitation. Des mécanismes de copie en mémoire des données sont utilisés afin d'accélérer leur lecture, et les écritures peuvent être différées pour être regroupées et ainsi optimiser les transferts. De ce fait, l'état du système de fichiers sur le support n'est quasiment jamais exactement en phase avec celui du système de fichiers en mémoire lorsqu'il est en cours d'utilisation. Il est donc nécessaire de signaler au système d'exploitation que l'on désire l'arrêter avant de couper le courant, ou que l'on désire retirer un lecteur amovible avant de le faire, afin qu'il puisse effectuer les synchronisations nécessaires. Ne pas le faire risquerait de provoquer des pertes de données irrémédiables. Cette opération s'appelle simplement le démontage.

Note : Bien entendu, les commandes d'arrêt du système se chargent (entre autres) de démonter tous les systèmes de fichiers avant d'éteindre l'ordinateur.

La commande permettant de démonter un système de fichiers est beaucoup plus simple que celle permettant de les monter, car aucune option n'est nécessaire. Il suffit en effet d'exécuter l'une des commandes suivantes :

```
umount fichier
```

ou :

```
umount base
```

Dans ces commandes, `fichier` représente le fichier contenant le système de fichiers à démonter, et `base` est le répertoire dans lequel ce système de fichiers est monté. On peut utiliser l'un ou l'autre de ces paramètres, la commande `umount` se débrouillera pour retrouver l'autre automatiquement. On notera qu'il est impossible de démonter un système de fichiers qui est en cours d'utilisation par quelqu'un. En particulier, il ne faut pas être dans le répertoire servant de point de montage pour pouvoir démonter un système de fichiers, car dans ce cas on est en train de l'utiliser. Faites bien attention à l'orthographe de la commande `umount`, elle a perdu son 'n' depuis bien longtemps déjà, et on ne l'a jamais retrouvé. Si vous savez où il se trouve, faites-le savoir.

Pour les supports de systèmes de fichiers amovibles, le démontage du système de fichiers peut ne pas être suffisant. En effet, il peut être nécessaire d'arrêter le périphérique correctement avant de l'éjecter. C'est en particulier le cas pour les systèmes de fichiers sur les clefs USB par exemple. Cette opération peut être réalisée à l'aide de la commande `eject` :

```
eject périphérique
```

où `périphérique` est le fichier spécial du périphérique à éjecter. Cette commande réalise également l'opération de démontage sur les systèmes de fichiers montés avant d'éjecter le disque.

VI-F-4 - Vérification des systèmes de fichiers

La vérification des systèmes de fichiers est une opération que l'on ne devrait jamais avoir à faire. Il y a plusieurs raisons à cela. Premièrement, si l'on arrête le système correctement avant d'éteindre la machine, et si l'on démonte bien les systèmes de fichiers avant de retirer les lecteurs amovibles, les systèmes de fichiers sont normalement toujours dans un état correct. Deuxièmement, les systèmes de fichiers Unix sont réputés pour être très fiables. Troisièmement, une vérification périodique est faite par le système au bout d'un certain nombre de démarrages. Quatrièmement, si un système de fichiers n'est pas démonté correctement avant l'arrêt du système, celui-ci sera vérifié automatiquement au démarrage suivant, ce qui fait qu'il n'y a pas lieu de le faire soi-même. Enfin, pour les systèmes de fichiers journalisés tels que EXT3, JFS ou ReiserFS, cette opération peut être réalisée très rapidement à l'aide des informations qui sont stockées dans le journal des transactions du système de fichiers.

Toutefois, même le meilleur système du monde ne saurait être à l'abri des secteurs défectueux du disque dur sur lequel il est installé. Il est donc parfois nécessaire d'effectuer une vérification manuelle des systèmes de fichiers, et il faut savoir le faire même quand plus rien ne fonctionne.

Un système de fichiers ne se manipule que lorsqu'il est démonté. Cela pose évidemment quelques problèmes pour le système de fichiers racine, puisqu'on ne peut pas accéder aux outils de vérification sans le monter. Pour ce système de fichiers, il n'y a donc que deux possibilités :

- soit on utilise une disquette de démarrage contenant les outils de vérification et de réparation des systèmes de fichiers ;
- soit on monte le système de fichiers racine en lecture seule.

La deuxième solution est la seule réalisable si l'on ne dispose pas de disquette de démarrage. Par conséquent, c'est cette méthode qui sera décrite ici.

La première étape consiste à passer en mode mono utilisateur, afin de s'assurer que personne ni aucun programme n'accède au système de fichiers racine en écriture. Pour cela, il suffit de taper la commande suivante :

```
init 1
```

qui fait passer le système dans le niveau d'exécution 1. On peut également passer le paramètre `single` au noyau lors de l'amorçage du système, comme il l'a été expliqué dans la section précédente. Ensuite, il faut s'assurer que le système de fichiers racine est en lecture seule, ce qui se fait avec la commande suivante :

```
mount -n -o remount,ro /
```

L'option `remount` permet de démonter et de remonter le système de fichiers racine, et l'option `ro` indique qu'il doit être remonté en lecture seule (« `ro` » signifie « Read Only »). Les options sont séparées par des virgules (attention, il ne faut pas insérer d'espace). De plus, l'option `-n` indique à `mount` qu'il ne doit pas écrire dans le fichier `/etc/mtab` lorsqu'il aura remonté le système de fichiers, parce que ce fichier sera alors également en lecture seule et qu'on ne pourra pas y écrire. Ce fichier est utilisé par `mount` pour mémoriser les systèmes de fichiers qui sont montés, afin de pouvoir en donner la liste (ce que la commande `mount` fait lorsqu'elle est appelée sans paramètres) et de permettre à la commande `umount` de vérifier que les systèmes de fichiers à démonter ne le sont pas déjà.

Note : Normalement, le noyau monte toujours le système de fichiers racine en lecture seule lors de l'amorçage. Ce sont les scripts de démarrage du système, lancés par `init`, qui le remontent en lecture / écriture s'il est dans un état correct. Il est donc fortement probable, si votre système ne démarre plus correctement, que le système de fichiers racine soit déjà en lecture seule après un démarrage en mode de maintenance. La commande précédente n'est donc décrite qu'à titre indicatif.

Une fois le système de fichiers racine monté en lecture seule, on peut utiliser le programme `fsck` afin de le vérifier et éventuellement le réparer. En réalité, ce programme ne fait rien d'autre que d'appeler un programme spécifique pour chaque système de fichiers, de la même manière que `mkfs` appelle des programmes spécifiques pour les créer. Par exemple, pour les systèmes de fichiers EXT2 et EXT3, `fsck` appelle le programme `fsck.ext` (qui est un lien vers `e2fsck`).

La ligne de commande à utiliser pour vérifier un système de fichiers avec `fsck` est la suivante :

```
fsck -a fichier
```

où `fichier` est le fichier spécial du périphérique ou le fichier image contenant le système de fichiers à vérifier. Il faut donc, généralement, spécifier la partition sur laquelle le système de fichiers racine se trouve. L'option `-a` demande à `fsck` d'effectuer les éventuelles corrections automatiquement en cas d'erreur sur le système de fichiers ainsi vérifié, sans confirmation de la part de l'utilisateur.

Il est également possible de demander la vérification de tous les systèmes de fichiers enregistrés dans le fichier de configuration `/etc/fstab`. Ce fichier contient la liste des systèmes de fichiers les plus utilisés et leurs options respectives. Il suffit donc d'ajouter l'option `-A` :

```
fsck -a -A
```

Il n'est évidemment plus nécessaire de spécifier le fichier spécial du périphérique contenant le système de fichiers à vérifier, puisque cette information est enregistrée dans le fichier `/etc/fstab`. La syntaxe de ce fichier sera décrite dans la section suivante.

Si le disque dur contient des secteurs défectueux, il peut être nécessaire de les marquer comme tels dans les structures du système de fichiers afin de ne pas les utiliser par la suite. De manière générale, la recherche de ces blocs peut être faite à l'aide du programme `badblocks`. Cette commande effectue un test de lecture de tous les blocs du disque sur lequel le système de fichiers se trouve, et génère une liste des blocs défectueux. Vous pouvez l'appeler directement et fournir cette liste au programme `e2fsck` à l'aide de son option `-l`, mais le plus simple est encore de demander à `e2fsck` d'appeler `badblocks` lui-même. Pour cela, il suffit de lui passer l'option `-c`, ce qui se fait en faisant précéder cette option d'un double-tiret dans la ligne de commande de `fsck` :

```
fsck -a -- -c périphérique
```

Note : L'option `-c` est spécifique à `e2fsck` et peut ne pas fonctionner avec d'autres systèmes de fichiers. En particulier, certains systèmes de fichiers ne sont pas capables de gérer correctement les blocs défectueux des disques durs. C'est le cas du système de fichiers ReiserFS.

Le programme `badblocks` peut également effectuer un test d'écriture sur le disque dur, si on lui communique l'option `-w`. Il va de soi que ce type de test est destructif, car toutes les données du disque sont alors écrasées par des motifs particuliers. Il ne faut donc JAMAIS utiliser cette option sur un système de fichiers contenant des données !

De manière générale, il vaut mieux prévenir que guérir, aussi est-il recommandé d'utiliser la commande `badblocks` au moins une fois avant d'utiliser un système de fichiers. Cette vérification peut être réalisée de manière automatique lors de la création du système de fichiers à l'aide de l'option `-c` de la commande `mke2fs`.

Une fois que vous aurez terminé la vérification du système de fichiers, vous pourrez le remonter en lecture et écriture avec la commande suivante :

```
mount -n -o remount,rw /
```

Cette commande est similaire à celle que l'on a vue pour monter le système de fichiers en lecture seule, à ceci près que l'option `rw` est utilisée à la place de l'option `ro`. Cette option permet de remonter le système de fichiers en lecture et en écriture (« `rw` » est l'abréviation de l'anglais « `Read Write` »).

VI-F-5 - Configuration du montage des systèmes de fichiers

Le montage des systèmes de fichiers peut devenir très vite une opération assez fastidieuse. Heureusement, elle peut être automatisée au démarrage pour les systèmes de fichiers situés sur les disques fixes, et simplifiée pour les systèmes de fichiers amovibles. Pour cela, il faut enregistrer ces systèmes de fichiers et leurs options de montage dans le fichier de configuration `/etc/fstab`.

Ce fichier contient, entre autres, le répertoire de montage, le type du système de fichiers et le fichier de périphérique à utiliser pour chaque système de fichiers. De cette manière, il est possible d'utiliser la commande `mount` de manière simplifiée, en ne précisant que le répertoire servant de point de montage ou le fichier spécial de périphérique.

Le fichier `/etc/fstab` contient une ligne pour chaque système de fichiers enregistré. Chaque ligne contient plusieurs champs séparés par des espaces. Les informations suivantes sont enregistrées dans ces champs :

- le fichier spécial permettant d'accéder au système de fichiers ;
- le répertoire servant de point de montage par défaut ;
- le type du système de fichiers ;
- les options de montage pour ce système de fichiers ;
- un entier indiquant si le système de fichiers doit être sauvegardé ;
- un entier indiquant l'ordre que ce système de fichiers doit avoir dans la liste des systèmes de fichiers à vérifier.

Grâce à ces informations, l'emploi de la commande `mount` est plus simple :

```
mount périphérique
```

ou :

```
mount répertoire
```

où `périphérique` est le fichier spécial de périphérique contenant le système de fichiers à monter, et `répertoire` est le répertoire servant de point de montage indiqué dans le fichier `/etc/fstab`. Il est possible d'utiliser indifféremment le fichier spécial de périphérique ou le répertoire du point de montage.

Le type du système de fichiers est l'un des types disponibles acceptés par la commande `mount`. Consultez la page de manuel de cette commande pour plus de renseignements à ce sujet. Les principales options disponibles pour le montage sont les suivantes :

- l'option `defaults`, qui permet de choisir les options par défaut pour ce système de fichiers ;
- l'option `auto`, qui permet de faire en sorte que le système de fichiers soit monté automatiquement au démarrage du système ;
- l'option `user`, qui permet d'autoriser le montage de ce système de fichiers par les utilisateurs ;

- l'option `ro`, qui permet de monter le système de fichiers en lecture seule ;
- l'option `rw`, qui permet de monter le système de fichiers en lecture et écriture ;
- l'option `exec`, qui permet d'autoriser l'exécution des fichiers exécutables sur ce système de fichiers si celui-ci ne supporte pas la notion de droit d'exécution ;
- l'option `acl`, qui permet d'autoriser l'utilisation des ACL (« Access Control List ») pour fixer les droits des utilisateurs sur les fichiers ;
- l'option `uid=utilisateur`, qui permet de spécifier le numéro utilisateur de l'utilisateur propriétaire du répertoire racine de ce système de fichiers ;
- l'option `gid=groupe`, qui permet de spécifier le numéro groupe du groupe d'utilisateurs auquel le répertoire racine du système de fichiers appartient ;
- l'option `mode=valeur`, qui permet de fixer les droits sur le répertoire racine du système de fichiers à monter. La valeur `valeur` est donnée en octal ;
- l'option `umask=valeur`, qui permet de fixer les droits sur les fichiers qui ne sont pas gérés par le système de fichiers. La valeur `valeur` est donnée en octal ;
- les options `codepage=cp` et `iocharset=charset`, qui permettent de fixer les tables de conversion des caractères pour les systèmes de fichiers pour lesquels les noms de fichiers doivent être transcodés.

Nous allons détailler un peu quelques-unes de ces options.

Par défaut, les utilisateurs n'ont pas le droit de monter et de démonter les systèmes de fichiers. L'option `user` permet de désactiver cette protection. Elle peut être utile pour permettre le montage et le démontage des disquettes et des CD-ROM. De même, l'exécution des fichiers exécutables n'est par défaut pas autorisée sur les systèmes de fichiers. Cette restriction permet d'éviter l'exécution de programmes placés sur des systèmes de fichiers de systèmes d'exploitation différents. Elle peut être levée grâce à l'option `exec`.

Tous les systèmes de fichiers disposant de l'option `auto` seront montés automatiquement au démarrage du système par la commande `mount -a`. Les autres systèmes de fichiers sont montables manuellement, avec les autres options indiquées dans le fichier `/etc/fstab`.

Les options `ro` et `rw` permettent d'indiquer à `mount` si le système de fichiers doit être monté en lecture seule ou en lecture et écriture. Les systèmes de fichiers devant être réparés doivent être montés en lecture seule si l'on ne peut pas les démonter (c'est le cas notamment du système de fichiers racine). Il en va de même pour les CD-ROM, car on ne peut bien entendu pas écrire dessus.

Les options `uid` et `gid` permettent de spécifier le propriétaire et le groupe du répertoire racine du système de fichiers à monter. Par défaut, c'est l'utilisateur `root` qui devient propriétaire de ce système de fichiers.

L'option `mode` permet de spécifier les droits d'accès sur tous les fichiers du système de fichiers à monter. L'option `umask` permet quant à elle de fixer les droits qui ne sont pas gérés par le système de fichiers. Ce peut être utile pour les systèmes de fichiers FAT et FAT32. Il est ainsi possible de donner les droits de lecture et d'exécution pour les fichiers de ces systèmes avec une valeur de masque nulle. Cela permet de monter les systèmes de fichiers FAT et FAT32 de telle sorte que tous les fichiers appartiennent à l'utilisateur `root` par défaut, et de donner cependant tous les droits à tous les utilisateurs sur ces fichiers. On prendra garde à ces options, car elles permettent à quiconque d'écrire des fichiers sous le nom de `root`, et donc constituent un grave défaut dans la sécurité du système.

Les options `codepage` et `iocharset` permettent de spécifier respectivement la page de codes et le jeu de caractères utilisés pour les systèmes de fichiers qui ne stockent pas les noms de fichiers Unix nativement. En particulier, elles permettent de modifier la page de code et le jeu de caractères sélectionnés par défaut dans la configuration du noyau pour les systèmes de fichiers FAT. L'option `codepage` est utilisée pour donner la page de codes utilisée pour la conversion des noms de fichiers en noms de fichiers courts. En général, pour les systèmes français, la page de codes utilisée est la page de codes 850. Il faut donc donner à cette option la valeur `cp850`. L'option `iocharset` quant à elle est utilisée pour faire la conversion des noms de fichiers Unix en Unicode. Elle est utilisée pour les systèmes de fichiers VFAT, car ceux-ci stockent les noms de fichiers longs en Unicode. Pour la plupart des systèmes, le jeu de caractères le plus approprié est sans doute le jeu de caractères ISO8859-1. Aussi faut-il généralement donner à cette option la valeur `iso8859-1`. Cette option n'est pas nécessaire pour les systèmes de fichiers FAT purs, puisque ceux-ci ne savent pas gérer les noms de fichiers longs.

Les deux derniers champs de `/etc/fstab` spécifient des options pour des programmes annexes. L'avant-dernier contrôle le comportement du programme de sauvegarde dump, et le dernier celui du programme de vérification de système de fichiers fsck. Consultez les pages de manuel pour plus de détails à ce sujet.

Si vous disposez de systèmes de fichiers FAT ou FAT32, vous pourrez monter ces partitions automatiquement lors du démarrage du système. Comme les systèmes de fichiers basés sur la FAT ne peuvent pas gérer les droits des utilisateurs, vous allez devoir faire un choix pour fixer ces droits à une valeur raisonnable. Vous pouvez par exemple donner le droit de lecture à tous les utilisateurs, mais le droit d'écriture uniquement à l'administrateur système. La ligne à ajouter dans le fichier `/etc/fstab` sera alors la suivante :

```
/dev/partition  répertoire  vfat  auto,exec,codepage=cp850,iocharset=iso8859-1  0  0
```

où `partition` est la partition contenant le système de fichiers FAT, et `répertoire` est le répertoire servant de point de montage pour cette partition. Cette ligne permettra de monter automatiquement ce système de fichiers en tant que FAT32 au démarrage du système. Les fichiers binaires seront exécutables, bien qu'ils ne soient pas stockés sur un système de fichiers EXT2. Si vous voulez laisser les droits d'écriture aux utilisateurs, vous pouvez utiliser la ligne suivante à la place de celle indiquée ci-dessus :

```
/dev/partition  répertoire  vfat  auto,codepage=cp850,iocharset=iso8859-1,umask=0  0  0
```

Cette ligne permet de monter le système de fichiers FAT en laissant les droits d'exécution et d'écriture aux utilisateurs. Cependant, aucun fichier exécutable de ce système de fichiers ne pourra être lancé, car l'option `exec` n'a pas été précisée.

Par ailleurs, certaines distributions spécifient des options incorrectes pour le système de fichiers `/dev/pts/` dans le fichier `/etc/fstab`. Veuillez vous assurer que la ligne utilisée pour ce système de fichiers est bien identique à la ligne suivante :

```
none  /dev/pts  devpts  auto,gid=5,mode=620  0  0
```

Si ce n'est pas le cas, certains émulateurs de terminaux développés récemment ne fonctionneront pas correctement. Le système de fichiers `/dev/pts/` est en effet un système de fichiers virtuel, géré directement par le noyau, dans lequel des fichiers spéciaux de périphériques utilisés par les émulateurs de terminaux sont placés. Si les droits ne sont pas correctement fixés sur le répertoire racine de ce système de fichiers, les émulateurs de terminaux utilisant cette fonctionnalité ne pourront se connecter au système que sous le compte root. Il faut donc impérativement corriger cette ligne, si vous voulez que les utilisateurs normaux puissent utiliser ces émulateurs. Notez également qu'il faut que tout le monde ait les droits d'écriture et de lecture sur le fichier spécial de périphérique `/dev/ptmx` pour que les utilisateurs non privilégiés puissent utiliser ce système de fichiers virtuel.

Vous devrez également ajouter le système de fichiers virtuel `/dev/shm/` dans votre fichier `/etc/fstab`. Ce système de fichiers permet aux applications qui le désirent d'utiliser des segments de mémoire partagée de manière compatible avec la norme POSIX, par exemple pour réaliser des communications interprocessus. Ce système de fichiers permet en effet de créer des fichiers qui sont directement stockés dans la mémoire virtuelle et qui peuvent être ouverts par plusieurs processus simultanément, ce qui permet d'utiliser les fonctionnalités classiques de partage de fichiers pour partager des zones de mémoire entre plusieurs processus. Ce type de communication peut être utilisé par des processus courants, aussi faut-il vous assurer que la ligne suivante se trouve bien dans le fichier `fstab` :

```
tmpfs  /dev/shm  tmpfs  defaults  0  0
```

Bien entendu, vous devrez créer le point de montage `/dev/shm/` si celui-ci n'existe pas avant de monter ce système de fichiers virtuels.

Vous pouvez monter les systèmes de fichiers virtuels du noyau, qui vous permettront d'obtenir des informations sur votre système et d'en modifier le comportement, en ajoutant les deux lignes suivantes dans le fichier `/etc/fstab` :

```
proc  /proc  proc  defaults  0  0
```

```
sysfs /sys sysfs defaults 0 0
```

La première ligne monte le système de fichiers `/proc/`, qui contient tous les paramètres du noyau et permet de paramétrer son comportement, et la deuxième contient le système de fichiers `/sys/`, qui contient une vue de tous les objets systèmes gérés par le noyau. Ce système de fichiers n'est disponible qu'avec les versions 2.6 et plus du noyau. Ces deux lignes ne sont généralement pas nécessaires, car ces systèmes de fichiers sont montés automatiquement dans les scripts d'initialisation des distributions.

Enfin, si vous utilisez des périphériques USB, il vous faudra également monter le système de fichiers virtuel `/proc/bus/usb/` en ajoutant la ligne suivante dans le fichier `fstab` :

```
none /proc/bus/usb usbfs defaults 0 0
```

Cette ligne doit être placée après celle qui effectue le montage du système de fichiers virtuel `/proc/`.

VI-F-6 - Montage des systèmes de fichiers à la demande

Les systèmes de fichiers sur les supports amovibles ne peuvent pas être montés automatiquement au démarrage du système, étant donné qu'ils ne sont pas forcément présents à ce moment. Les entrées qui les décrivent dans le fichier `fstab` doivent donc utiliser l'option `noauto`. De ce fait, l'utilisation des systèmes de fichiers stockés sur les supports amovibles nécessite toujours un montage et un démontage manuel, et même si ces opérations sont simplifiées, elles restent gênantes.

Il est toutefois possible de faire en sorte que ces systèmes de fichiers soient montés à la demande, c'est-à-dire dès qu'un accès est réalisé dans les répertoires de montage dédiés à ces systèmes par un quelconque programme. Ainsi, il n'est plus nécessaire de monter ces systèmes de fichiers manuellement. Le démontage quant à lui peut être réalisé automatiquement après un temps d'inactivité, ou manuellement, lorsqu'on veut retirer le lecteur amovible. Notez que le démontage reste vital si le système de fichiers est accédé en écriture, car des données pourraient être perdues si l'on supprime le lecteur avant qu'elles ne soient écrites.

Les opérations de montage et de démontage sont réalisées par un démon, que le noyau appelle automatiquement dès qu'une opération susceptible de concerner un système de fichiers amovible se produit. Ce démon se nomme `automount`, et il s'appuie donc sur une fonctionnalité dédiée du noyau Linux. Pour que le montage automatique des systèmes de fichiers fonctionne, il faut activer soit l'option « Kernel automounter support », soit l'option « Kernel automounter version 4 support (also supports v3) » du menu « File systems » du programme de configuration du noyau.

Note : Un démon est un processus qui tourne en arrière-plan dans le système. Les démons fonctionnent souvent dans le compte `root` et offrent des services de base que les autres programmes utilisent. Le terme « démon » provient de la traduction littérale « `daemon` », ce qui signifie « ange » en réalité. Le démon se place donc en tant qu'intermédiaire entre Dieu (c'est-à-dire le noyau) et les hommes (c'est-à-dire les applications normales). Le terme « `daemon` » a été ensuite éclairci et défini comme l'acronyme de l'anglais « Disk And Execution MONitor ».

Le démon `automount` peut surveiller plusieurs points de montage, s'ils sont tous placés dans le même répertoire. Le noyau surveille simplement les accès sur les sous-répertoires de ce répertoire, et signale à `automount` toutes les opérations susceptibles de nécessiter un montage de système de fichiers.

Les informations spécifiques aux points de montage doivent être stockées dans un fichier de configuration, que l'on doit fournir à `automount` en ligne de commande. Cette ligne de commande est donc typiquement de la forme suivante :

```
automount -t durée repertoire file fichier
```

où `durée` est la durée d'inactivité avant démontage automatique du système de fichiers, `repertoire` est le répertoire dont les sous-répertoires seront utilisés comme points de montage des systèmes de fichiers (il s'agit généralement du répertoire `/mnt/`), et `fichier` est le fichier de configuration pour les points de montage placés dans ce répertoire.

La durée d'inactivité doit être spécifiée en secondes, et la valeur par défaut est de cinq minutes. Pour désactiver le démontage automatique, il suffit de spécifier une valeur nulle.

Note : automount est capable d'utiliser d'autres méthodes de paramétrage des systèmes de fichiers amovibles que des fichiers de configuration, principalement utilisées dans le contexte du montage automatique des systèmes de fichiers en réseau. Ces méthodes de paramétrage ne seront toutefois pas décrites ici, vous pouvez consulter la page de manuel du démon automount pour plus de détails à ce sujet.

Les fichiers de configuration pour automount sont généralement placés dans le répertoire `/etc/`, et portent un nom de la forme « `auto.nature` », où `nature` est la nature des systèmes de fichiers décrits par ce fichier. Par exemple, le fichier de configuration pour les lecteurs amovibles locaux pourrait s'appeler `/etc/auto.local`. Ce fichier de configuration est constitué de plusieurs lignes, à raison d'une ligne par point de montage. Chaque ligne utilise la syntaxe suivante :

```
répertoire -options périphérique
```

où `répertoire` est le sous-répertoire du répertoire spécifié en ligne de commande à automount et devant servir de point de montage, `options` est la liste des options de montage du système de fichiers, séparées par des virgules, et `périphérique` est la localisation du système de fichiers. Pour les lecteurs amovibles, il s'agit du fichier spécial de périphérique permettant d'y accéder, précédé du caractère `'`.

Par exemple, pour réaliser le montage automatique du lecteur de CD-ROM et du lecteur de disquettes, vous devez ajouter deux lignes telles que celles-ci dans le fichier de configuration d'automount :

```
cdrom -fstype=iso9660,ro :/dev/cdrom
floppy -fstype=auto,umask=0 :/dev/fd0
```

Vous noterez que les options sont précédées d'un tiret (`'`), et que les fichiers spéciaux de périphérique sont précédés d'un deux-points (`:'`). Une fois ce fichier défini, vous n'aurez plus qu'à lancer automount avec la ligne de commande suivante :

```
automount -t 60 /mnt file /etc/auto.local
```

En général, les distributions lancent automatiquement le démon automount au démarrage, via le script d'initialisation `rc.autofs`. Ce script utilise le fichier de configuration `/etc/auto.master` pour déterminer les noms des fichiers de configuration à fournir en paramètre aux démons automount, ainsi que pour déterminer les options avec lesquelles ils doivent être lancés. Ce fichier est constitué de lignes, à raison d'une ligne par instance du démon automount qui doit être lancée. Le format de ces lignes est le suivant :

```
répertoire fichier options
```

où `répertoire` est le répertoire spécifié en ligne de commande à automount, `fichier` est le fichier de configuration, et `option` est la liste des options à lui fournir. Par exemple, pour lancer automatiquement automount comme dans l'exemple précédent, la ligne suivante devrait être ajoutée dans le fichier `/etc/auto.master` :

```
/mnt /etc/auto.local -t 60
```

Note : Vous noterez que lorsque les systèmes de fichiers ne sont pas montés, le répertoire contenant leurs points de montage est vide. De ce fait, les chemins sur ces répertoires devront être saisis explicitement pour forcer le montage des systèmes de fichiers. Cela peut être gênant lorsque l'on utilise les fonctionnalités de complétion automatique des shells, ou lorsqu'on cherche à localiser un fichier dans un de ces répertoires avec un gestionnaire de fichiers ou une boîte de dialogue d'un programme graphique. Une solution pour résoudre ce problème est de forcer une référence aux points de montage. Pour cela, il suffit de définir des liens symboliques sur les points de montage, et d'utiliser ces liens symboliques vers ces points de montage. Par exemple, on pourra définir les

liens symboliques /cdrom et /floppy pour référencer les points de montage /mnt/cdrom et /mnt/floppy, et laisser automount monter les systèmes de fichiers relatifs et créer les répertoires des points de montage dès qu'on accèdera aux liens symboliques.

VI-G - Gestion des volumes

Linux fournit des mécanismes très puissants permettant de créer des fichiers spéciaux de périphérique de type bloc virtuels, et d'indiquer les données qui seront accédées au travers de ces fichiers. Cela permet de manipuler des systèmes de fichiers stockés sur un fichier. Linux permet également de réaliser des agrégats de volumes. Ces agrégats peuvent ensuite être accédés via différentes techniques, par exemple pour réaliser de la réplication de données ou tout simplement pour simuler un périphérique de très grande taille. Ils permettent également de réaliser à la volée des opérations de chiffrement sur les données écrites, et de les déchiffrer en lecture. Ainsi, ces mécanismes d'agrégation sont utilisés pour gérer les volumes de manière beaucoup plus souple que les mécanismes de partitionnement classiques et pour créer des systèmes de fichiers chiffrés à l'aide des algorithmes cryptographiques du noyau. Nous allons présenter tous ces mécanismes dans les sections suivantes.

VI-G-1 - Gestion des fichiers images

Linux permet de stocker des systèmes de fichiers dans des fichiers image et de les manipuler de manière transparente, grâce à un mécanisme de périphériques de type bloc virtuel. Le principe est d'associer le fichier à l'un de ces périphériques, et d'utiliser simplement le fichier spécial de périphérique correspondant comme un fichier spécial de périphérique de type bloc classique. Il est ainsi possible de construire une image de systèmes de fichiers de manière tout à fait transparente, et d'accéder aux données stockées dans une image de CD-ROM de manière tout aussi facile.

Les fichiers spéciaux de périphérique utilisés pour manipuler les fichiers images sont les fichiers /dev/loopX, où X est un numéro allant de 0 à 7. Ces fichiers ne sont utilisables que si l'option « Loopback device support » du menu « Block devices » a été activée dans la configuration du noyau.

L'association entre le périphérique de type bloc virtuel et le support des données peut être réalisée à l'aide de la commande losetup. La syntaxe la plus simple de cette commande est la suivante :

```
losetup /dev/loopX fichier
```

où loopX est le fichier spécial de périphérique de type bloc virtuel à utiliser, et fichier est le fichier dans lequel les données de ce périphérique doivent être stockées. Une fois cette association faite, on peut utiliser ce périphérique comme un véritable périphérique. Ainsi, pour créer un système de fichiers dans un fichier image de 10 Mo, on procédera comme suit :

```
dd if=/dev/zero of=image bs=1024 count=10240
losetup /dev/loop0 image
mkfs -t ext2 /dev/loop0
```

Note : La commande dd permet de copier un certain nombre de blocs, ici 10240 (option count) de 1024 octets (taille indiquée par l'option bs) du fichier spécial de périphérique /dev/zero dans le fichier image. Cette commande a déjà été présentée dans le chapitre d'installation pour réaliser une copie du secteur de boot principal du disque dur.

En réalité, les commandes de création de systèmes de fichiers peuvent également créer directement les systèmes de fichiers dans des fichiers images. La manière classique de procéder a toutefois été présentée ici pour la bonne compréhension du mécanisme de gestion des fichiers images.

Le montage d'un système de fichiers placé dans l'image disque est ensuite réalisable de manière tout à fait classique, en utilisant le fichier spécial de périphérique `/dev/loopX` adéquat dans la commande `mount`. Toutefois, cette commande permet de réaliser le montage d'une image de manière simplifiée grâce à son option `loop`. Celle-ci permet de configurer le périphérique virtuel image de manière automatique avant de réaliser le montage. Elle prend donc en paramètre le fichier spécial de périphérique virtuel à utiliser, ainsi que le chemin sur le fichier image contenant le système de fichiers. Ainsi, pour monter une image de CD-ROM, vous pouvez utiliser la commande suivante :

```
mount -t iso9660 -o ro,loop=/dev/loop0 image /mnt/cdrom
```

Lorsque l'on n'a plus besoin du périphérique de type bloc virtuel, il est possible de supprimer l'association avec le support de données simplement avec la commande suivante :

```
losetup -d /dev/loopX
```

où `X` est toujours le numéro du fichier spécial de périphérique virtuel. Lorsque l'on utilise l'option `loop` de la commande `mount`, cette opération est réalisée automatiquement par la commande `umount` et ne doit plus être effectuée manuellement.

VI-G-2 - Agrégation de volumes

Les agrégats de volumes de Linux permettent de construire un périphérique virtuel constitué d'un certain nombre de volumes de données. Ceux-ci peuvent être regroupés pour réaliser un volume de grande taille, ou mis en parallèle afin de répliquer les données sur plusieurs disques par exemple. Il est donc possible, grâce à cette fonctionnalité, de regrouper plusieurs disques durs ou plusieurs partitions en une seule partition logique, ou au contraire de subdiviser de manière fine des partitions ou des disques existants. De même, il est possible de prendre en charge au niveau logiciel les fonctionnalités RAID normalement disponibles uniquement avec du matériel spécialisé. Nous ne présenterons toutefois pas ces fonctionnalités ici.

La réalisation des agrégats de volumes se fait à l'aide de l'outil `dmsetup`. Cet outil peut être récupéré sur le [site de Redhat](#) s'il n'est pas fourni avec votre distribution. Son installation se fait par compilation des sources, par exécution des commandes suivantes dans le répertoire des sources :

```
./configure
```

```
make  
make install
```

La principale commande de `dmsetup` est celle qui permet de créer un nouvel agrégat. Cela se fait simplement grâce à l'option `create` :

```
dmsetup create agrégat définition
```

où `agrégat` est le nom de l'agrégat et `définition` est le nom d'un fichier contenant la définition de cet agrégat. Si vous ne fournissez pas ce paramètre, `dmsetup` attendra que vous le définissiez en ligne de commande.

La définition d'un agrégat se fait de manière générale à l'aide de plusieurs lignes, à raison d'une ligne par composante de l'agrégat. Chaque ligne a la forme suivante :

```
début taille méthode arguments
```

`début` est le secteur de début de la plage de secteurs de l'agrégat qui seront stockés dans la composante. `taille` est le nombre de secteurs de cette plage. `méthode` est la méthode d'agrégation utilisée, elle est suivie par des options qui lui sont spécifiques. La méthode la plus classique est sans doute `linear`, qui permet de stocker des données de l'agrégat dans les secteurs correspondants de la composante. Dans le cas de la méthode linéaire, ces secteurs sont

spécifiés simplement avec le nom du fichier spécial de périphérique contenant les secteurs, et le numéro du premier secteur à partir duquel la composante sera stockée.

Par exemple, pour créer un agrégat des deux partitions `/dev/hda2` et `/dev/hdb1`, respectivement de tailles 16579080 et 59665410 secteurs, il suffit d'utiliser les deux lignes suivantes :

```
0 16579080 linear /dev/hda2 0
16579080 59665410 linear /dev/hdb1 0
```

On constatera que le secteur de départ dans l'agrégat de la deuxième composante est le secteur suivant le dernier secteur défini par la première composante. De manière générale, il n'est pas possible de définir des trous dans un agrégat. La totalité des deux partitions est utilisée dans cet exemple, pour créer un volume de taille totale 40 Go environ (pour rappel, un secteur de disque dur fait 512 octets, les deux partitions de cet exemple font donc environ 8 et 32 Go).

Une fois l'agrégat créé, un fichier spécial de périphérique portant le nom de l'agrégat doit apparaître dans le répertoire `/dev/mapper/`. Ce fichier spécial de périphérique peut être utilisé comme n'importe quel fichier spécial de périphérique, pour y créer un système de fichiers, pour le monter et pour le démonter. Par exemple, pour créer un agrégat `group0` avec les définitions précédentes, il suffit d'exécuter la commande suivante :

```
dmsetup create group0 group0.def
```

où `group0.def` est le fichier contenant les deux lignes définissant les composantes vues ci-dessus. La création et le montage du système de fichiers se font ensuite avec les commandes suivantes :

```
mke2fs /dev/mapper/group0
mount /dev/mapper/group0 /mnt/group
```

où `/mnt/group/` est le nom du répertoire où le système de fichiers agrégé doit être monté.

Lorsqu'un agrégat n'est plus utilisé, il peut être supprimé à l'aide de l'option `remove` de `dmsetup`

```
dmsetup remove agrégat
```

Cela ne supprime bien entendu pas les données, mais détruit l'association entre les composantes de cet agrégat.

Note : Les fonctionnalités d'agrégation de volumes ne sont disponibles que si les options « Multiple devices driver support (RAID and LVM) », « Device mapper support » et « Crypt target support » du noyau ont été activées dans le menu « Multidevice support (RAID and LVM) » du programme de configuration du noyau.

`dmsetup` utilise le fichier spécial de périphérique `/dev/mapper/control` pour communiquer avec le noyau. Vous devrez donc créer ce fichier si votre distribution ne le fournit pas. Pour cela, vous pouvez exécuter le script `devmap_mknot.sh` fourni dans le répertoire `scripts/` des sources de `dmsetup`. Ce script permet également de créer le répertoire `/dev/mapper/` si celui-ci n'existe pas.

VI-G-3 - Chiffrement des systèmes de fichiers

Il est possible de créer des systèmes de fichiers chiffrés afin de garantir la confidentialité des données qui y sont stockées. Ces systèmes de fichiers sont chiffrés à la volée par le noyau avec des algorithmes de chiffrement très puissants, que l'on paramètre avec une clef de chiffrement. Cette clef doit bien entendu rester secrète, faute de quoi tout ce travail de chiffrement devient inutile. Le chiffrement et le déchiffrement sont réalisés de manière complètement transparente par le noyau lors des accès aux répertoires et aux fichiers du système de fichiers, qui apparaît donc en clair lorsqu'on l'utilise. La différence est donc que le système de fichiers reste illisible pour tous les autres ordinateurs, ou pour tous les autres utilisateurs s'ils n'ont pas la clef de décryptage et que le système de fichiers n'est pas monté

(rappelons que le mécanisme des droits Unix permet à un utilisateur d'interdire aux autres utilisateurs de consulter ses données).

Pour créer un tel système de fichiers, il est nécessaire de créer un volume agrégé, et d'utiliser un algorithme de chiffrement comme algorithme d'agrégation. Ceci se fait simplement en indiquant que les composantes de l'agrégat utilisent la technique d'agrégation `crypt`. Cette technique est semblable à la technique `linear`, à ceci près qu'elles permettent de chiffrer les données à la volée. L'algorithme de chiffrement utilisé peut alors être spécifié et prendre en paramètre une clef privée, qui doit être fournie à chaque fois que l'agrégat est redéfini, faute de quoi les données resteront indéchiffrables. Pour des raisons de sécurité, un vecteur d'initialisation peut être utilisé afin de rendre plus difficiles les tentatives de décryptage par la force brute. Ce vecteur d'initialisation est ajouté à la clef de chiffrement et permet ainsi d'obtenir des données chiffrées différentes pour deux systèmes de fichiers même si les clefs utilisées sont identiques. Dans l'implémentation actuelle, le vecteur d'initialisation utilisé est le numéro du secteur chiffré, ce qui n'est pas la panacée étant donné que ce numéro est prédictible, mais c'est déjà mieux que rien.

La technique d'agrégation `crypt` prend en paramètre l'algorithme de chiffrement à utiliser, la clef privée à utiliser pour chiffrer le volume agrégé, un décalage à ajouter au numéro du secteur pour déterminer le vecteur d'initialisation, le fichier spécial de périphérique sur lequel les données chiffrées seront écrites, et le numéro du secteur de ce périphérique à partir duquel les données seront écrites. Les décalages fournis permettent de n'utiliser qu'une partie d'un périphérique de type bloc pour stocker les données chiffrées. En général, ces valeurs seront nulles, car les données sont souvent stockées sur la totalité du périphérique hôte.

Ainsi, la définition d'une composante de l'agrégat chiffré se fait selon la syntaxe suivante :

```
début taille crypt algorithme clef décalage périphérique origine
```

où `début` et `taille` définissent la composante dans l'agrégat, `algorithme` et `clef` l'algorithme de chiffrement et la clef privée utilisée, `décalage` le décalage pour le vecteur d'initialisation, `périphérique` le fichier spécial de périphérique dans lequel les données chiffrées devront être écrites, et `origine` le secteur de départ dans ce périphérique à partir duquel les données seront écrites.

Les algorithmes utilisables sont les algorithmes de chiffrement inclus dans la configuration du noyau, dont vous pourrez trouver la liste et les paramètres dans le fichier `/proc/crypto`. Pour chaque algorithme de chiffrement, vous pouvez indiquer si vous désirez utiliser un vecteur d'initialisation ou non en ajoutant un suffixe à leur nom. Le suffixe `ecb` signifie qu'il n'y a pas de vecteur d'initialisation, et le suffixe `plain` signifie que le vecteur utilisé est le numéro de secteur. Bien entendu, vous devriez utiliser `plain`, en attendant qu'une meilleure technique ne soit mise au point. Ainsi, le nom d'algorithme `des-plain` indique que l'algorithme DES doit être utilisé avec un vecteur d'initialisation. La clef de cet algorithme doit faire 64 bits (soit huit octets).

Une fois l'agrégat défini, celui-ci peut être accédé comme un périphérique de type bloc classique, via son fichier spécial de périphérique dans le répertoire `/dev/mapper/`. La création du système de fichiers se fait de la même manière que pour les autres fichiers spéciaux de type bloc. Par exemple, pour chiffrer la partition `/dev/hda2` (dont la taille est supposée être de 10000000 blocs) avec l'algorithme AES, vous devrez définir l'agrégat suivant :

```
0 10000000 crypt aes-plain clef 0 /dev/hda2 0
```

où `clef` est une clef de taille comprise entre 16 et 32 octets. Si cet agrégat a été nommée `cr0`, vous pourrez ensuite créer le système de fichiers simplement avec la commande suivante :

```
mkfs -t ext2 /dev/mapper/cr0
```

et le monter dans le répertoire `/mnt/secure/` avec une commande telle que celle-ci :

```
mount /dev/mapper/cr0 /mnt/secure
```

Enfin, il est possible d'utiliser un fichier en tant que périphérique de stockage des données chiffrées. Pour cela, il faut associer ce fichier à un fichier de type bloc virtuel à l'aide de la commande `losetup`. Ce fichier peut ensuite être utilisé pour définir un agrégat, comme n'importe quel autre fichier spécial de périphérique.

VI-H - Configuration des terminaux virtuels

Un terminal est, comme son nom l'indique, un équipement qui se trouve au bout d'une connexion à un ordinateur et qui permet de travailler sur l'ordinateur. Un terminal comprend généralement un clavier et un écran (graphique pour les terminaux X), et parfois une souris. Initialement, les terminaux étaient des périphériques passifs connectés sur un port série de l'ordinateur. Cette architecture permettait de partager une même unité centrale avec plusieurs utilisateurs. De nos jours, la plupart des terminaux sont simplement d'autres ordinateurs du réseau, qui se connectent à l'aide d'un programme que l'on appelle « émulateur de terminal ». L'émulateur de terminal se contente de simuler un terminal réel et permet de se connecter à toute machine gérant des terminaux clients. Du point de vue du système, tous les utilisateurs sont connectés via des terminaux, qu'ils soient physiques ou émulés.

Les connexions locales sont donc réalisées par l'intermédiaire d'un terminal local, la console. La console est tout simplement le périphérique prenant en charge le clavier et l'écran. Ce périphérique est géré directement par le noyau, et émule un modèle de terminal standard, le VT102. En fait, la console n'est utilisée directement en tant que terminal de login qu'en mode monousateur (c'est-à-dire en mode maintenance). En mode multiutilisateur, la console est partagée entre plusieurs terminaux dits « virtuels », parce qu'ils simulent la présence de plusieurs terminaux sur la même console. Ces terminaux virtuels se partagent le clavier et l'écran, mais seulement un de ces terminaux peut accéder à la console à chaque instant : celui qui traite les caractères saisis au clavier et qui réalise l'affichage. Pour pouvoir passer d'un terminal virtuel à un autre, il faut utiliser une séquence de touches spéciale, par exemple `ALT+Fn` (ou `CTRL+ALT+Fn` si vous êtes sous XWindow), où `Fn` est l'une des touches de fonction du clavier. Si l'on utilise cette combinaison avec la touche `F1`, on accédera au premier terminal virtuel. Avec la touche `F2`, ce sera le deuxième, avec `F3`, le troisième, etc.

Il est donc possible de simuler la présence de plusieurs écrans et claviers sur une même machine, ce qui est très pratique lorsqu'on commence à lancer plusieurs programmes en même temps (cela nécessite évidemment de se connecter plusieurs fois sur des terminaux différents). Bien entendu, la configuration du clavier et la police de caractères utilisée sont communes à tous les terminaux virtuels, puisqu'ils utilisent tous la même console.

La plupart des distributions utilisent au moins quatre terminaux virtuels, plus éventuellement un terminal pour le serveur X. Mais il est possible de définir d'autres terminaux, qu'ils soient graphiques ou non. Nous allons maintenant voir comment modifier le nombre de terminaux virtuels disponibles. L'utilisation des terminaux avec XWindow sera traitée dans le chapitre traitant de la [configuration de XWindow](#).

Chaque terminal virtuel utilise un fichier spécial de périphérique du répertoire `/dev/`. Le nom de ce fichier commence toujours par « `tty` » et est complété par le numéro du terminal. Ainsi, le fichier spécial de périphérique `/dev/tty1` correspond au premier terminal virtuel, accessible avec la combinaison de touches `CTRL+ALT+F1`, le fichier spécial de périphérique `/dev/tty2` correspond au deuxième terminal virtuel, etc. Linux peut gérer jusqu'à 64 terminaux virtuels, cependant, il est nécessaire de définir d'autres combinaisons de touches pour accéder aux terminaux 13 et suivants (puisque'il n'existe que 12 touches de fonctions). Il serait possible d'utiliser les combinaisons de touches `ALT+DROITE` et `ALT+GAUCHE` pour les atteindre, mais d'une part ce ne serait pas très pratique, et d'autre part, vous ne devriez pas avoir besoin de plus de quatre ou cinq terminaux virtuels. Nous n'utiliserons donc ici que les douze premiers terminaux virtuels.

Les terminaux virtuels sont créés par le noyau à la demande, dès qu'un processus cherche à y accéder. Ainsi, le système n'alloue les ressources utilisées pour la gestion de ces terminaux que lorsque cela est nécessaire. Les terminaux peuvent être créés par différents processus, et leur emploi n'est pas restreint à la simple connexion des utilisateurs. Par exemple, il est possible d'afficher un message sur un terminal simplement en écrivant dans son fichier spécial de périphérique. Ainsi, si vous tapez la commande suivante sous le compte `root` :

```
echo Coucou > /dev/tty11
```

la chaîne de caractères « Coucou » devrait apparaître sur le terminal virtuel 11.

En général, les terminaux virtuels sont utilisés soit pour afficher les messages du système, soit pour permettre aux utilisateurs de se connecter, soit pour XWindow. Les terminaux peuvent donc être attribués à différents programmes, selon l'emploi qu'on leur réserve. Il faudra cependant bien prendre garde au fait que les terminaux ne sont pas partageables entre tous les processus. Ainsi, on ne devra pas essayer de lancer un serveur X sur un terminal utilisé par un processus de connexion en mode texte.

Pour créer des terminaux de login, il suffit de demander au système de lancer les processus de connexion sur chaque terminal désiré. Ce travail est à la charge du processus fondamental du système : `init`. La définition des terminaux de login se fait donc dans le fichier de configuration `/etc/inittab`. Si vous regardez le contenu de ce fichier, vous trouverez quelques lignes semblables à la suivante :

```
1:2345:respawn:/sbin/getty 9600 tty1 linux
2:2345:respawn:/sbin/getty 9600 tty2 linux
etc.
```

Ces lignes indiquent à `init` que plusieurs processus `getty` doivent être lancés sur les différents terminaux virtuels. Le programme `getty` est le programme qui vous demande votre nom d'utilisateur sur les terminaux virtuels. Plus précisément, `getty` initialise le terminal, demande le nom d'utilisateur et lance le programme `login` en lui fournissant le nom saisi, afin que celui-ci demande le mot de passe de cet utilisateur.

Si vous désirez rajouter des terminaux de login à votre configuration, vous devrez donc rajouter des lignes de ce genre dans le fichier `/etc/inittab`. En fait, ces lignes sont constituées de quatre champs :

- le premier champ est le numéro de la ligne. En pratique, ce numéro doit être celui du terminal virtuel qui sera utilisé par `getty` ;
- le champ suivant (« 2345 ») contient les numéros des niveaux d'exécution dans lesquels cette ligne est valide. Ces numéros doivent être spécifiés les uns à la suite des autres, sans séparateur. Dans l'exemple donné ci-dessus, ces lignes sont valides dans les niveaux d'exécution 2 à 5 compris, c'est-à-dire tous les niveaux d'exécution classiques ;
- le troisième champ indique à `init` des options pour le lancement du programme `getty`. Dans l'exemple donné ci-dessus, le programme `getty` doit être relancé immédiatement dès qu'il se termine. Cela est le comportement désiré, puisque la terminaison de `getty` correspond à la déconnexion de l'utilisateur courant, et qu'il faut laisser la possibilité de se reconnecter aux suivants ;
- le dernier champ indique le programme à lancer et ses options de ligne de commande. Dans notre cas, il s'agit de `/sbin/getty`. Les options indiquées sont la vitesse de la ligne de communication utilisée, le nom du terminal sur lequel `getty` doit travailler et son type (ici, ce sont des terminaux de type « `linux` »).

Vous pouvez bien entendu vous baser sur les lignes existantes pour en créer de nouvelles. L'opération est très simple : il suffit de renuméroter les lignes et les terminaux virtuels utilisés. Prenez garde cependant à ne pas affecter à `getty` un terminal utilisé par XWindow. Il est recommandé d'effectuer ces modifications dans le niveau d'exécution 2 pour ne pas être gêné par XWindow.

Une fois les modifications ajoutées, vous pourrez demander à `init` de relire son fichier de configuration avec la commande suivante :

```
init Q
```

Dès lors, les nouveaux terminaux sont prêts à être utilisés.

VI-I - Configuration de la console

Comme nous venons de le voir, tous les terminaux virtuels utilisent la même console. La suite logique des opérations est donc de voir comment on réalise la configuration de celle-ci... Nous allons donc voir dans ce chapitre la manière de paramétrer le clavier et l'affichage du texte à l'écran.

Pour cela, il est nécessaire de bien comprendre les mécanismes mis en œuvre pour le traitement des codes émis par le clavier d'une part, et pour l'affichage des symboles des lettres à partir de leurs codes d'autre part. Ces mécanismes sont relativement évolués et complexes, mais permettent de paramétrer avec précision la disposition des touches du clavier, le comportement des applications et l'allure des symboles affichés à l'écran.

Le plus simple pour comprendre le fonctionnement de la console est encore de voir les différentes étapes entrant en ligne de compte de l'appui sur une touche jusqu'à l'action correspondant à cet appui. Notez bien que cette action n'est pas forcément l'affichage d'une lettre à l'écran : cela dépend de l'application qui a traité l'événement correspondant à l'appui sur la touche. Il est toutefois nécessaire de présenter auparavant quelques notions sur la manière dont les caractères sont représentés dans les programmes.

VI-I-1 - Pages de codes et Unicode

De par leur nature de calculateurs, les ordinateurs n'ont jamais été conçus pour manipuler nativement du texte. Cela signifie qu'ils n'ont aucune notion de caractère ou de symbole : pour eux, tout est numérique. Par conséquent, il a fallu trouver le moyen de représenter les caractères humains sous une forme numérique afin de pouvoir réaliser des programmes de manipulation de texte. Cette représentation est effectuée en associant à chaque caractère un numéro donné, et en travaillant directement sur ces numéros. Par exemple, le caractère 'A' est classiquement représenté par le nombre 65, la lettre 'B' par le nombre 66, etc. L'opération consistant à effectuer cette association sur chaque caractère d'un texte constitue ce que l'on appelle l'encodage du texte. Plusieurs manières de réaliser cet encodage ont été inventées, mais l'un des standards les plus utilisés est l'encodage ASCII. Lorsqu'il a été créé, l'encodage ASCII codait les caractères sur 7 bits. Depuis, il a été étendu pour utiliser 8 bits, ce qui fait que chaque caractère est dorénavant codé sur un octet. Ainsi, il est possible de représenter 256 caractères différents avec un octet, ce qui est suffisant pour toutes les langues occidentales.

Cependant, le standard ASCII initial ne spécifiait que l'encodage des caractères utilisés en anglais, tout simplement parce que les Américains parlent anglais. Évidemment, cela ne convenait pas pour les pays qui utilisent des lettres accentuées, c'est-à-dire pour quasiment tout le monde. Il a donc fallu définir d'autres conventions que celle initialement utilisée, afin d'associer des codes numériques aux caractères utilisés par les autres pays. Ces conventions constituent ce que l'on appelle les pages de codes. Chaque pays est donc susceptible d'utiliser une page de codes qui lui est spécifique. Par exemple, la page de codes 437 représente l'encodage utilisé aux États-Unis, et la 850 celle utilisée en France.

Historiquement, les pages de codes n'ont pas été immédiatement standardisées, ce qui a conduit à la prolifération de pages différentes et parfois incompatibles. Ainsi, les pages de codes utilisées en France sont les pages de codes 850 sous DOS, 1252 sous Windows, ISO 8859-1 et ISO 8859-15 sous Unix. Ces deux dernières constituent la norme actuelle, et sont celles qui doivent être utilisées de préférence. La norme ISO 8859-1 est également connue sous le nom latin-1, et la norme ISO 8859-15 sous le nom latin-0. La norme latin-0 est une extension de la latin-1, qui ne prévoyait pas le codage de certains caractères européens (comme le o e dans l'o français) et le symbole de l'Euro.

Le défaut majeur de l'ASCII et de ses dérivés est de travailler sur des nombres à 8 bits. Si le nombre de 256 caractères différents convient pour la plupart des pays occidentaux, ce n'est pas le cas de quelques autres pays, qui utilisent un nombre de symboles de très loin supérieur à 256. C'est notamment le cas du Japon et de la Chine, qui ne peuvent pas encoder tous leurs idéogrammes sur des nombres à 8 bits. Il a donc fallu introduire d'autres types d'encodages, plus riches, permettant de satisfaire aux besoins de tout le monde.

Après des tentatives infructueuses d'encodages à taille variable (un ou plusieurs octets selon le caractère codé), Unicode a été introduit et normalisé sous le nom ISO 10646. Unicode est une convention de codage universelle des caractères, qui utilise pour cela des nombres 32 bits (il existe également une version plus ancienne qui n'utilise que 16 bits). Chaque caractère est représenté par un nombre et un seul, comme pour l'ASCII. Cependant, avec ses 16 ou 32 bits, le jeu de caractères Unicode est suffisamment large pour coder tous les caractères de toutes les langues du monde. Bien entendu, tous les codes ne sont pas utilisés, et le jeu de caractères Unicode est discontinu. Pour des raisons de compatibilité, les 256 premiers caractères Unicode sont les mêmes que ceux définis dans la norme ISO 8859-1 (ce qui rend malheureusement non compatible la norme ISO 8859-15, plus complète). Les autres caractères sont affectés à d'autres plages de codes, qui sont parfaitement définies. Ainsi, l'utilisation d'Unicode permettra, à terme, de n'avoir plus qu'une seule page de codes pour tous les pays.

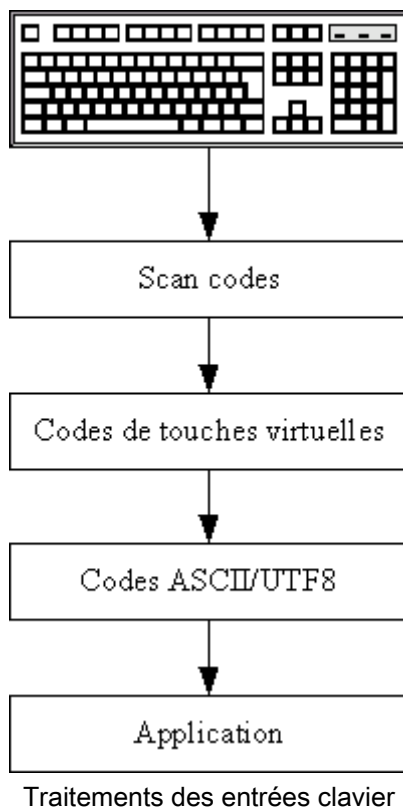
Malheureusement, Unicode est une évolution relativement récente, et la plupart des programmes travaillent encore avec des caractères 8 bits, ce qui rend l'utilisation d'Unicode prématurée. Linux, quant à lui, est capable de gérer l'Unicode. Cependant, pour des raisons d'économie de place, il ne l'utilise pas directement. Il préfère en effet utiliser l'encodage UTF-8 (abréviation de l'anglais « Unicode Transfer Format »). Cet encodage est un encodage à taille variable, qui permet d'encoder les caractères Unicode avec de un à six octets selon leur emplacement dans la page de codes Unicode.

VI-I-2 - Principe de fonctionnement du clavier

En général, les claviers envoient une série de codes à l'unité centrale lorsqu'on appuie sur une touche. Certaines touches génèrent un seul code, d'autres peuvent en produire jusqu'à une dizaine. Ces codes, que l'on appelle scancode, sont récupérés par le pilote du clavier dans le noyau de Linux, et constituent le début du traitement des saisies clavier. Les scancodes permettent normalement de déterminer avec certitude l'événement qui s'est produit, pour peu que l'on connaisse parfaitement le type de clavier utilisé. Malheureusement, ils sont spécifiques à chaque modèle de clavier, et il est difficilement concevable pour un programme de prendre en charge les particularités de tous les claviers existants. De plus, qui peut prévoir aujourd'hui combien de touches les claviers du futur auront, et quels scancodes ceux-ci utiliseront ?

Linux effectue donc un travail d'uniformisation en interprétant les scancodes et en les traduisant en d'autres codes, les keycodes. Ces codes sont toujours les mêmes, quel que soit le clavier utilisé. Les keycodes simplifient le traitement des données provenant du clavier en définissant un code de touche unique à chacune des touches du clavier. Les keycodes sont également souvent appelés les codes de touches virtuelles, car ils correspondent aux scancodes d'un clavier virtuel uniforme et commun à toutes les plates-formes. La gestion des événements clavier par l'intermédiaire des keycodes est donc beaucoup plus aisée, car il n'y a plus à se soucier ici que de ce clavier virtuel. La correspondance entre les scancodes, donc les touches physiques, et les keycodes, ou codes de touches virtuelles, est définie dans le pilote du clavier. La plupart des claviers courants sont pris en charge et, en général, peu de personnes ont besoin de modifier ce type d'information. Toutefois, afin de permettre l'intégration des claviers futurs, il est possible de compléter la table de conversion des scancodes en codes de touches virtuelles.

L'interprétation des keycodes est une opération relativement compliquée et peu commode à cause du grand nombre de dispositions de clavier existantes et, pour chaque type de clavier, des différentes déclinaisons existantes en fonction des langages des divers pays qui les utilisent. C'est pour cela qu'une autre association, qui permet de définir le comportement à obtenir pour chaque combinaison de codes de touches virtuelles, a été introduite. Cette correspondance est décrite dans un fichier que l'on appelle couramment le plan de clavier (ou keymap en anglais). Les keymaps contiennent donc, pour chaque touche ou combinaison de touches virtuelles utilisée, la définition d'une action à effectuer ou d'un code de caractère à renvoyer. Ces codes sont renvoyés en ASCII, codés selon la page de codes définie dans le plan de clavier. Cependant, certaines touches ne sont pas associées à une lettre ASCII et ne peuvent donc pas être représentées par des codes simples. C'est le cas, par exemple, des touches du curseur et des touches de suppression du clavier. Ces touches sont donc signalées aux programmes à l'aide de séquences de codes appelées les codes d'échappement. Les codes d'échappement sont en réalité des séquences de codes ASCII dont le premier est le code du caractère d'échappement Escape (dont le numéro est 27, quelle que soit la page de codes utilisée). Ces séquences sont donc utilisées typiquement pour signaler aux programmes qui les lisent qu'un traitement particulier doit être effectué, par exemple le déplacement du curseur.



Les programmes peuvent travailler à n'importe lequel des trois niveaux de traitement que l'on vient de décrire. Les programmes qui récupèrent directement les scancodes travaillent en mode raw (ce qui signifie en anglais que les données sont « brutes »). Ils n'utilisent donc pas la traduction en codes de touches virtuelles et doivent nécessairement connaître les caractéristiques physiques des claviers qu'ils veulent supporter. C'est le cas par exemple des serveurs X. Les programmes qui utilisent les keycodes travaillent dans le mode nommé medium raw ou keycode. Cependant, la plupart des programmes travaillent avec les codes issus du plan de clavier, qui sont normalisés et font bénéficier d'une plus grande portabilité (y compris avec d'autres systèmes Unix que Linux). Dans ce cas, on dit qu'ils travaillent en mode ASCII, ou xlate (abréviation de l'anglais « translate », ce qui signifie « traduction »).

Note : En fait, la console peut également travailler en Unicode, dans le mode UTF-8. Ce mode permet de travailler directement en Unicode, si vous le désirez. Cependant, vous devrez vous assurer dans ce cas que tous les programmes que vous utilisez sont capables de fonctionner avec un encodage à taille variable.

Dans tous les cas, les programmes lisent les données provenant du clavier sur la console, par l'intermédiaire du fichier spécial de périphérique `/dev/console`. En fait, ces programmes ne savent pas qu'ils lisent les données du clavier local. Pour eux, ces données semblent provenir d'un terminal VT102, et ils traitent ces données comme celles provenant de n'importe quel terminal. Cela signifie que ces données semblent provenir d'une ligne de communication, à laquelle le terminal local est connecté. Et qui dit ligne de communication, dit paramètres de transmission des informations sur la ligne ! Les caractères issus de la keymap peuvent donc subir un traitement supplémentaire, en fonction des paramètres de la ligne de communication (virtuelle) utilisée par la console. Cependant, il n'est en général pas nécessaire de modifier ces paramètres, puisque la ligne de communication utilisée est bien entendu idéale (quitte à être virtuelle, autant être idéale, non ?).

Gloups !? Que c'est compliqué ! C'est effectivement la réflexion que vous êtes en droit de vous faire. Cependant, si l'on prend un peu de distance par rapport à ce mécanisme en trois passes, on comprend son intérêt. Premièrement, la contrainte initiale est la complexité des scancodes. Sur ce point, il n'y a rien à dire. Grâce aux keycodes, il n'est plus nécessaire de se soucier du modèle de clavier utilisé. Ensuite, grâce aux keymaps, il est possible de faire l'association entre les keycodes et les lettres écrites sur les touches du clavier. Ainsi, la disposition du clavier, ainsi que les raccourcis clavier, peuvent être complètement paramétrés. Enfin, les applications qui gèrent les touches spéciales du clavier peuvent interpréter les codes d'échappement, qui sont ceux renvoyés par le terminal de la console. Dans tous les cas, les programmes qui lisent les données du clavier considèrent que ces données proviennent d'un terminal

classique. Ils n'ont donc pas besoin de faire des hypothèses sur l'origine des données, et leur programmation en est d'autant plus simple.

Passons maintenant aux travaux pratiques. La commande `kbd_mode` permet de déterminer le mode de fonctionnement du clavier. Si vous l'essayez à partir d'une console, vous verrez que le clavier est en mode ASCII. Inversement, si vous tapez cette commande à partir d'une session X, vous constaterez que XWindow utilise le clavier en mode raw, et gère donc les scancodes lui-même. Vous pouvez également visualiser les codes renvoyés dans les trois modes de fonctionnement du clavier avec la commande `showkey`. Lancée avec l'option `-s`, elle permet d'afficher les scancodes lors de l'appui sur chaque touche. Vous pourrez constater que toutes les touches ne renvoient pas le même nombre de codes. Vous pourrez utiliser l'option `-k` pour visualiser les keycodes renvoyés par le noyau lors de l'appui et du relâchement des touches. Enfin, si vous utilisez l'option `-a`, vous verrez les codes de touches ASCII. Vous pourrez constater que les touches spéciales renvoient des séquences de codes d'échappement.

Note : Notez également que certaines touches renvoient des caractères de contrôle (notés `^A` à `^Z`). Ces caractères correspondent en fait aux codes ASCII allant de 1 à 26 et ne sont pas des séquences d'échappement. Ils peuvent également être générés avec les combinaisons de touches `CTRL+A` à `CTRL+Z`. Si vous ne me croyez pas, tapez la commande `ls` dans une console et tapez la combinaison de touches `CTRL+M` (celle affectée à la touche `Entrée`). Vous verrez, c'est comme si vous aviez tapé `Entrée` !

Le programme `showkey` ne peut pas être lancé sous XWindow, parce que celui-ci gère lui-même le clavier. `showkey` se termine de lui-même au bout de dix secondes après la dernière frappe, ou lors de l'appui de la séquence de touches `CTRL+D` s'il est en mode ASCII.

VI-I-3 - Principe de fonctionnement de l'écran de la console

Il est temps maintenant de passer au mécanisme d'affichage des caractères sur l'écran. Comme vous avez pu le voir dans les paragraphes précédents, le fait d'appuyer sur une touche ne fait rien d'autre que de fournir des codes numériques aux programmes. Contrairement à ce que vous pourriez penser, les codes de ces touches ne sont pas forcément transmis directement à l'écran (heureusement, comment taperiez-vous votre mot de passe à l'abri des regards indiscrets sinon ?). En fait, l'affichage des caractères à l'écran est un des paramètres de la ligne du terminal, que l'on nomme `echo`. Pour la plupart des programmes, l'écho automatique est désactivé. Pour d'autres ce sont les applications qui, en fonction des codes qu'elles lisent sur la console, effectuent une action. Mais la majorité de ces actions nécessitent effectivement d'afficher des résultats sur le moniteur. Pour réaliser cela, les programmes écrivent les données à afficher dans le fichier spécial de périphérique de la console.

Comme nous l'avons vu, la console de Linux se comporte comme un terminal local. Cela implique que les applications qui désirent écrire des données doivent les envoyer par la ligne de communication via laquelle elles sont connectées à ce terminal. Tout comme les scancodes du clavier, les caractères envoyés à la console par les applications subissent un traitement imposé par les paramètres de la ligne de communication virtuelle utilisée. Bien entendu, ce traitement est encore une fois minimal, puisque cette ligne est idéale. Notez que les programmes n'en savent absolument rien, car, pour eux, tout se passe comme s'ils écrivaient sur un terminal réel. Quoi qu'il en soit, le flux de caractères arrive au niveau du gestionnaire de la console après ce traitement de base.

Comme nous l'avons déjà dit plus haut, la console Linux émule un terminal VT102, et recherche donc les codes de contrôle et les séquences d'échappement de ces consoles. Ces codes d'échappement sont classiquement utilisés pour effectuer des opérations sur la console, par exemple le déplacement du curseur, la prise en compte de la couleur et des attributs des caractères affichés, l'effacement de l'écran, etc. Pour information, tous les codes de la console sont décrits dans la page de manuel `console_codes`.

Les codes qui ne sont pas reconnus comme étant des codes d'échappement sont traités en tant que caractères normaux. Le gestionnaire de la console doit donc déterminer quel caractère doit être imprimé à l'écran pour chaque code fourni par l'application. Cela n'est pas une opération facile, parce que les polices de caractères, qui contiennent la définition de la représentation graphique des caractères, n'utilisent pas forcément la page de codes active dans le système. Cela signifie que les caractères définis dans la police n'apparaissent pas forcément dans le même ordre que celui de la page de codes, et pour cause : une police peut parfaitement être utilisable dans plusieurs pays différents !

Cela complique donc un peu plus les choses, puisqu'il faut utiliser une table de conversion entre la page de codes du texte à afficher et l'encodage propre de la police.

Afin de réaliser cette conversion, le gestionnaire de la console Linux commence par convertir tous les caractères qu'il reçoit en « Unicode ». Si la console est en mode UTF-8, cette opération est immédiate, car l'UTF-8 est un encodage Unicode. En revanche, si la console est en mode ASCII, cette opération nécessite l'emploi d'une table de conversion. Linux dispose de quatre tables de conversion :

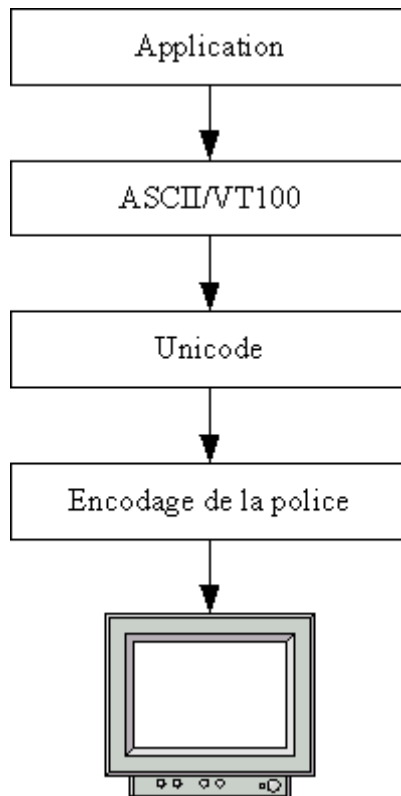
- la première table permet de faire la conversion entre la page de codes ISO 8859-1 et Unicode (cette conversion est, par définition, immédiate) ;
- la deuxième table permet de faire la conversion entre les codes graphiques des terminaux VT100 et Unicode ;
- la troisième table permet d'effectuer la conversion entre la page de codes 437 et Unicode ;
- et la quatrième et dernière table est réservée à l'usage personnel de l'utilisateur.

La dernière table est très utile, car elle permet de définir une nouvelle table de conversion si nécessaire. Par défaut, elle convertit les codes reçus en codes Unicode spéciaux, qui représentent les codes utilisés par la police de caractères chargée. Ainsi, lorsque cette table est utilisée, les codes reçus par la console seront utilisés directement pour localiser les caractères dans la police de caractères. Cela ne peut fonctionner que si les applications utilisent le même encodage que la police de caractères courante.

Linux dispose de deux jeux de caractères, nommés respectivement G0 et G1, qui permettent de sélectionner la table de conversion à utiliser. Par défaut, ces jeux de caractères pointent respectivement sur la première et la deuxième table de conversion, mais cela peut être modifié à l'aide de codes d'échappement. Seul un jeu de caractères est actif à un moment donné. Par défaut, il s'agit du jeu G0, ce qui implique que la table de conversion utilisée par défaut est la table ISO 8859-1 vers Unicode. Il est également possible de changer de jeu de caractères, à l'aide d'un code de contrôle. Ce mécanisme ne sera pas décrit plus en détail ici, car il ne nous sera pas utile.

Note : En fait, la recherche des codes de contrôle est effectuée a posteriori, après la conversion en Unicode.

Dans tous les cas, les codes envoyés par l'application sont donc convertis en Unicode (16 bits). Ils peuvent donc être reconvertis aisément dans l'encodage utilisé par la police de caractère chargée en mémoire vidéo. Par défaut, cette opération est triviale, et l'encodage cible est donc l'encodage ISO 8859-1, sauf si la quatrième table de conversion a été utilisée. Dans ce cas en effet, l'encodage cible est le même encodage que celui utilisé par l'application (tout se passe donc comme si rien ne s'était passé). Enfin, la dernière opération est tout simplement l'écriture du code dans la mémoire vidéo. Ce code est utilisé par la carte graphique comme index dans la police de caractère, et le caractère ainsi localisé est affiché.



Traitement des caractères pour l'affichage

Gloups !? Que c'est compliqué ! Ãƒa ne va pas recommencer ? Eh si ! Mais, encore une fois, ce m canisme permet de rendre ind pendantes les applications des polices de caract res. Chacun peut utiliser l'encodage qu'il d sire, et gr ce   un petit passage en Unicode, tous les caract res finissent par  tre repr sent s par le bon symbole   l' cran... En fait, le passage par Unicode donne la possibilit  de d finir les tables de conversion des polices uniquement par rapport   Unicode, et non par rapport   tous les encodages possibles que les applications peuvent utiliser. Cela permet donc l'utilisation de polices de caract res diverses et vari es, sans avoir   modifier les applications.

VI-I-4 - Configuration du clavier

Bien que les distributions modernes fournissent les outils n cessaires   la configuration correcte du clavier, vous aurez peut- tre   intervenir pour le personnaliser selon vos propres d sirs. La configuration du clavier comprend la d finition des scancodes, la d finition du plan de clavier, et le r glage de la vitesse de r p tition et de la touche de verrou num rique.

VI-I-4-a - D finition de scancodes

La d finition des scancodes est une op ration qui n cessite une bonne connaissance du fonctionnement du clavier des PC. Heureusement, rares sont les personnes qui disposent de claviers non standards, c'est- -dire, en pratique, de claviers disposant de plus de 105 touches. Notez que Linux reconna t parfaitement les touches « Windows » qui ont  t  introduites par Microsoft, et il n'y a donc pas lieu de d finir leurs scancodes. Cependant, les claviers les plus r cents disposent de touches « Internet » ou « Multimedia », qui ne sont pas encore reconnues par Linux en standard. L'utilisation de ces touches se traduit donc simplement par un message d'erreur dans les traces syst mes. Si vous disposez d'un tel clavier, et que vous d sirez utiliser ces touches, vous allez devoir les d finir dans la table des scancodes du noyau.

Pour r aliser cette op ration, il faut avant tout d terminer les scancodes envoy s par le clavier lors de l'appui sur la touche   d finir. Comme nous l'avons indiqu  plus haut, les scancodes peuvent  tre visualis s avec l'option `-s` de `showkey` :

```
showkey -s
```

Les scancodes sont exprimés en hexadécimal, c'est-à-dire en base 16. Dans cette base, les lettres A à F représentent les chiffres manquants à la base 10, c'est-à-dire les chiffres ayant respectivement pour valeur 10, 11, 12, 13, 14 et 15.

En général, le clavier envoie un scancode lors de l'appui sur une touche, puis le même scancode augmenté de 128 lors de son relâchement. Le nombre de 128 provient du fait que le bit de poids fort du scancode est mis à un lors du relâchement de la touche. Ainsi, la touche 'A' des claviers français renvoie le scancode 16 lors de l'appui, soit 0x10 en hexadécimal, et le scancode 144, soit 0x90, lors du relâchement (notez que la valeur 128 se note 0x80 en hexadécimal, et que l'on a bien 0x90 = 0x10 + 0x80).

Cependant, certaines touches renvoient des scancodes plus complexes. Ces touches sont généralement des touches qui ont été ajoutées après la définition des premiers claviers PC, et qui ont sensiblement le même rôle qu'une touche déjà présente. C'est exactement le cas de la touche **CTRL** de droite par rapport à la touche **CTRL** de gauche. Pour cette raison, les scancodes de ces touches sont les mêmes que ceux de certaines touches « classiques », mais ils sont précédés du préfixe 0xE0, qui indique qu'il s'agit d'une touche étendue. Par exemple, la touche **CTRL** de gauche (c'est-à-dire la première touche de contrôle qui ait été utilisée) utilise le scancode 0x1D à l'appui, et le scancode 0x9D au relâchement. La touche **CTRL** de droite renvoie donc la séquence suivante à l'appui :

```
0xE0 0x1D
```

et la séquence suivante au relâchement :

```
0xE0 0x9D
```

L'intérêt de procéder ainsi est que les vieux programmes, incapables de gérer les codes de touches étendus, ignoraient purement et simplement le code 0xE0. Ainsi, ils confondaient les deux touches **CTRL**, ce qui est le comportement désiré.

D'autres touches étendues utilisent des séquences de scancodes variables. Par exemple, les touches du curseur (entre la touche **Entrée** et le pavé numérique) ont été ajoutées pour simplifier le déplacement du curseur à l'écran. Initialement, on n'utilisait que les touches du pavé numérique, en combinaison de la touche **Majuscule** de droite si le verrou numérique était enclenché. Par conséquent, la séquence de scancodes générée lors de l'utilisation de la touche **Flèche Haute** du pavé numérique était 0x48 si le verrou numérique n'était pas enclenché, et 0x2A 0x48 sinon (soit le scancode de la touche **Majuscule** de gauche suivi du scancode de la touche **8** du pavé numérique). Lors du relâchement, la séquence était inversée (et augmentée de 128) : 0xC8 ou 0xC8 0xAA selon l'état du verrou numérique. La touche **Flèche Haute** étendue renvoie donc deux séquences de scancodes différents selon l'état du verrou numérique 0xE0 0x48 ou 0xE0 0x2A 0xE0 0x48.

Lors du relâchement, la séquence suivante complexe 0xE0 0xC8 ou 0xE0 0xC8 0xE0 0xAA est émise selon l'état du verrou numérique. Notez que l'état du verrou numérique est maintenu en interne par l'électronique du clavier, indépendamment de l'état de la diode lumineuse du verrou numérique. Celui-ci peut en effet être enclenché sans que la diode soit allumée, si le programme de gestion du clavier ne synchronise pas le clavier et sa diode.

Enfin, pour couronner le tout, certaines touches spéciales utilisent une séquence de scancodes spécifiques. Par exemple, la touche **Pause** ne renvoie que la séquence 0xE1 0x1D 0x45 0xE1 0x9D 0xC5 à l'appui. Elle utilise donc le préfixe 0xE1 au lieu de 0xE0, et simule l'appui et le relâchement immédiat des touches **CTRL** + **Verrou Numérique** (ce qui était la manière de provoquer la pause avant l'introduction de la touche **Pause**).

Comme vous pouvez le constater, tout cela est très compliqué. Heureusement, le noyau fait le ménage pour vous et élimine les touches suivantes :

- les touches non étendues sont traitées immédiatement (scancodes 0x01 à 0x58) ;
- la touche **Pause** est gérée automatiquement ;
- les codes 0x2A et 0xAA de la touche **Majuscule** de droite sont automatiquement éliminés.

Grâce à ce traitement, les seuls scancodes que vous manipulerez sont les scancodes simples non connus, et les scancodes précédés du préfixe 0xE0.

L'association d'un scancode inconnu à un keycode se fait à l'aide de l'utilitaire `setkeycodes`. Sa syntaxe est la suivante :

```
setkeycodes scancode keycode
```

où `scancode` est le numéro du scancode, exprimé en hexadécimal et sans le préfixe « 0x », et `keycode` est bien entendu le numéro de keycode à attribuer à cette touche. Les numéros de scancodes peuvent être simples (par exemple 48) ou précédés de leur préfixe (par exemple E048). Les numéros de keycodes doivent être exprimés en décimal, et doivent être compris entre 1 et 127.

Par exemple, mon clavier dispose de 12 touches « Multimédia » supplémentaires, dont les scancodes ne sont pas reconnues par le noyau. La touche Arrêt du son générant les scancodes suivants à l'appui :

```
0xE0 0x20
```

je peux lui associer le keycode 119 avec la commande suivante :

```
setkeycodes E020 119
```

Il va de soi qu'il faut s'assurer que chaque keycode n'est utilisé qu'une fois (à moins, bien entendu, de vouloir que plusieurs touches aient le même comportement). Pour cela, vous aurez sans doute besoin de voir les correspondances entre scancodes et keycodes. La commande suivante vous donnera la liste de ces correspondances :

```
getkeycodes
```

Les keycodes sont présentés à raison de huit par ligne. Le scancode de la touche décrite par le premier élément de chaque ligne est affiché en tête de la ligne. Les autres scancodes peuvent être déduits en ajoutant le numéro de la colonne dans laquelle se trouve chaque keycode au numéro du scancode du premier keycode.

VI-I-4-b - Définition d'un plan de clavier

Comme nous l'avons expliqué dans les paragraphes précédents, les plans de clavier définissent les associations entre les keycodes et les caractères ou les séquences d'échappement renvoyés par le clavier. Les plans de clavier permettent également de définir des séquences de composition, afin d'obtenir de nouveaux caractères en composant les actions de plusieurs touches.

La définition d'un plan de clavier est donc constituée de trois parties. La première partie décrit les symboles accessibles pour certaines touches du clavier. Par exemple, les symboles accessibles pour la touche A peuvent être les caractères 'a', 'A', 'Ã' et 'Ä', à l'aide des touches de majuscule et AltGr. La deuxième partie permet d'affecter des chaînes de caractères à certaines touches particulières. Cette partie peut contenir par exemple la définition de touches raccourci pour les commandes les plus utilisées. En pratique cependant, elles sont souvent utilisées pour définir des séquences d'échappement pour les touches de fonction et les touches du curseur, séquences qui seront interprétées ensuite par les applications. Enfin, la dernière partie permet de définir les compositions de touches pour obtenir des caractères qui n'auraient été accessibles autrement que par des combinaisons de touches compliquées.

Pour chaque touche normale du clavier, il est possible de définir jusqu'à 256 symboles différents. Ces symboles sont sélectionnés grâce à des touches de modification du comportement de base des autres touches, par exemple la touche Ctrl ou la touche Alt Gr. Linux est capable de gérer des combinaisons de touches faisant intervenir jusqu'à huit touches de modification différentes en plus de la touche affectée (remarquez qu'avec dix doigts, on peut encore y parvenir !). Ces touches sont récapitulées dans le tableau suivant :

Touche de modification	Valeur	Description
Shift	1	Touches de passage en majuscule (situées au-dessus des touches Ctrl sur les claviers français).
altgr	2	Touche Alt Gr, située à droite de la barre d'espacement.
control	4	Touches Ctrl, situées aux extrémités inférieures du clavier.
alt	8	Touche Alt, située à gauche de la barre d'espacement.
shiftl	16	Touche de majuscule de gauche.
shiftr	32	Touche de majuscule de droite.
ctrl	64	Touche Ctrl de gauche.
ctrlr	128	Touche Ctrl de droite.

Chaque touche dispose d'un code numérique qui est une puissance de deux. Lorsqu'elles sont utilisées dans une combinaison avec une touche donnée, la somme de ces valeurs est calculée pour déterminer le numéro du symbole dans la liste des symboles associés à la touche. Comme on peut le voir, toutes les valeurs possibles allant de 0 à 255 sont réalisables selon la combinaison de touches utilisée, ce qui permet donc de définir effectivement 256 symboles différents pour chaque touche du clavier.

Cependant, les touches Shift et Ctrl sont utilisées plusieurs fois dans le tableau, précisément trois fois, la première ne faisant pas de distinction entre la touche de droite et la touche de gauche. En pratique donc, toutes les combinaisons ne sont pas réalisables. Mais en réalité, les touches de contrôle du clavier sont des touches comme les autres, et peuvent être placées n'importe où dans le plan de clavier. Il est donc parfaitement possible de réaliser une distinction entre les touches Majuscule, Majuscule Droite et Majuscule Gauche (et de même pour les touches Ctrl). Par exemple, on peut associer la touche Majuscule à la touche Echap, ce qui permet de faire la distinction entre les trois variantes de touches de majuscules... Heureusement, il n'est pas nécessaire d'aller jusque là. Les plans de clavier n'utilisent en pratique que les quatre premières touches de contrôle, qui sont celles que vous connaissez.

La définition des symboles accessibles pour une touche utilise la syntaxe suivante :

```
keycode = symbole symbole symbole &#8230;
```

où `keycode` est le keycode qui identifie la touche en question, et `symbole` est un des symboles acceptés dans les plans de clavier. Vous pourrez obtenir la liste des symboles utilisés grâce à la commande suivante :

```
dumpkeys --long-info
```

Vous pourrez voir par exemple les symboles `A`, `B`, `C`, etc. qui représentent les lettres classiques, ainsi que des symboles spéciaux comme `exclamdown`, `hyphen`, `cedilla`, etc. pour les lettres non alphabétiques. Il existe également des symboles pour les touches spéciales comme `Backspace`, `Delete`, `Escape`. Enfin, le symbole `VoidSymbol` permet de signaler l'absence de symbole pour la combinaison de touches considérée.

En théorie, il faut définir la liste des 256 symboles accessibles pour chaque touche. Le premier symbole est donc le symbole obtenu par appui direct de la touche, le deuxième est celui obtenu par la combinaison Majuscule + Touche, le troisième celui de la combinaison AltGr + Touche, le quatrième par Majuscule + AltGr + Touche, etc. Évidemment, il est très fastidieux de définir ces 256 possibilités. Pour simplifier le format des plans de clavier, il est possible de ne spécifier que les combinaisons utilisées, ce qui réduit en pratique à quatre colonnes de symboles un plan de clavier français. De plus, les derniers symboles sont facultatifs s'ils sont tous `VoidSymbol`. Vous pouvez indiquer au début du plan de clavier les valeurs des combinaisons de touches de modifications qui seront effectivement utilisées avec la syntaxe suivante :

```
keymaps valeurs
```

où `valeurs` est la liste des valeurs ou des plages de valeurs des combinaisons de touches utilisées. Les éléments de cette liste sont séparés par des virgules, et les plages de valeurs sont indiquées par leurs première et dernière valeurs, séparées par un tiret.

Vous pouvez également utiliser une autre syntaxe, qui permet de ne modifier que les symboles associés à certaines combinaisons de touches. Cette syntaxe est très utile lorsque vous ne désirez modifier que quelques affectations de touches :

```
modificateur keycode = symbole
```

Dans cette syntaxe, `modificateur` est la liste des modificateurs devant intervenir dans la combinaison de touches, et `keycode` et `symbole` sont toujours le keycode de la touche et le symbole à générer. Les modificateurs autorisés sont les noms des touches de modification indiquées dans le tableau ci-dessus, plus le modificateur `plain`, qui signifie qu'aucune touche de modification n'est utilisée. Par exemple, la ligne suivante :

```
plain keycode 16 = q
```

permet d'affecter le symbole `q` à la touche `A` de votre clavier, et la ligne :

```
alt keycode 30 = a
```

permet d'affecter le symbole `a` à la combinaison de touches `Alt + Q` (n'essayez surtout pas ces deux exemples, vous deviendriez fou).

Note : Par défaut, les symboles utilisables dans les plans de clavier sont les symboles du jeu de caractères ISO 8859-1. D'autres encodages sont utilisables, mais celui-ci convient parfaitement pour un clavier français.

Comme nous l'avons dit plus haut, la deuxième partie d'un plan de clavier permet d'affecter des chaînes de caractères à certaines touches. Cela est facilement réalisable, avec la syntaxe suivante :

```
string symbole = "chaîne"
```

où `symbole` est le nom d'un des symboles affecté précédemment à une touche, et `chaîne` est une chaîne de caractères. Les chaînes de caractères étant délimitées par des guillemets anglais (« " »), ceux-ci ne peuvent pas être utilisés en tant que caractères de ces chaînes. Pour résoudre ce problème, on peut utiliser un antislash comme caractère d'échappement :

```
\"
```

Vous pouvez également spécifier des caractères directement à l'aide de leur valeur en base huit, à l'aide de la syntaxe suivante :

```
\0valeur
```

où `valeur` est la valeur du caractère. Bien entendu, l'antislash étant utilisé comme caractère d'échappement, il doit lui-même être précédé d'un caractère d'échappement si l'on désire l'utiliser dans une chaîne de caractères :

```
\\
```

Ainsi, si l'on veut faire en sorte que la touche `F4` affiche la chaîne de caractères « Coucou », il suffit d'utiliser la ligne suivante dans le plan de clavier :

```
string F4 = "Coucou"
```

Bien entendu, les chaînes de caractères les plus utiles sont celles qui définissent les séquences d'échappement pour les touches spéciales. Par exemple, la définition de la touche Page Haut est la suivante :

```
string PageUp = "\033[5~"
```

Vous pouvez reconnaître ces séquences d'échappement à la présence du caractère octal 033, soit 27 en décimal, qui n'est rien d'autre que le caractère Escape dans le jeu de caractères ISO 8859-1. Ne vous inquiétez pas pour l'instant de la forme apparemment compliquée de ces séquences d'échappement, nous verrons plus loin comment elles sont utilisées.

Enfin, les plans de clavier contiennent la définition des compositions. Les compositions de touches sont accessibles à l'aide d'une touche spéciale dite de composition, dont le symbole est Compose dans le plan de clavier (il est donc nécessaire que ce symbole soit affecté à l'une des touches du clavier pour utiliser les compositions). Lorsqu'on appuie sur la touche de composition, le clavier attend deux autres touches, dont les caractères serviront de base au caractère composé. Par exemple, les caractères 'A' et 'a' donnent le caractère composé 'à'.

Les compositions utilisent la syntaxe suivante :

```
compose 'caractère' 'caractère' to 'résultat'
```

où caractère est un des deux caractères à composer et résultat est le caractère résultant de la composition. Notez bien que les compositions se font sur les caractères, et pas sur les touches. Elles sont donc actives, quelle que soit la méthode obtenue pour générer ces caractères (combinaison de touches ou touches simples).

Note : Le noyau définit la plupart des compositions nécessaires pour le jeu de caractères ISO 8859-1. Ne les cherchez donc pas en vain dans votre plan de clavier, vous ne les trouverez pas forcément...

D'une manière générale, vous pourrez trouver de plus amples renseignements concernant les plans de clavier dans la page de man keymaps.

Il est long et difficile de créer un plan de clavier de toutes pièces. Heureusement, encore une fois, cette description n'était que didactique. Vous n'aurez certainement même pas à modifier votre plan de clavier (sauf si vous voulez faire des expériences), car des plans de claviers prédéfinis sont fournis avec toutes les distributions. Ces plans de clavier sont en général stockés dans le répertoire `/usr/lib/kbd/keymap`. Pour les claviers de PC français, je vous conseille tout particulièrement le plan de clavier `fr-latin0.map.gz` du sous-répertoire `i386/azerty`. Ce plan de clavier permet d'accéder à toutes les touches utilisées par le français, plus la plupart des touches utilisées en Europe. Il utilise l'encodage ISO 8859-15, afin d'avoir accès aux rares caractères manquants dans l'encodage ISO 8859-1.

Le chargement d'un plan de clavier est très simple. Il suffit de taper la commande suivante sous le compte root :

```
loadkeys keymap
```

où keymap est le nom du fichier contenant le plan de clavier à utiliser. Si l'on ne spécifie pas de fichier, loadkey attend que vous tapiez la spécification des touches directement. Vous pourrez valider en générant le caractère EOF (abréviation de « End Of File », ce qui signifie « Fin de fichier ») à l'aide de la combinaison de touche `CTRL + D`, ou abandonner avec le classique `CTRL + C`.

En général, les distributions utilisent la commande loadkey dans les fichiers d'initialisation du système, pour charger le plan de clavier que vous avez choisi à l'installation dès le démarrage. Il est recommandé de ne modifier le plan de clavier courant que par l'intermédiaire du programme de configuration du système fourni avec votre distribution.

VI-I-4-c - Modification des paramètres du clavier

Il nous faut encore voir deux petits utilitaires permettant de fixer quelques paramètres du clavier pour finir notre tour d'horizon de ce périphérique. Le premier de ces outils permet de manipuler l'état des diodes lumineuses qui indiquent

si le verrou numérique, le verrou des majuscules ou l'arrêt défilement sont actifs. Cet utilitaire se nomme logiquement `setleds`, et il permet non seulement de fixer l'état de ces diodes, mais également l'état du clavier. Rappelons en effet que les diodes peuvent être désynchronisées par rapport à l'état réel du clavier. Ce cas de figure peut se produire si l'on passe d'un terminal X à un terminal en mode console, tout simplement parce que les serveurs X ne savent pas comment déterminer l'état du clavier sur une console afin de le restaurer.

La syntaxe de `setleds` est la suivante :

```
setleds -D num | caps | scroll
```

où `num`, `caps` et `scroll` sont des options permettant de préciser respectivement l'état des verrous numériques, majuscules et défilement. Ces options sont de la forme `+num`, `-num` pour le verrou numérique, et de forme similaire pour les autres verrous. Comme leur syntaxe l'indique, elles permettent d'enclencher ou de désenclencher l'état des verrous correspondants du clavier.

L'état des verrous du clavier est conservé par chaque terminal virtuel, indépendamment les uns des autres. Cette commande devra donc être répétée pour tous les terminaux virtuels utilisés. L'option `-D` permet de rendre les changements permanents pour le terminal sélectionné. Ainsi, si ce terminal est réinitialisé, la modification sera conservée. C'est en général l'effet désiré. Sachez qu'il est également possible d'effectuer une modification temporaire, et de ne modifier que l'affichage des diodes (sans changer l'état des verrous).

Il est recommandé de placer les quelques lignes suivantes dans les fichiers d'initialisation de votre système afin d'activer le verrou numérique pour tous les terminaux, si votre distribution ne vous permet pas de le faire par l'intermédiaire de son programme de configuration :

```
INITTTY=`ls /dev/tty[1-9] /dev/tty1[0-2]`  
for tty in $INITTTY; do  
    setleds -D +num < $tty  
done
```

Ces lignes appellent la commande `setleds` sur les terminaux virtuels 1 à 12.

Le deuxième utilitaire est `kbdrate`. Sa fonction est de fixer les paramètres de délai d'attente avant répétition lorsqu'on maintient une touche enfoncée, ainsi que la vitesse de répétition utilisée une fois que ce délai d'attente est dépassé. Sa syntaxe est elle aussi très simple :

```
kbdrate -s -r taux -d délai
```

où `taux` est le taux de répétition, et `délai` est le délai d'attente avant le début des répétitions de la touche enfoncée. L'option `-s` indique à `kbdrate` de ne pas afficher de messages pendant l'exécution de la commande. Les délais d'attente spécifiés peuvent être de 250, 500, 750 ou 1000 millisecondes. Les taux de répétition utilisables vont de 2 à 30 caractères par seconde. Toutefois, les claviers ne peuvent pas accepter toutes les valeurs intermédiaires pour le taux de répétition. Vous trouverez la liste exhaustive des valeurs admissibles dans la page de manuel `kbdrate`.

VI-I-5 - Choix de la police de caractères

Par défaut, la police de caractères utilisée par Linux est la police enregistrée dans la carte graphique. Cette police convient pour l'affichage des textes anglais, car elle utilise la page de codes 437. Il est donc souhaitable, sinon recommandé, de changer de police de caractères à la fois pour bénéficier d'un encodage adapté à la langue française, et pour bénéficier d'améliorations esthétiques éventuelles.

Linux est fourni avec un certain nombre de fichiers de polices, qui sont normalement installées dans le répertoire `/usr/lib/kbd/consolefonts`. Comme vous pourrez le constater si vous y jetez un coup d'œil, ce répertoire contient un grand nombre de polices, utilisant divers encodages. Les polices les plus utiles pour un ordinateur situé en France sont sans doute les polices `iso01*`, `lat1-*` et `lat0-*`. Cette dernière police utilise l'encodage ISO 8859-15, et contient donc tous les symboles utilisés en Europe. Il est recommandé, mais non nécessaire, d'utiliser une des polices encodées

en ISO 8859-1 ou en ISO 8859-15, avec un plan de clavier approprié. Par exemple, le plan de clavier `fr-latin0.map.gz` pourra être utilisé avec la police de caractères `lat0-16.psfu.gz`.

Les polices de caractères sont chargées aisément avec l'utilitaire `setfont`. La syntaxe de ce programme est très simple :

```
setfont police
```

où `police` est le fichier de police à charger. Si vous ne spécifiez aucun fichier de police, la police par défaut sera chargée.

Notez que les fichiers de polices peuvent utiliser un encodage particulier spécifique, qui ne correspond à aucune des tables de conversion prédéfinies du noyau. Pour ces polices, il est nécessaire de spécifier des tables de conversion qui leur sont propres. Ces tables permettent de convertir les codes des caractères reçus par la console en codes Unicode spéciaux. Tous ces codes appartiennent à une plage de codes Unicode réservée au système, et que Linux utilise pour définir les codes qui permettront d'accéder directement aux caractères de la police. L'association entre les codes de caractères de l'application et leurs descriptions dans la police de caractères est donc réalisée grâce à ces tables de conversion.

Vous pouvez charger une table de conversion spécifique à l'aide de l'utilitaire `mapscrn`. Cet utilitaire prend en ligne de commande un fichier contenant la table de conversion, et charge cette dernière dans la quatrième table de conversion du noyau (c'est-à-dire la table réservée à l'utilisateur) :

```
mapscrn fichier
```

(où `fichier` est le fichier contenant la description de la table de conversion).

Notez que l'utilitaire `mapscrn` charge la table de conversion, mais ne s'assure pas que le jeu de caractères courant utilise cette table. Par conséquent, vous aurez sans doute à utiliser les codes d'échappement `\033(K` et `\033)K` pour faire pointer respectivement les jeux de caractères G0 et G1 sur la quatrième table. Ces codes d'échappement peuvent être saisis à l'aide de la commande `cat` :

```
cat
```

Il vous faudra alors taper sur la touche `Echap`, puis saisir « `(K` » ou « `)K` », et enfin signaler la fin de fichier avec un `CTRL + D`. Faites bien attention à ce que vous faites : en cas d'erreur, le jeu de caractères utilisé peut rendre l'écran totalement illisible. Si cela devait se produire, vous devrez taper à l'aveugle la commande `reset`. Cette table s'assure que le jeu de caractères actif est le jeu de caractères G0, et que celui-ci utilise la première table de conversion. Vous pourrez également recharger la police par défaut à l'aide de `setfont`.

Heureusement, la plupart des polices de caractères contiennent également la table de conversion à utiliser, et `setfont` effectue tout le travail en une seule passe. Il charge la police de caractères dans la carte graphique, puis sa table de conversion dans le noyau, et s'assure enfin que le jeu de caractères G0 soit actif et utilise cette table. Ainsi, l'utilisation de `mapscrn` est devenue facultative.

La dernière étape dans la configuration de la police de caractères est le chargement de la table de conversion Unicode vers les indices des caractères dans la police. Cette table est essentielle, puisque c'est elle qui indique l'emplacement des caractères de la police à utiliser pour chaque code Unicode. Ces tables dépendent donc de l'encodage utilisé par la police de caractères.

Le chargement des tables de conversion est réalisé par le programme `loadunimap`. La syntaxe de ce dernier est la même que celle de `mapscrn` :

```
loadunimap fichier
```

où `fichier` est un fichier de conversion approprié. Vous trouverez de tels fichiers dans le répertoire `/usr/lib/kbd/consoletrans`. Les fichiers proposés permettent d'utiliser les polices de caractères encodées selon les encodages les plus courants.

Bien entendu, certaines polices disposent de leur propre table d'encodage. Encore une fois, l'utilisation de `loadunimap` est rendue facultative par `setfont`, qui se charge d'effectuer tout le travail. C'est notamment le cas pour les polices `fr-latin0.psfu.gz` et `fr-latin1.psfu.gz`, dont le « u » en fin d'extension indique la présence de la table de conversion Unicode.

VI-I-6 - Configuration des paramètres du terminal

La configuration du gestionnaire de la console ne concerne à proprement parler que les plans de clavier et les polices de caractères. Pour les applications, tout se passe comme si elles accédaient à un terminal comme les autres, donc par l'intermédiaire d'une ligne de communication. En général, les lignes de communication utilisées pour les terminaux sont des lignes série, c'est-à-dire des lignes sur lesquelles les données sont envoyées sous la forme de petits paquets de 5 à 8 bits, et contrôlées par d'autres bits éventuellement facultatifs (bit de stop et bit de parité). Il va de soi que tous les terminaux ne sont pas identiques, et qu'un certain nombre de paramètres doivent être définis tant au niveau des terminaux qu'au niveau des lignes de communication utilisées.

Ces paramètres peuvent être fixés avec la commande `stty`. Leur nombre interdit ici une description exhaustive, d'autant plus que pour un terminal local, ils sont tous initialisés à des valeurs par défaut correctes, et vous n'aurez pas à les modifier. Vous pouvez toutefois consulter la page de manuel de `stty` pour de plus amples renseignements à ce sujet.

Si vous êtes curieux, vous pourrez obtenir à tout moment la liste des paramètres de la ligne de communication d'un terminal avec l'option `-a` de `stty` :

```
stty -a
```

Comme vous pourrez le constater, le nombre des options utilisées est assez impressionnant. Sont définis, entre autres, les paramètres de vitesse de communication de la ligne (option `speed`), le format des données transférées sur la ligne (option `parodd` pour la parité impaire, option `cs8` pour des paquets 8 bits, etc.), la géométrie du terminal (options `rows` et `columns`), les caractères de contrôle affectés à un certain nombre d'actions (comme, par exemple, le caractère « `^Z` », accessible par la combinaison de touches `CTRL + Z`, qui permet de suspendre l'exécution du programme utilisant la ligne, et le caractère « `^C` », qui permet de le tuer) et des options de gestion des caractères transférés (par exemple, `echo` fait en sorte que tout caractère entrant est immédiatement réémis sur la console, ce qui permet de voir ce que l'on tape).

Parmi ces paramètres, les plus intéressants sont sans doute ceux définissant les actions associées aux différentes touches de contrôle. Vous pourrez modifier ces paramètres avec la syntaxe suivante :

```
stty action caractère
```

où `action` est l'une des actions gérées par `stty`, par exemple `susp` pour suspendre le processus en cours, `intr` pour le tuer, etc. Remarquez que l'action `kill` n'a rien à voir avec la gestion des signaux des processus. Elle permet simplement d'effacer la ligne courante du terminal.

VI-I-7 - Description des terminaux

Le dernier maillon de la chaîne de gestion des caractères est bien entendu les applications. La plupart des applications sont capables de traiter les caractères simples comme les lettres de l'alphabet, les chiffres et la ponctuation. Cependant, il n'en va pas de même pour les codes d'échappement des terminaux. Comme nous l'avons vu, certaines touches sont programmées pour renvoyer des codes d'échappement dans le plan de clavier. Mais le gestionnaire de la console est également capable d'interpréter certains codes d'échappements, afin d'effectuer des actions spécifiques. Par exemple, les touches du curseur émettent des codes d'échappement spéciaux, et il est du ressort de chaque

programme de reconnaître ces codes et d'agir en conséquence. De même, les programmes peuvent envoyer des codes d'échappement à la console pour déplacer le curseur affiché, effacer l'écran, effectuer un défilement, etc.

Le malheur, c'est que les codes d'échappement utilisables diffèrent selon les terminaux utilisés. Les terminaux les plus standards sont les terminaux VT100 et VT102 (la console Linux en fait partie). Cependant, les programmes ne peuvent pas savoir, a priori, quels codes d'échappement doivent être utilisés pour effectuer telle ou telle action.

C'est pour ces diverses raisons que les systèmes Unix disposent d'une base de données de définition des terminaux. Cette base de données contient tous les codes d'échappement que chaque type de terminal est capable de gérer, décrits d'une manière uniforme. Ainsi, les programmes désirant gérer correctement les terminaux n'ont qu'à consulter cette base de données pour interpréter et récupérer les codes d'échappement utilisés par le terminal. Historiquement, la définition des terminaux était réalisée dans le fichier de configuration `/etc/termcap`. Ce fichier est obsolète et a été remplacé par la base de données `terminfo`, que tous les programmes modernes doivent à présent utiliser. Cependant, le fichier de configuration `/etc/termcap` a été conservé par compatibilité avec les vieux programmes qui l'utilisent encore.

Le fichier `termcap` comprend une ligne pour chaque type de terminal décrit. Cette ligne est constituée d'un certain nombre de champs, séparés par le caractère `:`. Le premier champ de chaque ligne contient la description du terminal que cette ligne décrit. Les champs suivants sont des définitions des variables contenant les séquences d'échappement du terminal.

Les informations descriptives du terminal sont les suivantes :

- le nom abrégé du terminal ;
- le nom complet du terminal ;
- les autres noms sous lesquels le terminal est également connu ;
- la description détaillée du terminal (toujours en dernier).

Ces informations sont séparées les unes des autres par une barre verticale (`|`).

Viennent ensuite les définitions des variables. Chaque variable est définie selon la syntaxe suivante :

```
variable=séquence
```

où `variable` est le nom de la variable, et `séquence` est la séquence d'échappement associée à cette variable. Notez que certaines variables ne prennent pas de paramètres, et que leur présence dans la description du terminal signale simplement un comportement particulier de celui-ci. Notez également que pour les variables numériques, le caractère d'égalité est remplacé par un caractère dièse (`#`).

Un certain nombre de variables peuvent être définies pour chaque terminal. Ce sont ces variables qui sont utilisées par les programmes pour retrouver les séquences d'échappement du terminal. Par conséquent, les noms de ces variables sont fixés une fois pour toutes, et elles représentent toujours la même fonctionnalité. La liste des fonctionnalités est, encore une fois, très grande, et les variables utilisables sont listées exhaustivement dans la page de manuel `termcap`.

Note : Il est évident que les lignes du fichier `/etc/termcap` peuvent devenir très longues. Il est donc possible de les étaler sur plusieurs lignes physiques, en insérant le caractère de continuation de ligne antislash (`\`). Après chaque retour à la ligne, il faut utiliser une indentation à l'aide du caractère de tabulation.

Vous trouverez ci-dessous un exemple de définition de terminal :

```
lx|linux|Console Linux:\
:do=^J:co#80:li#25:cl=\E[H\E[J:sf=\E[D:sb=\E[M:\
:le=^H:bs:am:cm=\E[%i%d;%dH:nd=\E[C:up=\E[A:\
:ce=\E[K:cd=\E[J:so=\E[7m:se=\E[27m:us=\E[36m:ue=\E[m:\
:md=\E[1m:mr=\E[7m:mb=\E[5m:me=\E[m:is=\E[1;25r\E[25;1H:\
:ll=\E[1;25r\E[25;1H:al=\E[L:dc=\E[P:dl=\E[M:\
:it#8:ku=\E[A:kd=\E[B:kr=\E[C:kl=\E[D:kb=^H:ti=\E[r\E[H:\
```

```
:ho=\E[H:kP=\E[5~:kN=\E[6~:kH=\E[4~:kh=\E[1~:kD=\E[3~:kI=\E[2~:\n:
:k1=\E[[A:k2=\E[[B:k3=\E[[C:k4=\E[[D:k5=\E[[E:k6=\E[[17~:\n:
:k7=\E[[18~:k8=\E[[19~:k9=\E[[20~:k0=\E[[21~:K1=\E[[1~:K2=\E[[5~:\n:
:K4=\E[[4~:K5=\E[[6~:\n:
:pt:sr=\EM:vt#3:xn:km:bl=^G:vi=\E[?25l:ve=\E[?25h:vs=\E[?25h:\n:
:sc=\E7:rc=\E8:cs=\E[%i%d;%dr:\n:
:r1=\Ec:r2=\Ec:r3=\Ec:
```

Cette ligne permet de définir le terminal associé à la console Linux. Vous pourrez par exemple reconnaître le nombre de colonnes et de lignes (variables `co` et `li`), ainsi que les codes d'échappement associés aux touches du curseur (variable `ku`, `kd`, `kr` et `kl` respectivement pour les touches haut, bas, droite et gauche). Par exemple, la variable `cl` donne la séquence d'échappement utilisable pour effacer l'écran et faire revenir le curseur en haut à gauche (séquence d'échappement « Esc [H Esc [J »). Comme il l'a déjà été dit plus haut, la liste complète des variables peut être obtenue en consultant la page de manuel `termcap`, et ce fichier ne sera pas décrit plus en détail ici.

La base de données `terminfo` a été introduite pour combler certaines limitations du fichier `termcap`. Si le principe de fonctionnement est presque le même, les informations fournies tiennent compte des terminaux plus récents et de nouvelles fonctionnalités. Cela signifie en pratique que de nouvelles variables ont été définies pour décrire les nouveaux terminaux. Inversement, certaines variables de `termcap` ont disparu parce qu'elles devenaient obsolètes, et ont été remplacées par des variables équivalentes de `terminfo`.

La principale différence entre `terminfo` et `termcap` est que la description des terminaux n'est plus stockée dans un fichier de configuration en mode texte. Toutes les données sont désormais stockées dans des fichiers binaires, qui peuvent être générés à l'aide du programme `tic`. Ces fichiers binaires sont usuellement placés dans les sous-répertoires du répertoire `/usr/share/terminfo/`. En fait, le nombre de fichiers de description est tellement grand qu'ils ont été regroupés par ordre alphabétique. Ainsi, le répertoire `/usr/share/terminfo` contient des sous-répertoires dont les noms sont les premières lettres des fichiers de description, et chaque fichier de description est situé dans le répertoire correspondant. Par exemple, le fichier de description des terminaux Linux se nomme tout simplement `linux`. Comme la première lettre de son nom est 'l', il est stocké dans le répertoire `/usr/share/terminfo/l/`, avec les descriptions de tous les autres terminaux dont le nom commence par 'l'.

Note : En fait, les fichiers de description de terminaux peuvent être placés à un autre emplacement que l'emplacement par défaut. Les bibliothèques de programmes utilisant les informations de `terminfo` cherchent en effet en premier dans le chemin référencé par la variable d'environnement `TERMINFO`, puis dans le répertoire `.terminfo` du répertoire de l'utilisateur. Ce n'est que si ces deux recherches échouent qu'elles utilisent les informations du répertoire par défaut.

Comme nous l'avons dit, les fichiers de description sont des fichiers binaires, qui ont été compilés à l'aide du compilateur `tic`. Cependant, vous pouvez parfaitement visualiser le contenu de ces fichiers ou comparer deux fichiers à l'aide de l'utilitaire `infocmp`. Par exemple, vous pouvez visualiser sous une forme lisible les informations du fichier de description des terminaux Linux avec la commande suivante :

```
infocmp linux
```

Vous obtiendrez certainement un résultat semblable à ceci :

```
#Reconstructed via infocmp from file: /usr/lib/terminfo/l/linux
linux|linux console,
    am, bce, eo, mir, msgr, xenl, xon,
    colors#8, it#8, pairs#64,
    acsc=+020\,021-030.^Y0\333`004a\261f\370g\361h\260i\316j\\
331k\277l\32m\300n\305o~p\304q\304r\304s_t\303u\264v\301w\302x\263y\\
363z\362{\343|\330}\234~\376,
    bel=^G, blink=\E[5m, bold=\E[1m, civis=\E[?25l,
    clear=\E[H\E[J, cnorm=\E[?25h, cr=^M,
    csr=\E[%i%p1%d;%p2%dr, cub1=^H, cud1=^J, cuf1=\E[C,
    cup=\E[%i%p1%d;%p2%dH, cuu1=\E[A, cvvis=\E[?25h,
    dch=\E[%p1%dP, dch1=\E[P, dim=\E[2m, dl=\E[%p1%dM,
    dl1=\E[M, ech=\E[%p1%dX, ed=\E[J, el=\E[K, ell=\E[1K,
    flash=\E[?5h\E[?5l$<200/>, home=\E[H, hpa=\E[%i%p1%dG,
    ht=^I, hts=\EH, ich=\E[%p1%d@, ich1=\E[@, il=\E[%p1%dL,
    ill=\E[L, ind=^J, invis=\E[8m, kb2=\E[G, kbs=\177, kcbt=\E[Z,
```

```

kcub1=\E[D, kcud1=\E[B, kcufl1=\E[C, kcuu1=\E[A,
kdch1=\E[3~, kend=\E[4~, kf1=\E[[A, kf10=\E[21~,
kf11=\E[23~, kf12=\E[24~, kf13=\E[25~, kf14=\E[26~,
kf15=\E[28~, kf16=\E[29~, kf17=\E[31~, kf18=\E[32~,
kf19=\E[33~, kf2=\E[[B, kf20=\E[34~, kf3=\E[[C, kf4=\E[[D,
kf5=\E[[E, kf6=\E[17~, kf7=\E[18~, kf8=\E[19~, kf9=\E[20~,
khome=\E[1~, kich1=\E[2~, knp=\E[6~, kpp=\E[5~, kspd=^Z,
nel=^M^J, op=\E[39;49m, rc=\E[8, rev=\E[7m, ri=\EM,
rmacs=\E[10m, rmir=\E[4l, rmpch=\E[10m, rmso=\E[27m,
rmul=\E[24m, rs1=\Ec, sc=\E[7, setab=\E[4%p1%dm,
setaf=\E[3%p1%dm,
sgr=\E[0;10%p1%t;7%;%?%p2%t;4%;%?%p3%t;7%;%?%p4%t;5%;\
%?%p5%t;2%;%?%p6%t;1%;%?%p7%t;8%;%?%p9%t;11%;m,
sgr0=\E[m, smacs=\E[11m, smir=\E[4h, smpch=\E[11m,
smso=\E[7m, smul=\E[4m, tbc=\E[3g, u6=\E[%i%d;%dR,
u7=\E[6n, u8=\E[?6c, u9=\E[c, vpa=\E[%i%p1?,

```

Comme vous pouvez le constater, le format de ces informations est similaire à celui de celles qui sont enregistrées dans le fichier `/etc/termcap`. Les principales différences sont que les différents champs sont séparés par des virgules (',') au lieu du caractère deux points (':'), et qu'il est possible de les répartir sur plusieurs lignes physiques (il est donc inutile d'utiliser le caractère antislash en fin de ligne physique pour indiquer la continuation de la ligne logique). De plus, le nom des variables utilisées dans les fichiers terminfo n'est pas le même a priori. Cependant, le principe d'utilisation de ces variables reste le même, chacune d'entre elles permet de définir une des fonctionnalités gérées par le terminal et de définir la séquence d'échappement nécessaire à l'obtention de cette fonctionnalité si nécessaire. Vous pourrez trouver la liste exhaustive des variables utilisables dans la page de manuel `terminfo`. Le format des fichiers de configuration de terminfo ne sera donc pas décrit plus en détail dans ce document.

Qu'ils utilisent des bibliothèques basées sur `termcap` ou `terminfo`, les programmes doivent tous connaître le nom du terminal courant pour récupérer les informations qui permettent de l'utiliser dans ces bases de données. C'est précisément ce à quoi sert la variable d'environnement `TERM`. Cette variable contient en permanence le nom du terminal courant, que les programmes peuvent utiliser comme index dans les bases de données `termcap` ou `terminfo`. Ainsi, si vous désirez travailler sur un terminal Linux, vous pourrez fixer le type de terminal correct avec la commande suivante :

```
export TERM=linux
```

La liste des noms de terminaux utilisables peut être obtenue en lisant directement le fichier `termcap`. Cette liste peut être obtenue plus facilement pour `terminfo`, à l'aide de l'utilitaire `toe` (abréviation de l'anglais « Table Of terminfo Entry »).

Bien entendu, vous n'aurez à définir la valeur de `TERM` que si cette variable d'environnement n'est pas correctement définie, ce qui est très rare. En fait, cela ne peut se produire que lors d'une connexion à distance sur un autre ordinateur, dont les terminaux ne sont pas de type Linux.

VI-I-8 - Paramétrage des applications

En théorie, la description du terminal courant fournie par les bases de données `termcap` ou `terminfo` est suffisante pour faire fonctionner correctement la plupart des applications. En particulier, les programmes qui utilisent les bibliothèques de fonctions de manipulation des terminaux sont capables d'utiliser ces informations. C'est le cas par exemple de tous les programmes qui utilisent les bibliothèques « `curses` » ou « `ncurses` », car celles-ci s'appuient sur la base de données `terminfo`.

Cependant, certaines applications utilisent un mécanisme plus simple (mais moins souple) pour gérer les terminaux. Pour ces applications, une configuration spécifique doit être effectuée, souvent pour chaque terminal, en précisant les codes d'échappement à utiliser dans leurs fichiers de configuration. On remarquera la présence du shell « `bash` », de l'éditeur « `vi` » et du programme de pagination « `less` » dans cette catégorie de logiciels. Comme ce sont les logiciels les plus couramment utilisés, il est indispensable d'indiquer comment les configurer pour une utilisation correcte.

VI-I-8-a - Configuration du clavier pour la bibliothèque readline

Un certain nombre d'applications, dont le shell « bash », utilisent la librairie GNU « readline » pour obtenir les lignes de commandes saisies par l'utilisateur. Tous ces programmes peuvent donc être configurés de la même manière, grâce au fichier de configuration de la bibliothèque readline.

Cette bibliothèque recherche son fichier de configuration en premier à l'emplacement indiqué par la variable d'environnement INPUTRC. Si cette variable d'environnement n'est pas définie, le fichier de configuration `~/.inputrc` est lu pour déterminer les séquences d'échappement à utiliser. Il est donc recommandé de créer un fichier de configuration général `/etc/inputrc` et de définir la variable d'environnement INPUTRC afin de définir des paramètres communs à tous les utilisateurs. Vous pouvez également recopier ce fichier dans les répertoires personnels de tous les utilisateurs sous le nom `.inputrc`.

Ainsi, si vous voulez gérer correctement le clavier français sous le shell bash, vous devrez vous assurer que les lignes suivantes sont placées dans votre fichier `inputrc` :

```
# Active la gestion du huitième bit des caractères
# (par exemple pour les caractères accentués) :
set meta-flag on
set input-meta on
set output-meta on
set convert-meta off

# Définit les codes d'échappement associés aux touches du curseur :
"\e[1~": beginning-of-line
"\e[4~": end-of-line
"\e[2~": overwrite-mode
"\e[3~": delete-char
"\e[5~": history-search-backward
"\e[6~": history-search-forward
"\e[C": forward-char
"\e[D": backward-char
"\e[A": previous-history
"\e[B": next-history

# Redéfinit les codes d'échappement pour l'émulateur de terminal xterm :
$if term=xterm
"\e[H": beginning-of-line
"\e[F": end-of-line
$endif
```

Ce fichier commence par autoriser le traitement des caractères dont le bit « meta », c'est-à-dire le huitième bit, est positionné. C'est en particulier le cas de tous les caractères accentués dans les principales pages de codes ; ces options sont donc nécessaires pour pouvoir utiliser ces caractères dans le shell. La suite du fichier définit les actions associées à chaque code d'échappement du terminal. Le caractère d'échappement est représenté ici par la chaîne de caractères « `\e` ». Comme les codes d'échappement sont différents pour la console et pour les émulateurs de terminaux de XWindow, il est nécessaire de les redéfinir en fonction de la nature du terminal. Ce fichier présente un exemple d'expression conditionnelle sur le nom du terminal, tel qu'indiqué dans la variable d'environnement TERM, afin de ne redéfinir ces codes que pour ces terminaux.

VI-I-8-b - Configuration du clavier pour vi

L'éditeur en ligne de commande vi n'est pas réputé pour être agréable à utiliser. Cela provient essentiellement de sa distinction entre les modes de commande, d'édition et de visualisation. Mais cela n'est pas suffisant pour le rendre haïssable : la torture psychologique imposée par la mauvaise gestion du curseur vient souvent à bout des plus résistants. Heureusement, Linux fournit un clone nettement plus puissant et qui permet de résoudre ces problèmes : vim.

L'éditeur vim peut bien entendu être parfaitement compatible avec vi afin ne pas dérouter les habitués de vi. Mais il dispose en plus de fonctionnalités avancées qui en font un outil extrêmement configurable, et il est possible de

le rendre nettement plus ergonomique que son ancêtre. Malheureusement, il faut reconnaître que la plupart des distributions fournissent un vim « brut de fonderie », ce qui fait que seuls ceux qui se lancent dans la lecture de son aide peuvent parvenir à l'utiliser correctement. Les options qui sont proposées ici sont donc données simplement à titre indicatif, mais permettront peut-être de rendre vim un peu plus ergonomique.

Les options de configuration de vim sont stockées dans deux fichiers. Le premier fichier est le fichier de configuration commun à tous les utilisateurs, `vimrc`. Ce fichier peut être placé soit dans le répertoire `/etc/`, soit dans le répertoire `/usr/share/vim/`, selon votre distribution. Le deuxième fichier est le fichier de préférences personnelles de chaque utilisateur, et se nomme `.vimrc`. Il est normalement placé dans le répertoire personnel de l'utilisateur.

Plusieurs types d'options peuvent être indiquées dans ces fichiers de configuration. Les premières associent les actions à effectuer aux codes d'échappement générés par les touches du curseur. Les autres spécifient simplement le comportement de l'éditeur et le paramétrage de ses principales fonctionnalités. Vous trouverez ci-dessous les principales options que vous pourrez ajouter à votre fichier de configuration `vimrc`. Ces options rendront sans doute l'utilisation de vim beaucoup plus agréable.

```
" Exemple d'options de configuration pour vim.
" Notez que les commentaires sont introduits ici
" par des guillemets anglais (") et non
" par des dièses ('#'), contrairement aux fichiers de configuration
" de la plupart des autres applications.

" Éviter à tout pris la compatibilité avec vi, qui est insupportable :
set nocompatible

" Définir les touches du clavier :

" Gestion du curseur en mode de visualisation :
map ^[OA      k
map ^[[A      k
map ^[OB      j
map ^[[B      j
map ^[OD      h
map ^[[D      h
map ^?  h
map ^H  h
map ^[OC      l
map ^[[C      l
map ^[[2~     i
map ^[[3~     x
map ^[[1~     0
map ^[OH      0
map ^[[H      0
map ^[[4~     $
map ^[OF      $
map ^[[F      $
map ^[[5~     ^B
map ^[[6~     ^F
map ^[[E      ""
map ^[[G      ""
map ^[OE      ""

" Gestion du pavé numérique en mode de visualisation :
map ^[Oo      :
map ^[Oj      *
map ^[Om      -
map ^[Ok      +
map ^[Ol      +
map ^[OM      ^M
map ^[Ow      7
map ^[Ox      8
map ^[Oy      9
map ^[Ot      4
map ^[Ou      5
map ^[Ov      6
map ^[Oq      1
map ^[Or      2
```

```
map ^[Os      3
map ^[Op      0
map ^[On      .

" Gestion du pavé numérique en mode insertion :
map! ^[Oo :
map! ^[Oj *
map! ^[Om -
map! ^[Ok +
map! ^[Ol +
map! ^[OM ^M
map! ^[Ow 7
map! ^[Ox 8
map! ^[Oy 9
map! ^[Ot 4
map! ^[Ou 5
map! ^[Ov 6
map! ^[Oq 1
map! ^[Or 2
map! ^[Os 3
map! ^[Op 0
map! ^[On .

" Gestion du curseur dans les modes d'insertion et de commande :
map! ^[[H <Home>
map! ^[OH <Home>
map! ^[[F <End>
map! ^[OF <End>
map! ^[OA <Up>
map! ^[OB <Down>
map! ^[OC <Right>
map! ^[OD <Left>
map! ^[[3~ <Delete>
map! ^[OE <Space>

" Définir les autres options globales :

" Paramétrage des touches Backspace et Delete :
set t_kb=^?
set t_kD=ESC[3~

" Faire en sorte que le "backspace" efface même les sauts de lignes :
set bs=2

" Utiliser l'indentation automatique dans les fichiers C et C++
" (pour les programmeurs) :
set cindent

" Utiliser la coloration syntaxique pour les principaux langages
" de programmation :
set background=dark
if &t_Co > 1
    syntax on
endif

" Signaler les correspondances de parenthèses, accolades et crochets :
set showmatch

" Rappeler le mode de fonctionnement courant :
set showmode
```

Vous constaterez que certains caractères de contrôle sont utilisés dans ce fichier de configuration, dont le caractère de contrôle « `^` », qui représente le caractère d'échappement. Ces caractères sont représentés avec la notation classique « `^C` », où « `C` » est la lettre à utiliser avec la touche `CTRL` pour obtenir ce caractère. Ces notations ne font que représenter les caractères de contrôle, et doivent être remplacées par les caractères qu'elles représentent dans le fichier de configuration. Pour saisir ces caractères spéciaux, vous devrez passer en mode insertion dans vi, puis utiliser la combinaison de touches `CTRL+V`. Ce raccourci permet d'indiquer à vi qu'il doit insérer les codes d'échappement directement issus du clavier, sans les interpréter. Vous pourrez alors taper la séquence de touches générant le caractère de contrôle ou la séquence d'échappement désirée. Par exemple, vous pourrez obtenir le

caractère « `^H` » en tapant la combinaison de touches `CTRL+H`, et le caractère « `^?` » en appuyant sur la touche `Backspace` (retour arrière). Les caractères d'échappement peuvent être générés par la touche `Echap` ou directement par les touches du curseur.

Note : En fait, vous pouvez également utiliser les chaînes de caractères « `\e` » et « `<Esc>` » pour représenter le caractère d'échappement. Mais certaines options ne fonctionnent pas avec ces notations, et je vous les déconseille.

Vous pouvez bien entendu ajouter d'autres options dans ce fichier de configuration. En pratique, toutes les options utilisables dans le mode de commande de `vim` peuvent être fixées définitivement dans ce fichier. Vous obtiendrez de l'aide sur ces options grâce à la commande « `:help` » de `vim`.

VI-I-8-c - Configuration du clavier pour less

Le programme de pagination `less` est naturellement bien plus agréable à utiliser que son ancêtre `more`, puisqu'il permet de revenir sur les pages déjà consultées. Cependant, il peut le devenir encore plus si l'on s'arrange pour qu'il reconnaisse les touches du curseur.

Le programme `less` lit ses informations de configuration dans un fichier binaire dont l'emplacement est spécifié par la variable d'environnement `LESSKEY`. Si cette variable d'environnement n'est pas définie, `less` utilise le fichier de configuration `.less` du répertoire personnel de l'utilisateur. Ce fichier binaire contient les associations entre les séquences d'échappement du terminal et les actions effectuées par `less`. Il est généré par le compilateur `lesskey`, à partir d'un fichier de configuration textuelle classique.

Ce fichier de configuration comprend plusieurs sections, qui permettent de définir les touches utilisées en mode de commande, les touches utilisées en mode d'édition de lignes (par exemple dans une commande de recherche), et enfin les variables d'environnement utilisées par `less`. Vous trouverez ci-dessous un exemple de fichier de configuration pour `less`. Vous trouverez de plus amples renseignements sur les actions qui peuvent être associées aux séquences d'échappement et aux caractères de contrôle dans la page de manuel `lesskey`.

```
# Exemple de fichier de configuration pour less

# Première section :

#command
\e[B    forw-line
\e[A    back-line
\e[6~   forw-scroll
\e[5~   back-scroll
\177    back-screen
^H      back-screen
\e[3~   back-screen
\e[2~   visual
\e[1~   goto-line
\eOH    goto-line
\e[4~   goto-end
\eOF    goto-end
\eOM    forw-line

# Deuxième section :

#line-edit
\177    backspace
^H      backspace
\e[3~   delete
\e[1~   home
\e[H~   home
\eOH    home
\e[4~   end
\e[F~   end
\eOF    end
\e[5~   up
\e[6~   down
```

```
\e[2~    insert
\e[E     insert
\e[G     insert
\eOE     insert
\eOo     insert :
\eOj     insert *
\eOm     insert -
\eOk     insert +
\eOl     insert +
\eOM     insert
\eOw     insert 7
\eOx     insert 8
\eOy     insert 9
\eOt     insert 4
\eOu     insert 5
\eOv     insert 6
\eOq     insert 1
\eOr     insert 2
\eOs     insert 3
\eOp     insert 0
\eOn     insert .

# Troisième section :

#env
LESSCHARSET=latin1
```

Conformément à un usage courant, les commentaires sont introduits par le caractère dièse ('#') dans ce fichier de configuration. Cependant, certains commentaires sont utilisés pour identifier le début des trois sections du fichier. Il s'agit des commentaires « `#command` », « `#line-edit` » et « `#env` ». Il ne faut donc surtout pas supprimer ces commentaires dans votre fichier de configuration.

Encore une fois, le caractère d'échappement est symbolisé par la chaîne de caractère « `\e` ». De même, les caractères de contrôle sont représentés par la notation classique « `^C` », où `C` est la touche utilisée en combinaison avec CTRL pour générer ce caractère de contrôle. Notez que, contrairement au fichier de configuration `/etc/vimrc`, ces notations peuvent être utilisées directement dans le fichier de configuration de `less`.

Ce fichier de configuration pourra être compilé avec le programme `lesskey` afin de générer le fichier binaire utilisé par `less`. Pour cela, il faudra simplement utiliser la syntaxe suivante :

```
lesskey fichier
```

où `fichier` est le nom du fichier de configuration à compiler. Le fichier binaire généré est par défaut celui référencé par la variable d'environnement `LESSKEY`. Si cette variable n'est pas définie, un fichier `.less` sera créé dans le répertoire personnel de l'utilisateur.

VI-I-9 - Configuration de la souris

L'installation de la souris est une opération très simple à réaliser. La seule chose importante est de bien connaître les différents types de souris et de ne pas les confondre. Autrefois, la plupart des souris étaient des souris connectées sur le port série (souris sérieelles). Aujourd'hui, ces souris se font de plus en plus rares, et le port de prédilection est le port PS/2. Ce port a été introduit par IBM dans ses ordinateurs PS/2 et est quelque peu plus pratique que le port série, car il définit une interface standard pour toutes les souris. De plus, il permet de dégager un des ports série, ce qui simplifie la configuration des modems. Le port PS/2 ressemble au port clavier du même type, et en fait on peut se tromper et brancher la souris à la place du clavier et inversement. Il ne faut surtout pas confondre les souris PS/2 avec les souris Bus, qui ont été vendues autrefois et que l'on ne trouve quasiment plus à présent. Ces souris pouvaient se connecter sur des cartes spéciales voire, parfois, sur la carte graphique.

Pour que l'installation de la souris se fasse correctement, il faut s'assurer que les options concernant la souris ont bien été définies dans la configuration du noyau de Linux. Cela n'est pas nécessaire pour les souris série. Vous pouvez

consulter la [Section 7.3](#) pour plus de détails sur la configuration du noyau. Lorsque cette étape est faite, il ne reste plus qu'à indiquer au programme de gestion de la souris à quel type de souris il a affaire. Ce programme, nommé `gpm`, permet d'utiliser la souris en mode texte. La configuration de la souris pour XWindow sera vue dans le [Chapitre 10](#).

La configuration de `gpm` se fait normalement par l'intermédiaire du programme de configuration de votre distribution. Lorsqu'on n'utilise pas XWindow, `gpm` est lancé automatiquement au démarrage. Il se peut que votre programme de configuration vous demande le type de souris à utiliser. Dans ce cas, il faut choisir le bon type, faute de quoi `gpm` ne fonctionnera pas correctement. Attention, si vous désirez utiliser une souris à molette (souris disposant d'une petite roue entre les deux boutons, et permettant de faire défiler le contenu des fenêtres), le type de souris à utiliser est « `imps2` » et non simplement « `ps2` ». Pour avoir la liste des types de souris gérés par `gpm`, il suffit de le lui demander avec la ligne de commande suivante :

```
gpm -t help
```

Normalement, vous aurez à utiliser `gpm` avec la ligne de commande suivante :

```
gpm -t type -m /dev/mouse
```

où `type` est le type de la souris que `gpm` doit utiliser et `/dev/mouse` est un lien vers le fichier spécial de périphérique gérant votre souris.

VI-J - Configuration de l'imprimante

Il existe deux systèmes d'impression concurrents sous Linux : LPRng (« Line Printer Next Generation ») et CUPS (« Common Unix Printing System »). LPRng est une évolution du système initial, LPR, qui est devenu très vite obsolète en raison de l'évolution des technologies d'impression. En effet, celui-ci a été conçu à l'époque où les imprimantes étaient encore des imprimantes matricielles et ne pouvaient imprimer qu'en noir et blanc. CUPS, quant à lui, a été créé pour fournir une infrastructure complètement nouvelle et pour s'affranchir des limitations de LPRng. C'est donc la solution d'avenir, mais il n'est pas rare de trouver encore des systèmes basés sur LPRng. D'autre part, la compatibilité au niveau des commandes d'impression est assurée par CUPS, ce qui fait que la présentation de LPRng n'est pas superflue.

Quelle que soit la technologie d'impression utilisée, l'imprimante reste généralement accessible par un port local. En effet, la plupart des imprimantes sont connectées sur le port parallèle ou sur un port USB (les imprimantes professionnelles mises à part, celles-ci disposant généralement d'une interface réseau). Nous supposons donc dans la suite de ce document que l'imprimante est connectée soit sur le port parallèle (accessible via le fichier spécial de périphérique `/dev/lp0`), soit sur un port USB (accessible via le fichier `/dev/usb/lp0`).

VI-J-1 - Concepts de base de l'impression sous Unix

La principale difficulté lors de l'impression d'un document est de le convertir de son propre format de fichier vers un format de données que l'imprimante est capable de comprendre. Le problème ici est qu'il existe une multitude de formats de documents et une multitude d'applications qui peuvent les générer, d'une part, et, d'autre part, que chaque fabricant d'imprimantes a développé ses propres jeux de commandes pour ses imprimantes. Il existe bien un standard pour les imprimantes laser avec le langage PostScript développé par Adobe, mais ce langage n'est pas utilisable avec la majorité des imprimantes à jet d'encre. De fait, la plupart des applications sont capables de générer un fichier PostScript lors d'une impression, mais ce fichier doit encore être traduit dans le langage spécifique de l'imprimante effectivement installée.

Il est donc toujours possible d'envoyer directement un fichier à imprimer à l'imprimante, en le recopiant directement sur le fichier spécial de périphérique de l'imprimante, mais cette technique suppose que ce fichier soit généré directement dans le langage de l'imprimante. Ceci ne peut que compliquer les applications qui permettent l'impression, et les empêcher de fonctionner ultérieurement avec de nouvelles imprimantes dont le langage leur sera inconnu. Cette technique ne convient donc, en général, que pour des fichiers textes simples. De plus, cette solution ne convient pas

dans un système multitâche et multiutilisateur, puisqu'un seul programme seulement peut accéder à l'imprimante à un instant donné.

Les systèmes d'impression ont donc pour principal but de résoudre ces problèmes. La multiplicité des formats de fichiers et des langages d'imprimante est prise en charge par tout un jeu de programmes de conversion que l'on appelle les filtres. Les filtres ne sont en fait rien d'autre que des programmes qui reçoivent un fichier en entrée et qui fournissent la traduction de ce fichier dans un autre format en sortie. Les systèmes d'impression LPRNG et CUPS fournissent un certain nombre de filtres, qui peuvent être agencés afin de permettre l'impression de n'importe quel fichier sur l'importe quelle imprimante.

Enfin, en ce qui concerne le partage des ressources d'impression, les systèmes d'impression utilisent généralement un mécanisme de files d'attente (« spool » en anglais). Tous les travaux d'impression soumis sont placés dans une file d'attente, et attendent leur tour pour être envoyés à l'imprimante associée à cette file. Une fois que l'impression est terminée, les travaux sont supprimés de la file. Ainsi, un seul programme accède à l'imprimante : le sous-système d'impression. Notez qu'il est généralement possible de définir plusieurs files d'attente sur une même imprimante, selon la nature du travail à effectuer. Par exemple, une file peut se charger des documents textes, et une autre des documents graphiques en couleur. Bien entendu, le sous-système d'impression contrôle tous les accès à l'imprimante et assure qu'un seul document est en cours d'impression à chaque instant.

VI-J-2 - Le système d'impression LPRng

Le système d'impression LPRng est constitué d'un démon nommé « lpd » et d'un jeu de commandes permettant de communiquer avec lui pour soumettre les travaux d'impression, visualiser ceux qui sont en attente ou en cours d'impression et les annuler.

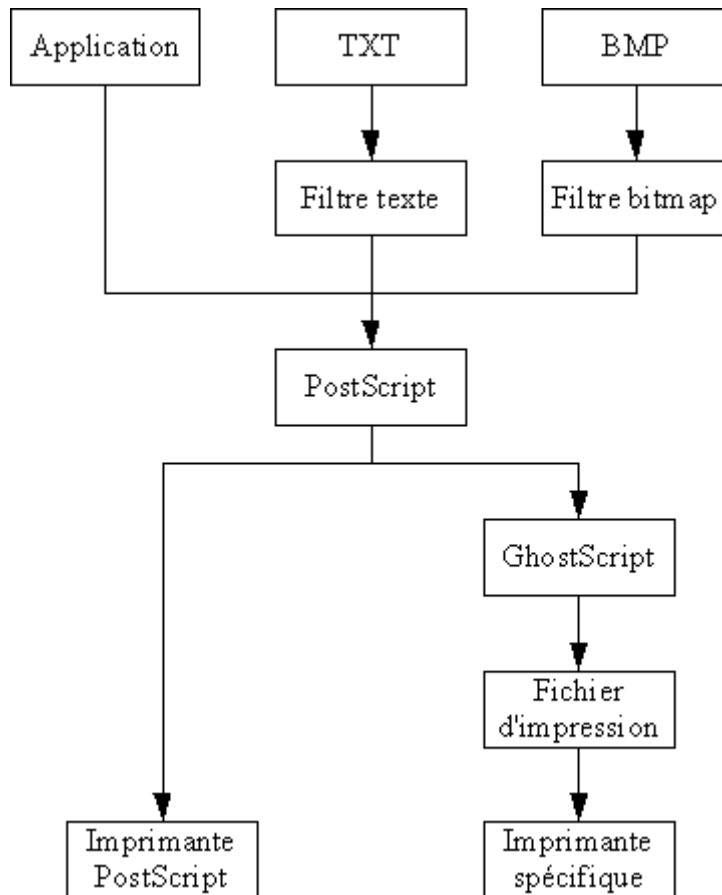
VI-J-2-a - Le mécanisme des filtres APSFILTER

Sous Linux, le mécanisme de filtres de LPRng est généralement APSFILTER. Ce mécanisme est profondément orienté vers le langage de description de pages PostScript, initialement inventé par Adobe et que nombre d'imprimantes laser comprennent. En fait, il s'agit d'un véritable langage de programmation, qui permet de programmer les périphériques dont la vocation est d'afficher ou d'imprimer des documents. Techniquement parlant, PostScript permet d'obtenir une qualité d'impression irréprochable, car c'est l'imprimante elle-même qui « dessine » la page à imprimer.

APSFILTER utilise donc un premier jeu de filtres pour convertir les fichiers à imprimer en PostScript s'ils ne sont pas déjà dans ce format. Le langage PostScript apparaît donc comme le langage d'impression universel, que toutes les imprimantes sont supposées comprendre...

Le problème, c'est que ce n'est pas le cas. En particulier, les imprimantes à jet d'encre ne comprennent généralement pas le PostScript. Aussi APSFILTER utilise-t-il un second jeu de filtres, capables de convertir le PostScript dans le langage graphique de l'imprimante (si l'imprimante n'est pas une imprimante PostScript, bien entendu). APSFILTER utilise pour cela un « interpréteur PostScript ». Un interpréteur PostScript est un programme capable de comprendre les fichiers PostScript et de les convertir dans le format compris par l'imprimante. L'interpréteur couramment utilisé sous Linux est GhostScript, parce que c'est un logiciel libre (cependant, la version courante est toujours commerciale). Il est également capable d'afficher les fichiers PostScript sous XWindow, et de gérer la plupart des imprimantes du marché.

L'avantage de cette technique est que toutes les imprimantes apparaissent comme étant des imprimantes PostScript pour les programmes désirant imprimer. Ainsi, leur programmation est beaucoup plus simple, puisqu'ils n'ont qu'un seul format à gérer. Bon nombre d'applications génèrent directement des fichiers PostScript, qui sont donc envoyés directement à GhostScript pour l'impression définitive.



Filtres d'impression et imprimantes PostScript

Comme on le voit, avec APSFILTER, le langage d'impression universel est le langage PostScript. Bien entendu, cela est idéal si l'on dispose effectivement d'une imprimante PostScript, mais même dans le cas contraire, les impressions se font parfaitement grâce à GhostScript.

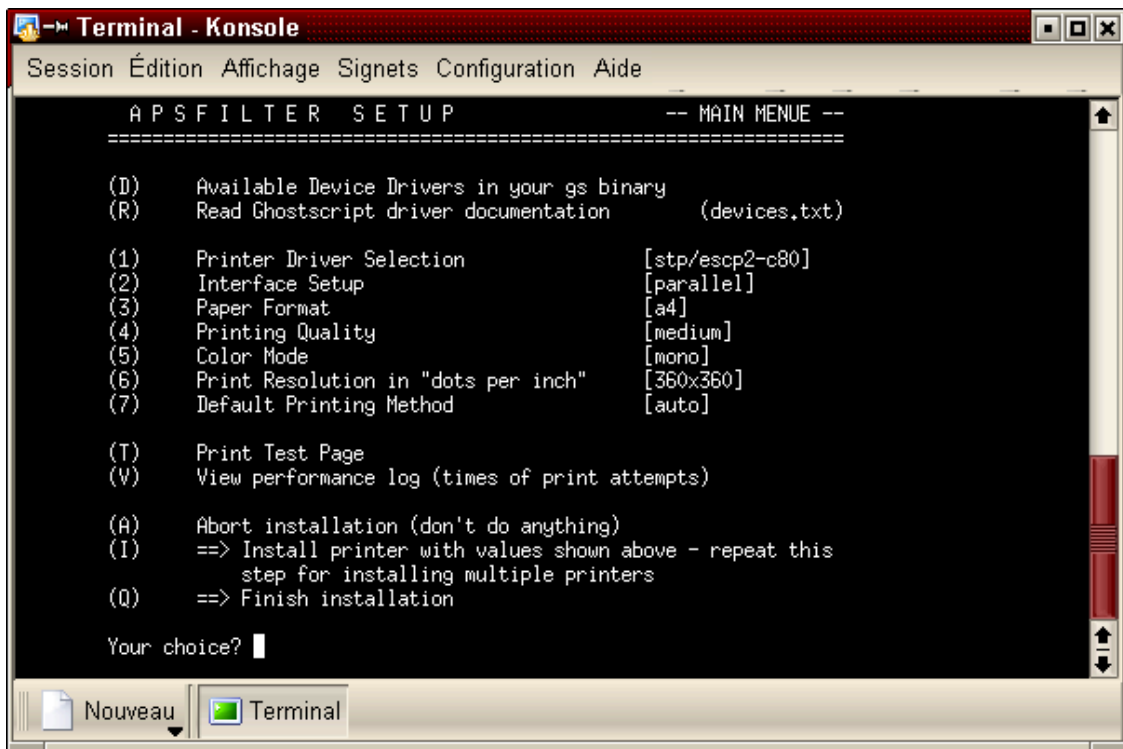
VI-J-2-b - Installation des filtres et configuration des files d'impression

Les distributions modernes fournissent toutes un outil permettant d'effectuer la configuration du sous-système d'impression. Il est évidemment recommandé d'utiliser ces outils, car le résultat est assuré et la vie en est d'autant plus facile. De plus, les distributions peuvent fournir des filtres complémentaires que seuls ces outils connaissent et sont capables d'installer.

Si le jeu de filtres utilisé de votre distribution est APSFILTER, vous pourrez installer votre imprimante manuellement en exécutant le programme SETUP du répertoire `/usr/share/apsfilter/`.

Ce programme est réellement affreux à utiliser, car il s'agit en réalité d'un script, mais il effectue correctement son travail. Dès son démarrage, il vous demande d'accepter la licence logicielle (la licence GPL avec une forte invitation à envoyer une carte postale à l'auteur, qui les collectionne). La question suivante est donc si vous désirez obtenir l'adresse postale de l'auteur. Une fois ces deux questions passées, le programme d'installation commence réellement.

Le programme vous demande de confirmer les droits sur le répertoire des fichiers d'impression en cours, puis, si vous le faites effectivement, si vous désirez compléter le fichier de configuration de LPRNG `/etc/printcap` ou en créer un complètement nouveau. Vient ensuite le menu de configuration des imprimantes :



Filtres d'impression et imprimantes PostScript

La première option vous permet de choisir le pilote de l'imprimante, qui donc en général est un des pilotes fournis avec l'interpréteur GhostScript. Vous devez sélectionner le pilote de votre imprimante ou, à défaut, celui de l'imprimante qui s'en rapproche le plus. La deuxième option vous permet de sélectionner l'interface de l'imprimante. Vous pourrez indiquer le port de l'imprimante ou, s'il s'agit d'une imprimante réseau, la manière d'y accéder. Les autres options du menu quant à elles sont relativement simples, et vous permettront respectivement de sélectionner le format du papier, la qualité et le mode d'impression, ainsi que la résolution de l'imprimante.

Vous pourrez (et devriez) tester la configuration ainsi définie avec l'option 'T'. Si le résultat vous convient, installez l'imprimante avec l'option 'I'. Vous pouvez installer plusieurs imprimantes ou plusieurs files d'impression pour la même imprimante en répétant les étapes précédentes. Finalement, terminez l'installation avec l'option 'Q', et envoyez une carte postale à l'auteur du logiciel.

VI-J-2-c - Commandes d'impression

La commande d'impression de LPRng est la classique commande `lpr` (abréviation de l'anglais « off Line PRint », ce qui signifie « impression différée »). Cette commande est très simple à utiliser, comme le montre la syntaxe suivante :

```
lpr fichier
```

où `fichier` est le nom du fichier à imprimer. Cette commande se contente de placer le fichier à imprimer dans un répertoire affecté à la file d'attente des travaux d'impression. Le travail d'impression est ensuite effectué par le démon `lpd`, qui fait passer chaque fichier à imprimer à travers la série de filtres pour le convertir dans le langage de l'imprimante, puis qui alimente l'imprimante.

La liste des travaux d'impression en attente peut être consultée avec la commande `lpq`. Chaque travail en attente porte un numéro, grâce auquel on peut le manipuler. Entre autres opérations, il est possible de l'abandonner à l'aide de la commande `lprm`.

Enfin, pour consulter et contrôler l'état des files d'impression, on peut utiliser la commande `lpc`. Cette commande peut prendre des options en ligne de commande afin de préciser l'opération à effectuer. Par exemple, l'option `status` permet d'obtenir l'état de chacune des files d'impression. Les autres options permettent d'arrêter le travail en cours, de le

suspendre, de désactiver l'imprimante pour les travaux suivants, et inversement de relancer les travaux d'impression sur cette file.

VI-J-2-d - Description du fichier `/etc/printcap`

Le démon `lpd` utilise le fichier de configuration `/etc/printcap` pour déterminer l'ensemble des files d'impression existantes et quels filtres doivent être utilisés. Ce fichier est généré automatiquement par les utilitaires des distributions ou, à défaut, par le script d'installation d'APSFILTER. Toutefois, comme il est bon de savoir ce qui se passe lorsqu'une commande d'impression est envoyée au démon `lpd`, une brève description de ce fichier sera faite dans cette section.

Chaque file est décrite par une ligne du fichier `/etc/printcap` et une seule. Ces lignes sont constituées de divers champs, séparés par des deux-points (':'). Comme ces lignes peuvent être relativement longues, elles peuvent être réparties sur plusieurs lignes physiques en plaçant le caractère d'échappement '\ ' à la fin de chaque ligne, sauf la dernière.

Le premier champ de la description d'une file d'attente est une liste des noms sous lesquels cette file sera connue. Les différents noms sont écrits les uns à la suite des autres, séparés par une barre verticale (caractère '|').

Les champs suivants décrivent l'imprimante à utiliser, ainsi que les options générales de la file d'attente. Ces champs utilisent tous la même syntaxe :

```
option = valeur
```

Il existe un grand nombre d'options, nombre d'entre elles sont facultatives. Cependant, il est impératif que le démon `lpd` puisse trouver l'imprimante à utiliser. Par conséquent, il faut lui fournir au moins l'une des deux options suivantes :

- l'option `lp` permet de spécifier le fichier spécial de périphérique auquel l'imprimante est connectée ;
- les options `rm` et `rp` permettent de spécifier respectivement le nom d'un serveur d'impression distant (« remote » en anglais) et de l'imprimante à utiliser sur ce serveur (« remote printer »).

Le démon `lpd` doit également connaître le répertoire dans lequel les travaux en attente seront stockés (répertoire dit de « spool »). Ce répertoire peut être défini avec l'option `sd`.

D'autres options peuvent être utiles, comme `sh` (cette option ne prend pas de valeur), qui permet de supprimer la page de garde au début de chaque impression, et `mx`, qui permet de spécifier la taille maximale des travaux d'impression soumis. Cette dernière option permet de fixer des quotas d'impression selon la taille des documents, afin de donner la possibilité aux autres documents d'être imprimés. Cette option utilise une syntaxe particulière :

```
mx#taille
```

où `taille` est la taille maximale autorisée, exprimée en kilo-octets. Le fait de spécifier une taille nulle permet de supprimer ce contrôle.

L'exemple ci-dessous correspond à la définition d'une file d'attente locale élémentaire :

```
ascii|lp:lp=/dev/lp:sd=/var/spool/lpd/ascii:mx#0:sh
```

Comme vous pouvez le constater, il n'y a aucune spécification des filtres d'impression à utiliser dans cet exemple. Les travaux sont donc directement envoyés à l'impression, sans traduction préalable. Il est donc nécessaire qu'ils soient déjà au format de l'imprimante. Si l'on veut utiliser des filtres d'impression, il faut utiliser l'une des options `if`, `cf`, `df`, `gf`, `nf`, `rf`, `tf` ou `vf`. Chacune de ces options permet de spécifier la ligne de commande d'un filtre d'impression spécifique. Le choix du filtre utilisé pour un travail d'impression est effectué lors de l'appel à la commande `lpr`, à l'aide d'une option en ligne de commande. Le filtre `if` est le filtre par défaut, il n'y a donc besoin d'aucune option pour l'utiliser. Les autres filtres peuvent être sélectionnés respectivement avec les options `-c`, `-d`, `-g`, `-n`, `-f`, `-t` et `-v`.

Comme on le voit, le sous-système d'impression ne reconnaît pas automatiquement le format de fichier utilisé. D'autre part, le nombre de filtres utilisables est limité à 8, ce qui peut ne pas suffire étant donné la prolifération des formats de fichiers. Pour résoudre ce problème, APSFILTER utilise un filtre générique (utilisé en tant que filtre par défaut) qui, lui, est capable de reconnaître le format du fichier à imprimer et de le diriger vers un autre filtre ou une série de filtres. Comme on l'a vu ci-dessus, l'ultime filtre utilisé est en général l'interpréteur GhostScript. Ainsi, il n'y a plus de limite sur le nombre de filtres utilisables, et les filtres sont sélectionnés automatiquement en fonction de la nature du document à imprimer.

VI-J-3 - Le système d'impression CUPS

Le système d'impression CUPS a pour but de remplacer LPRng et d'offrir de nouvelles fonctionnalités. En particulier, il est plus facile de configurer et d'ajuster les paramètres avancés des imprimantes. CUPS permet une meilleure intégration dans un réseau hétérogène, car il implémente le protocole d'impression réseau IPP (« Internet Printing Protocol »), qui est devenu un standard de fait pour tous les systèmes d'exploitation. Couplé au logiciel Samba, il permet également d'accéder aux imprimantes des postes Windows.

CUPS est constitué du démon `cupsd`, qui gère à la fois les requêtes d'impression effectuées via le protocole d'impression IPP, qui est un protocole d'impression encapsulé dans des requêtes HTML (le langage de description de pages Web utilisé sur Internet). Le démon `cupsd` étend même ce protocole d'impression pour prendre en charge des pages Web de configuration, ce qui permet de réaliser l'administration du système d'impression très simplement par l'intermédiaire d'une interface Web. Il est ainsi possible de réaliser la configuration et la gestion des imprimantes, ainsi que des groupes d'imprimantes sur lesquelles les travaux d'impression peuvent être répartis. Enfin, CUPS fournit les fonctionnalités de base du système d'impression LPRng, en définissant automatiquement un fichier `/etc/printcap` correspondant à sa propre configuration, et en fournissant un jeu de commandes `lp` classiques redirigeant tous les travaux d'impression vers lui.

CUPS utilise également un nouveau système de filtres, qui permettent d'éviter de passer systématiquement par le format d'impression PostScript. Plusieurs filtres peuvent être utilisés pour passer par différents formats intermédiaires. Le choix de ces filtres est déterminé en fonction du coût de chaque conversion dans la chaîne de transformations du document vers le format final utilisé par l'imprimante. On peut donc dire que CUPS est à la fois plus facile à configurer et plus performant que LPRng, tout en restant compatible avec lui pour les applications.

VI-J-3-a - Le mécanisme des filtres de CUPS

CUPS utilise une architecture légèrement différente de celle de LPRng. Son système de filtres est plus complexe et plus puissant, et ne nécessite pas un passage obligé par le langage d'impression PostScript.

En réalité, CUPS fournit tout un jeu de filtres élémentaires capables de transformer un document d'un format dans un autre. Tous ces filtres sont agencés pour effectuer la traduction complète du document à imprimer dans le langage d'impression utilisé par l'imprimante. Contrairement à LPRng, les formats intermédiaires ne cette fois sont ni uniques, ni forcément du PostScript.

Il est parfois possible de trouver plusieurs jeux de filtres distincts qui réalisent la conversion du document dans le langage d'impression destination, en passant par différents formats intermédiaires. CUPS doit donc effectuer un choix entre les différents jeux de filtres. Pour cela, il attribue un coût à chaque conversion, et calcule le coût total de chaque jeu de filtres utilisable pour imprimer le document. Il choisit ensuite le jeu de filtres qui a le coût total minimal, et garantit ainsi les performances du système d'impression.

Bien entendu, certains de ces filtres utilisent le bon vieux GhostScript, et la plupart des gestionnaires d'imprimantes sont en réalité des drivers créés pour GhostScript ou pour le logiciel de retouche d'images The Gimp. CUPS requiert donc une version modifiée de l'interpréteur GhostScript, mais les distributions l'installent automatiquement et cela ne pose pas de problème. Finalement, l'impression des documents est une opération très facile, et l'utilisateur ne voit rien de toutes les transformations effectuées sur les documents par le mécanisme des filtres.

Enfin, contrairement à LPRng, CUPS donne la possibilité de fournir des paramètres complémentaires sur les filtres d'impression, ce qui confère une grande souplesse au système d'impression. C'est pour cette raison qu'il est recommandé d'utiliser CUPS si l'on désire faire des impressions couleur ou contrôler finement les paramètres d'impression.

VI-J-3-b - Configuration d'une imprimante CUPS

La manière la plus simple de configurer CUPS est d'utiliser son interface Web. Celle-ci est accessible à l'aide de n'importe quel navigateur Internet, en utilisant l'adresse <http://localhost:631/admin>. Lorsque l'on désire se connecter au serveur Web de CUPS via cette interface, celui-ci exige une authentification de l'administrateur. Vous devrez donc vous identifier en tant que root pour effectuer la configuration du système d'impression.

La page Web affichée présente trois zones permettant de réaliser les principales opérations de configuration du système d'impression.

L'ajout d'une imprimante se fait en cliquant sur le bouton « Add Printer ». Ce bouton ouvre une autre page, dans laquelle il est possible de donner le nom de l'imprimante (champ « Name »), son emplacement sur le réseau (champ « Location ») et sa description (champ « Description/ »). La description permet de détailler la nature de l'imprimante et peut contenir n'importe quel texte. Le bouton « Continue » permet de passer à l'écran suivant, qui fournit la liste des périphériques disponibles. Cette liste contient tous les fichiers spéciaux de périphériques de votre système, ainsi que les protocoles d'impression en réseau supportés par CUPS. La page suivante permet de sélectionner la marque de l'imprimante, et la suivante son modèle. Le bouton « Continue » permet alors de terminer l'installation de l'imprimante.

La configuration des imprimantes installées se fait en cliquant sur le bouton « Manage Printers ». L'écran qui s'affiche contient les boutons « Print Test Page » (impression d'une page de test), « Stop Printer » (suspension des impressions), « Reject Jobs » (interdiction de soumettre de nouveaux travaux d'impression), « Modify Printer » (modification de la définition de l'imprimante), « Configure Printer » (configuration avancée de l'imprimante) et « Delete Printer » (suppression de l'imprimante). La fonctionnalité de configuration avancée des imprimantes permet de fixer les valeurs des paramètres du pilote d'impression, comme la résolution, la taille et le type de papier, ainsi que les paramètres de correction des couleurs.

L'ajout d'un groupe d'imprimantes se fait en cliquant sur le bouton « Add Class ». Comme pour la définition des imprimantes, le formulaire qui apparaît vous permet de décrire le groupe d'imprimantes. L'écran suivant donne ensuite la possibilité d'indiquer les imprimantes qui font partie du groupe. Une fois un groupe d'imprimantes défini, il est possible d'effectuer les tâches d'administration le concernant à l'aide du bouton « Manage Classes ».

VI-J-3-c - Les fichiers de configuration de CUPS

Les fichiers de configuration de CUPS sont généralement placés dans le répertoire `/etc/cups/`. Le fichier le plus important est sans doute le fichier `cupsd.conf`, qui contient tous les paramètres de configuration du démon `cupsd`. Ces paramètres sont nombreux et bien décrits dans la documentation de cups, que l'on peut obtenir simplement en ouvrant un navigateur sur l'adresse <http://localhost:631/documentation.html>. De plus, ce fichier est très bien commenté, aussi seules les options les plus importantes seront-elles décrites ici.

Les principales options concernent la sécurité du système d'impression. Normalement, le démon d'impression `cupsd` ne doit pas fonctionner sous le compte root, pour éviter de lui donner tous les droits. Pour cela, il change d'identité lorsqu'il démarre et se place dans le compte utilisateur spécifié par l'option `User`. De même, il se place dans le groupe utilisateur spécifié par l'option `Group`. Généralement, les valeurs utilisées pour ces deux options sont l'utilisateur `lp` et le groupe `sys`, ce sont donc leurs valeurs implicites.

Un autre option particulièrement importante est l'option `Listen`. Cette option permet d'indiquer au démon `cupsd` sur quelles interfaces réseau il doit se mettre en écoute pour les clients désirant s'y connecter. Il n'est généralement pas conseillé de laisser ouvert un port non utilisé lorsqu'on se connecte à Internet, aussi est-il recommandé de configurer le démon `cupsd` pour qu'il n'écoute pas les requêtes provenant d'autres machines que la machine locale. Pour cela, on utilisera la ligne suivante dans le fichier de configuration `cupsd.conf` :

```
Listen 127.0.0.1:631
```

Cette commande permet de ne répondre qu'aux requêtes provenant de la machine locale, sur le port dédié au protocole d'impression IPP, à savoir le port 631. Bien entendu, vous pourrez ajouter d'autres machines si vous disposez d'un réseau local, simplement en ajoutant d'autres options Listen. Notez que l'usage de cette option est incompatible avec l'option Port, qu'il faut donc commenter au préalable, faute de quoi cupsd ne répondra à aucune requête.

Note : Comme on le verra dans le chapitre sur la configuration réseau, l'adresse réseau 127.0.0.1 représente soi-même dans les communications réseau. Cela signifie ici que seuls les clients de la machine locale pourront utiliser le système d'impression.

Il est également possible de fixer des droits sur les différents répertoires de configuration de l'interface Web, ainsi que sur les répertoires virtuels utilisés par le protocole IPP lorsque des requêtes sont faites par les clients. Ainsi, il est possible de ne permettre l'impression que pour des clients qui vérifient certains critères. Les règles définissant les droits d'accès aux répertoires sont indiquées dans des sections `Location`, dont la syntaxe est similaire à celle des balises du langage XML. Par exemple, la section qui décrit les droits d'accès au répertoire de configuration `/admin` est typiquement de la forme suivante :

```
<Location /admin>
  AuthType Basic
  AuthClass System
  Order Deny,Allow
  Deny From All
  Allow From 127.0.0.1
</Location>
```

Vous constaterez que plusieurs informations sont données dans ce type de section. Premièrement, le type d'authentification utilisé est spécifié à l'aide de l'option `AuthType`. Les différentes valeurs possibles pour cette option sont décrites dans le tableau suivant :

Valeur	Signification
None	Aucune identification n'est faite.
Basic	L'identification est réalisée via les mécanismes d'authentification HTML classique. cupsd attend ici un nom d'utilisateur et un mot de passe Unix classiques, qui doivent donc exister dans le fichier <code>/etc/passwd</code> . Notez qu'avec ce mode d'authentification, le mot de passe est envoyé en clair du navigateur Internet utilisé pour réaliser la connexion vers le serveur Web du démon cupsd. Cela signifie que si la communication ne se fait pas en local, une tierce personne pourrait voir le nom d'utilisateur et le mot de passe en écoutant les communications réseau. Ce mode d'authentification est donc extrêmement dangereux.
Digest	L'identification se fait de la même manière que pour le mode Basic, mais le nom de l'utilisateur et le mot de passe ne sont pas transmis en clair. Au lieu de cela, seules les signatures de ces informations, obtenues à l'aide d'une fonction de chiffrement à sens unique, sont transmises sur le réseau. Il

	n'est donc plus possible de déterminer le nom de l'utilisateur et son mot de passe en clair. De plus, les mots de passe utilisés ne sont pas forcément les mêmes que ceux du système, et ils sont stockés de manière indépendante dans le fichier <code>passwd.md5</code> du répertoire <code>/etc/cups/</code> . Ces mots de passe peuvent être ajoutés avec la commande <code>lppasswd</code> . Ce mode d'authentification est donc plus sûr, notez toutefois qu'il reste possible pour un attaquant de se connecter avec le nom de l'utilisateur et son mot de passe chiffré une fois ceux-ci capturés.

Pour chaque méthode d'authentification, il est possible de préciser le critère utilisé pour vérifier les droits d'accès du client. Ce critère est fixé par l'option `AuthClass`. Celle-ci peut prendre les valeurs suivantes :

Valeur	Signification
Anonymous	Aucune critère n'est utilisé, tout le monde peut accéder à la page spécifiée dans la directive <code>Location</code> . Il va de soi que l'utilisation de ce critère avec les autres modes d'authentification que <code>None</code> est absurde.
User	L'utilisateur authentifié doit être un utilisateur du système Unix sous-jacent.
System	L'utilisateur authentifié doit être membre du groupe système. Ce groupe est, par défaut, le groupe <code>sys</code> , mais il peut être modifié à l'aide de l'option <code>SystemGroup</code> du fichier de configuration <code>cupsd.conf</code> .
Group	L'utilisateur authentifié doit être membre du groupe spécifié par l'option <code>AuthGroupName</code> du fichier de configuration <code>cupsd.conf</code> . Ce groupe doit être un des groupes du système Unix sous-jacent, sauf si le mode d'authentification <code>Digest</code> est utilisé. En effet, dans ce cas, le groupe de l'utilisateur doit être celui stocké avec son mot de passe dans le fichier de mots de passe de CUPS (c'est-à-dire le fichier <code>/etc/cups/passwd.md5</code>).

Les mots de passe cups utilisés pour l'authentification dans le mode d'authentification `Digest` peuvent être ajoutés à l'aide de la commande `lppasswd`. Cette commande permet également de fixer le groupe de l'utilisateur lorsque le critère d'accès utilisé est le critère `Group`. La syntaxe générale de cette commande est la suivante :

```
lppasswd [-g groupe] -a utilisateur
```

où `groupe` est le nom de groupe de l'utilisateur (uniquement utilisé pour le mode d'authentification `Digest`) et `utilisateur` est son nom d'utilisateur. À l'issue de cette commande, `lppasswd` demande deux fois le mot de passe de l'utilisateur et met à jour le fichier `passwd.md5`.

La deuxième série d'informations fournie dans les sections **Location** sont les droits sur les machines capables d'accéder à la page Web. L'option **Order** indique l'ordre d'évaluation des directives suivantes. Il est recommandé d'interdire en premier, et de regarder ensuite si la machine cliente a le droit de se connecter. C'est ce qui est fait dans l'exemple précédent, avec les options « **Deny From All** » et « **Allow From 127.0.0.1** », qui indiquent que seule la machine locale a le droit de se connecter au serveur d'impression.

VI-K - Configuration du lancement automatique des tâches

Il est possible de déclencher l'exécution de certaines opérations à intervalles réguliers sous Linux. Ces opérations sont définies pour le système et pour chaque utilisateur. Elles sont enregistrées dans des fichiers de configuration indiquant le moment où elles doivent être déclenchées, et quelle action elles doivent réaliser. Les opérations définies pour le système sont stockées dans le fichier de configuration **/etc/crontab**. Des commandes additionnelles peuvent être définies dans les répertoires **/etc/cron.d/**, **/etc/cron.daily/**, **/etc/cron.weekly/** et **/etc/cron.monthly/**. Par ailleurs, les fichiers de configuration des utilisateurs sont stockés dans le répertoire **/var/cron/tab/**, sous le nom de chaque utilisateur. Il est bien entendu possible d'éditer ces fichiers en tant que **root**, mais ce n'est pas recommandé. En effet, la commande **crontab** permet d'installer, de supprimer et de consulter les fichiers **crontab** de chaque utilisateur, et ce de manière sûre.

La commande **crontab** peut être utilisée pour afficher le contenu du fichier de configuration de l'utilisateur qui l'appelle, à l'aide de l'option **-l** :

```
crontab -l
```

Elle permet également de supprimer ce fichier, à l'aide de l'option **-r** :

```
crontab -r
```

Enfin, l'option **-e** permet d'éditer le fichier **crontab**, à l'aide de l'éditeur spécifié dans la variable d'environnement **VISUAL** ou **EDITOR**. Par défaut, l'éditeur **vi** sera utilisé.

En tant qu'administrateur du système, il est possible de modifier les paramètres pour n'importe quel utilisateur. Pour cela, il faut préciser le login de l'utilisateur avec l'option **-u**. Il est recommandé d'utiliser également l'option **-u** si l'on a effectué un **su**, car la commande **crontab** peut ne pas pouvoir déterminer l'utilisateur qui l'a appelé dans ce cas.

Le format des fichiers **crontab** est suffisamment riche pour permettre de spécifier avec finesse les conditions d'exécution des opérations programmées. En général, le début du fichier contient la définition de variables d'environnement utilisées par **crontab**. La suite du fichier est réservée aux commandes programmées. Chaque programmation est réalisée sur une ligne du fichier **crontab**. Les lignes contiennent 5 champs spécifiant la date et l'heure à laquelle la commande doit être exécutée, un nom d'utilisateur éventuel et la commande elle-même. Le nom d'utilisateur ne doit être spécifié que dans le fichier **/etc/crontab**, qui définit les commandes du système. Il spécifie alors au nom de quel utilisateur la commande doit être exécutée. Pour les fichiers **crontab** propres à chaque utilisateur, il n'est bien entendu pas nécessaire d'indiquer ce nom.

Les 5 champs de la partie décrivant la date d'exécution de la commande fournissent respectivement les informations suivantes :

- les minutes (comprises entre 0 et 59) ;
- les heures (comprises entre 0 et 23) ;
- le jour dans le mois (compris entre 0 et 31) ;
- le mois (compris entre 0 et 12, ou indiqué par les trois premières lettres du nom du mois en anglais) ;
- le jour dans la semaine (compris entre 0 et 7, ou indiqué par les trois premières lettres du nom du jour en anglais).

Les numéros de mois 0 et 12 correspondent à Janvier, et les numéros de jours 0 et 7 correspondent au Dimanche.

La commande sera exécutée à chaque fois que le jour, le mois, l'heure et les minutes du système correspondront avec ces 5 champs. Il suffit que l'une des spécifications du jour corresponde pour que la commande soit exécutée (c'est-à-dire qu'elle est exécutée une fois pour le jour du mois et une fois pour le jour de la semaine si ces deux champs sont spécifiés).

Il est possible d'utiliser un intervalle de valeurs pour chacun de ces champs, en indiquant la première et la deuxième valeur, séparées d'un tiret. Il est également possible de faire une liste de valeurs et d'intervalles, en séparant chaque donnée par une virgule. Si l'on veut spécifier toutes les valeurs possibles pour un champ, on peut utiliser le caractère '*'. Enfin, il est possible d'indiquer que la commande doit être exécutée toutes les n valeurs pour chaque champ. Pour cela, il suffit de faire suivre le champ d'une barre oblique de division ('/') et du nombre n. Ainsi, si l'on trouve l'expression « */3 » pour les heures, la commande sera exécutée toutes les trois heures.

La spécification de la commande doit être faite sur une seule ligne. Le caractère de pourcentage (%) a une signification spéciale, sauf s'il est précédé d'un antislash (\). Les données qui suivent le premier pourcentage sont passées telles quelles dans l'entrée standard de la commande. Les caractères pourcentages suivants sont interprétés comme des sauts de lignes (donc une validation). Ainsi, la commande suivante :

```
rm -i file.txt%y%
```

permet de supprimer le fichier file.txt et de répondre 'y' à la commande rm. Le caractère 'y' est passé ici dans le flux d'entrée standard de rm.

Comme vous pouvez le voir, le fichier /etc/crontab du système permet de programmer des opérations périodiques, comme les sauvegardes, la destruction des fichiers temporaires, ou toute autre tâche de maintenance. Ne vous étonnez donc pas si votre ordinateur semble s'activer tout seul régulièrement, à heure fixe (par exemple, sur le coup de 11 heures ou minuit). C'est le fonctionnement normal de votre système, qui s'occupe de toutes les tâches ménagères qu'il s'est réservées pour une heure où normalement tout le monde dort...

Les utilisateurs peuvent également définir leur propre crontab pour effectuer les opérations périodiques qu'ils désirent. Par exemple, ils peuvent programmer une commande qui leur rappellera un rendez-vous.

VI-L - Gestion de l'énergie

VI-L-1 - Généralités sur la gestion de l'énergie

La gestion de l'énergie sur les ordinateurs se fait classiquement via l'une des interfaces APM ou ACPI. APM (abréviation de « Advanced Power Management ») est la plus vieille, et s'appuyait essentiellement sur le BIOS pour prendre en charge la gestion de l'énergie. Elle peut encore n'être que la seule possibilité fonctionnelle sur la plupart des vieilles machines, mais est destinée à être remplacée par l'interface ACPI (abréviation de « Advanced Configuration and Power Interface »), développée par Intel. Cette nouvelle interface, bien que souffrant manifestement encore de quelques défauts de jeunesse, est nettement plus souple. En effet, elle se base sur un tout autre principe qu'APM, puisqu'elle permet cette fois de laisser le système prendre en charge la gestion de l'énergie. Étant donné que l'interface APM est vouée à disparaître, et sachant qu'on ne peut utiliser qu'une seule de ces interfaces, seule l'interface ACPI sera présentée dans ce document.

Le principe de fonctionnement de l'ACPI est le suivant. Le BIOS contient une table définie par le fabricant de l'ordinateur qui décrit le matériel fourni, et définit les opérations de base permettant de gérer l'énergie. Ces opérations sont définies dans un langage standard spécifié par Intel. Le système d'exploitation est en charge de lire ces informations et d'interpréter les opérations définies dans la table ACPI. Ainsi, c'est bien le système d'exploitation qui se charge de la gestion de l'ordinateur.

L'implémentation de l'interpréteur ACPI de Linux est l'implémentation de référence réalisée par Intel. Dans le meilleur des mondes, il ne devrait donc y avoir aucun problème pour utiliser l'ACPI sous Linux. Malheureusement, bon nombre de BIOS fournis par les fabricants sont bogués et ne respectent pas la norme ACPI. En effet, l'implémentation ACPI de Microsoft est incorrecte et son interpréteur accepte des erreurs de syntaxe que leur compilateur ACPI laisse passer

(ce qui somme toute est cohérent, ils ne pouvaient pas faire moins sans que le résultat ne cesse de fonctionner). Ce n'est pas le cas de l'interpréteur ACPI de Linux puisque, par définition, elle ne peut se conformer à celle de Microsoft et être compatible bogue à bogue.

Le résultat est, hélas, catastrophique, puisque les fonctionnalités ACPI de ces machines ne peuvent pas être utilisées totalement sous Linux. Microsoft aurait voulu exploiter sa position dominante et corrompre le standard d'Intel pour isoler les concurrents qu'il n'aurait pas fait mieux (avis personnel). La seule solution est de récupérer la table ACPI du BIOS, de la désassembler pour en obtenir le code source, et de tenter de la recompiler avec le compilateur d'Intel (pas celui de Microsoft). Les erreurs apparaissent donc et peuvent ainsi être corrigées. Une fois la table ACPI corrigée, il faut soit convaincre le fabricant de l'ordinateur de mettre un nouveau BIOS à disposition, soit paramétrer le noyau pour qu'il charge cette table au lieu de prendre celle du BIOS pour argent comptant. Corriger une table ACPI boguée est une opération extrêmement technique et absolument hors sujet, et faire en sorte que le noyau l'utilise nécessitait de patcher les sources du noyau jusqu'à la version 2.6.9. À présent, cela est faisable directement à partir du programme de configuration du noyau, mais je n'en parlerais pas plus ici. Vous pouvez consulter le site web de l'ACPI pour plus de détails à ce sujet. Si vous rencontrez des problèmes de ce type, je vous suggère de trouver un Linuxien confirmé, ou de faire une croix sur les fonctionnalités ACPI qui ne sont pas opérationnelles sous Linux.

VI-L-2 - Configuration de la gestion de l'énergie

La prise en charge des fonctionnalités de gestion d'énergie au niveau du noyau se fait simplement en activant les options correspondantes dans la configuration du noyau. Elles sont regroupées dans le menu « Power management options (ACPI, APM) ». Seules les plus intéressantes sont listées ci-dessous :

- l'option « Power Management support » permet d'activer les fonctionnalités de mise en veille du système sur disque. Il est possible ainsi de sauvegarder l'ensemble du système sur la partition d'échange du système, et de le redémarrer et le restaurer (options « Suspend-to-Disk Support » et « Default resume partition »). Cette option permet également d'activer le menu de gestion de l'énergie par APM.
- le menu « ACPI (Advanced Configuration and Power Interface) Support » permet d'activer les fonctionnalités ACPI. Il donne accès à l'option « ACPI Support », qui elle-même donne accès à de nombreuses options pour les différents composants gérant l'ACPI.
- le menu « CPU Frequency scaling » donne accès à l'option « CPU Frequency scaling ». Cette option permet d'activer la gestion des fonctions d'économie d'énergie des processeurs pour les portables. Ces processeurs sont en effet capables de changer de fréquence d'exécution pour, lorsque l'ordinateur est utilisé pour des tâches peu consommatrices de ressources, abaisser la consommation de l'ordinateur et augmenter l'autonomie du système lorsqu'il fonctionne sur batterie. Les sous-options permettent de spécifier les politiques de gestion de l'énergie ainsi que le type de processeur utilisé.

Les fonctionnalités ACPI prises en charge par le noyau sont exposées à l'utilisateur au travers des systèmes de fichiers virtuels `/proc/` et `/sys/`.

Le répertoire `/proc/acpi/` contient essentiellement un sous-répertoire pour chaque composant ACPI pris en charge par le noyau. Dans chacun de ces sous-répertoires, vous trouverez des fichiers contenant les informations courantes sur ces différents sous-systèmes.

Le sous-répertoire `/sys/power/` vous permettra quant à lui de modifier le niveau d'économie d'énergie utilisé par votre ordinateur. En particulier, le fichier `state` vous permettra de le suspendre ou de le mettre en veille. Pour cela, il vous faut simplement écrire le mode désiré dans ce fichier avec la commande `echo` :

```
echo -n mode > /sys/power/state
```

où `mode` est le mode désiré. Vous obtiendrez la liste des modes acceptés par Linux en lisant ce fichier avec la commande `cat` :

```
cat /sys/power/state
```

En général, les modes standby, mem et disk sont disponibles. standby correspond à la mise en veille simple de l'ordinateur. mem permet de réaliser une suspension du système en mémoire, et correspond donc à un niveau d'économie d'énergie supérieur. Enfin, disk permet d'éteindre complètement l'ordinateur après avoir stocké son état dans la partition d'échange. Cette commande correspond au niveau maximum d'économie d'énergie.

Note : Ces fonctionnalités sont relativement expérimentales et peuvent ne pas toutes fonctionner. Généralement, seule la suspension sur disque semble fonctionner, souvent sous condition que certains gestionnaires de périphériques soient déchargés avant la mise en veille.

Si vous disposez d'un portable, il est probable que votre processeur soit capable de fonctionner à différentes fréquences. Il est possible de lire la fréquence courante et de la modifier par l'intermédiaire des fichiers du répertoire `/sys/devices/system/cpu/cpu0/cpufreq/`. La fréquence courante peut être lue via le fichier `cpuinfo_cur_freq`. Il n'est pas possible de modifier directement cette valeur. Toutefois, le noyau donne la possibilité de la modifier via la politique de gestion d'énergie via le fichier `scaling_governor`. Les valeurs possibles pour ce fichier peuvent être lues dans le fichier `scaling_available_governors`.

Généralement, les politiques disponibles sont powersave et performance. Ces politiques permettent respectivement de changer la fréquence du processeur vers les fréquences indiquées dans les fichiers `scaling_min_freq` et `scaling_max_freq`. Ce sont donc dans ces fichiers que l'on pourra définir les fréquences utilisables. La liste des fréquences accessibles est fournie par le fichier `scaling_available_frequencies`.

VI-L-3 - Le démon ACPI

Le noyau expose les fonctionnalités de surveillance de l'ordinateur de l'interface ACPI via le fichier spécial de périphérique `/proc/acpi/event`. Dès qu'un événement se produit (passage de l'alimentation sur batterie pour un portable, hausse anormale de la température du processeur, appui sur le bouton de mise en marche / arrêt, etc.), une ligne signalant cet événement est ajoutée à ce fichier virtuel.

Les applications ne doivent pas lire ce fichier directement, cette tâche étant normalement attribuée au démon `acpid`. Ce démon surveille le fichier d'événement du noyau et fournit un mécanisme de notification plus générique (c'est-à-dire moins spécifique à Linux) aux applications. Ainsi, les applications peuvent se connecter au démon ACPI et être prévenues des événements que le noyau émet.

Le démon ACPI permet aussi de programmer des actions en réponse aux événements ACPI. Pour cela, il consulte les fichiers de configuration placés dans le répertoire `/etc/acpi/events/` et exécute les actions qui y sont enregistrées.

Ces actions sont définies par des paires de lignes événement / action permettant de renseigner, pour chaque type d'événement, l'action à effectuer. Les lignes de définition des événements utilisent des expressions rationnelles pour sélectionner les événements ou les groupes d'événements associés à l'action. Les lignes d'action quant à elles permettent d'indiquer la ligne de commande à exécuter en cas d'apparition de ces événements.

Par exemple, le fichier de configuration par défaut capte l'ensemble des événements ACPI et les passe au script `/etc/acpi/acpi_handler.sh` à l'aide de ces deux simples lignes :

```
event=.*
action=/etc/acpi/acpi_handler.sh %e
```

L'expression rationnelle utilisée dans la ligne de définition des événements signale que toute chaîne de caractères doit correspondre. L'action en réponse est l'exécution du script `acpi_handler.sh`. La chaîne de caractères complète définissant l'événement est passée en premier paramètre et est représentée par le symbole `%e`.

Si les événements intéressants étaient tous les événements concernant le bouton d'alimentation de l'ordinateur, l'expression rationnelle utilisée aurait été la suivante :

```
event=button power.*
```

Comme vous pouvez le constater, le format utilisé pour les descriptions des événements envoyés par le noyau doit être connu pour pouvoir écrire les expressions rationnelles devant les sélectionner. Ces descriptions sont envoyées par le script par défaut `acpi_handler.sh` dans les fichiers de traces du système. Vous pourrez donc les y trouver après avoir généré les événements ACPI correspondants.

Pour donner un exemple, voici comment ce script peut être modifié afin de faire en sorte que la fermeture du couvercle d'un portable le mette en veille immédiatement sur disque :

```
#!/bin/sh

# Ajoute / à la liste des séparateurs pour le découpage de la ligne de commande :
IFS=${IFS}/
set $@

case "$1" in
    button)
        case "$2" in
            power) /sbin/init 0
                ;;
            # Traite l'événement de fermeture du couvercle :
            lid)
                # Vérifie que le couvercle est fermé :
                grep -q "close" /proc/acpi/button/lid/LID/state
                if [ $? -eq 0 ] ; then
                    # Met en veille prolongée l'ordinateur :
                    /usr/sbin/suspend.sh disk
                fi
                ;;
            # Trace les événements de type "bouton" non traités :
            *) logger "ACPI action $2 is not defined"
                ;;
        esac
    ;;
    # Trace tous les autres événements non traités :
    *)
        logger "ACPI group $1 / action $2 is not defined"
    ;;
esac
```

Ce script capte l'événement `lid` (« couvercle » en anglais) et vérifie l'état du capteur de fermeture du couvercle, accessible via le fichier d'état `/proc/acpi/button/lid/LID/state` (l'emplacement de ce fichier peut varier selon les ordinateurs). Si ce fichier indique que le couvercle est fermé, le script utilitaire `suspend.sh` est appelé pour mettre en veille l'ordinateur.

Ce dernier script utilise le fichier `/sys/power/state` pour effectuer cette mise en veille :

```
#!/bin/sh

# Stocke l'heure système (pour mise à jour de /etc/adjtime) :
hwclock --systohc

# Suspend le système et sur disque (méthode sûre) :
echo -n $1 > /sys/power/state

# Rétablit l'heure système :
hwclock --hctosys
```

Comme vous pouvez le constater, il est nécessaire de prendre en compte l'horloge système avant et après une suspension. En effet, sans cela, l'horloge système reprendrait exactement à la date de la suspension lors du réveil. Le stockage de l'heure courante avant suspension est également nécessaire, afin de maintenir le fichier de configuration `/etc/adjtime` cohérent.

Note : Une autre solution aurait été de faire le test sur l'état du bouton dans le script de suspension et d'appeler celui-ci directement en réponse à l'événement « `button lid` ».

VII - Notions de compilation et configuration du noyau

Ce chapitre présente les notions de base de compilation et de fichiers sources de programmes. Il est certain que le lecteur de ce document n'a pas forcément l'intention de programmer sous Linux, cependant, il est nécessaire de connaître les notions qui vont être décrites ici. En effet, il n'est pas rare, voire il est même courant, d'avoir à recompiler une application lorsqu'on travaille avec Linux. Cela n'est pas étonnant, quand on sait que toute bonne installation passe par la recompilation du noyau de Linux ! La raison de cet état de fait provient sans nul doute du fait que la licence GNU impose de fournir les fichiers sources aux utilisateurs d'une part, et que le langage C et Unix sont historiquement fortement liés.

Nous allons commencer par donner la définition des notions de base et les étapes intervenant dans le processus de génération des fichiers binaires des programmes. Nous verrons ensuite comment installer (et compiler !) la suite de compilateurs GCC du projet GNU, et nous présenterons les étapes intervenant dans la compilation et l'installation d'un nouveau noyau. Ces informations pourront être utiles pour la lecture du chapitre traitant de la configuration du matériel puisque, bien souvent, la prise en charge d'un périphérique nécessite la compilation de ses pilotes au niveau du noyau.

VII-A - Notions de base

VII-A-1 - Définition des termes

Le langage naturel est le langage que les êtres humains utilisent pour communiquer, soit oralement, soit par écrit. Le langage naturel est très riche : il utilise des constructions grammaticales et syntaxiques complexes, et il dispose d'un vocabulaire très étendu. Il permet donc d'exprimer la plupart des idées humaines, et c'est bien sa fonction. En revanche, il n'est pas rigoureux, dans le sens où il laisse la porte ouverte à un grand nombre d'ambiguïtés. Ce qui n'est pas dit est souvent sous-entendu, et c'est donc pratiquement entre les mots que se trouve le sens du discours. Il n'est ainsi pas rare de pouvoir interpréter une même phrase différemment, selon le contexte socioculturel, géographique ou temporel dans lequel elle est dite. Les jeux de mots utilisent cette caractéristique à merveille. La conséquence est que l'on ne peut pas utiliser facilement le langage naturel pour décrire rigoureusement, voire mathématiquement, quelque chose.

Un langage formel est, au contraire, un langage restreint, qui ne dispose que de très peu de constructions syntaxiques et d'un vocabulaire très limité. La caractéristique principale des langages formels est qu'ils sont rigoureux, une expression ou une phrase donnée peut être validée selon les règles de syntaxe qui doivent être respectées. Tout texte écrit dans un langage formel est donc soit valide, soit invalide.

En définissant une association entre les constructions d'un langage formel et un jeu de concepts limité, il est possible de donner une sémantique à un langage formel. Cette sémantique donne la possibilité de qualifier de « vraies » ou de « fausses » les assertions écrites dans le langage formel (on notera qu'une expression peut être valide du point de vue de la syntaxe, mais sémantiquement fausse).

Un langage formel permet donc d'exprimer avec précision, sans aucune ambiguïté possible, une idée basée sur les concepts qu'il utilise. Les notations mathématiques constituent un langage formel par excellence.

Note : Par exemple, l'expression mathématique « $x=1$ » est toujours valide, c'est une simple équation. En revanche, elle n'est pas toujours vraie, cela dépend de la valeur de x . En particulier, si x représente 3, on a « $3=1$ », ce qui est valide, mais faux. Notez bien la différence.

Un langage de programmation est un langage formel qui permet de définir les tâches qu'un ordinateur doit effectuer, et de décrire les objets informatiques sur lesquels il doit travailler. Un langage de programmation est donc un code, et tout programme doit être écrit dans un tel langage. Pratiquement, les langages de programmation sont des langages très simples, disposant de constructions du type « si ... alors ... » ou « pour chaque ... fait ... ». Les programmes étant écrits avec de tels langages, il est clair qu'un programme ne fait que ce qui a été écrit : ni plus, ni moins. Il faut donc tout dire à l'ordinateur quand on écrit un programme, ce qui en général est relativement fatigant et compliqué.

C'est le prix à payer pour la rigueur de l'informatique : l'ordinateur ne se trompe jamais, parce qu'il ne fait qu'obéir aux ordres donnés dans un langage formel (donc précis). Celui qui se trompe, c'est en général le programmeur, et assez couramment l'utilisateur.

Note : Notez qu'un programme est un texte valide s'il vérifie la grammaire du langage formel dans lequel il est écrit. Cela ne l'empêchera pas de faire n'importe quoi si on l'exécute. Un programme « vrai » est donc un programme syntaxiquement correctement écrit et qui en plus fait ce que l'on désire qu'il fasse. Il n'y en a pas beaucoup... surtout que dans bien des cas, on ne s'est jamais posé clairement la question de savoir ce que doivent faire les programmes que l'on écrit !

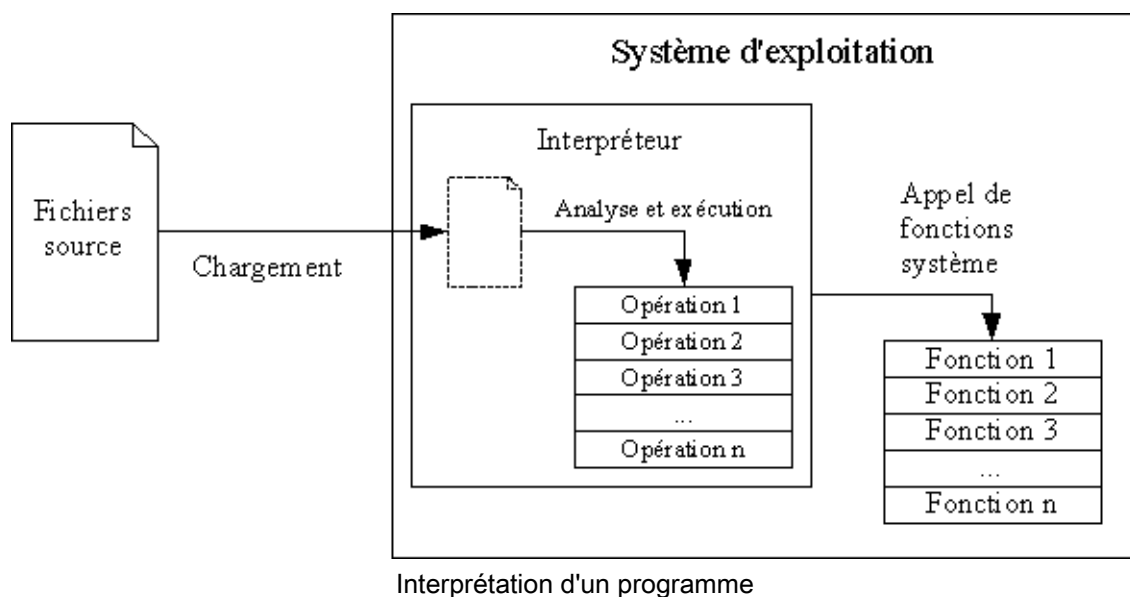
Le C est un langage de programmation relativement simple, qui se trouve sur tous les types d'ordinateurs et de systèmes d'exploitation. Les programmes sont plus difficiles à écrire en C que dans d'autres langages de programmation, mais ils sont plus performants. Le C est très utilisé pour écrire les systèmes d'exploitation : le noyau Linux lui-même est écrit en C. Le C++ est un langage plus évolué, qui est dérivé du C. Il est beaucoup plus riche, et il permet d'écrire des programmes plus compliqués et de les faire évoluer plus facilement. Bien que plus lents que ceux écrits en C, les programmes écrits en C++ sont toujours très performants par rapport à ceux écrits dans d'autres langages.

Les programmes sont en général écrits dans un certain nombre de fichiers. Ces fichiers sont appelés les fichiers sources, du fait qu'ils sont à l'origine du programme. Le texte de ces fichiers est appelé le code source, ou plus simplement le code.

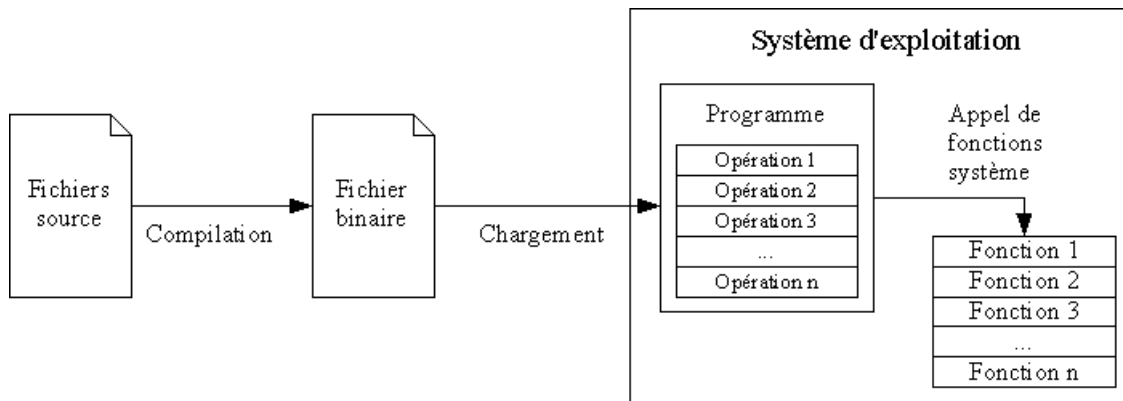
Il va de soi que le code source est écrit dans un langage de programmation, qui n'est pas compréhensible tel quel par le matériel de l'ordinateur. Pour exécuter un programme à partir de son code source, il n'y a que deux solutions :

- disposer d'un autre programme, nommé interpréteur, capable de lire le code source et d'effectuer les opérations décrites dans le code source ;
- disposer d'un autre programme, nommé compilateur, capable de traduire le code source en langage binaire, qui sera alors directement exécutable par l'ordinateur.

Les programmes interprétés sont fatalement relativement lents, puisque l'interpréteur doit analyser en permanence les fichiers sources pour le traduire en opération à exécuter à la volée.



En revanche, les programmes compilés sont beaucoup plus rapides à l'exécution, puisque la phase d'analyse du code source a été réalisée au préalable et se fait en dehors de la phase d'exécution. Ces programmes peuvent donc s'exécuter nativement, sans être ralentis par l'interpréteur.



Compilation et exécution d'un programme

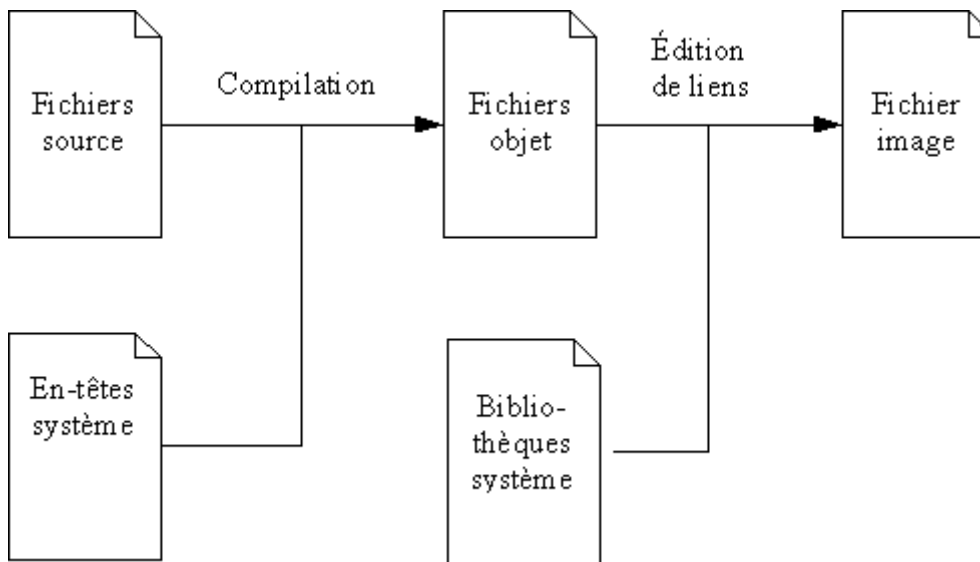
Note : Si vous avez bien suivi, on se trouve face au problème de l'œuf et de la poule. En effet, les interpréteurs et les compilateurs sont eux-mêmes des programmes écrits dans un langage informatique. Quel est donc le premier compilateur ou interpréteur ? Ce problème a effectivement dû être résolu au début de l'informatique, de la manière la plus simple : les premiers programmeurs entraînaient directement le langage machine en binaire dans les ordinateurs (via les câblages ou les cartes perforées...). Quand on sait le travail que cela représentait, et le nombre d'erreurs que cela a pu générer, on comprend facilement pourquoi les compilateurs et les interpréteurs font partie des tout premiers programmes qui ont été développés...

Dans les systèmes Unix, les deux types de programmes existent. Les programmes interprétés constituent ce que l'on appelle des scripts. Le système utilise le shell pour les interpréter. La plupart des opérations d'administration du système utilisent des scripts, car il est toujours plus facile d'écrire et de tester un script que d'écrire un programme en C. Les programmes compilés sont notamment le noyau lui-même, le shell, les commandes de base et les applications. Ce sont eux qui en fin de compte effectuent le travail, et ils sont souvent appelés à partir de scripts. Nous avons déjà vu des exemples de scripts lors de la configuration du système. Pour l'instant, nous allons nous intéresser au langage C et au processus de compilation.

La compilation est l'opération qui consiste à lire un fichier source et à le traduire dans le langage binaire du processeur. Ce langage est absolument incompréhensible par les êtres humains, cependant, il existe un langage de programmation qui permet de coder directement le binaire : il s'agit de l'assembleur.

En général, le processus de compilation génère un fichier binaire pour chaque fichier source. Ces fichiers binaires sont nommés fichiers objets, et porte de ce fait l'extension « .o » (ou « .obj » dans les systèmes Microsoft). Comme un programme peut être constitué de plusieurs fichiers sources, il faut regrouper les fichiers objets pour générer le fichier exécutable du programme, fichier que l'on appelle également le fichier image en raison du fait que c'est le contenu de ce fichier qui sera chargé en mémoire par le système pour l'exécuter et qu'il contient donc l'image sur disque des instructions des programmes en cours d'exécution. L'opération de regroupement des fichiers objets pour constituer le fichier image s'appelle l'édition de liens, et elle est réalisée par un programme nommé le linker (éditeur de liens en français). Ce programme regarde dans tous les fichiers objets les références partielles aux autres fichiers objets et, pour chaque « lien » ainsi trouvé, il complète les informations nécessaires pour en faire une référence complète. Par exemple, un fichier source peut très bien utiliser une fonctionnalité d'un autre fichier source. Comme cette fonctionnalité n'est pas définie dans le fichier source courant, une référence partielle est créée dans le fichier objet lors de sa compilation, mais il faudra tout de même la terminer en indiquant exactement comment accéder à la fonctionnalité externe. C'est le travail de l'éditeur de liens, lorsqu'il regroupe les deux fichiers objet.

Certains fichiers objet sont nécessaires pour tous les programmes. Ce sont notamment les fichiers objet définissant les fonctions de base, et les fonctions permettant d'accéder au système d'exploitation. Ces fichiers objet ont donc été regroupés dans des bibliothèques (également couramment appelées « librairies »), que l'on peut ainsi utiliser directement lors de la phase d'édition de liens. Les fichiers objets nécessaires sont alors lus dans la bibliothèque et ajoutés au programme en cours d'édition de liens. Les bibliothèques portent souvent l'extension « .a » (ou « .lib » dans les systèmes Microsoft).



Processus de génération d'un exécutable

Malheureusement, la solution consistant à stocker dans des bibliothèques les fonctions les plus utilisées souffre de la duplication du code contenu dans ces bibliothèques dans tous les programmes, d'où une perte de place considérable. De plus, la mise à jour d'une bibliothèque nécessite de refaire l'édition de liens de tous les programmes qui l'utilisent, ce qui n'est pas réalisable en pratique. C'est pour cela que les bibliothèques dynamiques ont été créées : une bibliothèque dynamique n'est pas incluse dans les fichiers des exécutables qui l'utilisent, mais reste dans un fichier séparé. Les bibliothèques sont regroupées dans un répertoire bien défini du système de fichiers, ce qui permet de les partager entre différents programmes. Ainsi, la mise à jour d'une bibliothèque peut se faire sans avoir à toucher tous les programmes qui l'utilisent. Le problème est cependant que l'édition de liens reste incomplète, parce que les références aux objets des bibliothèques dynamiques sont toujours externes. Il existe donc un programme spécial, l'éditeur de liens dynamiques (« ld », pour « Link Dynamically »), qui résout les dernières références incomplètes lors du chargement de chaque programme. Les bibliothèques dynamiques portent l'extension « .so » (pour « Shared Object »), ou « .dll » dans les systèmes Microsoft (pour « Dynamic Link Library »).

Évidemment, le chargement des programmes est un peu plus lent avec les bibliothèques dynamiques, puisqu'il faut réaliser l'édition de liens dynamiques lors de leur lancement. Cependant, ce processus a été optimisé, et les formats de fichiers binaires utilisés contiennent toutes les informations précalculées pour faciliter la tâche de l'éditeur de liens dynamiques. Ainsi, la différence de performance est devenue à peine décelable.

VII-A-2 - Processus de génération

Classiquement, les fichiers sources C et C++ se divisent en deux grandes catégories :

- les fichiers d'en-tête, qui contiennent les déclarations des symboles du programme ;
- les fichiers C (ou C++), qui contiennent leurs définitions.

Le fait de faire cette distinction entre la déclaration (qui consiste à dire : « telle chose existe ») et la définition (qui consiste à décrire la chose précédemment déclarée) permet de faire en sorte que l'on peut utiliser les fichiers objets sans avoir les fichiers sources. En effet, la déclaration permet de réaliser les références partielles dans les programmes, tandis que les fichiers objet contiennent bien entendu la définition binaire de ce qui a été déclaré.

Pour compiler un programme, on n'a donc réellement besoin que de trois types de fichiers :

- les fichiers de déclaration du système et des bibliothèques de base ;
- les fichiers des bibliothèques de base eux-mêmes ;
- les fichiers source (de déclaration et de définition) du programme à compiler.

En C et C++, les fichiers sources de déclaration portent l'extension « .h » (plus rarement « .hpp » pour les fichiers de déclaration C++), et les fichiers de définition portent l'extension « .c » pour les programmes écrits en C et « .C » (ou « .cpp », ou encore « .c++ ») pour les programmes écrits en C++.

Il va de soi que la réalisation d'un programme peut nécessiter la création d'un grand nombre de fichiers, tous plus ou moins dépendants les uns des autres. Pour pouvoir s'y retrouver plus facilement, les programmeurs utilisent un programme spécial : make. Ce programme est capable de réaliser toutes les opérations nécessaires à la création des fichiers exécutables d'un programme à partir de ses fichiers sources. Pour cela, il utilise un fichier contenant la définition des opérations à réaliser, ainsi que la description des dépendances entre les différents fichiers sources. Ce fichier est appelé le fichier makefile. Grâce à make, l'utilisateur de base que vous êtes va pouvoir compiler la plupart des programmes sans avoir la moindre notion de programmation.

En général, la compilation d'un programme passe par les étapes suivantes :

- récupération des fichiers sources du programme ;
- configuration du programme pour l'environnement courant ;
- appel à make pour la compilation ;
- appel à make pour l'installation.

La première étape est élémentaire et va de soi.

La deuxième étape se fait souvent en appelant un script dans le répertoire d'installation des fichiers sources. Ce script se nomme souvent configure, et peut être appelé avec la ligne de commande suivante :

```
./configure
```

à partir du répertoire où se trouvent les fichiers sources. Ce script effectue tous les tests sur le système et l'environnement de développement, et génère le fichier makefile. Le programme configure est en général fourni avec les logiciels GNU. Il permet de déterminer l'environnement logiciel et système et de générer le fichier makefile et des fichiers d'en-têtes contenant la définition de quelques options du programme. Dans quelques cas particuliers, vous aurez à utiliser des options de configure afin de préciser certains paramètres. L'option --host permet d'indiquer le type de la machine cible, dans le cas où configure ne parvient pas à le déterminer automatiquement :

```
--host=machine
```

où machine est la description de votre machine. Pour les PC fonctionnant sous Linux, cette description est de la forme « ix86-pc-linux-gnu », où le 'x' représente le numéro de la génération du processeur du PC. Par exemple, pour un Pentium ou un AMD K6, la description sera « i586-pc-linux-gnu ». L'option --enable-shared permet de générer des bibliothèques dynamiques lors de la compilation. Cela procure un gain de place évident. Enfin, l'option --prefix permet de préciser le répertoire de base dans lequel le programme sera installé a priori. Cette option s'utilise de la manière suivante :

```
--prefix=répertoire
```

où répertoire est le répertoire de base de l'installation. La connaissance de ce répertoire est utile aux autres programmes pour localiser les composants qui vont être installés. Dans la plupart des cas, la valeur par défaut du préfixe convient, mais il est parfois nécessaire de le modifier. C'est en particulier le cas lorsque vous désirez mettre à jour un programme déjà installé et que votre distribution n'utilise pas le répertoire par défaut. Dans ce cas, il convient d'indiquer le répertoire de base dans lequel ce programme a été installé, afin de le remplacer au lieu de le dupliquer d'une part, et afin que les autres programmes puissent trouver la nouvelle version à la place de l'ancienne d'autre part.

La troisième étape est très simple aussi. Il suffit de taper la commande suivante :

```
make
```

toujours dans le répertoire où se trouvent les fichiers sources.

Enfin, la dernière étape se fait en tapant la commande suivante :

```
make install
```

Bien entendu, ces différentes étapes varient parfois avec le logiciel à installer. Cependant, il existe quasiment toujours un fichier texte indiquant comment effectuer ces opérations dans le répertoire des fichiers sources. Ce fichier porte souvent le nom de « `readme` », « `install` », ou quelque chose de similaire.

Avec les notions que vous venez de voir, vous devriez pouvoir compiler quasiment tous les programmes que vous pourrez récupérer sous la forme de fichiers sources. Un dernier détail cependant : la compilation est une opération très gourmande en ressources. Cela signifie qu'elle peut consommer beaucoup de ressources processeur, de mémoire, de disque et de temps. Pour des gros programmes, il n'est pas rare de passer jusqu'à une heure de compilation, même sur une machine récente. Notez également que les facteurs limitants dans les compilations sont souvent la rapidité du processeur et du disque dur, ainsi que la quantité de mémoire vive disponible.

VII-B - Compilation de GCC

La clef de voûte du processus de compilation est bien entendu la suite de compilateurs GCC du projet GNU. Cette suite comprend un compilateur pour les langages C, C++, Objective C, Fortran, Java et Ada. La plupart des distributions fournissent ces outils sur leur CD d'installation, mais ne les installent pas par défaut. Il est toutefois recommandé d'installer au moins le compilateur C/C++ et les outils de développement associés, ainsi que les fichiers d'en-tête des bibliothèques de base du système (bibliothèque C, en-têtes de XWindow, des bibliothèques utilitaires et des bibliothèques de base des environnements graphiques Gnome et KDE).

Le choix de la version de GCC utilisée est extrêmement important, car la stabilité et les performances du système en dépendront directement. De plus, certains programmes exigeront une version spécifique du compilateur pour être compilés, soit parce qu'ils n'ont été testés qu'avec cette version par les développeurs, soit parce qu'ils utilisent une fonctionnalité qui n'est disponible que dans les versions récentes de GCC.

Les versions de GCC les plus utilisées sont actuellement les versions issues des branches 3.4 et 4.1. Il est déconseillé d'utiliser les versions les plus récentes (typiquement la version de la branche courante), car elles peuvent souffrir de bogues de jeunesse et elles peuvent être trop strictes pour compiler des programmes qui ne sont pas tout à fait à la norme (en particulier au niveau du langage C++, qui peut avoir des constructions d'usage courant et néanmoins extrêmement compliquées). Ce document suppose que votre distribution utilise la version 4.1.2 de GCC, qui offre les avantages suivants :

- elle permet de compiler des programmes pour les processeurs les plus récents, tels que les IA-64 d'Intel, les x86-64 d'AMD et les PowerPC d'IBM ;
- elle permet d'utiliser les instructions spécifiques des processeurs les plus récents, même d'architecture x86 ;
- elle génère un code bien plus optimisé que les versions précédentes ;
- elle permet de réduire la visibilité des fonctions, variables et types non publics des bibliothèques, et donc de réduire l'empreinte des binaires aussi bien en mémoire que sur disque et de réduire le temps de chargement des programmes récents ;
- la bibliothèque standard C++ fournie avec cette version respecte nettement mieux la norme C++ que les versions antérieures, et elle permet d'utiliser les caractères Unicode dans les programmes C++.

On prendra garde au fait que certaines distributions utilisent régulièrement des versions non stabilisées des programmes de base tels que GCC, la bibliothèque C ou le noyau. C'est en particulier le cas de la Redhat et de toutes ses dérivées, telle la Mandrake par exemple. Tout cela est fort dommageable, car les paquetages binaires de ces distributions sont systématiquement incompatibles avec ceux des autres d'une part, et il est quasiment impossible d'envisager une compilation correcte d'un programme avec ces distributions d'autre part. En pratique, les possesseurs de distributions Redhat et dérivées s'abstiendront donc de faire une mise à jour manuelle de GCC, de la bibliothèque C ou même du noyau, car les programmes installés sur leur machine risqueraient de ne plus fonctionner. Les utilisateurs de ces distributions devraient faire une mise à jour complète de leur distribution dans ce cas.

Note : En ce qui concerne les numéros de version, il faut savoir que le dernier chiffre caractérise souvent le numéro de correctif. Plus ce chiffre est élevé, moins le programme a de chances de comporter de bogues. En pratique, on évitera généralement d'installer systématiquement les dernières versions des logiciels lorsque ces versions sont les premières d'une nouvelle série. Ainsi, un programme de version x.y.0 est certainement peu fiable. Il faudra attendre au moins la version x.y.1 ou mieux la x.y.2, pour pouvoir l'utiliser sereinement. Sachez également que le numéro intermédiaire sert, pour certains logiciels, à indiquer les versions de développement. C'est par exemple le cas du noyau Linux, pour lequel les numéros de version impairs correspondent aux versions de développement et les numéros pairs aux versions stables. Ce n'est pas le cas pour la bibliothèque C ou GCC. De plus, certains logiciels utilisent un schéma de version à quatre chiffres, le dernier chiffre correspondant généralement aux correctifs mineurs n'entraînant théoriquement aucune régression. C'est également le cas du noyau Linux depuis la version 2.6.8. Ainsi, dans un numéro de version x.y.z.t, x représente généralement la version majeure du logiciel (et n'est changée qu'en cas d'incompatibilité majeure ou de refonte complète), y est le numéro de la branche (une convention permettant de distinguer les branches de développement des branches stables y est parfois appliquée), z est le numéro de version dans la branche, et t le numéro du patch regroupant les corrections de sécurité ou de petite taille.

VII-B-1 - Prérequis

À moins que vous ne réussissiez à mettre la main sur une version de GCC déjà compilée pour Linux, vous ne disposerez que des sources. Les sources de GCC peuvent être récupérées sur Internet sur le [site de GCC](#). Ces sources sont fournies sous la forme d'une archive au format tar-gzip. Le nom de cette archive est « gcc-3.4.4.tar.gz ».

Un autre prérequis est bien entendu que vous disposiez également d'un compilateur C sur votre système. Ce compilateur peut très bien être une ancienne version de GCC, mais il est impératif de disposer d'un compilateur C correct. Le code de GCC a été écrit pour pouvoir être compilé de manière minimale avec la plupart des compilateurs C existants. Il faut également que ce compilateur porte le nom de « cc » ou « gcc » et soit dans l'un des répertoires de la variable d'environnement PATH, ou que la variable CC ait été définie avec le chemin absolu du compilateur.

VII-B-2 - Installation des sources

Lorsque vous aurez installé le compilateur C natif de votre système, vous devrez décompresser les sources de GCC. Cela doit se faire avec la commande suivante :

```
tar xvfz archive
```

où archive est le nom de l'archive compressée contenant ces sources. Dans notre cas, ce doit être gcc-4.1.2.tar.gz. Cette commande a pour conséquence de créer l'arborescence des répertoires des sources de GCC dans le répertoire où elle a été exécutée.

Une fois cette opération effectuée, il est conseillé de créer un autre répertoire, ailleurs que dans le répertoire des sources de GCC, dans lequel la compilation aura lieu. Dans toute la suite, nous supposons que le répertoire des sources se nomme `srcdir`, et que le répertoire de compilation se nomme `objdir`.

VII-B-3 - Configuration

L'étape suivante est d'aller dans le répertoire de compilation et de lancer le programme de configuration de GCC dans ce répertoire. Cela peut être réalisé avec les commandes suivantes :

```
cd objdir
srcdir/configure [options]
```

Comme on l'a vu ci-dessus, le programme de configuration configure peut recevoir des options en ligne de commande. Il est recommandé de fournir l'option `--enable-__cxa_atexit`, qui permet de rendre GCC compatible avec la version 3 de la norme intervenant spécifiant le format des interfaces binaires pour les compilateurs C++, et donc de garantir la compatibilité binaire ascendante avec les versions futures de GCC. Cela vous permettra d'éviter d'avoir à recompiler toutes les bibliothèques C++ lors des mises à jour ultérieures de GCC. Il est également recommandé d'utiliser l'option

`--enable-long-long`, qui ajoute le support du type de données « long long, » qui permet de manipuler des données 64 bits sur les machines 32 bits comme les PC, ainsi que l'option `--enable-threads`, qui permet d'ajouter le support des threads aux langages.

Normalement, le programme de configuration détecte le type de machine et le système utilisé. Cependant, cette détection peut échouer si les types de machines indiqués pour les différents composants du système ne sont pas identiques. Dans ce cas, il vous faudra spécifier la machine hôte à l'aide de l'option `--host`. La ligne de commande pour la configuration sera alors la suivante :

```
srcdir/configure --enable-__cxa_atexit \
--enable-long-long --enable-threads --host=configuration
```

où `configuration` est un triplet de la forme `i686-pc-linux-gnu`.

Le nombre de compilateurs fournis avec la suite de compilateurs GCC est impressionnant. Cependant, seuls les compilateurs C et C++ sont réellement essentiels. Aussi est-il possible de ne prendre en charge que certains langages à l'aide de l'option `--enable-languages`. Cette option doit être suivie des noms des langages pour lesquels le compilateur doit être créé, séparés par des virgules. Ainsi, on peut n'installer que les compilateurs C/C++ et Java en ajoutant l'option suivante à la ligne de commande :

```
--enable-languages=c,c++,java
```

Enfin, si votre distribution utilise le compilateur GCC comme compilateur de base du système, et que vous désirez remplacer la version de votre distribution par la version que vous allez compiler, il faut changer le répertoire d'installation de GCC. Normalement, GCC s'installe dans le répertoire `/usr/local/` ce qui fait qu'il ne remplace pas la version de votre distribution. Vous devez donc spécifier les répertoires de base pour l'installation. Pour cela, il suffit d'ajouter les options suivantes :

```
--prefix=/usr
```

à la ligne de commande de `configure`.

VII-B-4 - Compilation

La compilation de GCC peut ensuite être réalisée. Autant vous prévenir tout de suite : c'est une opération très longue, qui de plus demande beaucoup d'espace disque. Il faut au moins prévoir 800 Mo d'espace disque, et presque trois heures sur une machine à 500 Mhz. Cela est dû au nombre de langages supportés par les versions récentes de GCC, à la technique employée lors de sa compilation. La plupart des compilateurs C ne sont en effet pas capables de compiler GCC avec toutes ses fonctionnalités, aussi la compilation se déroule-t-elle en trois étapes :

- une version allégée de GCC est compilée avec le compilateur natif du système dans une première passe ;
- cette version est utilisée pour compiler la version complète de GCC ;
- la version complète est utilisée pour se recompiler, afin de tester les différences entre les deux versions complètes.

Ainsi, GCC se compile lui-même !

Ces trois opérations peuvent être exécutées à l'aide d'une seule commande :

```
make bootstrap
```

Cette commande doit être exécutée à partir du répertoire de compilation `objdir`.

VII-B-5 - Installation de GCC

Lorsque la compilation s'est terminée, vous pouvez installer GCC. Il est recommandé de supprimer le compilateur que vous avez utilisé pour compiler GCC, sauf si, bien entendu, il s'agissait déjà de GCC. En effet, il n'est pas nécessaire de le conserver, puisque vous utiliserez désormais GCC. L'installation de GCC est, encore une fois, très simple :

```
make install
```

Cette commande installe GCC ainsi que la bibliothèque standard C++.

VII-C - Compilation du noyau Linux

La compilation du noyau est une spécificité des systèmes libres, qui n'est possible que parce que l'on dispose des sources du noyau. Cependant, même pour certains Unix commerciaux, il est possible d'effectuer une édition de liens, les modules du noyau étant fournis sous la forme de fichiers objet. La compilation ou l'édition de liens du noyau est une opération technique qui peut surprendre un habitué des systèmes fermés que sont par exemple Windows ou OS/2. Cependant, elle permet d'obtenir un noyau très petit, optimisé pour la machine sur laquelle il tourne, et donc à la fois économe en mémoire et performant. Il est donc recommandé d'effectuer cette compilation : pourquoi conserver un monstre capable de gérer des périphériques qui ne sont pas et ne seront jamais installés sur votre système ?

La compilation du noyau de Linux nécessite de disposer des sources du noyau fourni par votre distribution. En effet, la plupart des distributions modifient ces sources pour y ajouter des fonctionnalités ou des correctifs de sécurité, ce qui permet de ne pas prendre le risque d'utiliser la dernière version officielle du noyau, qui peut ne pas être considérée comme stable, car trop récente. Parfois même, ce sont des fonctionnalités non encore incluses dans le noyau officiel qui sont utilisées par les distributions, ce qui rend l'utilisation du noyau générique impossible. Rien ne vous empêche d'utiliser pour autant la dernière version officielle du noyau, mais vous vous exposez au risque d'avoir un système non fonctionnel, ou au moins à diagnostiquer des problèmes encore inconnus ou d'avoir à mettre à jour d'autres composants de votre distribution manuellement en raison des évolutions que ce noyau peut avoir subi.

Le noyau utilise des constructions spécifiques à gcc et ne peut donc être compilé de manière fiable que par ce compilateur (même si d'autres compilateurs compatibles avec GCC sont utilisables moyennant quelques efforts).

Note : La documentation du noyau indique que la version de gcc à utiliser pour le compiler est la version 3.2. La raison en est que les développeurs du noyau eux-mêmes changent rarement de compilateur, afin de ne pas avoir à se préoccuper d'éventuels bogues de celui-ci et ainsi de pouvoir se concentrer sur leur travail. La version 3.2 n'ayant pas de bogues majeurs, elle a été retenue, et les développeurs principaux sont supposés utiliser cette version. Ainsi, le noyau est théoriquement testé avec cette version, c'est la raison pour laquelle elle est recommandée. Toutefois, les versions plus récentes de GCC sont également tout à fait valables de manière générale, et parfois même elles sont nécessaires pour les architectures les plus récentes. Ainsi, les architectures 64 bits requièrent au minimum GCC 3.4... Quoi qu'il en soit, il n'y a généralement aucun problème à compiler le noyau avec les versions récentes de GCC, et en pratique on pourra utiliser la version installée dans le système sans inquiétude.

La compilation du noyau n'est pas très difficile. Elle nécessite cependant de répondre correctement aux questions de configuration. Les erreurs peuvent être multiples, et souvent fatales. Il est donc fortement conseillé de disposer d'une disquette de démarrage afin de réparer le système en cas d'erreur. Par ailleurs, il faut toujours conserver le dernier noyau utilisable en sauvegarde dans le répertoire `/boot/`. Il faut également ajouter une entrée spécifiant ce noyau dans le programme de démarrage (lilo), afin de pouvoir sélectionner l'ancien noyau en cas d'erreur. Ces opérations seront décrites en détail plus loin.

La compilation du noyau se passe en quatre étapes :

- installation des fichiers sources ;
- réponse aux questions de configuration ;
- compilation et installation du noyau ;
- compilation et installation des modules.

VII-C-1 - Installation des sources de Linux

Les sources du noyau sont normalement fournies par votre distribution. Vous pouvez toutefois trouver les sources génériques du noyau « officiel » sur le site kernel.org. Il est possible de récupérer les sources complètes, sous la forme d'une archive compressée d'environ 43 Mo. Toutefois, si l'on dispose déjà d'une version complète des fichiers sources, il est envisageable de ne télécharger que les fichiers différentiels de cette version à la version courante (ce que l'on appelle classiquement des « patches »).

Note : Les utilisateurs des distributions Redhat dérivées (telle que la Mandrake par exemple) doivent impérativement utiliser les sources fournies par leur distribution. En effet, les noyaux de ces distributions sont patchés à mort pour utiliser des fonctionnalités non stabilisées de la version de développement du noyau de Linux, et sans lesquelles la plupart des programmes et la bibliothèque C elle-même ne fonctionneraient plus. Je déconseille aux utilisateurs de ces distributions de se lancer dans l'aventure d'une quelconque compilation d'un des composants système.

Il est recommandé d'installer les sources du noyau dans un autre répertoire que celui où se trouvent les fichiers sources de votre distribution, car ceux-ci contiennent les fichiers d'en-tête C qui ont été utilisés pour générer la bibliothèque C du système et sont donc nécessaires à la compilation des programmes. Le remplacement des fichiers sources du noyau imposerait donc, en toute rigueur, de recompiler la bibliothèque C du système. Cette opération est extrêmement technique, risquée et longue, de plus, les fichiers d'en-tête du noyau évoluent souvent et provoquent des incompatibilités avec les programmes qui les utilisent. Les plus motivés trouveront toutefois en annexe la manière de procéder pour recompiler la bibliothèque C.

Généralement, les fichiers sources de Linux sont installés dans le répertoire `/usr/src/linux/`. Certaines distributions font une copie des fichiers d'en-tête du noyau qui ont servi pour la génération de la bibliothèque C dans les répertoires `/usr/include/linux/`, `/usr/include/asm/` `/usr/include/asm-generic`. Si ce n'est pas le cas, il faudra éviter d'écraser le contenu du répertoire `/usr/src/linux/` lors de l'installation des sources du nouveau noyau. Une solution est par exemple d'installer les fichiers du noyau dans le répertoire `/usr/src/linux<version>/` et de ne pas toucher au répertoire des sources originel `/usr/src/linux/`. Les commandes suivantes permettront d'extraire les sources dans le répertoire dédié aux sources de Linux :

```
cd /usr/src
tar xvfz linux-2.6.13.2.tar.gz
```

Si l'on dispose déjà d'une version complète des fichiers sources, et que l'on désire appliquer un patch, il faut décompresser le fichier de patch avec la commande suivante :

```
gunzip fichier.gz
```

où `fichier.gz` représente le fichier de patch compressé (en supposant qu'il ait été compressé à l'aide de `gzip`). L'application du patch se fait de la manière suivante :

```
patch -p1 < fichier
```

Cette commande doit être lancée à partir du répertoire des sources du noyau (par exemple `/usr/src/linux-2.6.13.2/`). Dans cette ligne de commande, `fichier` représente le nom du fichier de patch précédemment décompressé, et l'option `-p1` indique au programme `patch` d'utiliser les noms de répertoires relatifs au répertoire parent (à savoir le répertoire contenant le répertoire des sources du noyau, donc `/usr/src/` dans notre cas). Si le patch doit être appliqué depuis un autre répertoire, il faudra éventuellement modifier l'option `-px` passée en paramètre au programme `patch`, où `x` est le nombre de niveaux de répertoires à ignorer pour l'application du patch. Consultez la page de manuel `patch` pour plus de détails sur cette option.

VII-C-2 - Choix des options de configuration du noyau

La configuration du noyau peut se faire à l'ancienne avec la commande suivante :

```
make config
```

Cette commande pose une série de questions auxquelles il faut pouvoir répondre correctement du premier coup. On n'a pas le droit à l'erreur ici, faute de quoi il faut tout reprendre à zéro.

Il est nettement préférable d'utiliser la version texte, qui fournit les options de configuration sous la forme de menus. Cela peut être réalisé avec la commande suivante :

```
make menuconfig
```

Certains préféreront l'une des versions X11 du programme de configuration, que l'on peut obtenir avec les commandes

```
make xconfig
```

ou

```
make gconfig
```

Sachez cependant que certaines options ne sont toutefois pas correctement proposées dans ce mode, et que la version textuelle du programme de configuration reste recommandée.

Quelle que soit la méthode utilisée, il faut répondre par 'Y' (pour « Yes »), 'N' (pour « No ») ou 'M' (pour « Module ») lorsque c'est possible. 'Y' et 'M' incluent la fonctionnalité courante dans le noyau, 'N' la supprime. 'M' permet d'utiliser la fonctionnalité en tant que module du noyau. En général, l'utilisation des modules permet d'alléger le noyau, car les fonctionnalités sont chargées et déchargées dynamiquement. Cependant, les fonctionnalités nécessaires au démarrage de Linux, comme les gestionnaires de disques et systèmes de fichiers par exemple, ne doivent en aucun cas être placées dans des modules, car le système ne pourrait alors pas démarrer.

Le choix des options de configuration est réellement très large, car celles-ci couvrent un ensemble de fonctionnalités très large. La description exhaustive de ces options est à la fois fastidieuse et inutile, car vous n'utiliserez pas tous les gestionnaires de périphériques et toutes les fonctionnalités de Linux avec un même ordinateur. Il s'agit donc de répondre aux questions appropriées pour votre configuration, mais de le faire avec rigueur : la moindre erreur de configuration peut empêcher votre système de fonctionner correctement, voire l'empêcher de démarrer tout simplement. Vous trouverez une description rapide des principales options de configuration dans l'[Annexe A](#). Les options les plus utiles seront également décrites lorsque cela sera nécessaire dans le [chapitre de configuration du matériel](#).

VII-C-3 - Compilation et installation du noyau

Une fois la configuration du noyau réalisée, la compilation peut être lancée. Pour cela, il suffit de lancer la simple commande make dans le répertoire `/usr/src/linux`.

Une fois la compilation achevée, il faut installer le nouveau noyau. Cette opération nécessite beaucoup de prudence, car si le noyau nouvellement créé n'est pas bon, le système ne redémarrera pas. C'est pour cela qu'il est conseillé de conserver toujours deux versions du noyau, dont on est sûr que l'une d'entre elles fonctionne parfaitement. En pratique, cela revient à conserver la version originale du noyau installé par votre distribution. Pour cela, il faut en faire une copie de sauvegarde.

En général, le noyau est installé dans le répertoire `/boot/` (ou dans le répertoire racine pour les anciennes versions de Linux). Il porte souvent le nom de `vmlinuz`. Pour le sauvegarder, il suffit donc de taper par exemple la commande suivante :

```
cp vmlinuz vmlinuz.old
```

Il faut également indiquer au gestionnaire d'amorçage qu'il faut qu'il donne maintenant la possibilité de démarrer l'ancienne version du noyau sous ce nouveau nom. Pour LILO, il suffit d'éditer le fichier `/etc/lilo.conf` et d'y ajouter une nouvelle configuration. En pratique, cela revient à dupliquer la configuration du noyau actuel et à changer simplement le nom du noyau à charger (paramètre « `image` » de la configuration dans `/etc/lilo.conf`) et le nom de la configuration (paramètre « `label` »). Vous devrez aussi rajouter l'option « `prompt` » si elle n'y est pas déjà, afin que LILO vous demande la configuration à lancer à chaque démarrage. Dans notre exemple, le nom du noyau à utiliser pour la configuration de sauvegarde sera `vmlinuz.old`. De même, si la configuration initiale de Linux porte le nom « `linux` », vous pouvez utiliser le nom « `oldlinux` » pour la configuration de sauvegarde.

Une fois le fichier `lilo.conf` mis à jour, il faut vérifier que l'on peut bien charger l'ancien système. Pour cela, il faut réinstaller LILO et redémarrer la machine. La réinstallation de LILO se fait exactement de la même manière que son installation, simplement en l'invoquant en ligne de commande :

```
lilo
```

Si LILO signale une erreur, vous devez corriger immédiatement votre fichier `lilo.conf` et le réinstaller.

Pour le GRUB, la définition d'une nouvelle configuration se fait également en dupliquant la configuration initiale et en changeant le nom de l'option de menu du GRUB et le chemin sur le fichier du noyau sauvegardé. Veillez également à bien ajouter une option `timeout` pour avoir la moindre chance de sélectionner la configuration à lancer. Tout cela doit être effectué dans le fichier de configuration `/boot/grub/menu.lst`. Contrairement à LILO, il n'est pas nécessaire de réinstaller le GRUB pour que les modifications de ce fichier soient prises en compte au démarrage suivant.

Vous pourrez alors redémarrer la machine avec la commande suivante :

```
reboot
```

Le gestionnaire d'amorçage utilisé vous propose alors de choisir le système d'exploitation à lancer. Il faut ici sélectionner la configuration de sauvegarde pour vérifier qu'elle est accessible et fonctionne bien. Le système doit alors démarrer en utilisant la copie sauvegardée du noyau. Si cela ne fonctionne pas, on peut toujours utiliser le noyau actuel en sélectionnant le noyau initial et en corrigeant la configuration du gestionnaire d'amorçage.

Lorsque vous vous serez assuré que le système peut démarrer avec la sauvegarde du noyau, vous pourrez installer le nouveau noyau. Son image a été créée par `make` dans le répertoire `/usr/src/linux/arch/i386/boot/`, sous le nom `bzImage`. L'installation se fait donc simplement par une copie dans `/boot/` en écrasant le noyau actuel `vmlinuz` :

```
cp /usr/src/linux/arch/i386/boot/bzImage /boot/vmlinuz
```

Il faut également copier le fichier `System.map` du répertoire `/usr/src/linux/` dans le répertoire `/boot/` :

```
cp System.map /boot
```

Ce fichier contient la liste de tous les symboles du nouveau noyau, il est utilisé par quelques utilitaires systèmes.

Si vous utilisez LILO, il vous faudra le réinstaller à nouveau pour qu'il prenne en compte le nouveau noyau. Cela se fait avec la même commande que celle utilisée précédemment :

```
lilo
```

Cette opération n'est en revanche pas nécessaire avec le GRUB.

Encore une fois, il faut redémarrer la machine avec la commande suivante :

```
reboot
```

et vérifier que le nouveau noyau fonctionne bien. S'il ne se charge pas correctement, c'est que les options de configuration choisies ne sont pas correctes. Il faut donc utiliser le noyau sauvegardé, vérifier ses choix et tout recommencer. Attention cependant, cette fois, il ne faut pas recommencer la sauvegarde du noyau puisque cette opération écraserait le bon noyau avec un noyau défectueux.

Si le nouveau noyau démarre correctement, il ne reste plus qu'à installer les modules.

VII-C-4 - Compilation et installation des modules

Si le système a redémarré correctement, on peut installer les modules. Il n'est pas nécessaire de prendre les mêmes précautions pour les modules que pour le noyau. Il suffit donc ici de lancer la commande suivante dans le répertoire `/usr/src/linux/` :

```
make modules_install
```

Au préalable, il est recommandé de décharger tous les modules présents en mémoire. Cela peut être réalisé à l'aide de la commande `modprobe` et de son option `-r`.

Les modules sont installés dans le répertoire `/lib/module/version/`, où `version` est le numéro de version du noyau courant. Il est possible que des modules d'autres versions du noyau existent dans leurs répertoires respectifs. Si vous n'en avez plus besoin, vous pouvez les effacer. Attention cependant si vous avez installé des modules additionnels non fournis avec le noyau dans ces répertoires, vous pourriez encore en avoir besoin.

Comme on l'a déjà vu, les modules sont utilisés par le chargeur de module du noyau, grâce à la commande `modprobe`. Cette commande a besoin de connaître les dépendances entre les modules afin de les charger dans le bon ordre. Il faut donc impérativement mettre à jour le fichier `/lib/modules/version/modules.dep` à chaque fois que l'on installe les modules, à l'aide de la commande suivante :

```
depmod -a
```

Note : La commande `depmod -a` est exécutée automatiquement lors de l'installation des modules du noyau. Toutefois, elle devra être exécutée manuellement si l'on installe des modules non fournis avec le noyau.

VIII - Configuration du matériel et des périphériques

Ce chapitre présente les concepts nécessaires à l'installation de cartes filles additionnelles et à la configuration des périphériques complémentaires sous Linux. La procédure d'installation des disques durs, graveurs, cartes son, cartes graphiques, cartes vidéo et cartes réseau sera donc décrite, ainsi que la manière de les configurer pour obtenir un fonctionnement correct sous Linux.

VIII-A - Généralités sur le support matériel sous Linux

Comme il l'a déjà été expliqué dans les chapitres précédents, Linux gère la plupart des périphériques comme des fichiers spéciaux. De plus, la plupart des gestionnaires de périphériques peuvent être chargés dynamiquement, sous la forme de modules du noyau. Cette section présentera donc les notions de base concernant les modules du noyau, les fichiers spéciaux de périphériques, ainsi que leurs rôles respectifs dans la configuration d'un nouveau matériel. La syntaxe utilisée pour fournir des options au noyau lors de son amorçage et les mécanismes de détection du matériel et de chargement automatique des gestionnaires de périphériques seront également présentés.

VIII-A-1 - Modules du noyau

Les modules du noyau sont des bibliothèques que l'on peut charger dans le noyau lorsque celui-ci a besoin d'une certaine fonctionnalité. Une fois chargés, les modules font partie intégrante du noyau et ajoutent leurs fonctions à celles existantes. Ces bibliothèques sont normalement stockées dans le répertoire `/lib/modules/version/`, où `version` est le numéro de version du noyau pour lequel ces modules ont été créés.

Beaucoup de fonctionnalités du noyau peuvent être configurées pour être utilisées sous la forme de modules. Dans le cas des pilotes de périphériques, les modules permettent de réaliser un noyau générique et de prendre en charge dynamiquement les différents périphériques de l'ordinateur. Certains pilotes ne fonctionnent d'ailleurs que sous la forme de modules, aussi faut-il savoir les manipuler. Les modules sont également fortement utilisés pour la configuration automatique des périphériques connectables à chaud.

VIII-A-1-a - Chargement et déchargement des modules

Les modules peuvent être chargés manuellement à l'aide des commandes `insmod` et `modprobe`. `modprobe` est un peu plus évoluée, car elle gère les dépendances entre les modules et est capable de charger les modules utilisés par le module demandé. Leur utilisation est des plus simple :

```
insmod module
```

ou :

```
modprobe module
```

où `module` est le nom du module à charger.

En réalité, la commande `modprobe` appelle la commande `insmod` pour chaque module qui doit être chargé, dans l'ordre des dépendances des modules. De cette manière, chaque module peut être chargé sans problèmes, car toutes les fonctionnalités qu'il utilise sont déjà présentes dans le noyau lors de son chargement. La commande `modprobe` va chercher les informations de dépendances dans le fichier `modules.dep`, situé dans le répertoire `/lib/module/version/`. Ce fichier utilise une syntaxe très simple, et spécifie pour chaque module les modules dont il dépend. Bien qu'il puisse parfaitement être écrit à la main, cela nécessiterait d'avoir une connaissance approfondie des modules du noyau. C'est donc pour cela que l'outil `depmod` a été écrit. Cet outil permet de générer le fichier `modules.dep` automatiquement, pourvu qu'on l'appelle avec l'option `-a` en ligne de commande :

```
depmod -a
```

Il faudra donc appeler cette commande après chaque installation ou suppression de modules dans le système.

La liste de modules qui ont été chargés peut être obtenue aisément avec la commande `lsmod` :

```
lsmod
```

Enfin, la commande `rmmod` permet de décharger un module, avec la ligne de commande suivante :

```
rmmod module
```

On préférera cependant l'utilisation de la commande `modprobe` avec l'option `-r` pour décharger les modules, car cette commande est capable de décharger récursivement les modules dont dépendait le module à décharger lorsqu'ils ne sont eux-mêmes plus utilisés. La syntaxe est alors la suivante :

```
modprobe -r module
```


Note : En réalité, il n'est généralement pas nécessaire de charger les modules du noyau manuellement pour les pilotes de périphériques. En effet, comme nous allons le voir dans les sections suivantes, Linux dispose de mécanismes permettant de réaliser le chargement des modules de pilotes de périphériques automatiquement, lorsque ces pilotes sont détectés. Toutefois, il est possible que certains périphériques ne soient pas pris en charge par ces mécanismes (notamment les vieux périphériques ISA non Plug and Play).

VIII-A-1-b - Options des modules

La plupart des modules peuvent prendre un certain nombre d'options lors de leur chargement. Ces options permettent de fixer certains paramètres, et dans le cas des pilotes de périphériques, de préciser la configuration matérielle utilisée.

Les options des modules peuvent être spécifiées en ligne de commande lors de l'appel de `insmod` ou de `modprobe`. Toutefois, il est possible d'enregistrer ces options de manière permanente à l'aide du fichier de configuration `/etc/modprobe.conf`. Lors du chargement d'un module `modprobe` consulte en effet ce fichier et passe les paramètres indiqués au module à charger.

Afin de faciliter la maintenance du système, `modprobe` consulte également les fichiers de configuration situés dans le répertoire `/etc/modprobe.d/`. Ainsi, les options spécifiques à chaque module peuvent être placées dans un fichier de configuration dédié dans ce répertoire.

Le format de ces fichiers de configuration est assez simple, les options de chaque module se définissant sur une ligne commençant par le mot clé `options`. Ce mot clé doit être suivi du nom du module pour lequel ces options sont définies, lui-même suivi des paramètres de chargement du module. La syntaxe est donc la suivante :

```
options module paramètres
```

La liste des paramètres que l'on peut passer à un module dépend bien évidemment du module. Vous pouvez obtenir la liste des options supportées par un module à l'aide de l'option `-p` de la commande `modinfo` :

```
modinfo -p module
```

Nous verrons par la suite des exemples de passage de paramètres pour les modules les plus courants.

Les fichiers de configuration de `modprobe` permettent également de définir la manière dont le chargement et le déchargement des modules doivent être faits. Pour cela, il permet de spécifier une ligne de commande complète pour ces opérations. Ces lignes de commandes sont introduites via les mots-clés `install` et `remove`. La syntaxe de ces mots-clés est la suivante :

```
install module commande  
remove module commande
```

où `module` est le nom du module à charger ou à décharger, et `commande` est la ligne de commande de l'opération à effectuer pour cela.

Par exemple, si le port parallèle est utilisé pour accéder à un périphérique nécessitant une opération d'initialisation quelconque, celle-ci peut être exécutée lors du chargement du module à l'aide du mot clé `install` :

```
install parport_pc modprobe parport_pc --ignore-install && initperiph
```

(en supposant que la commande `initperiph` permet d'initialiser le périphérique en question). Évidemment, ce type d'opération dépend du matériel connecté sur le port parallèle, mais le principe est là. Vous noterez que pour éviter

que la commande `modprobe` ne boucle sur elle-même dans la commande indiquée par la directive `install`, l'option `--ignore-install` lui a été fournie. Dans le cas d'une directive `remove`, il faudrait utiliser l'option `--ignore-remove`.

Dans la plupart des cas, la définition de ces commandes de chargement et de déchargement est facultative, car les modules utilisent généralement un jeu de paramètres implicites qui permettent de les utiliser même si le fichier de configuration `modprobe.conf` est absent. Vous pouvez visualiser ces options en renommant le fichier de configuration `modprobe.conf` et en tapant la commande suivante :

```
modprobe -c
```

Cette commande vous permettra également de recréer un nouveau fichier de configuration si d'aventure vous perdiez celui fourni avec votre distribution.

VIII-A-1-c - Options des périphériques intégrés au noyau

La plupart des fonctionnalités disponibles sous la forme de modules peuvent également être intégrées directement dans le noyau, lors de la compilation de celui-ci. Il est évident qu'il n'est alors pas possible d'utiliser les fichiers de configuration de `modprobe` pour fixer les valeurs des différentes options que ces fonctionnalités peuvent utiliser. En effet, ces paramètres doivent dans ce cas être définis dès le démarrage du système, et il est nécessaire de les fournir dès le chargement du noyau en mémoire.

Cela se fait, comme nous l'avons vu dans la [Section 3.4.7](#), en fournissant au noyau les paramètres en question sur sa ligne de commande. Nous avons déjà vu comment la quantité de mémoire de l'ordinateur pouvait être passée au noyau à l'aide de l'option `mem` si elle n'était pas détectée correctement. Il est possible de spécifier de la même manière de nombreuses autres options, en particulier pour activer ou désactiver certains gestionnaires de périphériques ou pour leur indiquer les paramètres du matériel.

Les options fournies au noyau sur sa ligne de commande par le gestionnaire d'amorçage sont lues par le noyau dès son démarrage, afin de fixer les paramètres qu'elles décrivent jusqu'au redémarrage suivant. Pour certaines fonctionnalités, ces paramètres peuvent toutefois être modifiées dynamiquement après le démarrage, via le système de fichiers virtuel `/proc/`.

Il existe un grand nombre d'options, et leur syntaxe varie en fonction du sous-système qui les utilise. Le fichier de documentation `kernel-parameters.txt` du répertoire `Documentation/` des sources du noyau vous donnera la liste complète de ces options. Nous verrons quelques-unes de ces options spécifiquement dans la suite du chapitre, lorsque nous décrirons les paramètres des gestionnaires de périphériques.

En revanche, les options des modules intégrés au noyau suivent toujours la même syntaxe. Cette syntaxe est la suivante :

```
module.paramètre=valeur
```

où `module` est le nom du module du noyau qui implémente la fonctionnalité lorsqu'elle est utilisée sous la forme de module, `paramètre` est le nom de l'option de ce module du noyau telle qu'elle est donnée par la commande `modinfo`, et `valeur` est sa valeur. Ainsi, si vous voulez intégrer une fonctionnalité d'un module dans le noyau, il suffit simplement d'ajouter les paramètres complémentaires dans la ligne de commande de celui-ci en fonction des options définies dans les fichiers de configuration de `modprobe`. Nous verrons la manière de procéder plus loin dans ce chapitre.

Note : Intégrer un pilote de périphérique directement dans le noyau peut être intéressant si ce périphérique est inamovible et que l'on ne veut pas avoir à prendre en charge les modules. Toutefois, ce n'est pas une bonne idée lorsqu'on installe ce périphérique et qu'on cherche à le configurer, ou si l'on désire se réserver la possibilité de faire une mise à jour de ce périphérique sans redémarrer l'ordinateur. En effet, il est souvent nécessaire de redémarrer pour prendre en compte de nouveaux paramètres lorsque le pilote de périphérique est intégré au noyau, alors que lorsqu'il est sous la forme d'un module, un simple déchargement et rechargement suffit.

Par contre, lorsque votre configuration sera finie, vous pourrez parfaitement vous passer des modules et supprimer les modules des pilotes de périphériques dont vous ne disposez pas. Vous gagnerez ainsi de la place disque et potentiellement plusieurs dizaines de secondes au démarrage de votre ordinateur si votre distribution recherche les nouveaux périphériques à chaque démarrage.

VIII-A-2 - Les fichiers spéciaux de périphériques

VIII-A-2-a - Généralités

La notion de fichier spécial de périphérique simplifie énormément l'utilisation du matériel par les programmes d'application, puisque la plupart des opérations sur un périphérique reviennent simplement à réaliser une écriture ou une lecture. Évidemment, l'écriture sur un fichier spécial de disque permet d'enregistrer les données sur ce disque, et la lecture permet de les récupérer. Mais cela ne s'arrête pas là ! Par exemple, la communication avec le modem se fait simplement en écrivant et en lisant les données sur le fichier spécial du port série sur lequel le modem est connecté. De même, jouer un fichier son revient simplement à l'écrire dans le fichier spécial qui gère la carte son. Il est même possible d'accéder à la mémoire vidéo par l'intermédiaire d'un fichier spécial de périphérique...

Bien entendu, certaines fonctionnalités avancées des périphériques ne peuvent pas être accédées simplement par une écriture ou une lecture dans un fichier spécial. Le système fournit donc aux applications d'autres moyens d'accéder à ces fonctionnalités, par l'intermédiaire d'appels systèmes spécifiques (pour ceux qui sont intéressés par la programmation système, cet appel système est réalisé par la fonction `ioctl`, dont le nom provient de l'abréviation de l'anglais « Input / Output ConTrol »). Évidemment, cette méthode n'est utilisée que par les programmes qui connaissent bien le fonctionnement du gestionnaire du périphérique, car ils doivent spécifier une requête que seul ce gestionnaire comprend en général. Quoi qu'il en soit, les requêtes de ce type utilisent elles aussi un descripteur de fichier spécial de périphérique, ce qui fait que tous les accès au matériel ont lieu par l'intermédiaire de ces fichiers.

Il existe deux principaux types de fichiers spéciaux de périphériques. Le premier type correspond aux périphériques de type bloc, dont les données ne peuvent être lues que par blocs (c'est le cas des disques durs, des lecteurs de CD et de disquettes en particulier). Le deuxième type correspond aux périphériques de type caractère, dont les données peuvent être lues caractère par caractère (cartes son, ports série, etc.).

En plus de son type, chaque fichier spécial de périphérique est caractérisé par deux codes permettant d'identifier le type et le modèle du périphérique auquel il donne accès. Ces deux codes portent le nom de code majeur et de code mineur. C'est par l'intermédiaire de ces codes que le noyau est capable de retrouver le gestionnaire de périphériques à utiliser pour satisfaire aux requêtes des clients qui accèdent à un fichier spécial de périphérique. Il y a donc, en général, une association unique entre ces codes et les gestionnaires de périphériques.

Les fichiers spéciaux de périphériques sont tous stockés dans le répertoire `/dev/`. Selon votre distribution, ce répertoire peut être géré de manière totalement dynamique en fonction du matériel effectivement installé sur votre machine, ou être prérempli statiquement avec les fichiers spéciaux de la plupart des périphériques rencontrés sur le marché (y compris donc pour des périphériques inexistant sur votre machine). Nous allons voir dans les sections suivantes comment ces fichiers sont créés et utilisés par le système en relation avec les pilotes de périphériques.

VIII-A-2-b - Le système de fichiers virtuels udev

VIII-A-2-b-i - Présentation

Depuis la version 2.6 du noyau, Linux dispose d'une fonctionnalité permettant aux applications de détecter l'apparition et la suppression des périphériques, que ceux-ci soient détectés au démarrage du système ou qu'ils soient branchés à chaud une fois l'ordinateur allumé.

Cette fonctionnalité est principalement utilisée par le sous-système `udev` (abréviation de « Userspace /dev »). Comme son nom l'indique, `udev` a pour principale fonction de prendre en charge la gestion du répertoire `/dev/`, mais en réalité il est capable de faire beaucoup mieux que cela.

En particulier, udev peut réaliser les opérations suivantes lorsque le noyau signale la présence d'un nouveau périphérique :

- chargement du module du pilote de périphérique si nécessaire ;
- si le périphérique requiert un firmware, chargement de celui-ci ;
- création du fichier spécial de périphérique et de ses alias nécessaires à l'utilisation du périphérique ;
- exécution des opérations d'initialisation complémentaires ou lancement des applications utilitaires associées au périphérique ;
- notification de la présence du périphérique à l'ensemble des autres programmes qui s'intéressent à la gestion du matériel (par exemple le gestionnaire de bureau).

udev peut également effectuer les opérations nécessaires à la suppression d'un périphérique, ce qui est utile pour les périphériques amovibles. Ainsi, la détection, la configuration et la notification dynamique de la présence des périphériques Plug and Play sont totalement automatisées.

Note : Le noyau, et donc udev, ne sont en réalité pas capables de détecter que les périphériques connectés au bus IDE, USB, PCMCIA, FireWire ou PCI. En effet, ces bus utilisent des technologies suffisamment récentes et permettent la détection, la configuration automatique et l'identification précise des périphériques. De ce fait, seuls les pilotes de périphériques de ce type peuvent être chargés automatiquement par udev. Les pilotes des périphériques plus anciens (périphériques ISA par exemple) ne peuvent être chargés automatiquement, car ces périphériques ne sont détectés que si leur pilote est déjà chargé dans le noyau. Aussi est-il nécessaire de précharger ces pilotes au démarrage de l'ordinateur, ou de les intégrer directement dans le noyau, pour qu'ils soient reconnus automatiquement par le système.

udev permet de résoudre plusieurs problèmes concernant les fichiers spéciaux de périphériques. Le plus évident pour l'utilisateur est que seuls les fichiers spéciaux des périphériques effectivement présents et pris en charge par un gestionnaire de périphérique apparaissent dans le répertoire `/dev/`.

Le répertoire `/dev/` n'a donc plus à contenir les fichiers spéciaux de périphériques pour tous les périphériques pris en charge par Linux. Il en est d'autant réduit, ce qui permet de le placer dans un système de fichiers virtuel (c'est-à-dire un système de fichiers géré par le noyau en mémoire). La plupart des distributions basées sur udev utilisent cette technique afin d'accélérer la phase de chargement du système et les accès aux fichiers spéciaux de périphériques par les applications, et d'éviter de consommer inutilement de l'espace disque pour stocker des fichiers spéciaux qui sont de toute manière recréés dynamiquement à chaque démarrage.

Un autre avantage d'udev est que les codes majeurs et mineurs des fichiers spéciaux de périphériques peuvent à présent être attribués dynamiquement par le noyau lorsque les gestionnaires de périphériques s'initialisent. Cela a plusieurs conséquences :

- il n'y a plus besoin d'avoir recours à une autorité centrale pour obtenir les valeurs de codes mineurs et majeurs, ceux-ci pouvant être déterminés dynamiquement par le noyau ;
- il n'y a plus de risque de conflit entre deux périphériques, le noyau s'assurant de l'unicité des codes utilisés ;
- la limitation du nombre de périphériques due au nombre limité de codes majeurs et mineurs n'existe plus (il suffit de consulter la liste du fichier `/usr/src/linux/Documentation/devices.txt` pour constater qu'il reste peu de codes libres pour les périphériques à venir).

Enfin, udev permet de simplifier considérablement la configuration et l'utilisation du système. Par exemple, il est possible de fixer de manière permanente le nom des fichiers spéciaux des périphériques amovibles, et les environnements utilisateurs graphiques peuvent réagir à l'apparition et à la suppression des périphériques afin de permettre leur utilisation par l'utilisateur.

VIII-A-2-b-ii - Principe de fonctionnements de udev

Le principe de fonctionnement de udev est le suivant. Lorsque le noyau détecte un changement dans la configuration matérielle, il signale ce changement via un canal de communication « NetLink » aux applications qui s'y intéressent.

udev utilise le démon udevd pour écouter sur ce canal de communication, et il met chacun de ces événements dans une file pour les traiter dans leur ordre d'apparition.

Les événements envoyés par le noyau contiennent toutes les informations relatives au périphérique pour lequel le noyau a généré un événement. Ces paramètres comprennent, entre autres, la nature de l'événement (apparition ou disparition d'un périphérique inconnu sur un bus, demande de chargement de firmware, notification de chargement ou de déchargement d'un pilote de périphérique, etc.). udevd utilise ces informations, ainsi que les informations sur le périphérique fournies par le noyau au travers du système de fichiers `/sys/`, pour effectuer les opérations de chargement des modules de pilotes de périphérique, de chargement des firmware, et de création des fichiers spéciaux de périphériques.

Note : Le démon udevd exécute les commandes udev selon l'ordre indiqué par le noyau via un numéro de séquence. Cette sérialisation est nécessaire parce que le noyau est capable de générer plusieurs événements simultanément pour les périphériques complexes, et qu'il faut s'assurer que les pilotes de périphériques parents sont bien initialisés avant de charger les pilotes des sous-périphériques. Ce cas peut se produire par exemple pour les cartes d'extension ou pour les hub USB.

Les anciennes versions de udev n'écoutaient pas directement les événements générés par le noyau. Le mécanisme utilisé était légèrement plus complexe et passait par un programme utilitaire nommé hotplug. Le noyau appelait ce programme à chaque fois qu'un événement se produisait au niveau de la configuration matérielle. Ce programme déterminait les modules de pilotes de périphériques à charger, effectuait le chargement de ces modules, et initialisait les périphériques avec leur firmware si nécessaire. Puis, il envoyait les événements au démon udevd pour que celui-ci mette à jour le répertoire `/dev/`. Ce mécanisme a été abandonné au profit d'une gestion totalement intégrée dans udevd en raison des mauvaises performances dues à l'exécution de l'utilitaire hotplug pour chaque événement matériel.

Le comportement que udev doit adopter dépend bien évidemment de chacun des périphériques, ou du moins de la classe du périphérique pour lequel l'événement se produit. Ce comportement est donc paramétrable à l'aide des fichiers de configuration situés dans le répertoire `/etc/udev/` et `/etc/udev/rules.d/`. Ce dernier répertoire contient en particulier les règles qui permettent de définir les fichiers spéciaux de périphériques qui doivent être créés, les alias permettant de les nommer de manière constante, et les permissions que l'on doit leur affecter.

Le répertoire `/etc/udev/` ne contient généralement que le fichier de configuration principal d'udev, à savoir le fichier `udev.conf`. Ce fichier permet d'indiquer le répertoire cible dans lequel les fichiers spéciaux de périphérique seront créés (paramètre `udev_root`, dont la valeur doit obligatoirement être `/dev/`), le répertoire contenant les fichiers de définition des actions prises par udev en fonction des différents événements (paramètre `udev_rules`, dont la valeur est normalement `/etc/udev/rules.d/`), et le niveau d'erreur à partir duquel les traces d'udev seront enregistrées dans le journal du système syslog (paramètre `udev_log`).

L'essentiel de la configuration réside toutefois dans les fichiers du répertoire `/etc/udev/rules.d/`. Ces fichiers expriment, sous la forme de règles permettant de sélectionner les différents événements provenant du noyau en fonction des informations de cet événement, les actions à entreprendre. Ce mécanisme est suffisamment générique pour permettre d'identifier de manière unique les périphériques et de réaliser les actions dont udev a la charge.

Les règles sont décrites à raison d'une ligne par règle dans les fichiers de configuration. L'action identifiée par chaque règle n'est exécutée que si l'événement généré par le noyau correspond aux critères de sélection de la règle. Certains mots clés permettent d'obtenir les informations sur l'événement produit par le noyau, et d'autres mots clés permettent de décrire les actions à effectuer lorsque les critères de sélection de l'événement sont satisfaits.

Par exemple, le mot clé `ACTION` identifie la nature de l'événement. Le mot clé `KERNEL` donne le nom que le noyau donne au périphérique qui a généré l'événement, et `DEVPATH` le chemin du périphérique dans le système de fichiers virtuel `/sys/`. Inversement, le mot clé `NAME` permet de paramétrer l'action de création du fichier spécial de périphérique en définissant le nom de ce fichier, et `OWNER`, `GROUP` et `MODE` permettent de définir respectivement l'utilisateur propriétaire, le groupe, et les droits d'accès à ce fichier. Ainsi, les simples règles suivantes :

```
KERNEL=="mem", NAME="%k", GROUP="kmem", MODE="0640"  
KERNEL=="kmem", NAME="%k", GROUP="kmem", MODE="0640"
```

permettent de créer les fichiers spéciaux de périphériques `/dev/mem` et `/dev/kmem` (qui donnent respectivement accès à la totalité de la mémoire physique de l'ordinateur et à la mémoire du noyau). Bien entendu, ces fichiers spéciaux ne doivent être accessibles que par l'administrateur, mais il est également autorisé que les utilisateurs du groupe `kmem` puissent y accéder en lecture.

Note : Notez bien que dans les définitions de règle, le signe `'=='` est utilisé pour indiquer une contrainte d'égalité, alors que le signe `'='` est utilisé pour réaliser une affectation, et constitue donc une partie de l'action de la règle.

D'autres mots clés sont disponibles. Ainsi, `SYMLINK` permet de définir des liens symboliques supplémentaires (par exemple `/dev/modem` ou `/dev/mouse`) référençant les fichiers spéciaux de périphériques réels. Le mot clé `PROGRAM` permet d'exécuter un programme avant que le fichier spécial de périphérique ne soit créé (généralement pour déterminer le nom du fichier à créer, justement), et le mot clé `RUN` permet d'exécuter une commande après que le fichier spécial de périphérique a été créé (généralement pour configurer le périphérique ou signaler sa présence à l'utilisateur). Vous trouverez de plus amples renseignements sur toutes ces règles dans la page de manuel de `udev`, ainsi que dans la documentation fournie avec `udev` pour la rédaction de règles personnalisées.

Les fichiers du répertoire `/etc/udev/rules.d/` doivent tous avoir l'extension `.rules`. De plus, ces fichiers sont interprétés dans l'ordre lexicographique de leur nom. Ainsi, il est d'usage de préfixer le nom du fichier par un numéro indiquant l'ordre de priorité des règles de ce fichier par rapport aux autres. Par exemple, les règles du fichier `50-udev.rules` (fichier de définition des règles principales de `udev`) sont toujours lues avant les règles du fichier `90-hal.rules` (fichier de définition des règles permettant d'envoyer les événements concernant le matériel au démon d'abstraction du matériel `hald`), car 50 est plus petit que 90 dans l'ordre lexicographique.

Comme vous pouvez le constater, les fichiers de configuration de `udev` permettent de paramétrer de manière très souple et très précise les actions à effectuer lorsqu'un périphérique apparaît ou disparaît. Nous verrons dans les sections suivantes comment les principales actions sont réalisées. La contrepartie de ce système est tout de même une certaine complexité dans la définition des règles des fichiers de configuration d'`udev`, mais vous n'aurez généralement pas à vous en préoccuper, car les distributions fournissent des fichiers de configuration adaptés à la plupart des usages.

VIII-A-2-b-iii - Identification et chargement des pilotes de périphérique

Généralement, les périphériques modernes disposent d'identifiants qui leur sont propres et qui peuvent être récupérés par des mécanismes standards des bus modernes. Ces identifiants sont généralement au nombre de deux, le premier étant un code spécifique au fabricant du périphérique, et le deuxième étant un identifiant de ce périphérique. Par exemple, vous pouvez visualiser les différents périphériques connectés au bus PCI de votre ordinateur avec la simple commande `lspci -nn`. La commande `lsusb` vous donnera des identifiants similaires pour les périphériques USB.

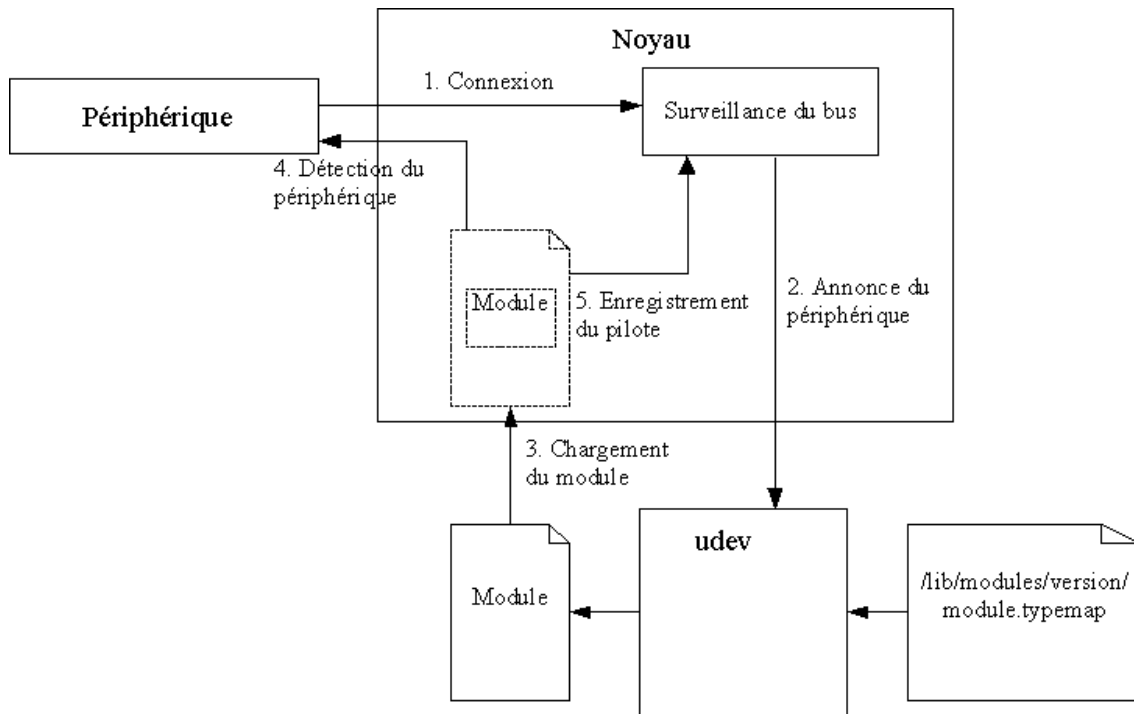
`udev` peut déterminer, à l'aide des identifiants du matériel, déterminer quels sont les pilotes de périphériques capables de gérer ce matériel. En effet, les fichiers sources de chaque module de pilote de périphérique contiennent une liste des identifiants des périphériques qu'ils prennent en charge. Lors de l'installation du module, la commande `depmod` extrait ces informations et les place dans un des fichiers situés dans le répertoire d'installation des modules du noyau (à savoir `/lib/module/version/`, où `version` est le numéro de version du noyau). `depmod` crée un fichier pour chaque type de périphérique. Ces fichiers ont tous un nom de la forme :

```
module.typemap
```

où `type` est le type de périphérique.

Par exemple, le fichier `module.usbmap` contient, pour chaque périphérique USB pris en charge par les modules installés, une ligne indiquant le nom du module et l'ensemble des identifiants du périphérique USB en question.

Ainsi, lorsque le noyau indique à `udev` l'apparition d'un nouveau périphérique, celui-ci consulte les fichiers d'association des modules aux périphériques pour déterminer quel module est capable de gérer ce périphérique. Ce module est chargé si nécessaire, et le gestionnaire de périphérique peut s'initialiser.



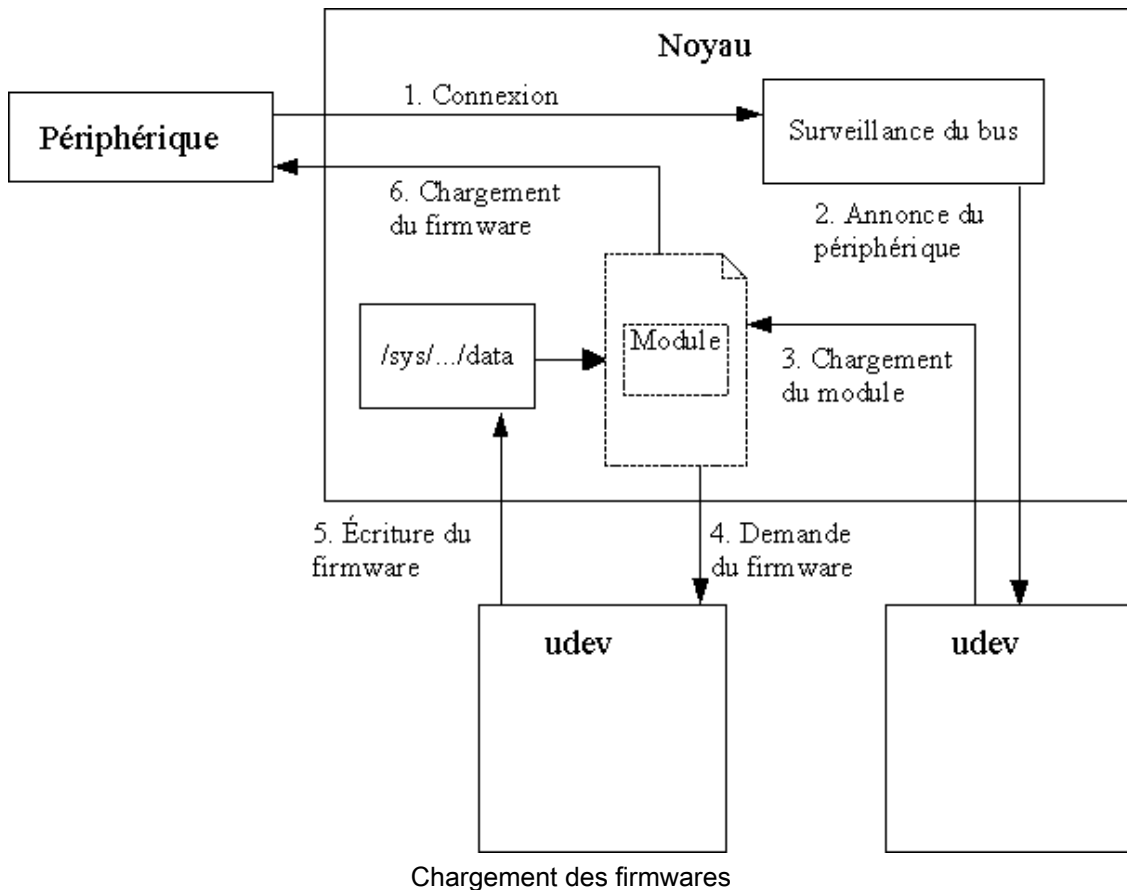
Chargement des modules de pilotes de périphérique

Note : La table des périphériques pris en charge par certains modules peut ne pas être complète ou à jour. C'est généralement le cas pour les nouveaux périphériques vendus par un fabricant qui utilisent une nouvelle révision d'une puce électronique que des anciens périphériques supportés par un pilote utilisaient déjà. Ainsi, lorsqu'un périphérique n'est pas reconnu immédiatement par les pilotes existants, vous pouvez tenter une recherche sur Internet avec les identifiants de périphériques tels qu'ils sont donnés par `lspci` ou `lsusb` et identifier le pilote nécessaire pour ce périphérique. S'il en existe un, vous pouvez essayer d'ajouter les identifiants de votre périphérique dans les sources du pilote, le recompiler et l'installer à nouveau. Si ce pilote vous permet d'utiliser effectivement ce périphérique de manière stable, vous pouvez alors demander au mainteneur du pilote d'ajouter les identifiants du périphérique dans la liste des périphériques supportés.

VIII-A-2-b-iv - Chargement des firmwares

Certains périphériques nécessitent un firmware (c'est-à-dire un micro logiciel pour les périphériques évolués) pour fonctionner correctement. Ce firmware peut être stocké de manière permanente dans le périphérique dans une mémoire non volatile, ou devoir être rechargé à chaque initialisation du périphérique. Dans ce cas, cette opération de chargement doit être faite par le pilote de périphérique.

Grâce à `udev`, il est possible de faire en sorte que cette opération se fasse automatiquement. Lorsqu'un pilote désire charger le firmware dans le périphérique, il appelle une fonction du noyau qui génère un événement `udev`. `udev` prend alors en charge les opérations et localise le firmware, puis le fournit au noyau. Le pilote de périphérique peut alors effectuer le chargement effectif du firmware dans le périphérique. Pour cela, `udev` copie le fichier de firmware demandé par le pilote de périphérique dans un fichier dédié à cet effet dans l'entrée du périphérique du système de fichiers virtuel `/sys/`. Bien entendu, le chemin permettant de trouver les informations du périphérique dans le système de fichiers virtuel `/sys/` est fourni aux scripts de configuration d'`udev` via le mot clé `DEVPATH`.

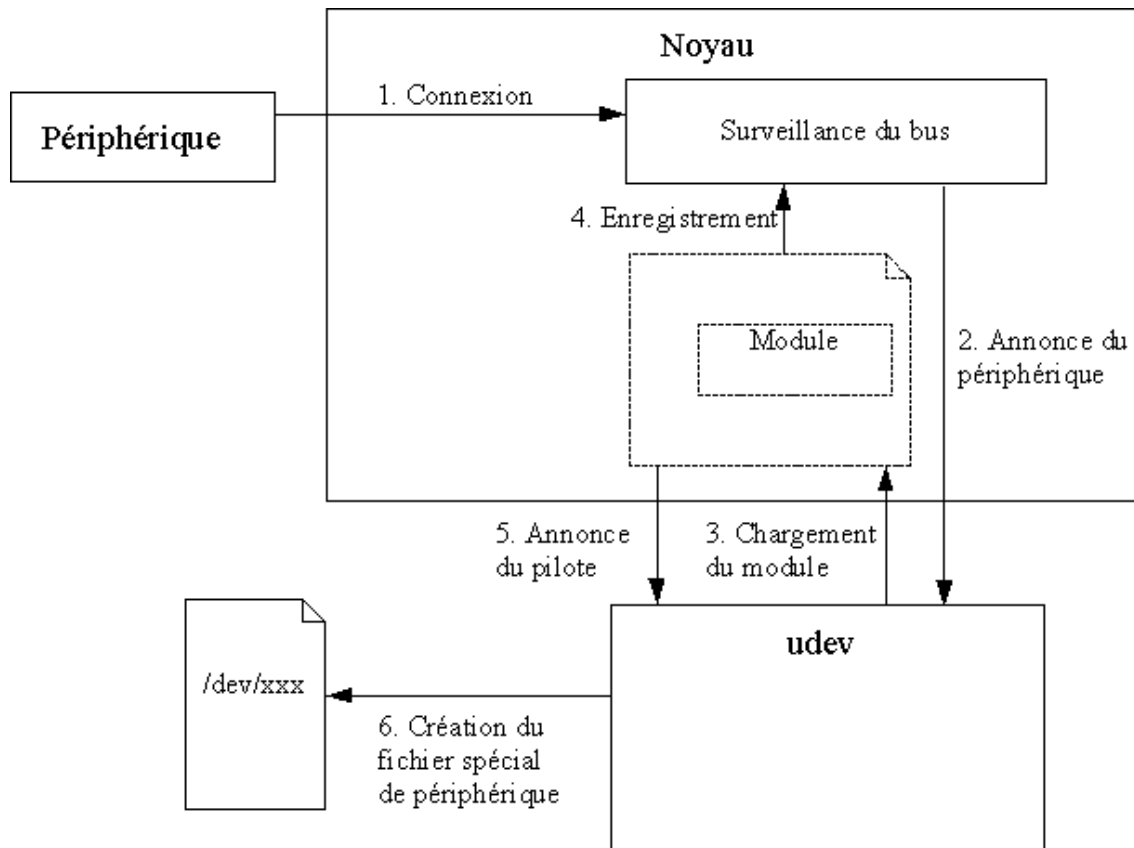


En général, les fichiers de firmware sont fournis avec les pilotes de périphériques ou sont disponibles sur le site Web indiqué dans leur documentation. Ils doivent être installés dans le répertoire `/lib/firmware/`. Il est également nécessaire d'activer l'option « Hotplug firmware loading support » du sous-menu « Generic Driver Options » du menu « Device Drivers » pour que cette fonctionnalité soit disponible.

VIII-A-2-b-v - Création des fichiers spéciaux de périphériques

À chaque périphérique détecté par un pilote de périphérique, le noyau génère un événement à destination d'udev. Ces événements indiquent le nom du périphérique tel qu'il est connu par le gestionnaire de périphérique, ainsi que les informations sur le périphérique dans le système de fichiers `/sys/`. udev utilise ces informations pour déterminer les règles qui indiquent quels sont les fichiers spéciaux de périphériques à utiliser.

Si aucune règle ne correspond, udev crée le fichier spécial de périphérique avec le nom utilisé par le pilote de périphérique pour l'identifier. Dans le cas contraire, c'est le nom fourni dans les fichiers de configuration qui est utilisé. De même, les fichiers sont créés par défaut au nom de l'utilisateur root, et seul cet utilisateur dispose des droits de lecture et d'écriture sur ces fichiers. Enfin, si des liens symboliques sont définis dans les règles appliquées, ceux-ci sont créés. Par exemple, il est classique d'avoir un lien symbolique cd-rom pointant vers le périphérique de type bloc gérant le lecteur de CD.



Création des fichiers spéciaux de périphériques

Les règles d'udev sont généralement fournies par les distributions et permettent de créer les fichiers spéciaux de périphériques de manière générique, ainsi que les liens symboliques standards. De même, bien que les interfaces réseau ne sont pas accédées par l'intermédiaire de fichiers spéciaux de périphériques, les règles udev sont également appliquées afin de les renommer sous un nom spécifique.

VIII-A-2-b-vi - Persistance des fichiers spéciaux de périphériques

L'un des principaux avantages d'udev est de permettre un nommage fixe des différents périphériques. Par exemple, si vous disposez de deux imprimantes, chaque imprimante se verra affecter un fichier spécial de périphérique dédié. Toutefois, il est possible qu'une de ces imprimantes ne soit pas disponible, ou pas allumée. Dans ce cas, un décalage des fichiers spéciaux de périphériques peut se produire, ce qui peut être gênant si une configuration spécifique doit être faite pour la deuxième imprimante. De même, vous pouvez disposer de plusieurs disques amovibles ou clefs USB, et il peut être utile que les fichiers spéciaux de ces périphériques aient un nom constant.

udev permet tout cela, en faisant en sorte que les règles des fichiers de configuration sélectionnent les périphériques sur une de leurs caractéristiques principales (par exemple, le nom de modèle du périphérique).

Pour fixer les idées et donner un exemple, supposons que l'on ait deux clefs mémoire USB de modèles différents et que l'on désire affecter de manière permanente des fichiers de périphériques à chacune d'elle. Il suffit pour cela simplement de trouver un paramètre unique dans les identifiants de ces périphériques afin de les distinguer.

La manière la plus simple pour obtenir la liste des paramètres d'un périphérique est d'utiliser la commande `udevinfo`. Pour obtenir la liste des attributs utilisables dans les règles de udev, il faut utiliser l'option `--attribute-walk`. Le périphérique dont les attributs doivent être récupérés peut être spécifié soit grâce au nom de son fichier spécial de périphérique utilisé par le noyau (avec l'option `--name`), ou avec son chemin dans le système de fichiers `/sys/`.

Note : La commande `udevmonitor` peut vous permettre de déterminer le chemin du périphérique dans le système de fichiers `/sys/`.

Par exemple, la commande suivante permet d'obtenir les informations sur un disque amovible connecté sur le port USB et apparaissant sous le nom `/dev/sdb` :

```
udevinfo --name=/dev/sdb --attribute-walk
```

Supposons que cette commande donne le résultat suivant :

```
Udevinfo starts with the device specified by the devpath and then
walks up the chain of parent devices. It prints for every device
found, all possible attributes in the udev rules key format.
A rule to match, can be composed by the attributes of the device
and the attributes from one single parent device.

looking at device '/block/sdb':
  KERNEL=="sdb"
  SUBSYSTEM=="block"
  DRIVER==" "
  ATTR{capability}=="12"
  ATTR{stat}=="      52      127      556      486      0      0      0      0      0
    426      486"
  ATTR{size}=="156301488"
  ATTR{removable}=="0"
  ATTR{range}=="16"
  ATTR{dev}=="8:16"

looking at parent device '/devices/pci0000:00/0000:00:02.2/usb1/1-2/1-2:1.0/host5/
target5:0:0/5:0:0:0':
  KERNELS=="5:0:0:0"
  SUBSYSTEMS=="scsi"
  DRIVERS=="sd"
  ATTRS{modalias}=="scsi:t-0x00"
  ATTRS{ioerr_cnt}=="0x0"
  ATTRS{iodone_cnt}=="0x3b"
  ATTRS{iorequest_cnt}=="0x3b"
  ATTRS{iocounterbits}=="32"
  ATTRS{timeout}=="30"
  ATTRS{state}=="running"
  ATTRS{rev}=="09.0"
  ATTRS{model}=="WD800VE-00HDT0  "
  ATTRS{vendor}=="WDC      "
  ATTRS{scsi_level}=="3"
  ATTRS{type}=="0"
  ATTRS{queue_type}=="none"
  ATTRS{queue_depth}=="1"
  ATTRS{device_blocked}=="0"
  ATTRS{max_sectors}=="240"

&#8230;
```

Comme on peut le voir, les informations du périphérique `/dev/sdb` sont d'abord affichées. Viennent ensuite les informations des périphériques parents, donc en particulier les informations sur le disque physique (ici, un Western Digital WD800VE-00HDT0), et ainsi de suite (seuls les deux premiers blocs d'information ont été recopiés dans cet exemple, les suivants identifiant l'adaptateur USB et les composants du bus USB eux-mêmes).

Ainsi, les informations spécifiques au matériel connecté que l'on doit utiliser pour définir un fichier spécial de périphérique unique sont données dans le deuxième groupe d'information (relative au disque physique). Il est donc possible de définir la règle udev suivante pour créer un fichier spécial de périphérique `/dev/hd_externe` dédié à ce disque :

```
SUBSYSTEMS=="scsi", ATTRS{model}=="WD800VE-00HDT0", NAME="hd_externe"
```

Notez cependant que l'on ne peut pas, dans une même règle udev mélanger des critères de sélection utilisant des informations des différents groupes d'informations renvoyés par `udevinfo`, parce que chaque groupe identifie un périphérique logique et un seul dans le noyau.

Une fois vos règles définies, vous pouvez les tester avec la commande `udevtest`. Cette commande prend en paramètre le nom du périphérique de plus haut niveau renvoyé par `udevinfo`. Dans notre exemple, il s'agit de `/block/sdb` :

```
udevtest /sys/block/sdb
```

Cette commande affichera l'ensemble des opérations que `udev` effectuerait si ce périphérique était connecté. Attention toutefois, les informations affichées proviennent d'une simulation, et peuvent être incorrectes si des règles `udev` utilisent des commandes externes, puisqu'en simulation ces commandes ne sont pas exécutées.

En pratique, les distributions font en sorte que les règles de nommage des périphériques soient écrites ou mises à jour dynamiquement lors de la première détection du périphérique. Elles s'appuient pour cela sur des programmes utilitaires, généralement situés dans le répertoire `/lib/udev/`. Par exemple, le script `/lib/udev/cdrom-symlinks.sh` a pour but de créer le fichier de configuration `/etc/udev/rules.d/75-optical-devices.rules`, qui contient les définitions des liens symboliques sur les lecteurs de disque optique (CDROM et graveurs de CD par exemple).

Note : Les scripts de génération de règles peuvent ne pas nommer les périphériques comme vous le désirez (par exemple, le fichier spécial `/dev/cdrom` peut référencer votre graveur de CD, alors que le fichier spécial `/dev/cdwriter` un simple lecteur de CD. Dans ce cas, vous pourrez modifier directement le fichier de configuration `75-optical-devices.rules` pour remettre les choses dans l'ordre. Si vous vous trompez, supprimez simplement ce fichier, il sera recréé automatiquement au prochain redémarrage d'`udev`.

VIII-A-2-b-vii - Notification des applications utilisateurs

`udev` est également livré avec un fichier de configuration contenant une règle permettant de notifier les applications de haut niveau des apparitions et disparition des périphériques.

Le mécanisme utilisé se base ici sur le sous-système HAL (« Hardware Abstraction Layer Daemon »). Ce sous-système a pour principal but de fournir une interface unifiée pour que l'ensemble des applications de haut niveau (par exemple les gestionnaires de bureau KDE et Gnome) puissent obtenir des informations sur les périphériques et être notifiées de toutes modifications de la configuration matérielle.

Toutes ces opérations sont implémentées au niveau du démon `hald`. Ce démon reçoit les informations concernant les périphériques directement de `udev`, grâce à la règle mentionnée ci-dessus. Grâce à ces informations, il maintient une base de données des périphériques présents et de leurs principales caractéristiques.

De plus, `hald` fournit une interface de programmation standard pour toutes les applications utilisateurs. Cette interface se base sur un protocole de communication standard, le protocole `D-BUS`. Ce protocole a été développé de manière commune avec les deux principaux environnements de bureau de Linux, à savoir KDE et Gnome. Ainsi, les applications de ces environnements sont capables de gérer les modifications intervenant sur le matériel. Par exemple, le fait de brancher une clef USB ou un disque externe permet de demander à l'utilisateur s'il veut le monter ou non.

Note : Le protocole de communication `DBus` a été conçu de manière générique et n'a pas pour vocation de n'être utilisé que par HAL. En réalité, c'est tout un système de communication inter processus de haut niveau et d'utilisation simplifiée (par rapport aux systèmes de composants lourds tels que CORBA par exemple).

`DBus` permet les communications entre les applications utilisateurs d'un même utilisateur, ainsi que les communications transverses entre les applications système. En pratique, ces communications sont gérées par un démon, dont une instance est lancée pour le système, et potentiellement une instance supplémentaire est exécutée pour chaque utilisateur. Les mécanismes de `DBus` et de HAL ne sont d'intérêt que pour les programmeurs et ne seront donc pas détaillés plus dans ce document.

VIII-A-2-b-viii - Initialisation du système

Pour terminer cette description des mécanismes d'udev, signalons que le démon udevd ne peut évidemment pas écouter les événements du noyau avant qu'il ne soit lancé. Par conséquent, les périphériques détectés par le noyau avant le démarrage du démon udevd ne sont pas détectés automatiquement.

Ce problème est géré au niveau des scripts de démarrage des distributions, en rejouant les événements matériels des périphériques déjà détectés par le noyau. Cela se fait en appelant la commande `udevtrigger`. Cette opération est totalement prise en charge par les distributions, et vous n'avez normalement pas à utiliser cette commande.

VIII-A-2-c - Gestion statique des fichiers spéciaux de périphériques

La gestion des périphériques et du répertoire `/dev/` par udev est utilisée par la plupart des distributions récentes. Toutefois, les anciennes distributions sont encore susceptibles d'utiliser un répertoire `/dev/` complètement statique, c'est-à-dire rempli à l'avance avec les fichiers spéciaux des périphériques les plus courants.

Normalement, votre distribution fournit un répertoire `/dev/` relativement complet. Vous n'avez donc pas à toucher au répertoire `/dev/`. Cependant, si vous devez créer un fichier spécial de périphérique vous-même, par exemple pour un gestionnaire de périphérique exotique non pris en charge par votre distribution ou non reconnu par le système, vous devrez utiliser la commande `mknod`. Sa syntaxe est relativement simple :

```
mknod fichier type majeur mineur
```

où `fichier` est le nom du fichier spécial de périphérique à créer, `type` est une lettre indiquant le type du fichier spécial ('c' pour les fichiers de type caractère et 'b' pour les fichiers de type bloc), et `majeur` et `mineur` sont les codes majeur et mineur du périphérique. Je vous invite à consulter la page de manuel de la commande `mknod` pour plus d'informations.

Note : La liste des codes majeurs et mineurs déjà affectés à des périphériques est donnée dans le fichier `/usr/src/linux/Documentation/devices.txt`. Comme vous pouvez le constater, l'utilisation d'un répertoire `/dev/` statique impose donc de disposer d'une autorité centrale régissant l'affectation des codes majeurs et mineurs aux gestionnaires de périphériques. De plus, la plupart des codes sont déjà affectés, ce qui est une des raisons pour laquelle le système udev a été développé.

Lorsqu'une application tente d'accéder à un fichier spécial de périphérique, trois cas peuvent se présenter :

- soit un pilote de périphérique capable de gérer le périphérique est déjà chargé ;
- soit il n'existe aucun pilote capable de gérer le périphérique, mais le noyau peut en trouver un ;
- soit le noyau est incapable de trouver un pilote pour ce périphérique.

Dans les deux premiers cas, la requête de l'application est satisfaite, éventuellement après le chargement automatique du pilote de périphérique sous la forme de module. Dans le troisième cas, évidemment, la requête est refusée.

Ainsi, lors de l'ouverture d'un fichier spécial de périphérique par une application, le noyau vérifie s'il dispose d'un pilote de périphérique capable de gérer ce périphérique, en s'appuyant sur les codes majeur et mineur du fichier spécial de périphérique. Si ce n'est pas le cas, il effectue un appel à `modprobe` pour charger le module contenant ce pilote de périphérique.

Le nom du module dont le noyau demande le chargement à `modprobe` dépend du pilote et ne correspond pas forcément à un nom de module existant. Par exemple, le module de gestion des ports parallèles se nomme, sur les ordinateurs de type PC, `parport_pc`. Toutefois, lorsque le noyau désire accéder au port parallèle, il n'utilise pas ce nom, mais plutôt un nom générique, qui est le même pour toutes les architectures matérielles supportées par Linux. Ce nom est `parport_lowlevel`. Ainsi, le noyau utilise la commande suivante pour charger le module de prise en charge du port parallèle :


```
modprobe -k parport_lowlevel
```

L'option `-k` permet d'indiquer à `modprobe` qu'il est appelé par le noyau, donc dans le contexte de chargement automatique des modules.

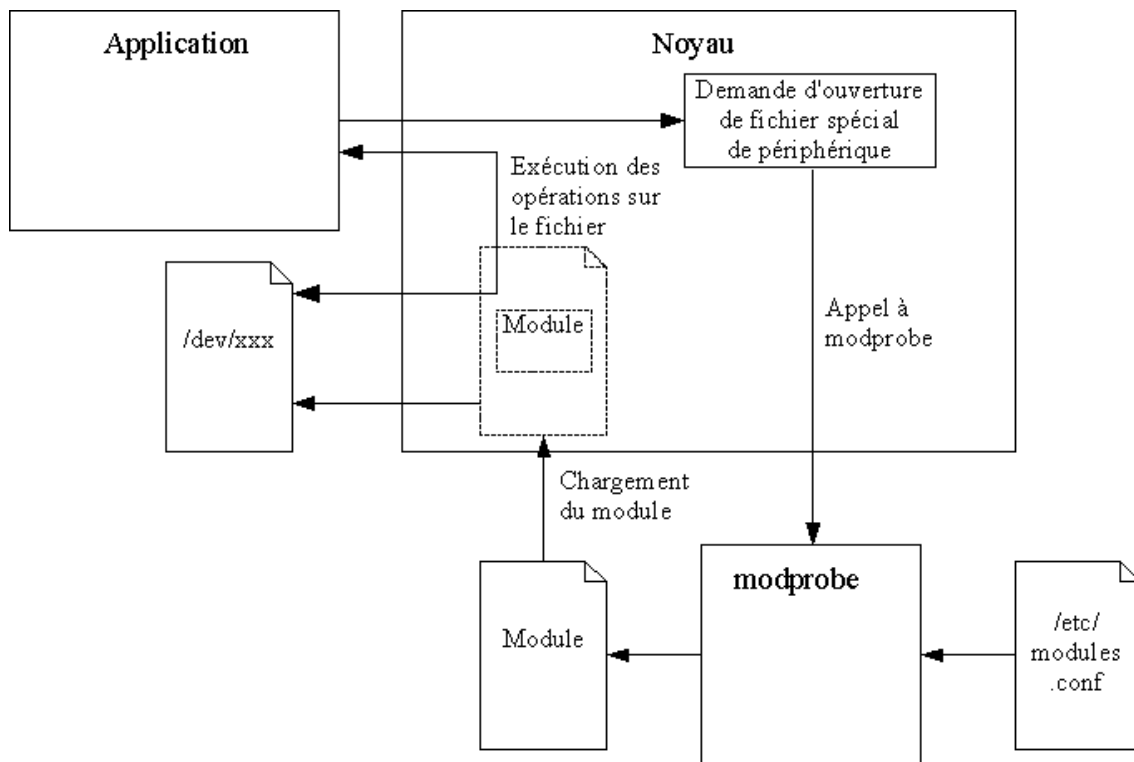
`modprobe` doit donc faire la correspondance entre le nom de module demandé par le noyau et le nom d'un module réel. C'est donc dans les fichiers de configuration de `modprobe` que cette correspondance est définie. Pour cela, un certain nombre d'alias peuvent être définis pour identifier les modules réels. Chaque alias est introduit par le mot clé `alias`, dont la syntaxe est donnée ci-dessous :

```
alias nom module
```

où `nom` est le nom de l'alias, et `module` est le nom du module réel. Lorsque la commande `modprobe` est appelée avec le nom d'un alias en paramètre, elle commence par rechercher le nom du module réel dans ses fichiers de configuration, puis elle le charge en mémoire.

Par exemple, pour charger le module `parport_pc` automatiquement lorsque le noyau a besoin d'accéder au port parallèle, on pourra ajouter la définition d'alias suivante dans le fichier `modprobe.conf` :

```
alias parport_lowlevel parport_pc
```



Chargement automatique des modules du noyau

Vous pouvez constater que le mécanisme d'alias permet de rendre le noyau indépendant des modules utilisés, puisque l'association entre le nom du module utilisé par le noyau et le module réel et ses paramètres est maintenue dans des fichiers de configuration. L'inconvénient de cette méthode est en revanche qu'il faut connaître les noms de modules utilisés par le noyau pour chaque pilote et pour leurs fonctionnalités. Ces noms sont assez variables et dépendent de la fonctionnalité demandée. En général, ce nom apparaît dans les messages de traces du noyau lorsqu'il ne parvient pas à localiser un module lors de son chargement automatique. Ces messages de traces peuvent être affichés à l'aide de la commande `dmesg`. Il est donc possible de déterminer la liste des modules que le noyau a tenté de charger relativement facilement. Toutefois, cela n'indique pas forcément la fonctionnalité implémentée par le module en question, et encore moins l'application qui a tenté d'utiliser cette fonctionnalité.

Il est heureusement beaucoup plus facile de savoir quelle est la fonctionnalité demandée lorsque celle-ci a trait à un périphérique de l'ordinateur. En effet, le nom de module utilisé par le noyau pour chaque gestionnaire de périphérique est construit à partir du type de fichier spécial de périphérique et de ses codes majeur et mineur. Ainsi, le noyau utilise un nom de module de la forme « char-major-xxxx » pour les périphériques de type caractère, et un nom de la forme « block-major-xxxx » pour les périphériques de type bloc. Les caractères xxxx identifient le code majeur de ce périphérique, plus rarement son code mineur. Il est donc facile de déduire le gestionnaire de périphérique nécessaire à la gestion de ce fichier spécial de périphérique, et de définir l'alias permettant de faire la correspondance entre le nom du module utilisé par le noyau et le module réel à charger.

Par exemple, pour les cartes son, le nom de module char-major-14 est utilisé pour demander le chargement du module de gestion du son, parce que les cartes son utilisent toutes le code majeur 14. Il faut donc définir un alias sur ce nom de module vers le module du gestionnaire de cartes son. Comme on le verra dans la section traitant de la configuration des cartes son, ce module pourra lui-même demander le chargement de modules complémentaires, en fonction de la carte son réellement utilisée et des fonctionnalités demandées. Le numéro de code mineur sera indiqué lors de ces opérations.

Si vous devez modifier les fichiers de configuration de modprobe, sachez qu'il s'agit d'une opération délicate, car elle nécessite de bien connaître les mécanismes de nommage utilisés par le noyau, les noms des modules réels et leurs paramètres de configuration. En général, vous trouverez les informations sur les entrées à ajouter ou à modifier dans le fichier d'aide du module correspondant, que vous pourrez trouver dans les sources du noyau (dans le répertoire `/usr/src/linux/Documentation/`). Quoi qu'il en soit, la modification des fichiers de configuration de modprobe peut générer de nouvelles dépendances entre les modules. Il est donc nécessaire d'exécuter la commande `depmod -a` après chaque modification de ce fichier.

VIII-B - Configuration des périphériques de masse

Les périphériques de masse sont les périphériques permettant de manipuler de grandes quantités de données. Il s'agit généralement des périphériques connectés sur bus SCSI, des disques durs et lecteurs de CD/DVD, des graveurs et des scanners.

VIII-B-1 - Configuration des périphériques SCSI

Les périphériques de qualité sont généralement connectés sur bus SCSI, en raison des capacités de celui-ci en termes de débit et de performances. En particulier, les disques durs SCSI restent privilégiés sur les systèmes haut de gamme comme les serveurs par exemple, parce que le bus SCSI permet d'utiliser plusieurs périphériques simultanément.

Sous Linux, les périphériques SCSI sont accédés par l'intermédiaire des fichiers spéciaux de périphériques `/dev/sdx` pour les disques durs ou `/dev/scdn` pour les lecteurs ou les graveurs de CD-ROM, où x est la lettre définissant le disque dur et n est le numéro du lecteur ou du graveur de CD-ROM. Les autres périphériques SCSI sont accédés par l'intermédiaire des fichiers spéciaux de périphériques SCSI génériques, à savoir les fichiers `/dev/sgn`. Notez que certains périphériques pourront être utilisés par l'un ou l'autre de ces fichiers spéciaux, selon la fonctionnalité à laquelle on cherche à accéder. Par exemple, les programmes de gravage utilisent les fichiers spéciaux de périphériques `/dev/sgn` pendant le gravage, alors que la simple lecture d'un CD avec un graveur se fait via le fichier spécial de périphérique `/dev/scdn` correspondant.

La prise en charge des périphériques SCSI se fait au niveau du noyau pour les fonctionnalités de base. En réalité, la configuration des périphériques SCSI se restreint à celle de la carte SCSI utilisée. Les fonctionnalités de plus haut niveau sont reléguées au niveau applicatif. Par exemple, la prise en charge des différents graveurs SCSI du marché est réalisée par les logiciels de gravage eux-mêmes, de même que la prise en charge des différents scanners l'est par les logiciels d'acquisition d'image.

La gestion des contrôleurs SCSI au niveau du noyau se fait par l'intermédiaire des options du menu « SCSI support » du programme de configuration du noyau. Outre l'option « SCSI support » que l'on activera bien évidemment, ce menu comprend les options suivantes, que l'on choisira en fonction du type des périphériques SCSI à utiliser :

- « SCSI disk support », pour les disques durs SCSI, ainsi que pour certains périphériques de masse qui apparaissent comme des disques durs SCSI (c'est en particulier le cas pour les clefs USB et un grand nombre d'appareils photo numériques) ;
- « SCSI tape support », pour les lecteurs de cartouches SCSI. Les possesseurs de lecteurs de cartouches OnStream SC-x0 devront plutôt utiliser l'option « SCSI OnStream SC-x0 tape support », car le gestionnaire de périphériques standard n'est pas capable de les prendre en charge ;
- « SCSI CD-ROM support », pour les lecteurs de CD-ROM SCSI et pour les graveurs de CD-ROM IDE, si l'on décide d'utiliser l'émulation SCSI avec certains logiciels de gravage (voir plus loin pour plus de détails à ce sujet) ;
- « SCSI generic support », pour les autres périphériques dont l'utilisation requiert l'utilisation d'un programme applicatif spécifique. Il s'agit ici des graveurs de CD, des scanners et autres périphériques spéciaux.

Certains périphériques SCSI disposent de plusieurs numéros d'unités logiques sur leur bus. Il peut donc être nécessaire, pour ces périphériques, d'activer l'option « Probe all LUNs on each SCSI device » afin de pouvoir les utiliser correctement. Les autres options ne sont pas essentielles, consultez leur aide pour plus de détails à leur sujet.

En plus des options générales de la prise en charge du SCSI, il est impératif de sélectionner le gestionnaire de périphériques capable de piloter votre contrôleur SCSI. Cela peut être fait dans le sous-menu « SCSI low-level drivers ». Vous devrez y activer la prise en charge pour votre matériel, aucune règle ne peut donc être définie à ce niveau. Consultez la documentation de votre matériel pour déterminer le gestionnaire de périphériques adéquat à utiliser.

Une fois la configuration du noyau effectuée, vous n'avez plus qu'à le compiler et à l'installer pour pouvoir utiliser vos périphériques. La manière de procéder a été décrite en détail dans la [Section 7.3](#).

VIII-B-2 - Configuration des disques durs IDE

Les disques durs IDE ont longtemps souffert d'un gros défaut par rapport aux disques SCSI : celui de ne pas pouvoir effectuer des opérations sans l'intervention du processeur. En effet, l'envoi des données aux disques se faisait classiquement par de nombreuses écritures dans les ports d'entrée/sortie des contrôleurs de disques, obligeant ainsi le système à consacrer une part importante des ressources processeur uniquement pour les accès disque. Ce défaut a été corrigé avec la montée en puissance du mode DMA (abréviation de l'anglais « Direct Memory Access »). Le mode DMA des périphériques IDE existe depuis les tous débuts de l'architecture PC, mais les premiers contrôleurs DMA n'étaient pas capables de suivre la cadence des disques durs, même des plus anciens. Il n'a donc malheureusement pas été utilisé pendant longtemps, jusqu'à ce qu'une nouvelle technologie soit mise au point : l'Ultra DMA. La plupart des jeux de composants des cartes mères modernes sont à présent capables de gérer l'Ultra DMA, dont les fréquences de bus les plus courantes sont 33, 66 et 100MHz.

Note : Même si les interfaces IDE ont à présent réduit l'écart avec les interfaces SCSI au niveau du taux de transfert des données, elles ont toujours un train de retard par rapport à elles. En effet, les périphériques SCSI sont d'une part capables de communiquer entre eux sur leur bus, déchargeant ainsi totalement le processeur de ce travail de transfert des données, et d'autre part capables de traiter plusieurs requêtes d'affilée sans l'intervention du processeur. Ces fonctionnalités déchargent le bus système, mais sont également extrêmement utiles pour les systèmes d'exploitation multitâches. Ceux-ci peuvent en effet envoyer plusieurs commandes aux périphériques et effectuer d'autres traitements pendant qu'elles s'exécutent en asynchrone. Les disques SCSI ont donc un taux d'occupation du processeur et du bus système nettement moindre, et permettent aux systèmes tels que Linux d'atteindre un degré de réactivité inégalé avec les disques IDE classiques. Cela dit, les périphériques IDE coûtent nettement moins cher que les périphériques SCSI, et c'est sans doute là la clef de leur succès.

En général, Linux configure les contrôleurs des disques durs de manière quasi optimale, et il n'est généralement pas nécessaire de modifier leur paramétrage. Cependant, les contrôleurs IDE peuvent être configurés ou optimisés manuellement grâce à l'utilitaire `hdparm`. Comme son nom l'indique, cet utilitaire permet de modifier les paramètres des disques durs et des contrôleurs IDE. Il permet également de tester les performances de votre configuration, et peut donc servir d'outil de diagnostic très utile. La mesure du débit de votre sous-système disque (sans les mécanismes de cache du système) peut être en effet réalisée facilement avec la commande suivante :

```
hdparm -t périphérique
```

où périphérique est le fichier spécial de périphérique de votre disque dur.

Note : Si vous obtenez des valeurs inférieures ou égales à 6 Mo/s, vous avez réellement un problème de configuration. Les premiers disques UltraDMA 33 peuvent atteindre facilement 8 à 10 Mo/s, les disques UltraDMA 66 atteignent facilement 17 à 22 Mo/s, et les disques les plus récents en UltraDMA 100 peuvent aller encore bien au-delà.

Faites bien attention à ne pas utiliser l'option -T à la place de l'option -t. En effet, vous mesureriez le débit dans le sous-système disque complet de Linux, avec ses mécanismes de cache. Vous obtiendriez alors des taux de transfert nettement plus grands (dix fois plus au moins), qui ne représenteraient pas le taux de transfert réel de votre interface IDE.

L'utilitaire `hdparm` permet également de fixer un certain nombre de paramètres des disques durs. Par exemple, l'activation ou la désactivation de l'UltraDMA se fait simplement avec l'option `-d`. Cette option prend en paramètre un entier pouvant valoir 1 ou 0, selon que l'UltraDMA doit être activé ou non. Cette option peut être complétée d'autres options d'optimisation pour les disques modernes, pour lesquels on peut également préciser le mode de transfert à utiliser. Ce mode peut être indiqué à l'aide de l'option `-X`, qui prend en paramètre un nombre dont la valeur est le numéro du mode plus une constante identifiant le type de transfert utilisé. Les transferts de type PIO (c'est-à-dire le mode dans lequel le processeur effectue lui-même les transferts par l'intermédiaire des ports d'entrée / sortie) utilisent la constante 8, les transferts de type DMA simple utilisent la constante 32, et les transferts en UltraDMA utilisent la constante 64. Ainsi, la sélection du mode UltraDMA mode 2 pour le disque maître du premier contrôleur IDE se fait avec la commande suivante :

```
hdparm -d1 -X66 /dev/hda
```

La valeur 66 utilisée ici signifie donc que les transferts se font en UltraDMA (constante 64), mode 2 (valeur 2).

Note : Pour que l'UltraDMA soit utilisable, il faut bien entendu que les pilotes IDE de votre noyau Linux le gèrent. Généralement, les noyaux des distributions en sont capables, mais si d'aventure ce n'était pas le cas, vous devriez recompiler votre noyau. Les options à activer dans le programme de configuration du noyau sont les options suivantes du menu « ATA/IDE/MFM/RLL support » :

- « Generic PCI IDE chipset support » du menu « IDE, ATA and ATAPI Block devices » ;
- « Generic PCI bus-master DMA support » ;
- « Use PCI DMA by default when available ».

Vous devrez également activer le support pour les jeux de composants utilisés par votre carte mère.

L'utilitaire `hdparm` permet également d'activer et de désactiver le mode 32 bits de ces contrôleurs à l'aide de l'option `-c`. Comme pour l'option `-d`, l'option `-c` prend en paramètre un indicateur pouvant valoir 1 ou 0 et permettant d'activer ou de désactiver le mode 32 bits du contrôleur. Notez que certains contrôleurs ne supportent pas correctement le mode de fonctionnement 32 bits et peuvent perdre des données si vous l'activez.

Note : Prenez garde lorsque vous utilisez la commande `hdparm`. Si vous spécifiez des paramètres incorrects ou non pris en charge par votre matériel, vous pouvez fort bien bloquer complètement vos contrôleurs IDE, et planter ainsi le système en un temps très court. Si d'aventure cela se produisait, il faudrait attendre un peu de temps, jusqu'à ce que le système s'aperçoive de la mauvaise configuration des contrôleurs et les réinitialise avec leurs paramètres initiaux. Vous pouvez détecter ce genre de réinitialisation dans les messages du noyau, que vous pouvez à tout moment afficher avec la commande `dmesg`. Sachez toutefois que le risque de corruption du système de fichiers en cas de mauvaise configuration reste bien réel, et qu'il vaut mieux un système légèrement plus lent qu'un système qui détruit vos données en un temps record.

Lorsque vous aurez trouvé les paramètres optimaux de vos contrôleurs de disque, vous pourrez demander au système de conserver ces paramètres par défaut en ajoutant l'option `-k1` dans la ligne de commande de `hdparm`. Par exemple, pour configurer de manière permanente le premier disque IDE en UltraDMA mode 2 et en accès 32 bits, il faut utiliser la commande suivante :

```
hdparm -c1 -d1 -X66 -k1 /dev/hda
```

Note : Prenez garde toutefois au fait que les réinitialisations du contrôleur en cas d'erreur reprendront systématiquement les mêmes paramètres, ce qui peut réellement bloquer complètement votre sous-système disque (et planter irrémédiablement votre système). Dans ce cas, il ne vous restera plus que le redémarrage brutal, avec les risques de pertes de données qui en découlent.

La configuration des disques durs et des contrôleurs IDE pourra également être ajoutée dans le script de démarrage de votre système. Ce script est généralement placé dans le répertoire `/etc/rc.d/` ou dans le répertoire `/sbin/init.d/` selon votre distribution. Bien entendu, il faut réellement être sûr de la commande de configuration utilisée, car elle sera exécutée systématiquement à chaque démarrage.

VIII-B-3 - Installation d'un graveur de CD/DVD

Linux dispose de tous les logiciels permettant de graver des CD ou des DVD, et ce avec la plupart des graveurs disponibles sur le marché. Il permet de copier des CD et de créer des images disques. Cependant, il n'est pas encore possible d'utiliser le gravage par paquet, qui permettrait d'utiliser les graveurs de CD comme des périphériques de sauvegarde amovibles. Cette fonctionnalité est en effet encore à l'étude, elle sera sans doute disponible sous peu.

VIII-B-3-a - Notions de base sur le gravage sous Linux

Originellement, tous les graveurs de CD/DVD utilisaient l'interface SCSI. Ce n'est que plus tard que les graveurs sur port parallèle et les graveurs IDE ATAPI sont apparus. Actuellement, les graveurs externes connectés sur port parallèle sont de moins en moins vendus. Les graveurs IDE ont fait une belle percée et sont les plus courants actuellement. Les logiciels de gravage doivent donc toutefois être capables d'utiliser n'importe quel type de graveur, ce qui n'est pas simple. Heureusement, le protocole de communication ATAPI n'est rien d'autre qu'un mode de transport de commandes SCSI pour les périphériques IDE, ce qui fait que les logiciels de gravage peuvent utiliser les deux types de périphériques de manière relativement générique.

Les anciennes versions des logiciels de gravage ne permettaient d'utiliser que l'interface SCSI pour graver des CD. Il n'était donc pas possible d'utiliser les graveurs IDE de manière native. Une couche d'émulation SCSI a donc été développée dans Linux afin de simuler une interface SCSI pour les périphériques IDE, et de traduire à la volée les commandes SCSI en commandes ATAPI. À présent, les logiciels de gravage sont tous capables d'utiliser nativement les graveurs IDE, et cette couche d'émulation n'est plus que très rarement nécessaire. Elle ne sera donc pas décrite plus en détail ici.

VIII-B-3-b - Configuration du noyau

L'utilisation des graveurs de CD/DVD ne requiert généralement pas d'option particulière au niveau de la configuration. Autrement dit, si vous pouvez utiliser votre graveur comme un simple lecteur, alors vous pouvez également l'utiliser pour graver. Cependant, il peut-être intéressant d'activer le support de certains systèmes de fichiers pour utiliser correctement les lecteurs et les graveurs.

Les principales options de configuration du noyau ayant trait aux lecteurs et aux graveurs de CD/DVD sont récapitulées ci-dessous :

« Loopback device support »

Cette option permet d'activer une fonctionnalité permettant d'utiliser un fichier comme un périphérique normal. Elle est très utile pour tester les images de CD, car il suffit simplement de monter le fichier image comme un périphérique « loopback ». Il est donc recommandé d'activer cette fonctionnalité, que l'on dispose d'un graveur SCSI ou IDE. Cette fonctionnalité peut être activée sous la forme de module ou non, la réponse recommandée est 'Y'.

« Include IDE/ATA-2 CDROM support »

Cette option permet d'activer la gestion des lecteurs de CD IDE ATAPI. Vous devez répondre à cette question par 'Y' si vous disposez d'un graveur ou d'un lecteur IDE, ce qui est généralement le cas. La réponse recommandée est 'Y'.

« SCSI support »

Cette option permet d'activer la gestion des périphériques SCSI dans votre noyau. Il va de soi qu'il faut l'activer si vous disposez d'un graveur SCSI. Cette fonctionnalité peut être activée sous forme de module ou non. La réponse recommandée est 'N'.

« SCSI CD-ROM support »

Cette option permet d'activer la gestion des lecteurs de CD SCSI. Il faut l'activer si l'on dispose d'un graveur SCSI. La réponse recommandée est 'Y' si vous disposez d'un lecteur ou d'un graveur SCSI, et 'N' dans le cas contraire.

« Enable vendor-specific extensions (for SCSI CDROM) »

Cette option permet d'autoriser l'emploi d'extensions au protocole SCSI définies par les fabricants de matériels. Certains graveurs utilisent de telles extensions, c'est en particulier le cas des graveurs de CD HP. Toutefois, si vous ne disposez pas d'un tel graveur, il est peu probable que vous ayez à activer cette fonctionnalité. La réponse recommandée est donc 'N'.

« SCSI generic support »

Cette option permet d'utiliser les périphériques SCSI avec des commandes non standards, ce qui requiert l'emploi de programmes capables de communiquer directement avec les périphériques SCSI. C'est le cas pour les graveurs, qui seront pilotés directement par les logiciels de gravage. Il faut généralement activer cette fonctionnalité si l'on dispose de périphérique SCSI, aussi la réponse recommandée est-elle 'Y' si l'on se trouve dans cette situation.

« Probe all LUNs on each SCSI device »

Cette option permet, lorsqu'on dispose de périphériques SCSI capables de gérer plusieurs numéros d'unité logiques, de leur demander tous ces numéros. Ce type de périphérique est assez rare, et en général chaque périphérique n'utilise qu'un et un seul numéro d'unité logique. Il ne faut donc pas, en général, activer cette fonctionnalité. La réponse recommandée pour cette question est donc 'N'.

« ISO 9660 CDROM file system support »

Cette option permet de prendre en charge les systèmes de fichiers des CD-ROM. Il est impératif d'activer cette fonctionnalité, aussi la réponse recommandée pour cette question est-elle 'Y'.

« Microsoft Joliet CDROM extensions »

Cette option permet de prendre en charge les extensions Microsoft au système de fichiers ISO 9660 (pour la gestion des noms longs du DOS). Étant donné le nombre de CD qui utilisent ces extensions, il est impératif d'activer cette fonctionnalité. La réponse recommandée pour cette question est donc 'Y'.

« UDF file system support »

Cette option permet de prendre en charge les systèmes de fichiers UDF, utilisé par certains CD-ROM gravés par paquets et par tous les DVD. Il est impératif d'activer cette fonctionnalité, aussi la réponse recommandée pour cette question est-elle 'Y'.

Enfin, il faut choisir le pilote bas niveau permettant de gérer son graveur de CD. Pour les graveurs IDE ATAPI il n'y a pas de pilote bas niveau à inclure. Pour les graveurs SCSI, il faut choisir le pilote approprié de votre carte ou de votre contrôleur SCSI.

Une fois cette configuration effectuée, il ne reste plus qu'à compiler le noyau et les modules et à les installer. La manière de procéder a été décrite dans la section traitant de la [compilation du noyau](#).

VIII-B-3-c - Configuration des modules du noyau

Vous aurez sans doute à modifier le fichier de configuration `/etc/modprobe.conf` si vous avez activé certaines fonctionnalités SCSI sous la forme de modules, afin de charger les modules nécessaires lorsqu'une commande sera effectuée sur l'un des fichiers spéciaux de périphériques SCSI. Si toutes les fonctionnalités SCSI sont intégrées au noyau, cette étape est bien entendu facultative. Les fichiers spéciaux utiles pour les périphériques blocs SCSI sont les fichiers `scdx` et `sgx`, où `x` est le numéro du périphérique. Les fichiers spéciaux du premier groupe sont de type bloc, et permettent d'accéder aux lecteurs de CD SCSI. Les fichiers du deuxième groupe en revanche sont de type caractère, et permettent d'accéder aux périphériques SCSI de manière générique. Les codes majeurs pour ces deux groupes de fichiers spéciaux sont respectivement 11 et 21, il faut donc ajouter les entrées du fichier `modprobe.conf` pour les modules nommés `block-major-11` et `char-major-21`. Normalement, ces entrées sont déjà présentes dans le fichier fourni par votre distribution.

VIII-B-3-d - Installation des logiciels de gravage

Lorsque le noyau aura été correctement configuré et installé, il ne vous restera plus qu'à installer les logiciels de gravage. Les logiciels indispensables sont les suivants :

- `cdrecord`, qui permet de piloter le graveur de CD et d'effectuer les tâches nécessaires pour le gravage ;
- `mkisofs`, qui permet de créer des images disques ISO 9660, éventuellement avec les extensions Joliet (CD Windows 95), Rock Ridge (CD Unix) ou HFS (CD Macintosh) ;
- `cdda2wav`, qui permet d'extraire les données numériques des CD audio ;
- `cdrecord`, qui permet de réaliser des gravages bas niveau et en mode « Disk At Once ».

Vous pourrez vérifier que votre configuration fonctionne correctement si vous disposez d'un périphérique SCSI ou si vous utilisez l'émulation SCSI avec la commande suivante :

```
cdrecord -scanbus
```

Cette commande recherche tous les périphériques SCSI du système. Vous devriez normalement y trouver, en plus de vos autres périphériques SCSI, votre graveur de CD. Cette commande ne fonctionne pas si vous utilisez un graveur IDE.

VIII-B-3-e - Utilisation des logiciels de gravage

Nous allons à présent voir les commandes permettant de faire les principales opérations ayant trait au gravage des CD. Bien que ce document soit consacré à l'installation de Linux et non à son utilisation, elles vous permettront de tester si tout fonctionne correctement.

La copie directe d'un CD peut se faire avec la commande suivante :

```
cdrecord -dummy -v dev=graveur speed=n -isosize /dev/source
```

L'option `dev` permet de spécifier le périphérique du graveur à utiliser. Cette option peut prendre en paramètre soit le fichier spécial de périphérique du graveur (par exemple `/dev/hdc`), soit un triplet de valeur permettant d'identifier un périphérique sur le bus SCSI (par exemple `0,1,0`). La première valeur représente le bus SCSI sur lequel le graveur est branché, la deuxième le numéro de périphérique du graveur sur ce bus SCSI et la troisième son numéro d'unité

logique (« Logical Unit Number »). Le numéro du bus SCSI et le numéro de périphérique peuvent être obtenus avec la commande `cdrecord -scanbus` présentée ci-dessus si vous utilisez un graveur SCSI. Le numéro d'unité logique est, dans la majorité des cas, 0.

Le nombre `n` donné dans la ligne de commande précédente doit valoir le facteur multiplicateur de la vitesse de gravage. Vous pouvez essayer avec des valeurs faibles pour commencer, afin d'être sûr que tout se passe bien. Le fichier de périphérique `/dev/source` quant à lui représente le fichier spécial de périphérique du lecteur de CD utilisé pour lire le CD-ROM. L'option `-isosize` permet de demander à `cdrecord` de lire la taille de la piste à graver dans le système de fichiers ISO9660 du CD source. Enfin, l'option `-dummy` permet de faire toutes les opérations en conservant le laser du graveur éteint, ce qui revient à faire un test. Si vous voulez réellement graver votre CD, il suffit de supprimer cette option.

Il va de soi que si vous ne disposez pas d'un lecteur de CD en plus de votre graveur, vous ne pouvez pas utiliser la commande précédente. Dans ce cas, vous devrez faire une image disque en extrayant toutes les données du CD source. Cela peut être réalisé avec la commande suivante :

```
dd if=/dev/source of=image
```

où `/dev/source` est le fichier spécial de périphérique du lecteur utilisé pour faire l'image, et `image` est le fichier image qui doit être créé.

Le gravage d'une image disque peut être réalisé avec la commande suivante :

```
cdrecord -dummy -v dev=graveur speed=n -data image
```

où les options sont les mêmes que dans les commandes précédentes.

Si vous désirez créer une image disque à partir des fichiers de votre disque dur, vous pouvez utiliser `mkisofs`. Ce programme permet de créer une image disque ISO9660, avec les extensions Rock Ridge pour activer les fonctionnalités des systèmes de fichiers Unix (noms de fichiers plus longs, liens symboliques, droits d'accès...), et éventuellement les extensions Joliet utilisées par les systèmes Windows 95, Windows NT, Windows 2000 et XP (noms de fichiers longs Microsoft). Le prix à payer pour cette amélioration est la perte de quelques centaines de kilo-octets, ce qui est dérisoire étant donné le gain en portabilité. La ligne de commande à utiliser pour créer une image disque est la suivante :

```
mkisofs [-J][-r | -R][-hfs | -apple] [-C début,fin [-M disque]] \
[-V "nom"] -o image répertoires
```

Les options `-r` ou `-R`, `-J` et `-hfs` ou `-apple` permettent respectivement d'utiliser les extensions Rock Ridge, Joliet ou HFS (pour les disques devant être lus sur les Macintosh). Il est possible de faire des images disque hybrides, qui pourront être lues sur plusieurs systèmes d'exploitation différents. La seule contrepartie est la perte de quelques dizaines de kilo-octets, ce qui est dérisoire. La distinction entre l'option `-r` et l'option `-R` est que `-r` modifie les attributs des fichiers Unix afin que le CD puisse être utilisé sur un autre ordinateur que celui sur lequel le CD a été créé. En particulier, le propriétaire et le groupe des fichiers voient leurs valeurs fixées à 0. L'option `-V` permet de fixer le nom du volume dans l'image. Il est possible de placer ce nom entre guillemets. Cette fonctionnalité est intéressante si ce nom comprend des espaces.

Il est possible de créer des CD multisessions à condition de spécifier l'emplacement de cette session lors de la création de l'image disque. Cela se fait avec l'option `-C` de `mkisofs`. Cette option prend en paramètre le numéro du secteur de début et le numéro du secteur de fin de la session à la suite de laquelle la nouvelle session doit être écrite, séparés par des virgules. Si l'on veut également importer la session précédente (c'est-à-dire permettre l'accès aux données des sessions précédentes comme si elles étaient également présentes dans la session à écrire), il faut utiliser l'option `-M`, à laquelle il faut indiquer le fichier spécial de périphérique du lecteur contenant le disque dont la session doit être importée.

La détermination des secteurs de début et de fin de la session précédente peut être difficile, mais `cdrecord` est capable d'obtenir cette information avec la commande suivante :

```
cdrecord dev=graveur -msinfo
```

Appelé ainsi, `cdrecord` affiche les secteurs de début et de fin de la dernière session du disque du lecteur spécifié par son option `dev`, directement dans le format attendu par `mkisofs`.

L'option `-o` de `mkisofs` permet de spécifier le nom du fichier image qui doit être créé. Les paramètres suivant l'option `-o` constituent la liste des répertoires et des fichiers qui doivent être insérés dans le CD-ROM. Par défaut, chaque répertoire ou fichier indiqué en paramètre est placé dans le répertoire racine du CD-ROM, et l'arborescence des sous-répertoires est respectée. Cependant, il est possible de placer un des répertoires ou fichiers indiqués en paramètre dans un autre répertoire du CD-ROM, à l'aide de la syntaxe suivante :

```
destination=source
```

où `destination` est le répertoire destination du CD-ROM, et `source` le répertoire ou le fichier source à ajouter à l'image disque. De même, il est possible d'exclure certains sous-répertoires de l'image disque à l'aide de l'option `-x` :

```
-x répertoire
```

Les sous-répertoires du répertoire indiqué seront également supprimés de l'image disque.

Vous pouvez tester votre image disque en la montant par l'intermédiaire du périphérique virtuel du noyau loopback. Il faut pour cela que vous ayez activé la gestion de ce périphérique dans la configuration du noyau. La commande à utiliser est la suivante :

```
mount -t iso9660 -o ro,loop=/dev/loop0 image /cdrom
```

où `image` est le nom de votre fichier image à tester. Cette commande monte le système de fichiers de cette image dans le répertoire `/cdrom`. La commande `umount` peut être utilisée pour démonter ce système de fichiers de la manière habituelle.

Lorsque votre image disque est satisfaisante, vous pouvez la graver avec la première ligne de commande suivante :

```
cdrecord -v dev=graveur speed=n -multi -data image
```

Vous noterez la présence de l'option `-multi`, qui permet de demander à `cdrecord` de créer un disque multisession. Bien entendu, vous pouvez vous passer de cette option si votre disque n'est pas multisession.

La création d'une image disque nécessite d'avoir une place disque égale à la taille de la session à ajouter, ce qui peut être contraignant. Sur les systèmes suffisamment puissants, il est possible de créer l'image disque et de la graver à la volée en créant un tube entre `mkisofs` et `cdrecord`. Dans ce cas, il ne faut pas donner de nom à l'image disque dans la ligne de commande de `mkisofs`, et utiliser l'option `-` dans la ligne de commande de `cdrecord` pour lui demander de prendre les données sur son entrée standard. Malheureusement, cette technique n'est pas utilisable facilement pour les disques multisessions, parce que la lecture des informations de la session précédente peut se faire en même temps que le début du gravage. Pour résoudre ce problème, il faut utiliser l'option `-waiti` de `cdrecord`, qui lui demande d'attendre l'arrivée des premières données dans le tube avant d'ouvrir le fichier spécial de périphérique du graveur. La ligne de commande complète devient alors assez complexe, comme vous pouvez en juger :

```
mkisofs -r -M périphérique -C `cdrecord dev=graveur -msinfo` \  
fichiers | cdrecord -v dev=graveur speed=vitesse -multi -waiti -
```

Les commandes présentées ci-dessus ne permettent pas de travailler avec des CD audio. Pour extraire les données audio d'un CD, vous devrez utiliser le programme `cdda2wav` :

```
cdda2wav [-H] -B nom [-tdébut[+fin]] -O wav dev=graveur
```

L'option `-H` permet d'éviter la création des fichiers d'information `.inf` sur les pistes extraites par `cdda2wav`. L'option `-B` permet d'utiliser le nom `nom` complété d'un numéro pour les fichiers son créés. Les pistes seront donc stockées dans des fichiers portant les noms `nom_01`, `nom_02`, etc. L'option `-t` permet d'indiquer la piste début et la piste fin afin de sélectionner les pistes dont les données audio doivent être extraites. Si la piste de fin n'est pas précisée, toutes les pistes depuis la piste de début jusqu'à la dernière piste du CD seront extraites. De même, si la piste de début n'est pas précisée, l'extraction commencera à partir de la première piste. L'option `-O` permet d'indiquer le type de fichier de sortie, `wav` indique ici que les fichiers créés seront au format WAV des fichiers son de Windows. Il est également possible d'utiliser le type de fichier `raw`, avec lequel les fichiers seront prêts à être gravés tels quels. Enfin, l'option `dev` permet de spécifier le périphérique à utiliser pour lire les données audio. La syntaxe de cette option est la même que celle utilisée par `cdrecord`.

Le gravage des pistes audio peut être réalisé avec la commande suivante :

```
cdrecord -dummy -v dev=graveur speed=n -nofix -audio piste1.wav \
piste2.wav &#8230;
```

L'option `-nofix` permet ici d'éviter de fermer la session. Elle doit être utilisée si l'on désire rajouter des pistes audio sur le CD-ROM ultérieurement. Si, en revanche, elle est omise, `cdrecord` fermera automatiquement la session après l'écriture de la dernière piste, et plus aucune donnée ne pourra être ajoutée. La commande suivante vous permettra de fixer le disque a posteriori :

```
cdrecord -dummy -v dev=graveur speed=n -fix
```

Enfin, si vous disposez d'un graveur de CD réinscriptible, vous pourrez utiliser la commande suivante pour effacer un CDRW :

```
cdrecord -dummy -v dev=graveur -blank=fast
```

où `graveur` représente toujours le périphérique du graveur.

Il est peu probable que vous ayez à utiliser ces commandes directement, car il existe de nombreux programmes graphiques capables de les piloter de manière conviviale (`k3b`, `gcombust`, `xcdroast`, etc.). Il est vivement recommandé d'en installer un et de l'utiliser. Ils sont relativement simples d'emploi et leur utilisation ne devrait pas poser de problème particulier.

VIII-C - Configuration des cartes filles

Il existe un grand nombre de cartes filles sur le marché, dont les plus courantes sont sans doute les cartes graphiques, les cartes son et les cartes réseau. Toutes ces cartes permettent d'ajouter des fonctionnalités de manière modulaire à une machine, en les enfichant dans les emplacements prévus à cet effet sur la carte mère.

Les cartes les plus vieilles que l'on peut encore rencontrer sont les cartes ISA. La configuration de ces cartes a toujours été extrêmement difficile, en raison des nombreux conflits matériels qu'elles provoquaient entre elles et avec les ressources de la carte mère. Le standard Plug and Play a été introduit pour limiter ces problèmes par la suite, mais ce n'est qu'avec l'arrivée des cartes PCI que l'on a enfin pu commencer à brancher n'importe quoi n'importe comment dans un ordinateur (cela dit, ce n'est pas encore la panacée parce que l'on est toujours obligé d'éteindre l'ordinateur...). Enfin, les cartes graphiques ne se connectent plus de nos jours ni sur les ports PCI, ni sur les ports ISA, pour des raisons de performances et de débit de données. Elles utilisent un port spécial, le port AGP. Cela ne pose cependant pas de problème de configuration spécial (du moins si l'on se contente d'une seule carte graphique).

Ce chapitre traite de la manière de configurer les cartes ISA et PCI en général, et donne des informations plus détaillées pour chacun des types courants de cartes filles.

VIII-C-1 - Généralités sur les cartes ISA, Plug And Play et PCI

L'architecture initiale des PC a très mal vieilli et souffre actuellement de défauts majeurs qui nuisent à leur développement. Sans parler des goulots d'étranglement sur les différents bus systèmes qui ne suivent plus les vitesses des processeurs actuels, la plupart des périphériques sont obligés de se restreindre à des protocoles de communication obsolètes pour conserver une compatibilité ascendante. L'une des limitations majeures de cette architecture est le nombre incroyablement restreint de lignes d'interruption permettant à un périphérique de signaler au processeur qu'il a besoin de l'intervention de son gestionnaire pour poursuivre son travail, et la quasi-absence des possibilités d'accès directs à la mémoire, qui impose l'utilisation du processeur pour réaliser de simples transferts de données entre les périphériques et la mémoire.

Ces ressources limitées, en plus de pénaliser les performances, posent des problèmes évidents d'extensibilité. En effet, qui dit ressource limitée dit qu'il est impossible de connecter plus de périphériques « intelligents » que ce que l'architecture matérielle autorise. L'installation de plusieurs cartes ISA gérant chacune une interruption ou un canal DMA était réellement un casse-tête il y a encore peu de temps. C'est pour résoudre ces problèmes de configuration du matériel que le standard Plug and Play a été développé. Ce standard établit un protocole de communication matériel permettant au BIOS de configurer automatiquement les périphériques ISA en évitant les conflits sur les ressources partagées (le plus souvent, les lignes d'interruption).

Ce protocole a soulagé bien des gens, mais suppose un support spécifique de la part du système d'exploitation, puisque les paramètres matériels de chaque carte Plug and Play ne sont pas fixés d'un démarrage à l'autre de la machine. Il faut donc que le système soit ou capable de récupérer ces paramètres, ou de refaire la configuration du matériel. Cette deuxième solution est souvent imposée par le fait que bon nombre de BIOS sont bogués et effectuent une mauvaise configuration des cartes Plug and Play.

Le Plug and Play a été une amélioration sensible, mais n'a pas résolu les problèmes concernant les ressources limitées des PC. Heureusement, depuis l'avènement du bus PCI, ces limitations se sont estompées. En effet, le bus PCI permet non seulement aux périphériques de s'autoconfigurer et de se déclarer dans le système PCI, mais également de partager une même ligne d'interruption et de prendre le contrôle du bus mémoire. Ainsi, il n'y a virtuellement plus aucune difficulté pour installer une carte PCI dans un ordinateur. De nos jours, on ne trouve quasiment plus de cartes ISA, et la plupart des cartes mères n'acceptent même plus ces cartes.

Mais ne croyez pas pour autant que le bus PCI soit idéal, c'est loin d'être le cas. En effet, il ne permet pas encore l'ajout et la suppression de cartes à chaud (c'est-à-dire lorsque le système est allumé), et fonctionne désormais à une vitesse bien trop faible compte tenu de la vitesse des processeurs, cartes graphiques et circuits mémoire. Il est donc encore appelé à évoluer.

Le sous-système PCI de Linux permet d'utiliser directement les cartes PCI, sans configuration particulière. Les périphériques ISA, en revanche, peuvent être plus difficiles à configurer. S'ils ne sont pas Plug and Play, il n'y a pas d'autre choix que de spécifier leurs paramètres matériels directement, soit lors de la compilation du noyau, soit à l'aide d'options de boot lorsque le système démarre, soit à l'aide d'options dans le fichier `/etc/modprobe.conf` si leur gestionnaire est compilé sous forme de module. La première solution est la plus technique, puisqu'elle nécessite de reconfigurer et de recompiler le noyau, les deux autres relèvent de la configuration simple.

Les cartes ISA Plug and Play constituent un cas à part, car, comme il l'a été dit ci-dessus, la plupart des BIOS Plug and Play sont bogués et les initialisent mal. Il est donc nécessaire que le système les initialise lui-même, et détermine par la même occasion leur configuration matérielle. En pratique, il existe deux possibilités pour effectuer ces deux opérations. La première possibilité est de laisser le noyau faire tout le travail de configuration des cartes et d'allocation des ressources. La deuxième possibilité est d'effectuer la configuration des cartes ISA Plug and Play au niveau applicatif dans les scripts de démarrage du système. Cette solution est relativement technique, et ne doit être utilisée que lorsqu'on désire contrôler finement les ressources allouées par les périphériques dans le système.

Si l'on désire utiliser les fonctionnalités Plug and Play du noyau, il n'y a strictement rien à faire. Linux se charge simplement d'initialiser automatiquement les cartes ISA Plug and Play lors du démarrage du système, de leur attribuer les paramètres matériels adéquats, et de communiquer ces paramètres aux gestionnaires de périphériques. Il prendra

également en charge la gestion des conflits avec les autres périphériques, qu'ils soient ISA non Plug and Play ou PCI. Par conséquent, vous n'aurez aucune configuration spécifique à effectuer.

Mais Linux fournit également la possibilité de contrôler soi-même les ressources allouées à chaque périphérique, pour ceux qui veulent avoir la maîtrise totale de leur matériel ou ceux qui ont une configuration tellement spécifique qu'elle nécessite un paramétrage manuel. Dans ce cas, l'initialisation des cartes ISA Plug and Play ne se fera pas lors du démarrage du noyau, mais plutôt ultérieurement. En général, on effectue cette tâche dans les scripts d'initialisation du système, mais ce n'est pas une obligation. Quoi qu'il en soit, comme l'attribution des ressources aux cartes est différée, les gestionnaires de périphériques ne peuvent pas être inclus dans le noyau. Il est donc nécessaire d'utiliser les modules du noyau pour ces gestionnaires. Les paramètres matériels pourront alors être communiqués simplement à ces gestionnaires lors du chargement des modules, à l'aide d'options dans le fichier de configuration `/etc/modprobe.conf`.

La configuration manuelle des cartes ISA Plug and Play se fait classiquement à l'aide des outils « isapnp ». Parmi ces outils se trouve le programme `isapnp`, que l'on utilise pour initialiser les cartes ISA Plug and Play. Cet utilitaire utilise les informations qui se trouvent écrites dans le fichier de configuration `/etc/isapnp.conf` pour déterminer les plages d'entrée/sortie, les canaux DMA et les lignes d'interruption à utiliser pour chaque carte.

La rédaction manuelle du fichier `isapnp.conf` n'est pas une tâche aisée. Heureusement, cela peut être réalisé automatiquement, à l'aide d'un autre utilitaire nommé `pnpdump`. Celui-ci affiche la liste des différentes possibilités de configuration pour chaque périphérique ISA Plug and Play. Cette liste est affichée exactement sous une forme exploitable par `isapnp`, ce qui fait qu'il est très simple de créer un fichier de configuration correct en faisant une redirection de sa sortie standard dans un fichier :

```
pnpdump > /etc/isapnp.conf
```

Le fichier de configuration ainsi créé contient les différentes configurations possibles. Cependant, elles sont toutes commentées, et aucun périphérique ISA ne sera configuré sans intervention supplémentaire. Il va donc falloir éditer ce fichier et retirer les commentaires devant les options de configuration qui vous intéressent. Les lignes de commentaires commencent toutes par un caractère dièse (`#`). Il suffit donc d'effacer ce caractère pour les décommenter. Vous devez choisir les options à décommenter de telle manière qu'aucun conflit d'adresse, d'interruption ou de canal DMA n'existe dans votre système. Pour chaque périphérique, plusieurs possibilités sont offertes, mais vous ne devez retirer les commentaires que devant les lignes d'une seule de ces possibilités. Les zones à décommenter sont clairement identifiées dans le fichier `isapnp.conf` généré par `pnpdump`, il vous suffit donc d'effectuer le choix de la configuration à utiliser. Enfin, il ne faut pas oublier de retirer le commentaire devant la ligne suivante :

```
(ACT Y)
```

à la fin des différentes options de configuration pour chaque carte correctement configurée, faute de quoi elle ne sera pas activée lors de l'appel à `isapnp`.

En général, il est souhaitable d'appeler la commande `isapnp` à chaque démarrage du système, dans un des scripts de démarrage. Vous devrez donc inclure une ligne telle que celle-ci :

```
/sbin/isapnp /etc/isapnp.conf
```

dans le fichier de démarrage principal de votre système. Ce fichier peut se trouver dans le répertoire `/etc/rc.d/` (ou dans `/sbin/init.d/` selon la distribution que vous utilisez). La plupart des distributions permettent de paramétrer l'appel à cette commande à l'aide d'une option de configuration modifiable à l'aide de leur programme de configuration. Consultez la documentation de votre distribution pour plus de détails à ce sujet.

Une fois que vous aurez déterminé la configuration correcte pour vos périphériques dans le fichier `/etc/isapnp.conf`, vous pourrez charger les modules du noyau gérant ces périphériques. Cela peut nécessiter la modification du fichier `/etc/modprobe.conf`. Afin de limiter les risques d'erreur, vous devriez procéder comme suit :

- déterminer le fichier spécial de périphérique utilisé par les applications pour accéder à votre périphérique ;
- déterminer le nom du module que le noyau tentera de charger lorsqu'un programme tentera d'utiliser ce fichier spécial de périphérique ;
- rechercher la ligne définissant l'alias pour ce nom de module, afin de savoir quel est le module qui sera effectivement chargé par modprobe. Vous pouvez également trouver ce nom dans la documentation de la configuration du noyau pour le gestionnaire du périphérique que vous configurez, ou en regardant directement dans le répertoire d'installation des modules `/lib/modules/` ;
- ajouter éventuellement la ligne « options module paramètres » permettant de spécifier les paramètres de chargement paramètres pour le module module ;
- ajouter éventuellement les lignes « install » et « remove », permettant d'effectuer des actions complémentaires lors du chargement et du déchargement du module.

Comme il l'a été indiqué dans la [Section 8.1.1](#), la détermination des noms de modules utilisés par le noyau pour les requêtes de chargement automatique n'est pas facile. Vous aurez sans doute à vous servir du type, du code majeur et du code mineur de ce fichier spécial de périphérique. Vous pourrez obtenir ces informations à l'aide de la commande `ls -l` fichier, où fichier est le nom du fichier spécial de périphérique. Le type de ce fichier est indiqué dans le premier caractère des droits du fichier. Le caractère 'c' indique que le périphérique est un périphérique de type caractère, et le caractère 'b' indique qu'il s'agit d'un périphérique de type bloc. Les numéros de codes majeur et mineur quant à eux sont indiqués juste après les informations concernant le propriétaire et le groupe du fichier, généralement égaux à « root ».

Quoi qu'il en soit, il est fortement recommandé de lire la documentation du module gérant votre périphérique Plug and Play. Cette documentation est en général située dans le répertoire `/usr/src/linux/Documentation/`.

Pour tester si les modifications que vous avez effectuées sont correctes, vous pouvez essayer de charger le module avec la commande suivante :

```
modprobe module
```

où module est le nom du module à charger. Vous pouvez également vérifier que ce module se charge bien lorsqu'une requête sur le fichier spécial de périphérique correspondant est effectuée, avec par exemple la commande suivante :

```
echo Coucou > /dev/périphérique
```

où périphérique est le fichier spécial de périphérique à tester. Vous pouvez voir si le module s'est correctement chargé en demandant la liste des modules chargés à l'aide de la commande `lsmod`. Cette commande vous indique également l'état de chaque module, ainsi que le nombre de fois qu'il est utilisé et par qui. Si un module est marqué « uninitialized », vous avez de fortes chances de devoir redémarrer l'ordinateur et de revoir votre configuration, car un module qui se trouve dans cet état est inutilisable et peut parfois même refuser de se décharger.

VIII-C-2 - Configuration des cartes son

VIII-C-2-a - Fonctionnalités du matériel

Les fonctionnalités fournies par les gestionnaires de son de Linux varient dans de larges proportions, selon le matériel utilisé. La quasi-totalité des cartes son gèrent bien entendu l'essentiel : la restitution de fichiers audio numériques. Les fonctionnalités des cartes son sont à ce niveau relativement disparates : les différences vont du taux d'échantillonnage utilisé jusqu'à la possibilité de les faire fonctionner en full duplex (enregistrement et lecture simultanée) ou non. Cette dernière fonctionnalité est essentielle pour les applications de téléphonie sur Internet par exemple.

De plus, très peu de cartes son sont capables de gérer la restitution des fichiers MIDI correctement (les fichiers MIDI sont des fichiers musicaux permettant de stocker les séquences de notes à jouer pour restituer le morceau, au lieu de stocker le son lui-même sous forme numérique). La plupart des cartes son ne fournissent aucun support matériel pour cela, et la restitution des notes des fichiers MIDI se fait généralement à l'aide d'un synthétiseur logiciel. C'est en particulier le cas pour toutes les cartes son intégrées sur les cartes mères des ordinateurs récents. Les gestionnaires

de périphériques de Linux ne fournissent pas de synthétiseur logiciel, en revanche il existe plusieurs programmes permettant de lire les fichiers MIDI, moyennant une consommation plus ou moins importante des ressources de calcul de l'ordinateur.

La plupart des cartes son bas de gamme disposent toutefois d'un synthétiseur en modulation de fréquence (synthétiseur dit « FM »), qui permet de simuler des notes d'instrument de musique, généralement avec une qualité très médiocre. Certains pilotes de cartes son de Linux permettent d'utiliser ces synthétiseurs, mais ce n'est toujours pas la panacée. En réalité, si l'on veut restituer correctement les notes des fichiers MIDI, il faut utiliser une carte son disposant de ce que l'on appelle une table d'échantillons. Ces tables permettent de stocker dans la mémoire de la carte son des échantillons de sons d'instruments de musique enregistrés en qualité numérique. La carte son peut ainsi reconstituer le son de toutes les notes de musique fidèlement, avec une très grande ressemblance avec l'instrument réel. Ces cartes sont bien entendu plus coûteuses, mais la qualité sonore est incomparable à ce niveau (les cartes SoundBlaster AWE et Live en particulier disposent de tables d'échantillons). Linux permet parfaitement d'utiliser ces cartes, et même de charger de nouveaux jeux d'échantillons dans la mémoire de la carte pour changer la sonorité des instruments.

Je ne saurais donc que trop vous conseiller de bien vous renseigner avant l'achat d'une carte son, car ce sujet n'est pas très souvent pris en considération lors de l'achat. Bien entendu, si vous ne désirez pas utiliser votre carte son dans ses derniers retranchements, vous pouvez vous contenter des cartes intégrées sur les cartes mères. Dans les autres cas, renseignez-vous.

VIII-C-2-b - Configuration du noyau

La plupart des cartes son vendues actuellement sont des cartes PCI, qui se configurent relativement aisément. Cependant, il existe encore un bon nombre de cartes son ISA, dont la configuration peut être plus technique, surtout si elles ne sont pas Plug and Play.

La première étape lors de la configuration de votre carte son sera la sélection du gestionnaire de périphériques à utiliser dans le programme de configuration du noyau. En ce qui concerne les cartes son, les options de configuration se trouvent dans le menu « Sound ». Vous y trouverez deux jeux de pilotes : les pilotes ALSA (« Advanced Sound Linux Architecture ») et les pilotes OSS (« Open Sound System »). Les pilotes ALSA sont les plus modernes, et devront être utilisés en priorité, car ils fournissent le plus de fonctionnalités. Les pilotes OSS sont des pilotes plus anciens, qui sont toutefois maintenus à titre de compatibilité d'une part et, d'autre part, parce que certaines cartes son ne sont pas encore totalement prises en charge par les pilotes ALSA. Les pilotes OSS sont eux-mêmes classés en deux catégories : les pilotes classiques (première partie des options du menu) et les pilotes OSS purs. Ces derniers utilisent une spécification d'interface de programmation commune permettant l'accès aux cartes son, mais il n'y a pas de différence au niveau des applications pour l'utilisation de ces pilotes. Vous devez choisir le gestionnaire de périphériques ALSA ou OSS qui convient le mieux à votre matériel.

L'erreur la plus classique que l'on peut faire au niveau du choix du pilote est de supposer que l'on possède une carte compatible Sound Blaster alors que ce n'en est pas une. Je tiens à préciser que quasiment aucune carte dite « compatible » sous Windows ne l'est sous Linux. La compatibilité sous Linux, c'est le fait d'avoir quasiment la même électronique ou du moins les mêmes interfaces au niveau matériel. Sous Windows, la compatibilité n'existe qu'au niveau des interfaces fournies par le pilote de la carte son. Par conséquent, il vous faut savoir exactement de quelle nature est votre carte son, et non ce que vous avez retenu des arguments commerciaux du fabricant. Notez en particulier que certaines cartes Sound Blaster ne sont pas compatibles Sound Blaster (Creative Labs est renommé en ce qui concerne les incompatibilités entre les différents modèles de cartes son). C'est notamment le cas pour les cartes sons SB64 PCI et SB128 PCI, qui sont en réalité des cartes son ESS1370 ou ESS1371, et dont l'électronique est fabriquée par la société Ensoniq (cette société a été rachetée par Creative Labs, qui vend ces cartes en s'appuyant sur son image de marque et qui sème ainsi la confusion sur le marché). En conclusion, si vous ne voulez pas essayer plusieurs pilotes et recompiler le noyau jusqu'à ce que vous trouviez le bon, renseignez-vous bien sur la nature de votre carte son (si possible avant de l'acheter, au moins afin de savoir exactement ce que vous aurez). Vous pouvez également taper la commande `lspci` afin de voir les périphériques réellement présents sur le bus PCI. La commande système `dmesg` peut également vous être utile. Elle permet de réafficher la liste des messages de traces générés par le noyau, et vous y trouverez donc les messages relatifs à la phase de détection des périphériques.

Lorsque vous aurez choisi le gestionnaire de périphériques adéquat, vous aurez le choix entre le compiler à l'intérieur du noyau (en choisissant l'option 'Y') ou le compiler sous forme de module (en choisissant l'option 'M'). Il est possible soit de charger ces pilotes en tant que module, soit de les intégrer au noyau. En général, il est préférable d'utiliser les modules, car les programmes de configuration audio peuvent ainsi détecter automatiquement les cartes son utilisées. De plus, certaines fonctionnalités non essentielles sont souvent désactivées par défaut par les pilotes, et doivent être activées explicitement à l'aide d'une option lors du chargement du module ou d'un paramètre à fournir au noyau lors de son amorçage. Les modules sont donc là aussi plus faciles à utiliser, car ils éviteront de redémarrer le système à chaque essai. Cependant, pour la plupart des paramètres matériels (en particulier les lignes d'interruptions, les ports d'entrée/sortie et les canaux d'accès direct à la mémoire), Linux déterminera automatiquement les valeurs correctes. Cette configuration n'est donc pas à faire, et votre carte son fonctionnera généralement immédiatement.

Les vieilles cartes son ISA qui ne sont pas Plug and Play devront généralement également être intégrées au noyau. En effet, pour ces cartes sons, la configuration logicielle est très simple, puisqu'on ne peut pas les configurer du tout, et l'on n'a donc pas besoin de modules du tout. Bien entendu, il faut que vous ayez résolu manuellement les conflits possibles de matériel de votre ordinateur en fixant les switches adéquats sur vos cartes filles, mais cela ne concerne pas Linux. Pour ces cartes, il faut généralement indiquer au noyau les paramètres matériels (IRQ, DMA et ports d'entrée/sortie) lors de la configuration.

Il se peut toutefois que vous ne puissiez pas spécifier ces paramètres matériels dans les menus de configuration de Linux. Bien qu'en théorie il soit possible de modifier les fichiers sources de Linux directement pour indiquer ces paramètres, ce n'est pas à la portée du commun des mortels. Par conséquent, on utilisera malgré tout dans ce cas les modules du noyau, et les options matérielles seront indiquées dans le fichier de configuration `/etc/modprobe.conf`. Notez que c'est également de cette manière que vous devrez procéder si vous désirez configurer vous-même l'allocation des ressources pour les cartes son ISA Plug and Play, ou, autrement dit, si vous préférez utiliser l'outil `isapnp` au lieu des fonctionnalités Plug and Play du noyau.

VIII-C-2-c - Configuration des modules du noyau

Si vous compilez les gestionnaires de son sous la forme de modules, vous devrez ajouter les alias et les options nécessaires à leur chargement dans le fichier `/etc/modprobe.conf`.

Les fichiers spéciaux de périphériques utilisés ne sont pas les mêmes pour les pilotes OSS et les pilotes ALSA. Les premiers utilisent les fichiers spéciaux `/dev/audio?`, `/dev/dsp?` et `/dev/mixer?`, dont le numéro de code majeur est 14, alors que les seconds utilisent les fichiers spéciaux de périphériques `/dev/aloadCn` et `/dev/aloadSEQ`, ainsi que tous les fichiers spéciaux de périphérique du répertoire `/dev/snd/`, tous de numéro de code majeur 116. Les pilotes ALSA étant également capables de simuler le comportement des pilotes OSS par compatibilité, ils utilisent le numéro de code majeur 14 également.

Lorsque le noyau désire accéder aux fonctionnalités d'ALSA ou d'OSS, il cherche donc à charger, selon les fichiers spéciaux de périphériques utilisés, les modules `char-major-116` ou `char-major-14`. Le premier doit être associé au module `snd` d'ALSA, alors que le second doit être associé au module `soundcore`. Ces modules prennent en charge respectivement les systèmes de son d'ALSA et d'OSS. Le fichier `/etc/modprobe.conf` devra donc définir ces alias comme suit :

```
# Permet le chargement des pilotes ALSA :
alias char-major-116 snd

# Permet le chargement des pilotes OSS ou l'émulation des pilotes OSS par ALSA :
alias char-major-14 soundcore
```

Les systèmes de son gérés par ces modules permettent de prendre en charge plusieurs cartes son. Chaque carte son est bien entendu référencée par le numéro de code mineur du fichier spécial de périphérique utilisé. Il faut donc définir, pour chaque carte son, le module du gestionnaire de périphérique de cette carte son. Cela se fait en définissant les alias pour les noms de modules génériques `snd-card-n` pour ALSA et `sound-slot-n` pour OSS, où 'n' est le numéro de la carte son.

Par exemple, si l'on dispose d'une carte son SoundBlaster 128 avec les pilotes ALSA, le module à utiliser est `snd-ens1370` ou `snd-ens1371` selon le modèle (parce que ces cartes son sont en réalité des cartes Ensoniq). Le fichier de configuration `/etc/modprobe.conf` devra donc contenir les alias suivants :

```
# La première carte son est gérée par le module snd-ens1371 :
alias snd-card-0 snd-ens1371
alias sound-slot-0 snd-card-0
```

Si vous désirez utiliser les pilotes OSS, le module à utiliser pour ce type de carte se nomme `es1371`, ce qui fait que le fichier de configuration `/etc/modprobe.conf` se réduit dans ce cas à :

```
# La première carte son est gérée par le module es1371 :
alias sound-slot-0 es1371
```

Les modules des gestionnaires de périphériques peuvent prendre des options permettant de contrôler les ressources matérielles et les fonctionnalités prises en charge. Vous pourrez trouver la liste des différents modules de pilotes de périphériques disponibles, ainsi que leurs options, dans la documentation fournie dans le répertoire `Documentation/sound/` des sources du noyau.

Enfin, les pilotes de périphériques OSS n'implémentent pas forcément l'ensemble des fonctionnalités classiques des cartes son. Lorsqu'une de ces fonctionnalités est demandée, le pilote peut donc demander le chargement d'un module complémentaire. Le nom utilisé par le noyau pour ce module est « `sound-service-n-m` », où 'n' est le code majeur du fichier spécial de périphérique de la carte son et m le code de la fonctionnalité utilisée. Ces codes correspondent aux numéros de codes mineurs des fichiers spéciaux de périphériques permettant d'accéder à ces fonctionnalités. Ainsi, la valeur 0 correspond au mixeur (fichier spécial de périphérique `/dev/mixer`), la valeur 2 correspond au port MIDI (fichier spécial de périphérique `/dev/midi`), et les codes 3 et 4 au canal de la carte son (fichiers spéciaux de périphériques `/dev/dsp` et `/dev/audio`). La liste des fichiers spéciaux de périphériques utilisés par OSS peut être consultée dans le fichier `Documentation/devices.txt`.

Si vous utilisez les pilotes ALSA, vous devrez ajouter les alias suivants dans le fichier `/etc/modprobe.conf` afin de charger à la demande les modules d'émulation des services OSS :

```
# Charge les modules ALSA des services OSS :
alias sound-service-0-0 snd-mixer-oss
alias sound-service-0-1 snd-seq-oss
alias sound-service-0-3 snd-pcm-oss
alias sound-service-0-8 snd-seq-oss
alias sound-service-0-12 snd-pcm-oss
```

VIII-C-2-d - Ajustage des paramètres audio avec ALSA

Ces opérations vous permettront de faire fonctionner votre carte son pour toutes les opérations de base, à savoir la lecture et l'enregistrement de fichiers son, le réglage du volume sonore et éventuellement la lecture des CD audio par l'intermédiaire de votre carte son (si, bien entendu, vous avez relié la sortie audio de votre lecteur de CD à l'entrée auxiliaire de votre carte son).

Le réglage des paramètres de votre carte son peut être réalisé très simplement si vous utilisez les pilotes ALSA. En effet, il existe une application en mode texte permettant de réaliser ces réglages. Cette application se lance simplement avec la commande `alsamixer`

Comme vous pouvez le constater, cette application présente une interface très simple, dans laquelle chaque canal de la carte son peut être réglé indépendamment. Pour passer d'un canal à l'autre, il suffit d'utiliser les flèches droite et gauche, et pour régler les volumes, d'utiliser les flèches haut et bas du clavier. Vous pouvez également rendre un canal complètement muet en appuyant sur la touche 'M'. Cette touche permet également d'activer ou de désactiver certaines fonctionnalités de votre carte son. Lorsque les réglages vous conviendront, vous pouvez simplement quitter `alsamixer` en appuyant sur la touche 'Échap'.

Les réglages réalisés de cette manière ne sont pas permanents. Cela signifie que d'un redémarrage à l'autre de la machine, les valeurs par défaut seront reprises. ALSA fournit donc également l'outil `alsactl`, qui permet d'enregistrer les paramètres des cartes son dans le fichier de configuration `/etc/asound.conf`. Pour cela, il suffit simplement d'appeler cette commande avec l'option `store` :

```
alsactl store
```

`alsactl` permet également de restaurer les paramètres sauvegardés, à l'aide de son option `restore`. On aura donc tout intérêt à placer une ligne telle que celle-ci dans les fichiers d'initialisation de sa distribution, si ce n'est déjà fait :

```
alsactl restore
```

Par ailleurs, pour les cartes son qui disposent d'un synthétiseur FM, il peut être intéressant d'activer cette fonctionnalité. Généralement, les pilotes ALSA désactivent cette fonctionnalité. Elle peut toutefois être réactivée en spécifiant le port d'entrée / sortie du synthétiseur FM de la carte son à l'aide de l'option `fm_port` du module prenant en charge votre carte son. De même, vous pourrez activer la gestion des pilotes MIDI en ajoutant l'option `mpu_port` et en indiquant le numéro de port pour le port MIDI. Vous pouvez consulter le fichier de documentation `Documentation/sound/alsa/ALSA-Configuration.txt` des sources du noyau Linux pour plus d'informations à ce sujet. En général, le numéro de port utilisé pour les synthétiseurs FM est le numéro `0x388`, et le numéro du port MIDI est le `0x300`. Par exemple, pour une carte son CMI, la ligne suivante pourra être ajoutée dans le fichier de configuration `modprobe.conf` :

```
options snd-cmipci fm_port=0x388 mpu_port=0x300
```

Si vous désirez compiler en dur le pilote de votre carte son, vous devrez fournir ces informations en ligne de commande à l'amorçage du noyau. Vous devez donc ajouter, pour chacun de ces paramètres, une option à la ligne de commande du noyau. Par exemple, pour une carte son CMI, il faudrait ajouter les options suivantes dans le fichier de configuration du GRUB ou dans le fichier `lilo.conf` :

```
snd-cmipci.index=0 snd-cmipci.fm_port=0x388 snd-cmipci.mpu_port=0x300
```

La première option permet d'indiquer que cette carte son est la première carte son du système. Elle peut être nécessaire si vous avez intégré dans le noyau des pilotes pour d'autres cartes son, réelles ou virtuelles. Les autres options sont directement déduites de celles spécifiées dans le fichier `modprobe.conf`.

Enfin, vous aurez sans doute à charger une banque de sons dans le pilote, que vous utilisiez une synthèse FM ou une table d'échantillons. La manière de procéder est toutefois différente selon la carte son utilisée. Pour les cartes sons qui ne disposent que d'une émulation FM, il faut utiliser le programme `sbiload`. Ce programme est fourni avec les outils complémentaires d'ALSA, et disponible sur le [site web du projet ALSA](http://www.alsa-project.org/). Ce programme permet de charger dans le pilote les définitions des notes en modulation de fréquence. La syntaxe est la suivante :

```
sbiload -p65:0 --opl3 std.o3 drums.o3
```

Cette commande permet de charger dans le périphérique MIDI ALSA `65:0` les fichiers de définition de notes `std.o3` et `drums.o3`. Ces fichiers sont fournis avec `sbiload` (pas dans toutes les versions toutefois). L'option `--opl3` permet d'indiquer le format de ces fichiers. Quant à l'option `-p`, elle permet d'indiquer le port MIDI dans le pilote ALSA. Les numéros de ports disponibles peuvent être obtenus avec la commande suivante :

```
sbiload -l
```

Note : La commande `sbiload` ne fonctionne qu'avec les ports MIDI FM.

Si le périphérique MIDI FM ne s'affiche pas, c'est que soit votre carte son n'en dispose pas, soit il n'a pas été activé dans le pilote et qu'il faut fournir l'option `fm_port` au module ou au noyau lors de son démarrage, soit que le module `snd-synth-opl3` n'a pas été chargé.

Pour les cartes son qui disposent d'une table d'échantillons, le programme à utiliser est différent. Pour les cartes Sound Blaster AWE et Sound Blaster Live, ce programme se nomme sfxload. Ce programme peut être trouvé sur le site <http://mitglied.lycos.de/iwai/awedrv.html>. Vous trouverez également les fichiers de patches utilisables avec les cartes son AWE sur ce site. Vous pouvez utiliser le fichier de définition des patches de Creative, que vous trouverez normalement soit sur une installation de Windows avec une carte son AWE, soit sur vos CD d'installation, soit sur Internet. Le fichier fourni par Creative se nomme synthgm.sbk.

Pour charger les patches dans la mémoire de la carte, il suffit d'utiliser la commande suivante :

```
sfxload /usr/lib/synthgm.sbk
```

(en supposant que le fichier de patches soit placé dans le répertoire `/usr/lib/`). Vous pourrez placer cette commande dans les scripts de démarrage de votre système ou dans une option de chargement du module de prise en charge de votre carte son.

VIII-C-2-e - Fichiers MIDI et synthétiseurs logiciels

Comme indiqué en introduction, la plupart des cartes son ne permettent pas de lire les fichiers MIDI au niveau matériel. Les fichiers MIDI sont des fichiers ne contenant que les « partitions » des morceaux enregistrés. Ils ne contiennent pas les données audio numériques comme c'est le cas pour les fichiers son classiques ou les pistes de CD audio. Cela explique leur taille réduite, mais pose un problème pour leur restitution. En effet, la lecture d'un fichier MIDI suppose que votre carte son dispose d'un jeu de sons numérisés dans sa mémoire afin de pouvoir reconstituer le signal sonore à partir des informations du fichier MIDI. Ces sons sont couramment appelés des « échantillons » (« patches » en Anglais), et ils peuvent être enregistrés soit en dur dans la mémoire de la carte son, soit chargés lors de son initialisation. Il va de soi que certaines cartes son ne disposent pas des fonctionnalités nécessaires à la lecture des fichiers MIDI. Dans ce cas, il faut utiliser un programme externe capable de synthétiser le son à partir d'échantillons enregistrés sur le disque dur et d'envoyer les données audio ainsi créées à la carte son pour la lecture.

Pour les cartes son ne disposant pas de la possibilité de lire les fichiers MIDI, il ne reste que la solution de l'émulation logicielle. Le meilleur synthétiseur logiciel sous Linux est incontestablement le logiciel Timidity. Timidity est un programme de lecture de fichiers MIDI très complet, paramétrable et capable d'utiliser des banques de sons externes variées. De plus, le moteur de Timidity est utilisé par de nombreux autres programmes, en particulier par le lecteur KMidi de KDE.

L'auteur originel de Timidity ne le maintient plus. Cependant, d'autres programmeurs ont pris le relais et l'ont amélioré pour en faire la version Timidity++. Cette dernière version peut être trouvée sur Internet sur le [site de Timidity](#). L'archive que vous récupérerez ne contient pas forcément le programme : il peut ne s'agir que des fichiers sources permettant de le générer. Les notions de fichiers sources et de compilation de programmes exécutables seront expliquées en détail dans le [chapitre suivant](#). En attendant, si vous récupérez les sources, vous pourrez suivre les indications données ci-dessous pour compiler et installer Timidity.

Timidity peut utiliser différentes interfaces utilisateur, accessibles via différentes bibliothèques. De plus, il est capable d'utiliser différentes interfaces pour lire et jouer les morceaux. En particulier, il peut être utilisé comme un serveur MIDI via ALSA afin que tous les programmes MIDI puissent y accéder via les interfaces standard d'ALSA. De même, il est capable de partager le canal audio de sortie de la carte son via le serveur de son aRts de KDE. Pour utiliser ces fonctionnalités, il faut configurer les sources avec la ligne de commande suivante :

```
./configure --prefix=/usr --enable-dynamic --enable-server \  
--enable-interface=gtk,ncurse --enable-audio=arts,alsa,oss --enable-alsaseq
```

Une fois cela fait, vous pourrez générer l'exécutable avec la commande suivante :

```
make
```

et l'installer avec la commande :


```
make install
```

Cela installera Timidity dans le répertoire `/usr/bin/`.

La deuxième étape est bien entendu d'installer les fichiers de patches. Timidity utilise des patches pour les cartes son Gravis Ultrasound. Vous pourrez trouver des archives contenant de tels patches sur Internet très facilement. En particulier, la collection de patches [EAWPATS](#) réalisée par Eric A. Welsh est très complète. Cette collection a de plus l'avantage d'être préconfigurée pour l'emploi avec Timidity.

Lorsque vous aurez installé les fichiers de patches, vous devrez indiquer à Timidity leur emplacement dans le système de fichiers. Cette information doit être stockée dans le fichier de configuration `timidity.cfg` du répertoire `/usr/share/timidity/`. Vous pourrez vous inspirer du fichier fourni avec la collection de patches EAWPATS. Cependant, vous aurez à modifier les chemins indiqués dans ce fichier, car il a été écrit pour la version Windows de Timidity. Le premier répertoire à indiquer est celui des bibliothèques. Il faut ici simplement remplacer la ligne « `dir c:\timidity` » par la ligne « `dir /usr/lib/timidity` ». Notez que cette ligne est facultative et peut être commentée si vous n'avez pas déplacé ce répertoire. Le deuxième répertoire est plus important, puisque c'est le répertoire d'installation des patches. Il s'agit cette fois de remplacer la ligne « `dir c:\leawpats` » par la ligne « `dir repertoire` », où `repertoire` est le répertoire où vous avez extrait les fichiers patches.

Une fois que vous aurez terminé ces modifications, vous pourrez lire un fichier MIDI simplement en utilisant la ligne de commande suivante :

```
timidity fichier
```

où `fichier` est le nom du fichier MIDI à lire. Si vous désirez utiliser l'interface graphique de Timidity, vous n'avez qu'à ajouter l'option `-ig` à la ligne de commande précédente. Enfin, si vous désirez utiliser Timidity en tant que séquenceur système ALSA et permettre le mixage du flux audio qu'il produit avec les flux des autres applications par l'intermédiaire du serveur de son aRts de l'environnement de développement de KDE, vous devrez utiliser les options `-iA -OR -B2,8 -EFverb=0`. Les programmes MIDI devront ensuite se connecter sur le port MIDI 128:0 d'ALSA.

L'installation de KMidi ne pose pas de problème particulier, puisqu'il est fourni avec KDE et que KDE est normalement inclus dans toute bonne distribution. La seule opération que vous ayez à faire est simplement de modifier le fichier de configuration de KMidi pour lui indiquer l'emplacement des fichiers de patches. Or comme KMidi est basé sur les sources de Timidity, il utilise le même fichier de configuration `timidity.cfg` que Timidity. Ce fichier est normalement situé dans le répertoire `/base/kde/share/apps/kmidi/config/`, où `base` représente le répertoire de base d'installation de KDE. Vous n'aurez donc qu'à répéter les opérations faites sur le fichier de configuration de Timidity avec le fichier de configuration de KMidi.

VIII-C-3 - Installation d'une carte graphique 3D

De nos jours, toutes les cartes graphiques disposent d'une accélération matérielle non seulement pour les opérations graphiques en 2D, mais également pour le calcul de scènes en 3D. Ce développement des cartes 3D a été principalement poussé pour les jeux, mais ces fonctionnalités sont également très utiles pour les applications de modélisation et de conception.

La manière dont les opérations graphiques en 3D sont réalisées dépend évidemment du modèle de carte graphique utilisée, car il n'existe pas de standard au niveau matériel. Comme il est impensable que les applications qui désirent utiliser les fonctionnalités 3D des ordinateurs modernes aient à prendre en compte le type de carte utilisée, des interfaces logicielles standards ont été définies.

De toutes ces interfaces, Linux n'en gère qu'une seule : OpenGL. Cette interface a été définie par Silicon Graphics pour ses stations de travail haut de gamme et s'est imposée comme étant la norme en la matière pour ce qui est de la 3D. Cette interface a l'avantage d'être ouverte et disponible sur la plupart des systèmes, qu'ils soient de type Unix ou non.

La prise en charge de l'interface OpenGL est réalisée à deux niveaux sous Linux. La partie principale réside dans le serveur X qui, comme nous l'avons déjà signalé, n'est rien d'autre que le programme fournissant les services graphiques aux autres applications sous Unix. Cette partie est implémentée par une extension de l'interface de programmation XWindow, que l'on nomme tout simplement GLX. Cependant, le serveur X ne peut pas tout faire à lui tout seul, car, pour des raisons de performances, il lui faut accéder directement au matériel lors du rendu de scènes 3D. Il s'appuie donc pour cela sur un module du noyau, dont le but est de contrôler les accès aux ressources matérielles et de garantir ainsi la stabilité globale du système.

La configuration des fonctionnalités 3D des cartes graphiques sous Linux nécessite donc d'intervenir à la fois dans le noyau et au niveau du serveur X. Pour ce qui est du noyau, il faut tout d'abord s'assurer que les fonctionnalités d'accès direct au matériel sont bien supportées. Ces fonctionnalités sont couramment appelées « DRI », ce qui est l'abréviation de l'anglais « Direct Rendering Infrastructure ». Pour activer les fonctionnalités DRI, vous devrez, dans la configuration du noyau, valider l'option « Direct Rendering Manager » du menu « Character devices », ainsi que le type de carte graphique utilisée (3Dfx, 3dlabs, ATI Rage 128 ou ATI Radeon, chipset i810 ou Matrox G200/G400).

Note : Les fonctionnalités 3D des cartes graphiques basées sur les puces NVidia ne sont pas supportées directement par X.org et par le noyau. En revanche, NVidia fournit un pilote pour Linux pour ces cartes, qui, bien qu'il n'utilise pas DRI, dispose également d'un module du noyau et d'un module pour X.org.

Quelle que soit votre carte graphique, vous aurez également sans doute intérêt à activer le support du bus AGP dans le noyau, si bien sûr votre carte graphique est une carte AGP. Pour cela, il suffit d'activer l'option « /dev/agpgart (AGP support) » dans le menu « Character devices » de la configuration du noyau, ainsi que le type de chipset utilisé par votre carte mère.

Une fois la configuration du noyau faite, vous pourrez le recompiler et l'installer. La manière de procéder a été décrite en détail dans la [Section 7.3](#).

Vous devrez également vous assurer que le fichier spécial de périphérique `/dev/agpgart` est bien présent dans le répertoire `/dev/`. Son code majeur est 10, et son code mineur est 175. De même, si vous avez compilé les options précédentes sous la forme de modules du noyau, assurez-vous qu'ils sont bien référencés dans le fichier `modprobe.conf`.

La suite des opérations se passe alors au niveau de la configuration du serveur X de X.org. L'activation du support de l'AGP et d'OpenGL se fait simplement en rajoutant deux options dans le fichier de configuration `/etc/X11/xorg.conf`. Vous devez trouver la section « Modules » et lui ajouter les deux lignes suivantes :

```
Section "Module"
.
.
.
Load "dri"
Load "glx"
.
.
.
EndSection
```

Vous trouverez de plus amples renseignements sur la manière de procéder dans le [Chapitre 10](#).

Note : Pour le pilote fourni par NVidia pour X.org, il n'est pas nécessaire de demander le chargement du module DRI, car il ne l'utilise pas.

Il est supposé ici que le serveur X utilisé correspond bien à la carte graphique et dispose des fonctionnalités 3D. Si ce n'est pas le cas, vous devrez sans doute réinstaller X.org.

Les programmes utilisant OpenGL utilisent souvent une bibliothèque complémentaire nommée GLUT. Cette bibliothèque est fournie avec la couche d'émulation logicielle d'OpenGL MESA. Toutefois, la bibliothèque GLUT n'est

disponible que dans les programmes d'exemples de MESA. Vous devrez donc réinstaller MESA complètement si votre distribution ne fournit pas la bibliothèque GLUT.

VIII-C-4 - Installation d'une carte d'acquisition vidéo

Linux fournit toutes les fonctionnalités nécessaires à la manipulation des flux de données vidéo par l'intermédiaire d'une interface de programmation nommée `video4linux` (ce qui se lit « Video for Linux »). Linux est capable de gérer la plupart des cartes d'acquisition TV du marché et quelques-unes des cartes d'acquisition vidéo. Comme d'habitude, seuls les constructeurs de matériel qui ont accepté de jouer le jeu de fournir les informations nécessaires à la programmation de gestionnaires de périphériques libres voient leur matériel supporté sous Linux. Par conséquent, il est encore une fois nécessaire de bien se renseigner sur la nature du produit et la politique du fabricant lorsqu'on achète une carte d'acquisition TV.

En pratique, quasiment toutes les cartes d'acquisition TV utilisent les circuits intégrés Bt848 ou un de leurs dérivés, et Linux sait les gérer sans problèmes. Les cartes d'acquisition et de montage vidéo utilisent d'autres circuits plus puissants, dont les spécifications ne sont généralement pas disponibles. Seule la configuration des cartes TV sera donc décrite ci-dessous.

Les applications, pour accéder aux périphériques vidéo, utilisent l'un des fichiers spéciaux de périphérique `/dev/video*`. Ces fichiers sont des fichiers de type caractère, dont le code majeur vaut 81. Le code mineur est utilisé pour différencier les différents périphériques installés sur votre système. En général, il existe un lien symbolique `/dev/video` sur l'un de ces fichiers spéciaux, qui sera utilisé pour accéder au périphérique vidéo par défaut. Normalement, tous ces fichiers sont installés d'office dans le répertoire `/dev/` soit par les distributions, soit par `udev` au démarrage du système, et vous n'aurez donc généralement pas à les créer vous-même.

Le support de la vidéo sous Linux passe bien entendu par la configuration du noyau. Cette fois, il n'est pas certain du tout que le noyau fourni avec votre distribution supporte les cartes d'acquisition TV, aussi aurez-vous peut-être à recompiler vous-même votre noyau. La manière de procéder a été décrite en détail dans la [Section 7.3](#).

Sachez toutefois que les options à valider dans la configuration du noyau se trouvent dans le menu « Multimedia devices ». Vous devrez activer l'option « Video For Linux ». Si vous compilez cette fonctionnalité sous la forme de module, celui-ci sera nommé `videodev`. Ce pilote se chargera de répondre aux requêtes du client sur les fichiers spéciaux de périphérique `/dev/video*`. Lorsque vous aurez activé la fonctionnalité de vidéo pour Linux, vous aurez le choix des pilotes de cartes vidéo dans le sous-menu « Video adapters ». Il est recommandé de compiler ces gestionnaires de périphériques sous la forme de modules, car il vous faudra sans doute spécifier des options afin d'adapter ces gestionnaires à votre matériel.

Note : La configuration des cartes basées sur la puce Bt848 (option de menu « BT848 Video For Linux ») n'est accessible que si vous avez également activé l'option « I2C bit-banging interfaces » du menu « I2C support » (lui-même situé dans le menu « Character devices »).

De plus, le pilote prenant en charge les puces du type Bt848 ne prend pas en charge la gestion du son. Ces puces n'intègrent en effet pas de gestion du son, et nécessitent de brancher la sortie audio de la carte TV sur l'entrée ligne d'une carte son. Toutefois, un mixeur est disponible. Celui-ci est pris en charge automatiquement pour les pilotes de son ALSA. Pour les pilotes OSS en revanche, un gestionnaire spécifique devra être activé. L'option correspondante est l'option « TV card (bt848) mixer support ». Pour les puces Bt878, qui sont plus récentes, une carte son intégrée est fournie, et il existe un pilote complet ALSA et OSS, que l'on devra également activer.

Une fois le noyau recompilé et installé, il faut modifier les paramètres adaptés à votre matériel. En effet, bien que Linux soit parfaitement capable de déterminer les ressources requises par les cartes vidéo, il est rare que le matériel soit correctement identifié par les gestionnaires de périphériques. Cela est dû au fait que les gestionnaires se basent plus sur les composants courants permettant de faire l'acquisition vidéo que sur les modèles de cartes de chaque fabricant. La palme revient sans doute au gestionnaire pour les cartes basées sur les puces Bt848 et ses dérivées, puisqu'il est capable de faire fonctionner toutes les cartes vidéo de tous les fabricants qui utilisent cette puce. Par conséquent, il faut indiquer le modèle de la carte au gestionnaire, et bien souvent le modèle de tuner doit également être spécifié.

Si vous avez compilé les gestionnaires de périphériques sous forme de module, vous pourrez spécifier le type de carte et le type de tuner simplement dans le fichier de configuration `/etc/modprobe.conf`, à l'aide des options du module du gestionnaire de périphérique adéquat. Pour les cartes basées sur la puce Bt848, le gestionnaire de périphérique est pris en charge par le module `bttv`. Le type de carte peut lui être communiqué à l'aide de l'option `card`. Cette option peut prendre comme valeur un numéro identifiant le modèle de la carte. Les valeurs actuellement supportées sont indiquées dans le fichier `CARDLIST.bttv` du répertoire `/usr/src/linux/Documentation/video4linux/bttv/`. Le type de tuner quant à lui doit être spécifié à l'aide de l'option `tuner`. Cette option prend, elle aussi, une valeur numérique indiquant la nature du tuner utilisé. Les valeurs supportées sont données dans le fichier `CARDLIST.tuner`. En pratique, il est fort probable que vous utilisiez le type de tuner 3, qui correspond au tuner Philips SECAM, car la France utilise le standard SECAM pour les émissions TV.

Ainsi, si vous disposez d'une carte MIRO PCTV (carte de type 1) basée sur le tuner Philips SECAM, vous devriez avoir les lignes suivantes dans votre fichier `modprobe.conf` :

```
# Nom du module de gestion de Video For Linux :
alias char-major-81 videodev

# Nom du module à utiliser pour le premier périphérique vidéo :
alias char-major-81-0 bttv

# Options du module de gestion du Bt848 :
options bttv card=1 tuner=3
```

Vous devrez bien entendu adapter ces lignes selon votre propre configuration.

Si vous ne désirez pas utiliser les modules du noyau, vous devrez fournir ces paramètres au gestionnaire de périphérique via la ligne de commande du noyau. Les options à utiliser peuvent alors être déduites des options du module, comme on l'a expliqué dans la [Section 8.1.1.3](#). Pour notre exemple, ces paramètres seraient les suivants :

```
bttv.card=1 bttv.tuner=3
```

Si vous avez compilé les fonctionnalités de l'interface I2C sous forme de module (option de menu « I2C bit-banging interfaces »), vous aurez également à ajouter ces lignes dans votre fichier de configuration `modprobe.conf` :

```
# Nom du module de gestion des fonctionnalités I2C :
alias char-major-89 i2c-dev

# Activation de l'agorithme bit-banging :
options i2c-algo-bit bit_test=1
```

Pour information, I2C est un protocole de communication entre microcontrôleurs, que la plupart des cartes mères sont capables de gérer. Cette fonctionnalité n'est nécessaire que pour les cartes basées sur la puce Bt848.

Enfin, si vous utilisez les pilotes OSS pour la gestion du mixer de votre carte d'acquisition vidéo (ce qui est nécessaire pour les cartes vidéo basées sur les puces Bt848, parce qu'ALSA ne fournit pas de pilote pour ces cartes à l'heure actuelle), vous devrez définir les alias nécessaires au chargement des modules de cette carte lors de l'accès aux périphériques audio. Par exemple, pour une carte basée sur la puce Bt848, les alias suivants doivent être ajoutés dans le fichier de configuration `modprobe.conf` :

```
# Charge le pilote de la carte vidéo en tant que deuxième carte son :
alias sound-slot-1 bttv

# Charge le mixer de la carte son de cette carte :
alias sound-service-1-0 tvmixer
```

Note : Vous pouvez rencontrer quelques problèmes lors de la configuration de votre carte TV. Généralement, si vous n'obtenez aucune image, c'est que vous vous êtes trompé de tuner. Revoyez dans ce cas l'option type du module tuner. Si vous obtenez bien une image, mais pas de son, c'est sans doute que vous

vous êtes trompé dans le type de carte, ou que vous avez oublié d'inclure le support du son pour les cartes à base de Bt848. On notera que, comme pour les cartes son, seule la compatibilité matérielle importe. Par exemple, les cartes Studio PCTV de Pinnacle vendues en France sont en réalité des cartes Miro PCTV et ne sont pas reconnues comme des cartes Studio PCTV par Linux... Si vous avez des problèmes de son, vous devrez donc revoir la configuration du noyau ou modifier la valeur passée à l'option `card` du module `bttv`. Dans tous les cas, n'hésitez pas à utiliser la commande `lspci`, qui permet de visualiser les informations du bus PCI, et la commande `dmesg`, qui permet de voir la liste des messages du noyau.

Vous pouvez également avoir quelques problèmes de droits sur les fichiers spéciaux de périphériques `/dev/videoX`. Le symptôme classique est dans ce cas que tout fonctionne parfaitement sous le compte `root`, mais pas sous les comptes utilisateurs. Dans ce cas, on pourra résoudre ces problèmes en créant un groupe d'utilisateurs `video`, auquel appartiendront tous les utilisateurs ayant le droit d'utiliser la carte d'acquisition TV, et de changer le groupe des fichiers spéciaux de périphériques `/dev/videoX`.

Enfin, l'utilisation des cartes d'acquisition TV nécessite d'activer les fonctionnalités DGA ou Xv du serveur X. Ces fonctionnalités permettent aux programmes d'accéder directement à la mémoire vidéo, et donc de faire l'incrustation de l'image décodée par la carte TV dans la surface d'affichage d'un écran. La manière d'activer les fonctionnalités DGA et Xv sera précisée dans le chapitre traitant de la configuration du serveur X.

VIII-C-5 - Configuration des cartes réseau

Le support des cartes réseau est très complet sous Linux et la prise en charge d'une nouvelle carte revient souvent simplement à ajouter le gestionnaire de périphériques de cette carte dans la configuration du noyau. La plupart des cartes réseau sont des cartes Ethernet compatibles soit avec les cartes NE2000, soit avec les cartes 3COM, aussi suffit-il en général d'activer le gestionnaire de périphériques pour l'une de ces cartes pour qu'elle soit utilisable.

La configuration des gestionnaires de périphériques se fait dans le menu « Networking support » de la configuration du noyau. Il faut activer l'option « Network device support » en premier lieu, puis sélectionner le gestionnaire de périphériques approprié dans le sous-menu « Ethernet (10 or 100Mbit) ». Ce menu présente en premier lieu les gestionnaires de périphériques pour les cartes ISA, et regroupe les gestionnaires de toutes les autres cartes dans les sous-options de l'option « EISA, VLB, PCI and on board controllers ». Vous aurez donc certainement à activer cette option et la sous-option correspondante au gestionnaire de périphériques de votre carte réseau. Vous pouvez activer ces options sous la forme de module du noyau ou non, le plus simple étant ici de ne pas utiliser les modules afin d'éviter de compléter le fichier de configuration `/etc/modprobe.conf`.

La compilation du noyau elle-même est décrite en détail dans la [Section 7.3](#) et ne sera donc pas reprise plus en détail ici. Remarquez également que la prise en charge de la carte réseau par un gestionnaire de périphériques ne suffit en aucun cas à l'utilisation de l'ordinateur en réseau. En effet, il faut également configurer l'ensemble des paramètres réseau du système, ce qui sera décrit dans le [Chapitre 9](#). Le nombre des fonctionnalités réseau de Linux interdit de les décrire toutes ici, veuillez vous référer à ce chapitre pour plus de détails.

VIII-C-6 - Configuration des adaptateurs Wi-Fi

Le Wi-Fi (abréviation de l'anglais « Wireless Fidelity ») est une technologie permettant de mettre plusieurs ordinateurs en réseau par ondes radio. La plage de fréquences utilisée est située dans la gamme des micro-ondes (2.44 GHz) et est soumise à une réglementation stricte dans de nombreux pays, car elle est en partie réservée aux applications militaires. En pratique, seuls certains canaux sont donc utilisables, et des dérogations peuvent être nécessaires pour une utilisation en extérieur. De plus, cette plage de fréquences est la même que celle utilisée par de nombreux autres appareils sans fil, comme les téléphones portables. Les communications sont donc fortement perturbables par ces appareils, ainsi bien sûr que par les fours à micro-ondes. Enfin, du fait même que ce sont des ondes radio, la confidentialité et la sécurité des communications peuvent être compromises.

Note : Comme pour toute onde proche des 2.4 GHz, l'influence des ondes radio Wi-Fi sur les corps organiques humides n'est pas neutre (la problématique se pose bien entendu également pour les téléphones portables). En effet, les effets sont strictement les mêmes que ceux des ondes des fours à micro-ondes : ces ondes sont à la fréquence propre des molécules d'eau, qui entrent donc en résonance en leur présence (cette interaction est due à la polarité de la molécule H₂O, elle-même due à son asymétrie - les atomes d'hydrogène et d'oxygène ne sont pas en ligne - et à la différence d'électronégativité entre les atomes d'hydrogène et d'oxygène qui la constituent). Les molécules d'eau restituent ensuite l'énergie de l'onde par diffusion de leur agitation moléculaire dans les molécules avoisinantes, ce qui provoque une augmentation de température. On est donc en droit de se poser la question du danger de l'utilisation de ces ondes pour l'organisme. En général, les puissances émises restent très faibles, et leur influence décroît de manière inversement proportionnelle au carré de la distance à l'émetteur, ce qui fait que le danger est relativement réduit. Toutefois, à titre personnel, et sans prétendre avoir fait la moindre étude sur le sujet, je vous invite à la prudence lors de l'utilisation d'ordinateurs ou de téléphones portables, en raison même de leur proximité (note : je n'ai pas de téléphone portable). Notez également que la chaleur directe dégagée par ces appareils par effet Joule est certainement la plus nocive, de même que leur utilisation prolongée pour l'équilibre mental...

Plusieurs normes de transmission ont été définies pour spécifier les interactions entre les appareils Wi-Fi, ainsi que les technologies de chiffrement utilisées. La norme la plus répandue est la norme 802.11b, qui permet d'atteindre un débit de 11 Mbits/s théorique sur des distances allant jusqu'à 300 mètres en extérieur (quelques dizaines de mètres en intérieur). Toutefois, la plupart des adaptateurs Wi-Fi implémentent également la norme 802.11g à présent. Cette norme permet en effet une compatibilité ascendante avec les équipements 802.11b d'une part (elle utilise la même plage de fréquences), et en constitue une amélioration conséquente puisque le débit théorique est relevé à 54 Mbits/s d'autre part (30 Mbits/s réels). Notez qu'une autre norme concurrente du 802.11g a été définie, mais, n'étant pas compatible avec les équipements existants, elle n'a pas eu autant de succès. Il s'agit de la norme 802.11a, qui utilise une plage de fréquence à 5 GHz. Les problèmes d'interférences sont donc moindres, mais la portée est également réduite (plus la fréquence d'une onde radio est élevée, plus sa portée est réduite et plus elle est susceptible d'être bloquée par des objets).

Ces normes Wi-Fi définissaient également un mécanisme de sécurité primitif, qui peut être considéré comme absolument inefficace de nos jours (il s'agit du WEP). En particulier, les algorithmes utilisés n'étaient pas fiables du tout et peuvent être cassés simplement par écoute du trafic pendant une période de temps relativement courte. De ce fait, de nouveaux mécanismes de sécurités ont été définis (notamment le WPA et le WPA2). Enfin, la norme 802.11i, encore peu répandue, a pour but d'améliorer la sécurité des communications en utilisant un véritable procédé cryptographique. Cette norme peut être transposée pour les appareils 802.11g et 802.11a.

Comme vous pouvez le constater, le terme Wi-Fi cache bien des choses complexes, et le choix d'une technologie sur une autre peut être assez difficile.

Malgré tous ces inconvénients, la technologie Wi-Fi est très pratique, car elle permet de connecter simplement des machines sans s'emmêler les pattes dans les câbles réseau. La plupart des ordinateurs portables vendus disposent maintenant d'un adaptateur réseau sans fil Wi-Fi, et il est très facile d'en connecter un à un ordinateur de bureau, soit avec une carte fille, soit grâce à un adaptateur USB (il existe même des clefs USB dont l'encombrement est réduit à celui d'une gomme). Il est donc certain que ces technologies sont appelées à se développer et à se généraliser.

Certains matériels Wi-Fi ne sont pas supportés nativement par Linux, soit parce que les constructeurs ne désirent pas fournir les informations permettant aux développeurs de faire un gestionnaire de périphérique libre, soit parce que les constructeurs n'ont eux-mêmes pas le droit de divulguer la technologie utilisée dans leurs appareils ou parce que les réglementations locales sur les émissions radio leur imposent de tenir secret les caractéristiques de leurs appareils. Le choix d'un périphérique Wi-Fi doit donc se faire avec précaution. Vous pourrez consulter le [site Web](#) de Jean Tourrilhes pour savoir le matériel supporté par les différents pilotes disponibles.

La prise en charge des périphériques Wi-Fi nécessite d'activer le support du Wi-Fi dans le noyau lui-même. Cela peut être fait en activant l'option « **Wireless LAN drivers (non-hamradio) & Wireless Extensions** » du menu « Network device support » du noyau. Cette option vous donnera également accès aux options de configuration des pilotes de périphériques Wi-Fi (sauf pour certains périphériques USB, qui sont accessibles dans les options de configuration des périphériques USB). Certains pilotes ne sont pas encore intégrés au noyau, et vous devrez peut-être chercher sur Internet le pilote correspondant à votre matériel. Plusieurs projets ont en effet été créés afin de prendre en charge

le Wi-Fi sous Linux. Vous trouverez également sur le site de Jean Tourrilhes des liens vers les projets des pilotes pour les autres puces Wi-Fi.

Si aucun pilote Linux n'est disponible pour votre matériel, vous pourrez vous rabattre sur le projet NDISWrapper (<http://ndiswrapper.sourceforge.net>). Ce projet a pour but de simuler la couche réseau NDIS de Windows XP afin de pouvoir charger les gestionnaires de périphériques pour Windows fournis par les fabricants, et les duper en leur faisant croire qu'ils s'exécutent bien sous Windows. Cette technique reste toutefois une magouille et ne résout bien entendu pas le problème de la disponibilité de pilotes libres.

L'installation des pilotes ne sera pas détaillée plus ici, car la manière de procéder est spécifique à chaque matériel. Dans le cas de NDISWrapper, l'installation se fait simplement en installant NDISWrapper lui-même et en exécutant la commande suivante :

```
ndiswrapper -i driver.inf
```

où driver.inf est le fichier .inf du gestionnaire de périphérique pour Windows de votre matériel. Cette commande copie les fichiers nécessaires de ce gestionnaire de périphérique dans le répertoire `/etc/ndiswrapper/` et définit la configuration pour l'utilisation de ce périphérique. Une fois cela fait, le chargement du module ndiswrapper suffit pour charger le gestionnaire de périphérique :

```
modprobe ndiswrapper
```

Pour les autres gestionnaires de périphériques, il suffit généralement également de charger le module du noyau du pilote, et éventuellement de charger un firmware.

Lorsque le pilote est chargé, une nouvelle interface réseau « wlanX » apparaît dans le système, où X est le numéro de l'interface réseau sans fil. Cette interface peut ensuite être utilisée comme n'importe quelle interface réseau, à ceci près qu'il faut au préalable configurer les paramètres du réseau sans fil. En particulier, il est nécessaire de spécifier le mode de fonctionnement, le canal à utiliser et l'identifiant du réseau sans fil. Une clef de chiffrement des communications peut également être donnée (cela ne garantit toutefois pas une sécurité absolue, sauf en 802.11i).

Les réseaux Wi-Fi peuvent être établis de plusieurs manières. S'il s'agit de relier deux machines par Wi-Fi, le mode de fonctionnement utilisé le plus simple est le mode « ad hoc ». Dans ce mode, les deux machines peuvent communiquer directement, sans avoir recours à une infrastructure quelconque. Inversement, si plusieurs machines doivent être reliées à un réseau existant (ou simplement entre elles), il est nécessaire d'installer ce qu'on appelle un « point d'accès », qui synchronisera les communications entre toutes ces machines. Dans ce cas, le mode de fonctionnement utilisé est le mode « managé ».

Le canal permet de spécifier la fréquence utilisée par les périphériques sans fil. Bien entendu, toutes les machines devant participer à la communication doivent utiliser le même canal. Le canal est donc un des paramètres essentiels du réseau. Il existe plusieurs canaux, mais en pratique, seuls quelques-uns peuvent être utilisés librement en France. Il s'agit des canaux 10 (2457 MHz), 11 (2462 MHz), 12 (2467 MHz) et 13 (2472 MHz).

Comme on peut être amené à créer plusieurs réseaux Wi-Fi utilisant un même canal, il est nécessaire de pouvoir les distinguer afin d'éviter de mélanger les transmissions. Deux possibilités s'offrent alors à l'utilisateur : soit on utilise ce qu'on appelle un numéro de réseau virtuel (« Network ID »), soit on utilise un nom de réseau. La première solution ne permet de définir qu'une cellule du réseau, alors que la deuxième permet de regrouper plusieurs de ces cellules et autorise un utilisateur à passer d'une de ces cellules à l'autre (une cellule est une zone géographique prise en charge par un point d'accès).

Tous ces paramètres peuvent être spécifiés avec la commande iwconfig, sauf pour certains pilotes de périphériques non standards qui nécessitent des commandes spécifiques. La syntaxe générale de cette commande est la suivante :

```
iwconfig interface options
```

où `interface` est le nom de l'interface réseau sans fil (par exemple `wlan0`), et `options` est la liste des options permettant de définir les paramètres du réseau sans fil. Les principales options sont récapitulées dans le tableau suivant :

Option	Signification
<code>channel</code>	Numéro du canal.
<code>mode</code>	Mode de fonctionnement.
<code>ssid</code>	Nom de réseau.
<code>nwid</code>	Numéro de réseau.
<code>key</code>	Clef de chiffrement WEP du réseau.

Les modes de fonctionnement les plus courants sont le mode « `ad hoc` » pour le mode ad hoc, et le mode « `managed` » pour l'utilisation d'un point d'accès. La page de manuel de `iwconfig` vous donnera la liste des autres modes utilisables.

Par exemple, pour configurer une interface sans fil sur le canal 11, en mode managé et avec un nom de réseau « `MonWLAN` », il faudra utiliser la commande suivante :

```
iwconfig wlan0 mode managed ssid MonWLAN channel 11
```

À la suite de cette commande, l'interface `wlan0` sera utilisable comme n'importe quelle interface réseau.

Si vous désirez définir une clef de chiffrement pour votre réseau sans fil, vous devrez utiliser l'option `key`. L'utilisation du chiffrement peut être facultative en Wi-Fi, sauf si l'interface est configurée pour n'accepter que les communications chiffrées. L'option `key` peut donc accepter, en plus de la clef de chiffrement (dont la taille est de 64 ou 104 bits, soit 8 ou 13 octets), un paramètre indiquant si le chiffrement est obligatoire ou non. Ce paramètre peut prendre la valeur `open` si le chiffrement est facultatif, `restricted` s'il est obligatoire, ou tout simplement `off` si le chiffrement doit être désactivé.

Par exemple, pour imposer l'utilisation d'une clef de chiffrement sur l'interface `wlan0`, la commande suivante devra être utilisée :

```
iwconfig wlan0 key restricted 74151B6DCD98D808
```

Comme vous pouvez le constater, la clef est fournie en hexadécimal (c'est-à-dire en base 16, les chiffres au-dessus de 9 étant représentés par les lettres A à F). Pour sortir du mode chiffré, la commande suivante pourra être utilisée :

```
iwconfig wlan0 key off
```

Sachez toutefois que le chiffrement WEP n'est pas fiable du tout, et qu'il ne faut pas se reposer dessus. Le chiffrement WPA, bien que n'étant pas encore supporté par tous les adaptateurs et tous les pilotes, est donc à privilégier. Outre l'utilisation d'algorithmes de chiffrement plus évolués, le chiffrement WPA permet de renégocier régulièrement les clefs de chiffrement utilisées par les interlocuteurs.

Sous Linux, cette négociation périodique est à la charge du démon `wpa_supplicant`. Ce démon prend en charge tous les aspects de la liaison Wi-Fi, et rend par conséquent obsolète l'utilisation de la commande `iwconfig` pour la configuration du réseau sans fil. En effet, cette configuration doit se faire dans le fichier de configuration `/etc/wpa_supplicant.conf`, dont vous trouverez un exemple ci-dessous :

```
ctrl_interface=/var/run/wpa_supplicant
network={
    ssid="MonRezo"
    key_mgmt=WPA-PSK
    psk=1cc5a2db98bfd1387de265397772b9ea36e4d049e17fd1d660c014576d5e71a4
    proto=WPA
}
```

L'option `ctrl_interface` permet de spécifier un répertoire dans lequel `wpa_supplicant` placera des fichiers servant de canaux de communication avec des programmes complémentaires permettant de le contrôler et d'obtenir des

informations sur l'état de la liaison Wi-Fi. La définition des paramètres des réseaux sans fil se fait ensuite dans une section network, comme vous pouvez le constater.

Le nom du réseau est obligatoire. Il doit être défini avec l'option `ssid`. Le protocole de gestion de clef utilisé peut ensuite être spécifié avec l'option `key_mgmt`. Dans cet exemple, il a été choisi d'utiliser une clef secrète partagée (« PSK », abréviation de « Private Shared Key ») par toutes les machines connectées au réseau sans fil. Cette configuration est très classique, et convient en particulier pour tous les modems routeurs sans fil que l'on peut trouver dans le commerce.

La clef elle-même doit être spécifiée avec l'option `psk`. Cette option peut prendre une chaîne de caractères ASCII entre guillemets (exactement comme on a donné le nom du réseau à l'option `ssid`), ou une valeur hexadécimale. La commande `wpa_password` permettra de créer une telle valeur à partir d'une chaîne de caractère et du nom du réseau.

Enfin, l'option `proto` permet d'indiquer le protocole de chiffrement utilisé. Dans le cas présent, nous avons utilisé le protocole WPA, qui renégocie les clefs régulièrement. Le protocole WPA2, intrinsèquement plus sûr, aurait pu être utilisé, mais n'a pas été choisi parce que les points d'accès les plus anciens ne savent pas gérer ce protocole.

Note : Il est possible de définir plusieurs réseaux sans fil dans le fichier de configuration, dans ce cas, `wpa_supplicant` utilisera le premier réseau défini dans le fichier, en privilégiant les réseaux utilisant les protocoles de chiffrement les plus sûrs.

Cet exemple de fichier de configuration ne présente que le cas d'un réseau sans fil classique, dont les différentes machines partagent une même clef secrète de chiffrement. Sachez toutefois que `wpa_supplicant` sait gérer des configurations plus complètes, dans lesquels des mécanismes de chiffrement plus évolués et des serveurs d'authentification peuvent être utilisés par exemple. Ces configurations ne seront pas abordées ici.

Une fois le fichier de configuration des réseaux sans fil mis en place, il ne reste plus qu'à lancer le démon `wpa_supplicant`. Typiquement, la commande à utiliser est la suivante :

```
wpa_supplicant -Bw -c/etc/wpa_supplicant.conf -Dwext -iwlan0
```

Les options `B` et `w` permettent respectivement de faire passer le processus ainsi lancé en arrière-plan, et de lui demander d'attendre que l'interface réseau à utiliser soit créée, si elle ne l'est pas encore lors de son lancement. Le nom de cette interface est spécifié avec l'option `i`, et l'interface de programmation à utilisée est spécifiée avec l'option `D`. Dans le cas présent, l'interface de programmation utilisée est l'interface des extensions sans fil classique de Linux (« wext », abréviation de « Wireless Extensions »). Enfin, l'option `c` permet de spécifier le fichier de configuration à utiliser. Il s'agit ici du fichier que l'on vient de décrire dans les paragraphes précédents.

Sachez enfin que la liste des réseaux sans fils disponibles peut être récupérée avec l'option `scan` de la commande `iwlist`. Par exemple, la commande suivante vous affichera tous les réseaux sans fils disponibles via l'interface réseau `wlan0`, ainsi que leurs caractéristiques :

```
iwlist wlan0 scan
```

La commande `iwlist` accepte d'autres options, dont vous trouverez la liste et la description dans sa page de manuel.

Pour finir, la commande `iwconfig`, utilisée sans option, affiche la configuration des interfaces sans fil du système. En particulier, vous y trouverez le mode de fonctionnement, le numéro de canal et les identifiants du réseau, éventuellement la clef de chiffrement utilisée, ainsi que la qualité du signal radio. La page de manuel de `iwconfig` vous donnera de plus amples informations sur la manière dont ces informations peuvent être interprétées.

VIII-D - Configuration des ports de communication

Les ports de communication sont les seuls moyens de communication de l'ordinateur avec d'autres appareils, et sont donc essentiels au même titre que les réseaux.

Historiquement, le plus vieux type de port de communication utilisé est sans doute le port série, qui permet de connecter une multitude de périphériques en raison de sa grande simplicité. Les ports parallèles sont également courants, et sont encore souvent utilisés pour connecter les imprimantes. Cependant, ces ports souffrent de défauts indéniables, dont l'un des principaux est sans doute le faible débit de données qu'ils autorisent. Ils ne peuvent donc pas être utilisés pour les applications les plus exigeantes. C'est pour cela que les ports USB (abréviation de l'anglais « Universal Serial Bus ») et IEEE1394 (alias « FireWire ») ont été inventés.

Cette section présente la manière de configurer les différents ports de communication actuellement utilisés, à savoir le port parallèle, les ports série standards, et enfin les ports USB et IEEE1394.

VIII-D-1 - Prise en charge des périphériques ISA standards

La plupart des PC disposent encore de périphériques d'entrée/sortie standards ISA intégrés aux cartes mères, comme les ports série et les ports parallèles, même s'ils ne permettent plus de brancher des cartes filles ISA. Il est donc encore nécessaire de prendre en charge ces fonctionnalités au niveau du noyau, même si vous ne disposez d'aucun périphérique ISA.

Pour cela, il suffit d'activer les options « Plug and Play support » et « Plug and Play BIOS support » du menu « Plug and Play support » de la configuration du noyau. Ces options permettent de prendre en charge les ressources Plug and Play des cartes mères, et donc de réaliser la configuration automatique des ports parallèles et série de l'ordinateur.

VIII-D-1-a - Configuration du port parallèle

La configuration du port parallèle sous Linux est relativement automatique. Il suffit en effet, dans le programme de configuration du noyau, d'activer les options « Parallel port support », « PC-style hardware », « Use FIFO/DMA if available » et « IEEE 1284 transfer modes » du menu « Parallel port support ». Si vous désirez connecter une imprimante sur votre port parallèle, ce qui est généralement le cas, il vous faudra également activer l'option « Parallel printer support » du menu « Character devices ». De même, si vous désirez connecter un disque dur, un lecteur de CD-ROM ou un lecteur de bande sur votre port parallèle, vous devrez activer l'option « Parallel port IDE device support » du menu « Block devices », ainsi qu'au moins une des sous-options pour le type de périphérique à utiliser et un des modules de prise en charge des protocoles de communication sur port parallèle.

Pour des raisons de compatibilité, le gestionnaire de ports parallèles utilise par défaut une interrogation périodique pour déterminer l'état des ports. Cela consomme généralement des ressources de calcul, mais c'est une technique fiable. Toutefois, si votre matériel le permet, il est possible de le configurer pour utiliser une interruption matérielle, et soulager ainsi le processeur de cette tâche. Nous allons donc voir comment configurer le port parallèle ci-dessous.

Afin d'utiliser une interruption matérielle pour le port parallèle, il est nécessaire de spécifier explicitement la ligne de cette interruption. Cela peut se faire avec l'option `irq` du module prenant en charge le port parallèle. Sur les ordinateurs de type PC, le nom de ce module est « `parport_pc` », mais, comme on l'a vu dans la [Section 8.1.1](#), le noyau demande toujours le chargement du module « `parport_lowlevel` » pour des raisons de portabilité. Il faut donc définir un alias dans le fichier de configuration `/etc/modprobe.conf`.

Outre la ligne d'interruption à utiliser, le module `parport_pc` peut prendre en paramètre une autre option permettant d'indiquer le port d'entrée / sortie qu'il doit utiliser pour accéder au port parallèle. Cette option est l'option `io`. En général, le port utilisé est le port numéro 378h et la ligne d'interruption est souvent la 7. Les options suivantes doivent donc être définies dans le fichier `modprobe.conf` pour obtenir un fonctionnement optimal du port parallèle sous Linux :

```
alias parport_lowlevel parport_pc
options parport_pc io=0x378 irq=7
```

D'autres options pourront être définies dans le fichier de configuration `modprobe.conf` pour prendre en charge les périphériques de type bloc connectés sur port parallèle. Cela est inutile pour les imprimantes connectées sur le port parallèle.

Si l'on désire intégrer le gestionnaire de périphérique du port parallèle dans le noyau et faire en sorte qu'il utilise les paramètres détectés automatiquement pour ce port, la solution la plus simple est de fournir l'option `parport` du noyau et de lui affecter la valeur `auto` :

```
parport=auto
```

Cela activera automatiquement l'utilisation de la ligne d'interruption du port parallèle. Les autres paramètres du port parallèle peuvent être trouvés dans le fichier `kernel-parameters.txt` de la documentation du noyau.

VIII-D-1-b - Configuration des ports série

Les ports série fournis en standard sur les cartes mères sont détectés et configurés automatiquement par le noyau lorsque l'ordinateur démarre. Pour que cela fonctionne, il suffit que l'option « Standard/generic (8250/16550 and compatible UARTs) serial support » du menu « Character devices » soit activée.

Cela dit, il se peut que votre ordinateur contienne un port série non standard, ou que vous ayez à sélectionner des options spécifiques pour votre matériel. Il vous faudra, dans ce cas, activer les options complémentaires relatives aux ports série dans le menu « Character devices ».

Le noyau n'est, en général, capable d'initialiser que les ports série standards intégrés dans les cartes mères (c'est-à-dire les ports COM1 à COM4), auxquels on accède, sous Linux, par l'intermédiaire des fichiers spéciaux de périphériques `/dev/ttyS0` à `/dev/ttyS3`. Pour les autres ports série, il peut être nécessaire de les initialiser manuellement, par exemple dans un fichier de configuration lancé au démarrage du système.

Cette initialisation se fait à l'aide de la commande `setserial`, dont la syntaxe simplifiée est la suivante :

```
setserial fichier [uart type] [port adresse] [irq ligne]
```

où `fichier` est le fichier spécial de périphérique permettant d'accéder au port série à initialiser, `type` est le type de port série utilisé, `adresse` est son adresse d'entrée / sortie et `ligne` est la ligne d'interruption qu'il utilise. La commande `setserial` dispose de nombreuses autres options. Je vous invite à en lire la page de manuel pour les découvrir.

Une fois le port série correctement initialisé, il est possible de fixer les paramètres de la ligne de communication avec la commande `stty`. Comme on l'a déjà vu précédemment, cette commande permet de fixer tous les paramètres des lignes de communication des terminaux et non seulement les paramètres des lignes série, et elle dispose donc d'un grand nombre d'options. Nous n'allons nous intéresser ici qu'aux options utilisées pour les lignes série. La page de manuel de `stty` pourra être consultée pour plus de détails.

La syntaxe à utiliser pour lire les paramètres de communication d'une ligne série est la suivante :

```
stty -a -F périphérique
```

où `périphérique` est le fichier spécial de périphérique de la ligne. Si vous exécutez cette commande, vous constaterez qu'un grand nombre d'informations est donné. Les informations les plus utiles sont sans doute `speed`, qui donne la vitesse de la ligne série, `csN`, qui donne le nombre `N` de bits de données par caractère, `[-]parenb`, qui indique si un bit de parité est utilisé ou non (auquel cas le caractère `'-'` est présent), `[-]parodd`, qui indique le type de parité utilisée (paire ou impaire selon la présence ou l'absence du caractère `'-'`), et `[-]cstopb`, qui indique le nombre de bits de stop utilisés (un ou deux selon la présence ou l'absence du caractère `'-'`). L'option `[-]crtscts` indique si le contrôle de flux matériel est utilisé ou non. À titre d'exemple, voici une sortie typique de la commande `stty` sur le premier port série :

```
speed 9600 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <indéfini>;
eol2 = <indéfini>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R;
werase = ^W; lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 hupcl -cstopb cread cllocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl
```

```
ixon -ixoff -iuclic -ixany -imaxbel opost -olcuc -ocrnl  
onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0  
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop  
-echopr echoctl echoke
```

Les mêmes options peuvent être utilisées pour fixer les paramètres de la ligne de communication. La ligne de commande à utiliser est alors la suivante :

```
stty -F périphérique options
```

Il existe une différence cependant : le paramètre permettant de fixer la vitesse de la ligne est décomposé en deux options `ispeed` pour le flux de données entrant et `ospeed` pour le flux de données sortant. On peut fixer la vitesse de la ligne avec l'option `ispeed`. Par exemple, pour faire passer la ligne de communication du deuxième port série à 115200 bauds, 7 bits de données, un bit de stop et une parité paire, il faut utiliser la ligne de commande suivante :

```
stty -F /dev/ttyS1 ispeed 115200 cs7 -cstopb parenb -parodd
```

Il est important de bien configurer les ports série avec les mêmes paramètres de ligne que ceux utilisés par les périphériques qui y sont connectés pour que la communication se fasse correctement.

VIII-D-2 - Installation des périphériques USB

Bien qu'ayant eu quelques difficultés à ses débuts, le port USB est désormais en pleine expansion. Il s'agit d'une extension des ports série classiques, qui offre les avantages suivants :

- possibilité de connecter jusqu'à 127 périphériques sur un même port, selon une structure arborescente ;
- bande passante accrue jusqu'à 12 Mbits/s théoriques pour l'USB 1.1, et de 480 Mbits/s pour l'USB 2.0 ;
- capacité de connexion des périphériques « à chaud » et détection automatique par le système d'exploitation ;
- possibilité d'alimentation des périphériques par le bus lui-même, évitant ainsi d'avoir des câbles supplémentaires pour les périphériques consommant peu d'électricité.

Tous ces avantages font que le bus USB est appelé à remplacer les ports série que l'on connaît, et que l'on n'utilise plus désormais que pour les modems, les vieilles souris série et quelques appareils extérieurs. Notez que la simplicité du port série fera qu'il restera encore présent sur bon nombre d'appareils pour plusieurs années encore, mais les périphériques informatiques risquent de s'en détourner de plus en plus.

La gestion de l'USB sous Linux se base profondément sur les mécanismes de détection dynamique des périphériques connectables à chaud. Les gestionnaires de périphériques sont donc généralement des modules du noyau chargés automatiquement lorsque ces périphériques sont détectés. Il est toutefois également possible d'intégrer ces gestionnaires au sein du noyau, et de laisser les opérations de configuration des périphériques (chargement du firmware, création des fichiers spéciaux de périphériques par `udev` et configuration du matériel) se faire via les mécanismes de gestion des périphériques amovibles. Linux est capable d'utiliser la plupart des périphériques USB existant actuellement sur le marché.

Au niveau du noyau, la configuration des périphériques USB se fait dans le menu « **USB support** ». Il faut simplement activer l'option « **Support for USB** », sélectionner un gestionnaire pour le port USB et choisir les gestionnaires des différents types de périphériques que l'on voudra connecter sur la machine.

Il est impératif d'activer l'option « **USB device filesystem** » afin de permettre à certains outils d'accéder, via le répertoire `/proc/bus/usb/` du système de fichiers virtuel `/proc/`, aux périphériques USB connectés à l'ordinateur. Ce système de fichiers doit être monté à l'aide de la commande suivante :

```
mount -t usbfs none /proc/bus/usb
```

pour pouvoir être utilisé. Il est également possible de le monter automatiquement dans le fichier `/dev/fstab`.

En fait, il existe trois types d'interfaces USB sur le marché : les interfaces EHCI (abréviation de l'anglais « Enhanced Host Controller Interface »), qui prennent en charge l'USB 2.0, les interfaces UHCI (abréviation de l'anglais « Universal Host Controller Interface »), spécifiées par Intel et que les contrôleurs de la plupart des cartes mères utilisent (chipsets Intel et VIA), et les interfaces OHCI (« Open Host Controller Interface »), spécifiées par Compaq, et qui sont utilisées par les chipsets Compaq et ALi principalement. Les ports USB 2.0 comprennent également un port USB 1.1 UHCI ou OHCI, qui permet donc de connecter les anciens périphériques USB. La configuration de l'USB se restreint donc simplement à la sélection des pilotes nécessaires pour votre matériel (option « EHCI HCD (USB 2.0) support » si vous disposez d'un port USB 2.0, et le pilote UHCI « UHCI HCD (most Intel and VIA) support » ou le pilote OHCI « OHCI HCD support »). Notez que le pilote EHCI ne gère que la partie USB 2.0 de votre contrôleur USB, et que vous devez également inclure le support UHCI ou OHCI pour pouvoir utiliser les périphériques USB 1.1.

Comme vous pouvez le constater dans le menu de configuration du noyau, un grand nombre de périphériques USB sont gérés par Linux. Vous devez bien entendu activer les options permettant de prendre en charge votre matériel. Il est conseillé d'inclure ces fonctionnalités sous forme de modules afin de permettre le chargement dynamique des gestionnaires de périphériques. Cela est nécessaire pour la configuration des périphériques connectés à chaud dans le système. Les options les plus importantes sont sans doute « USB Audio support », pour la prise en charge des périphériques USB audio, « USB Printer support », pour toutes les imprimantes USB et « USB Mass Storage support », pour la prise en charge des clefs USB et de la majorité des appareils photo numériques. Les scanners USB sont pris en charge automatiquement par les logiciels et ne requièrent pas de configuration particulière au niveau du noyau.

Les périphériques d'entrée tels que le clavier et la souris devront, si possible, être inclus directement au sein du noyau afin d'éviter de se retrouver sans clavier et sans souris au démarrage, en cas de problème dans la configuration du système. Notez qu'il existe deux types de pilotes pour les périphériques d'entrée : un pilote général (option « USB Human Interface Device (full HID) support »), qui fonctionne avec tous les périphériques d'entrée du menu « Input core support », et des pilotes simplifiés (options du sous-menu « USB HID Boot Protocol drivers »), que l'on utilisera pour réaliser des systèmes embarqués ou des noyaux légers. Il est évidemment recommandé d'utiliser le premier pilote, si réellement le support du clavier USB est nécessaire (normalement, les claviers USB sont pris en charge par le BIOS de l'ordinateur et apparaissent exactement comme des claviers classiques pour le système d'exploitation).

Il est fortement recommandé, si l'on désire utiliser des périphériques USB, d'activer les fonctionnalités de chargement dynamique des modules, de gestion des périphériques connectables à chaud et de chargement des firmware automatiques de Linux.

VIII-D-3 - Installation des périphériques IEEE1394

Le bus IEEE1394 (initialement dénommé bus FireWire) est un bus série à grande vitesse, capable de transférer des données avec un très grand débit (400 mégabits par secondes). Ce bus peut être utilisé pour connecter de nombreux périphériques (disques durs et lecteurs de CDROM externes par exemple), mais sa principale utilisation reste pour les transferts de données vidéo provenant des caméscopes numériques. Certaines cartes mères disposent d'un port IEEE1394 intégré, mais la plupart nécessitent l'ajout d'une carte fille PCI. La plupart de ces cartes sont vendues en tant que cartes de montage vidéo et disposent de deux ports IEEE1394 afin de permettre une entrée et une sortie du flux vidéo.

Le bus IEEE1394 dispose d'avantages indéniables, puisqu'il permet de brancher et de débrancher un périphérique à chaud (c'est-à-dire sans redémarrer l'ordinateur), les données sont transférées en numérique (donc sans perte de qualité pour les séquences vidéo), et de piloter les appareils qui y sont connectés (ce qui permet de contrôler par exemple un caméscope à partir d'un logiciel de montage vidéo). Il est également possible d'utiliser le bus IEEE1394 pour connecter deux ordinateurs en réseau, bien que cette fonctionnalité soit encore expérimentale.

Linux dispose des pilotes permettant d'utiliser les ports IEEE1394 de manière générique. Cependant, il faut bien le reconnaître, les logiciels spécifiques aux différents types d'appareils sont encore très rares ou incomplets. Cet état de fait est particulièrement vrai en ce qui concerne les caméscopes numériques, pour lesquels il n'existe quasiment aucun logiciel de montage vidéo dont la qualité s'approche de celle de ceux qui ont été développés pour Windows. Cela dit, les autres périphériques tels que les disques durs et les lecteurs de CD-ROM externes sont mieux supportés, et la configuration du port IEEE1394 peut être réalisée complètement en attendant que des logiciels plus évolués

ne voient le jour. Vous pouvez obtenir les dernières informations concernant le support des périphériques FireWire sous Linux sur le site de linux1394.

VIII-D-3-a - Configuration du noyau

La prise en charge du bus IEEE1394 au niveau du noyau est réalisable par l'intermédiaire des options du menu « IEEE 1394 (FireWire) support (EXPERIMENTAL) ». Comme le support des périphériques IEEE1394 sous Linux en est encore à ses balbutiements, ce menu ne peut être accédé que si l'on a activé l'option « Prompt for development and/or incomplete code/drivers » du menu « Code maturity level options ». Les principales options utilisables sont les suivantes :

- l'option « IEEE 1394 (FireWire) support (EXPERIMENTAL) », qui est l'option principale qui active toutes les autres options. Il faut donc impérativement activer cette option et répondre 'Y' ;
- l'option « Texas Instruments PCILynx support », qui permet de prendre en charge les cartes IEEE1394 basées sur la puce PCILynx de Texas Instrument. Ce ne sont pas les cartes les plus couramment utilisées, aussi la réponse recommandée est-elle 'N'. Si vous activez cette option, vous pourrez configurer des paramètres complémentaires du gestionnaire de périphériques avec les options « Use PCILynx local RAM » (utilisation de la mémoire embarquée sur la carte PCI) et « Support for non-IEEE1394 local ports » (utilisation de fonctionnalités complémentaires non FireWire de ces cartes) ;
- l'option « OHCI-1394 support », qui active la prise en charge des périphériques IEEE1394 compatible OHCI (abréviation de l'anglais « Open Host Controller Interface »), qui sont les périphériques les plus courants à présent. La réponse recommandée est 'Y' ;
- l'option « OHCI-1394 Video support », qui permet de prendre en charge les caméras vidéo numériques. Cette option ajoute également une fonctionnalité intéressante qui permet aux programmes de partager les données avec le gestionnaire de périphériques directement. Cela permet d'éviter qu'une copie de ces données ne soit réalisée entre la mémoire du noyau et la mémoire de l'application et d'obtenir ainsi de meilleures performances. La réponse recommandée est bien évidemment 'Y'. Cette option n'est toutefois disponible que pour les périphériques OHCI ;
- l'option « SBP-2 support (Harddisks, etc.) », qui permet de prendre en charge les disques durs et les lecteurs de DVD connectés sur bus IEEE1394. Ces lecteurs apparaissent alors comme des périphériques SCSI standards et pourront être montés via l'un des périphériques `/dev/sdx` ;
- l'option « Raw IEEE1394 I/O support », qui permet aux applications de communiquer directement avec le bus IEEE1394 par l'intermédiaire d'un fichier spécial de périphérique. Cette option est nécessaire au bon fonctionnement de la plupart des applications IEEE1394 et la réponse recommandée est donc 'Y' ;
- l'option « Excessive debugging output », qui active l'émission de messages de traces pour toutes les données survenant sur le bus IEEE1394. Cette option saturera vos fichiers de traces très vite et n'est réellement utile que pour les développeurs, aussi faut-il répondre 'N'.

Comme pour tous les périphériques sous Linux, les fonctionnalités IEEE1394 du noyau seront accessibles par l'intermédiaire de fichiers spéciaux de périphériques situés dans le répertoire `/dev/`. Les fonctionnalités vidéo seront accédées par l'intermédiaire d'un fichier spécial de périphérique de type caractère et de code majeur 172. De même, l'interface de données brute activée par l'option « Raw IEEE1394 I/O support » est exposée par l'intermédiaire d'un fichier spécial de périphérique de type caractère et de code majeur 171. Vous devrez donc créer ces fichiers spéciaux à l'aide des deux commandes suivantes :

```
mknod /dev/raw1394 c 171 0
mknod -m 666 /dev/video1394 c 172 0
```

Le numéro de code mineur est utilisé pour distinguer les différents ports IEEE1394 présents dans la machine. Les lignes de commandes précédentes ne montrent que la manière de créer les fichiers spéciaux de périphériques que pour le premier port IEEE1394.

VIII-D-3-b - Installation des bibliothèques complémentaires

Les fonctionnalités IEEE1394 peuvent être utilisées par les applications directement par l'intermédiaire des fichiers spéciaux de périphériques `/dev/raw1394` et `/dev/video1394`, mais quasiment aucune ne le fait. En effet, des

bibliothèques spécifiques ont été écrites afin de fixer les interfaces de programmation de Linux avec ces applications. Il est donc nécessaire d'installer ces bibliothèques pour pouvoir utiliser les périphériques IEEE1394.

Il est encore rare que ces bibliothèques soient installées par les distributions, certainement parce qu'elles sont encore en cours de développement. Leur installation nécessite donc de les compiler soi-même, ce qui est une tâche facile si on sait le faire, mais qui peut effrayer un débutant. Rappelons une fois de plus que le support des périphériques IEEE1394 sous Linux est encore expérimental.

La bibliothèque de programmes la plus importante est celle qui permet d'utiliser la fonctionnalité d'accès direct aux périphériques IEEE1394 par l'intermédiaire du fichier spécial de périphérique `/dev/raw1394`. Les fichiers sources de cette bibliothèque peuvent être trouvés sur le site du projet [libraw1394](#). Une autre bibliothèque utilisée par les programmes vidéo est la bibliothèque [libavc1394](#). Cette bibliothèque permet en effet aux programmes de piloter les caméscopes numériques par l'intermédiaire du bus IEEE1394. Enfin, les programmes d'édition vidéo peuvent avoir besoin de la bibliothèque [libdv](#), qui permet de manipuler les données au format DV (c'est-à-dire le format de données utilisé par la plupart des caméscopes numériques). Cette bibliothèque ne fait pas à proprement parler partie des bibliothèques permettant de communiquer avec les périphériques IEEE1394, mais elle est extrêmement utile.

La compilation de ces bibliothèques se fait classiquement avec les commandes suivantes :

```
./configure --prefix=/usr  
make
```

à partir du répertoire des sources. Celui-ci pourra être extrait des archives à l'aide de la commande `tar xzf archive`, où `archive` est le nom de l'archive en question. Une fois compilées, les bibliothèques pourront être installées avec la commande :

```
make install
```

Ces opérations ne devraient pas poser de problème particulier. Consultez le [Chapitre 7](#) pour plus de détails sur ces opérations.

Une fois ces bibliothèques installées, vous devriez pouvoir installer et utiliser des applications dédiées aux périphériques IEEE1394, comme `dvgrab` (outil de capture vidéo), `kino` ou `broadcast2000` (outils d'édition de séquences vidéo). Force est de reconnaître que ces programmes sont très loin d'être finis et réellement difficiles à utiliser.

IX - Configuration du réseau

Linux est un système d'exploitation fabriqué par l'Internet et pour l'Internet. Inutile de préciser que c'est l'un des meilleurs systèmes pour gérer et exploiter un réseau. Certains ne l'utilisent d'ailleurs que pour cela, et profitent de ses excellentes performances sur les petites machines afin de récupérer du matériel autrement voué à la casse. En fait, les fonctionnalités réseau de Linux sont si nombreuses que j'ai été obligé d'y consacrer un chapitre à part entière.

La configuration d'un réseau est une opération qui nécessite quelques connaissances théoriques sur le fonctionnement des réseaux TCP/IP. Ces informations sont assez techniques, mais indispensables pour bien configurer les services réseau de toute machine connectée, et pas seulement les machines fonctionnant sous Linux. Il n'est en effet pas rare de trouver des réseaux de machines fonctionnant sur des systèmes dont la configuration est supposée être plus simple, mais dont l'organisation est une hérésie absolue et qui risquent de nécessiter une remise à plat complète à chaque interconnexion.

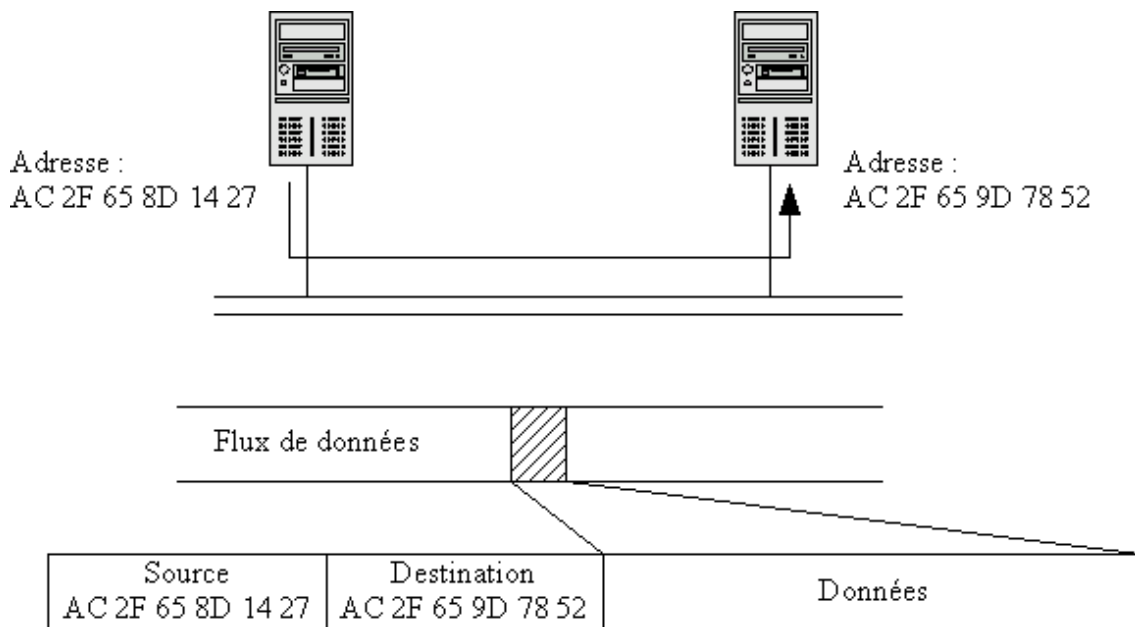
Cette section va donc donner quelques explications sur les notions fondamentales des réseaux informatiques. Il traitera ensuite de la configuration des réseaux locaux, puis celle des connexions temporaires à Internet. Les services réseau évolués tels que le partage de connexion à Internet et la création d'un serveur de fichiers seront finalement traités en fin de chapitre.

IX-A - Notions de réseau TCP/IP

IX-A-1 - Généralités sur les réseaux

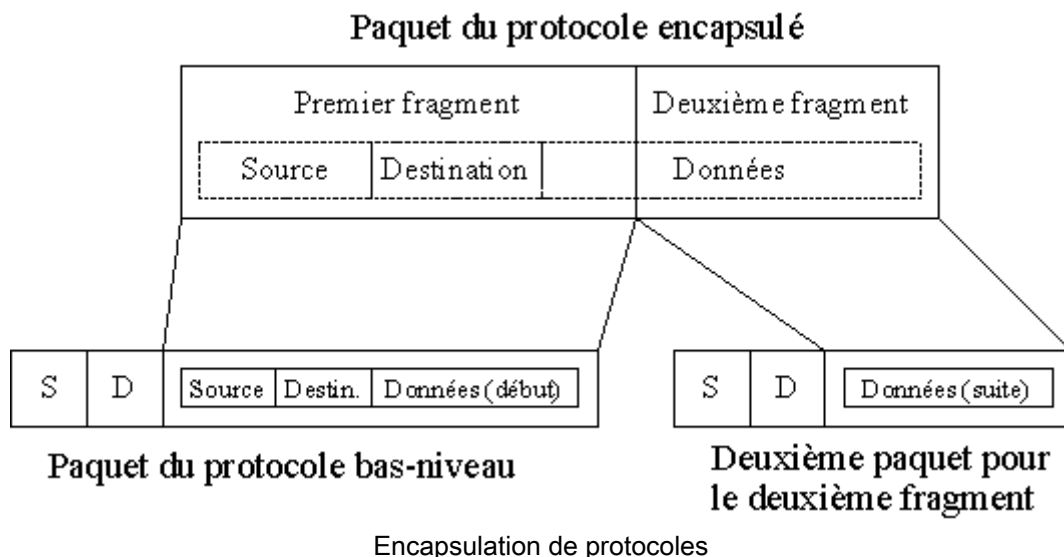
Un réseau n'est en général rien d'autre qu'une interconnexion entre plusieurs machines qui leur permet d'échanger des informations. Il existe de nombreux moyens de réaliser cette interconnexion, qui utilisent parfois des supports physiques variés. Les techniques les plus utilisées sont la liaison radio et la liaison par câble. Cette dernière technique comprend diverses variantes, dont les réseaux Ethernet, TokenRing et simplement la liaison téléphonique.

Il est évident que la manière d'envoyer et de recevoir des informations est différente pour ces différents supports physiques, parce qu'elle dépend tout simplement des possibilités techniques offertes par la technologie sous-jacente utilisée. Cependant, il est très courant de découper les informations à échanger en paquets, qui sont ensuite transmis sur le réseau. Ces paquets peuvent être de tailles variées, et contenir des informations utiles à la gestion du réseau. L'information la plus importante est sans doute celle permettant de connaître la machine destinataire du paquet. On l'appelle l'adresse de la machine cible, il s'agit généralement d'un numéro permettant de l'identifier de manière unique sur le réseau. En général, les paquets contiennent également l'adresse de la machine source, afin qu'une réponse puisse lui être envoyée.



Notion de paquet et d'adresse réseau

Du fait de la diversité des supports physiques de réseau, il n'est pas simple d'écrire une application réseau qui puisse travailler dans des environnements réseau hétérogènes. Cela supposerait de connaître les protocoles de communication pour chaque type de réseau, ce qui compliquerait à l'infini le moindre programme et le rendrait inutilisable avec les nouveaux réseaux. Par conséquent, cette tâche ingrate a été reléguée au plus profond des couches réseau spécifiques au support physique. Les applications quant à elles utilisent un protocole de communication plus évolué, dont le but est d'assurer l'interopérabilité des différents supports physiques. Ce protocole utilise toujours des paquets et une notion d'adresse, mais cette fois ces informations sont standardisées et utilisables par toutes les applications. Les paquets de ce protocole sont stockés dans les paquets des réseaux physiques et transmis tels quels. Ils peuvent éventuellement être découpés en sous-paquets dans le cas où la taille des paquets du réseau serait trop petite pour les contenir. Cette technique s'appelle l'encapsulation d'un protocole dans un autre protocole.



IX-A-2 - Le protocole IP

Les machines Unix utilisent toutes le protocole de communication de bas niveau IP (« Internet Protocol »). Ce protocole a été inventé pour permettre l'interconnexion d'un grand nombre de réseaux physiques différents (le nom d'Internet provient d'ailleurs de cette caractéristique : « INTERconnected NETworks »). Il permet de transmettre des informations de manière uniforme sur tous ces réseaux physiques. Ainsi, les programmes qui utilisent IP ne voient pas les spécificités des différents réseaux physiques. Pour eux, il ne semble y avoir qu'un seul réseau physique, dont le protocole de communication de base est IP. Autrement dit, les applications qui utilisent le réseau se contentent d'utiliser le protocole IP, et n'ont plus à se soucier de la manière dont il faut formater et transmettre les informations sur chaque support physique du réseau. Ce genre de détail est laissé aux couches réseau de chaque machine et aux passerelles reliant les divers réseaux physiques.

Comme il l'a déjà été dit ci-dessus, le protocole IP utilise des adresses pour identifier les machines sur les réseaux. Les adresses IP sont codées sur quatre octets (nombres binaires à huit chiffres, permettant de représenter des valeurs allant de 0 à 255), chacun définissant une partie du réseau. Ces adresses sont utilisées un peu comme les numéros de téléphone : le premier octet définit le numéro d'un « super-réseau » dans lequel le correspondant se trouve (ces « super-réseaux » sont appelés les réseaux de classe A), le deuxième octet définit le numéro du sous-réseau dans le super-réseau (ces sous-réseaux sont appelés réseaux de classe B), le troisième octet définit encore un sous-sous-réseau (réseaux dits de classe C) et le quatrième octet donne le numéro de la machine dans ce sous-sous-réseau.

Cette numérotation permet d'affecter des adresses similaires pour les différentes machines d'un réseau, et de simplifier ainsi la gestion de ce dernier. Elle dispose en revanche d'un inconvénient majeur : beaucoup d'adresses sont gaspillées, car il n'y a pas suffisamment de réseaux de classe A d'une part, et qu'on ne peut pas mélanger les machines de deux sous-réseaux dans un même réseau de classe A d'autre part. Si l'on reprend la comparaison avec les numéros de téléphone, il y a énormément d'abonnés dont le numéro commence par 01, mais beaucoup moins dont le numéro commence par 02, et quasiment aucun dont le numéro commence par 08. Si l'on venait à manquer de place dans la liste des numéros commençant par 01, on ne pourrait pas pour autant utiliser les numéros commençant par 02 pour des raisons géographiques. C'est la même chose pour les adresses IP, sauf que les zones géographiques sont remplacées par des sous-réseaux. Le problème est que, malheureusement, on commence à manquer d'adresses disponibles (alors qu'il y en a plein de libres, mais inutilisables parce qu'elles se trouvent dans d'autres sous-réseaux !). Il va donc falloir effectuer une renumérotation d'ici peu, exactement comme il y en a déjà eu dans le monde de la téléphonie...

Note : Le protocole IPv6, qui remplacera le protocole IP classique (encore appelé IPv4), a pour but de résoudre les limitations du protocole IP utilisé actuellement. Les adresses du protocole IPv6 sont codées sur 16 octets, ce qui résoudra définitivement le problème du manque d'adresses. De plus, les services modernes que sont l'authentification de l'émetteur, ainsi que la qualité de service (c'est-à-dire la garantie du délai de transmission des données, garantie nécessaire pour transmettre de façon correcte les flux multimédias tels que le son et la vidéo en temps réel) sont

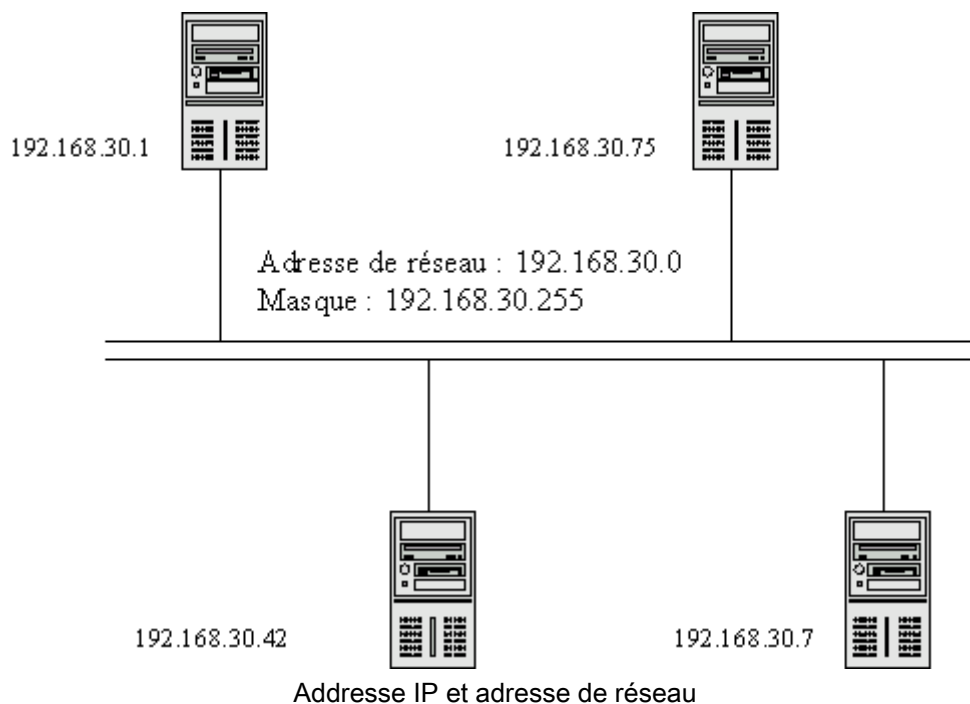
fournis par IPv6. Bien entendu, Linux est déjà capable d'utiliser IPv6 (combien de systèmes peuvent aujourd'hui l'affirmer ?) ! Notez toutefois que pour cela, il faut recompiler le noyau et toutes les applications réseau du système, ce qui est tout de même très lourd. Par conséquent, il vaut mieux se contenter du protocole IP actuel. Malgré ses limitations, ce protocole reste sans doute le meilleur protocole réseau du monde, car il allie souplesse et fonctionnalité. Il est difficilement concevable de créer un réseau aussi grand qu'Internet avec les autres protocoles existant sur le marché...

Les adresses IP sont donc parfaitement définies à l'aide de leurs quatre nombres, que l'on note les uns à la suite des autres et en les séparant d'un point. Comme on l'a vu, les adresses IP sont classées en sous-réseaux, de classe A, B et C. Les adresses des réseaux de classe C ont leurs trois premiers nombres fixés, et seul le quatrième nombre change pour chaque machine du réseau. De la même manière, les réseaux de classe B ont leurs deux premiers nombres fixés, et seuls les deux derniers nombres permettent de distinguer les différentes machines du réseau. Enfin, les réseaux de classe A n'ont de fixé que leur première composante, les autres sont libres. Il est donc clair qu'il existe peu de réseaux de classe A, mais que ce sont de très gros réseaux (ils peuvent contenir jusqu'à 16 millions de machines !). En revanche, il existe beaucoup plus de réseaux de classe C, dont la taille est plus modeste (seulement 256 machines).

Pour un réseau donné, les adresses ont donc toutes la même forme. Les premiers octets des adresses du réseau sont toujours les mêmes (ce peut être le premier octet pour les réseaux de classe A, les deux premiers pour les réseaux de classe B ou les trois premiers pour les réseaux de classe C). On peut donc définir la notion d'adresse de réseau, qui est l'adresse IP d'une machine du réseau dont les parties variables ont pour valeur 0. Par exemple, si une machine d'un réseau de classe C a pour adresse 192.168.1.15, alors l'adresse de son sous-réseau est 192.168.1.0. Cela signifie que toutes les machines de ce réseau auront une adresse de la forme « 192.168.1.xxx ».

Un réseau n'appartient qu'à une et une seule classe. Les adresses IP sont réparties sur les différentes classes de réseaux, selon la valeur des bits de poids fort de leur premier octet. Par exemple, les réseaux de classe A sont identifiables au fait que le bit de poids fort de leur adresse est nul. Les adresses de réseau valides pour les réseaux de ce type sont donc les adresses comprises entre 0.0.0.0 et 127.0.0.0. Il n'existe donc que 128 réseaux de classes A en tout et pour tout. Les autres réseaux ont donc le bit de poids fort de leur adresse fixé à 1, et c'est le deuxième bit de poids fort qui est utilisé pour distinguer les réseaux de classe B des autres. Les réseaux de classe B utilisent toujours la valeur 0 pour ce bit, leurs adresses varient donc entre 128.0.0.0 et 191.255.0.0. De même, les réseaux de classe C utilisent la valeur 1 pour le deuxième bit de leur adresse, et ont nécessairement un troisième bit nul. Leurs adresses vont donc de 192.0.0.0 à 223.255.255.0. Les adresses pour lesquelles le troisième bit (en plus des deux premiers) est à 1 sont réservées (soit pour une utilisation ultérieure, soit pour s'adresser à des groupes d'ordinateurs en multicast) et ne doivent pas être utilisées. Il s'agit des adresses 224.0.0.0 à 255.255.255.255. Cette dernière adresse a une signification spéciale et permet de s'adresser à tous les ordinateurs d'un réseau.

Il est possible de déterminer l'adresse du réseau auquel une machine appartient en utilisant ce qu'on appelle le masque de sous-réseau. Le masque de sous-réseau est une série de quatre nombres ayant le même format que les autres adresses IP, mais dont les composantes ne peuvent prendre que la valeur 0 ou la valeur 255, les 255 devant nécessairement apparaître en premier. Les composantes des adresses IP qui correspondent à la valeur 255 dans le masque de sous-réseau font partie de l'adresse dudit sous-réseau. Les composantes qui correspondent à la valeur 0 dans le masque de sous-réseau n'en font pas partie, et varient pour chaque machine du réseau. Pour reprendre l'exemple précédent, si une machine a pour adresse IP 189.113.1.15 et que son masque de sous-réseau est 255.255.255.0, alors l'adresse de son réseau est 189.113.1.0. Si le masque de sous-réseau avait été 255.255.0.0 (typiquement le masque d'un réseau de classe B), l'adresse du réseau aurait été 189.113.0.0. Comme on le voit, le masque de sous-réseau est utilisé par le système pour déterminer rapidement l'adresse de sous-réseau d'une machine à partir de son adresse IP. On notera que certaines combinaisons d'adresses IP et de masques de sous-réseau sont invalides. Par exemple, les adresses affectées aux réseaux de classe C (comme 192.168.0.1 par exemple) ne peuvent pas avoir de masque de sous-réseau en 255.255.0.0, car cela impliquerait que cette adresse serait une adresse de réseau de classe B.



Les adresses IP ne sont pas attribuées aux machines au hasard. Il est évident que chaque machine doit avoir une adresse unique, et que son adresse doit appartenir à la plage d'adresses utilisée pour le sous-réseau dont elle fait partie. Pour cela, les classes de réseau, ainsi que les adresses qu'ils utilisent, sont attribuées par l'IANA, un organisme de gestion de l'Internet. Le rôle de l'IANA (abréviation de l'anglais « Internet Assigned Numbers Authority ») est essentiellement d'assurer l'unicité des adresses IP sur l'Internet. Cependant, certaines adresses sont librement utilisables pour les réseaux locaux qui ne sont pas connectés directement à l'Internet. Les paquets utilisant ces adresses sont assurés de ne pas être transmis sur Internet. Ces adresses peuvent donc être utilisées par quiconque. Les plages d'adresse réservées sont les suivantes :

Tableau 9-1. Plages d'adresses IP réservées pour un usage personnel

Classe de réseau	Adresses de réseau réservées
A	10.0.0.0
B	172.16.0.0 à 172.31.0.0
C	192.168.0.0 à 192.168.255.0

Ainsi, un réseau de classe A (d'adresse 10.0.0.0), 16 réseaux de classe B (les réseaux 172.16.0.0 à 172.31.0.0) et 255 réseaux de classe C (d'adresses 192.168.0.0 à 192.168.255.0) sont disponibles. Vous pouvez donc les utiliser librement.

Il est également possible de configurer les machines pour qu'elles récupèrent leurs adresses IP auprès d'un serveur à l'aide du protocole DHCP (abréviation de l'anglais « Dynamic Host Configuration Protocol »). Cette technique est très intéressante quand on dispose d'un grand nombre de machines qui ne sont pas toujours toutes connectées à un réseau. Il est donc possible de redistribuer les adresses IP d'un stock d'adresses en fonction des machines qui se connectent, et d'économiser ainsi les précieuses adresses. En revanche, elle n'est pas appropriée pour les serveurs qui sont couramment accédés par des postes clients, et qui doivent donc avoir une adresse IP fixe.

Certaines adresses IP ont une signification particulière et ne peuvent pas être attribuées à une machine. Par exemple l'adresse 127.0.0.1 représente, pour une machine, elle-même. Cette adresse est souvent utilisée pour accéder à un programme réseau sur la machine locale. Elle fait partie du sous-réseau de classe A 127.0.0.0, qui ne comprend pas d'autres adresses. De plus, les adresses dont les derniers nombres (c'est-à-dire les nombres qui ne font pas partie de l'adresse du réseau) se terminent par 0 ou 255 sont réservées pour l'envoi des paquets à destination de tout le monde sur le réseau (émission dite « broadcast »). Par exemple, les adresses 192.168.1.0 et 192.168.1.255

ne peuvent pas être affectées à une machine. Ce sont typiquement ces adresses qui sont utilisées par le protocole DHCP pour émettre des requêtes sur le réseau alors que la machine n'a pas encore d'adresse fixe.

Il est important de savoir que, par défaut, une machine ne communiquera qu'avec les machines de son propre réseau. C'est-à-dire que si une machine utilise l'adresse IP 192.168.1.15 et que son masque de sous-réseau est 255.255.255.0, elle ne pourra contacter que des machines dont l'adresse est de la forme 192.168.1.xxx. Elle ne pourra donc pas voir par exemple une machine dont l'adresse IP est 192.168.0.2. Cela ne signifie pas que l'on doit toujours utiliser le masque 0.0.0.0 pour voir toutes les machines du monde, mais plutôt que la machine 192.168.0.2 ne fait pas partie, a priori, du même réseau physique que la machine 192.168.1.15. Il est donc inutile de chercher à la contacter (et mettre le masque de sous-réseau à 0.0.0.0 ne résoudrait évidemment pas le problème). Cependant, si deux réseaux physiques ont nécessairement deux adresses de réseau différentes, rien n'empêche de définir, sur un même réseau, plusieurs réseaux logiques. Ainsi, une même carte réseau peut avoir plusieurs adresses IP. La communication avec les machines des différents réseaux logiques se fait alors par l'intermédiaire de la même interface réseau.

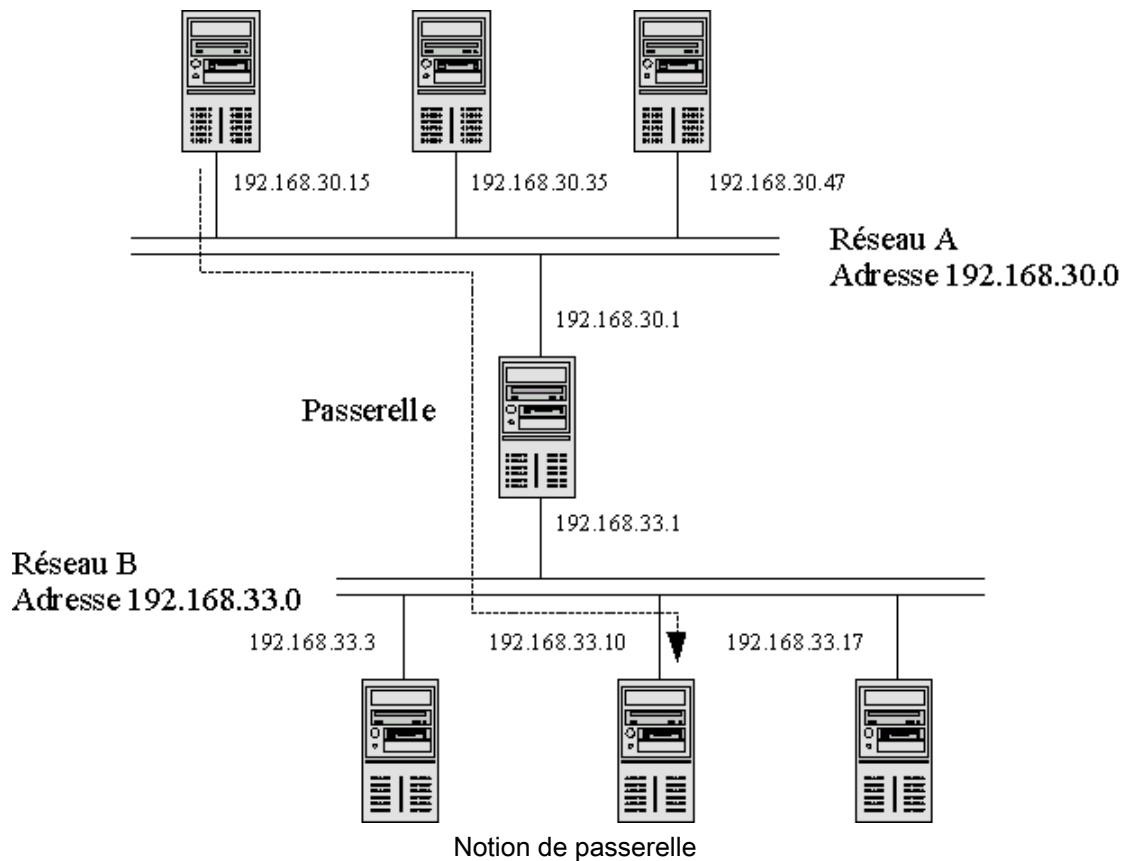
Arrivé à ce stade des explications, je sens venir la question suivante : « ?! Euhhh... Mais alors, comment peut-on voir les machines sur Internet ? Je n'ai pas de réseau, et quand je me connecte à Internet, je peux y accéder ! Et même si j'avais un réseau, elles ne feraient certainement pas partie de mon réseau... ».

Explications :

- premièrement, vous avez un réseau, même si vous ne le savez pas. Toute machine appartient généralement au moins à son propre réseau virtuel, sur laquelle elle est la seule machine, et où elle a l'adresse 127.0.0.1 ;
- deuxièmement, effectivement, les machines qui se trouvent sur Internet n'appartiennent pas à votre réseau, que celui-ci existe effectivement ou soit virtuel ;
- troisièmement, toutes les informations que vous envoyez et recevez transitent par un seul ordinateur, celui de votre fournisseur d'accès à Internet. C'est cet ordinateur qui se charge de faire le transfert de ces informations vers les machines situées sur Internet.

C'est donc ici qu'intervient la notion de passerelle (« Gateway » en anglais).

Une passerelle est une machine qui appartient à deux réseaux physiques distincts, et qui fait le lien entre les machines de ces deux réseaux. Les ordinateurs des deux réseaux peuvent communiquer avec la passerelle de part et d'autre, puisqu'elle appartient aux deux réseaux. Les ordinateurs de chaque réseau transmettent à cette passerelle tous les paquets qui ne sont pas destinés à une machine de leur propre réseau. Celle-ci se charge simplement de transférer ces paquets aux machines de l'autre réseau.



Lorsque vous vous connectez à Internet, vous ne faites rien d'autre que de créer un réseau (dont le support physique est la ligne téléphonique), et vous utilisez l'ordinateur que vous avez appelé comme passerelle par défaut. Tous les paquets destinés à un autre réseau que le vôtre (donc, en pratique, tous les paquets si vous n'avez pas de réseau local) sont donc envoyés sur le réseau constitué de votre connexion à Internet et arrivent donc chez votre fournisseur d'accès, qui se charge ensuite de les transmettre aux autres ordinateurs. Notez que celui-ci peut transmettre ces paquets à une autre passerelle à laquelle il a accès et ainsi de suite, jusqu'à ce que la destination soit atteinte.

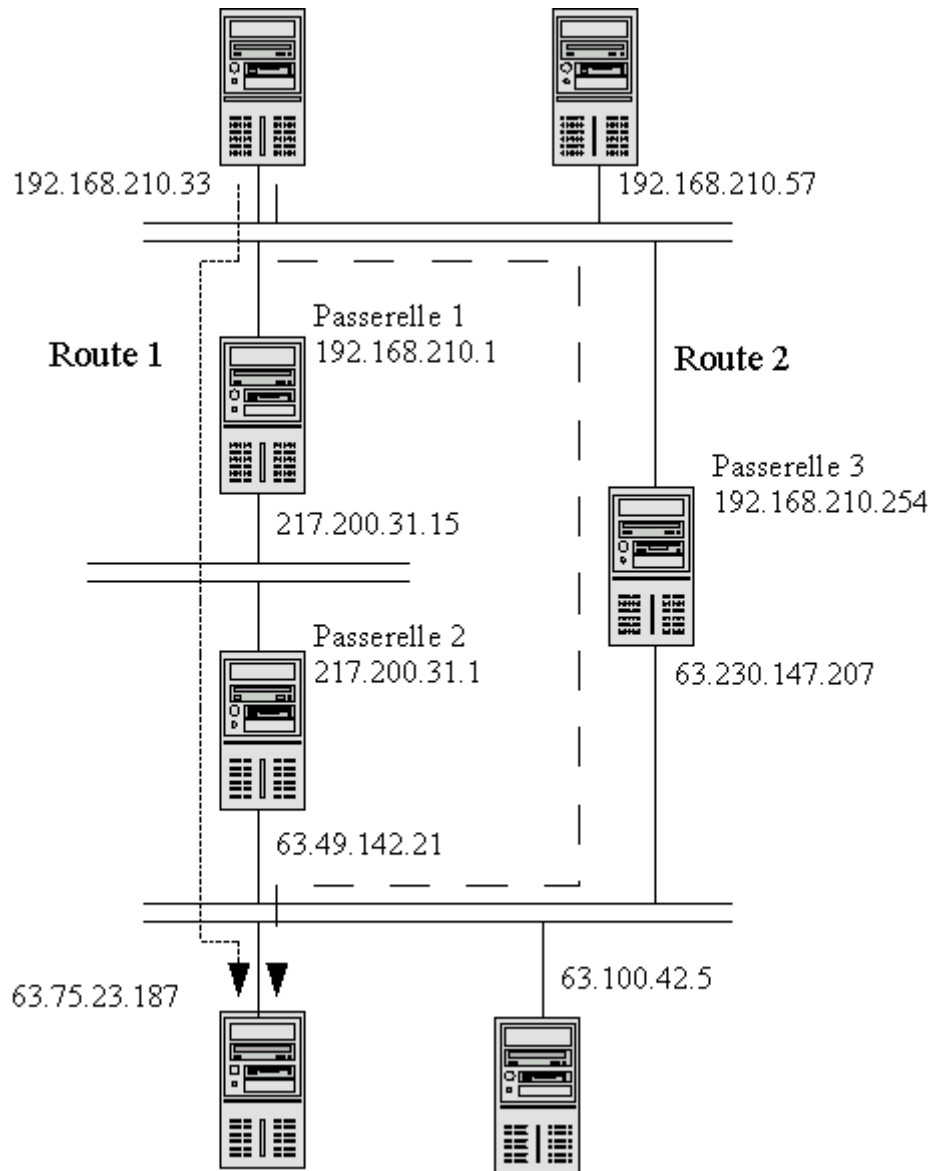
Dans le cas d'un particulier, le choix du réseau sur lequel les paquets doivent être transmis est très facile à faire puisqu'en général un paquet est soit à destination de la machine locale, soit à destination d'une machine sur Internet. Pour un paquet destiné à la machine locale, le réseau virtuel local est utilisé. Tous les autres paquets sont envoyés sur la connexion Internet. Cependant, il peut arriver qu'une machine ait le choix entre plusieurs passerelles différentes pour envoyer un paquet dont la destination n'est pas sur son propre réseau. Par exemple, les passerelles d'Internet peuvent être elles-mêmes connectées à différents autres réseaux, qui sont eux-mêmes connectés à d'autres réseaux. Un paquet peut donc être acheminé de plusieurs manières à sa destination, selon la topologie du réseau. L'ensemble des réseaux empruntés par un paquet dans son voyage constitue ce qu'on appelle sa route.

Chaque passerelle contribue donc à déterminer la route des paquets en choisissant, pour chaque paquet, l'interface réseau à utiliser pour transmettre ce paquet. Ce choix est fait en suivant un certain nombre de règles basées sur l'adresse destination des paquets. Par exemple, si une passerelle reçoit un paquet dont l'adresse destination est 129.46.10.15, et qu'elle est elle-même connectée à un réseau possédant cette machine, elle transmettra bien évidemment ce paquet sur ce réseau. Si en revanche elle ne peut localiser la machine cible sur l'un de ses réseaux, elle l'envoiera à une autre passerelle à laquelle elle a accès. Le choix de cette passerelle est en général déterminé par des règles de la forme suivante : « Tous les paquets destinés au sous-réseau 129.46.0.0 doivent être envoyés vers la passerelle 193.49.20.1 du réseau 193.49.20.0 ». Notez qu'à chaque étape de la route, seules les passerelles de l'étape suivante peuvent être choisies comme relais.

Ainsi, si la passerelle est connectée à trois réseaux d'adresses respectives 192.168.1.0, 193.49.20.0 et 209.70.105.10, elle transférera le paquet à destination de 129.46.10.15 à la passerelle 193.49.20.1. La transmission effective du paquet à cette passerelle se fera ensuite selon le protocole réseau bas niveau utilisé sur le réseau 193.49.20.0. Autrement dit, une passerelle est une machine qui, contrairement aux machines classiques, accepte de

recevoir des paquets qui ne lui sont pas directement adressés, et de les traiter en les réémettant vers leur destination ou vers une autre passerelle, selon les règles de routage qu'elle utilise.

Les règles utilisées par les passerelles sont stockées dans ce qu'on appelle des tables de routage. Les tables de routage peuvent être configurées statiquement dans la configuration des passerelles, c'est-à-dire initialisées dans les fichiers de configuration et ne jamais être modifiées. Cette technique convient parfaitement pour les petits réseaux. Elle se révèle cependant insuffisante sur les passerelles d'Internet. En effet, si les réseaux empruntés par une route sont saturés, voire hors service, il peut être intéressant d'utiliser une autre route. Ces passerelles utilisent donc des tables de routage dynamiques, qui sont automatiquement mises à jour en fonction de l'état du réseau. Bien que Linux puisse parfaitement effectuer ce type de routage, c'est une configuration réservée à un usage spécifique de Linux. Ce document ne traitera donc pas du routage dynamique.



Notion de route

Le problème des adresses IP est qu'elles ne sont pas très parlantes pour les êtres humains : que peut donc signifier 192.147.23.2 ? Pas grand-chose... C'est pour cela qu'on affecte des noms de machines plus humains, par exemple `krypton.andromede.galaxie`. Ces noms suivent une convention de nommage bien précise. En général, le premier nom est le nom de la machine elle-même, et la suite du nom constitue ce qu'on appelle le domaine dans laquelle la machine se trouve. Ainsi, dans l'exemple précédent, `krypton` est le nom d'une machine, et `andromede.galaxie` est le nom de son domaine. En général, il existe plusieurs machines dans un même domaine, on pourrait donc également avoir `altair.andromede.galaxie` par exemple (malheureusement pour mon exemple, l'étoile Altair ne se

trouve pas dans la galaxie d'Andromède, mais bon...). Souvent, les noms de domaines sont des noms de sociétés. La dernière partie du nom de domaine permet également de déterminer la nature du domaine, ou sa localisation. Par exemple, l'extension `.com` indique clairement que le domaine est de nature commerciale (de surcroît, il s'agit sans doute d'une société américaine, l'extension `.com` étant réservée aux États-Unis). De même, l'extension `.gov` est utilisée pour les organismes gouvernementaux américains, et l'extension `.edu` pour les universités ou les écoles américaines. L'extension `.org` est utilisée pour les organisations non commerciales. Enfin, les noms de domaines des autres pays que les États-Unis utilisent quasiment systématiquement une extension indiquant leurs pays d'origine. Ainsi, `.fr` représente la France, `.uk` les Royaumes-Unis, etc. Notez que la notion de domaine est a priori distincte de la notion de réseau.

La question technique qui se pose avec ces conventions de nommage « humaines » est de savoir comment déterminer, à partir d'un nom littéral, l'adresse IP de la machine correspondante. Cette opération n'est pas simple, et en fait elle est effectuée de deux manières :

- soit la machine locale demande à une autre machine qu'elle connaît d'effectuer la recherche de l'adresse du correspondant ;
- soit elle dispose d'une liste de noms et d'adresses lui permettant de déterminer directement les adresses de ses interlocuteurs.

La première solution utilise ce qu'on appelle un serveur de noms (« DNS » en anglais, abréviation de « Domain Name Service »), qui connaît toutes les machines du réseau soit directement, soit indirectement (donc via un autre DNS). L'opération qui consiste à retrouver l'adresse IP d'une machine à partir de son nom s'appelle la résolution de nom de domaine. Il est évident que si le serveur de noms ne peut être contacté, il sera impossible d'utiliser les noms de machines. En revanche, il sera toujours possible d'utiliser leurs adresses IP directement, si on les connaît. La deuxième solution ne souffre pas de ce défaut, elle nécessite cependant de mettre à jour la liste des noms sur tous les postes régulièrement, ce qui est beaucoup plus complexe que la gestion centralisée d'un DNS. On ne l'utilisera donc que pour les petits réseaux.

IX-A-3 - Le protocole TCP

Nous avons vu que le protocole IP fournit les briques de base de toute la communication réseau sous Unix (et Internet). Ses principales fonctionnalités sont le découpage des informations en paquets de taille suffisamment petite pour pouvoir transiter sur tous les types de réseaux physiques, la gestion des destinations des paquets à l'aide des adresses IP, et le choix de la route permettant d'acheminer les paquets à destination. En revanche, il est incapable d'assurer les services plus évolués comme la gestion de l'ordre d'arrivée des paquets et la gestion de la fiabilité du transfert des informations. C'est donc à des protocoles plus évolués, eux-mêmes encapsulés dans IP, d'effectuer ces tâches. L'un des protocoles les plus importants est le protocole TCP (abréviation de l'anglais « Transfer Control Protocol »). Ce protocole se charge d'établir les connexions entre deux ordinateurs, et assure la fiabilité des informations transmises et l'arrivée des informations dans leur ordre d'envoi. Il existe d'autres protocoles, moins connus que TCP, mais tout aussi importants. On retiendra en particulier les deux protocoles suivants :

- UDP (abréviation de l'anglais « User Datagram Protocol »), qui permet d'émettre des datagrammes sur le réseau, qui sont de simples paquets de données (c'est un protocole semblable à IP, destiné aux applications) ;
- ICMP (abréviation de « Internet Control Message Protocol »), qui est utilisé essentiellement pour transmettre des messages de contrôle du fonctionnement des autres protocoles (il est donc vital).

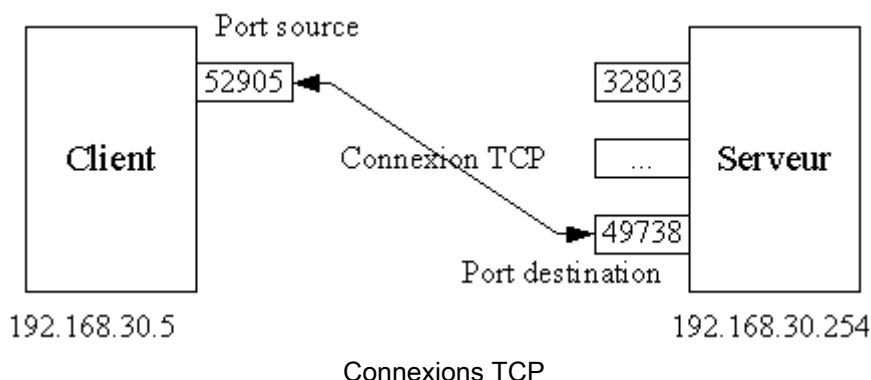
Les services réseau des machines sont organisés en couches logicielles, qui s'appuient chacune sur la couche inférieure. Ainsi, TCP utilise IP, qui lui-même utilise le pilote qui gère l'interface réseau. Du fait que ces couches s'appuient les unes sur les autres, on appelle souvent l'ensemble de ces couches une pile (« stack » en anglais). Vous avez peut-être déjà entendu parler de la pile TCP/IP. Lorsqu'un service réseau d'une machine n'est plus accessible, il se peut que ce service réseau ait planté. Si tout un ensemble de services réseau ne fonctionne plus, c'est certainement une des couches logicielles qui est plantée. Par exemple, une machine peut répondre à la commande ping (classiquement utilisée pour tester les connexions réseau) et ne plus accepter la plupart des connexions réseau. Cela signifie simplement que la couche TCP ne fonctionne plus, et que la couche ICMP (utilisée par ping) est toujours

valide. Évidemment, si la couche IP tombe en panne, la machine ne sera plus accessible du tout, sauf éventuellement avec d'autres protocoles réseau complètement différents (IPX, Appletalk, etc.).

Seuls les mécanismes du protocole TCP seront détaillés dans la suite de ce document. Le protocole TCP est en effet utilisé par un grand nombre de services, ce qui en fait certainement le plus connu.

Le protocole TCP utilise la notion de connexion. Une connexion est un canal de communication établi entre deux processus par TCP. Comme les processus sont susceptibles d'utiliser plusieurs connexions simultanément, TCP fournit la possibilité de les identifier par un numéro unique sur la machine, compris entre 0 et 65535. Chaque numéro identifie ce qu'on appelle un port TCP. Quand un processus désire établir une connexion avec un autre, il utilise un de ses ports et essaie de se connecter sur le port du deuxième processus.

Il faut bien comprendre que les deux numéros de ports utilisés ne sont a priori pas les mêmes pour les deux processus. Évidemment, il est nécessaire que les processus clients connaissent les numéros de port des processus serveurs auxquels ils se connectent. Les numéros de ports sont donc affectés à des services bien définis, et les serveurs qui fournissent ces services doivent bien entendu utiliser ces numéros de ports. Ainsi, il est possible de déterminer de manière unique un programme serveur sur un réseau avec l'adresse IP de la machine sur laquelle il fonctionne et le numéro de port qu'il écoute pour les connexions extérieures. Les clients qui se connectent savent donc parfaitement à quel service ils accèdent lorsqu'ils choisissent le numéro de port destination. Leur propre numéro de port est en général choisi par le système, afin d'éviter tout conflit avec un autre processus de la même machine.



Une fois établie, une connexion TCP permet d'effectuer des communications bidirectionnelles. Cela signifie que le client et le serveur peuvent tous deux utiliser cette connexion pour envoyer des données à l'autre. Le client envoie ses données sur le port destination du serveur, et le serveur peut renvoyer les résultats au client en utilisant le port de celui-ci. Les paquets TCP disposent donc toujours d'un port source (c'est-à-dire le port TCP de l'émetteur), et d'un port destination (le port du récepteur). Ainsi, le récepteur peut renvoyer sa réponse en utilisant le port source comme port de destination du paquet renvoyé, et inversement.

Comme il l'a déjà été dit, le protocole TCP s'assure que les informations transmises arrivent à bon port (c'est le cas de le dire !). Pour cela, il utilise un mécanisme d'accusés réception, qui indiquent à l'émetteur si le destinataire a bien reçu chaque paquet envoyé. Si l'accusé réception n'est pas reçu dans un temps fixé, le paquet est réémis. Un paquet reçu en double à cause d'un retard dans la communication de l'accusé réception est tout simplement ignoré. La fiabilité des informations est également assurée. Cette fiabilité est réalisée par un mécanisme de sommes de contrôle. Si le récepteur constate que la somme de contrôle des données reçues n'est pas celle que l'émetteur a calculée, il rejette le paquet parce qu'il sait que les informations ont été corrompues pendant la transmission. Enfin, TCP s'assure que les informations émises en plusieurs passes sont bien reçues dans leur ordre d'émission. Cette réorganisation se fait grâce à une numérotation des paquets (cette numérotation sert également à détecter les paquets reçus en double). Elle peut paraître inutile, mais la vitesse d'arrivée des paquets est hautement dépendante de la route IP qu'ils prennent pour parvenir à destination. Les paquets qui arrivent en avance sont donc mémorisés jusqu'à ce que tous les paquets qui les précèdent soient reçus.

IX-A-4 - Les protocoles de haut niveau

TCP fournit donc les fonctionnalités requises pour la plupart des services réseau existant. Il est logique que ceux-ci s'appuient sur lui pour effectuer leur travail. Cependant, il ne se charge que de communiquer les informations, rien de plus. Des protocoles de plus haut niveau ont donc été créés. Leur valeur ajoutée provient souvent du formatage et de la structuration des flux de données échangés.

La plupart des services réseau définissent donc un protocole qui leur est propre. Il est d'ailleurs assez courant de confondre le service et le protocole, tellement ils sont intrinsèquement liés. Ainsi, le service FTP utilise le protocole FTP (« File Transfer Protocol », protocole de transfert de fichiers), les serveurs Internet utilisent essentiellement le protocole HTTP (« Hyper Text Transfer Protocol », protocole de transfert d'hypertexte), le service de courrier électronique utilise les protocoles POP (« Post Office Protocol ») pour la réception des courriers et SMTP (« Simple Mail Transfer Protocol ») pour leur envoi. Cette liste de protocoles n'est pas exhaustive, et de nouveaux services apparaissent régulièrement. Les protocoles de haut niveau ne seront donc pas décrits dans ce document.

Les protocoles de haut niveau transmettent, en général, toutes leurs données en clair sur le réseau. Cela comprend non seulement les données applicatives, mais également les noms d'utilisateurs et les mots de passe. Toute personne ayant physiquement accès au réseau ou à une machine par laquelle les paquets passent peut donc, s'il le désire, récupérer tous les mots de passe avec une simplicité extrême (il existe même des programmes spécialisés pour cela). Il est donc inconcevable d'utiliser tous ces protocoles sans prendre de précautions particulières. Heureusement, il est possible d'encapsuler les protocoles dans un réseau virtuel qui, lui, transmet les données sous forme chiffrée. L'outil le plus utilisé de nos jours est ssh (abréviation de l'anglais « Secure SHell »), qui permet de se connecter sur une machine et de travailler à distance en toute sécurité. Il en existe plusieurs implémentations, dont au moins une libre : OpenSSH. Nous verrons comment configurer et utiliser OpenSSH pour améliorer la sécurité du réseau dans la [Section 9.5.2.2](#).

IX-B - Configuration du réseau sous Linux

La configuration du réseau nécessite donc la configuration du protocole IP et des services TCP, UDP et ICMP (entre autres). Cette opération se fait en définissant l'adresse IP le masque de sous-réseau et les routes à utiliser. Vient ensuite la configuration du nom de la machine locale, de son domaine, des noms de machines qu'elle peut résoudre elle-même et des serveurs de DNS qu'elle doit utiliser pour les autres noms.

Il est quasiment certain que votre distribution dispose d'un outil permettant d'effectuer la configuration du réseau simplement. Connaissant à présent la signification des termes utilisés dans les réseaux TCP/IP, vous devriez pouvoir parvenir à une configuration valide relativement simplement. Il est fortement recommandé de consulter la documentation de votre distribution. Les commandes de configuration du réseau sont souvent appelées dans les scripts de démarrage de la machine, ou dans les scripts de changement de niveau d'exécution. Toutefois, il peut être utile de connaître ces commandes, ne serait-ce que pour comprendre comment votre système fonctionne. Cette section a donc pour but de vous présenter ces commandes, ainsi que les principaux fichiers de configuration du réseau utilisés sous Linux.

IX-B-1 - Configuration statique des interfaces réseau

La principale commande permettant de configurer le réseau est la commande `ifconfig`. Comme son nom l'indique (« InterFace CONFiguration »), elle permet de configurer les interfaces réseau de la machine. Il faut savoir qu'il existe plusieurs types d'interfaces réseau. Les plus courants sont les trois types d'interfaces suivants :

- l'interface `loopback`, qui représente le réseau virtuel de la machine, et qui permet aux applications réseau d'une même machine de communiquer entre elles même si l'on ne dispose pas de carte réseau ;
- les interfaces des cartes réseau (que ce soient des cartes Ethernet, TokenRing ou autres) ;
- les interfaces `ppp`, `plip` ou `slip`, qui sont des interfaces permettant d'utiliser les connexions sérieelles, parallèles ou téléphoniques comme des réseaux.

La configuration d'une interface comprend l'initialisation des pilotes nécessaires à son fonctionnement et l'affectation d'une adresse IP à cette interface. La syntaxe générale que vous devrez utiliser est la suivante :

```
ifconfig interface adresse netmask masque up
```

où `interface` est le nom de l'interface réseau que vous voulez configurer, `adresse` est l'adresse IP que cette interface gèrera, et `masque` est le masque de sous-réseau que vous utilisez. Les interfaces que vous aurez à configurer seront certainement des interfaces Ethernet, auquel cas vous devrez utiliser les noms `eth0`, `eth1`, etc. dans la commande `ifconfig`. Si vous désirez configurer l'interface loopback, vous devrez utiliser le nom d'interface `lo`.

Le paramètre `up` donné à `ifconfig` lui indique que l'interface doit être activée. Cela signifie que dès que la commande `ifconfig` s'achèvera, votre interface réseau sera active et fonctionnelle. On ne peut faire plus simple... Bien entendu, il existe le paramètre inverse : `down`. Ce paramètre s'utilise tout simplement dans la commande `ifconfig` avec la syntaxe suivante :

```
ifconfig interface down
```

où `interface` est toujours le nom de l'interface.

Un exemple de configuration très classique est le suivant :

```
ifconfig eth0 192.168.0.1 netmask 255.255.255.0 up
```

Note : Prenez bien garde, lorsque vous écrivez vos adresses IP, à ne pas rajouter de 0 supplémentaire devant les nombres qui la constituent. En effet, il existe une convention en informatique qui dit que les nombres préfixés d'un 0 sont codés en octal, c'est-à-dire en base 8. Il va de soi qu'une adresse IP codée en octal n'utilise pas les mêmes nombres que lorsqu'elle est exprimée en décimal, aussi éprouveriez-vous quelques difficultés pour diagnostiquer le non-fonctionnement de votre réseau si vous faisiez cette erreur !

Le noyau utilisera par défaut le nombre 255 pour les adresses de broadcast dans les composantes de l'adresse IP qui ne fait pas partie de l'adresse de sous-réseau. Si vous désirez utiliser une autre adresse (en général, l'alternative est de prendre l'adresse du sous-réseau), vous devrez utiliser l'option `broadcast adresse` dans la commande `ifconfig`. Cependant, le comportement par défaut convient à la plupart des réseaux. La commande de configuration donnée en exemple ci-dessus sera alors :

```
ifconfig eth0 192.168.0.1 netmask 255.255.255.0 broadcast 192.168.0.0 up
```

Enfin, il est possible d'affecter plusieurs adresses IP à certaines interfaces réseau. C'est en particulier le cas pour toutes les interfaces réseau classiques, mais bien entendu cela n'est pas réalisable avec les interfaces de type point à point comme les interfaces des connexions ppp. Lorsqu'une interface dispose de plusieurs adresses, la première est considérée comme l'adresse principale de l'interface, et les suivantes comme des alias. Ces alias utilisent comme nom le nom de l'interface réseau principale et le numéro de l'alias, séparés par un deux-points (caractère `:`). Par exemple, si l'interface `eth0` dispose d'un alias, celui-ci sera nommé `eth0:0`. Ainsi, pour fixer l'adresse d'un alias d'une interface réseau, on utilisera la syntaxe suivante :

```
ifconfig interface:numéro add adresse netmask masque
```

où `interface` est toujours le nom de l'interface, `numéro` est le numéro de l'alias, `adresse` est l'adresse IP à attribuer à cet alias, et `masque` est le masque de sous-réseau de cette adresse.

La commande `ifconfig` est en général appelée dans les scripts d'initialisation du système, qui ont été générés par l'utilitaire de configuration réseau de votre distribution. Si vous effectuez un `grep` sur « `ifconfig` » dans le répertoire `/etc/rc.d/` (ou `/sbin/init.d/`, selon votre distribution), vous trouverez la commande de démarrage du réseau. Il se peut

qu'une variable d'environnement soit utilisée à la place de l'adresse IP que vous avez choisie. Quoi qu'il en soit, vous devez sans aucun doute avoir les lignes suivantes, qui permettent l'initialisation de l'interface loopback :

```
ifconfig lo 127.0.0.1 netmask 255.0.0.0 up
```

IX-B-2 - Définition des règles de routage

La deuxième étape dans la configuration du réseau est la définition des règles de routage. Il est possible de définir plusieurs règles de routage actives simultanément. L'ensemble de ces règles constitue ce qu'on appelle la table de routage. La règle utilisée est sélectionnée par le noyau en fonction de l'adresse destination du paquet à router. Chaque règle indique donc un critère de sélection sur les adresses, et l'interface vers laquelle doivent être transférés les paquets dont l'adresse destination vérifie cette règle.

La commande utilisée pour définir une route est, chose surprenante, la commande système `route`. Sa syntaxe est donnée ci-dessous :

```
route opération [-net | -host] adresse netmask masque interface
```

où `opération` est l'opération à effectuer sur la table de routage. L'opération la plus courante est simplement l'ajout d'une règle de routage, auquel cas `add` doit être utilisé. L'option suivante permet d'indiquer si le critère de sélection des paquets se fait sur l'adresse du réseau destination ou plus restrictivement sur l'adresse de la machine destination. En général, il est courant d'utiliser la sélection de toutes les adresses d'un même réseau et de les router vers une même interface. Dans tous les cas, `adresse` est l'adresse IP de la destination, que celle-ci soit un réseau ou une machine. Si la destination est un réseau, il faut indiquer le masque de sous-réseau `masque` à l'aide de l'option `netmask`. Enfin, `interface` est l'interface réseau vers laquelle doivent être envoyés les paquets qui vérifient les critères de sélection de cette règle.

Par exemple, la règle de routage à utiliser pour l'interface loopback est la suivante :

```
route add -net 127.0.0.0 netmask 255.0.0.0 lo
```

Cette règle signifie que tous les paquets dont l'adresse de destination appartient au sous-réseau de classe A 127.0.0.0 doivent être transférés vers l'interface loopback. Cela implique en particulier que les paquets à destination de la machine d'adresse IP 127.0.0.1 (c'est-à-dire la machine locale) seront envoyés vers l'interface loopback (ils reviendront donc sur la machine locale).

Une autre règle de routage classique est la suivante :

```
route add -net 192.168.0.0 netmask 255.255.255.0 eth0
```

Elle permet d'envoyer tous les paquets à destination du réseau de classe C 192.168.0.0 vers la première interface Ethernet. C'est typiquement ce genre de règle qu'il faut utiliser pour faire fonctionner un réseau local.

Il n'est normalement pas nécessaire d'ajouter les règles de routage pour les réseaux auxquels la machine est connectée. En effet, la configuration d'une carte réseau à l'aide de la commande `ifconfig` ajoute automatiquement à la table de routage une règle pour envoyer les paquets à destination de ce réseau par l'interface réseau qui y est connectée. Cependant, la commande `route` devient réellement nécessaire lorsqu'il faut définir les passerelles à utiliser pour l'envoi des paquets destinés à une machine à laquelle la machine locale ne peut accéder directement. Les règles de routage faisant intervenir une passerelle sont semblables aux règles de routage vues ci-dessus, à ceci près que l'adresse IP de la passerelle à utiliser doit être fournie. Pour cela, on utilise l'option `gw` (abréviation de l'anglais « Gateway »). La syntaxe utilisée est donc la suivante :

```
route add [-net | -host] adresse netmask masque gw passerelle interface
```

où passerelle est l'adresse IP de la passerelle à utiliser pour router les paquets qui vérifient les critères de cette règle. Les autres paramètres sont les mêmes que pour les règles de routage classique.

Par exemple, supposons qu'une machine soit connectée à un réseau d'adresse 192.168.0.0, et que sur ce réseau se trouve une passerelle d'adresse 192.168.0.1 permettant d'atteindre un autre réseau, dont l'adresse est 192.168.1.0. Une machine du réseau 192.168.0.0 aura typiquement les règles de routage suivantes :

```
route add -net 192.168.0.0 netmask 255.255.255.0 eth0
route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.0.1 eth0
```

La première règle permet, comme on l'a déjà vu, de communiquer avec toutes les machines du réseau local. La deuxième règle permet d'envoyer à la passerelle 192.168.0.1 tous les paquets à destination du réseau 192.168.1.0.

Inversement, si la passerelle utilise l'adresse 192.168.1.15 sur le réseau 192.168.1.0, les machines de ce réseau qui voudront accéder au réseau 192.168.0.0 devront spécifier la règle de routage suivante :

```
route add -net 192.168.0.0 netmask 255.255.255.0 gw 192.168.1.15 eth0
```

Bien entendu, il est nécessaire que toutes les machines des deux réseaux utilisent ces règles de routage pour que la communication entre les deux réseaux se fasse dans les deux sens.

Le problème de ces règles de routage est qu'elles spécifient l'adresse du réseau destination. Il est évidemment hors de question d'utiliser une règle de routage différente pour toutes les adresses de réseaux possibles. Il est donc possible de définir ce qu'on appelle une passerelle par défaut, qui n'est rien d'autre que la passerelle vers laquelle doivent être envoyés tous les paquets qui n'ont pas vérifié les critères des autres règles de routage. La syntaxe à utiliser pour définir la passerelle par défaut est plus simple, puisqu'il n'est plus nécessaire de préciser les critères de sélection :

```
route add default gw passerelle interface
```

où la signification des paramètres passerelle et interface est inchangée.

Ainsi, pour reprendre l'exemple précédent, supposons que la machine 192.168.0.47 dispose d'une connexion à Internet et soit configurée pour partager cette connexion avec les machines du réseau local. Pour que toutes les machines du réseau local puissent profiter de cette connexion, il suffit de demander à ce que tous les paquets qui ne vérifient aucune autre règle de routage soient envoyés à la passerelle 192.168.0.47. Cela se fait avec la règle de routage suivante :

```
route add default gw 192.168.0.47 eth0
```

IX-B-3 - Définition du nom de la machine

La configuration du nom d'un ordinateur n'est pas à proprement parler une opération indispensable, mais elle permet de le nommer de manière plus conviviale qu'en utilisant son adresse IP. La commande de base permettant de manipuler le nom de la machine locale est la commande `hostname`. Appelée telle quelle, elle renvoie le nom actuel de la machine :

```
hostname
```

Cette commande permet également de modifier ce nom, simplement avec la syntaxe suivante :

```
hostname nom
```

où `nom` est le nom à utiliser. Il est d'usage de n'utiliser que le nom de la machine, sans son domaine. Le nom de domaine est déterminé automatiquement par le système à partir des informations issues de la configuration de la résolution des noms de domaine. Nous verrons cela dans le paragraphe suivant.

Cette commande est donc très simple à utiliser, et elle est en général appelée dans les scripts de démarrage du système. La plupart des distributions utilisent le fichier `/etc/HOSTNAME` pour stocker le nom de la machine. Vous êtes bien entendu libre de choisir le nom que vous voulez pour votre ordinateur, mais vous devez vous assurer que ce nom est unique dans votre domaine !

IX-B-4 - Résolution des noms de domaine

La commande `hostname` ne permet de fixer que le nom de la machine locale. Pour les autres machines du réseau, il faut mettre en place les mécanismes de résolution de noms de domaine. Comme il l'a déjà été indiqué au début de ce chapitre, il existe deux solutions pour trouver l'adresse IP d'une machine à partir de son nom : la consultation d'une liste de noms stockée en local, soit l'interrogation d'un serveur de noms de domaine.

Le fichier `/etc/host.conf` permet de définir le comportement du système lors de la résolution d'un nom. Sa structure est très simple, puisqu'il y a une option de recherche par ligne. Dans la plupart des cas, les lignes suivantes sont suffisantes :

```
order hosts,bind
multi on
```

Elles permettent d'indiquer que la recherche des noms pour leur résolution doit se faire d'abord localement, puis par appel aux DNS si la recherche précédente a échoué. C'est en général le comportement désiré. La deuxième ligne permet de faire en sorte que toutes les adresses correspondant à une machine soient renvoyées. Si l'on avait utilisé l'option `multi off`, seule la première adresse IP trouvée aurait été renvoyée.

La liste de noms locale est stockée dans le fichier `/etc/hosts` (cela explique le nom `hosts` utilisé pour l'option `order` dans le fichier `/etc/host.conf`). Votre ordinateur connaîtra directement l'adresse IP de toutes les machines référencées dans ce fichier. Il est bon de placer ici une entrée pour les ordinateurs les plus couramment utilisés sur le réseau. Chaque entrée commence par une adresse IP, et est suivie de la liste des noms de la machine possédant cette adresse, séparés par des espaces. Pour ceux qui ne disposent pas de réseau local, ce fichier doit être relativement simple : seule la ligne affectant l'adresse `127.0.0.1` à la machine locale (appelée « `localhost` ») doit s'y trouver.

```
127.0.0.1    localhost
```

De la même manière, le fichier `/etc/networks` contient les adresses des réseaux. Ce fichier est utilisé par la commande `route` pour donner un nom aux différents réseaux. Chaque entrée est constituée du nom du réseau, suivi de son adresse IP. Encore une fois, ce fichier se réduit à sa plus simple expression pour ceux qui n'ont pas de réseau local, puisqu'il ne contiendra tout au plus qu'une entrée pour le réseau « `loopback` », sur lequel se trouve l'adresse de retour `127.0.0.1`. Cette entrée aura donc la forme suivante :

```
loopback    127.0.0.0
```

La configuration des serveurs de noms est en revanche une opération nécessaire si l'on désire accéder à des machines dont on ne connaît que le nom. Le fichier de configuration utilisé est cette fois le fichier `/etc/resolv.conf`. Sa structure est encore une fois très simple, avec une option par ligne, chaque option étant introduite par un mot clé.

Le mot clé `domain` permet d'indiquer le nom du domaine dont fait partie votre machine. Par exemple, si votre nom de domaine est « `andromede.galaxie` », vous devrez utiliser la ligne suivante :

```
domain andromede.galaxie
```

Le mot clé `search` permet quant à lui de spécifier une liste de noms de domaines à ajouter par défaut aux noms de machines non complètement qualifiés. Les éléments de cette liste doivent être séparés par des espaces. La recherche d'une machine dont le nom ne comprend pas la partie de nom de domaine s'effectue en ajoutant au nom de la machine les noms des domaines indiqués ici, jusqu'à ce que la résolution du nom en adresse IP réussisse. Cette recherche commence bien entendu par le nom de domaine local, s'il a été défini. Il est donc recommandé d'indiquer

votre nom de domaine dans cette liste de noms de domaines. Par exemple, si votre machine fait partie du domaine « andromede.galaxie », vous devrez utiliser la ligne suivante :

```
search andromede.galaxie
```

Ainsi, si vous recherchez l'adresse de la machine « krypton », la requête au DNS se fera avec le nom complètement qualifié « krypton.andromede.galaxie ». Vous êtes bien entendu libre d'ajouter d'autres noms de domaines pour le cas où la résolution de nom échouerait sur ce domaine.

Enfin, l'option `nameserver` est essentielle, puisqu'elle permet de donner les adresses IP des serveurs de DNS auxquels doivent être adressées les requêtes de résolution de noms. Par exemple, si vous disposez de deux serveurs DNS, un primaire, d'adresse 192.168.0.10, et un secondaire, d'adresse 192.168.0.15, vous utiliserez la ligne suivante :

```
nameserver 192.168.0.10 192.168.0.15
```

Cette ligne est évidemment obligatoire, faute de quoi la résolution des noms de machines en adresse IP échouera pour toute machine qui ne se trouve pas dans votre fichier `/etc/hosts`.

IX-B-5 - Utilisation des protocoles DHCP et BOOTP

Généralement, la gestion des adresses IP des machines devient rapidement une tâche difficile sur les grands réseaux, pour trois raisons. Premièrement, il faut toujours s'assurer que chaque machine dispose bien d'une adresse qui lui est propre, ce qui peut être difficile si l'on ne s'organise pas en conséquence. Deuxièmement, il faut que les fichiers `/etc/hosts` de toutes les machines soient à jour, ce qui nécessite un travail proportionnel au nombre de machines administrées. Enfin, le nombre d'adresses IP disponibles peut se réduire, ce qui peut devenir gênant à terme.

Afin de résoudre ces problèmes de configuration réseau, le protocole DHCP (abréviation de l'anglais « Dynamic Host Configuration Protocol ») a été défini. Il s'agit d'un protocole qui permet aux machines connectées sur un réseau d'interroger un « serveur d'adresses » du réseau capable de gérer une liste d'adresses et de les affecter dynamiquement aux machines du réseau. En fait, ce protocole permet de fournir plus d'informations que les simples adresses IP, par exemple la route par défaut que les machines doivent utiliser, ainsi que les adresses des serveurs de noms du réseau.

Un autre protocole semblable à DHCP a également été développé dans le but de permettre la configuration réseau des machines dès leur démarrage : le protocole BOOTP (abréviation de l'anglais « BOOTstrap Protocol »). Ce protocole est évidemment plus léger que DHCP, mais permet aux machines d'obtenir dynamiquement leur configuration réseau dès leur démarrage, avant même que ne soient montés les systèmes de fichiers. Ce protocole est donc particulièrement utile pour faire démarrer des machines sans disque, pour lesquelles le système de fichiers racine est monté en NFS.

La manière la plus simple de configurer les protocoles DHCP et BOOTP sur un poste client est d'utiliser les fonctionnalités d'autoconfiguration du noyau. Cependant, il est également possible d'effectuer cette configuration au niveau utilisateur à l'aide de programmes complémentaires. Les deux sections suivantes décrivent ces deux techniques.

IX-B-5-a - Autoconfiguration des clients DHCP et BOOTP

La configuration des protocoles DHCP et BOOTP ne comporte aucune difficulté particulière lorsque l'on utilise les fonctionnalités d'autoconfiguration du noyau. Ces fonctionnalités étant prises en charge au niveau du noyau, il va vous falloir recompiler un nouveau noyau pour en bénéficier. Cette opération en revanche est relativement technique, et doit être faite avec soin. La manière de procéder a été décrite en détail dans la [Section 7.3](#).

L'option à activer pour permettre l'utilisation du protocole BOOTP est l'option « IP: kernel level autoconfiguration » du menu « Networking options ». Cette option vous permettra de sélectionner le protocole d'autoconfiguration réseau

que le noyau devra utiliser lors de son amorçage. Vous devrez alors choisir l'option « IP: DHCP support (NEW) » ou l'option « IP: BOOTP support (NEW) » pour activer respectivement les protocoles DHCP et BOOTP.

Note : Vous remarquerez qu'il existe également un autre protocole d'autoconfiguration du réseau au démarrage : le protocole RARP. Ce protocole fournit les mêmes services que le protocole BOOTP, mais est plus ancien. Il n'est donc plus conseillé de l'utiliser, sauf vous vous trouvez sur un réseau pour lequel seul le protocole RARP est disponible.

Une fois ces options sélectionnées, vous pourrez recompiler votre noyau, l'installer et redémarrer la machine. Lors du démarrage, le noyau doit chercher un serveur DHCP ou un serveur BOOTP sur le réseau local pour effectuer la configuration réseau de votre carte réseau.

Note : Vous devrez peut-être également activer les options « NFS file system support » et « Root file system on NFS » du menu « Network File Systems » si vous désirez faire démarrer votre machine sur un système de fichiers racine monté en NFS lors du démarrage.

IX-B-5-b - Configuration d'un client DHCP au niveau utilisateur

Il est également possible de configurer les clients DHCP au niveau utilisateur, à l'aide de programmes complémentaires. Comme sur la plupart des machines Unix, le programme à utiliser est le programme `dhclient`. Ce programme est généralement lancé dans les scripts de démarrage des machines, et envoie des paquets de demandes de configuration sur le réseau à l'adresse de diffusion générale 255.255.255.255. Ces paquets sont donc susceptibles d'être captés par toutes les machines du réseau, mais seuls les serveurs DHCP y répondent. Les réponses obtenues sont alors analysées par `dhclient`, qui configure en conséquence l'interface réseau et passe ensuite en arrière-plan.

Les informations envoyées par les serveurs DHCP peuvent être plus ou moins complètes, la base étant bien sûr l'adresse IP de la machine et son masque de sous-réseau. Il est toutefois possible de donner plus d'informations, par exemple les adresses des serveurs de noms, des routes par défaut et des passerelles à utiliser.

Les adresses IP attribuées aux clients ne sont pas permanentes, car le protocole DHCP est avant tout destiné à la configuration automatique des postes itinérants ou susceptibles de redémarrer souvent. Par conséquent, ces adresses sont fournies dans le cadre d'un bail, dont la durée maximum est fixée par le serveur. Dès que le bail obtenu par un client expire, celui-ci doit chercher à le renouveler. C'est encore le programme `dhclient` qui s'en charge. C'est la raison pour laquelle celui-ci passe en arrière-plan après avoir configuré l'interface pour la première fois : il attend la fin des baux de la machine sur laquelle il tourne et cherche à les renouveler. Si un client ne renouvelle pas ce bail (parce qu'il est arrêté par exemple), le serveur DHCP peut réutiliser son adresse IP et l'affecter à une autre machine. Bien que les serveurs DHCP s'efforcent généralement de conserver les adresses IP des clients à chaque bail, un client configuré par DHCP ne peut donc pas considérer que son adresse IP restera toujours la même. C'est la contrepartie de la flexibilité.

Si aucun serveur DHCP ne peut être contacté lors du démarrage, `dhclient` abandonne temporairement et réessaie au bout d'un temps aléatoire. Au bout d'un certain nombre d'essais non fructueux, il peut décider de configurer les interfaces réseau avec les adresses IP des anciens baux obtenus par la machine. Pour cela, il mémorise dans le fichier de configuration `/var/db/dhclient.lease` les adresses IP de ces anciens baux. Ce fichier est périodiquement réécrit avec la liste des adresses des baux valides afin d'éviter qu'il ne se remplisse *ad vitam eternam*.

Bien entendu, les postes clients ne peuvent pas choisir leurs adresses IP sans vérification d'unicité. Dans le cas de l'absence de serveur DHCP (et donc d'autorité centrale), les clients qui démarrent interrogent les machines déjà présentes sur le réseau pour déterminer si l'adresse qu'ils envisagent de prendre est bien libre. Dans le cas contraire, une autre adresse est essayée, et ainsi de suite. Ainsi, même en cas de panne de tous les serveurs DHCP d'un réseau, les postes clients peuvent toujours travailler ensemble sans conflit d'adresses IP.

Comme vous pouvez le constater, le comportement de `dhclient` est relativement complexe et dépend de nombre de paramètres. Tous ces paramètres peuvent être définis dans le fichier de configuration `/etc/dhclient.conf`. Ce fichier contient en particulier les différentes durées intervenant dans le choix des adresses IP, par exemple la durée minimale d'un bail, les durées entre chaque tentative de configuration, les informations qui doivent être récupérées des serveurs

DHCP, ainsi que les valeurs par défaut pour ces informations lorsque les serveurs ne les fournissent pas. Le fichier de configuration `dhclient.conf` est donc relativement complexe. Heureusement, `dhclient` utilise des options par défaut qui conviennent dans la plupart des cas, aussi est-il fortement probable que votre fichier `dhclient.conf` soit vide. Si toutefois vous désirez en savoir plus, vous pouvez consulter la page de manuel `dhclient.conf`.

La configuration DHCP pour les postes clients se réduit donc à l'essentiel : le lancement de `dhclient`. Celui-ci s'utilise avec la syntaxe suivante :

```
dhclient interface0 [interface1 [&#8230;]]
```

où `interface0`, `interface1`, etc., sont les interfaces réseau qui doivent être configurées par DHCP. On ne peut donc pas faire plus simple...

Note : Sous Linux, le programme `dhclient` utilise les fonctionnalités d'accès direct aux cartes réseau et de filtrage des paquets. Ces deux fonctionnalités peuvent être activées dans la configuration du noyau à l'aide des options « Packet socket », « Packet socket: mmaped IO » et « Socket Filtering » du menu « Networking options ». L'option « IP: multicasting » de la liste des options du protocole IP devra également être activée. Le détail de la configuration et de la compilation du noyau a été vu dans la [Section 7.3](#).

Le programme `dhclient` est assez facétieux depuis quelques versions. S'il refuse obstinément de fonctionner, vous devrez sans doute vous rabattre vers le programme `dhpcd`, beaucoup plus simple, mais également beaucoup plus fiable. La plupart des distributions utilisent de dernier en lieu et place de `dhclient`. Consultez la documentation de votre distribution pour déterminer la solution qu'elle utilise.

IX-B-6 - Définition des protocoles de haut niveau

Comme nous l'avons vu plus haut, le protocole IP fournit les mécanismes de base pour la transmission des paquets. Plusieurs protocoles de plus haut niveau ont été définis pour fournir des services à valeur ajoutée, qui satisfont donc plus aux besoins des applications. Tous ces protocoles sont encapsulés dans le protocole IP, ce qui signifie que leurs informations sont transmises en tant que données dans les paquets IP.

En réalité, les paquets du protocole IP contiennent un champ permettant de signaler le type de protocole de haut niveau dont ils transportent les informations. À chaque protocole est donc attribué un identificateur numérique qui lui est propre, et qui permet aux services réseau d'interpréter les données des paquets.

Tout comme les adresses IP, ces numéros identifiant les protocoles ne sont pas très intéressants pour les humains, qui leur préfèrent évidemment le nom du protocole. Certains programmes réseau utilisent donc ces noms pour identifier les protocoles. Pour leur permettre de faire l'association entre ces noms et les identificateurs numériques, le fichier `/etc/protocols` est utilisé.

Le format de ce fichier est très simple. Il contient une ligne pour chaque protocole, constituée du nom du protocole, de la valeur de son identificateur, et des autres noms que ce protocole peut avoir (c'est-à-dire ses alias). Parmi ces protocoles, les plus importants sont les suivants :

Nom	Numéro	Alias
ip	0	IP
icmp	1	ICMP
tcp	6	TCP
udp	17	UDP

Comme on le voit, le protocole IP dispose lui-même d'un identificateur de protocole (qui vaut 0 en l'occurrence). Cet identificateur est un identificateur de « pseudo protocole », parce qu'IP est en fait le protocole de base. ICMP est identifié par le numéro 1, TCP par le numéro 6 et UDP par le numéro 17. Il existe beaucoup d'autres protocoles,

qui ne seront pas décrits ici. Bien entendu, le fichier `/etc/protocols` fourni avec votre distribution doit déjà contenir la définition de la plupart des protocoles.

De la même manière, la plupart des ports TCP et UDP sont affectés à des services bien définis, et certains programmes réseau peuvent chercher à faire la correspondance entre les noms de ces services et les numéros de ports. Cette correspondance est stockée dans le fichier de configuration `/etc/services`.

Les informations concernant les services sont données à raison d'une ligne par service. Chaque ligne suit la syntaxe suivante :

nom	port/protocole	alias

où `nom` est le nom du service décrit par cette ligne, `port` est le numéro du port utilisé par ce service, `protocole` est le nom du protocole utilisé par ce service (ce peut être « `tcp` » ou « `udp` »), et `alias` la liste des autres noms sous lesquels ce service est également connu.

Vous trouverez les principaux services dans l'extrait donné ci-dessous :

Service	Port/Protocole
ftp	21/tcp
telnet	23/tcp
smtp	25/tcp
pop3	110/tcp
irc	194/tcp
irc	194/udp

Comme vous pouvez le constater, ces services n'ont pas d'alias. Ces informations sont données uniquement à titre d'exemple. Il va de soi que le fichier `/etc/services` fourni avec votre distribution contient la définition d'un grand nombre de services, et vous n'aurez en général pas à y toucher.

IX-B-7 - Les super-démons inetd et xinetd

La plupart des services réseau sont gérés par des programmes qui s'exécutent en permanence et qui attendent des connexions sur un port TCP ou UDP. Ces programmes consomment relativement peu de ressources, car ils passent la plupart de leur temps à attendre ces connexions. Ils ne se réveillent que lorsqu'un client se connecte effectivement et leur envoie une requête.

Cependant, ils peuvent être relativement nombreux, et si tous les services sont lancés simultanément, ils peuvent finir par consommer une part non négligeable des ressources système. C'est pour cette raison que les super-démons `inetd` (de l'anglais « InterNET Daemon ») et `xinetd` (de l'anglais « eXtended INETD ») ont été créés. Ces démons sont à l'écoute des demandes de connexion des clients pour les autres services réseau, et ne lancent ceux-ci que lorsqu'un client se connecte sur leurs ports. Une fois lancés, les véritables démons reprennent la main et communiquent directement avec leurs clients. Ainsi, `inetd` et `xinetd` écoutent les ports pour tout le monde, et sont la plupart du temps les seuls à fonctionner. Les ressources système sont donc économisées et les services réseau sont démarrés et arrêtés à la demande.

Le super-démon `xinetd` est appelé à remplacer `inetd`, qui lui est beaucoup plus ancien et nettement moins souple. En pratique, un seul de ces super-démons doit être démarré sur une machine (il est inutile de les lancer tous les deux, car les clients ne peuvent se connecter qu'à un seul d'entre eux de toute manière). Les deux sections suivantes décrivent la configuration de ces super-démons.

IX-B-7-a - Le super-démon inetd

Le super-démon inetd laisse de plus en plus la main au nouveau super-démon xinetd qui est beaucoup plus puissant, mais reste toutefois très utilisé sur de nombreuses machines. Sa description n'est donc pas inutile, car toutes les distributions n'utilisent pas encore xinetd.

Le super-démon inetd utilise le fichier de configuration `/etc/inetd.conf` pour déterminer les ports sur lesquels il doit attendre des connexions de la part des clients, et pour trouver le service réseau qu'il doit lancer lorsqu'une telle connexion arrive. Ce fichier est structuré en lignes, dont chacune décrit un des services que le démon inetd prend en charge. Les informations données sur ces lignes sont les suivantes :

- le nom du service (tel qu'il est défini dans la première colonne du fichier `/etc/services`) dont inetd doit surveiller les requêtes ;
- le type de canal de communication réseau utilisé, en général « stream » (pour les communications en mode connecté, donc en général celles qui utilisent le protocole TCP) ou « dgram » (pour les communications basées sur les datagrammes, donc typiquement les communications utilisant le protocole UDP) ;
- le protocole réseau utilisé (« tcp » ou « udp ») par ce service ;
- l'un des mots clés `wait` ou `nowait`, qui permettent d'indiquer si inetd doit attendre la fin de l'exécution du démon gérant le service ou s'il peut attendre de nouvelles requêtes de la part des clients ;
- le nom de l'utilisateur au nom duquel le démon gérant ce service doit fonctionner (en général, c'est l'utilisateur `root`) ;
- le chemin sur le fichier exécutable de ce démon ;
- les éventuels paramètres en ligne de commande pour ce démon, en commençant par l'argument 0, qui doit toujours être le nom du fichier exécutable du programme lui-même.

Par exemple, la ligne suivante permet de lancer le démon `telnetd` sur toute requête via le protocole TCP pour le service `telnet` :

```
telnet stream tcp nowait root /usr/sbin/in.telnetd in.telnetd
```

Il est supposé ici que le démon en charge de ce service peut être lancé avec le programme `/usr/sbin/in.telnetd`.

Le démon inetd est capable de fournir lui-même un certain nombre de services de base, et il n'est pas nécessaire de fournir un démon pour ces services. Dans ce cas, il faut utiliser le mot clé `internal` à la place du nom du fichier exécutable du démon de ce service. Les paramètres doivent également être remplacés par le mot clé `internal`.

En fait, il est fort peu probable que votre fichier de configuration `/etc/inetd.conf` définisse les services comme indiqué dans cette section. En effet, un programme intermédiaire en charge d'assurer des contrôles de sécurité est souvent intercalé entre inetd et les démons gérant les services. Nous verrons ce que fait exactement ce programme dans une prochaine section.

IX-B-7-b - Le super-démon xinetd

Le super-démon xinetd utilise un autre fichier de configuration que celui du super-démon inetd. Pour xinetd, la définition des services mis à disposition des clients se fait dans le fichier de configuration `/etc/xinetd.conf`. Toutefois, contrairement au fichier `inetd.conf`, le fichier `xinetd.conf` peut inclure d'autres fichiers de configuration et référencer des répertoires contenant les fichiers de configuration spécifiques aux services. Ainsi, la configuration des services est beaucoup plus modulaire et se fait plus facilement.

En pratique, il est d'usage de définir les options par défaut pour tous les services dans le fichier de configuration `/etc/xinetd.conf`, et de décrire les différents services dans des fichiers complémentaires stockés dans le répertoire `/etc/xinetd.d/`. Ce répertoire est alors inclus directement dans le fichier `xinetd.conf` à l'aide de la directive `includedir` dédiée à cet usage.

Les fichiers de configuration de xinetd sont constitués de sections permettant de décrire les services pris en charge, ou tout simplement les options par défaut applicables à tous les services. La forme générale de ces sections est la suivante :

```
service
{
    option opérateur valeur [valeur [#8230;]]
    option opérateur valeur [valeur [#8230;]]
    [#8230;
}
```

où `service` est le nom du service (ou `defaults` pour les options par défaut), `option` est un mot-clé identifiant une des options de ce service, et `opérateur` est l'un des opérateurs '=' (pour la définition de la valeur d'une option ou l'écrasement de sa valeur précédente), '+=' (pour compléter la liste de valeurs d'une option) ou '-=' (pour supprimer une valeur de la liste de valeurs d'une option). Les valeurs des options peuvent être multiples, et leur nature dépend des options utilisées.

Les principales options utilisables dans ces sections sont les suivantes :

Option	Signification
id	Identificateur du service, pour le cas où plusieurs sections devraient être définies pour un même service. Cette possibilité est utilisée lorsque l'on désire fournir des options différentes pour un même service. L'entrée choisie (et donc le jeu d'options choisi) dépend dans ce cas de critères définis par exemple sur l'interface réseau ou sur les adresses des clients.
type	Permet d'indiquer la nature du service décrit par cette section. Généralement, cette option n'a pas à être donnée, sauf lorsque l'on veut forcer la méthode d'implémentation d'un service. Par exemple, il est possible de donner la valeur <code>INTERNAL</code> à cette option pour utiliser l'un des services implémentés par xinetd en interne.
server	Permet de donner le chemin sur l'exécutable du démon prenant en charge le service décrit.
server_args	Permet de donner les paramètres en ligne de commande du démon prenant en charge le service. Notez que, contrairement à ce qui se fait avec inetd, il ne faut généralement pas donner le nom de l'exécutable en premier argument dans cette option. Cette règle peut toutefois être rendue fautive en ajoutant la valeur <code>NAMEINARGS</code> dans l'option <code>flags</code> décrite ci-dessous.
socket_type	Le type de canal de communication utilisé (« <code>stream</code> » ou « <code>dgram</code> », comme pour inetd).
protocol	Le protocole réseau utilisé par le service (« <code>tcp</code> » ou « <code>udp</code> », comme pour inetd).
port	Le port sur lequel le service peut être trouvé. Cette option est facultative. Si elle n'est pas indiquée, le numéro de port utilisé sera celui indiqué dans le fichier de configuration <code>/etc/services</code> pour le service en cours de configuration.
wait	L'indicateur de capacité du démon a géré plusieurs connexions simultanément. Les valeurs que l'on peut donner à cette option sont « <code>yes</code> » et « <code>no</code> », respectivement pour indiquer que le démon peut gérer plusieurs connexions simultanément ou non. Dans ce dernier cas, le démon xinetd lancera plusieurs instances du démon gérant le service si plusieurs clients cherchent à se connecter simultanément.

	Le nombre maximum d'instances peut toutefois être contrôlé à l'aide des options <code>instances</code> et <code>per_source</code> décrites ci-dessous.
<code>flags</code>	Les paramètres permettant de contrôler différents aspects du service. Les options les plus courantes sont « <code>IDONLY</code> », qui permet de n'autoriser que les connexions provenant de machines disposant d'un serveur d'identification des clients, « <code>NORETRY</code> », qui permet d'éviter de réessayer de lancer le démon du service si le lancement précédent a échoué, et « <code>NAMEINARGS</code> », qui permet d'indiquer que le nom de l'exécutable doit être spécifié en premier paramètre dans l'option <code>server_args</code> .
<code>user</code>	Le compte utilisateur dans lequel le démon prenant en charge le service doit être lancé. Cette option ne peut bien entendu pas être utilisée pour les services gérés en interne par <code>xinetd</code> .
<code>interface</code>	L'interface à laquelle le service est attaché. Cette option permet de définir plusieurs configurations pour un même service et d'affecter ces différentes configurations à des interfaces réseau distinctes. Pour l'heure, seule l'adresse IP de l'interface réseau peut être spécifiée grâce à cette option, ce qui impose d'avoir des adresses IP fixes.
<code>only_from</code>	La liste des adresses des machines autorisées à se connecter. Il est possible de spécifier les adresses IP explicitement ou à l'aide d'une adresse et d'un masque de sous-réseau. Les noms de domaines peuvent également être utilisés. Dans ce cas, le nom n'est pas transformé en adresse IP. Au contraire, c'est l'adresse du client qui est retransformée en nom de machine pour vérifier s'il a le droit de se connecter. Notez que l'absence de ce champ indique que, par défaut, l'accès est accordé à toutes les machines (sauf celles explicitement interdites de connexion par l'option <code>no_access</code> décrite ci-dessous). En revanche, la présence de ce champ, mais sans valeur permet d'interdire l'accès à toutes les machines.
<code>no_access</code>	La liste des adresses des machines qui n'ont pas le droit de se connecter. Les adresses peuvent être spécifiées de la même manière que pour l'option <code>only_from</code> . Si une adresse vérifie les critères des deux options <code>only_from</code> et <code>no_access</code> , c'est l'option dont le critère est le plus précis qui est choisie.
<code>access_times</code>	La période pendant laquelle le service est accessible. Cette période peut être exprimée sous la forme d'une liste d'intervalles « <code>heure:minute-heure:minute</code> ».
<code>instances</code>	Permet d'indiquer le nombre maximum d'instances d'un même démon que <code>xinetd</code> peut lancer. Cette option permet donc de spécifier un nombre de connexions maximum pour chaque service.
<code>per_source</code>	Permet d'indiquer le nombre maximum d'instances d'un même démon que <code>xinetd</code> peut lancer pour un même client (identifié par son adresse IP). Cette option permet donc de limiter le nombre de connexions d'un client, de manière indépendante du nombre de connexions total donné par l'option <code>instances</code> .
<code>cps</code>	Permet de limiter dans le temps le nombre de connexions entrantes pour un service, afin d'éviter les attaques par déni de service. Cette option prend deux paramètres, le premier

	étant le nombre maximum de demandes de connexion par seconde que xinetd peut accepter. Si ce nombre est dépassé, le service est désactivé pendant le nombre de secondes indiqué par le deuxième paramètre.
disabled	Permet de désactiver globalement des services, en indiquant leurs noms en paramètre. Par défaut, aucun service n'est désactivé. Cette option ne peut être utilisée que dans la section defaults, car elle permet de désactiver globalement et rapidement un ensemble de services.
enabled	Permet de donner la liste des services pris en charge. Cette option fonctionne de manière similaire à l'option disabled, à ceci près qu'elle fonctionne en logique inverse. L'absence de cette option implique l'activation de tous les services, sauf ceux qui sont listés dans l'option disabled. En revanche, dès que cette option est définie, seuls les services listés sont démarrés. Tout comme l'option disabled, cette option ne peut être utilisée que dans la section globale defaults.
disable	Permet de désactiver les services de manière unitaire. Cette option est utilisée dans les sections des services, afin d'indiquer de manière plus fine qu'avec les options enabled et disabled s'ils doivent être activés ou non. Cette option peut prendre deux valeurs : « yes » ou « no ». La première permet de désactiver le service et la deuxième de le garder fonctionnel. Notez que les options enabled et disabled ont priorité l'option disable. Ainsi, un service désactivé par l'une de ces options ne peut pas être réactivé en attribuant la valeur no à l'option disable.
log_type	Méthode d'enregistrement des événements du démon xinetd. Il est possible d'utiliser le démon syslog pour enregistrer les événements de connexion et de déconnexion des utilisateurs, ou directement un fichier texte. Dans le premier cas, il faut utiliser la valeur SYSLOG suivie de la classe d'enregistrement (généralement daemon) et du niveau de trace (généralement info). Dans le deuxième cas, il faut spécifier la valeur FILE et indiquer les limites basse et haute de la taille du fichier au-delà desquelles un avertissement est généré, puis les traces sont arrêtées.
log_on_success	Informations enregistrées lors de l'acceptation d'une connexion. xinetd peut enregistrer différentes informations lorsqu'une nouvelle connexion est acceptée. Ces informations sont indiquées grâce à la liste des mots-clés spécifiés par cette option. Les mots-clés utilisables sont « PID », pour enregistrer l'identifiant de processus du démon qui traitera la requête, « HOST », pour enregistrer le nom de la machine cliente, « USERID », pour enregistrer le nom de l'utilisateur (si celui-ci peut être obtenu par le service d'identification de sa machine), « EXIT », pour enregistrer la terminaison du démon qui traite la requête, et « DURATION », pour enregistrer la durée de la connexion.
log_on_failure	Informations enregistrées lors d'un refus de connexion. Les informations enregistrées lors d'un refus de connexion peuvent être spécifiées à l'aide de cette option, de la même manière que les informations de connexion peuvent être enregistrées à l'aide de l'option log_on_success. Les mots-clés utilisables avec cette option sont « ATTEMPT », pour signaler simplement que la tentative a échoué, « HOST », pour enregistrer le nom de la machine depuis

laquelle la tentative a été effectuée, et « **USERID** », pour enregistrer le nom de l'utilisateur tel qu'indiqué par le service d'identification de sa machine.

Toutes les options disponibles ne sont pas décrites dans le tableau précédent. Vous pourrez obtenir de plus amples renseignements en consultant la page de manuel `xinetd.conf`.

La section « **defaults** » permet de définir les options par défaut applicables à tous les services. Les options qui y sont définies peuvent toutefois être redéfinies ou précisées à l'aide des opérateurs `=`, `+=` et `-=` dans les sections des différents services. Les options utilisables dans la section **defaults** et que l'on a vu ci-dessus sont les options `bind`, `log_type`, `log_on_success`, `log_on_failure`, `instances`, `per_source`, `only_from`, `no_access`, `disabled` et `enabled`.

Généralement, le fichier `/etc/xinetd.conf` ne contient que la section **default** et une ligne d'inclusion du répertoire `/etc/xinetd.d/`, dans lequel se trouvent les fichiers de configuration des différents services. L'exemple suivant présente donc un fichier `xinetd.conf` typique :

```
# Définition des options par défaut :
defaults
{
    # On interdit la connexion à tout le monde par défaut :
    only_from =

    # On désactive les services internes :
    disabled = echo time daytime chargen discard

    # On désactive les services d'administration de xinetd :
    disabled += servers services xadmin

    # On limite le nombre d'instances total des services :
    instances = 15

    # On définit les règles de suivi des connexions :
    log_type      = FILE /var/log/servicelog
    log_on_success = HOST PID USERID DURATION EXIT
    log_on_failure = HOST USERID RECORD
}

# Inclusion des fichiers de configuration des services :
includedir /etc/xinetd.d
```

L'exemple suivant présente la définition du service `telnet` située dans le fichier de configuration `/etc/xinetd.d/telnet` :

```
service telnet
{
    socket_type      = stream
    # Le numéro de port et le protocole utilisés peuvent être omis
    #, car ils sont déjà définis dans /etc/services.

    server           = /usr/sbin/in.telnetd
    user             = root
    wait            = no
    only_from        = localhost
    disable          = no
}
```

Note : `xinetd` n'inclue pas les fichiers contenant un point (caractère `.`) ni ceux finissant par un tilde (caractère `~`). En particulier, on évitera de mettre une extension aux fichiers de configuration des services, puisque dans ce cas ils contiendraient un point et ne seraient pas lus.

IX-C - Configuration de la connexion à Internet

Les connexions à Internet font partie des connexions réseau temporaires permettant de relier deux machines. La machine locale se connecte en effet au serveur du fournisseur d'accès à Internet via la ligne téléphonique. Bien entendu, ce serveur est considéré comme la passerelle par défaut pour tous les paquets à destination d'Internet. Ce type de connexion a de particulier que les adresses IP des deux machines sont, en général, déterminées dynamiquement, lors de l'établissement de la connexion. La configuration réseau des connexions à Internet se fait donc légèrement différemment de la configuration d'un réseau local normal. Heureusement, tous les détails de la configuration réseau sont pris en charge automatiquement.

IX-C-1 - Le protocole PPP

Les connexions à Internet utilisent le protocole PPP (abréviation de l'anglais « Point to Point Protocol »), qui permet à deux ordinateurs d'échanger des paquets TCP/IP au travers d'un canal de communication simple (donc, en particulier, au travers d'une ligne téléphonique en utilisant un modem). Ce protocole est géré par le noyau et par le démon `pppd`. Il permet d'effectuer la négociation initiale entre les deux machines, ce qui comprend notamment l'identification, l'authentification et l'attribution des adresses IP.

Le démon `pppd` ne gère pas le modem directement, car il est supposé être utilisable sur d'autres types de lignes que les lignes téléphoniques. La gestion du modem est donc relayée à un autre programme, qui se nomme `chat` (il tient son nom du fait qu'il permet de dialoguer directement avec le modem). Le programme `chat` ne connaît pas les différents types de modems qui existent sur le marché, il se contente d'exécuter des scripts qui décrivent les informations à envoyer au modem et les réponses attendues en retour. Cela permet de reporter la configuration spécifique aux modems dans des scripts de connexion. L'écriture de ces scripts nécessite bien entendu de connaître les commandes que votre modem est capable de comprendre. Notez également qu'il est indispensable ici que votre modem soit un vrai modem (qu'il soit interne ou externe), et non pas un modem logiciel (ces modems sont des modèles bas de gamme qui nécessitent un support logiciel pour interpréter les informations analogiques reçues sur la ligne téléphonique). Renseignez-vous donc bien sur la nature du modem que vous achetez, si vous voulez ne pas avoir de problèmes sous Linux...

La séquence des opérations lors d'une connexion est donc l'initialisation du modem et l'établissement de l'appel téléphonique par le programme `chat`, puis le démarrage du démon `pppd`. Celui-ci effectue la connexion sur le serveur du fournisseur d'accès et détermine l'adresse IP, les adresses des DNS, la passerelle et la route à utiliser pour cette connexion. Une fois cela réalisé, toutes les fonctionnalités réseau peuvent être utilisées via Internet, et votre ordinateur fait alors partie du réseau mondial.

L'identification et l'authentification peuvent se faire de différentes manières selon le fournisseur d'accès à Internet utilisé. Un certain nombre de fournisseurs exige l'authentification lors de l'appel téléphonique, avec un mécanisme de login classique. Pour ces fournisseurs, l'authentification est faite par le programme `chat`, auquel il faut communiquer le nom d'utilisateur et le mot de passe. D'autres fournisseurs utilisent un protocole d'authentification spécifique. Pour ceux-ci, l'authentification est réalisée directement par le démon `pppd`, une fois que la ligne a été établie. Les deux principaux protocoles d'authentification sont PAP (abréviation de l'anglais « Password Authentication Protocol ») et CHAP (abréviation de « Challenge Handshake Authentication Protocol »). PAP réalise l'authentification comme le mécanisme de login classique : le client envoie son nom et le mot de passe correspondant en clair au serveur. CHAP utilise en revanche la notion de défi. Le serveur envoie une requête contenant son nom, et le client doit s'identifier et authentifier son identité en lui renvoyant une réponse contenant son propre nom et une valeur calculée à partir du mot de passe correspondant. Les deux méthodes seront présentées ci-dessous. Si vous ne parvenez pas à vous connecter à Internet avec la première de ces méthodes, tentez votre chance avec PAP ou CHAP. Sachez que pour utiliser ces protocoles il est nécessaire de connaître, en plus de votre login et de votre mot de passe, le nom du serveur utilisé. Cette information doit vous être communiquée par votre fournisseur d'accès à Internet. Si vous avez le choix, utilisez de préférence le protocole CHAP, car il n'envoie jamais de mot de passe en clair sur la ligne téléphonique.

La configuration de l'accès à Internet pour un fournisseur d'accès requiert donc la configuration du réseau, la configuration de `ppp`, la configuration de `chat` et l'écriture des scripts de connexion. Ces étapes seront détaillées ci-dessous. Nous supposons dans la suite que vos paramètres de connexion sont les suivants :

Paramètre	Valeur
Fournisseur d'accès Internet	www.monfai.fr
Nom de domaine	monfai.fr
Adresse du DNS	192.205.43.1
Numéro de téléphone	08 36 76 30 18
Nom de login	jdupont
Mot de passe	gh25k;f
Nom de serveur (pour PAP et CHAP)	serveurfai

Note : Les informations données dans le tableau ci-dessus sont fictives et ne servent qu'à donner un exemple. Vous devez bien entendu les remplacer par les valeurs vous concernant en chaque endroit où elles apparaissent dans la suite de ce document. Si par hasard une de ces informations correspondait à un numéro de téléphone, un nom ou une adresse IP valide, ce serait une pure coïncidence.

Certains fournisseurs d'accès refuseront de vous donner toutes ces informations, sous prétexte qu'ils vous fournissent un kit de connexion pour Windows. Ce n'est pas trop grave, car il est possible de leur extorquer ces informations malgré eux. Les seules informations réellement indispensables sont le numéro de téléphone, le nom de login, le mot de passe et le nom de domaine. Les adresses de DNS peuvent être déterminées automatiquement dans le cadre du protocole PPP, et le nom de serveur est arbitraire et ne sert qu'à identifier la connexion.

IX-C-2 - Création d'une connexion à Internet

La première étape dans la création d'une connexion à Internet est avant tout l'ajout des serveurs DNS du fournisseur d'accès à votre configuration réseau. Cela n'est pas toujours nécessaire, car il est possible de configurer le démon pppd pour qu'il demande au fournisseur d'accès les adresses IP des DNS de celui-ci lors de l'établissement de la connexion. Cependant, il est plus facile d'utiliser le mécanisme de connexion à la demande si l'on indique les adresses des DNS du fournisseur d'accès dans la configuration réseau.

Pour l'exemple de connexion utilisé ici, vous devez simplement ajouter les lignes suivantes dans le fichier `/etc/resolv.conf` :

```
search monfai.fr
nameserver 192.205.43.1
```

Si vous ne connaissez pas les adresses de DNS de votre fournisseur d'accès, vous pourrez l'obtenir en regardant dans les traces du noyau lors de l'établissement d'une connexion. Nous verrons cela en détail plus loin.

La deuxième étape est d'écrire un script de numérotation pour le programme chat. Ce script pourra être placé dans le répertoire `/etc/ppp/peers/`, et pourra être nommé `monfai.chat` par exemple. Son contenu sera le suivant :

```
REPORT CONNECT
ABORT ERROR
ABORT BUSY
ABORT VOICE
ABORT "NO CARRIER"
ABORT "NO DIALTONE"
"" ATZ
OK AT&F1
OK ATMOL0DT0836763018
CONNECT ""
```

Ce script contient le programme que chat doit exécuter pour appeler le fournisseur d'accès à Internet. Il indique toutes les erreurs possibles susceptibles de faire échouer l'opération, puis il envoie les commandes d'initialisation et de numérotation au modem. Vous devrez remplacer le numéro de téléphone utilisé dans la commande `DT<numéro>` par le numéro de téléphone de votre fournisseur d'accès à Internet.

Note : Dans les scripts pour chat, il est possible d'utiliser indifféremment l'apostrophe simple, l'apostrophe double ou aucune apostrophe si la chaîne attendue de la part du modem ou à lui envoyer ne contient aucun espace.

Si d'aventure ce script ne fonctionne pas, vous serez sans doute obligé de demander au programme chat d'afficher tous les messages envoyés et reçus par le modem. Cela se fait normalement avec les options -v et -s. Vous verrez alors les messages en provenance du modem, ce qui vous permettra de déterminer comment modifier ce script.

Si votre fournisseur d'accès à Internet utilise un mécanisme de login classique, vous devez faire l'identification directement dans le script chat. En général, le serveur du fournisseur d'accès envoie la demande de login dès que la connexion a été établie. Pour cela, il utilise une chaîne de caractères telle que « login: », à laquelle le script chat doit répondre par votre nom d'utilisateur. Le serveur demande alors le mot de passe avec une chaîne telle que « password: », demande à la suite de laquelle le script chat doit envoyer votre mot de passe. Ce n'est que lorsque ces deux informations auront été fournies que la connexion sera réellement établie. Vous pouvez compléter le script chat pour répondre à ces deux questions avec les deux lignes suivantes :

```
login: jdupont
password: gh25k;f
```

Vous devrez bien entendu remplacer le nom du login et le mot de passe par vos propres données dans ce script. Notez qu'il n'est pas nécessaire (ni recommandé) de demander la réception de la chaîne complète « login: », car une erreur de transmission sur la ligne peut encore arriver à ce stade et provoquer la perte des premiers caractères envoyés par l'ordinateur distant. De plus, il se peut que la première lettre soit en majuscule ou en minuscule, selon le fournisseur d'accès à Internet que vous utilisez. Il en va de même pour la demande de mot de passe (« password: »).

Si, en revanche, votre fournisseur d'accès utilise le mécanisme d'authentification PAP ou CHAP, vous ne devez pas ajouter ces lignes, car le serveur n'enverra pas la chaîne de caractères « login: ». Il tentera de communiquer directement avec votre démon pppd pour réaliser l'authentification. Bien entendu, les défis lancés par le serveur sont simplement constitués de la demande du login et de la demande du mot de passe correspondant à ce login. Le démon pppd utilisera alors les « secrets » stockés dans le fichier `/etc/ppp/pap-secrets` ou le fichier `/etc/ppp/chap-secrets` pour répondre à ces défis, selon que le protocole utilisé par le serveur du fournisseur d'accès est PAP ou CHAP. C'est donc dans ces fichiers que vous devrez enregistrer votre login et votre mot de passe. Le format de ces fichiers est très simple. Les deux fichiers utilisent la même syntaxe pour les secrets. Vous devez simplement y ajouter une ligne telle que celle-ci :

```
# Secrets for authentication using PAP/CHAP
# client      server  secret  IP addresses
jdupont serveurfai  gh25k;f
```

pour que l'identification et l'authentification se fasse correctement. Comme on le voit, le nom du serveur est indiqué dans ce fichier : il permet de déterminer quel login et quel mot de passe doivent être utilisés pour les protocoles PAP et CHAP.

Note : En fait, le protocole PPP ne permet pas d'identifier des utilisateurs, mais des machines. Cependant, pour votre fournisseur, votre machine est identifiée par votre login, et il faut donc indiquer ce nom comme nom de machine cliente dans les fichiers pap-secrets et chap-secrets.

Le nom de serveur n'est pas utilisé pour la connexion. Il ne sert qu'à déterminer quel secret doit être utilisé pour une connexion donnée. Comme la plupart des gens n'ont qu'un seul fournisseur d'accès à Internet, ce nom est purement et simplement facultatif. Dans ce cas, on peut parfaitement remplacer le nom du serveur par une étoile. Ce caractère générique indique simplement que le même secret doit être utilisé pour toutes les connexions.

Pour les connexions à Internet, il est souvent impossible de connaître a priori l'adresse IP que le serveur donnera au client. On peut donc laisser vide la colonne contenant les adresses IP utilisables par les clients.

Lorsque vous aurez écrit le script chat et éventuellement complété les fichiers de secrets de PAP ou de CHAP, vous pourrez écrire le script de connexion à Internet. Ce script est très simple :

```
#!/bin/sh
# Script de connexion à Internet

# Effectue le ménage :
rm -f /var/spool/uucp/LCK* /var/lock/LCK* /var/run/ppp*.pid

# Établit la connexion :
/usr/sbin/pppd /dev/modem 115200 connect "/usr/sbin/chat -v \
-f /etc/ppp/peers/monfai.chat" defaultroute usepeerdns \
ipcp-accept-remote ipcp-accept-local
```

Prenez garde à écrire la ligne exécutant pppd et la ligne des options en une seule ligne, sans saut de ligne. Notez que le caractère '\ ' placé en fin de ligne permet d'indiquer que la commande se poursuit sur la ligne suivante. Ce script commence par faire le ménage sur les fichiers éventuellement laissés par les anciennes connexions, et appelle pppd en lui demandant d'utiliser la connexion établie par le programme chat avec la vitesse maximale de connexion indiquée. Lorsque la connexion est établie, le serveur du fournisseur d'accès est enregistré comme passerelle par défaut dans la table de routage du noyau. L'envoi et la réception des paquets IP en direction de l'Internet se feront donc en passant par cette passerelle. Les adresses de DNS du fournisseur d'accès sont également récupérées et ajoutées dans le fichier /etc/ppp/resolv.conf. Vous pourrez les déterminer simplement en consultant les fichiers de log de votre distribution dans le répertoire /var/log/. Les deux dernières options permettent d'autoriser le serveur du fournisseur d'accès à fixer les adresses IP locales et distantes.

Dans ce script de connexion, le programme chat est appelé avec le script de numérotation pour chat que l'on a déjà écrit. Si l'on désire voir exactement ce qui se passe, on peut ajouter l'option -s à la commande d'exécution de chat :

```
/usr/sbin/chat -v -s -f /etc/ppp/peers/monfai.chat
```

dans le script de connexion vu ci-dessus. Cela vous permettra de déboguer votre script de numérotation.

Vous remarquerez que le programme pppd utilise le fichier spécial de périphérique /dev/modem pour la communication. Il va de soi que si ce fichier spécial n'existe pas, vous devrez le créer. La solution la plus simple est de faire un lien symbolique vers /dev/ttS0 ou /dev/ttS1, selon que votre modem est connecté sur le premier ou le deuxième port série de votre ordinateur.

La commande de lancement donnée ci-dessus suppose que le script de connexion /etc/ppp/peers/monfai.chat réalise l'identification et l'authentification. Si vous utilisez l'un des protocoles PAP ou CHAP, il faut demander à pppd d'effectuer ces deux opérations lui-même. Pour cela, il faut ajouter des options dans la ligne de commande utilisée pour le lancer dans le script de connexion, afin de préciser le nom du serveur et le nom d'utilisateur pour l'identification. La ligne complète à utiliser pour PAP ou CHAP doit donc être celle-ci :

```
# Établit la connexion :
/usr/sbin/pppd /dev/modem 115200 connect "/usr/sbin/chat -v \
-f /etc/ppp/peers/monfai.chat" defaultroute usepeerdns \
ipcp-accept-remote ipcp-accept-local noauth \
remotename serveurfai user jdupont
```

Encore une fois, cette commande doit être écrite sur une seule ligne, ou sur plusieurs lignes séparées par le caractère '\ '. L'option noauth signale qu'il n'est pas nécessaire que le serveur du fournisseur d'accès s'authentifie en tant que tel (ce n'est pas nécessaire, car les fournisseurs d'accès ne jouent pas vraiment aux pirates du web), et les deux options suivantes permettent respectivement d'indiquer le nom de ce serveur ainsi que le nom d'utilisateur à utiliser pour le login. Lors de l'authentification, le démon pppd lira le fichier de secret du protocole choisi par le fournisseur d'accès pour trouver le mot de passe à utiliser. Notez encore une fois qu'il est facultatif de préciser le nom du serveur du fournisseur d'accès si vous avez utilisé le caractère générique '*' dans les fichiers de secrets pour PAP et CHAP.

Il est possible de faire en sorte que la connexion à Internet s'établisse automatiquement dès qu'un programme cherche à utiliser une adresse devant passer par la route par défaut. Toute demande à destination d'une machine dont l'adresse est inconnue localement provoque dans ce cas l'établissement de la connexion à Internet. Pour cela, il suffit simplement :

- d'ajouter l'option `demand` à la ligne de commande de `pppd` dans le script de connexion ;
- de lancer ce script en arrière-plan dans un des scripts de démarrage du système.

Si vous désirez utiliser cette fonctionnalité, il est recommandé d'activer également la déconnexion automatique, faute de quoi vous risqueriez de rester connecté involontairement. Cela se fait simplement en ajoutant l'option `idle n` dans la ligne de commande de `pppd`, où `n` est le nombre de secondes pendant lequel le lien PPP doit rester inactif avant que la connexion ne se coupe automatiquement. Un choix judicieux est par exemple 600 (ce qui correspond à dix minutes).

Note : L'interface réseau est automatiquement créée lorsqu'on lance le démon `pppd` avec l'option `demand`. Par conséquent, le démon `pppd` définira automatiquement une adresse IP locale pour cette interface et une adresse IP distante pour la passerelle par défaut. Malheureusement, ces adresses peuvent changer d'une connexion à une autre. En effet, la plupart des fournisseurs attribuent les adresses IP en fonction du modem qui répond au client qui appelle et, en général, la connexion ne se fait pas toujours sur le même modem (il peut y avoir plusieurs modems pour un même numéro de téléphone). C'est pour cela qu'il faut ajouter les options `ipcp-accept-remote` et `ipcp-accept-local` pour indiquer à `pppd` de modifier ces adresses lorsqu'il établira la connexion. Le problème, c'est que les connexions TCP/IP utilisant ces adresses seront systématiquement cassées à la suite de ce changement. C'est en particulier le cas pour le programme qui a initié l'établissement de la liaison PPP. C'est pour cette raison qu'il est recommandé de demander une adresse IP fixe à son fournisseur d'accès lorsqu'on veut utiliser la connexion à la demande. Hélas, ce service peut faire l'office d'une facturation supplémentaire.

Même si l'on utilise l'option `usepeerdns` dans la ligne de commande de PPP, il est recommandé de rajouter les adresses des DNS dans le fichier `/etc/resolv.conf`. En effet, en l'absence de DNS, les noms de domaine ne seront pas résolus et aucune requête vers l'extérieur ne se fera. Par conséquent, la route par défaut ne sera pas utilisée, et `pppd` n'établira donc pas la connexion automatiquement. Notez que si vous définissez les DNS dans le fichier `/etc/resolv.conf`, vous pouvez vous passer de l'option `usepeerdns` dans la ligne de commande de `pppd`. Si vous désirez procéder de cette manière, vous devrez vous assurer que l'ordre spécifié pour la résolution des noms dans le fichier `/etc/host.conf` est bien le fichier `/etc/hosts` (option `hosts`) avant le DNS (option `bind`), faute de quoi votre ordinateur se connectera systématiquement à Internet dès que vous utiliserez un nom de machine local.

La plupart des options passées en ligne de commande peuvent être spécifiées dans le fichier d'options `/etc/ppp/options` du démon `pppd`. Cela peut permettre de simplifier les scripts de connexion.

Si l'on n'utilise pas les mécanismes de connexion / déconnexion automatiques, on devra se déconnecter manuellement. Pour cela, rien de plus simple : il suffit de détruire le démon `pppd`. Cela peut être réalisé automatiquement avec le script suivant :

```
#!/bin/sh
# Script de terminaison de la connexion PPP
#

# Détermine la connexion à fermer (fournie sur la ligne de commande,
# ppp0 par défaut) :
if [ "$1" = "" ]; then
    DEVICE=ppp0
else
    DEVICE=$1
fi

# Teste si la connexion indiquée est active :
if [ -r /var/run/$DEVICE.pid ]; then

    # Détruit le processus ppp correspondant (son PID est stocké
    # dans /var/run/) :
    kill -INT `cat /var/run/$DEVICE.pid`

    # Effectue le ménage :
    if [ ! "$?" = "0" ]; then
        rm -f /var/run/$DEVICE.pid
        echo "ERREUR: Un fichier de PID résiduel \
a dû être effacé manuellement"
        exit 1
    fi
fi
```

```
rm -f /var/spool/uucp/LCK* /var/lock/LCK*
echo "La connexion PPP sur $DEVICE a été fermée correctement&#8230;"
echo

# Termine le script :
exit 0

fi

# La connexion indiquée n'est pas active :
echo "ERREUR: Il n'y a pas de connexion PPP sur $DEVICE"
exit 1
```

IX-C-3 - Connexion à l'ADSL

Les connexions ADSL ne requièrent pas d'appel téléphonique vers un modem du fournisseur d'accès. Elles s'établissent donc d'une manière différente, généralement bien plus simple. Toutefois, la manière de procéder dépend des fournisseurs d'accès et de la nature de votre modem ADSL. Il est donc difficile de décrire de manière générique la manière dont une connexion ADSL se met en place sous Linux. Seules les grandes lignes seront donc indiquées dans ce paragraphe.

Sachez pour commencer qu'il existe trois types de modems ADSL : les modems USB, les modems Ethernet simples, et les modems routeurs. Les premiers sont connectés à votre ordinateur via un câble USB. Les deuxièmes utilisent plutôt un câble réseau Ethernet classique. Les modems routeurs quant à eux sont des modems ADSL couplés d'un switch et, de plus en plus souvent, d'un point d'accès sans fil Wi-Fi. Ces modems sont intéressants, car ils permettent de connecter plusieurs ordinateurs à Internet via la même ligne, et ces ordinateurs peuvent être mis en réseau directement en les câblant au modem routeur à l'aide de câbles Ethernet, et via un réseau sans fil Wi-Fi pour les modems avec point d'accès.

En général, les modems USB requièrent des pilotes de périphériques spécifiques à chaque type de modem. Ce n'est pas le cas des modems Ethernet et des modems routeurs, puisqu'ils se connectent directement à votre carte Ethernet par un câble Ethernet (droit ou croisé selon votre modem). Par conséquent, si votre carte réseau est prise en charge par Linux (ce qui est fort probable), vous n'aurez absolument aucun problème pour vous connecter à Internet avec un modem Ethernet, alors que les pilotes USB pour Linux sont rarement développés avec sérieux par les fabricants de modems (ayant essuyé les plâtres avec le modem Sagem Fast800 fourni par Free à ses clients et participé au débogage de celui-ci, je suis bien placé pour le savoir). De plus, les modems USB ne permettent pas de partager la connexion Internet entre plusieurs ordinateurs, fonctionnalité très intéressante dans le cadre d'un petit réseau familial ou si vous disposez d'un ordinateur fixe et d'un portable. Enfin, les modems Ethernet sont alimentés, alors que les modems USB utilisent l'alimentation du port USB, ce qui dans 50% des cas provoque des instabilités de la connexion Internet, si ce n'est de l'ordinateur lui-même.

Il est donc vivement conseillé de choisir un modem routeur ou, à défaut, un modem Ethernet si vous en achetez un ou si votre fournisseur d'accès à Internet vous en propose. La plupart des modems évolués des offres multiples (téléphone + Internet et éventuellement TV) distribués de nos jours sont généralement des modems routeurs, et disposent souvent d'une possibilité de point d'accès Wi-Fi.

Note : Même si, comme nous le verrons plus loin, Linux est parfaitement capable de réaliser un partage de connexion à Internet, l'achat d'un routeur domestique ou, pour simplifier l'installation, d'un modem routeur prenant tout en charge, reste intéressant. En effet, ces appareils, prenant totalement en charge la connexion à Internet, rendent la configuration de l'accès à Internet triviale au niveau de l'ordinateur. De plus, ils peuvent fournir des services complémentaires, tels que la configuration dynamique des ordinateurs clients par le protocole DHCP, ou encore le relai de serveur de nom avec un mini serveur DNS, accélérant ainsi de manière non négligeable le temps de réponse lors de la navigation.

De plus, et c'est sans doute l'un des plus gros avantages, les modems routeurs peuvent jouer le rôle d'un pare-feu « naturel » dès lors que la fonction de partage de connexion à Internet est activée. En effet, même si un seul ordinateur est connecté au modem routeur, celui-ci doit utiliser un jeu d'adresses IP privées pour votre réseau local, car votre fournisseur ne vous attribuera qu'une seule adresse IP lors de la connexion. Or, les adresses privées utilisées de ce type ne sont pas, par définition, routables sur Internet. Par conséquent, du point de vue d'Internet, seul le modem

routeur est visible : aucun de vos ordinateurs n'est accessible ! Il est donc quasiment impossible à un pirate et encore moins à un virus de passer au travers de cet appareil, sauf à le pirater en premier lieu. Dans tous les cas, cela complique sérieusement les choses, et le type de protection fourni par ces appareils est donc extrêmement fiable. Ne serait-ce que pour cet aspect des choses, je vous recommande vivement l'utilisation de ce type d'appareil (surtout si vous avez un ordinateur qui fonctionne sous Windows).

Même si vous utilisez un modem USB, il apparaîtra généralement comme une interface Ethernet dans le système. L'installation des pilotes pour les modems USB étant extrêmement spécifique, cette étape ne sera pas décrite plus en avant ici. La suite du document suppose donc que vous disposiez d'un modem Ethernet, ou que votre modem soit accessible par l'intermédiaire d'une interface Ethernet simulée par le pilote du modem.

Lorsque vous aurez branché votre modem et que vous pourrez y accéder via son interface Ethernet, vous devrez établir la connexion avec votre fournisseur d'accès. Cela dépend encore une fois du fournisseur d'accès, et parfois même, pour un même fournisseur, de la région dans laquelle vous vous trouvez. Généralement, deux méthodes sont utilisées pour établir la connexion :

- soit les communications se font de manière totalement transparente via l'interface Ethernet du modem, auquel cas la connexion s'établit aussi facilement que si vous vous connectiez à un réseau Ethernet classique ;
- soit les communications sont encapsulées dans le protocole de communication PPP, via un protocole du type PPPoE (« PPP over Ethernet ») ou PPPoA (« PPP over ATM »), auquel cas vous devrez utiliser le démon `pppd` pour établir la connexion.

Dans le premier cas, la configuration est simpliste. Il suffit d'utiliser l'interface Ethernet du modem comme interface de sortie dans la route par défaut, et éventuellement de mettre un client DHCP à l'écoute sur cette interface si le fournisseur d'accès utilise ce protocole pour attribuer les adresses IP de ses clients. La manière de procéder est décrite dans la section traitant de [DHCP](#). Cette technique est celle utilisée par le fournisseur d'accès Free en France en zone dégroupée. Pour ce fournisseur, les adresses IP fournies étant fixes en zone dégroupée, la connexion à Internet se résume donc aux deux commandes suivantes :

```
ifconfig eth0 adresse up
route add default eth0
```

où `adresse` est l'adresse IP que votre fournisseur d'accès vous a attribuée. La connexion à Internet se fait donc dans ce cas comme la configuration d'une simple carte réseau.

Dans le deuxième cas, le plus simple est d'utiliser le protocole d'encapsulation PPPoE. Comme son nom l'indique, ce protocole permet d'encapsuler les paquets de la liaison PPP avec votre fournisseur Ethernet directement dans des trames Ethernet, qui peuvent ensuite être envoyées via l'interface Ethernet du modem. Cette solution peut être mise en place avec l'outil `rp-pppoe` de la société [Roaring Penguin](#), ou directement avec un gestionnaire PPPoE au niveau du noyau.

La mise en place d'une connexion avec `rp-pppoe` est réellement très simple. En effet, un fichier d'options pour PPP nommé `pppoe.conf` et des scripts de connexion et déconnexion `adsl-start` et `adsl-stop`, permettant de lancer le démon `pppd` avec les bonnes options et dans le bon contexte pour établir la connexion, sont fournis. Les seules opérations à faire sont dans ce cas de s'assurer que les identifiants de connexion (login et mot de passe) fournis par votre fournisseur d'accès sont bien enregistrés dans les fichiers `pap-secrets` et `chap-secrets`, et que le fichier `pppoe.conf` y fait bien référence (au niveau de l'option `USER`). La manière de procéder ayant déjà été décrite ci-dessus, elle ne sera pas détaillée outre mesure.

L'utilisation du gestionnaire PPPoE du noyau est encore plus directe, car il suffit dans ce cas de lancer le démon `pppd` en lui demandant de travailler avec l'interface réseau du modem. Pour utiliser ce gestionnaire, vous devez activer, outre le support de PPP, l'option « PPP over Ethernet (EXPERIMENTAL) » dans les options de configuration du noyau. Une fois les identifiants de connexion écrits dans les fichiers `pap-secrets` ou `chap-secrets`, la connexion à Internet se fera simplement avec la commande suivante :

```
pppd interface
```

où interface est le nom de l'interface réseau du modem.

Note : Il est recommandé de limiter la taille des paquets en réception et en émission à 1492 octets, afin de prendre en compte les frais dus à l'encapsulation PPPoE. Pour cela, il suffit de placer les options suivantes dans le fichier d'options de pppd :

```
mtu 1492  
mru 1492
```

Il est impératif de mettre en place un pare-feu si vous disposez d'une connexion ADSL. La manière de procéder est décrite dans la [Section 9.4](#).

IX-C-4 - Les autres outils de connexion

Vous en savez suffisamment à présent pour vous connecter à Internet avec Linux. Les opérations qui ont été décrites ici sont légèrement compliquées, et vous les trouvez sans doute un peu lourdes. Cela est naturel, car elles le sont effectivement. En fait, les opérations décrites ici vous montrent comment vous connecter manuellement, mais elles ne constituent pas la manière de faire la plus facile. En effet, si vous installez XWindow, vous pourrez utiliser des programmes graphiques permettant de vous connecter à Internet, d'envoyer et recevoir des courriers électroniques (« mails »), de naviguer (« surfer ») et de lire les groupes de discussions (« news »). Cependant, l'utilisation de ces outils ne sera pas décrite ici, car il en existe trop pour que l'on puisse tous les présenter. Heureusement, ces outils sont prévus pour être très simples d'emploi et leur configuration ne pose réellement aucun problème.

IX-C-5 - Configuration d'un cache de DNS

La résolution des noms de domaines est une opération dont le coût est loin d'être négligeable au niveau du temps de connexion aux serveurs, si ce n'est au niveau de la bande passante utilisée. En effet, il est relativement courant qu'une page Web fasse de multiples références à des machines inconnues et dont les adresses doivent être résolues. De ce fait, la résolution des noms de domaines peut parfois se mesurer en secondes, surtout si les serveurs de noms des fournisseurs d'accès sont surchargés, ce qui arrive très couramment, même pour les clients des offres ADSL à 20 Mbits/s.

De plus, il est courant qu'un internaute reste toujours sur le même site pendant qu'il navigue sur Internet. Il peut donc être très intéressant de disposer d'un cache local pour les requêtes DNS. Un tel cache permet de mémoriser les interrogations faites par les programmes clients (essentiellement les navigateurs Web) et surtout les réponses renvoyées par le serveur de nom du fournisseur d'accès. Ainsi, lorsqu'un client effectue une requête sur un nom de domaine pour lequel il a déjà fait une requête auparavant (par exemple lors de l'accès à la page précédente), la réponse de cette requête est accessible immédiatement, puisqu'elle se trouve dans le cache local. Ce mécanisme permet donc de réduire sensiblement les temps de connexions aux machines situées sur Internet : seule la première connexion prend le temps habituel, et toutes les autres sont accélérées.

Note : Les mécanismes de « caches » sont très souvent utilisés en informatique pour permettre d'obtenir rapidement des informations habituellement difficiles à obtenir, mais qui varient très peu. Du fait de leur quasi-constance, il est en effet possible de les copier dans une mémoire locale. Par exemple, les caches sont utilisés au niveau des disques durs, dans les processeurs pour accéder aux données en mémoire, et au niveau du réseau. Ils permettent d'augmenter considérablement les performances, au détriment d'une plus grande complexité des opérations d'écriture des données, puisque celles-ci invalident les données des caches et doivent forcer leur mise à jour.

En pratique, la mise en place d'un cache de DNS consiste à installer un petit serveur de nom local sur la machine, qui répond à toutes les requêtes des applications clientes. Ce petit serveur est lui-même client des serveurs de noms effectifs qui se trouvent sur Internet. L'idéal est dans ce cas simplement de s'adresser aux serveurs de noms du fournisseur d'accès à Internet, afin de bénéficier de son propre cache et de sa bande passante pour les requêtes intermédiaires. C'est exactement ce que se propose de faire le programme DNSmasq.

La configuration de DNSmasq se fait essentiellement au niveau du fichier de configuration `/etc/dnsmasq.conf`. Ce fichier est constitué d'une série d'options auxquelles une valeur est affectée, les lignes commençant par un signe dièse étant des commentaires. Le fichier suivant peut vous servir d'exemple, sachant que les options sont toutes commentées dans le fichier fourni par défaut :

```
# Exemple de fichier de cache de DNS :

# Fichier de définition des serveurs de noms en amont :
# Si on utilise ppp pour se connecter :
# resolv-file=/etc/ppp/resolv.conf

# Si on utilise DHCP directement :
# resolv-file=/var/dhcp/resolv.conf

# Adresses des serveurs utilisés par les sociétés et les fournisseurs d'accès
# tordus qui ont mis en place une entrée par défaut pour capter toutes les requêtes
# sur les noms de domaines incorrects et qui redirigent leurs clients vers des sites
# publicitaire au lieu de renvoyer une erreur comme la nétiquette l'impose :

# Verisign (ils se sont calmés récemment) :
# bogus-nxdomain=64.94.110.11

# Mon fournisseur débile :
# bogus-nxdomain=adresse.du.site.publicitaire
```

Ce fichier d'exemple ne montre que les options réellement nécessaires. L'option `resolv-file` permet d'indiquer au démon `dnsmasq` où trouver les adresses des serveurs DNS amont (c'est-à-dire ceux du fournisseur d'accès en pratique). Il faut indiquer ici un fichier qui contient les lignes `nameserver` donnant les adresses de ces serveurs. Comme nous allons le voir ci-dessus, il ne faut pas spécifier ici le fichier `/etc/resolv.conf` du système, car nous allons le modifier. Au contraire, il faut indiquer un fichier dans lequel les outils de connexion réseau (client DHCP ou démon `pppd` par exemple) enregistrent les adresses des serveurs de noms du fournisseur d'accès.

Par exemple, lorsqu'une connexion est établie avec le démon `ppp`, celui-ci enregistre les serveurs DNS primaires et secondaires du fournisseur d'accès dans le fichier `/etc/ppp/resolv.conf`. Il suffit donc de référencer ce fichier dans l'option `resolv-file`.

En revanche, si la connexion s'établit directement grâce au protocole DHCP, il faut configurer le client DHCP du système pour qu'il enregistre les adresses des serveurs DNS dans un fichier autre que le fichier `/etc/resolv.conf` du système. Ici, nous utilisons le fichier `/var/dhcp/resolv.conf`. Par exemple, si votre distribution utilise le client DHCP `dhcpcd`, il suffit de lui ajouter l'option `-e /var/dhcp` pour que ce fichier soit créé dans le répertoire `/var/dhcp/` au lieu du répertoire `/etc/`.

L'option `bogus-nxdomain` est un palliatif à une tentative de détournement des requêtes DNS que certaines sociétés effectuent pour obtenir un profit publicitaire direct ou des statistiques comportementales sur les internautes. Ces sociétés redirigent en effet toutes les requêtes mal formées ou mal orthographiées vers un de leur serveur Web, ce qui leur permet d'obtenir des informations sur les sites accédés, d'afficher une page de publicité, ou encore faire une redirection directe vers un site supposé intéresser l'internaute.

Bien entendu, ce comportement est pour le moins cavalier, car l'internaute ne comprend pas forcément pourquoi il tombe sur un site inconnu alors qu'il demandait l'adresse d'un autre site. D'autre part, ces redirections perturbent le bon fonctionnement de certains outils qui s'attendent explicitement à trouver une erreur en cas d'échec de résolution de nom.

Il n'y a malheureusement pas grand-chose à faire pour faire revenir à la raison ces sociétés (à part porter plainte, passer à la concurrence, ou plus méchamment bombarder leurs serveurs DNS de requêtes invalides avec un programme, ce que je vous déconseille de faire vivement...). En revanche, `dnsmasq` peut détecter ces redirections et générer une erreur de résolution de nom classique. Pour cela, il suffit d'identifier l'adresse IP du serveur cible de la redirection, et de l'indiquer à `dnsmasq` à l'aide de l'option `bogus-nxdomain`. Ainsi, lorsque `dnsmasq` reçoit une réponse référençant ces serveurs, il renvoie directement une erreur aux programmes clients.

Une fois que vous aurez configuré et lancé le démon dnsmasq, vous devrez encore indiquer au système qu'il devra interroger ce démon pour effectuer les résolutions des noms de domaines. Comme il l'a déjà été indiqué dans les sections précédentes, cela se fait simplement en ajoutant une ligne `nameserver` référençant la machine locale dans le fichier de configuration `/etc/resolv.conf` :

```
nameserver 127.0.0.1
```

Note : Le démon dnsmasq ne permet de cacher que les résultats des requêtes DNS. Or, de nombreux fichiers qui ne changent quasiment jamais, comme les images de fond, les logos et les pages statiques, sont récupérés de multiples fois lorsqu'un internaute navigue sur des pages Web. Il est donc très intéressant de mettre également ces fichiers en cache. Nous verrons comment réaliser cela dans la section suivante.

IX-C-6 - Installation d'un proxy HTTP

Lors d'une connexion à Internet, bon nombre d'informations sont échangées entre le serveur et le client. En particulier, lorsqu'on navigue, un certain nombre d'images et d'informations plus ou moins lentes à télécharger transitent systématiquement. Or la plupart des gens visitent régulièrement les mêmes sites, dont seulement quelques pages sont modifiées entre deux visites successives. Par conséquent, on peut se demander pourquoi les pages complètes devraient être rechargées à chaque visite, si seulement quelques informations ont été modifiées.

Cette question prend encore plus de sens lorsqu'on réalise un partage de connexion à Internet. Il est tout à fait concevable que plusieurs clients du même réseau demandent plusieurs fois les mêmes informations, ce qui provoque l'engorgement du lien partagé inutilement. C'est pour résoudre ce genre de problème que les programmes que l'on nomme « proxys » ont été développés.

Un proxy n'est rien d'autre qu'un programme qui s'intercale entre des clients et un serveur. Il a pour principale fonction de mémoriser les réponses les plus courantes renvoyées par le serveur, afin de répondre à la fois plus rapidement aux clients et d'éviter de surcharger le serveur. Il existe différents types de proxy, mais nous allons nous intéresser uniquement aux proxys Web, qui permettent donc de stocker des pages Web en local afin de soulager une liaison trop faible.

Le principe de fonctionnement d'un proxy est le suivant. Les navigateurs utilisés par les clients se connectent, en temps normal, directement sur le port 80 des serveurs Web. Leur configuration doit être modifiée pour utiliser le proxy à la place. De leur point de vue, le proxy se comporte comme un serveur Web capable de répondre aux requêtes des clients. Lorsqu'il peut le faire, il renvoie les données qu'il a stockées dans son cache. Sinon, il va les chercher directement sur Internet à la place du client. Le cache est maintenu de telle manière que les données obsolètes ou les données les moins utilisées sont supprimées, afin de laisser la place aux données qui sont le plus souvent demandées et qui changent le moins. Pour utiliser un proxy, il faut donc modifier la configuration des navigateurs des clients, afin de donner l'adresse de la machine sur laquelle le proxy fonctionne, ainsi que le port sur lequel il peut être contacté (généralement, c'est le port 8080).

Note : Il est également possible d'utiliser les mécanismes de translation d'adresse du noyau pour rendre l'utilisation du proxy transparente aux clients. La manière de procéder sera décrite dans la section traitant des mécanismes de [filtrage](#) du noyau.

Il existe plusieurs proxys sur le marché, mais le plus utilisé sous Linux est sans doute squid. C'est un logiciel performant, puissant et libre, fourni avec la plupart des distributions modernes. Sa configuration peut être relativement technique, mais nous n'allons pas entrer dans tous les détails. Il s'agit simplement ici de donner un aperçu des fonctionnalités disponibles.

squid utilise le fichier de configuration `/etc/squid.conf` pour lire ses options de fonctionnement. Ce fichier contient en particulier les informations concernant le réseau, une liste de règles indiquant qui a le droit de lui demander des pages Web, et l'emplacement du cache dans lequel les données du cache seront enregistrées.

L'option la plus importante dans ce fichier de configuration est sans doute `cache_dir`, qui indique l'emplacement du cache ainsi que sa taille et sa structure. La syntaxe de cette option est la suivante :

```
cache_dir répertoire taille n m
```

où `répertoire` est le répertoire dans lequel le cache sera stocké (usuellement `/var/squid/cache/`), `taille` est la taille de ce cache en mégaoctets, et `n` et `m` deux nombres donnant la structure de l'arborescence du cache. La signification de ces deux nombres doit être expliquée un peu plus en détail.

Par défaut, squid essaie de répartir tous les objets qu'il place dans le cache de manière uniforme dans plusieurs répertoires, afin d'accélérer les temps de recherche sur ces objets lorsqu'un client les demande. Il utilise à cette fin une fonction de répartition (les programmeurs disent souvent une fonction de hash) qui indique dans quel répertoire se trouve un objet. La recherche dans ce répertoire se fait donc plus rapidement, car, si la fonction de répartition est bien faite (et c'est le cas), les objets sont répartis de manière équilibrée dans tous les répertoires du cache, qui sont donc tous relativement peu peuplés. En fait, squid peut stocker tellement d'objets que le nombre de répertoires peut lui-même devenir très grand. Il utilise donc un découpage à deux niveaux, les répertoires du deuxième niveau étant répartis dans les répertoires du premier niveau. Le premier nombre spécifié dans l'option `cache_dir` indique le nombre de répertoires du premier niveau, et le deuxième nombre indique le nombre de sous-répertoires dans chacun de ces répertoires. En général, on utilise respectivement les valeurs 16 et 256 :

```
cache_dir /var/squid/cache 100 16 256
```

Cette ligne permet de cacher 100 Mo, en répartissant les objets dans 4096 répertoires répartis en 16 groupes de 256 répertoires.

Les options suivantes spécifient les paramètres réseau du cache :

- l'option `http_port` indique le port TCP que les clients doivent utiliser pour accéder au proxy ;
- l'option `ftp_user` permet de donner l'adresse email à utiliser comme mot de passe dans les connexions FTP anonymes ;
- l'option `cache_peer` permet d'intégrer le proxy local dans une hiérarchie de proxies.
Les proxies peuvent en effet être organisés dans une structure arborescente. Lorsqu'un proxy d'un niveau ne dispose pas d'une donnée demandée par un client, il peut contacter ses frères pour le cas où ceux-ci auraient cette donnée dans leurs caches, ou demander à son proxy père de récupérer cette donnée pour lui. Les proxies communiquent entre eux en utilisant un protocole dédié, le protocole ICP (abréviation de l'anglais « Inter Cache Protocol »).
En général, les petits réseaux ne disposent que d'un seul proxy, l'option `cache_peer` est donc souvent utilisée de la manière suivante :

```
cache_peer père parent port port_icp options
```

- L'option `prefer_direct` permet d'indiquer au proxy s'il doit chercher avant tout à récupérer lui-même les données qui lui manquent (valeur `on`) ou s'il doit contacter d'abord son proxy père (valeur `off`). Si l'on a spécifié un proxy père à l'aide de l'option `cache_peer`, on aura intérêt à utiliser la valeur `off` ;
- L'option `dns_children` permet de donner le nombre de processus en charge de cacher les requêtes sur les DNS. squid peut en effet lancer des serveurs de cache de DNS permettant d'optimiser les temps d'accès aux pages Web. Plus le réseau local est important, plus il faudra de tels processus, afin que tous les clients puissent obtenir les adresses IP des machines à partir de leur nom. La valeur par défaut est de 5 processus, ce qui devrait convenir dans la plupart des cas.

Les valeurs données par défaut à toutes ces options conviennent dans la plupart des cas. Elles sont parfaitement documentées dans le fichier de configuration `squid.conf`.

Par défaut, squid n'autorise personne à accéder au cache. Il est donc nécessaire de donner les droits nécessaires pour permettre aux machines de votre réseau d'accéder au cache. La gestion de la sécurité se fait par l'intermédiaire de ce que l'on appelle des ACL (abréviation de l'anglais « Access Control List »).

Une ACL est simplement un critère permettant de classifier les requêtes que le proxy reçoit. Ces critères sont définis à l'aide du mot clé `acl`, suivi du nom de la liste, suivi lui-même d'un critère que toutes les requêtes concernées par cette ACL devront vérifier. Les critères sont eux-mêmes exprimés à l'aide d'un mot clé qui indique sur quoi porte le critère et des options de ce critère. Les principaux mots clés que vous pourrez utiliser sont les suivants :

- `src`, qui permet de filtrer les requêtes par les adresses des machines qui les ont émises, en indiquant l'adresse de leur réseau, donnée sous la forme `adresse/masque` ;
- `dst`, qui permet de filtrer les requêtes par leurs adresses destination ;
- `port`, qui permet de sélectionner les requêtes s'adressant sur les ports fournis en paramètres ;
- `proto`, qui permet de spécifier la liste des protocoles utilisés par les requêtes ;
- `method`, qui permet de sélectionner les requêtes HTTP par la méthode utilisée dans la requête.

Ce tableau n'est pas exhaustif, mais il permet d'avoir une idée des capacités de filtrage des requêtes dont dispose squid. Vous trouverez ci-dessous quelques exemples pratiques de définitions d'ACL :

```
# ACL caractérisant toutes les machines :
acl all src 0.0.0.0/0.0.0.0

# ACL définissant la machine locale :
acl localhost src 127.0.0.1/255.255.255.255

# ACL définissant les machines d'un réseau privé local :
acl localnet src 192.168.0.0/255.255.255.0

# ACL caractérisant le proxy lui-même :
acl manager proto cache_object

# ACL spécifiant les ports acceptés par le proxy :
acl safe_ports port 80 21 443 563 70 210 1025-65535

# ACL définissant les requêtes de connexion :
acl connect method CONNECT
```

La définition des règles de sécurité utilise les ACL de manière séquentielle. Les règles doivent être données les unes après les autres, suivant le format suivant :

```
ressource politique ACL
```

où `ressource` est un mot clé indiquant la ressource sur laquelle la règle de sécurité porte, `politique` est l'action prise pour les requêtes vérifiant cette règle, et `ACL` une liste d'ACL permettant de spécifier les requêtes concernées par cette règle. La ressource la plus utilisée est sans doute le protocole HTTP, que l'on peut spécifier à l'aide du mot clé `http_access`. Les actions peuvent être l'acceptation (mot clé `allow`) ou le refus (mot clé `deny`) de la requête. Enfin, les requêtes concernées par une règle sont celles qui appartiennent à toutes les ACL de la liste d'ACL indiquée.

Vous trouverez ci-dessous quelques exemples de règles de sécurité courantes :

```
# Autorise les accès au gestionnaire de cache local :
http_access allow manager localhost

# Interdit les accès aux gestionnaires de cache étrangers :
http_access deny manager

# Interdit les accès pour toutes les requêtes provenant de ports non autorisés :
http_access deny !safe_ports
http_access deny connect !safe_ports

# Autorise les accès aux clients de la machine locale :
http_access allow localhost

# Autorise les accès aux machines du réseau local :
http_access allow localnet

# Politique par défaut :
```

```
http_access deny all
```

Notez que si une requête ne correspond à aucune règle, la politique par défaut utilisée par squid est de prendre l'action opposée à la dernière règle spécifiée. En pratique, il est plus sage de toujours indiquer une politique par défaut pour toutes les requêtes restantes, par exemple en utilisant l'ACL all définie ci-dessus.

L'exemple de jeu de règles de sécurité donné ci-dessous convient pour un réseau local d'adresses 192.168.0.0. Vous pouvez bien entendu modifier les options de sécurité comme bon vous semble. Encore une fois, rappelons que le fichier de configuration de squid est très bien documenté et que sa lecture est relativement facile. Je ne saurais que trop vous recommander de le consulter si vous désirez en savoir plus.

IX-D - Pare-feu et partages de connexion à Internet

Supposons qu'une machine située sur un réseau local ait accès à Internet. Il peut être intéressant de faire en sorte que les autres machines du réseau puissent également y accéder, en utilisant cette machine comme passerelle. Cela est parfaitement réalisable et ne pose aucun autre problème que la définition des règles de routage si toutes les machines ont une adresse IP attribuée par l'IANA. Cependant, cela est rarement le cas, car les réseaux locaux utilisent normalement les adresses réservées à cet usage, qui ne sont pas routables sur Internet. Dans ce cas, il est évident que les machines du réseau local ne pourront pas envoyer de paquets sur Internet, et qu'a fortiori elles ne recevront jamais de paquets provenant d'Internet. Heureusement, il existe une technique nommée masquerading, basée sur un mécanisme de translation d'adresse (« NAT » en anglais, abréviation de « Network Address Translation »), qui permet de modifier les paquets émis à destination d'Internet à la volée afin de pouvoir partager une connexion Internet même avec des ordinateurs qui utilisent des adresses locales. Comme nous allons le voir, partager une connexion à Internet avec d'autres ordinateurs d'un réseau local est un jeu d'enfant sous Linux grâce à cette technique.

Il faut toutefois bien se rendre compte que le fait de fournir un accès à Internet à un réseau local pose des problèmes de sécurité majeurs. Pour des réseaux locaux familiaux, les risques de piratages sont bien entendu mineurs, mais lorsque la connexion à Internet est permanente ou lorsque les données circulant sur le réseau local sont sensibles, il faut tenir compte des risques d'intrusion. Lorsqu'on utilise des adresses IP dynamiques, il est relativement difficile d'accéder à des machines du réseau local, sauf si la passerelle expose des services internes au reste du réseau. En revanche, si les adresses utilisées sont fixes et valides sur Internet, le risque devient très important. Par ailleurs, une machine Linux connectée sur Internet ouvre toujours des ports indésirés, même si elle est bien configurée. Par exemple, il est impossible d'empêcher le serveur X d'écouter les demandes de connexion provenant d'Internet, même s'il est configuré pour toutes les refuser. La configuration d'un ordinateur connecté à Internet doit donc se faire avec soin dans tous les cas, et l'installation d'un pare-feu est plus que recommandée (un « pare-feu », ou « firewall » en anglais, est un dispositif qui consiste à protéger un réseau local du « feu de l'enfer » d'Internet, sur lequel tous les crackers sont supposés grouiller).

Les paragraphes suivants exposent les mécanismes de filtrage de paquets réseau de Linux, qui sont utilisés tant pour définir les règles de protection contre les intrusions indésirables sur un réseau par l'intermédiaire d'une passerelle que pour effectuer des traitements sur les paquets. Cependant, de tous ces traitements, nous ne verrons que la translation d'adresse, car c'est sans doute celui que la plupart des gens cherchent à utiliser.

IX-D-1 - Mécanismes de filtrage du noyau

Linux est capable de filtrer les paquets circulant sur le réseau et d'effectuer des translations d'adresses depuis la version 2.0. Cependant, les techniques utilisées ont été profondément remaniées dans la version 2.4.0 du noyau, et une architecture extensible a été mise en place et semble répondre à tous les besoins de manière simple : Netfilter.

Netfilter est simplement une série de points d'entrée dans les couches réseau du noyau au niveau desquels des modules peuvent s'enregistrer afin d'effectuer des contrôles ou des traitements particuliers sur les paquets. Il existe une entrée en chaque point clé du trajet suivi par les paquets dans les couches réseau du noyau, ce qui permet d'intervenir de manière souple et à n'importe quel niveau. Un certain nombre de modules permettant d'effectuer les traitements les plus courants sont fournis directement dans le noyau, mais l'architecture Netfilter est suffisamment

souple pour permettre le développement et l'ajout des modules complémentaires qui pourraient être développés par la suite.

Les fonctionnalités fournies par ces modules sont regroupées par domaines fonctionnels. Ainsi, les modules permettant de réaliser des pare-feu se chargent spécifiquement de donner les moyens de filtrer les paquets selon des critères bien définis, et les modules permettant d'effectuer des translations d'adresses ne prennent en charge que la modification des adresses source et destination des paquets à la volée. Afin de bien les identifier, ces fonctionnalités sont regroupées dans ce que l'on appelle des tables. Une table n'est en fait rien d'autre qu'un ensemble cohérent de règles permettant de manipuler les paquets circulant dans les couches réseau du noyau pour obtenir une fonctionnalité bien précise. Les tables les plus couramment utilisées sont les tables « filter », « mangle » et « nat », qui permettent respectivement de réaliser des pare-feu, de marquer les paquets pour leur appliquer un traitement ultérieur, et d'effectuer des translations d'adresses.

Les différentes tables de NetFilter s'abonnent chacune aux points d'entrée qui sont nécessaires pour implémenter leur fonctionnalité, et les paquets qui passent au travers de ces points d'entrée doivent traverser les règles définies dans ces tables. Afin de faciliter la définition des règles stockées dans les tables, les règles sont regroupées dans des listes de règles que l'on appelle communément des chaînes. Chaque table dispose donc toujours d'une chaîne de règles prédéfinie pour chacun des points d'entrée de Netfilter qu'elle utilise. L'utilisateur peut bien entendu définir de nouvelles chaînes et les utiliser dans les chaînes prédéfinies, ces chaînes apparaissent alors simplement comme des sous-programmes des chaînes prédéfinies.

Les règles des tables de Netfilter permettent de spécifier les paquets qui les vérifient, selon des critères précis, par exemple leur adresse source ou le type de protocole qu'ils transportent. Les règles indiquent également les traitements qui seront appliqués à ces paquets. Par exemple, il est possible de détruire purement et simplement tous les paquets provenant d'une machine considérée comme non sûre si l'on veut faire un pare-feu. On pourra aussi envoyer ces paquets dans une autre chaîne, qui peuvent indiquer d'autres règles et d'autres traitements.

Les critères de sélection de paquets des règles de filtrage se basent sur les informations de l'en-tête de ces paquets. Rappelez-vous que chaque paquet de données émis sur le réseau transporte avec lui des informations nécessaires au fonctionnement du dit réseau, entre autres les adresses source et destination du paquet. Ce sont sur ces informations que la sélection des paquets est effectuée dans Netfilter. Dans le cas d'un protocole encapsulé dans un autre protocole, par exemple un paquet TCP dans un paquet IP, les informations du protocole encapsulé peuvent également être utilisées, si l'on a chargé des modules complémentaires dans NetFilter. Par exemple, le numéro de port TCP peut faire partie des critères de sélection d'une règle et donc de différencier le trafic des différents services réseau.

Les traitements que l'on peut appliquer à un paquet sont également fournis par des modules de Netfilter. Les traitements les plus simples sont bien entendu fournis en standard avec le noyau. En général, on peut chercher à réaliser les opérations suivantes sur un paquet :

- l'enregistrer, pour analyse ultérieure ;
- le rediriger vers un port local, pour traitement par un programme dédié (par exemple, par un serveur proxy) ;
- l'accepter, l'abandonner ou le rejeter directement (le rejet se distingue de l'abandon par l'émission d'une notification « hôte inaccessible » à la machine ayant émis le paquet) ;
- le modifier et, en particulier, effectuer une translation d'adresse ou le marquer pour le reconnaître dans un traitement ultérieur ;
- l'envoyer dans une autre chaîne pour effectuer des vérifications complémentaires ;
- le faire sortir immédiatement de la chaîne courante.

De plus, les statistiques tenues par le noyau pour la règle ainsi vérifiée sont mises à jour. Ces statistiques comprennent le nombre de paquets qui ont vérifié cette règle ainsi que le nombre d'octets que ces paquets contenaient.

Lorsqu'un paquet arrive à la fin d'une chaîne (soit parce qu'il n'a pas été rejeté ou détruit par une règle de filtrage de pare-feu, soit parce qu'il ne correspond aux critères d'aucune règle, ou soit parce qu'il est arrivé à ce stade après avoir subi des modifications), le noyau revient dans la chaîne appelante et poursuit le traitement du paquet. Si la chaîne au bout de laquelle le paquet arrive est une chaîne prédéfinie, il n'y a plus de chaîne appelante, et le sort du

paquet est déterminé par ce qu'on appelle la politique de la chaîne prédéfinie. La politique (« policy » en anglais) des chaînes prédéfinies indique donc ce que l'on doit faire avec les paquets qui arrivent en fin de chaîne. En général, on utilise une politique relativement stricte lorsqu'on réalise un pare-feu, qui rejette par défaut tous les paquets qui n'ont pas été acceptés explicitement par une règle de la chaîne.

Note : Certains paquets peuvent être découpés en plusieurs paquets plus petits lors de la traversée d'un réseau. Par exemple, un paquet TCP un peu trop gros et initialement encapsulé dans un seul paquet IP peut se voir éclater en plusieurs paquets IP. Cela peut poser quelques problèmes aux règles de filtrage, puisque dans ce cas les données spécifiques à TCP (par exemple les numéros de ports) ne sont disponibles que sur le premier paquet IP reçu, pas sur les suivants. C'est pour cette raison que le noyau peut effectuer une « défragmentation » des paquets lorsque cela est nécessaire, et que les paquets transmis par la passerelle ne sont pas toujours strictement identiques aux paquets qu'elle reçoit.

IX-D-2 - Translations d'adresses et masquerading

Nous avons vu que grâce aux mécanismes de filtrage du noyau, il est possible de modifier les paquets à la volée. Il peut être intéressant de modifier un paquet pour diverses raisons, les trois plus intéressantes étant sans doute de le marquer à l'aide d'un champ spécial dans son en-tête afin de le reconnaître ultérieurement, de modifier sa priorité pour permettre un traitement privilégié ou au contraire placer en arrière plan ce type de paquet (libérant ainsi les ressources réseau pour d'autres types de connexion), et de modifier ses adresses source et destination, pour effectuer une translation d'adresse. Nous allons nous intéresser plus particulièrement aux différentes formes de translations d'adresses dans la suite de cette section.

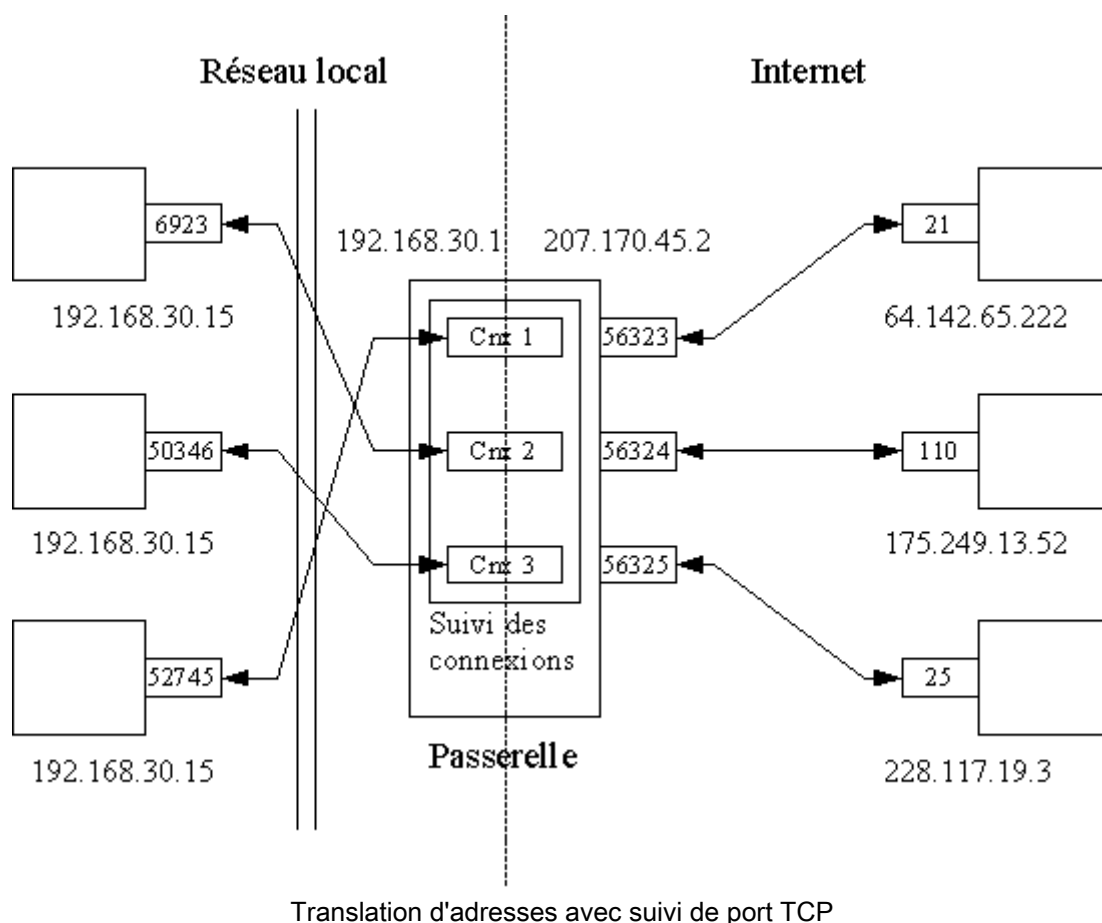
Une translation d'adresse n'est en fait rien d'autre qu'un remplacement contrôlé des adresses source ou destination de tous les paquets d'une connexion. En général, cela permet de détourner les connexions réseau ou de simuler une provenance unique pour plusieurs connexions. Par exemple, les connexions à Internet peuvent être détournées vers un proxy fonctionnant sur la passerelle de manière transparente, en modifiant les adresses destination des paquets émis par les clients. De même, il est possible de simuler un seul serveur en remplaçant à la volée les adresses source des paquets que la passerelle envoie sur Internet.

On distingue donc deux principaux types de translations d'adresses : les translations d'adresse source et les translations d'adresse destination. Parmi les translations d'adresse source se trouve un cas particulier particulièrement intéressant : le « masquerading ». Cette technique permet de remplacer toutes les adresses source des paquets provenant des machines d'un réseau local par l'adresse de l'interface de la passerelle connectée à Internet, tout en conservant une trace des connexions réseau afin d'acheminer vers leur véritable destinataire les paquets de réponses. Ainsi, une seule connexion à Internet peut être utilisée par plusieurs machines distinctes, même si elles ne disposent pas d'adresses IP fournies par l'IANA.

Lorsqu'une machine du réseau local envoie un paquet à destination d'Internet, ce paquet est acheminé vers la passerelle, qui est la route par défaut. Celle-ci commence par modifier l'adresse source de ce paquet et mettre sa propre adresse à la place, puis le transfère vers la machine du fournisseur d'accès. Tous les paquets émis par les machines du réseau local semblent donc provenir de cette passerelle, et seront acheminés normalement à destination. En fait, la complication provient des paquets de réponses, puisque les machines situées sur Internet croient que la machine avec laquelle elles communiquent est la passerelle. Ces paquets sont donc tous envoyés à la passerelle directement, et celle-ci doit retrouver la machine du réseau local à laquelle ils sont destinés. Cette opération est réalisée de différentes manières selon les protocoles utilisés, et elle suppose que la passerelle conserve en permanence une trace des connexions réseau effectuées par les machines du réseau local.

Pour TCP, ce suivi de connexion est réalisé en modifiant également le port source des paquets provenant des machines locales. La passerelle utilise un port unique pour chaque connexion, qui va lui permettre de retrouver la machine à laquelle un paquet est destiné lorsqu'elle en reçoit un provenant d'Internet. Par exemple, si une machine locale fait une connexion à Internet, la passerelle alloue un nouveau numéro de port pour cette connexion et modifie tous les paquets sortants comme ayant été émis par la passerelle elle-même, sur ce port. Lorsque l'autre machine prenant part à la connexion, située sur Internet, envoie un paquet de réponses, celui-ci sera à destination de la passerelle, avec comme port destination le port alloué à cette connexion. La passerelle peut donc retrouver, à partir de ce port, l'adresse de la machine destination réelle, ainsi que le port source que cette machine utilisait initialement.

La passerelle modifie donc ces deux champs du paquet, et le renvoie sur le réseau local. Finalement, la machine destination reçoit le paquet sur son port, et ce paquet semble provenir directement d'Internet, comme si la connexion avait été directe. Notez bien que la passerelle ne modifie pas les adresses source des paquets provenant d'Internet, elle ne fait que les réacheminer vers la bonne destination.



Ainsi, le masquering est un mécanisme complètement transparent pour les machines du réseau local. En revanche, pour les machines de l'Internet, il ne semble y avoir qu'un seul interlocuteur : la passerelle. Celle-ci utilise des numéros de ports variés, mais cela ne les regarde pas. Les machines du réseau local sont donc complètement « masquées » par la passerelle, d'où le nom de masquering.

Tout ce travail effectué par la passerelle nécessite un traitement spécifique sur chaque paquet qu'elle achemine, et consomme bien entendu des ressources système. Cependant, les débits utilisés pour les connexions à Internet, même les plus rapides, sont très loin de mettre à genoux une passerelle sous Linux, même sur les plus petites machines (386 et 486). Vous voilà rassurés, et peut-être aurez-vous trouvé d'ici peu une utilité à l'un de ces dinosaures qui traînent encore dans votre garage...

IX-D-3 - Trajet des paquets dans le code de Netfilter

Il existe cinq points d'entrée au total pour les modules de Netfilter dans le code des couches réseau IP de Linux. À chacun de ces points d'entrée, les paquets sont soumis aux différentes chaînes des tables qui s'y intéressent. Comme on l'a déjà dit plus haut, chaque table dispose d'une chaîne prédéfinie pour chacun des points d'entrée auxquels elle s'intéresse. Le nom de ces chaînes prédéfinies est le nom du point d'entrée au niveau duquel elles sont appelées. Les paragraphes suivants décrivent ces cinq points d'entrée, ainsi que l'ordre dans lequel les chaînes des différentes tables sont traversées par les paquets.

Les paquets réseau provenant de l'extérieur sont d'abord contrôlés par le noyau afin de détecter les erreurs les plus simples. Les paquets mal formés ou corrompus sont systématiquement éliminés à ce niveau, avant même

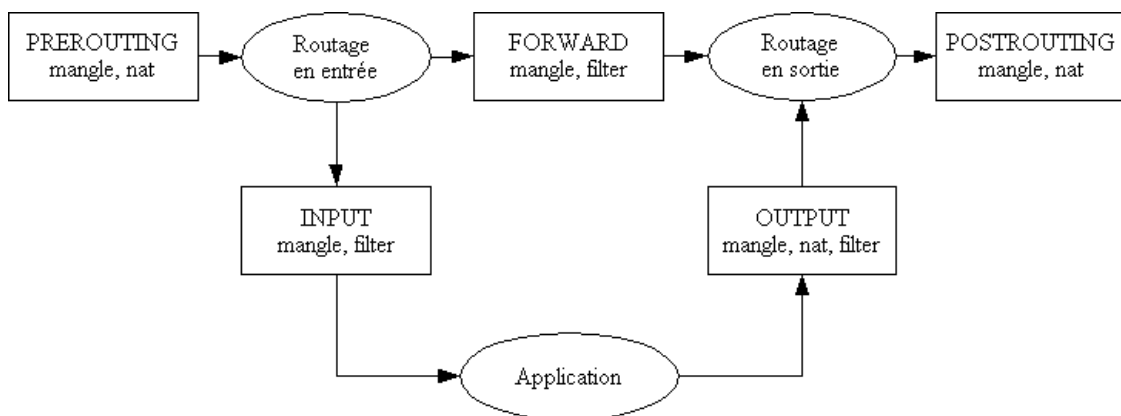
d'atteindre le code de Netfilter. Les autres paquets commencent alors leur trajet dans les couches de plus haut niveau. C'est à ce moment-là qu'ils rencontrent le point d'entrée PREROUTING. Les règles des chaînes PREROUTING des tables mangle et nat sont appliquées à ce moment. Ces chaînes précèdent le code de routage, qui est en charge de déterminer le trajet que le paquet devra suivre par la suite à partir de l'adresse destination des paquets. C'est donc dans ces chaînes que l'on pourra entre autres modifier l'adresse destination des paquets, si l'on veut faire une translation d'adresse destination. Cette opération se fait typiquement dans la table nat.

Les paquets qui passent le point d'entrée PREROUTING sont ensuite orientés par le code de routage de Linux. S'ils sont à destination de la machine locale, ils rencontrent un autre point d'entrée : le point d'entrée INPUT. Les règles des chaînes INPUT des tables mangle et filter sont exécutées à ce moment. C'est donc à ce niveau que l'on peut empêcher un paquet d'entrer dans le système, ce qui se fait naturellement dans la table filter.

Les autres paquets doivent être transférés vers une autre machine, souvent par une autre interface réseau. Ils atteignent donc le point d'entrée FORWARD. Ce point d'entrée permet de contrôler les paquets qui transitent au travers d'une passerelle. Les paquets y traversent les chaînes FORWARD des tables mangle et filter, cette dernière étant justement dédiée au filtrage des paquets qui transitent par la passerelle.

Les paquets qui sont acceptés par les chaînes FORWARD retournent ensuite dans le code de routage du noyau, afin de déterminer l'interface réseau par laquelle ils doivent être réémis. Une fois cela réalisé, ils sont prêts à être réémis, mais ils doivent auparavant traverser les chaînes des tables abonnées au point d'entrée POSTROUTING. Il s'agit bien sûr des chaînes POSTROUTING des tables mangle et nat. C'est dans cette dernière chaîne que se font les translations d'adresse source.

Enfin, les paquets émis par la machine locale à destination de l'extérieur doivent impérativement passer par le point d'entrée OUTPUT, qui exécute les chaînes OUTPUT des tables mangle, nat et filter. C'est donc à ce niveau que l'on pourra modifier les adresses destination des paquets de la machine locale avant leur routage et effectuer le filtrage des paquets sortants, dans les chaînes OUTPUT des tables nat et filter. Les paquets qui se révèlent avoir le droit de sortir traversent alors de code de routage, puis, tout comme les paquets qui ont été transférés, passent par le point d'entrée POSTROUTING.



Trajet des paquets dans le code de filtrage

IX-D-4 - Configuration du noyau et installation des outils

Les fonctionnalités de Netfilter sont toutes fournies par le noyau lui-même, aussi leur configuration doit-elle se faire au niveau du noyau. La manière de procéder pour configurer le noyau a été indiquée dans le paragraphe traitant de la compilation du noyau. Les options à activer pour mettre en place les fonctionnalités de filtrage des paquets et de translation d'adresse sont les suivantes :

- « Network packet filtering (replace ipchains) », pour activer les fonctionnalités de filtrage des paquets en général ;
- « Network packet filtering debugging », pour obtenir les messages de débogage du noyau. Ces messages peuvent être très utiles pour mettre au point les règles de filtrage ;

- « Connection tracking (required for masq/NAT) » (menu « IP: Netfilter Configuration »), pour permettre le suivi des connexions auxquelles appartiennent les paquets IP reçus par la passerelle. Cette option est nécessaire pour réaliser un partage de connexion à Internet ;
- « FTP protocol support », pour permettre la gestion des connexions FTP, qui exigent un traitement particulier pour les translations d'adresses ;
- « IP tables support (required for filtering/masq/NAT) », pour permettre la gestion des tables de Netfilter ;
- « Packet filtering », pour inclure le support de la table « filter ». Cette option est nécessaire pour réaliser un pare-feu ;
- « REJECT target support », pour permettre le rejet des paquets (par défaut, seul l'abandon des paquets est fourni dans le code de filtrage) ;
- « Full NAT », pour inclure la table « nat » de Netfilter. Cette option est nécessaire pour réaliser un partage de connexion à Internet ;
- « MASQUERADE target support », pour permettre le masquering afin de réaliser un partage de connexion à Internet.

Vous devrez ensuite compiler le noyau et les modules, et les installer comme indiqué dans la partie traitant de la [compilation du noyau](#).

Note : Le noyau de Linux dispose d'un paramètre global permettant de contrôler le routage des paquets IP d'une interface réseau à une autre. Par défaut, ce paramètre est à 0, ce qui implique que la transmission des paquets ne sera pas autorisée. Il faut donc activer ce paramètre à chaque démarrage afin de pouvoir utiliser votre passerelle Linux. Cela peut être réalisé en écrivant la valeur 1 dans le fichier `/proc/sys/net/ipv4/ip_forward` du système de fichiers virtuel `/proc/` :

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

La manipulation des chaînes et la manipulation des règles de Netfilter se fait grâce à l'outil iptables. Cette commande doit normalement avoir été installée par votre distribution. Toutefois, si ce n'est pas le cas, vous pouvez la compiler manuellement. Pour cela, il vous faudra récupérer l'archive des sources d'iptables. Cette archive peut être trouvée [sur Internet](#). La version courante est la 1.2.9, aussi l'archive se nomme-t-elle iptables-1.2.6a.tar.bz2.

L'installation d'iptables ne pose pas de problème particulier en soi. Vous devrez d'abord décompresser l'archive avec les deux commandes suivantes :

```
bunzip2 iptables-1.2.9.tar.bz2
tar xf iptables-1.2.9.tar
```

puis modifier le fichier Makefile pour définir les répertoires d'installation dans les variables LIBDIR, BINDIR et MANDIR. En général, les répertoires utilisés seront respectivement les répertoires `/usr/lib/`, `/usr/bin/` et `/usr/man/`. Une fois cela fait, il ne restera plus qu'à compiler le tout avec la commande make et à faire l'installation avec la commande make install.

IX-D-5 - Utilisation d'iptables

Il est possible, grâce à iptables, d'effectuer toutes les opérations d'administration désirées sur les chaînes des tables de Netfilter. Vous pourrez donc créer des nouvelles chaînes, les détruire, définir les politiques par défaut, ainsi que manipuler les règles de ces chaînes (en ajouter, en supprimer ou en remplacer une). En fait, iptables dispose d'un grand nombre d'options, et seules les options les plus importantes seront présentées ici. Vous pouvez lire la page de manuel iptables si vous désirez plus de renseignements sur la manipulation des chaînes et des règles de filtrage du noyau.

IX-D-5-a - Manipulation des chaînes

Toutes les options peuvent être utilisées avec toutes les tables gérées par le noyau. La table sur laquelle une commande s'applique peut être précisée à l'aide de l'option -t. Si cette option n'est pas fournie, la table utilisée par

défaut sera la table `filter`, qui est celle qui est normalement utilisée pour définir des règles de filtrage des paquets dans un pare-feu.

La création d'une nouvelle chaîne se fait simplement avec l'option `-N` :

```
iptables [-t table] -N chaîne
```

où `chaîne` est le nom de la chaîne à créer. Il est interdit d'utiliser un des mots clés réservés par `iptables`. En pratique, il est recommandé d'utiliser des noms de chaînes en minuscules, car les chaînes prédéfinies sont en majuscules.

Une chaîne peut être détruite avec l'option `-X` :

```
iptables [-t table] -X chaîne
```

Une chaîne ne peut être détruite que lorsqu'elle ne contient plus de règle. De plus, il est impossible de détruire les chaînes prédéfinies d'une table.

Vous pouvez lister l'ensemble des règles d'une chaîne avec l'option `-L` :

```
iptables [-t table] -L chaîne
```

Enfin, il est possible de supprimer toutes les règles d'une chaîne avec l'option `-F` :

```
iptables [-t table] -F chaîne
```

où `chaîne` est le nom de la chaîne dont on veut supprimer les règles.

IX-D-5-b - Manipulation des règles

La syntaxe générale pour ajouter une règle dans une chaîne est la suivante :

```
iptables [-t table] -A chaîne [-s source] [-d destination] [-p protocole]
[-i itfsource] [-o itfdest] [-j cible]
```

où :

- `table` est la table dans laquelle se trouve la chaîne manipulée (par défaut, il s'agit de la table `filter`) ;
- `chaîne` est le nom de la chaîne à laquelle la règle doit être ajoutée ;
- `source` est l'adresse source du paquet ;
- `destination` est l'adresse destination du paquet ;
- `protocole` est le protocole du paquet (spécifié avec son numéro de port ou par le nom du protocole tel qu'il est déclaré dans le fichier `/etc/services`) ;
- `itfsource` est le nom de l'interface réseau source par laquelle le paquet doit arriver ;
- `itfdest` est le nom de l'interface destination par laquelle le paquet doit sortir ;
- `cible` est la cible des paquets qui vérifient les critères de sélection de la règle.

Chacun des paramètres placés entre crochets dans la syntaxe est facultatif, mais il faut au moins qu'un critère de sélection soit donné (sur les adresses source ou destination, ou sur le port ou l'interface).

Les adresses source et destination peuvent être indiquées directement par le nom des machines (comme par exemple « `www.monrezo.com` ») ou par leurs adresses IP. Vous pouvez spécifier directement toutes les adresses d'un réseau avec la syntaxe « `réseau/masque` ». Il est également possible d'utiliser la syntaxe « `réseau/n` », où « `n` » est le nombre de bits mis à 1 dans le masque de sous-réseau. Ainsi, `réseau/24` est équivalent à `réseau/255.255.255.0`, `réseau/16` à `réseau/255.255.0.0`, etc. Par extension, « `réseau/0` » peut être utilisé pour spécifier toutes les adresses possibles (dans ce cas, l'adresse donnée pour le réseau n'est pas prise en compte).

La cible détermine les opérations qui seront appliquées à ces paquets. Les cibles autorisées dépendent de la chaîne dans laquelle la règle est ajoutée, ainsi que des modules de Netfilter qui ont été compilés. Les principales cibles sont les suivantes :

- **ACCEPT**, pour accepter le paquet qui vérifie le critère de sélection de la règle ;
- **DROP**, pour éliminer purement et simplement le paquet ;
- **REJECT**, pour rejeter le paquet (en signalant l'erreur à la machine émettrice). Cette cible n'est utilisable que dans les chaînes **INPUT**, **FORWARD** et **OUTPUT**, ainsi que dans les chaînes utilisateur appelées à partir de ces chaînes ;
- **QUEUE**, pour envoyer le paquet à un programme utilisateur capable de communiquer avec NetFilter ;
- **RETURN**, pour sortir de la chaîne immédiatement, ou appliquer la règle de la politique par défaut pour les chaînes prédéfinies ;
- **REDIRECT**, pour rediriger le paquet sur une autre machine, souvent la machine locale. Cette cible n'est utilisable qu'avec la table **nat**, car il s'agit d'une translation d'adresse. On ne peut l'utiliser que dans les chaînes **PREROUTING** et **OUTPUT** et les chaînes utilisateur appelées à partir de ces chaînes ;
- **SNAT**, pour permettre la modification de l'adresse source du paquet. Cette cible n'est bien entendu accessible qu'au niveau de la table **nat**. Comme la modification de l'adresse source n'a de signification que pour les paquets devant sortir de la passerelle, cette cible ne peut être utilisée que dans la chaîne **POSTROUTING** et les chaînes utilisateur appelées à partir de cette chaîne ;
- **DNAT**, pour effectuer la modification de l'adresse destination du paquet, afin par exemple de le détourner vers une autre machine que celle vers laquelle il devait aller. Cette cible n'est accessible que dans la table **nat**, et n'est utilisable que dans les chaînes **PREROUTING** et **OUTPUT** ainsi que dans les chaînes utilisateur appelées à partir de ces chaînes ;
- **MASQUERADE**, pour effectuer une translation d'adresse sur ce paquet, dans le but de réaliser un partage de connexion à Internet. Cette cible n'est accessible que dans la chaîne **POSTROUTING** de la table **nat**, ainsi que dans les chaînes utilisateur appelées à partir de cette chaîne.

Toute autre spécification de destination doit être le nom d'une chaîne utilisateur, avec les règles de laquelle le paquet devra être testé. Si aucune cible n'est spécifiée, aucune action n'est prise et la règle suivante est traitée. Cependant, les statistiques de la règle sont toujours mises à jour.

La suppression d'une règle d'une chaîne se fait avec la commande suivante :

```
iptables -D chaîne numéro
```

où chaîne est la chaîne dont on veut supprimer la règle, et numéro est le numéro de la règle à supprimer dans cette chaîne. Il est également possible d'utiliser l'option **-D** avec les mêmes options que celles qui ont été utilisées lors de l'ajout de la règle, si l'on trouve cela plus pratique.

Enfin, la politique d'une chaîne, c'est-à-dire la cible par défaut, peut être fixée avec l'option **-P** :

```
iptables -P chaîne cible
```

où chaîne est l'une des chaînes prédéfinies, et cible est l'une des cibles **ACCEPT** ou **DROP**. Remarquez que l'on ne peut pas définir de politique pour les chaînes créées par l'utilisateur, puisque les paquets sont vérifiés avec les règles restantes de la chaîne appelante lorsqu'ils sortent de la chaîne appelée.

IX-D-6 - Exemple de règles

Ce paragraphe a pour but de présenter quelques-unes des règles les plus classiques. Le but est ici de présenter les principales fonctionnalités d'iptables et de réaliser un pare-feu simpliste.

IX-D-6-a - Exemple de règles de filtrage

Les règles de filtrage peuvent être utilisées pour mettre en place un pare-feu afin de protéger votre réseau. La définition des règles de filtrage constitue bien souvent l'essentiel du problème, et nécessite bien entendu de parfaitement savoir ce que l'on veut faire. De manière générale, la règle d'or en matière de sécurité informatique est de tout interdire par défaut, puis de donner les droits au compte-goutte. C'est exactement ce que permet de faire la politique des chaînes. On définira donc toujours la politique par défaut des chaînes pour utiliser la cible DROP. Cela peut être réalisé simplement avec les trois commandes suivantes :

```
# Définition de la politique de base :
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP

# Suppression des règles existantes :
iptables -F INPUT
iptables -F FORWARD
iptables -F OUTPUT
```

Ces règles commencent par définir les politiques de base pour toutes les tables en blindant la machine pour qu'elle ne laisse rien entrer, ne rien passer et ne rien sortir. Toutes les règles des tables sont ensuite détruites. Dans ce cas, on ne craint plus rien, mais on ne peut plus rien faire non plus (pas même en local). Aussi faut-il redonner les droits nécessaires pour permettre l'utilisation normale de votre système. Le minimum est d'autoriser la machine à communiquer avec elle-même, ce qui implique d'autoriser les paquets provenant de l'interface loopback à entrer et à sortir :

```
# Communications locales :
iptables -A INPUT -d 127.0.0.0/8 -i lo -j ACCEPT
iptables -A OUTPUT -s 127.0.0.0/8 -o lo -j ACCEPT
```

Si l'on dispose d'un réseau local, il peut également être utile d'autoriser toutes les connexions avec les machines de ce réseau :

```
# Connexions au réseau local :
iptables -A INPUT -s 192.168.0.0/24 -i eth0 -j ACCEPT
iptables -A OUTPUT -s 192.168.0.1 -o eth0 -j ACCEPT
```

Comme vous pouvez le constater, ces règles ne permettent l'entrée des paquets prétendant provenir du réseau local (c'est-à-dire les paquets ayant comme adresse source une adresse de la forme 192.168.0.0/24) que par l'interface réseau connectée au réseau local (dans notre exemple, il s'agit de l'interface réseau eth0). De même, seule la machine locale (à laquelle l'adresse 192.168.0.1 est supposée être affectée) peut émettre des paquets sur le réseau local.

Notez que ces règles sont beaucoup trop restrictives pour permettre un accès à Internet (et, a fortiori une intrusion provenant d'Internet...). Il est donc nécessaire d'autoriser les connexions vers les principaux services Internet. Pour cela, il faut ajouter les lignes suivantes :

```
# Autorise les communications sortantes
# pour DNS, HTTP, HTTPS, FTP, POP, SMTP, SSH et IPSec :
iptables -A OUTPUT -o ppp0 -p UDP --dport domain -j ACCEPT
iptables -A OUTPUT -o ppp0 -p TCP --dport domain -j ACCEPT
iptables -A OUTPUT -o ppp0 -p TCP --dport http -j ACCEPT
iptables -A OUTPUT -o ppp0 -p TCP --dport https -j ACCEPT
iptables -A OUTPUT -o ppp0 -p TCP --dport ftp -j ACCEPT
iptables -A OUTPUT -o ppp0 -p TCP --dport pop3 -j ACCEPT
iptables -A OUTPUT -o ppp0 -p TCP --dport smtp -j ACCEPT
iptables -A OUTPUT -o ppp0 -p TCP --dport ssh -j ACCEPT
iptables -A OUTPUT -o ppp0 -p AH -j ACCEPT
iptables -A OUTPUT -o ppp0 -p ESP -j ACCEPT
```

Il est supposé ici que la connexion Internet se fait via le démon pppd, et que l'interface réseau que celui-ci a créée se nomme ppp0.

Vous pouvez constater que les règles présentées jusqu'ici se basaient uniquement sur les adresses source et destination des paquets, ainsi que sur les interfaces réseau par lesquelles ils entraient et ressortaient. Vous voyez à présent que l'on peut réaliser des règles beaucoup plus fines, qui se basent également sur les protocoles réseau utilisés (que l'on peut sélectionner avec l'option -p) et, pour chaque protocole, sur des critères spécifiques à ces protocoles (comme les ports source et destination, que l'on peut sélectionner respectivement avec les options --sport et --dport). Vous pouvez consulter la page de manuel d'iptables pour plus de détails à ce sujet.

Certains protocoles requièrent plusieurs connexions TCP/IP pour fonctionner correctement. La connexion est amorcée avec une première connexion, puis une deuxième connexion est établie après négociation. C'est en particulier le cas pour le protocole FTP, qui utilise une connexion de contrôle et une connexion pour le transfert des données. De plus, le protocole FTP peut fonctionner dans deux modes distincts (respectivement le mode dit « actif » et le mode dit « passif »), selon que c'est le serveur contacté qui établit la connexion de données ou que c'est le client qui l'établit vers le serveur. Ces connexions additionnelles ne doivent bien entendu pas être filtrées, pour autant, on ne sait pas avec précision le port qui sera choisi pour elles, car ce port fait partie de la négociation lors de l'établissement de la connexion. Heureusement, Netfilter dispose d'un mécanisme de suivi de connexions capable d'analyser les données protocolaires (si le protocole n'est pas chiffré, bien entendu). La solution est donc de lui indiquer qu'il doit laisser les paquets relatifs aux connexions établies au niveau protocolaire. La règle à utiliser est la suivante :

```
# Autorise l'établissement des connexions protocolaires additionnelles :
iptables -A OUTPUT -o ppp0 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Vous découvrez ici l'option --match, qui permet de sélectionner des paquets selon certains critères. Le critère choisi ici est bien entendu l'état des connexions. Des options complémentaires peuvent être fournies à ce critère à l'aide de l'option --state. Dans le cas présent, nous acceptons tous les paquets sortants qui appartiennent à une connexion établie (paramètre « :ESTABLISHED ») ou qui a été amorcée à partir d'une autre connexion déjà établie (paramètre « RELATED »).

Les règles précédentes permettent à la machine locale de se connecter à Internet, mais la connexion ne s'établira malgré tout pas, car aucun paquet réponse n'a le droit d'entrer. Il est donc nécessaire d'autoriser l'entrée des paquets réponse provenant des serveurs auxquels on se connecte. Toutefois, il ne faut pas tout autoriser. Seuls les paquets relatifs aux connexions dont la machine locale est initiatrice doivent être autorisés. Cela est réalisable encore une fois grâce au mécanisme de suivi de connexions de Netfilter. Pour notre exemple, la simple règle suivante suffit :

```
# Autorise les paquets entrants pour les connexions
# que l'on a établies :
iptables -A INPUT -i ppp0 --match state --state ESTABLISHED,RELATED -j ACCEPT
```

Elle dit que les paquets entrants des connexions établies et des connexions relatives aux autres connexions sont autorisés. Tous les autres paquets seront refusés.

Si vous désirez laisser un service ouvert vers Internet, vous devrez ajouter des règles complémentaires qui autorisent les paquets entrant et sortant à destination du port de ce service. Par exemple, si vous voulez laisser un accès SSH ouvert, vous devrez ajouter les deux règles suivantes :

```
# Autorise les connexions SSH entrantes :
iptables -A INPUT -i ppp0 -p TCP --dport ssh -j ACCEPT
iptables -A OUTPUT -o ppp0 -p TCP --sport ssh -j ACCEPT
```

Enfin, il peut être utile d'accepter certains paquets du protocole ICMP, qui permet de contrôler l'état des liaisons. En particulier, il faut pouvoir recevoir les paquets indiquant qu'une machine n'est pas accessible, faute de quoi certaines connexions impossibles risqueraient d'être très longues à détecter. De plus, il peut être intéressant de pouvoir envoyer des demandes de réponse aux machines pour vérifier leur présence. Les règles suivantes pourront donc être ajoutées :


```
# Autorise les paquets ICMP intéressants :
iptables -A OUTPUT -p ICMP --icmp-type echo-reply -j ACCEPT
iptables -A INPUT -p ICMP --icmp-type echo-request -j ACCEPT
iptables -A OUTPUT -p ICMP --icmp-type echo-request -j ACCEPT
iptables -A INPUT -p ICMP --icmp-type echo-reply -j ACCEPT
iptables -A INPUT -p ICMP --icmp-type destination-unreachable -j ACCEPT
```

Notez qu'il n'est théoriquement pas nécessaire de répondre aux paquets echo-request. Je vous ai toutefois laissé les deux règles pour le cas où le fournisseur d'accès utiliserait ces requêtes pour déterminer si la liaison reste correcte.

IX-D-6-b - Exemple de partage de connexion à Internet

Le masquerading est un cas particulier de translation d'adresse source. En effet, celle-ci est couplée à un suivi des connexions afin d'effectuer la translation d'adresse destination des paquets de réponses renvoyés par les serveurs sur Internet pour les acheminer vers la machine du réseau local qui a initié la connexion.

En pratique, seule la chaîne POSTROUTING de la table nat sera utilisée pour le masquerading, parce que c'est à ce niveau que la translation d'adresse est effectuée. La mise en œuvre du masquerading se fait extrêmement simplement, puisqu'il suffit de spécifier que tous les paquets du réseau local sortant de l'interface réseau connectée sur Internet doivent subir le masquerading. En général, l'interface de la connexion à Internet est une interface PPP, aussi la règle à utiliser est-elle simplement la suivante :

```
# Effectue la translation d'adresses :
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o ppp0 -j MASQUERADE
```

Cette règle permet de réaliser la translation des adresses des machines du réseau local et à destination d'Internet. Remarquez que les paquets provenant de la machine locale ne subissent pas le masquerading, car c'est complètement inutile.

Il faut ensuite autoriser les paquets provenant des machines du réseau local à traverser la passerelle. Cela se fait avec des règles similaires à celles vues dans la section précédente, mais adaptées pour être utilisées dans la chaîne FORWARD de la table filter :

```
# Autorise le trafic DNS, HTTP, HTTPS,
# FTP, POP, SMTP et SSH au travers de la passerelle :
iptables -A FORWARD -s 192.168.0.0/24 -i eth0 -o ppp0 -p UDP --dport domain -j ACCEPT
iptables -A FORWARD -s 192.168.0.0/24 -i eth0 -o ppp0 -p TCP --dport domain -j ACCEPT
iptables -A FORWARD -s 192.168.0.0/24 -i eth0 -o ppp0 -p TCP --dport http -j ACCEPT
iptables -A FORWARD -s 192.168.0.0/24 -i eth0 -o ppp0 -p TCP --dport https -j ACCEPT
iptables -A FORWARD -s 192.168.0.0/24 -i eth0 -o ppp0 -p TCP --dport ftp -j ACCEPT
iptables -A FORWARD -s 192.168.0.0/24 -i eth0 -o ppp0 -p TCP --dport pop3 -j ACCEPT
iptables -A FORWARD -s 192.168.0.0/24 -i eth0 -o ppp0 -p TCP --dport smtp -j ACCEPT
iptables -A FORWARD -s 192.168.0.0/24 -i eth0 -o ppp0 -p TCP --dport ssh -j ACCEPT

# Autorise l'établissement des connexions protocolaires additionnelles :
iptables -A FORWARD -s 192.168.0.0/24 -i eth0 -o eth0 --match state \
    --state ESTABLISHED,RELATED -j ACCEPT

# Autorise le trafic en retour pour les connexions établies par les clients :
iptables -A FORWARD -i ppp0 -d 192.168.0.0/24 --match state \
    --state ESTABLISHED,RELATED -j ACCEPT

# Autorise le trafic ICMP :
iptables -A FORWARD -p ICMP --icmp-type echo-request -j ACCEPT
iptables -A FORWARD -p ICMP --icmp-type echo-reply -j ACCEPT
iptables -A FORWARD -p ICMP --icmp-type destination-unreachable -j ACCEPT
```

Enfin, le routage des paquets étant, par défaut, désactivé sous Linux, il faut le réactiver. Si votre distribution ne le fait pas, vous aurez donc également à ajouter une ligne telle que celle-ci :

```
# Activation du routage :
echo "1" > /proc/sys/net/ipv4/ip_forward
```

dans vos scripts de démarrage de votre système.

Certains routeurs et certaines passerelles d'Internet sont bogués et ne répondent pas correctement aux messages ICMP. Cela a pour conséquence, entre autres, que la taille maximum des paquets transmis via la passerelle peut ne pas être déterminée correctement. En effet, les clients de la passerelle utiliseront généralement la taille maximum utilisable pour communiquer avec la passerelle, et cette taille peut malheureusement être supérieure à la taille maximum des paquets pour atteindre certains sites Web. Les messages d'erreurs ICMP étant filtrés, personne ne peut s'en apercevoir, et certaines connexions peuvent rester bloquées ad vitam eternam. Pratiquement parlant, cela se traduit par l'impossibilité, pour certains clients bénéficiant du partage de connexion à Internet, d'accéder à certains sites Web ou à certains gros fichiers d'une page (images par exemple).

Pour pallier ce problème, Linux propose une technique approximative, dont le but est de fixer une option des paquets TCP d'établissements de connexion, afin de limiter la taille des paquets suivants de la connexion à la taille maximum déterminée pour la route vers les ordinateurs cibles. Ainsi, les clients se limitent d'eux-mêmes, pourvu que la taille maximum des paquets de la route soit correctement déterminée, bien entendu. Il est recommandé d'activer cette fonctionnalité, ce qui se fait à l'aide de la commande suivante :

```
# Restreint la taille des paquets des connexions TCP
# lors de leur établissement :
iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu
```

Cette règle un peu compliquée permet de modifier le champ MSS (« Maximum Segment Size ») des paquets d'établissements des connexions TCP (paquets disposant des flags TCP SYN ou RST) qui traversent la passerelle, et de forcer ce champ à la valeur de la taille maximum des paquets sur le chemin (« Path Maximum Transmission Unit » dans l'option `--clamp-mss-to-pmtu`).

Note : Les chaînes et leurs règles ne sont pas enregistrées de manière permanente dans le système. Elles sont perdues à chaque redémarrage de la machine, aussi faut-il les redéfinir systématiquement. Cela peut être réalisé dans les scripts de démarrage de votre système. Vous pouvez également utiliser les commandes `iptables-save` et `iptables-restore` pour sauvegarder toutes les chaînes et leurs règles dans un fichier et les restaurer au démarrage.

N'oubliez pas que votre passerelle doit définir une route par défaut pour que tous les paquets qui ne sont pas destinés au réseau local soient envoyés par l'interface réseau connectée à Internet. Cette route par défaut est établie automatiquement lorsque vous vous connectez à Internet à l'aide de PPP. Dans les autres cas, vous aurez à la définir manuellement.

IX-D-7 - Configuration des clients

La configuration des autres machines du réseau est très simple. Vous devrez tout d'abord définir la machine possédant la connexion à Internet comme passerelle par défaut de tous vos postes clients. Cette opération peut se réaliser de différentes manières selon le système d'exploitation utilisé par ces machines. Sous Linux, il suffit d'utiliser la commande suivante :

```
route add default gw passerelle eth0
```

où `passerelle` est l'adresse de votre passerelle vers Internet sur le réseau local. Cette commande suppose que l'adaptateur réseau utilisé est `eth0`

La deuxième étape est ensuite de donner accès aux postes clients au DNS de votre fournisseur d'accès. Cela permettra en effet d'utiliser les noms de machines autres que ceux de votre réseau local. Encore une fois, la manière de procéder dépend du système utilisé. Sous Linux, il suffit d'ajouter les adresses IP des serveurs de noms de domaine de votre fournisseur d'accès dans le fichier de configuration `/etc/resolv.conf`.

Si les clients ont des difficultés à accéder à certaines pages Web, c'est que le MTU (« Maximum Transmission Unit », taille maximum des paquets réseau) est trop élevé. Ceci se corrige généralement tel qu'on l'a indiqué dans la

définition des règles de la passerelle. Toutefois, si les ordinateurs qui bénéficient du partage de connexion à Internet continuent d'avoir des problèmes, c'est que le MTU de la route permettant d'accéder aux sites Web posant problème n'a pas pu être déterminé correctement. Dans ce cas, il faut le trouver soi-même, et le fixer dans la configuration réseau des clients. Pour cela, le plus simple est d'utiliser la commande ping avec son option -s sur les clients, avec plusieurs tailles de paquets possible. La taille maximum est de 1500 octets sur Ethernet. En général, une taille de 1400 octets convient :

```
ping -s 1400 adresse
```

Une fois le MTU correctement déterminé, il suffit de fixer cette valeur lors de la configuration des interfaces réseau des clients :

```
ifconfig interface mtu taille
```

`interface` représente ici l'interface à configurer, et `taille` la taille maximum trouvée à l'aide de ping.

Note : La taille effective du MTU est en réalité la taille fournie à la commande ping plus vingt-huit octets, car les paquets utilisés contiennent toujours l'en-tête ICMP en plus des données générées par ping.

IX-E - Configuration de la sécurité du réseau

La sécurité d'un réseau n'est pas quelque chose à prendre à la légère, surtout lorsqu'on commence à transmettre des données personnelles sur le réseau. Internet n'est absolument pas sûr, car nombre de personnes curieuses, peu discrètes ou franchement mal intentionnées ont un accès privilégié aux machines qui s'y trouvent. Il est donc essentiel de prendre les problèmes sécuritaires très au sérieux, d'autant plus que Linux est un système disposant d'un grand nombre de fonctionnalités serveur.

Bien qu'il soit impossible de réaliser un système sûr à 100%, la situation actuelle est franchement catastrophique et si cela continue comme cela, la seule chose de sûre, c'est qu'on va aller droit dans le mur. Entre ne rien faire du tout et se fixer un niveau de sécurité raisonnable permettant de limiter les dégâts, il y a un grand pas. Cette section n'a donc pas pour but de vous transformer en expert en sécurité (n'en étant pas un moi-même, je serais bien incapable d'en écrire une ayant cet objectif), mais de vous présenter les principes de base et le minimum requis pour laisser les mauvaises surprises aux autres.

L'un des principes de base de la sécurité est de ne faire confiance à personne. Ce n'est pas un comportement paranoïaque, mais simplement du bon sens : la plupart des surprises désagréables dans le monde environnant proviennent du mauvais jugement des personnes. De plus, même des personnes de bonne foi peuvent être manipulées par un tiers sans qu'elles s'en rendent compte.

Partant de ce constat, et quand bien même on serait capable de garantir la fiabilité d'un système à 100%, il est nécessaire de limiter et de cloisonner les services fournis, et d'en restreindre l'accès aux personnes habilitées. Cela est d'autant plus important que la plupart des démons ont besoin d'avoir les privilèges du compte root pour s'exécuter. Cela peut être dramatique si d'aventure une personne mal intentionnée découvrait le moyen de les détourner de leur fonction initiale et de leur faire accomplir des tâches au nom de l'administrateur. Bien entendu, les démons sont des programmes très bien écrits, qui vérifient généralement toutes les requêtes demandées par les clients avant de les satisfaire. Cependant, tout programme peut comporter des failles et laisser ainsi un accès indésiré au sein du système. Autant éliminer directement le problème en ne lançant que les services réellement nécessaires sur votre machine.

De plus, il va de soi que lorsqu'il y a un loup dans la bergerie, tout peut aller très mal très vite. La sécurité doit donc être mise en place en profondeur, c'est-à-dire que l'on ne doit pas négliger la sécurisation d'un réseau local sous prétexte qu'il est protégé par des dispositifs en amont. En effet, ces dispositifs peuvent très bien être défectueux, et dans ce cas, l'intrus se retrouve directement en terrain conquis. C'est pour cela qu'il faut prévoir plusieurs remparts pour se protéger.

En outre, rien n'est jamais figé dans le domaine de la sécurité. De nouvelles attaques sont développées régulièrement, et de nouvelles failles sont découvertes dans les logiciels quasiment quotidiennement. Il est donc nécessaire de réaliser un tant soit peu de veille et de mettre à jour les systèmes régulièrement.

Enfin, la sécurité ne peut se concevoir que dans un contexte global, en prenant compte l'environnement dans lequel on se trouve. Il est tout à fait inutile d'utiliser des systèmes d'authentification par analyse de code génétique pour protéger des documents ultra-sensibles, si ces documents peuvent être sortis simplement par les poubelles.

En résumé, les principes fondamentaux de la sécurité sont les suivants :

- il faut limiter le nombre des services lancés au strict minimum ;
- il faut en restreindre l'accès aux seules personnes autorisées ;
- il faut être proactif, se tenir au courant et mettre à jour les logiciels dès l'apparition des correctifs de sécurité ;
- il faut prévoir plusieurs systèmes de sécurité ;
- il faut prendre en compte les problèmes de sécurité de manière globale.

De tous ces points, nous ne traiterons ici que ceux qui concernent un système de particulier. La mise à jour des logiciels et des antivirus reste bien entendu extrêmement importante (surtout si des machines Windows sont placées sur le réseau), mais n'appelle pas beaucoup de commentaires. La description de la réalisation d'un pare-feu a également déjà été faite dans la section précédente. Nous ne présenterons donc que la manière de limiter les services lancés et l'accès à ces services. Cela n'est toutefois pas suffisant pour dormir tranquille, car, et c'est sans doute là l'un des plus gros problèmes, la plupart des protocoles réseau ne chiffrent pas leurs données et laissent passer toutes les informations en clair sur le réseau, y compris les mots de passe ! Nous verrons donc également les principales techniques de cryptographie existantes afin de rendre les communications plus sûres.

IX-E-1 - Limitation des services et des accès

IX-E-1-a - Réduction du nombre des services

Les services réseau d'un système Linux sont nombreux, mais la plupart du temps inutiles pour l'emploi que l'on veut faire de son ordinateur. Par conséquent, il vaut mieux tout simplement ne pas les proposer au monde extérieur, simplement en ne lançant pas les démons correspondants. Pour cela, il suffit de commenter les lignes des services inutiles dans le fichier `/etc/inetd.conf` ou d'ajouter une ligne « `disable = true` » dans les fichiers de configuration des services inutiles du répertoire `/etc/xinetd.d/` si votre distribution utilise le super-démon `xinetd`. Faire le ménage dans ces fichiers et dans les fichiers d'initialisation du réseau réduit donc de 90% les risques d'une intrusion dans votre système.

IX-E-1-b - Définition de règles de contrôle d'accès

Restreindre le nombre de services accessibles n'est toutefois pas suffisant, il faut également restreindre le nombre des personnes et des machines auxquels les services réseau restants sont proposés. Sur une machine de particulier, l'équation est souvent simple : personne ne doit pouvoir s'y connecter par le réseau, pour quelque service que ce soit ! Bien entendu, certains particuliers possèdent un réseau privé et devront donc affiner cette règle pour ne permettre les connexions que sur le réseau local. Il faut donc être capable d'empêcher les accès provenant de certaines machines, considérées comme peu fiables. La règle est encore une fois très simple : doit être considérée comme dangereuse tout machine étrangère au réseau local.

La manière de procéder dépend du super-démon utilisé. En effet, alors que `xinetd` permet de définir les adresses des machines autorisées à se connecter de manière précise, `inetd` ne permet pas de le faire lui-même et requiert l'aide d'un autre démon, le démon `tcpd`.

IX-E-1-b-i - Restrictions d'accès avec tcpd

Le principe de fonctionnement du démon tcpd est le suivant : il s'intercale entre le super-démon inetd et les démons que celui-ci est supposé lancer lorsqu'un client le demande. Ainsi, il lui est possible d'effectuer des vérifications de sécurité primaires, principalement basées sur les adresses IP des clients.

Comme vous pouvez sans doute le constater dans le fichier `/etc/inetd.conf` de votre système, tcpd est lancé par inetd pour quasiment tous les services, et reçoit en paramètres de la ligne de commande le nom du démon à lancer si la machine cliente en a l'autorisation, ainsi que les paramètres à communiquer à ce démon. L'accès à quasiment tous les services est donc contrôlé par tcpd.

tcpd utilise les deux fichiers de configuration `/etc/hosts.allow` et `/etc/hosts.deny` pour déterminer si une machine a le droit d'accéder à un service donné. Le premier de ces fichiers indique quelles sont les machines autorisées à utiliser certains services locaux. Le deuxième fichier indique au contraire les machines qui ne sont pas considérées comme sûres et qui doivent se voir refuser toute demande de connexion sur les services locaux.

Le format de ces deux fichiers est identique. Il est constitué de lignes permettant de définir les règles d'accès pour certains démons. Chaque ligne utilise la syntaxe suivante :

```
démons : machines [: commande]
```

où `démons` est une liste de noms de démons, séparés par des virgules, et `machines` est une liste de noms de machines ou d'adresses IP, également séparés par des virgules. `commande` est un paramètre optionnel permettant de faire exécuter une action à tcpd lorsque la règle indiquée par cette ligne est prise en compte.

Une règle définie par une de ces lignes est utilisée dès qu'une des machines indiquées dans la liste des machines tente une connexion à l'un des services de la liste des services. Si la règle se trouve dans le fichier `/etc/hosts.allow`, la connexion est autorisée et le service est lancé par tcpd. Si elle se trouve dans le fichier `/etc/hosts.deny`, la connexion est systématiquement refusée. Si aucune règle n'est utilisée, la connexion est acceptée par défaut.

Les listes de machines peuvent contenir des noms de machines partiels, et utiliser des caractères génériques. Il est également possible d'interdire la connexion à toutes les machines d'un domaine en ne donnant que le nom de domaine (précédé d'un point). Enfin, des mots clés spéciaux permettent de représenter des ensembles de machines. Par exemple, le mot clé `ALL` représente toutes les machines et `LOCAL` représente toutes les machines du réseau local. Le mot clé `ALL` permet également de représenter l'ensemble des démons dans la liste des démons.

Par exemple, le fichier `/etc/hosts.deny` devrait contenir la ligne suivante :

```
# Exemple de fichier /etc/hosts.deny :  
ALL : ALL
```

Cela permet de garantir que, par défaut, aucune demande de connexion n'est acceptée, ce qui est un comportement sain. Les machines ayant le droit de se connecter doivent être spécifiées au cas par cas dans le fichier `/etc/hosts.allow`, comme dans l'exemple suivant :

```
# Exemple de fichier /etc/hosts.allow :  
telnetd, ftpd : LOCAL
```

Cela permet d'autoriser les connexions telnet et ftp pour toutes les machines du réseau local uniquement.

Vous trouverez de plus amples renseignements sur le fonctionnement de tcpd dans la page de manuel `hosts_access(5)`.

Note : Comme vous pouvez le constater si vous comparez les lignes du fichier de configuration `/etc/inetd.conf` utilisant tcpd et celles qui ne l'utilisent pas, la liste des paramètres passés à tcpd par inetd est différente de celle utilisée pour

les démons lancés directement. En effet, elle ne comprend pas le nom de `tcpd` lui-même, alors que pour les démons, elle contient le nom du démon en premier paramètre. Cette différence provient du fait que le premier argument passé en ligne de commande est le nom du programme lui-même, sauf pour `tcpd`. En effet, `tcpd` suppose que ce paramètre contient le nom du démon dont il contrôle l'accès, et non son propre nom.

La sécurité basée sur `tcpd` suppose que les adresses IP source des paquets reçus sont effectivement les adresses IP des machines qui les ont envoyés. Malheureusement, cette hypothèse ne peut pas être vérifiée pour les adresses autres que les adresses des réseaux considérés comme sûrs (par exemple le réseau local), car le protocole IP n'inclut aucun mécanisme d'authentification. Par conséquent, n'importe qui peut tenter de communiquer avec votre machine au nom d'une autre machine (technique nommée « IP spoofing » en anglais), qu'il aura au préalable mis hors d'état de se manifester (par exemple par une attaque de type DoS, « Denial of Service » en anglais). `tcpd` tente par défaut de contrôler ce genre de pratique en vérifiant que le nom indiqué par la machine qui se connecte correspond bien à l'adresse IP qu'elle utilise. Ainsi, pour passer cette protection, il faut d'abord infecter un DNS. Cependant, ce genre d'attaque (dite d'interposition) reste courant, surtout sur les réseaux dont les DNS sont insuffisamment protégés (une étude récente a montré que c'était le cas de trois DNS sur quatre en France). Par conséquent, vous ne devez pas vous reposer uniquement sur `tcpd` et les techniques de pare-feu si vous devez créer un site réellement sûr. En particulier, vous devriez vous intéresser aux réseaux virtuels et au chiffrement des données, choses que l'on décrira dans la [Section 9.5.2.2](#). Le protocole `Ipv6` intégrera des mécanismes d'authentification et sera donc nettement plus sûr. Bien entendu, cela dépasse le cadre de ce document.

Le démon `tcpd` peut également être utilisé avec le démon `xinetd`. Toutefois, une telle utilisation est superflue, étant donné que `xinetd` permet de définir ses propres règles de contrôle d'accès.

IX-E-1-b-ii - Restrictions d'accès avec `xinetd`

Le super-démon `xinetd` intègre d'office un mécanisme de contrôle d'accès similaire à celui utilisé par `tcpd`. Grâce aux options `only_from` et `no_access`, il est possible de spécifier pour chaque service les machines qui sont autorisées à l'utiliser, ou inversement les machines qui doivent être explicitement rejetées.

Il faut toutefois prendre garde au fait que, comme pour `tcpd`, ne rien dire signifie accepter les connexions provenant de toutes les machines. La règle d'or est donc, encore une fois, de tout interdire par défaut et d'autoriser au compte-goutte les accès aux services ouverts. Pour cela, il suffit d'ajouter la ligne suivante :

```
only_from =
```

dans la section de définition des options par défaut de `xinetd`. Avec cette ligne, contrairement à ce qui se passe si rien n'est spécifié, les accès ne sont autorisés qu'à partir d'aucune machine. Autrement dit, ils sont toujours interdits.

Il faut ensuite redonner les droits sur les machines autorisées service par service. Ainsi, les connexions par `telnet` peuvent être autorisées sur le réseau local (supposé sûr) grâce à la ligne suivante :

```
only_from += 192.168.0.0/24 127.0.0.1
```

dans la section décrivant le service `telnet` de la configuration de `xinetd`.

On prendra garde également au fait que `xinetd` implémente quelques services internes permettant de l'administrer par le réseau. Il est évident que ces services doivent être désactivés. Ces services sont respectivement les services `servers`, `services` et `xadmin`. Leur désactivation se fait classiquement en ajoutant la ligne suivante :

```
disabled = servers services xadmin
```

dans la section `defaults` du fichier de configuration `/etc/xinetd.conf`.

IX-E-1-c - Contrôle des utilisateurs au niveau des services

Certains services ne peuvent pas être supprimés complètement, tout simplement parce que l'on peut en avoir besoin. Dans ce cas, il faut au moins s'assurer que l'accès à ces services n'est autorisé que pour les utilisateurs qui en ont réellement besoin. La règle est cette fois que l'utilisateur root ne doit pas avoir accès aux services réseau par une connexion non locale. En effet, un pirate attaque toujours par le réseau et, même s'il arrive à trouver le mot de passe d'un utilisateur normal, il lui restera encore pas mal de boulot pour trouver le mot de passe de l'utilisateur root (attention cependant, si vous êtes dans cette situation, il vaut mieux réagir très très vite).

Le service le plus sensible est bien entendu le service de login. Il est possible de fournir le service de login aux connexions extérieures, et il est évident que cela constitue déjà un point d'entrée pour qui peut trouver un mot de passe valide. Il est impératif ici d'interdire une telle connexion à l'utilisateur root. Il reste toutefois concevable que cet utilisateur puisse se connecter sur la console locale (c'est-à-dire, avec votre clavier et votre écran...). Il est donc nécessaire de choisir les terminaux à partir desquels l'utilisateur root pourra se connecter. Ces informations sont stockées dans le fichier de configuration `/etc/securetty`.

Le format de ce fichier est très simple, puisqu'il est constitué de la liste des terminaux à partir desquels l'utilisateur root peut se connecter, à raison d'un terminal par ligne. Un fichier de configuration `/etc/securetty` typique contient donc la liste des terminaux virtuels de la machine :

```
# Exemple de fichier /etc/securetty
tty1
tty2
tty3
tty4
tty5
tty6
```

Il est fortement déconseillé de placer dans ce fichier les autres terminaux (en particulier, les terminaux série `ttyS0` et suivants).

Un autre service sensible est le service de téléchargement de fichiers. Il est recommandé d'interdire aux utilisateurs privilégiés les transferts de fichiers par FTP. En effet, si une personne parvient à accéder au système de fichiers, il peut supprimer tous les mécanismes de sécurité. De la même manière que l'on a interdit à l'utilisateur root de se connecter sur les terminaux distants, il faut lui interdire (ainsi qu'aux autres comptes sensibles) les connexions par FTP.

Cette opération peut être réalisée en ajoutant le nom de chaque utilisateur sensible dans le fichier de configuration `/etc/ftpusers`. Ce fichier a la même structure que le fichier `securetty`, puisqu'il faut donner un nom d'utilisateur par ligne :

```
# Exemple de fichier /etc/ftpusers
root
uucp
mail
```

Bien entendu, la liste des utilisateurs privilégiés de votre système dépend de la distribution que vous avez installée. Le fichier `/etc/ftpusers` fourni avec cette distribution est donc, en général, approprié.

IX-E-2 - Chiffrement des communications

Les protocoles réseau de haut niveau ne garantissent en général pas la confidentialité des données qu'ils transmettent sur le réseau, ce qui signifie qu'elles peuvent être lues par tout le monde. Ils n'en garantissent pas non plus l'intégrité, chacun pouvant les modifier à volonté et fausser ainsi la communication. Ces défauts sont incurables, car ils sont en réalité inhérents au protocole IP lui-même. En effet, ce protocole a été conçu initialement pour assurer l'acheminement des informations, et les protocoles de haut niveau basés sur lui se contentent généralement de la garantie de livraison.

Note : La confidentialité des données est le fait que seules les personnes autorisées peuvent y accéder. L'intégrité des données est le fait que les données n'ont pas été altérées ou modifiées de manière non autorisée. Enfin, l'authenticité des données est le fait que les données sont certifiées comme provenant bien de celui qui prétend en être l'émetteur.

Généralement, le moyen le plus sûr pour garantir la confidentialité des données est de les chiffrer de telle sorte que seuls les destinataires autorisés puissent les déchiffrer et donc y accéder. C'est pour cette raison que la cryptographie, ou science du chiffrement est souvent utilisée pour résoudre les problèmes de sécurité informatique. Mais la cryptographie fournit également les techniques nécessaires pour assurer l'intégrité des données, généralement en calculant une empreinte ou condensé cryptographique des données. En effet, les algorithmes de calcul des empreintes sont conçus pour fournir des résultats très différents même si les données ne sont modifiées que très légèrement. Ainsi, si une personne mal intentionnée modifie un tant soit peu les données, le recalcul de l'empreinte associée indiquera tout de suite la supercherie. Enfin, l'authenticité des données peut être assurée en faisant en sorte que leur émetteur puisse prouver son identité. Cela se fait généralement en exigeant qu'il puisse répondre à une question dont on sait que lui seul connaît la réponse. Toute la difficulté est alors de trouver un système d'authentification où celui qui pose la question ne doit lui-même pas en connaître la réponse (faute de quoi celui qui pose la question pourrait se faire passer pour celui qui doit répondre !).

Note : Il est évident que pour garantir la confidentialité et l'authenticité des données, il faut garantir l'authenticité des interlocuteurs. En effet, sans cela, une tierce personne pourrait s'intercaler entre deux interlocuteurs et faire croire à chacun qu'il est l'autre. Même dans un canal de communication chiffré, cette personne serait en mesure de déchiffrer les messages dans les deux sens, voire même de les modifier et d'en recalculer les empreintes. Cette technique d'interposition s'appelle l'attaque du « man in the middle » (« l'homme du milieu » en Français)

Il est donc nécessaire, du fait des faiblesses des protocoles réseau standards, de recourir à des solutions cryptographiques pour fournir les services de confidentialité, d'authenticité et d'intégrité des données. Parmi ces solutions, les plus utilisées sont sans doute le protocole SSH (abréviation de l'anglais « Secure SHell ») et l'extension IPsec au protocole IP. Cette extension a initialement été développée pour le successeur du protocole IP, à savoir le protocole IPv6, mais a également été adaptée pour fonctionner avec l'implémentation actuelle d'IP.

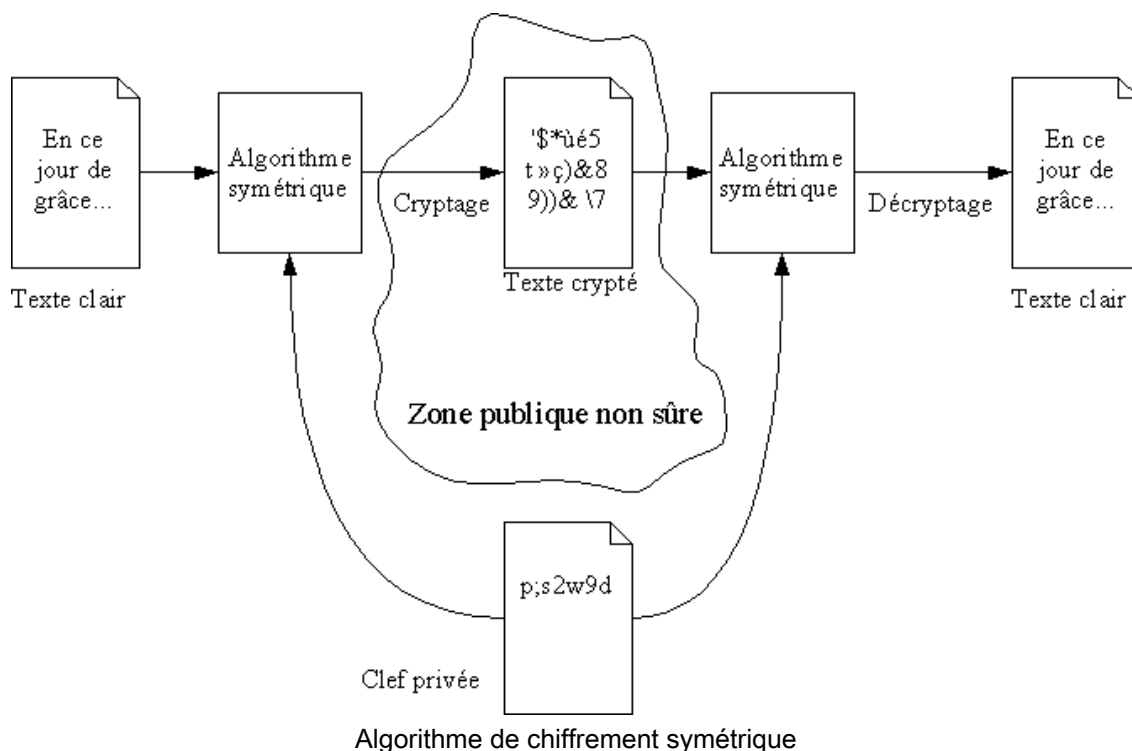
IX-E-2-a - Principes de base de cryptographie

Tout d'abord, commençons par quelques définitions, beaucoup d'articles disponibles actuellement utilisant un vocabulaire imprécis. Le chiffrement (également appelé à tort cryptage), est l'opération qui consiste à transformer une information en clair en une information codée afin d'éviter qu'elle ne puisse être utilisée par une personne non autorisée. Le déchiffrement est l'opération inverse du chiffrement, qui permet donc de retrouver l'information initiale à partir de l'information codée. Le décryptage est l'opération qui consiste à déchiffrer une information sans y être autorisé. La cryptographie est la science qui se charge d'étudier la manière de chiffrer les informations. La cryptoanalyse est l'étude des moyens de décryptage. Cette science va de pair avec la cryptographie, puisqu'elle permet de déterminer la fiabilité des techniques mises en œuvre pour le chiffrement. Enfin, un pirate (également appelé « cracker ») est une personne désirent s'introduire de manière illicite dans un système, ou désirent détourner une application de son contexte d'utilisation normal. Il ne faut pas confondre les crackers avec les hackers, qui sont des personnes extrêmement compétentes en informatique et qui ne font rien de mal (par exemple, les développeurs du noyau Linux sont souvent des hackers).

Il existe un grand nombre de techniques de cryptographie, qui sont plus ou moins efficaces. La première technique, qui est aussi la plus simple et la moins sûre, est celle qui consiste à utiliser un algorithme de chiffrement réversible (ou une paire d'algorithmes ayant un effet inverse l'un de l'autre) pour chiffrer une information. Dans ce cas, la sécurité des données est assujettie au maintien au secret de l'algorithme utilisé. Il est évident que ce type de technique n'est pas utilisable avec les logiciels dont les sources sont diffusées (comme Linux), puisque l'algorithme est alors connu de tout le monde. C'est également la technique la moins sûre, parce qu'il est en général très facile de déterminer cet algorithme. Un exemple d'un tel algorithme est le codage classique qui consiste à décaler toutes les lettres de l'alphabet (A devient B, B devient C, etc.).

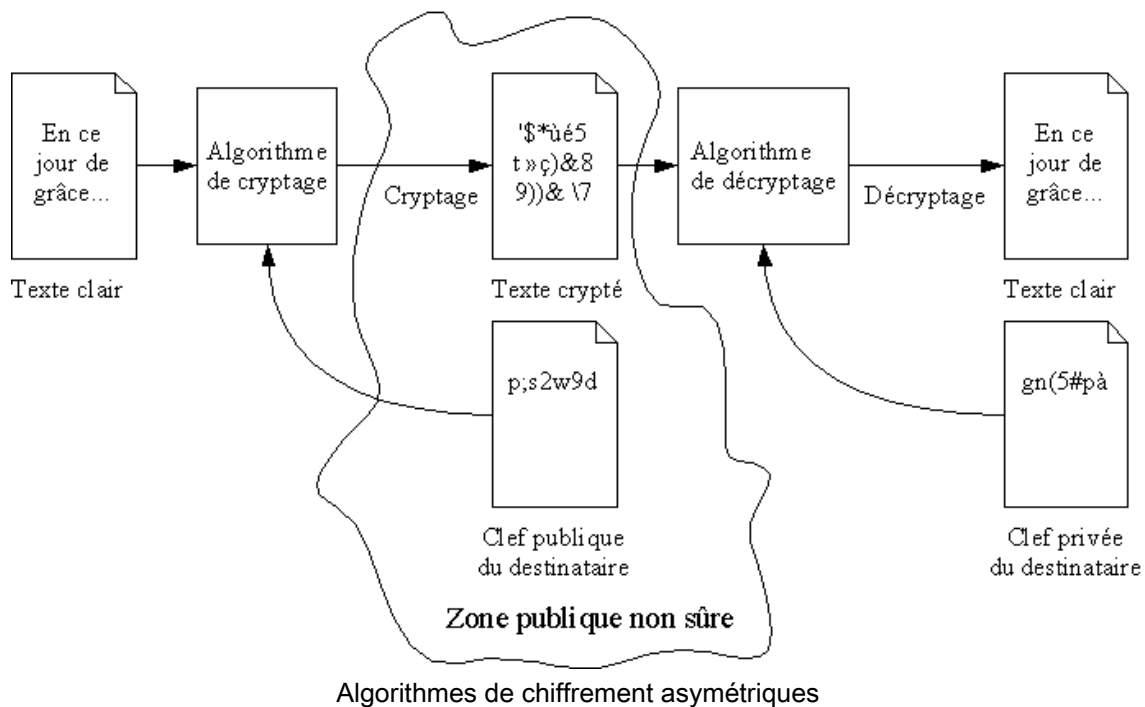
Une autre technique se base sur un algorithme connu de tout le monde, mais dont l'effet est paramétré par une clef tenue secrète. Le déchiffrement des informations ne peut se faire que si l'on connaît cette clef. Les algorithmes de ce type sont souvent qualifiés de « symétriques », en raison du fait que la clef utilisée pour le déchiffrement

est la même que celle utilisée pour le chiffrement. Cette technique n'est satisfaisante que dans la mesure où les personnes qui communiquent peuvent s'échanger cette clef de manière sûre. La clef utilisée ne devant être connue que des personnes autorisées à participer à la communication, cet échange doit se faire soit de visu, soit dans un canal déjà chiffré (mais dans ce cas, il faut échanger la clef de déchiffrement du canal, et on se retrouve devant le problème de l'œuf et de la poule), soit à l'aide d'un protocole d'échange de clef sûr. Les algorithmes de ce type sont également connus sous le nom d'algorithmes à clef secrète en raison de cette particularité. Il existe un grand nombre d'algorithmes à clef secrète, les plus connus étant DES, AES et IDEA.



La technique la plus sûre actuellement, et aussi la plus rusée, est d'utiliser un algorithme de chiffrement asymétrique. À la différence des algorithmes symétriques, les algorithmes asymétriques utilisent deux clefs différentes et duales pour le chiffrement et le déchiffrement. Ces algorithmes se basent sur des fonctions mathématiques dont la fonction inverse est extrêmement difficile à déterminer sans une information également difficile à calculer. Cette information fait office de clef permettant de retrouver l'information en clair. Toute la ruse ici est que la clef servant au déchiffrement peut ne plus être échangée du tout, ce qui supprime la nécessité de réaliser un échange de clefs généralement risqué. En fait, seule la clef de chiffrement doit (et peut sans crainte) être publiée. C'est en raison de cette propriété que les algorithmes asymétriques sont également appelés algorithmes à clef publique.

Le principe des algorithmes asymétriques est le suivant. Une clef publique, utilisée pour chiffrer les informations, est fournie à tout le monde, et seul celui qui dispose de la clef privée associée à cette clef publique peut déchiffrer le texte. Toute personne qui désire communiquer avec le propriétaire de la clef publique utilise celle-ci pour chiffrer les informations qu'il doit lui transférer. Ainsi, du fait que seul le destinataire dispose de la clef de déchiffrement (sa clef privée), il est le seul à pouvoir récupérer ces informations. Il est donc possible d'échanger des informations de manière sûre avec tous les intervenants en utilisant simplement leurs clefs publiques respectives. Aucun échange d'information sensible n'est effectué en clair. Les algorithmes à clef publique les plus connus sont RSA et DSA.



La gestion des clefs des algorithmes à clef publique est techniquement laborieuse, de même que les calculs mis en œuvre par les algorithmes utilisés. Il est donc courant d'amorcer une communication avec un algorithme à clef publique, comme RSA par exemple, puis de l'utiliser pour échanger une clef secrète permettant de poursuivre la communication avec un algorithme symétrique. Il existe également des algorithmes spécialisés dans les échanges de clefs dans un canal non sûr, par exemple l'algorithme Diffie Hellman.

Enfin, il existe des algorithmes de chiffement à sens unique qui ne permettent pas de retrouver l'information initiale, mais qui sont utilisés pour calculer une empreinte des informations à transférer. Cela permet d'identifier l'information de manière unique. Ces algorithmes sont donc souvent utilisés pour garantir l'intégrité des données diffusées dans un environnement peu sûr. Pour cela, celui qui réalise la diffusion chiffre un condensé des informations diffusées avec sa clef privée. Tout le monde peut donc déchiffrer ce condensé avec la clef publique de cette personne, et vérifier que le message qu'ils reçoivent a bien la même empreinte. Chacun a donc la certitude que le message provient bien de son auteur, car toute modification du message changerait son empreinte d'une part, et il est impossible de chiffrer cette nouvelle empreinte sans connaître la clef privée de l'expéditeur. Les algorithmes de calcul d'empreinte les plus connus sont MD5 et SHA.

Note : Notez que les algorithmes de chiffement à clef publique nécessitent toujours que l'on soit sûr que l'on utilise la bonne clef publique pour s'adresser à une personne. Or, rien ne nous garantit que la clef publique d'une personne diffusée sur le réseau provienne bien de cette personne ! Il est donc toujours nécessaire d'échanger les clefs publiques de main à main, avec vérification de l'identité de la personne avec qui l'échange se fait. Il est également possible de recourir à une tierce personne digne de confiance et dont on connaît déjà la clef publique pour garantir que la clef que l'on reçoit est bien celle de la personne qui prétend en être le propriétaire. On peut ainsi construire ce qu'on appelle un « réseau de confiance », dans lequel chacun sait que quelqu'un en qui il a confiance, directement ou indirectement, a authentifié ses interlocuteurs.

Bien entendu, tout cela est contraignant. Mais, contrairement aux algorithmes de chiffement à clef secrète, il n'est pas grave de perdre les clefs échangées avec les algorithmes asymétriques, puisque ces clefs sont destinées à être publiques...

IX-E-2-b - Utilisation de SSH

SSH est en réalité un protocole de communication réseau chiffré couplé d'une suite d'outils permettant de l'utiliser. SSH a été initialement développé par une société commerciale, qui diffusait les outils en Open Source et gratuitement pour les particuliers. Ces outils n'étaient toutefois absolument pas libres. Une implémentation libre a donc été

développée pour le système FreeBSD, puis portée pour les autres systèmes d'exploitation, dont Linux. Cette implémentation est celle qui est utilisée par la plupart des distributions actuellement, il s'agit de « OpenSSH ». Nous ne décrivons ici que l'utilisation d'OpenSSH.

Pour garantir la confidentialité des données, SSH utilise des techniques de cryptographie avancées. Ces techniques permettent de s'assurer que personne ne peut lire les informations qui circulent sur le réseau d'une part, et de garantir l'authentification des machines auxquelles on se connecte d'autre part. De plus, SSH permet de rediriger les protocoles réseau non chiffrés dans son propre canal de communication, et résout donc la plupart des problèmes de sécurité. SSH a donc pour but premier de remplacer les applications non sûres permettant de se connecter à distance sur une machine ou d'y exécuter des commandes. Les applications ainsi rendues obsolètes sont, entre autres, telnet, rlogin et rsh. Enfin, SSH est également utilisable pour chiffrer automatiquement les connexions X11, ce qui permet d'utiliser l'environnement graphique X Window à distance en toute sécurité.

Note : Officiellement, il est interdit d'utiliser en France des techniques de chiffrement aussi poussées que celles mises en œuvre par SSH, sauf à demander une dérogation. Il y a de fortes pressions pour permettre la légalisation de ces techniques, mais elles n'ont pas encore abouti. Plus précisément, le gouvernement a pris position pour une libéralisation de l'usage de la cryptographie, mais la loi n'a pas encore été votée. Les sinistres événements du 11 novembre 2001 risquent fort de retarder encore, si ce n'est de compromettre, cette libéralisation. En effet, l'argument principal des opposants à cette légalisation est qu'elle permettrait aux truands de chiffrer leurs communications. L'argument est bien entendu fallacieux : comme s'ils attendaient d'en avoir le droit pour le faire...

En fait, il existe une implémentation gratuite (mais non libre) de SSH qui a été déclarée et qui peut donc être utilisée en France. Il s'agit de la version de Bernard Perrot, nommée « SSF », que l'on pourra trouver à l'adresse <http://perso.univ-rennes1.fr/bernard.perrot/SSF/index.html>. Si vous désirez absolument être sans reproche vis-à-vis de la loi actuellement en vigueur en France, vous devriez remplacer la version d'OpenSSH fournie avec votre distribution (dont l'usage est donc illégal) par SSF. Sachez cependant que SSF restreint la taille des clefs de sessions à 128 bits seulement d'une part (condition sine qua non pour qu'on puisse l'utiliser), qu'il n'est pas utilisable commercialement, et qu'il est susceptible de ne pas être à jour en ce qui concerne les correctifs de bogues et les failles de sécurité que l'on a pu trouver récemment dans SSH ou OpenSSH. La sécurité apportée par SSF est donc très relative, mais c'est mieux que rien du tout.

Il existe deux versions du protocole réseau SSH, qui ont toutes deux été adoptées en tant que standards. La première version souffrait d'un trou de sécurité majeur, qui depuis a largement pu être utilisé. Il ne faut donc surtout pas utiliser cette version et se rabattre systématiquement sur le protocole SSHv2. Ce protocole permet de garantir l'authentification des machines et des utilisateurs et est nettement plus fiable. Aucune attaque n'a permis de le casser à ce jour, sachez toutefois qu'il existe encore des points douteux dans ce protocole qui permettraient, en théorie, de parvenir à s'immiscer dans une communication ou d'en décrypter une partie.

Sachez également que les différentes implémentations de SSH, aussi bien libres que commerciales, sont soumises au même régime que les autres programmes : elles peuvent avoir des bogues et des trous de sécurité qu'un attaquant peut utiliser. De nombreuses attaques ont été découvertes ces derniers temps, aussi est-il recommandé de s'assurer que l'on dispose de la dernière version d'OpenSSH. La dernière version stable est la 3.7.1p1 (les versions antérieures doivent être évitées à tout prix). Cela dit, mener une attaque contre SSH relève le niveau de difficulté nettement au-dessus de ce qui est nécessaire pour pénétrer un système qui ne l'utilise pas.

IX-E-2-b-i - Principes de base de l'authentification SSH

La grande difficulté dans les mécanismes d'authentification est d'être capable de prouver son identité sans fournir suffisamment d'information pour que quelqu'un d'autre puisse se faire passer pour soi par la suite. La technique du mot de passe utilisée par les systèmes Unix est absolument insuffisante pour des communications réseau, car elle requiert justement de fournir ce mot de passe. Même si les mots de passe ne sont pas stockés en clair sur la machine (ce qui est la moindre des choses), il est possible de se faire passer pour une tierce personne. En effet, les mécanismes d'authentification comparent souvent une empreinte du mot de passe avec l'empreinte stockée dans les fichiers de définition des mots de passe, mais rien n'empêche un intrus de capter cette empreinte et de l'utiliser directement pour s'authentifier si l'on utilise la même technique en réseau. Il n'a même pas besoin de connaître le mot de passe en clair (cette technique est connue sous le nom d'attaque par « rejeu », car l'attaquant se contente dans

ce cas de rejouer la séquence d'authentification). Une solution contre le rejeu est de demander à la personne que l'on cherche à authentifier de chiffrer un message aléatoire avec son mot de passe ou son empreinte, et de vérifier que le résultat qu'il renvoie est bien celui attendu, en chiffrant le même message de son côté. Cette technique est toutefois toujours vulnérable à l'attaque de l'homme du milieu, et elle souffre de l'énorme défaut que l'on doit stocker l'empreinte de son mot de passe sur tous les ordinateurs auxquels on désire se connecter. Cela multiplie d'autant les points de faiblesse : si un pirate s'introduit sur une seule machine, il obtient toutes les empreintes des mots de passe et peut donc se faire passer pour tout le monde !

SSH ne transmet donc ni les mots de passe ni leurs condensés, en clair sur le réseau. Il établit d'abord un canal de communication chiffré à l'aide d'un des algorithmes d'échange de clef (la technique utilisée dépend de la version utilisée du protocole SSH). Une fois le canal chiffré établi, la machine serveur est authentifiée à l'aide de sa clef publique. Pour que cela fonctionne, il faut bien entendu que la clef publique de la machine serveur soit connue du client et ait été authentifiée comme étant bien celle de ce serveur. Inversement, le client peut être authentifié par le serveur. SSH fournit plusieurs possibilités pour cela, les deux plus courantes étant l'authentification par mot de passe et l'authentification par un algorithme à clef publique.

L'authentification par mot de passe est strictement la même que celle utilisée pour le login Unix classique, à la différence près que les informations du mot de passe sont transmises dans le canal chiffré, vers un serveur authentifié. La sécurité de ce mot de passe est donc garantie, et l'attaque de l'homme du milieu ne fonctionne plus. Ce type d'authentification permet donc d'utiliser SSH exactement comme les commandes classiques rsh ou telnet. L'utilisation de SSH est donc complètement transparente pour les utilisateurs.

Cela dit, il est possible de faciliter encore plus l'authentification des clients tout en augmentant le niveau de sécurité, en utilisant également l'authentification par clef publique pour les utilisateurs. Dans ce mode d'authentification, chaque utilisateur dispose également d'un couple de clefs privée et publique. Les clefs publiques de ces utilisateurs sont diffusées dans les comptes des utilisateurs au niveau du serveur. Ainsi, il n'est plus nécessaire de fournir un mot de passe, car le serveur peut authentifier l'utilisateur du seul fait qu'il est le seul à connaître sa clef privée. La connexion est donc automatique (une fois le serveur correctement configuré, bien entendu) et aucun échange d'information sensible comme le mot de passe n'est réalisé (même si le fait que cet échange était réalisé dans un canal chiffré par SSH était déjà une bonne garantie).

Les clefs privées et publiques sont bien entendu stockées sur disque, dans les fichiers de configuration des serveurs et des clients. Il va de soi que si les clefs publiques peuvent être lues, il est impératif que les clefs privées ne soient lisibles que par leurs propriétaires. Les droits d'accès sur les fichiers de configuration doivent donc être fixés de manière correcte pour garantir la sécurité du système. En fait, le point faible du mécanisme d'authentification des clients est toujours au niveau des utilisateurs, qui peuvent perdre leur clef privée ou la communiquer à d'autres sans même savoir qu'ils font une erreur fondamentale (de la même manière qu'ils peuvent choisir un mot de passe trop facile à trouver ou simplement le communiquer à une tierce personne). C'est pour cela qu'il est possible de chiffrer les clefs privées avec un algorithme de chiffrement symétrique. L'utilisateur doit, dans ce cas, fournir une phrase clef à chaque fois qu'il désire réaliser une connexion SSH.

IX-E-2-b-ii - Compilation et installation d'OpentSSH

Comme nous l'avons dit plus haut, il est recommandé d'utiliser la dernière version d'OpenSSH, à savoir la version 3.7.1.p1. Il est probable que votre distribution fournisse une mise à jour sur son site, et il est recommandé d'utiliser ces mises à jour.

Si, toutefois, vous désirez effectuer l'installation à partir des fichiers sources, vous pourrez procéder comme suit. Il vous faudra récupérer l'archive des fichiers sources en premier lieu sur le site [OpenSSH](#). Pour Linux, il faut prendre l'archive 3.7.1p1, le 'p' signifiant ici qu'il s'agit d'un portage des sources pour Linux. Une fois cela fait, vous pourrez exécuter la commande suivante dans le répertoire des sources :

```
./configure --prefix=/usr --sysconfdir=/etc/ssh --with-tcp-wrappers --with-pam
```

Cette commande permet de configurer OpenSSH pour qu'il remplace la version existante dans votre système. Elle indique également que le support du démon tcpd doit être activé, ainsi que celui des modules d'authentification PAM

(options « `--with-tcp-wrappers` » et « `--with-pam` »). Bien entendu, si votre distribution n'utilise pas ces fonctionnalités, vous devez supprimer ces options.

La compilation et l'installation en soi ne posent pas de problème particulier. Pour les réaliser, il vous suffira d'exécuter les deux commandes suivantes :

```
make
make install
```

Vous constaterez que le programme d'installation génère automatiquement des fichiers de clefs publiques et privées pour votre machine. Nous verrons dans les sections suivantes comment ces clefs peuvent être utilisées.

Note : Vous devrez créer un compte utilisateur « `sshd` » non privilégié avant d'installer SSH. Ce compte est en effet utilisé par le démon `sshd` pour réaliser les communications réseau une fois la phase d'authentification réalisée. Cela permet de garantir que, même en cas d'exploitation d'une éventuelle faille de sécurité du démon `sshd`, l'accès à la machine sera extrêmement limité.

IX-E-2-b-iii - Configuration d'OpenSSH côté serveur

OpenSSH utilise un démon nommé `sshd` pour prendre en charge les requêtes de connexion de la part des clients. Ce démon est à l'écoute des demandes de connexion et effectue les opérations d'authentification du serveur auprès des clients. Il vérifie également les droits du client avant de le laisser utiliser la connexion chiffrée.

Le démon `sshd` utilise un fichier de configuration général `/etc/ssh/sshd_config` et plusieurs fichiers contenant les clefs publiques et privées de la machine pour les différents protocoles cryptographiques utilisés par les clients. Les protocoles disponibles sont le protocole RSA pour la version 1 du protocole SSH, et les protocoles RSA et DSA pour la version 2. Comme il existe deux fichiers pour les clefs du serveur (un premier fichier lisible par tous les utilisateurs, pour la clef publique, et un fichier lisible uniquement par l'utilisateur `root`, pour la clef privée), il peut exister au total six fichiers de clefs. Ces fichiers sont nommés respectivement `/etc/ssh_host_key` et `/etc/ssh_host_key.pub` pour les clefs privées et publiques du protocole de version 1, et `/etc/ssh_host_rsa_key`, `/etc/ssh_host_rsa_key.pub`, `/etc/ssh_host_dsa_key` et `/etc/ssh_host_dsa_key.pub` respectivement pour les algorithmes RSA et DSA du protocole SSH de version 2.

Normalement, les fichiers de clefs du serveur ont été créés automatiquement lors de l'installation de SSH. Cependant, si vous installez SSH à partir d'une archive de distribution, ils peuvent ne pas être présents, et vous devrez générer ces clefs vous-même. Pour cela, il faut utiliser la commande `ssh-keygen`, dont la syntaxe est la suivante :

```
ssh-keygen [-t type]
```

où `type` est le type de clef à générer pour le protocole de version 2. Les types disponibles sont `rsa` et `dsa`. Pour le protocole de version 1, il ne faut pas utiliser l'option `-t`, les clefs générées étant forcément des clefs RSA.

Lorsque l'on invoque la commande `ssh-keygen`, celui-ci demande le chemin du fichier dans lequel la clef privée doit être écrite. Le fichier contenant la clef publique aura le même nom, mais portera l'extension `.pub`. Vous devez donc saisir le chemin du fichier correspondant au type de clef généré. `ssh-keygen` demande ensuite le mot de passe à utiliser pour le chiffrement de la clef privée. Dans le cas d'un serveur, il ne faut pas utiliser de mot de passe, car dans ce cas le démon `sshd` ne pourrait pas accéder à la clef privée de la machine. Il faut donc, dans ce cas, valider simplement sans rien taper.

Une fois les fichiers de clefs du serveur générés, vous pouvez vous intéresser au fichier de configuration `sshd_config`. Ce fichier de configuration contient un certain nombre d'options qui indiquent les paramètres que le démon doit utiliser. Vous trouverez ci-dessous un exemple de fichier de configuration :

```
# Paramètres de connexion :
Port 22
ListenAddress 0.0.0.0
```

```
KeepAlive yes

# Protocoles utilisables :
Protocol 2

# Chemin sur les fichiers de clefs :
HostKey /etc/ssh_host_rsa_key
HostKey /etc/ssh_host_dsa_key

# Paramètres des messages de traces :
SyslogFacility AUTH
LogLevel INFO

# On définit les modes d'authentification utilisables :

# Authentification par clef publique :
RSAAuthentication no
PubkeyAuthentication yes

IgnoreRhosts yes
RhostsRSAAuthentication no
HostbasedAuthentication no

# Authentification par mot de passe :
PasswordAuthentication no
PermitEmptyPasswords no

# Options générales :

# L'utilisateur root ne doit pouvoir se loguer qu'en local :
PermitRootLogin no

# Utilise le compte non privilégié sshd
# pour les communications réseau :
UsePrivilegeSeparation yes

# On n'autorise la connexion que pour les utilisateurs dont
# les fichiers de configuration sont correctement protégés :
StrictModes yes

# Encapsulation automatique du protocole X :
X11Forwarding yes
X11DisplayOffset 10

# Affichage des informations habituelles au login :
PrintMotd no
PrintLastLog yes
```

Comme vous pouvez le constater, un grand nombre d'options sont disponibles. Les options `Port` et `ListenAddress` permettent de définir l'adresse IP sur laquelle le démon `sshd` se mettra en attente de demandes de connexion, ainsi que le port TCP utilisé pour cela. L'adresse `0.0.0.0` signifie ici qu'il doit se mettre en attente sur toutes les adresses de la machine. L'option `KeepAlive` permet de demander au démon de vérifier régulièrement que la liaison est toujours valide, afin de fermer les connexions automatiquement en cas d'arrêt du client. L'option `Protocol` permet de spécifier les versions utilisables du protocole SSH. La version 1 étant obsolète, on spécifiera systématiquement la valeur 2 ici. L'option `HostKey` permet, comme son nom l'indique, de donner les noms des fichiers contenant les clefs pour les différents protocoles d'authentification. Les options `SyslogFacility` et `LogLevel` permettent d'indiquer le type et le niveau des messages de traces générés par le démon `sshd` dans le fichier de traces du système.

Viennent ensuite les options spécifiques aux différents modes d'authentification. Les options `RSAAuthentication` et `PubkeyAuthentication` permettent d'activer et de désactiver l'authentification par clef publique des clients, respectivement pour les protocoles SSH de version 1 et 2. Les options `IgnoreRhosts`, `RhostsRSAAuthentication` et `HostbasedAuthentication` permettent de contrôler un protocole d'authentification basé sur l'identité des machines. Ce type d'authentification n'a pas été présenté, car il est extrêmement peu sûr, et il faut impérativement désactiver ces fonctionnalités. Enfin, les options `PasswordAuthentication` et `PermitEmptyPasswords` permettent d'activer ou de désactiver l'authentification par mot de passe. Vous êtes libre d'utiliser ou non ces fonctionnalités.

Le fichier de configuration peut également contenir des options complémentaires. Les plus importantes pour la sécurité sont sans doute `PermitRootLogin` et `StrictModes`, qui permettent d'interdire la connexion par le réseau à l'utilisateur `root` d'une part, et d'interdire la connexion aux utilisateurs qui n'ont pas bien fixé les droits d'accès sur leurs fichiers de configuration personnels dans leur compte local (il est donc impossible de garantir que ces fichiers n'ont pas été trafiqués par d'autres utilisateurs pour accéder à leur compte). L'option `UsePrivilegeSeparation` permet de demander au démon `sshd` de n'effectuer les communications réseau que dans le compte spécial `sshd`, compte auquel on n'affectera que le minimum de droits. Cette technique permet de protéger le système contre l'exploitation d'une éventuelle faille de sécurité dans le démon `sshd`. L'option `X11Forwarding` permet d'activer l'encapsulation automatique du protocole X et peut être relativement pratique. Le numéro de display auquel les utilisateurs devront connecter leurs applications sera alors décalé du nombre indiqué par l'option `X11DisplayOffset`. Cette option est utile pour éviter les conflits avec les serveurs X locaux. Vous trouverez de plus amples informations sur le protocole X11 et l'environnement graphique dans le [Chapitre 10](#). Enfin, les options `PrintMotd` et `PrintLastLog` permettent de spécifier les informations affichées par le démon `sshd` lorsque la connexion des clients est acceptée. Il existe bien d'autres options, vous pourrez obtenir de plus amples renseignements dans la page de manuel du démon `sshd`.

IX-E-2-b-iv - Utilisation d'OpenSSH côté client

Les clients peuvent se connecter à un serveur SSH à l'aide de la commande `ssh`. Cette commande s'utilise exactement de la même manière que les commandes `rsh` ou `rlogin`. La syntaxe de la commande `ssh` est la suivante :

```
ssh [-l nom] machine [commande]
```

ou :

```
ssh nom@machine [commande]
```

où `nom` est ici le nom d'utilisateur à utiliser pour la connexion, et `machine` la machine distante. Si aucun nom d'utilisateur n'est spécifié, le nom de l'utilisateur sur la machine local est pris par défaut. Il est possible d'exécuter une commande sur la machine distante en la spécifiant après le nom de la machine. Si aucune commande n'est spécifiée, un shell interactif est lancé.

Selon le mode d'authentification du client choisi par le serveur, il peut être nécessaire ou non de définir des clefs pour le client. Tout comme pour les clefs publiques et privées d'un serveur, la génération de ces clefs se fait à l'aide de la commande `ssh-keygen`. Cependant, contrairement à ce qui se passait pour le serveur, les chemins par défaut proposés pour stocker les clefs sont ici corrects, et il est vivement recommandé de saisir un mot de passe pour chiffrer la clef privée. Les fichiers de clefs générés par `ssh-keygen` sont les fichiers `identity` et `identity.pub` pour le protocole d'authentification RSA de la version 1 du protocole SSH, et les fichiers `id_rsa`, `id_rsa.pub`, `id_dsa` et `id_dsa.pub` respectivement pour les protocoles RSA et DSA de la version 2 du protocole SSH. Tous ces fichiers sont stockés dans le répertoire `.ssh/` du répertoire personnel de l'utilisateur qui les génère. Il va de soi que les fichiers des clefs privées ne doivent pas être lisibles pour les autres utilisateurs, et que les fichiers des clefs publiques ne doivent pas pouvoir être écrits si l'on veut éviter qu'un usurpateur prenne notre place sur notre propre machine.

Afin de bénéficier de l'authentification des serveurs auxquels on se connecte, il faut placer les clefs publiques de ces derniers dans le fichier `.ssh/known_hosts`. Si, lors d'une connexion à un serveur, la clef publique de celui-ci n'est pas connue, `ssh` vous le signalera. Il indiquera un message d'erreur indiquant que le serveur auquel on se connecte n'est pas forcément celui qu'il prétend être, et affiche un condensé de sa clef publique. Vous devrez vous assurer que ce condensé est bien celui de la clef publique du serveur auquel vous vous connectez, et ce par un moyen sûr (déplacement physique et connexion en local par exemple). Si vous acceptez cette clef, `ssh` la placera pour vous dans votre fichier `known_hosts`, ce qui vous permettra de vous connecter par la suite en toute quiétude. Attention, le fichier `known_hosts` ne doit pas être accessible en écriture aux autres utilisateurs, car dans ce cas un pirate pourrait capter toutes vos communications !

Si le mode d'authentification utilisé est le mot de passe, `ssh` vous demandera de saisir le mot de passe du compte distant. Dans ce cas, `ssh` se comporte exactement comme `rsh`, à ceci près que votre mot de passe ne circulera pas en clair sur le réseau (quel progrès !).

Si, en revanche, le mode d'authentification utilisé est une authentification par clef publique, alors vous devrez avoir défini vos clefs privées et publiques. Pour que le démon sshd de la machine distante accepte la connexion, vous devrez placer vos clefs publiques dans le fichier `authorized_keys` du répertoire `/ssh/` du répertoire du compte distant. Cela permettra au démon sshd de s'assurer que c'est bien vous qui vous connectez. Attention, le fichier `authorized_keys` ne doit pas être accessible en écriture, faute de quoi un autre utilisateur de la machine distante pourrait se connecter en votre nom. Il est également impératif de réaliser l'écriture soi-même dans ce fichier par un moyen sûr (déplacement physique). En aucun cas la communication de votre clef publique à l'administrateur distant ne pourra être considérée comme un moyen sûr pour écrire cette clef : en effet, un pirate pourrait tout aussi bien lui demander d'écrire sa propre clef dans votre fichier !

Vous constaterez qu'un certain nombre de précautions doivent être prises pour les manipulations de clefs. Comme il l'a déjà été indiqué plus haut, le point faible est bel et bien l'authentification des clients. Cela concerne en premier lieu le client bien entendu, dont le compte peut être piraté, mais c'est aussi un trou de sécurité énorme pour le serveur, car une fois le loup dans la bergerie, il peut faire beaucoup de dégâts et chercher à accroître ses droits (jusqu'à devenir root) par d'autres techniques.

IX-E-2-b-v - Création d'un tunnel SSH

L'un des principaux avantages de SSH est qu'il est possible de l'utiliser pour encapsuler n'importe quel protocole réseau TCP dans le canal de communication d'une connexion sécurisée. Un tel canal est communément appelé un « tunnel », en raison du fait qu'il est utilisé par les informations sensibles pour traverser un réseau non sûr. Grâce aux tunnels SSH, il est possible d'utiliser tous les protocoles classiques sans avoir à s'inquiéter de la confidentialité des informations transmises. SSH permet même de réaliser cette opération automatiquement pour le protocole X du système de fenêtrage X Window, ce qui est très commode si l'on désire afficher localement les fenêtres graphiques des programmes lancés sur le serveur.

Encapsuler le protocole X dans SSH est donc extrêmement aisé. En effet, il suffit de donner la valeur « yes » à l'option `X11Forwarding` dans le fichier de configuration du serveur `/etc/sshd_config` et dans le fichier de configuration des clients (les valeurs par défaut pour tous les utilisateurs peuvent être définies dans le fichier de configuration `/etc/ssh_config` et être redéfinies par chaque utilisateur dans le fichier `config` de leur répertoire `.ssh/`). Si ces options sont activées, SSH se charge de fixer la variable d'environnement `DISPLAY` de la machine distante à un serveur X virtuel de cette machine. Les programmes X se connectent alors à ce serveur, et celui-ci effectue automatiquement le transfert des communications vers le serveur X du client. Il ne faut donc pas, quand on utilise SSH, modifier manuellement la variable d'environnement `DISPLAY`. Vous trouverez plus d'information sur X Window dans le [Chapitre 10](#).

Pour les autres protocoles, les opérations ne sont pas réalisées automatiquement par SSH. Il faut donc établir un réseau privé virtuel manuellement. Le principe est le suivant : les programmes clients sont paramétrés pour se connecter sur un port local différent de celui utilisé habituellement pour leur protocole. Ainsi, ils ne s'adressent pas directement au serveur distant, mais plutôt à un proxy local établi par SSH pour l'occasion. Ce proxy établit un canal de communication chiffré avec le démon sshd de la machine distante. Toutes les informations du protocole encapsulé passent alors par le canal chiffré, et le démon sshd de la machine distante retransmet ces informations au service distant auquel le client devait se connecter.

Tout ce mécanisme ne fonctionne que parce qu'il est possible de rediriger les clients sur un port réservé par SSH sur leur machine locale. Cette redirection peut être initiée aussi bien du côté serveur que du côté client, lors de la connexion. Du côté du client, la redirection de port peut être réalisée avec la commande suivante :

```
ssh -Llocal:serveur:port machine [commande]
```

où `local` est le port de la machine locale auquel il faudra se connecter pour accéder au service distant, fonctionnant sur le port `port` de la machine serveur. Cette commande ouvre une session SSH sur la machine `machine` (qui, normalement, est la même que `serveur`), et réalise la redirection du flux destiné au port `local` dans le canal de cette session.

Il faut bien comprendre que cette redirection se fait en plus de la connexion SSH, et que la syntaxe donnée ci-dessus ouvre une connexion sur la machine distante (ou exécute la commande spécifiée en paramètre, s'il en est spécifié une). Si l'on désire simplement établir un tunnel, on peut refermer la session juste après avoir lancé le programme qui doit utiliser ce tunnel. En effet, SSH maintient le tunnel ouvert tant qu'un programme détient une connexion sur le port redirigé. On peut aussi exécuter une commande `sleep` sur un délai arbitraire, qui permettra de se connecter sur le port redirigé et qui se terminera automatiquement.

Du côté serveur, la syntaxe est similaire. Cette fois, la commande permet d'indiquer le port que le client devra utiliser pour accéder à un service non sécurisé du serveur. La syntaxe à utiliser est la suivante :

```
ssh -Rport:client:local machine [commande]
```

où `port` est le port que le client devra utiliser, `client` est la machine cliente, et `local` est le port local utilisé par le service que l'on veut exposer de manière sécurisée au client. Encore une fois, cette commande ouvre une connexion SSH, il faut donc spécifier la machine à laquelle on se connecte (en l'occurrence, le client). Une commande peut être spécifiée de manière optionnelle, s'il n'y en a pas, un shell sera lancé sur la machine distante.

Note : Prenez bien conscience que SSH ne s'assure du chiffrement que des données transmises sur le réseau. Si la machine cliente ou la machine serveur ne sont pas correctement configurées, un pirate peut espionner les communications réseau internes à cette machine. Pour éviter ce type de risque, il faut utiliser des protocoles réseau eux-mêmes chiffrés. Autrement dit, une fois sorti du tunnel SSH, vous vous exposez à nouveau aux risques d'intrusion.

Il existe d'autres méthodes de tunneling sous Linux, qui sont sans aucun doute plus ergonomiques que SSH. En effet, non seulement les tunnels SSH ne peuvent pas être créés automatiquement (ils nécessitent une connexion), mais en plus les connexions SSH se terminent dès que le dernier client disparaît. En fait, les fonctionnalités de SSH sont un plus, mais elles ne sont pas destinées à remplacer des solutions plus adaptées pour réaliser des tunnels. Vous trouverez plus d'informations sur les réseaux privés virtuels ci-dessous.

IX-E-2-c - Utilisation d'IPSec

Le protocole IPSec est une extension du protocole IP qui permet de répondre aux besoins d'authentification et de confidentialité, grâce aux techniques de cryptographie. Les principes utilisés sont les mêmes que pour SSH, mais ils sont mis en pratique au plus bas niveau, ce qui fait que tous les protocoles de haut niveau en bénéficient automatiquement.

IX-E-2-c-i - Fonctionnement d'IPSec

IPSec définit des protocoles complémentaires permettant d'encapsuler les données transférées dans les paquets IP et assurant les services d'authentification et de confidentialité. Le protocole AH (abréviation de l'anglais « Authentication Header ») permet, comme son nom l'indique, d'assurer que les machines sont bien celles qu'elles prétendent être. Cela permet de supprimer tous les risques d'attaque de l'homme du milieu. Le protocole AH permet également d'assurer l'intégrité des données. En revanche, il n'en assure pas la confidentialité. Le protocole ESP (abréviation de l'anglais « Encapsulating Security Payload ») permet d'assurer cette confidentialité. En revanche, contrairement à ce que de nombreuses documentations affirment, il ne permet pas d'assurer l'authenticité des interlocuteurs, ce qui fait que la confidentialité peut être brisée par une attaque de l'homme du milieu. Il est donc nécessaire d'utiliser les deux protocoles si les interlocuteurs ne sont pas authentifiés de manière physique. Ces deux protocoles permettent également de prévenir les attaques par rejeu au niveau protocolaire, car ils utilisent un numéro de séquence dont la prédétermination est très difficile pour chaque paquet échangé.

Note : En réalité, le protocole ESP peut également assurer l'authenticité des données, même s'il ne permet pas d'assurer l'authenticité des interlocuteurs. Cette fonctionnalité peut être utile si l'authenticité des machines est assurée par un autre mécanisme. Toutefois, cette condition n'est généralement pas vérifiée et il faut utiliser le protocole AH pour authentifier les machines. De ce fait, le mécanisme d'authentification d'ESP est rarement utilisé.

AH et ESP peuvent être utilisées dans deux modes différents : le mode transport et le mode tunnel. Le mode transport est le mode de communication natif, dans lequel les en-têtes des protocoles AH et ESP sont insérés entre l'en-tête IP et ses données. De ce fait, les protocoles de haut niveau sont simplement encapsulés dans les protocoles AH et ESP. Le mode tunnel, quant à lui, permet de créer un réseau virtuel complet, en encapsulant la communication entre les interlocuteurs dans un canal chiffré. Ce sont donc les paquets IP de la communication qui sont eux-mêmes encapsulés dans les protocoles AH et ESP.

IPSec utilise les algorithmes de cryptographie symétriques pour assurer ces fonctionnalités. Par exemple, l'authenticité des machines et l'intégrité des données sont garanties par le protocole AH grâce à la génération de condensés cryptographiques paramétrés par une clef privée. Seules les machines partageant cette clef peuvent donc communiquer avec ce protocole, et toute modification des données serait détectée lors de la vérification de la signature. De même, ESP assure la confidentialité des données en les chiffrant avec un algorithme de chiffrement à clef privée.

Chaque canal de communication IPSec peut utiliser ses propres paramètres. Bien entendu, il est hors de question de transférer ces paramètres (et encore moins les clefs privées utilisées !) dans chaque paquet IP ! IPSec identifie donc ces paramètres par un numéro d'identification (appelé « Security Parameter Index », « SPI » en abrégé), grâce auquel chaque interlocuteur peut retrouver les paramètres de la connexion chiffrée. L'association entre le SPI et ces paramètres est appelée une « SA » (abréviation de l'anglais « Security Association »). Les associations de sécurité sont stockées sur chaque machine dans une base de données que l'on appelle la SAD (de l'anglais « SA Database »).

Les associations de sécurité ne permettent que de définir la manière dont les communications doivent être protégées. Cela dit, une machine n'est pas obligée de chiffrer toutes ses connexions, et l'on peut parfaitement vouloir ne chiffrer les communications qu'entre deux machines, ou que pour certains protocoles. Il est donc également nécessaire de définir quelles communications doivent être chiffrées. Cela se fait grâce à une politique de sécurité (« Security Policy » en anglais, « SP » en abrégé). Les politiques de sécurité sont également stockées dans une base de données sur chaque machine, base de données que l'on appelle la « SPD » (abréviation de « SP Database »).

Bien entendu, la définition des clefs secrètes, des associations et des politiques de sécurité peut devenir une opération très lourde sur de grands réseaux. En effet, il est nécessaire de définir des clefs privées pour chaque couple de machines. Par conséquent, un protocole de gestion des clefs et des associations de sécurité a été défini pour automatiser l'établissement des connexions. Ce protocole est le protocole « ISAKMP » (abréviation de l'anglais « Internet Security Association and Key Management Protocol »). Comme SSH, ce protocole s'appuie sur les algorithmes de cryptographie asymétriques pour échanger les clefs privées et définir les associations de sécurité de manière sûre avec des machines authentifiées. Ainsi, grâce à ce protocole, seuls les politiques de sécurité et les couples de clefs publiques/privées des machines doivent être définis. Les associations de sécurité et les clefs privées n'ont plus à être manipulées.

IX-E-2-c-ii - Configuration manuelle d'IPSec en mode transport

La configuration manuelle d'IPSec reste intéressante, car elle permet de bien saisir la manière dont les opérations se déroulent. Elle se fait simplement en définissant les SA et les SP sur chaque machine, à l'aide de l'outil setkey. Cet outil peut être retrouvé sur le site Web consacré à [IPSec](#).

Les associations et les politiques de sécurité peuvent être spécifiées directement en ligne de commande si on lance setkey avec l'option -c. Toutefois, la manière la plus simple de charger les associations et les politiques dans les bases de données du système est de les définir dans un fichier de configuration et de demander à setkey de le lire en lui fournissant le chemin sur ce fichier à l'aide de l'option -f. En général, ce fichier de configuration est placé dans le répertoire `/etc/` et se nomme `ipsec.conf`.

Les commandes acceptées par setkey sont nombreuses. Le fichier de configuration d'exemple suivant vous en donnera un aperçu :

```
#!/usr/sbin/setkey -f

# Vide la base de données des SAs :
```



```
flush;

# Vide la base de données des SPs :
spdflush;

# Définit les SAs pour le protocole AH :
# add source dest protocole SPI algo clef
add 170.20.57.31 170.20.58.40 ah 0x200 \
    -A hmac-md5 0x848255320c7fd11b6082f9628f078af7;
add 170.20.58.40 170.20.57.31 ah 0x201 \
    -A hmac-md5 0xcd23ef4b73ea54b626d16092293a298a;

# Définit les SAs pour le protocole ESP :
# add source dest protocole SPI algo clef
add 170.20.57.31 170.20.58.40 esp 0x300 \
    -E 3des-cbc 0xe6933d365096e13be514eald881d87371100b15119e2ffb9;
add 170.20.58.40 170.20.57.31 esp 0x301 \
    -E 3des-cbc 0xb2b630c9e9692c36d248b7206bda17c47f772f96290ac05f;

# Définit les politiques :
# spadd source dest protocole -P politique
# politique = direction action regles
# regle = ipsecp/mode/options/niveau
# options = règles de définition des tunnels
# niveau = use ou require
spdadd 170.20.57.31 170.20.58.40 any -P out ipsec
    esp/transport//require
    ah/transport//require;

spdadd 170.20.58.40 170.20.57.31 any -P in ipsec
    esp/transport//require
    ah/transport//require;
```

Ce fichier permet de décrire les associations de sécurité utilisées pour communiquer entre deux machines d'adresses 170.20.57.31 et 170.20.58.40, ainsi que la politique de sécurité de la machine 170.20.57.31. Bien que largement commenté, ce fichier requiert quelques explications complémentaires.

Tout d'abord, la première ligne permet de considérer ce fichier comme un fichier exécutable, dont l'interpréteur est la commande `setkey` elle-même. Les deux lignes suivantes permettent de vider les bases de données contenant les associations et les politiques de sécurité. Attention, cela a pour conséquence de désactiver complètement IPSec, les services réseau qui doivent être protégés ne doivent donc pas être lancés lorsque ce fichier est interprété.

Viennent ensuite les définitions des associations de sécurité utilisées par les protocoles AH et ESP. Ces associations doivent être définies strictement de la même manière sur les deux machines pour lesquelles la liaison est sécurisée. Elles sont simplement ajoutées à la base de données des associations à l'aide de la commande `add` de `setkey`. Cette commande prend en paramètre les adresses IP source et destination des machines intervenant dans la communication sécurisée, suivies du nom du protocole pour lequel l'association est définie (à savoir, `ah` ou `esp`), suivi de l'identifiant de l'association (le « SPI »), et pour finir des options relatives au protocole.

Les options des protocoles dépendent évidemment de ceux-ci. Le protocole AH n'accepte que l'option `-A`, qui permet de spécifier l'algorithme d'authentification à utiliser (ici, l'algorithme MD5). ESP quant à lui peut également accepter l'option `-E`, qui permet de spécifier un algorithme de chiffrement des données. Il est inutile de spécifier un algorithme d'authentification pour ESP si l'on utilise également le protocole AH, comme c'est le cas ici. Il est également possible de n'utiliser que le protocole ESP et de lui demander de réaliser à la fois l'authentification et le chiffrement des communications. Toutefois, comme on l'a dit plus haut, cela nécessite d'avoir déjà une authentification des machines (ce qui n'est pas assuré en général, une adresse IP pouvant être vampirisée par un pirate).

Les algorithmes utilisables sont variés et dépendent des options que vous avez spécifiées dans la configuration du noyau. Ceux utilisés ici sont relativement fréquents, à savoir MD5 pour l'authentification et le triple DES pour le chiffrement. Vous remarquerez que ces algorithmes ont besoin d'une clef secrète pour fonctionner. Dans le fichier d'exemple précédent, cette clef est spécifiée en hexadécimal (c'est-à-dire en base 16). La taille des clefs dépend de l'algorithme utilisé. Pour MD5, il vous faudra utiliser une clef de 16 octets, et pour le triple DES, une clef de 24 octets.

Les clefs de cet exemple ont été générées par lecture du fichier spécial de périphérique `/dev/random`, dont le but est de fournir des données aléatoires, avec la commande suivante :

```
dd if=/dev/random count=n bs=1 | xxd -ps
```

où `n` est le nombre d'octets à créer. Bien entendu, ces clefs devant rester secrètes, le fichier de configuration `ipsec.conf` ne doit pas être accessible en lecture aux autres utilisateurs que l'administrateur.

Note : Vous remarquerez que les associations de sécurité utilisent deux clefs privées distinctes pour les deux directions du flux de données entre les machines qui communiquent par IPSec. Cela n'est pas une obligation, on peut très bien utiliser la même clef pour les deux directions.

Comme il l'a déjà été indiqué, définir les associations de sécurité ne suffit pas pour sécuriser les communications. En effet, il est également nécessaire de définir la politique de sécurité. Cela est réalisé à l'aide de la commande `spdadd`. Cette commande prend en paramètre l'adresse source et l'adresse destination, le protocole ou le port qui doit être sécurisé (any signifiant que toutes les communications doivent être sécurisées), et la politique de sécurité elle-même. Cette politique permet d'indiquer la direction du trafic réseau (in signifiant le trafic entrant et out signifiant le trafic sortant), et ce que l'on doit faire des données. L'option `ipsec` indique qu'elles doivent être sécurisées par IPSec. L'option `discard` permet d'interdire le trafic entre les machines, et l'option `none` permet d'autoriser ce trafic et de laisser les communications se faire classiquement.

Si l'on indique qu'IPSec doit être utilisé, il faut indiquer, respectivement pour AH et ESP, le mode de fonctionnement (ici, `transport`) et le niveau de sécurité exigé. La valeur utilisée est généralement `require`, ce qui permet de n'accepter les communications que si elles se font par IPSec. Vous trouverez le détail des autres options dans la page de manuel de `setkey`

Ainsi, la première ligne `spdadd` du fichier donné en exemple ci-dessus indique que tout le trafic sortant de `170.20.57.31` et allant vers `170.20.58.40` doit être encapsulé en mode `transport` dans les protocoles AH et ESP, en utilisant les associations de sécurité définies précédemment. De même, la deuxième ligne `spdadd` indique que tout le trafic entrant dans `170.20.57.31` et en provenance de `170.20.58.40` doit être sécurisé par IPSec.

Vous constaterez que les adresses IP sources et destination sont inversées dans les deux commandes, parce que la direction indiquée est inversée et que l'on ne peut pas définir, sur la machine `170.20.57.31`, les règles à appliquer sur la machine `170.20.58.40`. Bien entendu, sur la machine `170.20.58.40`, ces directions devront être échangées. Ainsi, pour reprendre l'exemple, les lignes permettant de définir la politique de sécurité sur cette machine doivent être les suivantes :

```
# Définit la politique de sécurité sur 170.20.58.40 :
spdadd 170.20.57.31 170.20.58.40 any -P in ipsec
    esp/transport//require
    ah/transport//require;

spdadd 170.20.58.40 170.20.57.31 any -P out ipsec
    esp/transport//require
    ah/transport//require;
```

Note : Pour que les communications se fassent par IPSec, il est nécessaire que vous laissiez passer le trafic AH ou ESP au travers de votre pare-feu. Le protocole AH utilise le numéro de protocole 51 et le protocole ESP le numéro 50.

IX-E-2-c-iii - Configuration manuelle d'IPSec en mode tunnel

La configuration d'IPSec en mode tunnel est similaire à celle utilisée pour le mode `transport`. Elle est toutefois légèrement plus compliquée, puisqu'il est nécessaire de spécifier les adresses des passerelles entre lesquelles le tunnel sera établi. Le fichier de configuration suivant donne un exemple de définition de tunnel IPSec utilisant les adresses `192.168.20.1` et `192.168.20.2`, entre deux machines d'adresses `170.20.57.31` et `170.20.58.40` :

```
#!/usr/sbin/setkey -f

# Vide la base de données des SAs :
flush;

# Vide la base de données des SPs :
spdflush;

# Définit les SAs pour les en-têtes :
# add source dest protocole SPI algo clef
add 170.20.57.31 170.20.58.40 ah 0x200 -m tunnel -A hmac-md5 0x848255320c7fd11b6
082f9628f078af7;
add 170.20.58.40 170.20.57.31 ah 0x201 -m tunnel -A hmac-md5 0xcd23ef4b73ea54b62
6d16092293a298a;

# Définit les SAs pour les paquets :
# add source dest protocole SPU algo clef
add 170.20.57.31 170.20.58.40 esp 0x300 -m tunnel -E 3des-cbc 0xe6933d365096e13b
e514eald881d87371100b15119e2ffb9;
add 170.20.58.40 170.20.57.31 esp 0x301 -m tunnel -E 3des-cbc 0xb2b630c9e9692c36
d248b7206bda17c47f772f96290ac05f;

# Définit les politiques :
# spadd source dest protocoles politique
# politique = direction action regles
# regle = ipsecp/mode/options/niveau
# options = règles de définition des tunnels
# niveau = use ou require
spadd 192.168.20.2 192.168.20.1 any -P in ipsec
    esp/tunnel/170.20.58.40-170.20.57.31/require
    ah/tunnel/170.20.58.40-170.20.57.31/require;

spadd 192.168.20.1 192.168.20.2 any -P out ipsec
    esp/tunnel/170.20.57.31-170.20.58.40/require
    ah/tunnel/170.20.57.31-170.20.58.40/require;
```

Ce fichier commence par définir les associations de sécurité décrivant comment le trafic réseau doit se faire entre les machines 170.20.57.31 et 170.20.58.40. La première différence avec le mode transport est l'utilisation de l'option -m, qui prend en paramètre le mode de communication. Ici, il est demandé que tout le trafic IP, et non pas les communications elles-mêmes, soit encapsulé dans les protocoles AH et ESP. Cela a effectivement pour effet de faire un tunnel « IP over IPSec ».

La deuxième différence provient ensuite de la définition des politiques de sécurité. À présent, ces politiques ne portent plus sur les véritables adresses réseau des machines (à savoir 170.20.57.31 et 170.20.58.40), mais sur les adresses des interfaces réseau permettant d'accéder au tunnel (respectivement 192.168.20.1 et 192.168.20.2). De plus, les informations de routage pour le trafic dans le tunnel sont spécifiées dans les règles de la politique : il doit passer dans le tunnel IPSec établi entre 170.20.57.31 et 170.20.58.40.

Note : Du fait que les communications doivent être transférées via le tunnel IPSec, il est nécessaire d'autoriser le routage sur les deux machines du tunnel. Cela se fait en autorisant le routage au niveau du pare-feu et en exécutant la commande suivante :

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Les directions indiquées dans les politiques correspondent à la machine 170.20.57.31. Bien entendu, ces directions devront être inversées sur la machine 170.20.58.40 pour rendre la communication symétrique.

IX-E-2-c-iv - Autoconfiguration avec ISAKMP

La configuration manuelle d'IPSec est réalisable pour l'établissement d'un tunnel ou pour le chiffrement des connexions entre deux ou trois machines, mais au-delà de ce nombre, elle devient très vite lourde et contraignante. En effet, il est nécessaire de définir une clef privée de communication pour chaque couple de machines, ce qui peut

faire un grand nombre de clefs. Par exemple, pour un réseau de trois machines, pas moins de six clefs AH et ESP sont nécessaires (douze si ces clefs ne sont pas utilisées pour les deux sens du trafic) !

C'est pour cette raison que le protocole d'échange de clef ISAKMP a été développé. Ce protocole utilise le port 500 du protocole UDP pour définir les associations de sécurité à utiliser pour les connexions IPSec, et de les renouveler régulièrement afin d'accroître la sécurité des communications. En effet, utiliser toujours la même clef dans une association de sécurité peut permettre à un attaquant de la calculer en observant certains paquets réseau dont il peut deviner une partie du contenu, par exemple les paquets de demandes de connexion TCP.

Pour l'établissement des associations de sécurité, le protocole ISAKMP utilise plusieurs phases. La première phase permet d'établir une association de sécurité de contrôle entre les deux machines, par l'intermédiaire de laquelle les autres associations de sécurité seront établies. La deuxième phase est l'établissement des associations de sécurité utilisées pour les communications.

La première phase peut être réalisée selon plusieurs modes d'authentification. Le mode « principal » est le mode le plus sûr. Le mode « agressif » est plus économe en terme de trafic réseau, mais il n'est pas utilisable de manière fiable en pratique. Dans chacun de ces modes, plusieurs techniques d'authentification sont utilisables, mais, en raison d'erreurs de conception du protocole ISAKMP, seule l'authentification basée sur les mécanismes de clefs publiques est fiable. De ce fait, le protocole ISAKMP fonctionne de manière tout à fait similaire à OpenSSH, puisqu'il commence par établir une association de sécurité privée à partir des clefs publiques et privées, puis il utilise ce canal chiffré pour effectuer les échanges des associations de sécurité IPSec des communications. Dans ce type de configuration, il n'est plus nécessaire de définir qu'un couple de clefs privée et publique par machine.

Note : Pour ceux qui sont intéressés par les problèmes de sécurité, les autres mécanismes d'authentification proposés par le protocole ISAKMP sont :

- l'authentification par signature, qui est sujette à l'attaque de l'homme du milieu ;
- et l'authentification par secret partagé.

Ce dernier type d'authentification est très problématique, car il ne résout pas le problème de la gestion des clefs d'une part, et il requiert de connaître l'identité des machines qui se connectent d'autre part. Cela implique que les adresses IP des machines qui se connectent soient fixes dans le mode principal, et que les informations d'identité soient transférées sur le réseau pour éviter d'avoir à connaître ces adresses IP dans le mode agressif. Cela n'est pas le plus grave, le mode agressif transférant également des données chiffrées dont on connaît le contenu sur le réseau, permettant ainsi à un attaquant de faire une attaque par dictionnaire et de trouver le secret partagé.

Sous Linux, le protocole ISAKMP est implémenté par le démon racoon. racoon utilise des certificats x509 pour authentifier les machines, aussi vous faudra-t-il en générer pour chacune de vos machines.

Les certificats x509 sont des fichiers contenant l'identité d'une personne ainsi que sa clef publique, ce qui permet donc de l'authentifier de manière fiable à condition que l'on soit sûr du certificat provient bien d'elle. Pour s'assurer de cela, il est courant de faire signer le certificat par une autorité de certification en laquelle on a confiance et dont le rôle est de bien s'assurer que la personne qui demande la signature de son certificat est bien celle qu'elle prétend être (moyennant finances bien entendu). Il est possible de signer soi-même ses propres certificats, ce qui ne résout bien sûr pas le problème de l'authenticité du certificat, mais permet au moins de s'assurer qu'il n'a pas été modifié par une tierce personne. Cela suppose bien entendu que le problème du transfert de l'empreinte du certificat vers le destinataire est résolu.

Les certificats peuvent être créés et manipulés à l'aide de l'outil OpenSSL, dont la ligne de commande est malheureusement relativement complexe. La création d'un nouveau certificat se fait typiquement avec la commande suivante :

```
openssl req -new -nodes -newkey rsa:1024 -sha1 \  
-keyform PEM -keyout machine1.private -outform PEM -out requete.pem
```

Cette commande permet de créer une requête de certification d'une clef publique RSA de 1024 bits avec l'algorithme de génération de condensé SHA1, et de stocker la clef privée correspondante dans le fichier `machine1.private`. La requête elle-même est stockée dans le fichier `requete.pem`. Des informations permettant de vous identifier vous seront demandées, et seront incluses dans la requête de certification.

La création du certificat se fait ensuite simplement, en signant la requête de certification. Nous pouvons le faire nous-mêmes, en signant la requête avec la clef privée que l'on vient de créer. Pour cela, il faut utiliser la commande suivante :

```
openssl x509 -req -in requete.pem \  
-signkey machine1.private -out machine1.public
```

Une fois cette opération effectuée, vous pouvez vous débarrasser du fichier de requête de certification (c'est-à-dire du fichier `requete.pem`). Vous disposerez alors d'un couple de clef privée/publique dans les fichiers `machine1.private` et `machine1.public`.

Une fois que vous aurez créé vos clefs pour toutes vos machines, vous pourrez les distribuer sur vos différentes machines. Chaque machine doit bien entendu disposer de sa clef privée, ainsi que de l'ensemble des clefs publiques des autres machines. L'échange des clefs publiques est l'opération sensible, puisqu'il s'agit de s'assurer que les clefs reçues sont bien celles que l'on a émises. Pour cela, on pourra utiliser un condensé de la clef et s'assurer que ce condensé est bien le même après échange. Attention toutefois, il faut être certain que le condensé soit échangé de manière sûre. OpenSSL vous permet d'obtenir un condensé de vos clefs publiques avec la commande suivante :

```
openssl dgst clef.public
```

où `clef.public` est la clef dont on désire obtenir le condensé.

Le démon `racoon` utilise le fichier de configuration `/etc/racoon.conf` pour décrire la manière dont les différentes phases de la négociation des associations de sécurité doivent être faites. Il est possible de définir des options différentes pour différents clients, identifiés par leurs adresses IP. Toutefois, en raison de l'utilisation courante des adresses IP dynamiques, une configuration implicite doit souvent être donnée. Vous trouverez ci-dessous un exemple de fichier de configuration `racoon.conf` :

```
path certificate "/etc/ssl/certs";  
remote anonymous  
{  
    exchange_mode main;  
    certificate_type x509 "neptune.public" "neptune.private";  
    verify_cert off;  
    my_identifier asn1dn;  
    peers_identifier asn1dn;  
    proposal  
    {  
        encryption_algorithm 3des;  
        hash_algorithm md5;  
        authentication_method rsasig;  
        dh_group modp1024;  
    }  
}  
  
sainfo anonymous  
{  
    pfs_group modp1024;  
    encryption_algorithm 3des;  
    authentication_algorithm hmac_md5;  
    compression_algorithm deflate;  
}
```

La première ligne indique le chemin du répertoire dans lequel les fichiers de clefs publiques des clients se trouveront, ainsi que les fichiers de clefs publique et privée de la machine locale. La section `remote` permet de définir les options d'authentification d'un client pendant la première phase du protocole ISAKMP. Le mot-clé `anonymous` indique ici que

tous les clients pour lesquels une section `remote` plus spécifique n'est pas disponible devront utiliser cette section. C'est donc la section à utiliser pour tous les clients disposant d'une adresse IP dynamique.

La définition de la première phase du protocole contient les directives suivantes :

- la directive `exchange_mode`, qui indique le mode de fonctionnement à utiliser pendant cette phase (le mode principal est choisi ici, parce que c'est le seul qui soit réellement sûr) ;
- la directive `certificate_type`, qui permet d'indiquer que le mécanisme d'authentification utilise des certificats X509 (grâce à l'option `x509`) ;
- la directive `verify_cert`, qui permet d'indiquer si les certificats manipulés doivent être vérifiés auprès d'une autorité de certification ou non (la configuration présentée ici ne le permet pas, car nous avons signé nous-mêmes nos certificats) ;
- les directives `my_identifier` et `peers_identifier`, qui indiquent la manière dont les machines sont identifiées (leur option `asn1dn` indique que les informations qui ont été demandées lors de la création des certificats des machines sont utilisées pour retrouver leurs certificats) ;
- et enfin, la directive `proposal`, qui permet de fournir les paramètres cryptographiques utilisés pour l'association de sécurité de contrôle négociée pendant la première phase du protocole ISAKMP.

Les paramètres de l'association de sécurité de contrôle permettent de spécifier l'algorithme de chiffrement et l'algorithme de génération des signatures (directives `encryption_algorithm` et `hash_algorithm` respectivement), ainsi que la méthode d'authentification des machines (utilisation du protocole RSA, directive `authentication_method`). La directive `dh-group` permet de fournir un paramètre à l'algorithme d'échange de clefs Diffie Hellman, utilisé lors de l'échange des clefs privées.

La section `sainfo` permet de définir les paramètres des associations de sécurité négociées pendant la deuxième phase du protocole ISAKMP. Son format est semblable à la section de définition de l'association de sécurité de contrôle vue précédemment. Toutefois, une différence importante est que la méthode d'authentification est à présent basée sur un secret partagé et non plus sur le mécanisme des clefs publiques et privées des machines. De ce fait, l'algorithme de génération de signatures utilisé par le protocole AH nécessite une clef secrète échangée pendant la première phase. Ce n'est donc plus un simple algorithme de génération d'empreintes. La directive `hmac_md5` permet d'utiliser un algorithme MD5 modifié pour être paramétré par la clef secrète.

Vous noterez que toutes ces informations permettent de définir la manière dont les associations de sécurité doivent être générées. Cependant, ces associations de sécurité ne sont pas forcément utilisées : il faut pour cela définir la politique de sécurité. Cette opération peut bien entendu se faire avec `setkey`. Le fichier de configuration `ipsec.conf` d'un serveur ressemble alors à celui-ci :

```
spdadd 0.0.0.0/0 serveur any -P in ipsec
    esp/transport//require
    ah/transport//require;

spdadd serveur 0.0.0.0/0 any -P out ipsec
    esp/transport//require
    ah/transport//require;
```

Les fichiers de configuration `ipsec.conf` des clients devront avoir des directions de trafic inversées. Vous noterez que les adresses spécifiées pour les clients utilisent des plages d'adresses dans les commandes d'ajout de règles de politique de sécurité. En effet, on ne sait a priori pas quelles sont les adresses de ces clients, car ils auront généralement une adresse IP allouée dynamiquement. La seule adresse connue est en général celle du serveur, référencée ici par le nom de machine `serveur`. Si l'adresse du serveur est elle aussi dynamique, la configuration est plus contraignante, car il faut que toutes les communications se fassent par IPSec (et les clients ne peuvent plus se connecter qu'aux serveurs qu'ils connaissent).

Il est également possible de demander à `racoon` de définir lui-même la politique de sécurité, lorsque des tentatives de connexion ont lieu. Cela peut être réalisé à l'aide de la directive `generate_policy` de la section `remote`.

Une fois le fichier de configuration `racoon.conf` défini, il ne reste plus qu'à lancer `racoon`. Ceci peut se faire avec la commande suivante :


```
racoon -f /etc/racoon.conf
```

L'option -f permet de lui indiquer le fichier de configuration qu'il doit utiliser. Si vous désirez lancer racoon en avant-plan pour pouvoir déboguer votre configuration, vous pouvez utiliser -F, et augmenter le niveau de traces avec l'option -d.

IX-F - Configuration des fonctions serveur

En général, les gens ne voient Internet que du point de vue client en termes d'architecture client / serveur. Cela signifie qu'ils se contentent de se connecter à Internet, et n'utilisent que des logiciels clients qui communiquent avec des serveurs situés sur Internet. Ce mode d'utilisation est de loin le plus courant et celui qui convient à la majorité des gens, qui n'ont en général même pas conscience de l'existence du monde des serveurs.

Cependant, il est parfois nécessaire de mettre en place des petits serveurs, sans forcément être un fournisseur d'accès à Internet ou un hébergeur de site Web. Par exemple, il peut être intéressant de fournir la possibilité de réaliser une connexion par téléphone à un ordinateur distant que l'on doit laisser allumé pour une raison ou une autre, afin de réaliser une opération de maintenance. Il est également possible de faire une liaison PPP entre deux ordinateurs par un câble série pour échanger des données, ou pour relier par l'intermédiaire du réseau téléphonique deux réseaux séparés physiquement.

Cette section présente donc les techniques de base permettant de paramétrer Linux pour accepter les communications entrantes, tant pour satisfaire aux demandes de connexion distantes que pour établir une liaison PPP.

IX-F-1 - Paramétrage des connexions extérieures

Nous avons vu dans la [Section 6.8](#) que le programme en charge de demander l'identification des utilisateurs sur les terminaux de login était le programme getty. Ce programme ne permet aux utilisateurs que de s'identifier. Lorsque quelqu'un saisit son nom, getty se contente de lancer le processus login pour que celui-ci réalise l'authentification de l'utilisateur en lui demandant son mot de passe. En fait, getty peut très bien lancer un autre programme que login si l'identification n'est pas nécessaire sur le terminal qu'il gère, mais, en général, il faut identifier les utilisateurs.

Le programme getty est donc le point d'entrée par lequel il faut nécessairement passer pour utiliser le système. Dans le cas des terminaux virtuels, getty utilise une ligne de communication virtuelle, mais il peut parfaitement utiliser une ligne de communication réelle (c'est d'ailleurs ce pour quoi il a été conçu à l'origine). En particulier, il est possible de lancer un processus getty sur une ligne série, afin de permettre l'utilisation de terminaux réels connectés sur le port série correspondant.

La ligne série sur laquelle le processus getty fonctionne n'est pas forcément connectée à un terminal physique. Elle peut parfaitement être connectée à un modem. Dans ce cas, les utilisateurs devront appeler le serveur pour se connecter à l'aide d'un logiciel d'émulation de terminal. Il existe plusieurs logiciels permettant d'appeler un ordinateur distant et d'émuler un terminal. On citera par exemple le logiciel HyperTerminal de Windows, et les programmes Telix et Minicom. Ce dernier est couramment fourni avec les distributions de Linux, car c'est un logiciel libre.

En fait, il existe plusieurs variantes de getty, spécialisées pour des usages spécifiques, et permettant de faciliter le paramétrage du fichier de configuration `/etc/inittab`, dans lequel sont définis les terminaux disponibles. Les distributions de Linux sont souvent fournies avec les versions suivantes :

- mingetty, qui est un getty simplifié, couramment utilisé pour les terminaux virtuels ;
- mgetty, qui est une version spécialement conçue pour être utilisée sur les lignes série connectées à un modem. mgetty est en particulier capable d'initialiser les modems, de décrocher automatiquement la ligne lorsqu'il y a un appel, et de gérer les principales commandes des modems compatibles Hayes ;
- agetty, qui est une version complète de getty, utilisable aussi bien sur les lignes série (avec ou sans modem) que pour les terminaux virtuels. On utilisera de préférence cette version si l'on désire proposer une connexion par l'intermédiaire d'un câble null-modem.

Note : Les câbles null-modem sont des câbles série symétriques, qui permettent de relier deux ordinateurs par l'intermédiaire de leurs ports série. Ces câbles peuvent être trouvés chez la plupart des revendeurs de matériel informatique (vous pouvez également en fabriquer un vous-même si vous avez du temps à perdre).

Connaissant les diverses possibilités de ces programmes, proposer un service de connexion extérieur est relativement simple. Il suffit en effet de rajouter dans le fichier de configuration `/etc/inittab` les lignes permettant de lancer les programmes `getty` sur les bonnes lignes. Par exemple, pour offrir une connexion modem sur le deuxième port série (couramment appelé COM2), on ajoutera la ligne suivante :

```
S1:23:respawn:/usr/sbin/mgetty -n 3 -s 38400 ttyS1
```

Cette commande indique à `init` que `mgetty` doit être relancé à chaque fois qu'il se termine dans les niveaux d'exécution 2 et 3. Il doit prendre le contrôle de la ligne série utilisée par le fichier spécial de périphérique `/dev/ttyS1`, utiliser la vitesse de communication de 38400 bauds sur cette ligne (option `-s`) et ne décrocher qu'au bout de la troisième sonnerie (option `-n`). Vous pouvez bien entendu ajuster ces options en fonction de vos besoins.

En revanche, si l'on désire offrir une possibilité de connexion sur un port série par l'intermédiaire d'un câble null-modem, on utilisera plutôt le programme `agetty`. La ligne à ajouter dans le fichier `/etc/inittab` sera alors :

```
S1:123:respawn:/sbin/agetty -L 38400 ttyS1
```

Notez que les options sont quelque peu différentes. La vitesse de la ligne est donnée directement, et l'option `-L` indique que la ligne série utilisée est locale et ne gère pas le signal de porteuse CD (abréviation de l'anglais « Carrier Detect ») habituellement utilisé pour détecter les coupures téléphoniques. Encore une fois, vous pouvez adapter ces commandes selon vos besoins. N'hésitez pas à consulter les pages de manuel des programmes `mgetty` et `agetty` pour obtenir plus de renseignements.

Note : Vous pouvez tester vos connexions série à l'aide de l'émulateur de terminal `minicom`. S'il est installé sur votre distribution, vous pourrez le lancer simplement en tapant la commande `minicom`. Si vous l'utilisez sur une connexion directe (c'est-à-dire par un câble null-modem), vous devrez utiliser l'option `-o`, afin d'éviter que `minicom` n'envoie les codes d'initialisation du modem sur la ligne série.

Il se peut que vous ne parveniez pas à vous connecter sur vos ports série. Dans ce cas, la première des choses à faire est de regarder les paramètres de la ligne série à l'aide de la commande `stty`. Vous devriez vous intéresser tout particulièrement à la vitesse utilisée, et éventuellement vous assurer que les deux ports série des deux ordinateurs utilisent la même vitesse, ou que le modem est capable de gérer la vitesse utilisée par le port série. Si besoin est, vous pouvez modifier la vitesse d'un port série à l'aide de la commande `stty`. Par exemple, pour configurer le port série COM2 à la vitesse de 115200 bauds, vous devez taper la commande suivante :

```
stty -F /dev/ttyS1 ispeed 115200
```

IX-F-2 - Configuration des liaisons PPP

Le protocole PPP utilisé par les fournisseurs d'accès à Internet est, comme son nom l'indique (« Point to Point Protocol », ou « protocole de liaison point à point »), un protocole permettant de relier deux ordinateurs uniquement. Ce protocole est complètement symétrique, et la mise en place d'un serveur PPP n'est pas plus compliquée que celle d'un client. Les liaisons PPP sont donc couramment utilisées pour relier deux réseaux distants par l'intermédiaire des lignes téléphoniques.

Lorsque nous avons présenté la configuration des accès à Internet, nous avons utilisé des options spécifiques à ce cas de figure afin de gérer le fait que les paramètres réseau sont imposés par le fournisseur d'accès. Dans le cas d'un serveur, ces paramètres ne sont pas nécessaires, puisque nous sommes cette fois à la place du fournisseur d'accès. En revanche, certains paramètres supplémentaires devront être spécifiés, notamment pour configurer le modem afin d'accepter les connexions externes.

En général, il faut lancer une instance du démon `pppd` pour chaque ligne série utilisée pour les connexions entrantes, car `pppd` ne peut gérer qu'une seule connexion à la fois. Cette technique peut paraître lourde, mais en fait elle est relativement pratique, car elle permet de spécifier des options spécifiques à chaque ligne série. Comme d'habitude, ces options peuvent être spécifiées sur la ligne de commande de `pppd`, mais elles peuvent également être enregistrées dans un fichier de configuration. Le fichier de configuration global `/etc/ppp/options` est généralement utilisé pour stocker les options communes à toutes les lignes série. Les options spécifiques à chaque ligne sont enregistrées quant à elles dans les fichiers `options.ttySx`, où `ttySx` est le nom du fichier spécial de périphérique utilisé pour accéder à cette ligne. L'usage de ces fichiers de configuration permet de simplifier notablement les lignes de commandes utilisées pour lancer `pppd`.

La ligne de commande à utiliser pour lancer `pppd` en mode serveur diffère peu de celle utilisée pour établir une connexion PPP en tant que client. Les deux principales différences proviennent du fait que :

- le serveur doit spécifier l'adresse IP qu'il utilisera pour l'interface PPP et l'adresse IP qu'il donnera au client lors de l'établissement de la liaison ;
- le script chat utilisé ne doit pas composer un numéro, mais initialiser le modem pour qu'il réponde aux appels entrants.

Les adresses IP sont spécifiées en ligne de commande à l'aide de la syntaxe suivante :

```
locale:distante
```

où `locale` est l'adresse IP de l'interface PPP du serveur, et `distante` est l'adresse IP que le client recevra lorsqu'il se connectera. Ces deux adresses peuvent appartenir au même réseau (auquel cas le lien PPP constitue un réseau en soi) ou non (le lien PPP constitue alors pas un pont entre deux réseaux distincts). Ainsi, l'option suivante :

```
192.168.30.1:192.168.31.1
```

indiquera à `pppd` que l'adresse IP de l'interface PPP locale sera `192.168.30.1`, et que les clients qui se connecteront sur la ligne correspondante devront recevoir l'adresse `192.168.31.1`.

Notez que les adresses utilisées ici font partie des adresses IP réservées pour les réseaux locaux. Ce n'est pas ce que font les fournisseurs d'accès à Internet en général, car ils disposent d'un ensemble d'adresses IP réelles redistribuables, mais les adresses réservées sont parfaitement utilisables. Le seul cas où cela pourrait être gênant serait si l'on voulait faire passer les adresses des deux passerelles de la liaison PPP sur Internet, mais les paquets utilisant ces adresses ne circulent justement que sur cette liaison (à moins que les passerelles ne se trouvent sur des réseaux locaux et utilisent des adresses de ces réseaux pour la liaison PPP, auquel cas, de toute manière, il faudrait utiliser la technique du masquering pour connecter ces réseaux sur Internet).

Le script d'initialisation du modem, quant à lui, peut être spécifié à l'aide de l'option `init` de `pppd`. Cette option fonctionne exactement comme l'option `connect` que l'on a utilisée pour établir la liaison téléphonique dans la [Section 9.3.2](#). La différence est ici que l'on ne cherche pas à composer un numéro de téléphone, mais plutôt à initialiser le modem pour qu'il décroche automatiquement la ligne lorsqu'il reçoit un appel.

Il existe malheureusement plusieurs dialectes de langages de commandes utilisés par les modems. En général, chaque modem dispose donc de ses propres commandes, et il faut consulter sa documentation pour savoir leurs syntaxes. Malgré cela, ce n'est pas trop gênant, car la plupart des modems sont compatibles avec le standard Hayes, dont les commandes sont bien connues. Pour l'initialisation d'un modem pour un serveur, les commandes d'initialisation suivantes seront très utiles :

Commande	Signification
ATS0=n	Décrochage automatique si <code>n</code> vaut 1, manuel si <code>n</code> vaut 0.
AT&C1	Activation du contrôle de la ligne CD (« Carrier Detect ») pour signaler les raccrochages des clients. Cette

	commande n'est utile que si l'option modem est passée en paramètre à pppd sur sa ligne de commande.
AT&K3	Activation du contrôle de flux matériel (pour optimiser les échanges de données entre le modem et l'ordinateur). Cette option nécessite d'utiliser l'option crtscts dans la ligne de commande de pppd.

Finalement, la ligne de commande à utiliser est donc la suivante :

```
pppd port vitesse detach modem crtscts init "/usr/sbin/chat -v \
-f /etc/ppp/script" &
```

où `port` est le nom du fichier spécial de périphérique à utiliser pour cette connexion, par exemple `/dev/ttyS0`, `vitesse` est la vitesse de la ligne, par exemple `115200`, et `/etc/ppp/script` est le nom du fichier du script chat devant servir à l'initialisation du modem. L'option `detach` permet de demander à pppd de se détacher du terminal courant, afin que l'on puisse le lancer en arrière-plan.

Vous trouverez ci-dessous un script chat d'exemple, permettant de spécifier les options vues ci-dessus :

```
"" ATZ
OK AT&K3=1
OK AT&C1
OK AT&K3
OK ""
```

Note : Vous remarquerez qu'il n'est pas nécessaire, contrairement à ce que l'on a vu lors de la configuration d'un client PPP, de spécifier le nom de l'utilisateur et le protocole d'identification et d'authentification utilisé. Cela est normal, car le serveur n'a pas à s'identifier auprès de la machine cliente en général (il peut bien entendu le faire s'il le désire, mais dans ce cas seuls les clients capables de gérer ce type de connexion pourront se connecter).

En revanche, il peut être nécessaire d'exiger que les clients s'identifient et s'authentifient à l'aide de l'un des protocoles PAP ou CHAP. Pour cela, il suffit simplement d'ajouter les noms des clients et leurs secrets respectifs dans les fichiers de secrets `/etc/ppp/pap-secrets` et `/etc/ppp/chap-secrets`, et d'ajouter l'une des options `require-pap` ou `require-chap` sur la ligne de commande du démon pppd. Remarquez que le format des fichiers `pap-secrets` et `chap-secrets` est le même aussi bien pour les connexions entrantes que pour les appels vers l'extérieur. Cela signifie que l'on doit donner d'abord le nom du client, ensuite le nom du serveur, et ensuite son secret et enfin la liste des adresses IP que les clients peuvent utiliser.

Remarquez également que la colonne spécifiant la liste des adresses IP ne doit pas rester vide lorsqu'on désire créer un secret pour le serveur. En effet, pppd se base sur l'adresse IP de la machine distante et sur le nom du client pour déterminer le secret à utiliser lors de l'authentification. Cependant, si l'on désire utiliser les mêmes secrets pour toutes les lignes (et donc pour toutes les adresses que le client peut se voir imposer), on pourra utiliser une étoile. Cela signifie que toutes les adresses correspondront à la ligne, et seul le nom du client sera utilisé pour déterminer le secret à utiliser. Par exemple, si le serveur se nomme `monserveur` et le poste client s'appelle `jdupont`, on utilisera la ligne suivante :

#Client	Serveur	Secret	Adresses
jdupont	monserveur	gh25k;f	*

Lorsque la liaison PPP est établie, le démon pppd configure l'interface réseau correspondante. Le protocole utilisé par défaut est bien entendu le protocole IP. En général, il est nécessaire de configurer cette interface afin de spécifier les règles de routage permettant au client d'envoyer et de recevoir des paquets. Lors de l'établissement de la liaison,

le démon `pppd` ajoute donc automatiquement une règle de routage pour indiquer que tous les paquets à destination de l'adresse du client doivent passer par l'interface `ppp` de la liaison courante. Cette règle convient dans la majorité des cas, et en tout cas pour les fournisseurs d'accès à Internet, car le client ne dispose que d'une seule machine. En revanche, si la liaison PPP est utilisée pour relier deux réseaux, les règles de routages par défaut ne suffiront plus, car des paquets à destination de toutes les machines du réseau distant doivent être envoyés via l'interface `ppp` (et inversement). Pour résoudre ces petits problèmes, on devra compléter ou modifier les scripts `/etc/ppp/ip-up` et `/etc/ppp/ip-down`. Ces deux scripts sont appelés par le démon `pppd` respectivement lors de l'établissement de la liaison PPP et lors de sa terminaison. On placera donc les commandes de routage complémentaires dans le script `ip-up`, et on fera le ménage correspondant dans le script `ip-down`.

Remarquez que ces deux scripts sont communs à toutes les interfaces `ppp` que les différents démons `pppd` peuvent utiliser. Ils sont également utilisés lors de l'établissement d'une connexion en tant que client. Leur modification devra donc être entourée d'un soin extrême. Afin de distinguer les différents cas d'utilisation, ces scripts sont appelés avec les paramètres suivants :

Paramètre	Signification
\$1	Nom de l'interface réseau (par exemple, « <code>ppp0</code> »).
\$2	Nom du fichier spécial de périphérique de la ligne série utilisée (par exemple, « <code>ttyS0</code> »).
\$3	Vitesse de la ligne série.
\$4	Adresse IP locale de l'interface réseau.
\$5	Adresse IP du client.
\$6	Paramètre complémentaire, que l'on peut passer au script à l'aide de l'option <code>ipparam</code> de <code>pppd</code> .

Un script classique fait un test sur le nom de l'interface réseau et ajuste les règles de routage en fonction de cette interface, en se basant sur les adresses IP locales et distantes reçues en paramètre. La plupart des distributions fournissent un script d'exemple que vous pouvez bien entendu modifier soit directement, soit à l'aide de leur outil de configuration.

Le démon `pppd` se termine à chaque fois que la liaison PPP se termine. Ce comportement est normal pour un client, mais ce n'est pas généralement ce que l'on cherche à faire pour un serveur. Il faut donc relancer `pppd` régulièrement à chaque fois qu'il se termine. Cela peut être réalisé en ajoutant sa ligne de commande dans le fichier de configuration `/etc/inittab` dans les niveaux d'exécution adéquats. Comme `init` impose une taille réduite sur les lignes de commandes des programmes qu'il peut lancer, il est nécessaire d'utiliser les fichiers d'options de `pppd`. Par exemple, si l'on veut assurer le service PPP pour une ligne entrante sur le deuxième port série dans les niveaux d'exécution 2 et 3, on utilisera la ligne suivante :

```
ppp1:23:respawn:/usr/sbin/pppd /dev/ttyS1 115200 \
192.168.30.1:192.168.31.1"
```

avec les options suivantes activées dans le fichier d'options `/etc/ppp/options.ttyS1` :

```
# Initialisation du modem :
init "/usr/sbin/chat -v -f /etc/ppp/client.ttyS1
# Protocole d'authentification :
require-pap
# Contrôle de la ligne téléphonique :
modem
# Contrôle de flux matériel :
crtscts
```

IX-F-3 - Liaison de deux ordinateurs par un câble série

La connexion entre deux ordinateurs par l'intermédiaire d'un câble série est un cas particulier des liaisons PPP. Les techniques utilisées sont les mêmes, mais plusieurs simplifications importantes peuvent être faites :

- les deux ordinateurs sont reliés directement par un câble null-modem, et aucun modem ne s'intercale pour compliquer les choses. Par conséquent, on peut supprimer les scripts chat d'initialisation et de connexion ;
- on a le contrôle physique sur les deux ordinateurs, ce qui implique que les mécanismes d'identification et d'authentification sont superflus. Il est donc inutile de préciser les secrets PAP ou CHAP sur chaque machine ;
- il n'y a que deux ordinateurs à relier, donc les règles de routages sont élémentaires ;
- enfin, les adresses choisies sont connues d'avance et peuvent être spécifiées de manière symétrique sur le client et le serveur.

Il découle de ces simplifications que l'établissement d'une liaison PPP au travers d'un câble null-modem est une opération très facile. Comme vous allez le constater, le protocole PPP est effectivement symétrique.

Sur le « serveur », il suffit de lancer par exemple la commande suivante :

```
pppd /dev/ttyS0 115200 192.168.30.1:192.168.31.1 detach crtscts &
```

Cette commande permet de lancer le démon pppd sur le port série COM1 (fichier spécial de périphérique `/dev/ttyS0`) en utilisant la vitesse de 115200 bauds (vitesse maximale généralement supportée par les ports série des ordinateurs), en utilisant l'adresse IP locale `192.168.30.1` et l'adresse IP distante `192.168.31.1`. Le démon pppd doit se détacher du terminal, car il est lancé en arrière-plan. Enfin, le contrôle de flux matériel est utilisé (option `crtscts`).

La commande sur le « client » est absolument symétrique, puisque seules les adresses locales et distantes sont interverties :

```
pppd /dev/ttyS0 115200 192.168.31.1:192.168.30.1 detach crtscts &
```

Bien entendu, le port série utilisé peut être différent de celui du serveur, mais il est impératif que les deux ports série soient configurés pour utiliser les mêmes paramètres de communication (vitesse, nombre de bits de données, bit de parité, bit d'arrêt). Vous aurez donc peut-être à utiliser la commande `stty` pour fixer ces paramètres sur le serveur et sur le client.

Une fois ces deux commandes lancées, la connexion PPP est établie. Le démon pppd ajoute alors automatiquement une règle de routage pour acheminer les paquets d'un ordinateur à l'autre. Vous n'aurez donc pas à modifier les scripts `/etc/ppp/ip-up` et `/etc/ppp/ip-down`. Comme vous pouvez le constater, l'utilisation de PPP pour relier simplement deux ordinateurs n'est pas une opération très compliquée...

IX-F-4 - Configuration d'un serveur DHCP

Le protocole DHCP est géré, au niveau serveur, par le démon `dhcpd`. Ce démon se configure par l'intermédiaire du fichier de configuration `/etc/dhcpd.conf`, dans lequel se trouvent les définitions des réseaux à configurer, ainsi que les différentes informations qui doivent être communiquées aux clients. Ce fichier de configuration peut contenir plusieurs déclarations de réseau différentes, chacune étant introduite par le mot clé `subnet`. Ces déclarations permettent de définir les options pour les clients de différents réseaux de manière indépendante, mais si certaines de ces options sont communes à tous vos sous-réseaux, vous pouvez les factoriser en tête du fichier `dhcpd.conf`. Par exemple, le fichier de configuration suivant :

```
# Exemple de fichier de configuration dhcpd.conf :
default-lease-time 600;
max-lease-time 86400;
option subnet mask 255.255.255.0;
```



```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.10 192.168.1.100;
    range 192.168.1.150 192.168.1.200;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.1;
    option domain-name "monrezo.perso";
}

subnet 192.168.53.0 netmask 255.255.255.0 {
    range 192.168.53.10 192.168.53.254;
    option broadcast-address 192.168.53.255;
    option routers 192.168.53.2;
    option domain-name "autrezezo.org";
}
```

permet de configurer les deux réseaux 192.168.1.0 et 192.168.53.0 en spécifiant un masque de sous-réseau égal à 255.255.255.0 et une durée de bail d'adresse IP par défaut de 10 minutes (option `default-lease-time`) et maximale d'un jour (option `max-lease-time`). Vous remarquerez que ces dernières options utilisent la seconde comme unité de temps.

Comme vous pouvez le constater, la syntaxe du fichier de configuration `dhcpd.conf` n'est pas bien compliquée. Le mot clé `subnet` doit être suivi de l'adresse du réseau sur lesquels les clients à configurer par DHCP se trouvent. Le mot clé `netmask` doit être suivi quant à lui par le masque de ce sous-réseau. Les adresses IP attribuées aux clients peuvent être choisies dans différentes plages de valeurs, chacune étant spécifiée par le mot clé `range` suivi de l'adresse de début et de l'adresse de fin de la plage. Il est également possible d'attribuer des adresses fixes à certaines machines, avec la syntaxe suivante :

```
hardware ethernet 08:00:2b:4c:59:23;
fixed-address 192.168.1.101;
```

où les mots clés `hardware ethernet` introduisent l'adresse Ethernet de l'interface réseau de la machine cliente, et où `fixed-address` spécifie l'adresse IP que cette machine doit utiliser pour cette interface.

Enfin, un grand nombre d'options peuvent également être spécifiées, par exemple pour définir les adresses de diffusion (mot clé `broadcast-address`), les adresses des passerelles (mot clé `routers`), les noms de domaines (mot clé `domain-name`) ou encore les durées des bails accordés aux machines clientes (mots clés `default-lease-time` et `max-lease-time`). La liste complète des options utilisables est donnée dans la page de manuel `dhcpd.conf` et je vous invite à la consulter si vous désirez en savoir plus.

Lorsque vous aurez rédigé le fichier de configuration adapté à votre installation, il ne vous restera plus qu'à lancer le démon `dhcpd`. Celui-ci utilise la syntaxe suivante :

```
dhcpd interface0 [interface1 [&#8230;]]
```

où `interface0`, `interface1`, etc., sont les interfaces réseau utilisées pour accéder aux réseaux à configurer en DHCP. Ces interfaces doivent être configurées avec des adresses de réseau pour lesquelles il est possible de trouver une section `subnet` correspondante dans le fichier de configuration `/etc/dhcpd.conf`. Enfin, la commande de démarrage du démon `dhcpd` pourra être placée dans les fichiers d'initialisation du serveur DHCP.

Note : Le démon `dhcpd` utilise les fonctionnalités d'accès direct aux cartes réseau et de filtrage des paquets du noyau Linux pour écouter sur le réseau les demandes de configuration par DHCP des clients. Par conséquent, vous devrez activer ces options dans la configuration du noyau si ce n'est déjà fait. Pour cela, il vous faudra cocher les options « **Packet socket** », « **Packet socket: mmaped IO** » et « **Socket Filtering** » du menu « **Networking options** ». Vous devrez également activer l'option « **IP: multicasting** » dans la liste des options pour le protocole IP afin de permettre au démon d'effectuer des émissions de paquets en mode broadcast. Le détail de la configuration et de la compilation du noyau a été vu dans la [Section 7.3](#).

IX-G - Systèmes de fichiers en réseau

L'une des utilités principales d'un réseau est de permettre le partage des fichiers entre plusieurs ordinateurs. Cela nécessite de disposer d'un système de fichiers réseau sur les machines qui désirent accéder à des fichiers d'une autre machine, et d'un programme serveur sur les machines ainsi accédées par les clients.

Il existe deux grands standards de protocole pour les systèmes de fichiers en réseau. Le premier standard fonctionne quasiment exclusivement sous Unix, il s'agit du protocole « NFS » (abréviation de l'anglais « Network File System »), introduit originellement par Sun Microsystems et repris par la suite par les autres éditeurs de systèmes Unix. Le deuxième protocole est le protocole « SMB » (abréviation de l'anglais « Server Message Block »), qui a été introduit par Microsoft pour les systèmes DOS et Windows.

Les deux protocoles sont incompatibles, et si les deux solutions fonctionnent parfaitement en environnements homogènes, il est assez difficile de faire communiquer les deux types de systèmes. Les principaux problèmes proviennent de l'impossibilité d'utiliser le protocole réseau NetBIOS d'IBM sur les machines Unix, et des limitations des systèmes Microsoft en ce qui concerne la gestion des utilisateurs, les droits d'accès et les liens symboliques. De plus, la différence de gestion des fins de lignes dans les fichiers textes entre Unix et les systèmes Microsoft pose des problèmes qui peuvent difficilement être résolus de manière systématique.

Malgré ces limitations, les machines fonctionnant sous DOS ou Windows peuvent accéder aux systèmes de fichiers NFS des machines Unix grâce à des programmes spéciaux. Ces programmes sont souvent des extensions propriétaires aux systèmes Microsoft, et peuvent être relativement coûteux sans pour autant garantir une grande fiabilité et des performances honnêtes. L'utilisation de NFS sur les machines Windows ne sera donc pas traitée ici. Inversement, les machines Unix peuvent accéder aux partages Microsoft, mais là encore, il peut se présenter des difficultés. La principale difficulté est ici la nécessité d'encapsuler NetBIOS dans le protocole TCP/IP. En effet, le protocole SMB utilise les messages NetBIOS, protocole qui n'est pas géré nativement par les machines Unix. Dans tous les cas, des logiciels complémentaires sont nécessaires.

IX-G-1 - Installation d'un serveur de fichiers NFS

L'installation d'un serveur NFS est la solution de prédilection pour le partage des fichiers sous Unix. Il faut savoir cependant que NFS n'offre pas des performances remarquables d'une part, et qu'il ouvre des brèches dans la sécurité du système d'autre part.

Comme il l'a été dit ci-dessus, l'installation d'un serveur NFS requiert l'utilisation de programmes spécifiques. Plus précisément, NFS utilise deux démons pour fournir les services NFS aux machines clientes. Ces deux démons se nomment respectivement « mountd » et « nfsd ». Le premier démon prend en charge les requêtes de montage des systèmes de fichiers NFS, et nfsd effectue les requêtes d'accès aux fichiers pour les clients. Par ailleurs, ces deux démons utilisent les mécanismes d'appels de procédure à distance « RPC » (abréviation de l'anglais « Remote Procedure Call » de Sun Microsystems). Ces services sont désormais disponibles sur toutes les machines Unix et constitue la norme en la matière. Il est donc également nécessaire de mettre en place les services RPC avant de lancer les démons NFS.

Le principe de fonctionnement des appels de procédures à distance est le suivant. Chaque programme désirant fournir des services RPC écoute sur un port TCP ou UDP pour des requêtes éventuelles. Les clients qui veulent utiliser ces services doivent envoyer leurs requêtes sur ce port, en précisant toutes les informations nécessaires à l'exécution de cette requête : numéro de la requête et paramètres de la requête. Le serveur exécute cette requête et renvoie le résultat. Les bibliothèques RPC fournissent les fonctions nécessaires pour effectuer le transfert des paramètres et les appels à distance eux-mêmes.

En pratique cependant, les clients ne savent pas sur quel port le serveur RPC attend leurs requêtes. Un mécanisme a donc été mis en place pour leur permettre de récupérer ce port et communiquer ensuite avec le serveur. Chaque serveur RPC est identifié par un numéro de programme unique, ainsi qu'un numéro de version. Lorsqu'ils démarrent, les serveurs s'enregistrent dans le système en précisant le port sur lequel ils écouteront les requêtes. Les clients

peuvent alors interroger le système distant pour demander le port sur lequel ils trouveront un serveur donné, à partir du numéro de programme et du numéro de version de celui-ci.

Il existe donc un service RPC particulier, nommé « portmapper », qui fournit aux clients qui le demandent les numéros de port des autres serveurs. Bien entendu, le portmapper doit être toujours contactable, ce qui implique qu'il utilise systématiquement le même numéro de port. Par convention, le portmapper est identifié par le numéro de programme 100000, et il écoute les requêtes des clients sur les ports 111 des protocoles TCP et UDP.

La mise en place des serveurs de fichiers passe donc par le lancement de ces trois démons : le démon portmapper, le démon mountd et le démon nfsd. Les fichiers exécutables de ces démons sont respectivement `portmap`, `rpc.mountd` et `rpc.nfsd`. Ils sont normalement placés dans le répertoire `/sbin/` ou `/usr/sbin/`. Vous devrez donc lancer ces trois démons sur la machine serveur de fichiers, il est probable que votre distribution fasse le nécessaire dans ses scripts de démarrage.

Note : Il est prévu d'intégrer les fonctionnalités de serveur NFS dans le noyau de Linux. Cependant, le serveur de fichiers du noyau n'est pas encore finalisé, et ne sera donc pas décrit ici.

Une fois les démons lancés, vous pourrez configurer les systèmes de fichiers exportés par votre serveur. Ces systèmes de fichiers sont en fait de simples répertoires que vous mettez à disposition de certaines machines. La liste de ces répertoires est placée dans le fichier de configuration `/etc/exports`. Chaque ligne de ce fichier caractérise un répertoire accessible par les autres machines du réseau. Ces lignes utilisent le format suivant :

```
répertoire machines
```

où `répertoire` est le chemin sur le répertoire à exporter, et `machines` est une liste de machines pouvant accéder à ce répertoire. Cette liste contient des noms de machines, des adresses IP ou des noms de réseaux, séparés par des espaces. Il est également possible d'utiliser les caractères génériques '?' et '*' afin de spécifier des groupes de machines.

Des options peuvent être utilisées pour chaque machine, à l'aide de la syntaxe suivante :

```
machine(options)
```

où `machine` est l'une des entrées de la liste de machines d'une ligne du fichier `exports`, et `options` est la liste des options pour cette entrée, séparées par des virgules. Les options les plus utilisées sont bien entendu `ro` et `rw`, qui permettent de fournir à cette machine ou à ce groupe de machine respectivement un accès en lecture seule et en lecture et écriture sur le répertoire.

Le problème fondamental de NFS est la sécurité. En effet, les fichiers sont exportés par défaut avec un identificateur d'utilisateur qui est celui qui possède le fichier sur la machine serveur. Or il est tout à fait possible que cet identificateur ne corresponde pas au même utilisateur sur tous les clients. Cela signifie que les utilisateurs des machines clientes peuvent parfaitement avoir accès à des fichiers qui ne leur appartiennent pas sur le serveur. Ce problème est fondamental, aussi faut-il le prendre en considération sérieusement.

NFS fournit plusieurs solutions pour assurer la sécurité sur les systèmes de fichiers exportés. La première, qui est aussi la plus restrictive, est d'attribuer les fichiers exportés à un utilisateur ne disposant de quasiment aucun droit (opération que l'on nomme souvent « squashing de l'utilisateur »). Cet utilisateur spécial est l'utilisateur « nobody », dont l'identificateur est 65534 par défaut. Ainsi, tous les clients accèdent à ces fichiers au nom de l'utilisateur nobody, et ne peuvent donc pas modifier les fichiers du serveur. Le squashing de l'utilisateur root est toujours réalisé par défaut, pour des raisons de sécurité évidentes.

La deuxième solution est nettement moins sûre. Elle nécessite de lancer le démon « `ugidd` » sur chaque machine client. Ce démon est appelé par le serveur NFS pour déterminer l'identificateur des utilisateurs du client à partir de leur nom. Ainsi, chaque fichier est exporté avec l'identificateur de l'utilisateur qui porte le même nom que celui qui possède le fichier accédé sur le serveur. Les problèmes de sécurité posés par cette solution sont énormes : rien ne garantit que deux utilisateurs distincts sur deux machines différentes ne puissent pas avoir le même nom d'une part,

et un attaquant potentiel peut utiliser le démon `ugidd` pour obtenir la liste des utilisateurs de la machine cliente d'autre part (ce qui constitue déjà la moitié du travail pour s'introduire dans le système de la machine cliente). Cependant, cette solution est très pratique pour les réseaux dont on contrôle chaque machine, à condition de restreindre l'accès au démon `ugidd` au serveur uniquement, par exemple en ayant recours à `tcpd`.

La troisième solution est de définir sur le serveur l'association entre les identificateurs des utilisateurs du serveur et les identificateurs du client, ce pour chaque machine cliente. Cette technique est sûre, mais nettement plus compliquée à mettre en œuvre.

Enfin, la dernière solution est d'utiliser les services d'information du réseau NIS (abréviation de l'anglais « Network Information Service »), qui permettent de définir un certain nombre d'informations de manière globale sur un réseau. En particulier, il est possible de centraliser la définition des utilisateurs et des mots de passe. Cette solution est très lourde à mettre en œuvre, puisqu'elle nécessite de configurer NIS au préalable. Elle n'est donc mise en place que sur les grands réseaux, et un particulier n'a pas de raisons d'y recourir (à moins de vouloir explorer ces mécanismes bien entendu). Nous n'en parlerons donc pas.

Toutes ces techniques peuvent être activées à l'aide d'options fournies dans la liste des options pour chaque machine déclarée dans le fichier de configuration `/etc/exports`. Les options utilisées sont décrites ci-dessous :

- l'option `all_squash` permet d'exporter tous les fichiers comme appartenant à l'utilisateur `nobody`. C'est l'option la plus sûre, mais aussi la plus restrictive ;
- l'option `map_daemon` permet d'utiliser le démon `ugidd`, qui doit être lancé sur les machines clientes et accessibles du serveur. L'accès aux répertoires exportés sera refusé si ce démon ne peut être contacté par le serveur NFS. C'est certainement la solution la plus pratique pour un particulier ;
- l'option `map_static=fichier` permet d'utiliser un fichier de correspondance des identificateurs des utilisateurs du serveur NFS sur les identificateurs des utilisateurs de la machine cliente.

Comme on peut le voir, cette dernière option permet de définir spécifiquement, pour chaque machine, la correspondance des identificateurs d'utilisateurs. Le fichier fourni en paramètre à l'option `map_static` contient un certain nombre de lignes, chacune définissant l'association entre les identificateurs d'utilisateurs de la machine distante et les identificateurs de ces utilisateurs sur le serveur. Ces lignes sont introduites à l'aide du mot clé `uid`. Il est également possible de donner les correspondances sur les groupes des utilisateurs avec le mot clé `gid` :

```
# Exemple de fichier de correspondance d'identificateurs :
#      client      serveur
uid  0-99         -
uid  500-1000     1000
gid  0-49         -
gid  50-100       1000
```

Dans l'exemple donné ci-dessus, les utilisateurs ayant un identificateur compris entre 0 et 99 inclus seront associés à l'utilisateur `nobody` (ils subissent le « squashing »). Il en est de même pour les groupes allant de 0 à 49. En revanche, les utilisateurs dont les identificateurs vont de 500 à 1000 sur la machine cliente se voient respectivement considérés comme les utilisateurs d'identificateur 1000 à 1500 par le serveur NFS. De même, les groupes d'identificateurs 50 à 100 sont considérés comme les groupes d'identificateurs 1000 à 1050.

L'exemple qui suit va permettre d'éclaircir un peu ces notions. Il montre comment les trois types de contrôle des identificateurs peuvent être mis en place, pour trois types de clients différents :

```
# Exemple de fichier /etc/exports :

# Répertoire /pub : accessible de tout le monde, avec les droits
# de l'utilisateur nobody :
/pub      (ro,all_squash)

# Répertoire /home : accessible de toutes les machines du réseau local,
# avec contrôle des utilisateurs effectué par le démon rpc.ugidd :
/home     *.monrezo.org(map_daemon)
```

```
# Répertoire /usr : accessible par la machine 192.168.5.2, avec mapping
# des utilisateurs :
/usr 192.168.5.2(map_static=/etc/nfs/192.168.5.2.map)
```

Le fichier `/etc/nfs/192.168.5.2.map` utilisé dans la troisième ligne définit la correspondance entre les utilisateurs de la machine 192.168.5.2 et le serveur NFS :

```
# Fichier de correspondance pour 192.168.5.2 :
#      client  serveur
uid    0-99    -
uid    500     507
gid    0-49    -
gid    50      50
```

Les informations stockées dans le fichiers `/etc/exports` sont prises en compte lors de l'exécution de la commande `exportfs`. Cette commande est généralement exécutée à chaque démarrage de la machine dans les scripts de démarrage du système, avec l'option `-a`. Cette option permet d'exporter tous les systèmes de fichiers référencés dans `/etc/exports`, tout comme la commande `mount` utilise cette option pour monter tous les systèmes de fichiers au démarrage.

À chaque modification du fichier `/etc/exports`, il est nécessaire de signaler cette modification au démon `mountd`. Cela peut être fait simplement à l'aide de la commande suivante :

```
exportfs -a
```

IX-G-2 - Configuration d'un client NFS

Une fois la configuration du serveur NFS effectuée, il ne reste plus qu'à configurer les postes clients. Bien entendu, la manière de réaliser cela dépend du système utilisé. Nous allons voir ici comment accéder à un répertoire exporté par un serveur à partir d'une machine fonctionnant sous Linux.

Du point de vue du client, les fichiers accédés par NFS sont considérés comme des fichiers normaux d'un système de fichiers classique. Ils sont donc gérés directement par le noyau, en tant que système de fichiers à part entière. L'utilisation de NFS est donc directe, pour peu que l'on ait activé les fonctionnalités NFS dans le noyau.

Encore une fois, vous devrez modifier la configuration du noyau si celui fourni avec votre distribution ne gère pas les systèmes de fichiers NFS (ce qui est fort peu probable). La seule fonctionnalité à activer, en plus de la gestion du réseau bien entendu, est l'option « NFS filesystem support » du menu « Network File Systems » (ce menu est un sous-menu du menu « Filesystems » si vous utilisez la configuration en mode texte).

Lorsque le noyau sera compilé et installé, il sera capable d'utiliser les systèmes de fichiers NFS nativement. Vous n'aurez donc pas besoin des démons `nfsd` et `mountd`, tout sera pris en charge par le noyau. Cependant, si le serveur exporte ses répertoires avec l'option `map_daemon`, il sera nécessaire de lancer le démon `ugidd` sur la machine cliente (le fichier exécutable se nomme `rpc.ugidd` et se trouve dans le répertoire `/usr/sbin/`). Il est conseillé de lancer ce démon à partir d'`inetd`, en l'encapsulant à l'aide du démon `tcpd`.

Le montage d'un système de fichiers NFS se fait classiquement, avec la commande `mount`. Il suffit ici de préciser le type de système de fichiers « `nfs` » avec l'option `-t`, et d'utiliser la syntaxe suivante pour le fichier spécial de périphérique à utiliser :

```
machine:répertoire
```

où `machine` est le nom du serveur NFS, et `répertoire` est le chemin absolu sur le répertoire exporté par cette machine et auquel on désire accéder. Ainsi, si la machine « `mon.serveur.nfs` » exporte le répertoire « `/mon/répertoire/exporté` », la commande suivante permettra de monter ce système de fichiers NFS dans le répertoire `/mnt` :

```
mount -t nfs \
```

```
mon.serveur.nfs:/mon/répertoire/exporté /mnt
```

Le système de fichiers NFS accepte des options permettant d'optimiser les transferts d'informations. Ces options peuvent être fournies en ligne de commande à mount à l'aide de l'option -o. Les plus utiles sont sans doute rsize, qui permet de fixer la taille des blocs de données transférés pour la lecture des fichiers, et wsize, qui permet de fixer cette taille pour l'écriture des fichiers. Il est recommandé d'utiliser des paramètres raisonnables afin d'éviter des erreurs de transfert et des pertes de données. La valeur par défaut est 1024, mais il est recommandé d'utiliser des blocs de taille 8192 pour obtenir de meilleures performances. Ainsi, la commande précédente pourra être optimisée comme suit :

```
mount -t nfs -o rsize=8192,wsize=8192 \  
mon.serveur.nfs:/mon/répertoire/exporté /mnt
```

Bien entendu, ces valeurs dépendent de la bande passante de votre réseau, et vous aurez sans doute à effectuer des tests de transferts de fichiers pour trouver les valeurs optimales.

IX-G-3 - Installation d'un serveur de fichiers SMB

Bien que les protocoles SMB et NFS soient profondément différents, les principes utilisés pour mettre en place un serveur SMB sont quasiment les mêmes que ceux utilisés pour un serveur NFS. L'installation d'un serveur SMB requiert en effet l'utilisation de deux démons qui prennent en charge les services fournis par les serveurs de fichiers SMB. Ces démons utilisent tous deux le fichier de configuration `smb.conf`, placé le plus souvent dans le répertoire `/etc/`. Seuls les serveurs doivent faire fonctionner ces démons, les clients Linux quant à eux se contentent d'utiliser le système de fichiers SMB du noyau.

Ces démons, ainsi que tous les outils nécessaires à l'utilisation du protocole SMB, sont fournis dans la suite logicielle « Samba », dont le nom est évidemment inspiré de « SMB ». Cette suite logicielle est distribuée sous la licence GNU et est donc complètement libre d'utilisation. Elle est normalement fournie avec toutes les distributions. La version actuelle est la 3.0.3. Si ce n'est pas celle fournie avec votre distribution, il est fortement recommandé d'effectuer une mise à jour.

L'installation de Samba ne pose pas de problème particulier. Nous ne la décrivons donc pas ici. Sachez cependant qu'elle est recommandée même sur les postes clients, car bien que les démons ne soient pas lancés sur ceux-ci, les outils permettant de se connecter à un serveur de fichiers sont fournis avec Samba et restent nécessaires.

Comme on l'a vu ci-dessus, le protocole SMB se base sur le protocole de bas niveau NetBIOS. NetBIOS est un protocole très primitif, qui n'utilise pas un mécanisme d'adressage semblable à celui de la plupart des protocoles modernes. Chaque machine est identifiée sur le réseau par un nom unique, qu'elle doit déclarer avant de pouvoir utiliser les ressources du réseau. Tout nouvel arrivant sur le réseau signale donc aux autres qu'il désire utiliser son nom, afin de déterminer si ce nom est déjà pris ou non par une autre machine.

L'un des plus gros défauts de NetBIOS est tout simplement qu'il n'est pas routable tel quel, et qu'il est impossible de structurer un réseau en se basant seulement sur les noms NetBIOS. C'est pour cette raison que l'on encapsule souvent NetBIOS dans un protocole plus évolué capable de traverser les routeurs, comme TCP/IP par exemple. C'est exactement ce que fait Samba, qui apparaît donc comme un service réseau Unix classique permettant de comprendre les paquets NetBIOS encapsulés dans les paquets TCP/IP. Cette technique suppose évidemment que tous les clients du réseau, y compris les postes Windows, soient configurés pour encapsuler NetBIOS dans TCP/IP. L'utilisation de TCP/IP est de toute manière fortement recommandée si l'on veut réaliser un réseau un tant soit peu sérieux, et les versions récentes de Windows utilisent également cette technique.

Outre les noms d'ordinateurs, le protocole NetBIOS permet de définir des groupes d'ordinateurs, qui contiennent tous les ordinateurs supposés échanger des informations couramment. Un ordinateur ne peut faire partie que d'un seul groupe, mais il peut également ne faire partie d'aucun groupe s'il le désire. Ces groupes d'ordinateurs sont créés uniquement en fonction du travail qu'ils ont à effectuer, et ne sont a priori absolument pas liés à leur emplacement géographique ni à leur emplacement sur le réseau. Ainsi, il est possible de communiquer avec tous les ordinateurs d'un groupe, un peu comme le protocole TCP/IP permet d'utiliser des adresses de diffusion. Dans le monde Windows, ces groupes de machines sont plus couramment dénommés des « workgroup », et il n'y a en fait aucune différence

technique. Pour le protocole SMB, chaque ordinateur du réseau doit faire partie d'un groupe bien défini, bien que ce ne soit pas une nécessité au niveau de NetBIOS.

Lorsqu'un groupe de travail possède une base de données centrale permettant de réaliser l'authentification des utilisateurs lors des accès aux machines par le réseau, on dit qu'il s'agit d'un domaine. En pratique, un domaine n'est donc rien d'autre qu'un groupe de travail dont font partie un ou plusieurs serveurs d'authentification. Il existe plusieurs types de serveurs de domaine. Un serveur primaire est un serveur central, sur lequel se fait la gestion globale des utilisateurs. Outre un serveur primaire, un domaine peut également contenir un ou plusieurs serveurs de sauvegarde si nécessaire. Comme leur appellation l'indique, ces serveurs sont capables de remplacer le serveur primaire en cas de défaillance de celui-ci. Ils maintiennent donc une copie des tables d'utilisateurs, et la synchronisent régulièrement avec celle du serveur primaire. Enfin, un domaine peut contenir des serveurs membres, qui sont comme des serveurs de sauvegarde, à ceci près qu'ils ne peuvent pas remplacer le serveur de domaine primaire. Bien entendu, le domaine contient également les stations de travail classiques, qui effectuent des requêtes sur le serveur primaire.

Afin de gérer la liste des noms NetBIOS du réseau, Samba fournit le démon « nmbd ». Ce démon est essentiel au fonctionnement correct de Samba et doit donc toujours être lancé. Il peut également être configuré en tant que serveur WINS (abréviation de l'anglais « Windows Internet Name Server »), afin de pouvoir fournir les noms NetBIOS sur Internet. WINS est une technique développée par Microsoft pour résoudre le problème du nommage des postes de travail sur Internet. Un serveur WINS est donc à NetBIOS un peu ce qu'un serveur DNS est à TCP/IP. Le démon nmbd est installé classiquement dans le répertoire `/usr/bin/` ou dans le répertoire `/usr/sbin/`. Vous devrez le lancer avec l'option `-D`, faute de quoi il ne démarrera pas en tant que démon :

```
/usr/sbin/nmbd -D
```

Le deuxième démon fourni par Samba est « smbd ». Celui-ci gère effectivement le protocole SMB, et est donc en charge d'effectuer la gestion des utilisateurs et des partages. Il doit être lancé également avec l'option `-D` si l'on veut qu'il démarre en tant que démon. Ce démon est normalement installé au même endroit que nmbd :

```
/usr/sbin/smbd -D
```

Vous pourrez éventuellement faire en sorte que ces démons soient lancés automatiquement en modifiant les scripts de démarrage de votre système. Une fois ces deux démons lancés, votre machine Linux se comportera exactement comme une machine Windows serveur classique. Ainsi, les ressources partagées par ce serveur seront accessibles par les clients SMB, comme les postes Windows par exemple.

Malheureusement, la plupart des clients SMB ne gèrent pas les notions avancées des systèmes de fichiers Unix. Le protocole SMB ne supporte d'ailleurs qu'une partie infime de ces fonctionnalités. De plus, la gestion des droits sur les fichiers exportés par un serveur de fichiers Windows se base généralement sur le modèle de sécurité des d'ACLs (« Access Control List »), plutôt que sur le modèle de sécurité Unix classique. Samba peut donc être configuré pour utiliser le support des ACLs natif des systèmes de fichiers Linux, ainsi que pour utiliser les modèles de sécurité primitifs des anciennes versions de Windows.

La première difficulté à résoudre concerne la gestion des droits d'accès aux fichiers des partages. Sous Unix, chaque fichier et chaque répertoire disposent de droits d'accès, et seuls les utilisateurs autorisés peuvent les manipuler. Dans le monde Windows, il n'y a quasiment pas de notion d'utilisateur, et en général pas de gestion de la sécurité. Seuls Windows NT/2000/XP disposent de ces fonctionnalités, à condition qu'ils utilisent des systèmes de fichiers NTFS et que le type de partage utilisé le permette.

En fait, la gestion de la sécurité se fait de manière globale, pour tous les fichiers d'un partage. Il existe principalement deux modes de contrôle d'accès aux partages SMB :

- le contrôle d'accès au niveau ressource, qui se fait à partir d'un mot de passe unique pour la ressource, sans tenir compte de l'utilisateur qui cherche à y accéder ;
- le contrôle d'accès au niveau utilisateur, qui nécessite de fournir un nom d'utilisateur et un mot de passe.

Le contrôle d'accès au niveau utilisateur nécessite de se trouver dans un domaine, et non dans un simple Workgroup. Il faut donc disposer d'un serveur de domaine afin de réaliser l'authentification des utilisateurs qui cherchent à accéder à ces ressources. Pour des raisons commerciales, Microsoft s'est arrangé pour que seuls Windows NT, Windows 2000 et Windows XP serveur puissent servir de contrôleurs de domaine, ce qui fait qu'il est impossible d'utiliser le contrôle d'accès au niveau utilisateur sur un réseau ne disposant que de machines sous Windows 95, 98, ou NT/2000/XP client (heureusement, Samba est capable de servir de contrôleur de domaine, cependant, nous ne verrons pas ce type de configuration). Il va de soi que le contrôle d'accès au niveau utilisateur est la seule solution réellement sûre du point de vue de la sécurité.

Pour couronner le tout, les partages ne permettent que de fixer les droits de lecture et d'écriture pour celui qui accède aux fichiers du partage. Les droits d'exécution n'ont pas de signification sous Windows puisque, contrairement aux fichiers Unix, le type des fichiers est déterminé par leur extension. Les notions de sécurité des partages sont donc profondément différentes de celles utilisées par Unix, surtout pour le contrôle d'accès au niveau ressource, puisqu'aucun nom d'utilisateur n'est requis lors de l'authentification ! Mais ce n'est pas tout. Les fichiers Unix appartiennent tous à un utilisateur et à un groupe d'utilisateurs, chose qui n'a pas de signification pour Windows. De plus, les systèmes de fichiers Unix font la distinction entre les majuscules et les minuscules dans les noms de fichiers, ce que ne fait pas Windows. Et ils disposent de la notion de liens physiques et symboliques. Autant de problèmes qui doivent être gérés par Samba.

Note : Le fait de ne pas disposer des informations concernant les droits des utilisateurs sur les fichiers accédés par SMB n'empêche pas le serveur de fichiers de contrôler les accès effectués par le client. Si par exemple, un fichier est marqué comme étant en lecture seule, aucun client ne peut y écrire, même par l'intermédiaire d'un partage accédé en lecture et écriture.

Pour résoudre tous ces problèmes, Samba dispose d'un grand nombre d'options, qui peuvent être utilisées dans le fichier de configuration `/etc/samba/smb.conf`. Ces options permettent d'indiquer la manière dont l'authentification doit être faite, les droits avec lesquels les fichiers sont accédés ou créés, les utilisateurs au nom desquels les fichiers sont créés, et le comportement à suivre lorsqu'un client essaie de suivre un lien symbolique. Seules les principales options gérées par Samba seront décrites ici. Les options avancées seront laissées de côté, car Samba propose un nombre incroyable de fonctionnalités et les traiter toutes dépasserait le cadre de ce document. Vous pourrez trouver des informations complémentaires en cas de nécessité dans la documentation de Samba, ou bien dans l'excellent livre « Using Samba » de Robert Eckstein, publié aux éditions O'Reilly. Une version en ligne de ce livre est fournie avec les sources de Samba et complète ainsi la documentation officielle de ce programme.

La plupart des distributions fournissent un fichier de configuration `/etc/samba/smb.conf` contenant la plupart des options par défaut, plus quelques exemples de configuration. Par exemple, les répertoires des utilisateurs sont souvent exportés par défaut. Vous pourrez bien entendu modifier ce fichier de configuration. Il est donc peut-être nécessaire de donner quelques précisions.

Le fichier `smb.conf` est structuré en sections permettant de fixer les paramètres de différentes fonctionnalités du logiciel. Chaque section est introduite par son nom, donné entre crochets. Certaines sections sont spécifiques à Samba et d'autres permettent de définir les partages que vous désirez créer. Les paramètres définis dans les sections se présentent sous la forme de couple « paramètre = valeur », où paramètre est le nom d'un des paramètres possibles dans cette section, et valeur est la valeur que ce paramètre peut prendre.

Parmi les sections propres à Samba, on compte la section « `[global]` ». Cette section est particulièrement importante, car elle fixe les paramètres globaux de Samba, ainsi que les valeurs par défaut des paramètres des autres sections. La section `[global]` suivante peut servir d'exemple :

```
[global]
# Définit le nom NetBIOS du serveur :
    netbios name = Linux
    server string = Linux fait la Samba (version %v)

# Définit le nom du groupe de travail :
    workgroup = monrezo

# Fixe les règles de gestion de la sécurité :
    security = user
```

```
encrypt passwords = yes
smb passwd file = /etc/samba/passwd

# Définit la correspondance entre les noms d'utilisateurs Windows
# et les noms Unix :
username map = /etc/samba/usermap

# Définit le compte Unix à utiliser pour invité :
guest ok = no
guest account = nobody

# Définit les paramètres réseau :
socket options = TCP_NODELAY
interfaces = eth0
bind interfaces only = yes
```

Cette section commence par définir le nom NetBIOS du serveur et sa description. Par défaut, le nom utilisé par Samba est le nom de domaine Unix de la machine, mais vous pouvez changer ce nom comme bon vous semble. Vous avez dû remarquer que dans la description de la machine (mot clé `server string`), la version de Samba est récupérée avec `%v`. En fait, Samba définit un certain nombre de variables qui sont remplacées dynamiquement lors de la connexion à un partage. Ainsi, `%v` représente la version de Samba utilisée, `%u` représente le nom de l'utilisateur qui se connecte, etc. Toutes ces variables sont décrites dans la page de man du fichier `smb.conf`.

Le nom du groupe de travail dont fait partie la machine est ensuite introduit avec le mot clé `workgroup`. Dans le cas présent, il s'agit du groupe de travail « monrezo ».

Viennent ensuite les options de sécurité. Le mot clé `security` permet de définir le mode de contrôle d'accès aux partages mis à disposition des clients par le serveur Samba. La valeur `user` indique que ce contrôle d'accès se fait au niveau utilisateur, ce qui est le plus cohérent sous Unix. Chaque utilisateur doit donc fournir son nom et son mot de passe pour accéder à ces partages. Samba peut également gérer le contrôle d'accès au niveau ressource, si l'on utilise la valeur `share`. Dans ce cas, Samba essaiera de retrouver l'utilisateur qui cherche à se connecter à partir de son mot de passe. Il faut donc ajouter dans ce cas une ligne telle que celle-ci dans la section globale ou dans l'une des sections définissant un partage :

```
username = utilisateurs
```

où `utilisateurs` représente la liste des utilisateurs que Samba utilisera pour tester le mot de passe fourni. Bien entendu, si un nom d'utilisateur est fourni par le client pour accéder à cette ressource, ce nom est utilisé directement. Samba mémorisera également ce nom pour les demandes d'accès ultérieures à la ressource partagée. Cet algorithme n'est pas très sûr, car un utilisateur peut se connecter par hasard au nom d'un autre. Aussi n'est-il pas recommandé d'utiliser le contrôle d'accès au niveau ressources.

Jusqu'à Windows 95 et Windows NT4 Service Pack 2 compris, les mots de passe fournis par les clients étaient transférés en clair sur le réseau. Cela n'étant pas sûr, Microsoft a décidé d'adopter une nouvelle technique, dans laquelle les mots de passe à transférer sont chiffrés par une clef fournie par le serveur. Ainsi, une personne mal intentionnée écoutant les paquets transférés sur le réseau ne pourrait pas capter ces mots de passe. Samba est capable de gérer les deux cas de configuration, à l'aide de l'option `encrypt password`. Cependant, si vous décidez d'activer le chiffrement des mots de passe, vous devrez également créer un fichier de mots de passe pour les clients SMB. L'emplacement de ce fichier est indiqué avec l'option `smb passwd file`.

Le format du fichier de mot de passe de Samba ressemble fortement à celui du fichier de mot de passe Unix `/etc/passwd`. Vous pourrez trouver dans la documentation de Samba une description détaillée de ce fichier. Cependant, la chose la plus importante, c'est bien sûr de pouvoir ajouter des entrées pour chaque utilisateur dans ce fichier. Cette opération se fait avec l'utilitaire `smbpasswd`. Exécutée sous le compte `root`, la commande suivante permet d'ajouter un utilisateur et de définir son mot de passe :

```
smbpasswd -a utilisateur
```

où `utilisateur` est le nom de l'utilisateur dont on veut créer un nouveau mot de passe. L'utilisateur peut bien entendu changer son mot de passe avec `smbpasswd`.

Il est possible que les noms des utilisateurs Windows ne soient pas identiques à leurs logins sur la machine serveur de fichiers. En fait, il est même possible que plusieurs utilisateurs utilisent un même compte, créé uniquement pour les partages SMB. Samba fournit donc la possibilité d'utiliser un fichier de correspondance entre les noms des utilisateurs avec l'option `username map`. Le format du fichier de correspondance est très simple, puisqu'il est constitué de lignes définissant chacune une association entre un nom de login Unix et un nom d'utilisateur Windows. Ces lignes utilisent la syntaxe suivante :

```
login = utilisateur
```

où `login` est le nom de login de l'utilisateur sur la machine Unix, et `utilisateur` est le nom qu'il utilisera pour accéder aux partages.

L'option `guest ok` permet d'autoriser les connexions sans mot de passe sous un compte spécial, que l'on nomme le compte invité. Pour cela, il suffit de lui affecter la valeur `yes`. Cette option peut être fixée à `no` dans la section de configuration globale, et redéfinie dans les sections spécifiques de certains partages. Dans tous les cas, les connexions qui s'effectuent en tant qu'invité doivent utiliser un compte utilisateur Unix classique. Ce compte peut être défini à l'aide de l'option `guest account`, en général, l'utilisateur `nobody` utilisé par NFS est le plus approprié.

Enfin, il est possible de définir des paramètres réseau dans la section de configuration globale. Ces paramètres réseau permettent de fixer des options de sécurité complémentaires et des options d'optimisation. L'option `interfaces` donne la liste des interfaces à partir desquelles les demandes de connexion des clients seront acceptées. Ce type de contrôle est activé lorsque le paramètre `bind interfaces only` prend la valeur `yes`. Ces options conviennent parfaitement pour un réseau local de particulier. L'option `socket options` quant à elle permet de définir les paramètres de fonctionnement des communications TCP. L'option `TCP_NODELAY` indique au système d'envoyer les paquets TCP dès que Samba le demande, sans chercher à en regrouper plusieurs pour optimiser la taille des paquets. Cette option accélère significativement le fonctionnement des programmes tels que Samba, qui utilisent beaucoup de petits paquets pour envoyer des requêtes ou des réponses. Sans cette option, le système chercherait à regrouper ces paquets, même si Samba n'en a pas d'autres à envoyer, et ralentirait ainsi inutilement les transferts de données.

Chaque partage fourni par le serveur SMB dispose également d'une section, dont le nom est le nom utilisé pour ce partage. Ces sections sont beaucoup plus simples que la section global, comme le montre l'exemple suivant :

```
[Données]
# Donne le chemin sur le répertoire utilisé par ce partage :
  path = /usr/share/samba/données

# Description du partage :
  comment = Disque de données

# Nom de volume Windows :
  volume = SMB-LNX

# Options d'accès :
  writeable = yes
  guest ok = yes
```

Comme vous pouvez le constater, l'option `path` permet de définir le répertoire utilisé pour stocker les fichiers du partage. Ce répertoire est un répertoire du serveur, et le chemin doit donc obligatoirement être un chemin Unix valide. En particulier, il faut tenir compte ici de la casse des noms de fichiers. L'option `comment` donne la description du partage, telle qu'elle apparaîtrait dans le voisinage réseau de Windows. Le nom de volume quant à lui est celui qui sera donné si un client Windows attache une lettre de lecteur à ce partage. Il est spécifié à l'aide de l'option `volume`. Enfin, les options d'accès utilisées dans cet exemple permettent d'autoriser l'écriture dans les fichiers de ce partage, et d'autoriser les accès en tant qu'invité (c'est-à-dire sans mot de passe).

Samba dispose d'une fonctionnalité intéressante afin de définir automatiquement un partage pour chacun des répertoires personnels des utilisateurs déclarés dans le serveur. Lorsqu'un client cherche à accéder à un partage qui

n'est pas défini par une section qui lui est propre, Samba tente de le localiser un répertoire personnel d'utilisateur portant ce nom. S'il le trouve, il utilisera ce répertoire pour effectuer le partage automatiquement. Les paramètres de ce partage automatique sont définis dans la section `[homes]`, dont vous trouverez un exemple ci-dessous :

```
[homes]
comment = Répertoires personnels
browsable = no
writable = yes
```

L'utilisation de l'option `browsable = no` ici sert simplement à indiquer qu'il ne doit pas apparaître de partage nommé `homes`. En effet, ce qui est désiré ici, c'est d'exposer les noms des répertoires personnels des utilisateurs, pas le partage `homes` lui-même. Les autres options sont les mêmes que pour les partages normaux.

La section `[homes]` peut poser quelques problèmes, puisqu'elle partage également le compte `root`. Cela n'est certainement pas le comportement désiré. Il est même fortement recommandé d'interdire les connexions pour les utilisateurs privilégiés du système. Cela peut être réalisé en utilisant l'option `invalid users` :

```
invalid users = root bin daemon adm sync shutdown \
halt mail news uucp operator gopher
```

En fait, il est conseillé de placer cette option dans la section `[global]`, afin d'interdire les connexions de ces utilisateurs sur tous les partages.

Enfin, Samba fournit une fonctionnalité comparable à la section `[homes]` pour les imprimantes partagées. Il est en effet capable d'analyser le fichier de configuration `/etc/printcap` pour déterminer la liste des imprimantes installées, et les partager pour les clients Windows. Ces imprimantes apparaîtront donc comme des imprimantes partagées classiques. Pour cela, il suffit d'ajouter les lignes suivantes dans la section `[global]` du fichier `smb.conf` :

```
[global]
# Définit le type d'impression utilisé (bsd ou cups) :
printing = bsd

# Lit la définition des imprimantes installées dans /etc/printcap
# (type de commande BSD uniquement) :
printcap name = /etc/printcap
load printers = yes
```

La première ligne indique à Samba le type de commande à utiliser pour réaliser une impression. Si vous utilisez le système d'impression BSD, vous devez fixer l'option `printing` à `bsd`. Si vous utilisez CUPS, la valeur à utiliser est `cups`.

Les options concernant toutes les imprimantes que Samba trouvera peuvent être précisées dans la section `[printers]`. Cette section joue donc le même rôle pour les imprimantes partagées que la section `[homes]` joue pour les répertoires personnels des utilisateurs. La section donnée ci-dessous pourra vous servir d'exemple :

```
[printers]
comment = Imprimantes
browseable = no
printable = yes
writeable = no
guest ok = no
path = /tmp
create mode = 0700
```

L'option `browseable` a ici la même signification que dans la section `[homes]`. On ne désire en effet pas qu'un partage portant le nom `printers` apparaisse dans les voisinages réseau des machines Windows ! En revanche, l'option `printable` permet d'indiquer clairement que ces partages sont des imprimantes. Il est évident que l'on ne peut pas écrire sur une imprimante, l'option `writeable` est donc fixée à `no`. Enfin, les impressions ne sont autorisées que pour les utilisateurs identifiés, et l'utilisateur invité n'a pas le droit de soumettre un travail d'impression.

Le mécanisme utilisé par Samba pour imprimer un document est le suivant. Lorsqu'un client demande une impression sur une des imprimantes partagées, Samba copie le fichier que le client lui envoie en local. Ce fichier est ensuite communiqué au gestionnaire d'impression Unix, en l'occurrence lpr sous Linux. Celui-ci imprime le fichier et le supprime du disque. Dans la section [printers], l'option `path` permet d'indiquer dans quel répertoire les fichiers temporaires envoyés par les clients doivent être stockés. Il est tout à fait cohérent de les placer dans le répertoire `/tmp` comme dans l'exemple donné ci-dessus. Enfin, il est logique de protéger ces fichiers contre les autres utilisateurs. Cela est réalisé à l'aide de l'option `create mode`, qui fixe les droits de ces fichiers à 0 pour tout le monde, sauf pour le compte root, à leur création. Ainsi, seuls les programmes fonctionnant sous le compte root (donc Samba et la commande lpr) peuvent lire, écrire et effacer ces fichiers.

Il est bien entendu possible de définir les imprimantes manuellement, à raison d'une section par imprimante. Ce type de configuration ne sera toutefois pas décrit ici.

Toute modification du fichier de configuration de Samba implique de la signaler aux démons `smbd` et `nmbd`. Cela peut être réalisé en leur envoyant un signal `SIGHUP`. Vous devrez donc taper les commandes suivantes :

```
killall -HUP smbd
killall -HUP nmbd
```

Comme on peut le voir, le fichier de configuration `smb.conf` n'est pas très compliqué, mais utilise un grand nombre d'options. Heureusement Samba fournit l'outil de configuration « SWAT » (abréviation de l'anglais « Samba Web Administration Tool »), qui permet d'effectuer la configuration de Samba graphiquement. Comme son nom l'indique, SWAT joue le rôle de serveur Web, et peut être utilisé avec n'importe quel navigateur en se connectant sur le port 901 de la machine sur laquelle il tourne.

Pour que cela puisse fonctionner, il faut ajouter le service réseau « `swat` » dans votre fichier de configuration `/etc/services` :

```
swat      901/tcp
```

Vous devrez également ajouter une ligne dans votre fichier `/etc/inetd.conf`, afin que le démon `inetd` puisse lancer SWAT automatiquement lorsqu'un client cherche à l'utiliser sur le réseau. Vous devrez donc ajouter la ligne suivante :

```
swat stream tcp nowait.400 root /usr/local/samba/bin/swat swat
```

et signaler à `inetd` la modification de `inetd.conf` à l'aide de la commande :

```
killall -HUP inetd
```

Vous pourrez alors accéder à la configuration graphique de SWAT à l'aide d'un navigateur Web, en utilisant simplement l'URL suivante :

```
http://localhost:901
```

SWAT demandera bien entendu le nom de l'utilisateur root et son mot de passe pour permettre la modification du fichier `smb.conf`. Notez qu'il est fortement déconseillé de laisser la possibilité de réaliser une connexion Web sur le compte root, et que vous ne devriez pas autoriser les connexions non locales sur le port TCP 901. L'utilisation de SWAT ne pose pas vraiment de problèmes et ne sera donc pas décrite plus en détail ici.

IX-G-4 - Configuration d'un client SMB

L'utilisation des volumes partagés par l'intermédiaire du protocole SMB sous Linux est très simple. Elle se fait exactement comme pour NFS : il suffit simplement de monter le volume dans un répertoire vide de l'arborescence de votre système de fichiers.

Pour que cela puisse être réalisable, il faut bien entendu que le noyau supporte le système de fichiers SMB. Ce système de fichiers est considéré par le noyau exactement comme le système de fichiers NFS : il s'agit d'un système de fichiers réseau. Vous pourrez donc activer le support de ce système de fichiers à l'aide de l'option de configuration « SMB filesystem support (to mount WfW shares etc.) », que vous trouverez normalement dans le menu « Network File Systems » (ce menu est un sous-menu du menu « Filesystems » si vous utilisez le programme de configuration en mode texte). La compilation du noyau et son installation ont été décrites en détail dans la [Section 7.3](#).

Lorsque vous aurez configuré et installé ce noyau, vous pourrez monter les volumes partagés par des serveurs de fichiers SMB (qu'il s'agisse de serveurs Linux ou Windows) exactement comme n'importe quel système de fichiers, et accéder à leurs ressources. Malheureusement, la commande `mount` du système ne supporte pas encore complètement le système de fichiers SMB. Elle utilise donc un programme complémentaire nommé `smbmount`, fourni avec la distribution de Samba. Notez bien que ce programme ne fait pas officiellement partie de Samba, il est seulement fourni en tant qu'utilitaire complémentaire pour Linux. Bien qu'il fasse également partie d'un paquetage indépendant, il est recommandé d'utiliser la version fournie avec Samba. Cela signifie que vous devrez installer Samba sur les postes clients, même si vous ne lancez pas les démons `smbd` et `nmbd`. La commande standard `mount` du système sera peut-être modifiée un jour pour intégrer toutes les fonctionnalités de la commande `smbmount`, ce qui rendra inutile l'installation de Samba sur les postes clients.

Quoi qu'il en soit, la syntaxe de la commande `smbmount` est très simple :

```
smbmount partage répertoire
```

où `partage` est le nom de partage du volume à monter, et `répertoire` est son point de montage (par exemple, `/mnt/`). Le nom de partage utilisé est exactement le même nom que celui utilisé par Windows, à ceci près que les antislashes (`\`) sont remplacés par des slashes (`/`). Ainsi, pour monter dans `/mnt/` le partage Données du serveur de fichiers SMBSERVER, vous devrez utiliser la commande suivante :

```
smbmount //smbserver/données /mnt
```

Remarquez que le protocole SMB ne fait pas la distinction entre les majuscules et les minuscules (tout comme Windows d'une manière générale), et que vous pouvez utiliser indifféremment les majuscules ou les minuscules.

L'utilisation des slashes à la place des antislashes dans le nom du partage est due au fait que l'antislash est un caractère spécial que shell interprète comme étant la poursuite de la commande courante sur la ligne suivante. Vous pouvez utiliser des antislashes si vous le désirez, mais dans ce cas, vous devrez mettre le nom de partage entre guillemets, pour empêcher le shell de les interpréter. Malheureusement, l'antislash est également un caractère d'échappement dans les chaînes de caractères, et est utilisé pour introduire des caractères spéciaux. Étant lui-même un caractère spécial, il doit lui aussi être précédé d'un antislash d'échappement ! Il faut donc doubler tous les antislashes, et la commande précédente devient particulièrement longue et peu pratique avec cette syntaxe :

```
smbmount "\\\\"smbserver\\"données" /mnt
```

Il est très facile de faire des erreurs avec de telles commandes, aussi je ne vous la conseille pas.

En fait, il est possible de faire en sorte que la commande `mount` classique du système puisse être utilisée pour réaliser les montages de partages SMB à partir de la version 2.0.6. ou plus de Samba. Pour cela, il suffit de créer dans le répertoire `/sbin/` un lien symbolique `mount.smbfs` vers la commande `smbmount`. Lorsqu'on utilisera la commande `mount` avec le type de système de fichiers `smbfs`, celle-ci utilisera ce lien et appellera ainsi automatiquement `smbmount`. Les paramètres fournis à `mount` seront transmis tels quels à `smbmount`. En particulier, vous pourrez utiliser une commande de ce type :

```
mount -t smbfs partage répertoire
```

où `partage` et `répertoire` sont toujours les noms de partage et de répertoire devant servir de point de montage.

Une fois le partage monté, vous pouvez l'utiliser comme un système de fichiers classique. Lorsque vous aurez fini de l'utiliser, vous pourrez simplement le démonter, avec la commande `smbmount` :

```
smbmount /mnt
```

Il est également possible d'utiliser la commande système classique `umount`, mais cela nécessite de repasser sous le compte `root`, ou de fixer le bit `setuid` sur l'exécutable de `umount`, ce qui est un énorme trou de sécurité. La commande `smbmount` en revanche peut être `setuid`, car elle vérifie que l'utilisateur ne cherche à démonter que les partages qu'il a lui-même montés.

Les partages au niveau ressource ne prennent pas en compte la notion d'utilisateur et de groupe d'utilisateurs des systèmes de fichiers Unix classiques. Par conséquent, le propriétaire et le groupe des fichiers accédés par un partage de ressources sont fixés par défaut à l'utilisateur qui a effectué le montage et à son groupe. Cela est naturel et ne pose pas de problèmes pour la plupart des utilisateurs, mais peut être gênant lorsqu'on monte un partage en tant que `root` pour le compte d'un autre utilisateur. Pour résoudre ce problème, il est possible de préciser le nom de l'utilisateur et son groupe à l'aide des options `uid` et `gid` de la commande `smbmount` :

```
smbmount partage répertoire -o uid=utilisateur,gid=groupe
```

où `utilisateur` et `groupe` sont respectivement les noms ou les numéros de l'utilisateur et du groupe auquel les fichiers devront être attribués. Les paramètres `uid` et `gid` peuvent également être utilisés avec la commande `mount`. Dans ce cas, ils sont passés tels quels à `smbmount`.

Vous l'aurez sans doute remarqué, la commande `smbmount` vous demande de taper un mot de passe pour accéder au partage. Ce mot de passe est le mot de passe attendu par le serveur de fichiers SMB. Pour Windows, il s'agit du mot de passe attribué au partage. Pour les serveurs de fichiers Samba, il s'agit du mot de passe de l'utilisateur au nom duquel se fait le partage, ou du mot de passe SMB enregistré dans le fichier de mots de passe `smbpasswd` de Samba. Dans tous les cas, ce mot de passe est demandé, même s'il est vide. La commande `smbmount` peut accepter une option `password`, qui permet de préciser ce mot de passe. Cependant, il est fortement déconseillé de l'utiliser, et ce pour deux raisons. Premièrement, le mot de passe apparaît en clair lorsqu'il est saisi, et la ligne de commande peut être vue par n'importe quel utilisateur avec la commande `ps`. Cela pose donc un problème de sécurité évident. Deuxièmement, l'option `password` n'est pas reconnue par la commande `mount`, et ne peut donc être utilisée qu'avec `smbmount`. C'est pour cela qu'une autre solution a été proposée, bien qu'encore imparfaite : définir le nom d'utilisateur et le mot de passe dans la variable d'environnement `USER`. Pour cela, il suffit d'utiliser l'une des commandes suivantes :

```
export USER=utilisateur%secret
export USER=utilisateur/workgroup%secret
```

où `utilisateur` est le nom d'utilisateur classique, `workgroup` est le groupe de travail et `secret` le mot de passe sur le partage à monter. Cette technique n'est pas non plus très sûre, et n'est en aucun cas pratique.

Une autre solution, plus sûre, est de définir les paramètres d'identification et d'authentification de l'utilisateur dans un fichier et de fournir ce fichier en paramètre à la commande `smbmount` à l'aide de l'option `credentials` :

```
smbmount partage répertoire -o credentials=fichier
```

où `fichier` est le fichier contenant l'identification de l'utilisateur. Ce fichier utilise la syntaxe suivante :

```
username = utilisateur
password = secret
```

où `utilisateur` est le nom de l'utilisateur et `secret` son mot de passe. Cette technique est la technique recommandée pour réaliser le montage de partages SMB directement dans le fichier de configuration `/etc/fstab`, car c'est la seule qui permette de ne pas avoir à saisir le mot de passe ni d'avoir à le définir en clair dans une ligne de commande.

X - Installation de XWindow

Ce chapitre décrit la manière de procéder pour installer et configurer X.org, une implémentation libre de XWindow dérivée du projet XFree86. Il indique également comment installer le serveur X pour le pilote de frame buffer du noyau. L'installation des polices Truetype, qui sont si chères aux utilisateurs de Windows et des Macintosh, est également traitée. En revanche, il ne décrira pas comment configurer les gestionnaires de fenêtres ni les gestionnaires de bureau, car ces opérations sont spécifiques à celui que vous choisirez d'une part, et spécifiques à vos desiderata d'autre part. Pour cela, vous devrez commencer à vous débrouiller tout seul, à lire les pages de manuel et à poser les questions qu'il faut aux personnes qu'il faut. Mais ne paniquez pas, si vous êtes arrivé jusqu'ici, c'est que vous commencez à savoir vous débrouiller. Vous ne devriez plus trop avoir de problèmes pour tirer de Linux tout ce dont vous avez besoin...

X-A - Généralités sur XWindow

Il est nécessaire de bien comprendre ce qu'est XWindow pour pouvoir le configurer et l'utiliser. Son architecture se distingue en effet fortement de celle des environnements graphiques des systèmes classiques comme Windows, OS/2 ou Macintosh, et ces différences se traduisent dans la manière de l'utiliser.

La principale différence entre XWindow et les autres environnements graphiques est qu'il s'agit d'un environnement graphique distribué sur un réseau. La notion de base de XWindow est que l'application qui effectue le traitement ne réalise pas elle-même la gestion de l'environnement graphique. Comme on l'a déjà vu, cette tâche est prise en charge par le serveur X, qui est un processus indépendant. L'application est donc cliente du serveur X, qui lui fournit les services graphiques dont elle a besoin. Cette séparation a plusieurs conséquences :

- premièrement, l'environnement graphique est isolé des fautes des applications qui l'utilisent. Ainsi, ce n'est pas parce qu'une application a planté en plein écran qu'on ne peut pas réduire sa fenêtre et accéder à nouveau au bureau sous-jacent. Inversement, une application est isolée des erreurs potentielles du serveur X, et peut poursuivre son traitement même si celui-ci s'est terminé. Par exemple, un processus de gravage de CD peut terminer le CD en cours même s'il a perdu son interface graphique ;
- deuxièmement, les clients doivent établir une connexion avec le serveur. Cette connexion est réalisée par l'intermédiaire du réseau. Cela implique naturellement que les mécanismes de sécurité liés au réseau sont applicables pour toutes les applications désirant se connecter au serveur X. XWindow fournit par ailleurs des mécanismes de sécurité complémentaires, et les ressources graphiques ne peuvent être utilisées que par les processus qui y sont autorisés ;
- enfin, comme le client et le serveur X sont deux processus distincts et qu'ils communiquent par l'intermédiaire d'une connexion réseau, rien n'interdit de lancer le client et le serveur X sur deux machines distinctes. Ainsi, tout processus déporté peut afficher ses données en local (et inversement).

Chaque client doit donc se connecter au serveur X avec lequel il désire travailler. En pratique, il n'existe souvent qu'un seul serveur X sur une machine, mais cela n'est pas une obligation. Par exemple, une même machine peut disposer de deux cartes graphiques et de deux écrans, et faire tourner deux serveurs X distincts. Une autre possibilité est d'utiliser les deux écrans avec un seul serveur X pour faire un écran virtuel beaucoup plus grand. Enfin, il est tout à fait concevable de lancer plusieurs fois un même serveur X, même si l'on ne dispose que d'une seule carte graphique et d'un seul écran, afin de pouvoir utiliser plusieurs terminaux X virtuels. Comme on le voit, l'architecture client/serveur de XWindow lui apporte une très grande flexibilité.

Les serveurs X utilisent la notion de display pour gérer l'affichage. En fait, un display est constitué d'un clavier, d'une souris et d'un ou plusieurs écrans. Le display est donc l'extension de la notion de terminal pour XWindow. Notez bien qu'il est possible d'utiliser plusieurs écrans sur le même terminal X. Cependant, un serveur X ne peut prendre en charge qu'un seul terminal X sur une machine, et chaque display est géré par un serveur X qui lui est propre. Si l'on veut utiliser plusieurs terminaux X sur une même machine, il est nécessaire de lancer plusieurs serveurs X, à raison d'un par terminal.

Les clients qui désirent se connecter à un serveur X doivent donc indiquer le display avec lequel ils désirent travailler. Le système XWindow se chargera d'établir la connexion avec le serveur X en charge de ce display. Les displays sont spécifiés avec la syntaxe suivante :

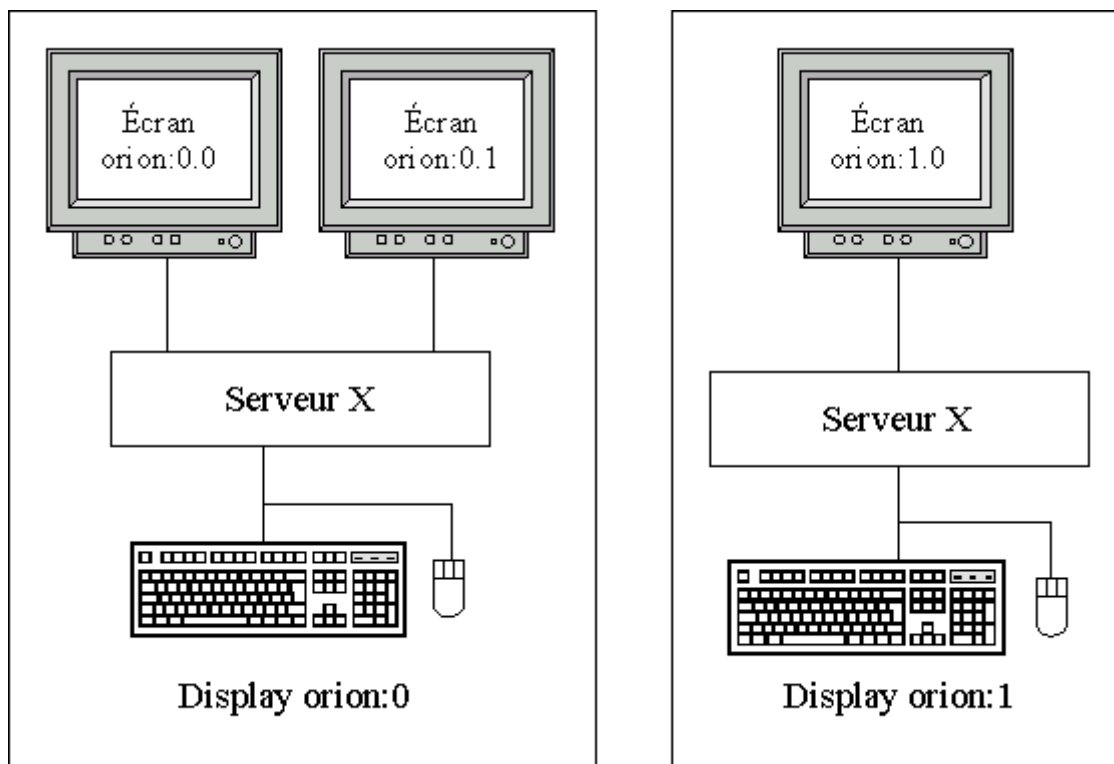
```
machine:display
```

Comme on le voit, la présence du champ machine confirme que XWindow est bien un système graphique réseau. Il peut contenir directement le nom de la machine ou son adresse IP. Si ce champ est absent, le serveur contacté sera l'un des serveurs X de la machine locale, avec un protocole de communication optimisé. Le champ display quant à lui est un numéro permettant d'identifier le display de la machine en question qui doit être utilisé. C'est ce champ qui déterminera le serveur X qui sera utilisé, dans le cas où plusieurs serveurs X fonctionnent sur la même machine.

Un display pouvant avoir plusieurs écrans, il est possible de spécifier l'écran sur lequel l'affichage doit être réalisé. Pour cela, il suffit de suffixer le nom du display par un point suivi du numéro de l'écran :

```
machine:display.écran
```

où le champ écran spécifie le numéro de l'écran de ce display sur lequel l'affichage doit avoir lieu. Ce champ sera rarement utilisé en pratique, car il est assez rare de disposer de plusieurs écrans. Il peut donc être omis, la valeur par défaut utilisée est dans ce cas 0 (pour le premier et unique écran du display).



Notion de display

Une fois la connexion établie, les programmes clients continuent d'indiquer à XWindow le display qui doit être utilisé pour chaque opération graphique à effectuer. Il est donc possible pour un programme de répartir son affichage sur plusieurs écrans, voire de communiquer avec plusieurs serveurs X et donc de gérer plusieurs displays simultanément, éventuellement sur des machines différentes. Ce genre de programme est cependant assez rare, et ne se trouve en pratique que dans le monde de la conception assistée par ordinateur, la visualisation d'images médicales et l'architecture. Les programmes classiques se contentent d'un seul display et effectuent toutes leurs opérations sur un même écran. En revanche, il est possible de configurer les serveurs X pour utiliser automatiquement plusieurs écrans pour un même display, afin de réaliser un écran virtuel gigantesque.

Le display utilisé pour un programme doit donc souvent être fixé par un paramètre de sa ligne de commande. L'option utilisée est `-display`, avec la syntaxe suivante :

```
programme -display nom
```

où `programme` est le programme à exécuter, et `nom` est le nom du display tel qu'il a été décrit ci-dessus. Par exemple, la commande suivante :

```
xterm -display :0
```

permet de lancer le programme `xterm` et de réaliser l'affichage sur le display `:0` (sur l'écran par défaut) de la machine locale.

Vous pouvez cependant vous passer de l'option `-display`, à condition de définir la variable d'environnement `DISPLAY` pour fixer le display par défaut. Vous pouvez fixer la valeur de cette variable à l'aide d'une commande telle que celle-ci :

```
export DISPLAY=:0.0
```

(si vous utilisez `bash`). Dans cet exemple, le serveur X à utiliser se trouve sur la machine locale, le display porte le numéro 0, et l'écran à utiliser est le numéro 0. En général, cette variable d'environnement est fixée à la valeur du display courant lorsqu'on est connecté sous `XWindow`. Par conséquent, vous pouvez lancer vos programmes sans avoir à vous préoccuper du display qu'ils doivent utiliser.

En revanche, vous serez obligé de préciser le display à utiliser lorsque vous lancerez une application à distance en voulant avoir l'affichage en local. Bien entendu, vous devrez au préalable donner les droits à l'utilisateur distant sur votre display local, faute de quoi les mécanismes de sécurité de `XWindow` lui interdiront de se connecter (message d'erreur « `Can't open display` »). Nous verrons plus loin la manière dont `XWindow` gère la sécurité.

X-B - Installation de X.org

Il existe actuellement deux implémentations libres de `XWindow` : l'implémentation historique, `XFree86`, et une branche de `XFree86` créée récemment pour des raisons politiques, `X.org`. Cette duplication est due au fait que beaucoup de développeurs n'ont pas eu le sentiment que leurs requêtes étaient prises en compte par l'équipe de développement de `XFree86`, qui est restée très conservatrice pendant de longues années. Ces problèmes ont même provoqué des querelles intestines au sein de l'équipe de `XFree86`. N'ayant pu se mettre d'accord, les différentes parties ont décidé de développer leur propre version de `XWindow` séparément. À ces problèmes s'est ajouté un changement de licence du projet `XFree86` qui, bien que restant libre, n'a pas été du goût de tout le monde. C'était en quelque sorte la goutte qui a fait déborder le vase.

J'ai pris le parti de décrire `X.org` dans ce document, bien que j'estime que l'on doit encore rendre crédit à l'équipe de `XFree86`. Je considère en effet que les reproches qui sont faits au projet `XFree86` sont justifiés (lenteur du projet, intégration et mises à jour des composants annexes trop lentes, refus d'intégration des modifications nécessaires aux autres projets utilisant `XFree86`, organisation du projet monolithique et sans volonté de le rendre plus modulaire, entre autres problèmes). De plus, il semble que les principales distributions et constructeurs de matériel informatique aient décidé de suivre cette voie. Enfin, je considère que le changement de licence de `XFree86` n'était pas nécessaire et relève de la tentative de putsch de la part de l'équipe de développement actuelle d'une part, et n'a été fait que pour des raisons politiques d'autre part. Il n'est pas certain que les deux projets subsistent à l'avenir, seul l'avenir dira qui avait raison.

Bien que développés par des équipes différentes, `XFree86` et `X.org` sont basés sur les mêmes sources et sont donc totalement compatibles. Cela ne durera pas forcément, car `X.org` est appelé à évoluer différemment et plus vite que `XFree86`, mais à l'heure actuelle, les deux produits n'ont pas eu le temps de diverger suffisamment pour créer des incompatibilités. Les seules différences se situent donc au niveau des noms des commandes et des fichiers de configuration, qui ont été renommés dans `X.org` pour bien se distinguer de `XFree86`. Par conséquent, si votre distribution utilise `XFree86`, vous n'aurez que très peu d'adaptations à faire pour lire ce chapitre.

L'installation de XWindow a été pendant longtemps une tâche ardue et risquée. À présent, il est possible d'installer et de configurer cet environnement relativement facilement, et en prenant beaucoup moins de risques que par le passé. La plupart des distributions installent XWindow et le configurent automatiquement, ce qui fait que vous n'aurez généralement rien à faire. Les informations de ce chapitre ne vous seront donc utiles que pour comprendre comment effectuer des modifications dans la configuration du système.

En fait, les seules difficultés que vous pourrez rencontrer résident dans le choix du pilote utilisé par le serveur X pour prendre en charge votre carte graphique et dans sa configuration. L'essentiel de ces opérations réside donc dans ces deux points stratégiques :

- il faut disposer d'un pilote adapté à sa carte graphique pour le serveur X ;
- si l'écran utilisé est relativement ancien (écran cathodique non compatible avec le protocole DDC), il vous faudra éventuellement connaître les caractéristiques de votre matériel (essentiellement les fréquences de balayage horizontales et verticales supportées par votre moniteur).

Le premier point est quasiment assuré de nos jours, car la plupart des cartes graphiques vendues sur le marché de nos jours sont des cartes de marque Intel (surtout pour les puces graphiques intégrées au chipset), nVidia et ATI. Des pilotes sont disponibles pour la plupart des modèles de ces fabricants.

En fait, vous devrez essentiellement choisir entre les pilotes libres fournis avec X.org, généralement plus stables, mais ne supportant pas forcément les dernières cartes graphiques ou la totalité des fonctions 3D, et les pilotes propriétaires fournis par le fabricant, qui ne supportent souvent pas les anciennes cartes ou qui ne sont pas compatibles avec tous les noyaux, qui sont parfois relativement instables (surtout les pilotes AMD et, dans une moindre mesure, les pilotes nVidia des « anciennes » cartes), mais qui donnent accès à l'ensemble des fonctions 3D.

Note : En fait, seul Intel joue le jeu des pilotes libres, bien qu'ATI semble à nouveau considérer la question depuis son rachat par AMD. NVidia quant à lui fournit des pilotes propriétaires de très bonne qualité et la prise en charge des cartes NVidia sous Linux est excellente. Toutefois, cette société s'est toujours opposée à ce qu'un pilote libre existe (à tel point qu'elle a même fait pression pour que le code source du pilote 2D de X.org soit crypté). À vous de faire votre choix en fonction de vos contraintes financières, matérielles et morales...

Si d'aventure vous ne disposez d'aucun serveur X adapté à votre matériel (matériel trop récent pour lequel le constructeur lui-même n'a pas encore fourni le pilote), vous pourrez vous rabattre sur le pilote VESA, qui fonctionne avec toutes les cartes graphiques compatibles avec le standard VESA. Toutefois, ce pilote ne vous fera bénéficier d'aucune accélération graphique, même 2D.

Pour ce qui est du deuxième point, l'avènement des écrans plats a simplifié considérablement la donne : non seulement ces écrans ne supportent plus qu'une seule résolution, mais la notion de balayage n'a plus de sens non plus. Ces écrans fournissent également les informations nécessaires aux pilotes de cartes graphiques permettant de déterminer les modes graphiques supportés. Toutefois, si votre écran est un vieil écran cathodique, vous aurez peut-être à connaître ses caractéristiques techniques. Pour cela, il suffit souvent de regarder la fiche technique de l'écran. Bien entendu, cela suppose de l'avoir conservée. Si ce n'est pas le cas, il faut espérer que les programmes d'installation connaissent la marque et le modèle du matériel. Il reste toujours la possibilité de demander des renseignements à des personnes qui ont également ce type de matériel (c'est là qu'Internet peut être utile). Les informations les plus importantes sont les plages de fréquences horizontales et verticales du moniteur, ainsi que les durées des signaux de synchronisation horizontale et verticale. Sans ces informations, vous ne parviendrez peut-être pas à installer XWindow. Rassurez-vous cependant, les programmes de configuration de XWindow connaissent la plupart des moniteurs à présent, ce qui fait qu'ils sont capables d'écrire les fichiers de configuration correctement sans que vous ayez à spécifier les paramètres du moniteur.

X-C - Configuration de X.org

La configuration de X.org commence avant tout par l'installation du serveur X. Un même serveur X peut prendre en charge plusieurs cartes graphiques sur une même machine, pourvu qu'il dispose des pilotes adéquats. Inversement, il est possible de lancer plusieurs serveurs X, chacun utilisant sa propre carte graphique, ou partageant la même carte si la machine n'a qu'un seul terminal X.

X.org fournit un unique serveur X, qui prend en charge tous les types de cartes graphiques à l'aide de pilotes spécifiques. Dans une installation normale, ces pilotes sont fournis sous forme de modules de ce serveur. Les pilotes peuvent ainsi être chargés dynamiquement par le serveur X, selon la configuration du système. Cette architecture permet à un même serveur X de charger plusieurs pilotes pour plusieurs cartes graphiques, afin de gérer les configurations disposant de plusieurs écrans connectés à des cartes graphiques de différents types. Il est donc nécessaire que les modules prenant en charge vos cartes graphiques soient installés, ce qui est normalement toujours le cas.

Par convention, le nom du serveur X est toujours X. Comme le nom du fichier programme du serveur X de X.org est Xorg (on s'en serait douté...), il doit exister un lien symbolique `/usr/bin/X` qui pointe vers le fichier `/usr/bin/Xorg`. Ce lien est, encore une fois, normalement toujours présent sur les systèmes correctement configurés.

Une fois XWindow installé, la suite de la configuration de X.org se fait uniquement dans le fichier de configuration `xorg.conf`. Ce fichier est classiquement stocké dans le répertoire `/etc/X11/`. Normalement, vous ne devez pas créer ce fichier vous-même. Votre distribution doit au moins vous en fournir un par défaut et, souvent, elle dispose d'un outil de configuration de X.org convivial qui fera quasiment tout le travail pour vous. Ce genre d'outil est de plus capable de configurer X.org pour les pilotes spécifiques fournis avec les distributions (il n'est pas rare que les sociétés éditrices de distributions développent des serveurs X pour les nouvelles cartes graphiques). Lisez donc votre documentation pour plus de détails à ce sujet.

Vous pouvez également utiliser les programmes de configuration fournis avec X.org, qui vous permettront de générer ce fichier. Il existe trois possibilités pour générer un fichier de configuration `xorg.conf`. La première méthode est de demander directement au serveur X de détecter le matériel présent sur votre machine et de générer le fichier `xorg.conf` correspondant. La deuxième méthode, qui est aussi la plus sûre, est d'utiliser le programme `xorgconfig`. Il s'agit d'un outil fonctionnant en mode texte, qui est effroyablement peu pratique à utiliser. Enfin, un autre outil, beaucoup plus convivial, est en cours de développement. Il s'agit de `xorgcfg`. Cet outil permet de générer et de modifier les fichiers `xorg.conf` de manière conviviale, soit en mode texte avec des menus, soit en mode graphique. Lorsque cet outil sera terminé, la configuration de X.org sera beaucoup plus aisée. Encore une fois, je ne saurai que vous recommander d'utiliser l'outil de configuration fourni avec votre distribution.

X-C-1 - Génération automatique du fichier `xorg.conf`

Le serveur X de X.org est capable de détecter le matériel installé sur une machine et de générer un fichier de configuration `xorg.conf` adapté à ce matériel. Pour cela, il suffit de le lancer avec l'option `-configure` sous le compte `root`, comme dans l'exemple suivant :

```
Xorg -configure
```

À l'issue de cette commande, le serveur X écrira un fichier `xorg.conf.new` dans le répertoire personnel de l'utilisateur `root`.

Cependant, le fichier de configuration ainsi généré n'utilisera que les options de configuration les plus sûres et devra souvent être révisé complètement. En pratique, seuls les paramètres de la carte graphique et de la souris seront correctement détectés. Il ne faut donc utiliser cette fonctionnalité que dans le but d'obtenir un squelette de fichier `xorg.conf`, que l'on personnalisera ensuite. Ce fichier pourra être copié dans le répertoire `/etc/X11/` sous le nom `xorg.conf` une fois qu'il aura été corrigé.

X-C-2 - Utilisation de `xorgconfig`

Comme il l'a été indiqué plus haut, `xorgconfig` est un programme fonctionnant en mode texte exclusivement. Son principe de fonctionnement est très simple : il pose une série de questions sur votre matériel, puis il génère un fichier `xorg.conf` générique pour votre matériel. Vous pourrez ensuite partir de ce modèle pour personnaliser votre configuration et pour préciser les paramètres que `xorgconfig` ne prend pas en charge.

Le principal problème de `xorgconfig` est qu'il ne laisse pas droit à l'erreur. La moindre faute de frappe ou la moindre hésitation est prise comme une réponse valide, et il est impossible de revenir en arrière. Cela est d'autant plus énervant que dans ce cas, il ne reste plus qu'une solution, qui est de tout recommencer à partir du début. Après trois ou quatre erreurs, on finit par faire extrêmement attention à ce que l'on tape.

Lorsqu'on lance `xorgconfig`, il commence par afficher un message indiquant qu'il va créer un nouveau fichier de configuration `xorg.conf` que l'on pourra utiliser par la suite comme point de départ pour paramétrer son système XWindow. Il faut valider pour passer ce message et commencer la configuration proprement dite.

La première question que `xorgconfig` pose est le type de la souris que vous voulez utiliser. Il existe un grand nombre de types de souris sur le marché, cependant seuls deux types sont réellement courants.

Initialement, les souris se connectaient sur le port série des ordinateurs. Ces souris, dites souris sérieelles, étaient relativement répandues et sont généralement référencées sous le terme de « compatible Microsoft ». Il faut donc utiliser l'option « Microsoft compatible (2-button protocol) » pour ces souris. Notez que certaines souris sérieelles utilisent un protocole différent pour gérer un troisième bouton. Si vous disposez d'une telle souris, il faut choisir l'option « Mouse Systems (3-button protocol) ».

Plus récemment, le port PS/2 est apparu pour les claviers et les souris. Ce port permet de gérer directement la plupart des souris actuelles, et il est probable que votre souris soit une souris PS/2. L'option à utiliser est cette fois l'option « PS/2 Mouse ». Il faut surtout ne pas confondre les souris PS/2 avec les souris bus (que l'on peut utiliser avec l'option « Bus Mouse »), qui sont des souris relativement peu répandues et qui utilisaient des bus spéciaux. Les autres options sont réservées à des souris peu répandues. Si vous en utilisez une, vous devez choisir l'option correspondante.

Notez que les souris à molette connectées sur le port PS/2 utilisent un protocole de communication légèrement différent de celui que les autres souris PS/2 utilisent. Malheureusement, `xorgconfig` ne propose pas d'option pour ces souris, et une intervention manuelle dans le fichier de configuration `xorg.conf` est nécessaire pour les configurer. La manière de procéder sera détaillée dans la [Section 10.3.4](#).

La question suivante demande si vous désirez activer la fonctionnalité d'émulation d'un troisième bouton pour les souris à deux boutons. Cette émulation permet d'utiliser les nombreux programmes pour XWindow qui nécessitent d'utiliser une souris à trois boutons. Le clic sur le troisième bouton est alors simulé en appuyant sur les deux boutons de la souris simultanément. Il est très vivement recommandé de répondre par 'y' à cette question si votre souris ne dispose que de deux boutons. En revanche, si elle dispose de plus de trois boutons, ou si elle dispose d'une roulette, il faut répondre par la négative.

`xorgconfig` demande ensuite le port sur lequel votre souris est connectée. Ce port est généralement le port `/dev/ttyS0` pour les souris sérieelles et `/dev/psaux` pour les souris PS/2. Les distributions créent souvent un lien symbolique `/dev/mouse` sur le port effectivement utilisé par la souris, si bien que la réponse par défaut utilise ce port. C'est la réponse recommandée, validez donc pour passer à la question suivante.

Vient ensuite la sélection du type de clavier. Encore une fois, un certain nombre de modèles de claviers ont été vendus sur le marché. Cependant, seuls quelques-uns sont réellement répandus. En France, on trouve essentiellement les claviers internationaux 102 touches et 105 touches, auxquels correspondent les réponses « Generic 102-key (Intl) PC » et « Generic 105-key (Intl) PC ». Si vous utilisez un clavier Microsoft Natural Keyboard, choisissez l'option « Microsoft Natural ».

Vous devrez ensuite indiquer la disposition de ce clavier. Il faut évidemment choisir l'option « French ». `xorgconfig` demande alors de saisir un nom de variante pour le clavier choisi. Comme le clavier français n'est décliné que sous une seule variante, vous pouvez simplement valider pour passer à la suite de la configuration du clavier.

`xorgconfig` vous propose alors de modifier des paramètres additionnels du clavier (emplacement des touches modificatrices telles que les touches de majuscule, de contrôle et de jeu de caractères alternatif, état des diodes, etc.). En général, cela n'est pas nécessaire et vous pouvez répondre 'n' à cette question.

La partie difficile vient ensuite. `xorgconfig` vous le signale avec un message indiquant que les informations de synchronisation horizontale et verticale sont extrêmement importantes pour configurer correctement les modes vidéo. La signification de ces valeurs vous sera décrite plus loin en détail, pour l'instant, contentez-vous de récupérer le manuel de votre moniteur et recherchez ses caractéristiques précises. Validez ensuite pour passer à la question suivante.

`xorgconfig` vous demande alors la plage de fréquences horizontales que votre moniteur est en mesure d'accepter. Il propose un certain nombre de choix standards, qui correspondent aux différents types de moniteurs existant sur le marché. Cependant, ces valeurs sont les plus mauvaises, car il est fort probable que votre moniteur sache faire mieux que ce que les standards imposent. Vous devez donc soit accepter une des plages de valeurs proposées, soit saisir la plage correspondant exactement à votre moniteur à l'aide de l'option « Enter your own horizontal sync range ». Si vous choisissez cette dernière option, vous devrez ensuite saisir la plage de valeurs des fréquences horizontales de votre moniteur. Ne les inventez pas, cela ne marchera pas. Saisissez les vraies valeurs.

La même question est alors posée pour la plage de fréquences verticales. Encore une fois, vous pouvez choisir l'une des plages proposées selon le type de votre moniteur, ou saisir vous-même la plage de fréquences citée dans ses caractéristiques techniques à l'aide de l'option « Enter your own vertical sync range ». À l'issue de cette question, `xorgconfig` vous demande de saisir le nom du moniteur que vous venez ainsi de configurer. Ce nom est arbitraire, il ne sert que pour l'identifier de manière lisible par la suite. Entrez, par exemple, le nom du modèle de l'écran dont vous disposez.

La question suivante vous demande simplement si vous désirez choisir votre carte graphique dans la liste des cartes graphiques supportées par X.org. Il est recommandé de répondre par l'affirmative en tapant 'y'. `xorgconfig` affiche alors la liste des cartes gérées, qui est assez longue. En fait, elle se présente sur plusieurs pages, et il faut appuyer sur la touche Entrée pour passer d'une page à la suivante. Si vous avez passé une page de trop, vous êtes bon pour passer toutes les pages pour revenir à la première, avant d'aller sur la page contenant votre carte graphique. Lorsque vous aurez trouvé votre carte, choisissez l'option correspondante et validez.

Si votre carte n'est pas présente dans la liste, cela ne signifie pas qu'elle n'est pas supportée par X.org. En effet, chaque pilote est capable de prendre en charge un type de puce électronique, et il arrive que plusieurs cartes graphiques soient basées sur une électronique provenant d'un même fabricant. C'est la raison pour laquelle les pilotes de X.org portent généralement le nom des puces utilisées par les cartes. Par conséquent, si vous ne trouvez pas votre carte dans la liste, vous pouvez essayer de spécifier le pilote correspondant à l'électronique de votre carte. Dans le pire des cas, vous pourrez quasiment toujours utiliser le pilote générique VESA, qui permet de piloter toutes les cartes graphiques compatibles avec les standards VESA 2.0. Cependant, vous ne bénéficierez d'aucune accélération matérielle avec ce pilote.

Les questions qui suivent dépendent fortement de la carte sélectionnée. En effet, chaque carte peut avoir besoin d'un certain nombre de paramètres complémentaires pour fonctionner. Encore une fois, ces paramètres sont, normalement, indiqués dans le mode d'emploi de votre carte graphique ou, au pire, sur les composants de la carte eux-mêmes. Une des questions courantes concerne la quantité de mémoire vidéo dont cette carte dispose. Pour cette question, les choix les plus courants sont proposés, mais vous pouvez également spécifier vous-même cette quantité à l'aide de l'option « Other ». L'unité est ici le kilo-octet, pensez donc bien à multiplier par 1024 le nombre de mégaoctets de mémoire vidéo de votre carte graphique avant de saisir la valeur. Lorsque la configuration de la carte graphique sera terminée, `xorgconfig` vous demandera de saisir le nom de cette carte. Encore une fois, ce nom est arbitraire, mais vous devriez saisir un nom cohérent avec votre modèle.

`xorgconfig` vous propose alors de modifier l'ordre d'utilisation des modes graphiques pour chaque profondeur de couleur. Cet ordre doit être spécifié en donnant les numéros des différents modes, les uns après les autres, et en ne les séparant pas par des espaces. Par exemple, le nombre 432 sélectionnera les modes 4, 3 et 2, soit les modes 1024x768, 800x600 et 640x480. Notez que l'ordre des numéros est important, c'est l'ordre dans lequel ces modes seront choisis lorsqu'on basculera de l'un à l'autre. Lorsque vous aurez saisi les modes graphiques à utiliser et leur ordre d'apparition, `xorgconfig` vous proposera d'utiliser un écran virtuel de taille supérieure à la résolution physique du plus grand des modes graphiques choisis. Cela signifie que la résolution de l'affichage sera supérieure à celle de votre écran, et que XWindow fera défiler automatiquement l'image affichée lorsque la souris sortira de la portion d'image visible. Il est en général conseillé de répondre par la négative à cette question, car cela peut être facilement déroutant. Cependant, c'est une affaire de goût, et vous êtes libre d'accepter ce comportement. Notez que, quelle

que soit la réponse que vous donnerez, XWindow utilisera comme résolution logique la résolution du plus grand mode que vous avez sélectionné. Par conséquent, ce mécanisme de défilement sera encore utilisé pour les modes graphiques de résolution inférieure, avec comme taille d'écran virtuel la taille du mode ayant la résolution la plus grande. Lorsque vous aurez configuré tous vos modes graphiques, vous pourrez passer à la suite en choisissant l'option « The modes are OK, continue. ».

xorgconfig vous demande alors le nombre de couleurs à utiliser par défaut lorsque vous démarrez en mode graphique. Choisissez bien, car il ne sera pas possible de changer cette valeur une fois que l'environnement graphique sera lancé. Vous pourrez bien entendu la modifier manuellement, mais il vous faudra relancer le serveur X après cela. La plupart des cartes graphiques modernes supportent les modes graphiques à 16 millions de couleurs, la réponse recommandée est donc « 24 bits (16 millions colors) ».

Enfin, la dernière question est tout simplement si vous désirez enregistrer le fichier `xorg.conf` correspondant aux options que vous avez choisies. Il faut bien entendu répondre par 'y', ou sinon vous devriez vous demander pourquoi vous êtes en train de lire ceci...

Le fichier `xorg.conf` généré est généralement fonctionnel, mais parfaitement améliorable. Le format de ce fichier sera décrit en détail dans la [Section 10.3.4](#). Si vous l'ouvrez, vous constaterez que xorgconfig a ajouté un nombre impressionnant de commentaires pour vous aider dans vos expérimentations. Bien entendu, vous trouverez également des informations complémentaires dans la page de manuel `xorg.conf`.

X-C-3 - Utilisation de xorgcfg

La manière la plus agréable d'effectuer la configuration de X.org est sans nul doute d'utiliser l'utilitaire `xorgcfg`. Ce programme permet aussi bien de créer un fichier de configuration `xorg.conf` initial que de modifier une configuration existante de manière graphique. Il dispose également d'un mode texte, qui reprend les mêmes questions que `xorgconfig`, mais d'une manière plus conviviale, à l'aide d'un système de menus.

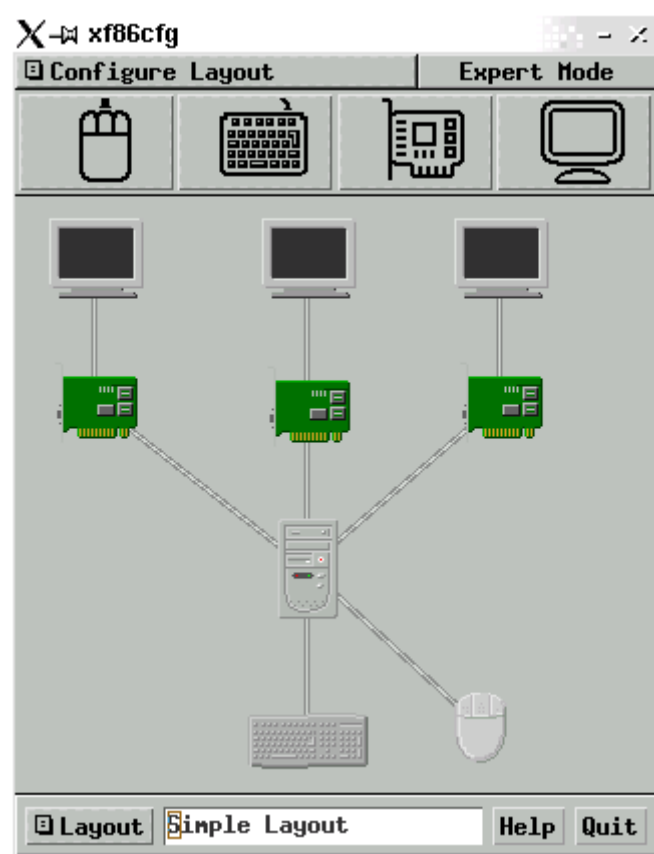
X-C-3-a - Configuration en mode graphique

Lorsqu'on lance `xorgcfg`, celui-ci commence par regarder si la variable d'environnement `DISPLAY` est définie ou non. Si elle est définie, il tente de se connecter au serveur X gérant ce display afin de permettre l'édition du fichier de configuration `xorg.conf` du système. Notez qu'il ne permet pas de modifier le fichier `xorg.conf` utilisé par le serveur X auquel il se connecte si celui-ci utilise un autre fichier que le fichier du système : le serveur X n'est utilisé par `xorgcfg` que pour son affichage.

Si, en revanche, la variable d'environnement `DISPLAY` n'est pas définie, `xorgcfg` considère que XWindow n'est pas installé sur la machine locale et appelle le serveur X avec l'option `-configure` afin de générer un nouveau fichier `xorg.conf`. Il lance ensuite le serveur X détecté et utilise ce serveur pour permettre la modification du fichier `xorg.conf` ainsi créé.

À son démarrage, il vérifie que la souris est correctement configurée. Si tel n'est pas le cas, il propose d'utiliser les touches du curseur du pavé numérique pour déplacer le pointeur de la souris, ainsi que les touches `/`, `*` et `-` respectivement pour le premier, le deuxième et le troisième bouton de la souris. Je déconseille fortement d'essayer d'effectuer la configuration dans ces conditions, car ce n'est réellement pas utilisable. Dans ce cas de figure, on cherchera plutôt à utiliser l'interface en mode texte de `xorgcfg`, que nous présenterons dans la section suivante.

L'interface de `xorgcfg` en mode graphique est très simple. La partie supérieure de la fenêtre comprend un menu avec deux entrées. La première entrée permet de choisir les différentes parties intervenant dans la configuration. La deuxième entrée (intitulée « Expert Mode ») donne accès quant à elle à un mode de paramétrage permettant d'éditer directement les propriétés du fichier `xorg.conf`. Ce mode de fonctionnement est réservé aux utilisateurs avertis et ne sera pas décrit ici.



La première entrée de menu comprend plusieurs options. L'option « **Configure Layout** » permet d'effectuer la configuration générale du display. L'option « **Configure Screen** » permet de configurer la disposition des différents écrans d'un même display les uns par rapport aux autres. Elle n'est réellement utile que dans le cas des configurations à plusieurs écrans. L'option « **Configure Modeline** » donne la possibilité de configurer les différents modes graphiques de chaque configuration. Enfin, l'option « **Configure AccessX** » permet de spécifier les options des différents périphériques d'entrée. Ces deux options de menus sont réservées aux utilisateurs expérimentés et ne seront pas décrites dans ce document.

L'essentiel de la configuration se fait dans l'écran affiché lorsque l'option « **Configure Layout** » est choisie. Cet écran est constitué d'une barre d'icônes permettant d'ajouter les différents périphériques à la configuration courante : souris, claviers, cartes graphiques et moniteurs. Il n'est possible d'éditer qu'une seule configuration à la fois, la configuration courante peut être sélectionnée à l'aide du bouton situé dans la partie inférieure gauche de la fenêtre. Le corps de la fenêtre elle-même contient une représentation de cette configuration, avec les différents périphériques utilisés et les relations qui existent entre eux.

En cliquant avec le bouton droit de la souris sur ces éléments, vous pouvez faire apparaître un menu contextuel concernant cet élément. Ce menu peut contenir l'option « **configure** », qui donne accès à une boîte de dialogue permettant d'éditer les propriétés de l'élément, l'option « **option** », qui permet d'ajouter des options générales à cet élément, les options « **enable** » et « **disable** », qui permettent d'activer ou de désactiver l'élément en question dans la configuration, et l'option « **remove** », dont le but est de supprimer l'élément concerné.

Une configuration typique comprend au moins une souris, un clavier, une carte graphique et un moniteur. Il est recommandé d'ajouter les éléments de la configuration dans cet ordre, afin d'éviter d'avoir à revenir plusieurs fois sur certains de ces éléments.

La boîte de dialogue ouverte par l'option « **configure** » sur une souris comprend un champ contenant l'identificateur de cette souris, la liste des fichiers spéciaux de périphériques pour la sélection du fichier spécial de périphérique auquel la souris est connectée, la liste des protocoles gérés par X.org et un bouton permettant d'activer l'émulation du troisième bouton pour les souris à deux boutons. Généralement, les souris sont connectées soit sur le port série

(fichier spécial de périphérique `/dev/ttyS0` ou `/dev/ttyS1`), soit sur le port PS/2 (fichier spécial de périphérique `/dev/psaux`). Les principaux protocoles gérés par X.org sont les suivants :

Microsoft

C'est le protocole utilisé par la plupart des souris connectées sur le port série. Ce type de souris est de plus en plus rare actuellement, ne choisissez cette option que si vous disposez de ce type de souris ;

IntelliMouse

C'est le protocole utilisé par les souris à molette connectées sur le port série. N'utilisez pas ce protocole pour les souris à molette connectées sur le port PS/2, la souris ne fonctionnerait pas.

PS/2

C'est le protocole qui convient pour la majorité des souris connectées sur un port PS/2. Notez toutefois que ce n'est pas le cas des souris à molette Logitech et Microsoft ;

IMPS/2

C'est le protocole des souris à molette connectées sur port PS/2. Malheureusement, ce protocole n'est pas proposé par `xorgcfg`. Il est donc impossible de configurer une souris de ce type avec cet utilitaire, sauf à passer dans le mode expert. La configuration à utiliser pour ce type de souris sera décrite dans la [Section 10.3.4](#).

Lorsque vous aurez configuré votre souris, vous pourrez utiliser le bouton « Apply changes » pour prendre en compte les changements si vous avez modifié les paramètres de la souris courante. Dès lors, vous devriez avoir une souris fonctionnelle, et vous pouvez l'activer avec l'option « enable » du menu contextuel. Lorsque la souris est activée, un lien apparaît entre l'unité centrale de l'ordinateur et cette souris dans la fenêtre de `xorgcfg`.

La boîte de dialogue ouverte par l'option « configure » du menu contextuel des claviers comprend, comme la boîte de dialogue des propriétés des souris, un champ permettant de saisir le nom de ce clavier dans la configuration. Vous pouvez également choisir le modèle de clavier et sa disposition en fonction de sa langue. Pour les claviers français, le modèle à utiliser est généralement celui des claviers internationaux 102 ou 105 touches, et la disposition à utiliser est la disposition « French ». Lorsque vous aurez configuré le clavier correctement, vous pourrez appliquer les changements avec le bouton « Apply changes ». Vous pourrez également activer le clavier dans la configuration courante à l'aide de l'option « enable » du menu contextuel du clavier. Un lien entre ce clavier et l'unité centrale doit alors apparaître dans la fenêtre principale de `xorgcfg`.

La boîte de configuration des cartes graphiques contient le champ habituel pour donner un nom à la carte et la liste des cartes graphiques reconnues automatiquement par X.org. Si votre carte ne se trouve pas dans cette liste, vous devrez choisir le pilote à utiliser. En général, les pilotes portent le nom de la puce électronique sur laquelle la carte graphique est basée. Si aucun pilote n'est adapté pour votre carte, vous pourrez malgré tout utiliser un des pilotes génériques `vga` ou `vesa`. Enfin, dans le cas où plusieurs cartes graphiques sont installées dans l'ordinateur, vous devrez spécifier l'adresse de la carte sur le bus PCI/AGP. N'oubliez pas d'activer la carte graphique à l'aide de l'option « enable » du menu contextuel lorsque vous aurez fini la configuration.

La boîte de dialogue des moniteurs permet de spécifier un nom pour le moniteur ainsi que ses plages de fréquences horizontales et verticales. Vous pouvez aussi sélectionner la carte graphique à laquelle ce moniteur est connecté. La liste ne propose que les cartes graphiques que vous avez configurées précédemment. Les moniteurs n'ont pas besoin d'être activés, car ils sont liés aux cartes graphiques.

Une fois la configuration générale définie, vous pouvez, si elle contient plusieurs écrans, spécifier la disposition relative de ces écrans à l'aide de l'option « Configure Screen » du menu principal de `xorgcfg`. Cette option affiche les différents écrans dans la fenêtre de `xorgcfg`, et vous pouvez les faire glisser pour les disposer comme vous le désirez.

Enfin, l'option de menu principal « Configure Modeline » vous permettra de définir de nouveaux modes graphiques et de les ajuster. Cet écran fournit les mêmes fonctionnalités que l'utilitaire xvidtune, que l'on décrira en détail dans la [Section 10.3.6](#). Cet écran ne sera donc pas traité plus en détail ici.

Lorsque vous quittez xorgcfg, il vous propose d'enregistrer le fichier de configuration `/etc/X11/xorg.conf` correspondant à la configuration que vous venez de définir. Vous pouvez spécifier un autre nom si vous le désirez, afin d'enregistrer ces paramètres dans un autre fichier. xorgcfg demande également si vous désirez sauvegarder la configuration du clavier, pour le cas où vous auriez modifié les paramètres des périphériques d'entrée.

X-C-3-b - Configuration en mode texte

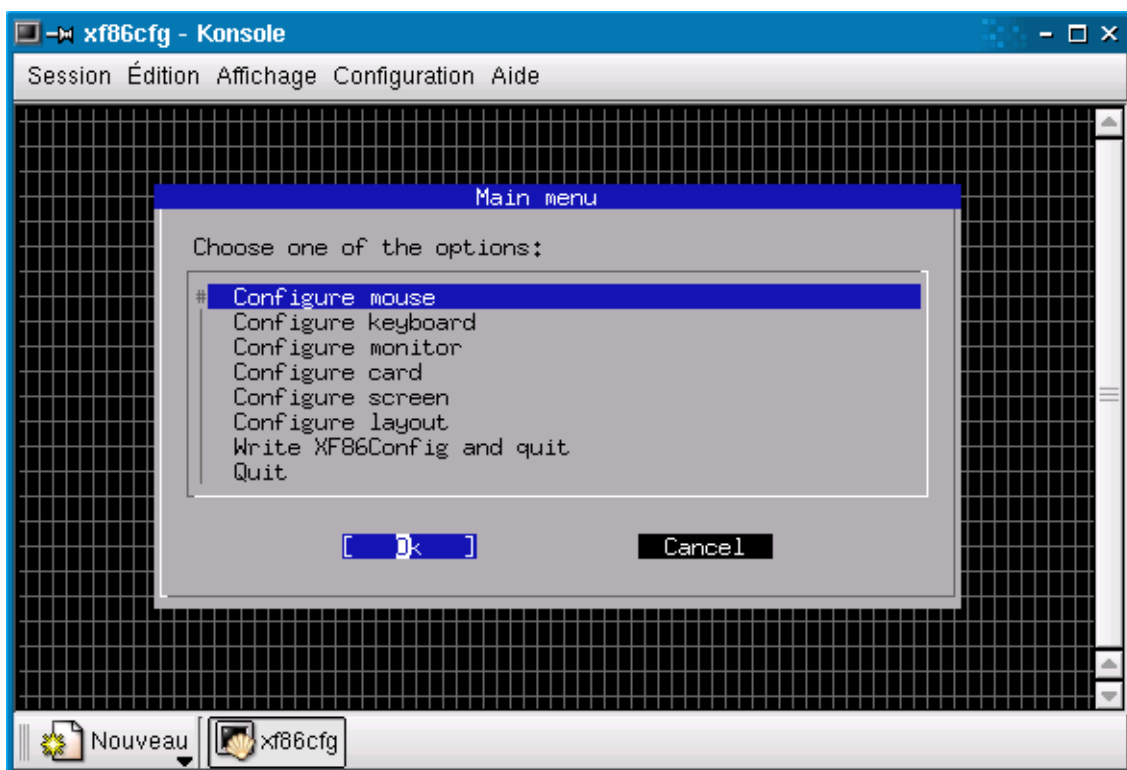
xorgcfg permet également d'effectuer la configuration de X.org en mode texte, à l'aide d'une interface basée sur un système de menus. Bien que ce mode de fonctionnement soit en mode texte, il permet d'effectuer les mêmes tâches que xorgconfig d'une manière beaucoup plus conviviale.

Pour lancer xorgcfg en mode texte, il suffit simplement de lui passer l'option `-textmode` en ligne de commande :

```
xorgcfg -textmode
```

Lors de son démarrage, xorgcfg affiche un écran d'accueil que vous pouvez passer en validant l'option « OK ».

xorgcfg affiche ensuite son menu principal, qui comprend des entrées pour ajouter des souris, claviers, moniteurs et cartes graphiques. Ce menu contient également une entrée « Configure screen », qui permet d'associer les cartes graphiques aux moniteurs, et une entrée « Configure layout », qui permet de configurer le display en spécifiant le clavier, la souris et les différents écrans à utiliser.



Le menu de configuration de la souris permet d'ajouter, de supprimer et de modifier des souris. L'ajout d'une souris nécessite de saisir un identificateur pour la souris, le protocole que cette souris utilise et si le troisième bouton doit être simulé ou non. xorgcfg demande également le fichier spécial de périphérique à utiliser pour accéder à cette souris. Notez que xorgcfg ne permet pas de choisir le protocole IMPS/2 utilisé par les souris à molette connectées sur le port PS/2, ce type de souris ne peut donc pas être configuré avec cet outil.

De la même manière, le menu de configuration du clavier demande de saisir un identificateur lorsqu'on ajoute un nouveau clavier. Le type de clavier est également demandé, ainsi que sa disposition. Pour les claviers français, il faut utiliser les claviers internationaux 102 ou 105 touches.

La configuration des moniteurs permet de spécifier un identificateur pour le moniteur, ainsi que les plages de fréquences horizontales et verticales. La configuration des cartes graphiques quant à elle permet également de spécifier un identificateur pour la carte, et de choisir le type de carte dans la liste des cartes prises en charge par X.org. Si votre carte n'est pas listée, vous devrez choisir l'option « **** Unlisted card **** » et indiquer le pilote à utiliser pour cette carte. Ce pilote porte généralement le nom de la puce électronique sur laquelle cette carte est conçue, mais vous pouvez également utiliser les pilotes génériques vga et vesa pour les cartes graphiques compatibles avec ces standards. Enfin, xorgcfg vous propose de spécifier l'adresse de la carte sur le bus PCI/AGP, pour le cas où vous auriez plusieurs cartes installées sur la même machine.

Les identifiants donnés aux moniteurs et aux cartes sont utilisés dans le menu de configuration des écrans, qui demande quelle carte et quel moniteur sont utilisés pour chaque écran. La profondeur de couleur par défaut est également demandée, ainsi que la liste des modes graphiques gérés par cet écran. Notez que l'interface en mode texte de xorgcfg ne permet pas de définir de nouveaux modes graphiques, pour cela, il faut recourir à l'interface en mode graphique ou éditer manuellement le fichier de configuration `xorg.conf`.

Enfin, le menu de configuration général permet de définir, de modifier et de supprimer les displays. Chaque display doit contenir une référence à une souris, un clavier et un ou plusieurs écrans. Remarquez encore une fois que l'interface en mode texte de xorgcfg ne permet pas de préciser la disposition relative des écrans les uns par rapport aux autres. Pour cela, vous devrez utiliser l'interface en mode graphique ou éditer manuellement le fichier de configuration de X.org.

Lorsque vous aurez fini la configuration de X.org, vous pourrez l'enregistrer à l'aide du menu « `Write xorg.conf and quit` ». Ce menu vous demande le chemin complet sur le fichier de configuration à écrire. Vous pouvez spécifier un autre fichier que le fichier de configuration du système si vous désirez l'éditer et le modifier manuellement de manière indépendante.

X-C-4 - Description du fichier `xorg.conf`

Les fichiers générés par les outils de configuration de X.org sont, comme nous l'avons déjà précisé, des fichiers devant servir de point de départ pour réaliser votre configuration. Normalement, ils permettent de démarrer le serveur X et de travailler dans de bonnes conditions, toutefois, il est possible d'améliorer sensiblement la qualité de l'affichage et l'ergonomie du système en mettant un peu la main à la pâte. Cette section entreprend donc de décrire la structure du fichier de configuration `xorg.conf` et les principales options que l'on peut utiliser. D'autre part, il peut être instructif de connaître la nature des informations que ce fichier contient, et quelles sont les conséquences des choix effectués dans les utilitaires de configuration.

X-C-4-a - Structure générale du fichier `xorg.conf`

Le fichier `xorg.conf` est constitué d'un certain nombre de sections contenant chacune les options pour une partie de l'architecture de X.org. La configuration de X.org est un sujet ardu, aussi seules les principales sections du fichier `xorg.conf` seront décrites dans ce document. Vous pouvez vous référer à la page de manuel `xorg.conf` et à la documentation de X.org pour plus de détails à ce sujet.

Certaines sections peuvent être définies plusieurs fois, avec différents jeux d'options à chaque fois, afin de permettre la définition de plusieurs configurations. Bien entendu, lorsqu'un serveur X tourne, une seule configuration est utilisée. Cette configuration est spécifiée soit directement en ligne de commande lors du lancement du serveur X, soit en indiquant une configuration par défaut dans le fichier de configuration.

D'autres sections sont globales, et contiennent les options de configuration générales du serveur X. Ces options concernent typiquement l'environnement dans lequel il est supposé fonctionner, et les options qui ne dépendent pas de la configuration utilisée.

La première section globale est la section « **Files** », qui porte relativement mal son nom, car au lieu de fichiers, elle indique les chemins vers les différentes ressources que le serveur X peut utiliser. Ces ressources comprennent les polices de caractères, l'emplacement des modules complémentaires et l'emplacement des fichiers de définitions de couleurs.

La deuxième section globale est la section « **ServerFlags** ». Cette section contient les options générales à utiliser par défaut pour tous les serveurs X. Elles permettent d'activer ou de désactiver certaines fonctionnalités ou extensions non standards. Ces options fournissent des valeurs par défaut, elles peuvent être modifiées pour chaque serveur X de manière indépendante si nécessaire.

La troisième section globale est la section « **Module** ». Cette section contient les commandes de chargement des différents modules du serveur X, lorsque ce serveur est compilé sous forme modulaire. Chaque module gère une certaine partie des fonctionnalités, et celles-ci peuvent donc être activées ou désactivées au niveau de cette section.

Viennent ensuite les sections de configuration des displays. Ces sections décrivent chacune un des aspects du display, et peuvent apparaître en de multiples exemplaires.

Le premier type de section de configuration des displays regroupe toutes les sections qui définissent les périphériques d'entrée de données (clavier, souris, table de digitalisation, etc.). Ces sections sont toutes nommées « **InputDevice** ».

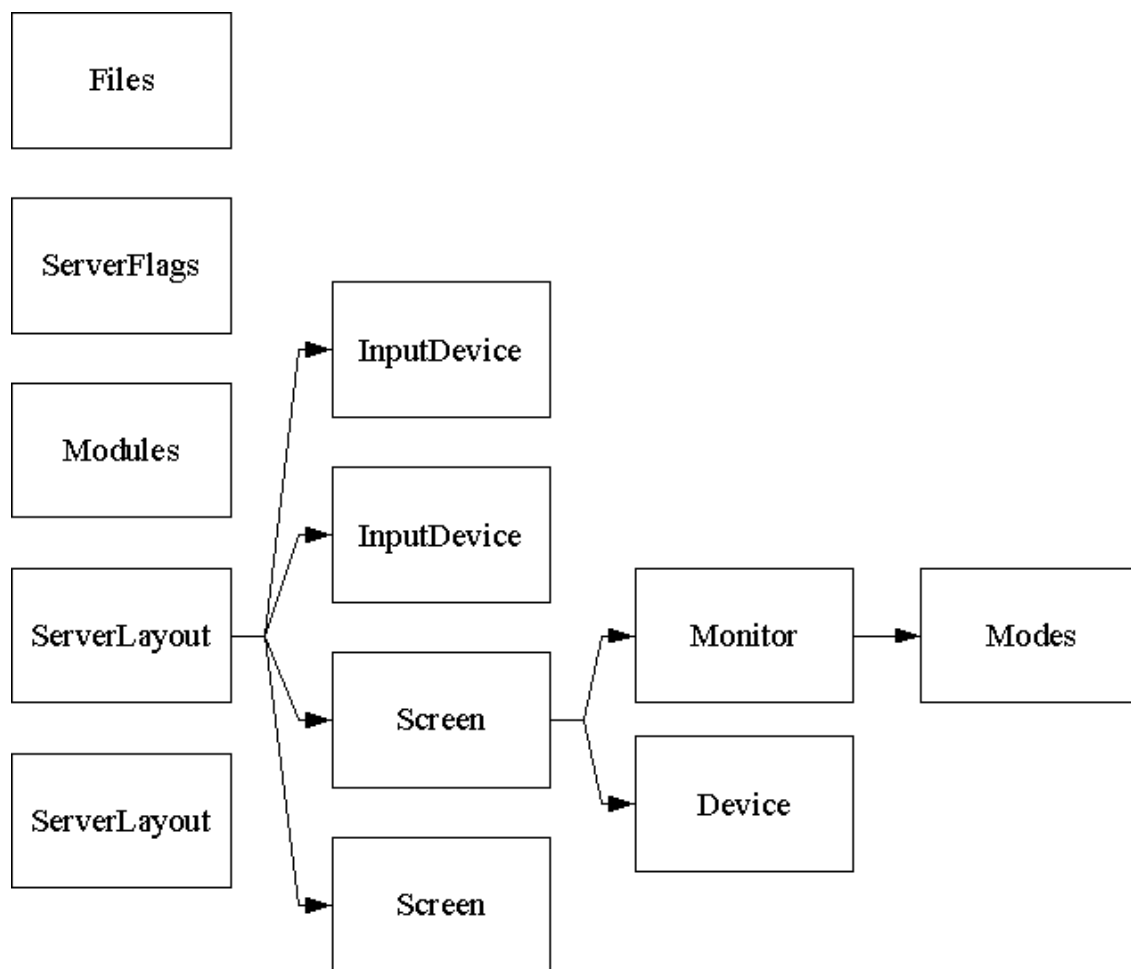
Le deuxième type de section de configuration contient la définition des adaptateurs graphiques. Ces sections permettent de donner les renseignements nécessaires pour la configuration des différentes cartes graphiques installées sur le système. Elles sont toutes nommées « **Device** ».

Viennent ensuite les sections « **Monitor** » qui, comme leur nom l'indique, permettent de définir les caractéristiques physiques des différents moniteurs utilisables. Encore une fois, il peut exister plusieurs sections de ce type, pour définir plusieurs moniteurs dans les configurations multitêtes.

Normalement, chaque moniteur est, en fonction de ses capacités, capable de gérer différents modes graphiques. Les sections « **Monitor** » peuvent donc contenir des définitions de modes graphiques, mais ce n'est pas la technique recommandée. En effet, il est possible d'utiliser plusieurs moniteurs avec des modes graphiques identiques, aussi X.org donne-t-il la possibilité de définir les modes graphiques en dehors des sections « **Monitor** » (ce n'est toutefois pas une obligation). Dans ce cas, les modes graphiques seront définis dans des sections nommées « **Modes** », et celles-ci seront référencées par les sections des moniteurs qui les utilisent.

Les moniteurs sont faits pour être branchés sur des cartes graphiques. Comme on l'a vu ci-dessus, il est possible de définir plusieurs sections « **Device** » et plusieurs sections « **Monitor** », afin de décrire plusieurs cartes graphiques et plusieurs moniteurs. Il faut donc spécifier quels moniteurs sont connectés à quelles cartes graphiques. C'est exactement le rôle des sections « **Screen** ». Pour X.org, un écran n'est donc rien d'autre qu'un couple fonctionnel moniteur / carte graphique. Notez que certaines cartes graphiques haut de gamme peuvent prendre en charge plusieurs écrans. Dans ce cas, il faudra définir plusieurs sections « **Device** » pour la même carte graphique, et spécifier les ports de sortie dans les options de ces sections.

Enfin, il ne reste plus qu'à définir le display lui-même. Nous avons déjà défini la notion de display ci-dessus comme étant le regroupement d'un clavier, d'une souris et d'un ou plusieurs écrans. Ces associations sont donc définies dans des sections nommées « **ServerLayout** ». Une section de ce type contient donc nécessairement une référence à une section « **InputDevice** » pour le clavier et une section « **InputDevice** » pour le périphérique de pointage, et une ou plusieurs sections « **Screen** » pour les écrans utilisés par le serveur X. D'autre part, comme leur nom l'indique, les sections « **ServerLayout** » permettent également de spécifier la disposition relative des écrans les uns par rapport aux autres, dans le cas des configurations multitêtes.



Structure du fichier xorg.conf

Nous allons à présent décrire un peu plus en détail les différentes sections qui ont été présentées ici. Toutes ces sections sont introduites par le mot clé « Section » suivi du type de la section, indiqué entre guillemets, et se terminent toutes par le mot clé « EndSection ». Entre ces deux balises, un certain nombre d'informations peuvent être spécifiées, et peuvent même être regroupées en sous-sections. Étant donné le grand nombre d'options qui peuvent être indiquées dans les sections du fichier xorg.conf, seules les plus classiques seront décrites ci-dessous.

X-C-4-b - Section « Files »

La section « Files », contient les chemins vers les ressources utilisés par X.org. Ce peut être les répertoires de polices de caractères, les répertoires d'installation des modules du serveur X, ou encore des chemins indiquant l'adresse et le port de serveurs de polices sur un réseau. Cette section est donc relativement riche de renseignements pour le serveur X. Un exemple typique de section « Files » est donné ci-dessous :

```

Section "Files"
    RgbPath      "/usr/share/X11/rgb"
    FontPath      "/usr/share/fonts/local"
    FontPath      "/usr/share/fonts/misc"
    FontPath      "/usr/share/fonts/75dpi"
    FontPath      "/usr/share/fonts/100dpi"
    ModulePath    "/usr/lib/modules"
EndSection
  
```

Les chemins indiqués sont des chemins Unix classiques, et sont introduits par les mots clés « FontPath », « RgbPath » et « ModulePath » respectivement pour les chemins sur les répertoires de polices, les répertoires de fichiers de définition de couleurs et les répertoires d'installation des modules. Cependant, comme on le verra plus loin, l'accès aux serveurs de polices se fait à l'aide d'une syntaxe de chemin particulière.

X-C-4-c - Section « ServerFlags »

Cette section regroupe les principales options globales de tous les serveurs X. Ces options sont généralement introduites à l'aide du mot clé « Option », suivi du nom de l'option entre guillemets, lui-même suivi d'une éventuelle valeur pour cette option, elle aussi entre guillemets. Par exemple, l'option :

```
Option "DontZap"
```

permet d'empêcher que l'utilisateur puisse utiliser la combinaison de touches CTRL+ALT+BACKSPACE (en temps normal, cette combinaison de touches a pour conséquence de tuer le serveur X, et de forcer la déconnexion de toutes les applications en cours de fonctionnement).

De même, l'option :

```
Option "Dont Zoom"
```

désactive les combinaisons de touches CTRL+ALT+PLUS et CTRL+ALT+MINUS, où PLUS et MINUS sont respectivement les touches plus et moins du pavé numérique. Ces deux combinaisons de touches sont normalement utilisées respectivement pour passer d'un mode vidéo au suivant ou au précédent.

Il existe beaucoup d'autres options générales du serveur X. Les plus intéressantes sont sans doute les options « BlankTime », « StandbyTime », « SuspendTime » et « OffTime », qui permettent de contrôler les durées des économiseurs d'écran matériels. Vous trouverez la description des autres options dans les fichiers xorg.conf générés automatiquement, et bien entendu dans la page de manuel xorg.conf.

X-C-4-d - Section « Module »

Cette section contient la liste des modules que les serveurs X doivent charger lorsqu'ils démarrent. Ces modules peuvent être spécifiés de deux manières différentes, une abrégée et une plus complète. La forme abrégée utilise le mot clé « Load », suivi du nom du module entre guillemets. Par exemple, la ligne suivante permet de demander le chargement du module de gestion des polices TrueType :

```
Load "freetype"
```

La forme complète utilise une sous-section, introduite par le mot clé « SubSection » suivi du nom du module entre guillemets, et se terminant par le mot clé « EndSubSection ». Ces sous-sections ont la même structure que les sections normales, et permettent de spécifier des options pour le module à l'aide du mot clé « Option ». Par exemple, les extensions du serveur X peuvent être chargées à l'exception de l'extension DGA à l'aide de la sous-section suivante :

```
SubSection "extmod"  
    Option "omit X.org-DGA"  
EndSubSection
```

Note : L'extension DGA permet aux applications d'accéder directement à la mémoire vidéo de la carte graphique. La désactiver accroît donc la sécurité du système, mais peut empêcher certaines applications de fonctionner correctement. C'est le cas des jeux, et surtout des programmes de télévision sous XWindow. Par conséquent, il est probable que vous vouliez laisser ces extensions activées. Dans ce cas, il suffit de ne pas spécifier l'option omit de l'exemple précédent.

Outre les modules freetype et extmod donnés dans les exemples précédents, vous pourrez peut-être utiliser les modules dri et glx. Ces modules permettent respectivement d'activer les fonctionnalités DRI (abréviation de l'anglais « Direct Rendering Infrastructure ») du noyau et le support de la 3D par OpenGL. Les différentes cartes capables de

gérer les fonctionnalités DRI sont celles listées dans la configuration du noyau, que l'on a déjà vue dans la [Section 7.3](#). Notez toutefois que ces fonctionnalités ne sont indispensables pour utiliser le module glx que pour ces cartes. En particulier, les cartes graphiques basées sur les puces NVidia peuvent utiliser le module glx sans charger le module dri, car le pilote fourni par NVidia utilise un autre mécanisme.

X-C-4-e - Section « InputDevice »

Les sections « InputDevice » permettent de décrire tous les types de périphériques d'entrée. En pratique, il s'agira souvent de claviers et de souris, mais il est également possible de connecter des périphériques plus exotiques tels que les joysticks et les tablettes de dessin.

Les sections « InputDevice » doivent être nommées avec un identificateur unique, introduit par le mot clé « Identifier ». Ce mot clé doit être suivi du nom de la section, donné entre guillemets. C'est ce nom qui sera utilisé pour référencer la section dans les sections « ServerLayout » pour définir les différents displays.

La nature du périphérique d'entrée est ensuite spécifiée par la valeur de l'option « Driver ». En pratique, on utilisera quasiment toujours Keyboard ou Mouse, qui correspondent bien évidemment aux pilotes pour les claviers et les souris.

La suite des options de la section « InputDevice » est fonction de la nature du pilote utilisé. Les options les plus courantes pour un clavier sont données dans l'exemple suivant :

```
Section "InputDevice"
    Identifier "Clavier 1"
    Driver      "Keyboard"

# L'option suivante permet de spécifier le délai
# avant répétition (en millisecondes) et la vitesse
# de répétition une fois ce délai passé (en caractères
# par seconde) :
    Option      "AutoRepeat" "500 30"

# L'option suivante indique que la définition de clavier
# à utiliser est celle de X.org :
    Option      "XkbRules" "xfree86"

# L'option suivante indique que le modèle de clavier
# utilisé est un clavier 105 touches :
    Option      "XkbModel" "pc105"

# L'option suivante indique que la disposition
# des touches est celle d'un clavier français :
    Option      "XkbLayout" "fr"

EndSection
```

De même, vous trouverez ci-dessous un exemple typique de section « InputDevice » pour une souris :

```
Section "InputDevice"
    Identifier "Souris 1"
    Driver      "Mouse"

# L'option suivante indique que la souris est
# une souris de type PS/2 :
    Option      "Protocol" "PS/2"

# L'option suivante indique le fichier spécial
# de périphérique à utiliser pour cette souris :
    Option      "Device" "/dev/psaux"

# L'option suivante indique le nombre de boutons
# de la souris :
    Option      "Buttons" "2"
```



```
# L'option suivante demande à X.org d'émuler
# le troisième bouton de la souris par clic
# simultanément des deux boutons :
Option      "Emulate3Buttons"

EndSection
```

Note : Pour les souris à molette connectées sur port PS/2, vous devrez choisir le protocole « IMPS/2 » au lieu du protocole « PS/2 » et indiquer que la souris a 5 boutons. De plus, vous devrez ajouter l'option « ZAxisMapping » dans la section « InputDevice » de votre souris, et lui donner la valeur « 4 5 ». Cette option permet de rediriger la rotation de la molette sur les événements des boutons 4 et 5 de la souris. De cette manière, vous pourrez utiliser votre molette dans toutes les applications qui gèrent ces deux boutons.

X-C-4-f - Sections « Device »

Les sections « Device » contiennent les descriptions des cartes graphiques utilisées. Ces descriptions n'indiquent absolument pas quel est le serveur X utilisé : elles ne contiennent qu'une description du matériel. Notez, encore une fois, qu'il est possible de définir plusieurs sections « Device », qui pourront être utilisées dans les sections « Screen » associant les cartes graphiques aux moniteurs.

Tout comme les sections « InputDevice », les sections « Device » doivent disposer d'un identificateur unique, afin de pouvoir les référencer dans les sections « Screen » qui les utilisent. Cet identificateur est introduit par le mot clé « Identifier » et suivi d'un nom indiqué entre guillemets.

Les sections « Device » peuvent contenir un certain nombre d'options permettant de caractériser la carte graphique et de préciser ses capacités (mémoire, rapidité, adresse sur le bus PCI, etc.). Toutes ces options sont bien entendu spécifiques aux différentes cartes graphiques, et ne seront pas décrites en détail ici.

L'option la plus importante est sans doute l'option « Driver », qui permet d'indiquer le pilote que le serveur X devra charger pour piloter cette carte graphique. C'est à ce niveau qu'il faut surtout ne pas se tromper, puisque la moindre erreur ferait échouer le démarrage du serveur X. En général, les pilotes portent le nom du chipset utilisé par la carte graphique, si bien qu'il n'est pas difficile de déterminer la valeur exacte à utiliser. X.org fournit également deux pilotes génériques vga et vesa, permettant respectivement de prendre en charge les cartes graphiques compatibles avec le standard VGA ou compatibles avec le standard VESA 2.0. Sachez cependant que ces pilotes ne disposent d'aucune accélération matérielle, et que leurs performances sont donc bien inférieures aux pilotes spécialisés pour chaque carte graphique.

Vous trouverez ci-dessous un exemple de section « Device » typique :

```
Section "Device"
    Identifier   "Carte 1"
    VendorName   "CHASTECH"
    BoardName    "TORNADO I7000"

    # Cette option indique au serveur X qu'il doit
    # charger le module i740_drv pour piloter cette
    # carte graphique :
    Driver       "i740"

    # Cette option indique la quantité de mémoire
    # vive dont cette carte dispose (en ko) :
    VideoRam     8192
EndSection
```

Note : Les options des sections « Device » ne sont pas très compliquées à utiliser. Cependant, quelques-unes peuvent poser des problèmes lorsqu'on réalise des configurations multitâches (c'est-à-dire des configurations disposant

de plusieurs écrans). Dans la majorité des cas, il faut installer plusieurs cartes graphiques dans le PC pour pouvoir utiliser plusieurs écrans. En général, on utilise une carte AGP et une ou plusieurs cartes PCI. Il va de soi qu'il faut que chaque pilote utilise la bonne carte, aussi faut-il indiquer, dans chaque section « Device », l'adresse de la carte décrite par la section. Cette adresse est définie à l'aide de l'option « BusID ». Cette option permet de retrouver la carte graphique sur les différents bus systèmes. Les cartes AGP utilisant le même mécanisme d'adressage que les cartes PCI, on utilisera la syntaxe suivante :

```
"PCI:bus:périphérique:fonction"
```

où bus est le numéro du bus PCI sur lequel se trouve la carte graphique, périphérique est le numéro du périphérique sur ce bus, et fonction est le numéro de la fonction à utiliser pour accéder à ce périphérique. Vous pouvez utiliser la commande `lspci` pour afficher les caractéristiques des différents bus PCI de votre machine et déterminer les valeurs utilisées par vos cartes graphiques.

Certaines cartes graphiques haut de gamme disposent de plusieurs contrôleurs vidéo et sont donc capables de gérer plusieurs écrans. Pour ces cartes graphiques, le principe de fonctionnement est le même : il faut écrire une section « Device » pour chaque contrôleur vidéo existant. Cependant, étant donné que toutes ces sections s'adressent à la même carte graphique, il est impossible de les distinguer par l'intermédiaire de l'option « BusID ». Dans ce cas, on utilise alors l'option « Screen », suivie du numéro du contrôleur vidéo que la section décrit. Ce numéro doit être donné juste après l'option « Screen », sans guillemets. La numérotation des contrôleurs vidéo commence à 0. C'est cette valeur qui est utilisée pour toutes les cartes graphiques ne disposant que d'un seul contrôleur vidéo.

X-C-4-g - Sections « Monitor »

Les sections « Monitor » contiennent les informations descriptives des écrans. Notez qu'il est possible de définir plusieurs sections « Monitor », pour plusieurs écrans potentiels. Cependant, chaque section « Screen » devra utiliser un moniteur et un seul.

Les sections « Monitor » sont certainement les sections les plus importantes du fichier `xorg.conf`. Elles contiennent en effet toutes les informations concernant les moniteurs et tous les paramètres des modes graphiques utilisés par ces moniteurs. Ces informations sont utilisées par le serveur X pour programmer les signaux vidéo que la carte graphique envoie au moniteur. Il est donc évident que chaque section « Monitor » est spécifique à un modèle de moniteur donné, et que la détermination des valeurs qui doivent y être écrites requiert une bonne connaissance du fonctionnement de ce moniteur et de ses caractéristiques physiques. Vous trouverez une description générale du fonctionnement des moniteurs dans les paragraphes qui suivent. Il est impératif de bien les assimiler si vous désirez créer ou modifier une section « Monitor », car si les informations qui y sont stockées sont erronées, le moniteur correspondant risque d'être gravement endommagé.

Si votre moniteur est un écran plat, le moniteur affiche directement les informations dans son mode vidéo natif. La configuration des modes graphiques au niveau de la carte graphique n'a donc que peu d'influence : seuls la résolution et le nombre de couleurs sont utilisés ici, et l'écran s'adapte aux signaux horizontaux et verticaux fournis par la carte graphique dynamiquement pour reconstituer l'image à afficher dans sa résolution native (moyennant une image plus ou moins nette et des couleurs plus ou moins précises). De ce fait, tous les réglages par défaut sont généralement valides, et la section « Monitor » peut donc être réduite à sa plus simple expression.

En revanche, pour les tubes cathodiques, les signaux envoyés par la carte graphique sont très importants. C'est réellement elle qui définit le mode graphique. Les informations temporelles des signaux horizontaux et verticaux, ainsi que le nombre de couleurs utilisées, sont essentiels pour définir le mode vidéo, tant en résolution qu'en nombre de couleurs utilisées, ou encore en fréquence de rafraîchissement de l'image. La définition des modes vidéo dans ce cas est plus complexe, sauf si les modes définis par défaut vous conviennent.

Le principe de fonctionnement des tubes cathodiques utilisés dans les moniteurs est très simple : à chaque instant, un faisceau d'électrons généré par un canon à électrons vient frapper un point d'une plaque phosphorescente située

juste derrière la vitre du moniteur. L'énergie cinétique des électrons du faisceau est transformée en lumière par le phosphore en ce point, qui reste ainsi lumineux tant que le faisceau frappe ce point. Comme un seul faisceau est utilisé pour tous les points de la plaque phosphorescente, il faut déplacer le faisceau de point en point pour pouvoir tous les éclairer. Cela a bien entendu pour conséquence que les points ne restent lumineux que pendant un temps très court. Cependant, nous percevons la luminosité de ce point pendant un temps beaucoup plus long, en raison de la rémanence des cellules de l'œil. Les points de l'écran semblent donc être tous allumés en même temps, bien qu'à chaque instant un seul d'entre eux soit atteint par le faisceau d'électrons. Nous voyons ainsi l'image complète du moniteur.

Le déplacement du faisceau d'électrons est assuré par un champ magnétique généré par des bobinages magnétiques à l'arrière du tube cathodique. Le faisceau balaye l'écran ligne par ligne, de gauche à droite (en regardant le moniteur de face). À chaque fois qu'il atteint le bord gauche, il retourne rapidement vers le début de la ligne suivante, sur le bord droit du moniteur. Ce retour se fait simplement en modifiant brusquement le champ magnétique qui dirige le faisceau. Il va de soi que le faisceau d'électrons est coupé pendant ce retour de balayage, car s'il ne l'était pas, on le verrait traverser l'écran de droite à gauche et perturber ainsi l'image. De plus, le champ magnétique n'est pas facilement contrôlable lorsque le retour du faisceau commence. Par conséquent, il est nécessaire que ce retour se fasse un certain temps après que le bord droit ait été éteint. Si ce n'était pas le cas, l'image serait déformée près des bords du moniteur. Le faisceau électronique continue donc de balayer l'écran sur une petite zone après le bord de l'image, sans émettre de lumière. Cette zone s'appelle le « blanking » horizontal.

De même, le faisceau électronique retourne rapidement en haut de l'écran dès qu'il en atteint le bas. Pour les mêmes raisons que pour les retours de balayage horizontal, le faisceau doit être éteint un peu avant et pendant la modification du champ magnétique lors des retours de balayage vertical. La zone ainsi parcourue par le faisceau après le bord de l'image s'appelle le « blanking » vertical.

Note : Il est possible d'utiliser d'autres méthodes de balayage que celle décrite ci-dessus. En particulier, les modes graphiques entrelacés ne suivent pas ce principe. Dans ces modes, seulement une ligne sur deux est balayée à chaque génération d'image. Les lignes paires sont d'abord affichées pour une première image, puis les lignes impaires sont affichées pour l'image suivante. Ces modes graphiques étaient utilisés lorsque les moniteurs n'étaient pas suffisamment rapides pour afficher une image complète sans que l'on puisse voir le balayage du faisceau d'électrons. De nos jours, il est déconseillé d'utiliser ces modes, car l'image a tendance à trembler du fait de l'écart de luminosité alternatif entre les lignes paires et impaires de l'écran.

Les limitations techniques de la technologie des tubes cathodiques imposent une limite supérieure pour la vitesse de balayage, et une limite inférieure pour la durée des retours de balayage horizontal et vertical, ainsi que pour la durée minimale du blanking. Les caractéristiques techniques des tubes sont donc souvent spécifiées en termes de fréquences de balayage et de durées des signaux de synchronisation. Les vieux moniteurs ne pouvaient supporter qu'un nombre limité de fréquences de balayage, que l'on qualifiait alors de « fixes ». De nos jours, les écrans multisynchrones acceptent toutes les fréquences comprises dans une plage de valeur, et s'adaptent donc aux fréquences des signaux émis par les cartes graphiques.

La fréquence de balayage vertical utilisée dans un mode graphique donné constitue le taux de rafraîchissement de l'écran, c'est-à-dire le nombre de fois que l'image de l'écran est affichée par seconde. Il est évident qu'un taux de rafraîchissement trop faible ne permet pas d'avoir une image agréable à regarder, car elle semble trembler ou clignoter. Un tube cathodique de bonne qualité est donc un tube qui permet d'utiliser de hautes fréquences verticales. Comme il est nécessaire d'afficher toutes les lignes pour afficher une image complète, il va de soi qu'un bon tube doit également être capable d'afficher les lignes rapidement, et donc accepter des fréquences de balayage horizontal élevées. Typiquement, les fréquences de balayage horizontal sont environ mille fois plus élevées que les fréquences de balayage vertical, puisque chaque ligne peut contenir environ mille pixels.

La sensibilité au taux de rafraîchissement dépend des personnes et de l'emploi que l'on fait du tube. Les tubes destinés à être regardés de loin peuvent utiliser des fréquences beaucoup plus faibles que les tubes de proximité. Ainsi, les télévisions affichent 25 images par seconde (en fait, elles affichent 50 demi-images par seconde, car elles utilisent l'entrelacement). En revanche, ce taux de rafraîchissement est très insuffisant pour les moniteurs d'ordinateurs, où la limite inférieure est de 60 Hz. Le standard VESA exige des taux de 72 Hz ou plus pour que l'image soit agréable à regarder, mais de nombreux écrans supportent facilement 85 Hz ou plus de nos jours. Certaines personnes ne sont cependant pas dérangées par des taux aussi faibles que 60 Hz. Il est bon de savoir quel taux de

rafraîchissement vous convient, car un balayage insuffisant peut fatiguer vos yeux, même sans que vous en soyez conscient. Et cette fatigue se traduit toujours un jour ou un autre par des maux de tête inexplicables... Ne vous vantez donc pas de voir le balayage, c'est un grave handicap par rapport à ceux qui ne le voient pas et qui donc ont moins de maux de tête que vous !

Vous pouvez utiliser la méthode suivante pour déterminer si le taux de rafraîchissement est suffisant pour vous. Commencez par afficher une image blanche sur la totalité de la surface visible de votre moniteur. Puis réglez la luminosité du moniteur à son maximum. Placez-vous ensuite de biais par rapport à votre moniteur, et regardez en face de vous. Concentrez ensuite votre attention sur l'image de l'écran, sans le regarder directement (regardez-le « en coin »). Normalement, vous devez percevoir l'image avec les cellules latérales de la rétine, qui donnent une image moins précise que les cellules centrales, mais qui sont plus sensibles aux mouvements (ces cellules nous servent en quelque sorte d'antennes). Ces cellules sont donc plus aptes à percevoir les variations rapides de luminosité de l'image dues au balayage du faisceau électronique. Si l'image semble clignoter, c'est que le taux de rafraîchissement de l'écran est insuffisant. Sinon, vous tenez le bon taux, ne le lâchez pas.

Certaines caractéristiques techniques des tubes cathodiques influent sur les autres et permettent donc de modifier le taux de rafraîchissement. Il est évident que si la durée du retour du faisceau est importante, l'écran ne peut pas générer beaucoup d'images par seconde. Par conséquent, plus la durée du retour de balayage est longue, moins bon peut être le taux de rafraîchissement. Il en va de même pour la durée du blanking. Les bons écrans disposent donc souvent de retours de balayage rapides et de courtes durées de blanking. Mais diminuer ces durées nécessite une électronique plus précise au niveau du contrôle des champs magnétiques dirigeant le faisceau d'électrons. Il est également évident que la résolution du mode graphique influe sur le taux de rafraîchissement. Si la résolution est élevée, il y a plus de pixels à afficher au total sur chaque ligne de l'écran. À vitesse de balayage des pixels fixée, le temps nécessaire pour balayer une ligne est donc plus grand. Et comme il y a plus de lignes à afficher, le temps nécessaire pour générer une image complète est plus grand, et le taux de rafraîchissement est plus faible. Donc, si le test décrit dans le paragraphe précédent vous indique que votre taux de rafraîchissement est trop faible, vous pouvez tenter de baisser la résolution.

Note : Notez que la profondeur de couleur utilisée n'intervient absolument pas dans le taux de rafraîchissement. Cette profondeur n'est un facteur limitant qu'au niveau de la carte graphique, qui peut ne pas avoir assez de mémoire pour afficher un mode de grande résolution avec toutes les couleurs désirées ou ne pas avoir une électronique assez rapide pour traiter toutes les données à une vitesse raisonnable.

Le signal vidéo émis par la carte graphique contient toutes les informations dont le moniteur a besoin pour générer l'image. Ces informations spécifient bien entendu l'intensité du faisceau d'électrons, mais aussi les signaux de synchronisation qui contrôlent les retours de balayage horizontal et vertical. Ce sont les informations de synchronisation que les sections « Monitor » du fichier `xorg.conf` contiennent. On comprend donc l'importance de ces informations pour la qualité de l'image, aussi bien en terme de stabilité qu'en termes de taux de rafraîchissement.

Les sections « Monitor » sont constituées grosso modo de deux parties. La première partie contient les caractéristiques générales de l'écran, telles que son nom, son modèle et le nom de son fabricant. Elle contient également les plages de fréquences de balayage horizontal et vertical. La deuxième partie contient les définitions des modes graphiques, à raison d'une ligne par mode graphique utilisable. La documentation de X.org appelle ces lignes des « lignes de mode » (« modelines » en anglais). Il est évident que les informations fournies dans les lignes de mode sont très sensibles, et déterminent tous les paramètres du mode graphique : résolution, position et taille de l'image sur l'écran, taux de rafraîchissement.

La première partie d'une section « Monitor » ressemble à ceci :

```
Section "Monitor"
    Identifier "Moniteur 1"
    # Marque et modèle du moniteur (informations facultatives) :
    VendorName "SONY"
    ModelName   "MULTISCAN 100ES"
    # Plage de fréquence horizontale (information capitale) :
    HorizSync   30-70
    # Plage de fréquence verticale (information capitale) :
    VertRefresh 50-120
```

Le mot clé « **Identifier** » permet de donner un nom à la section « **Monitor** ». Ce sera ce nom que l'on utilisera plus loin dans la section « **Screen** » pour indiquer que l'on désire utiliser ce moniteur. Les mots clés « **VendorName** » et « **ModelName** » permettent de donner la description du moniteur. Les informations stockées ici n'ont aucun effet sur la configuration du moniteur et sont simplement indicatives. En revanche, les plages de fréquences horizontales et verticales indiquées respectivement après les mots clés « **HorizSync** » et « **VertRefresh** » sont d'une importance capitale. Elles donnent les plages de fréquences auxquelles le moniteur peut travailler, et servent à contrôler la validité des lignes de mode données dans la deuxième partie de la section « **Monitor** ». Il est donc important de donner ici les valeurs exactes, que vous trouverez normalement dans la fiche technique de votre moniteur (cette fiche est souvent imprimée à la fin du mode d'emploi). Pour les moniteurs multisynchrones, les plages de fréquences utilisables peuvent être données par leurs fréquences minimale et maximale, séparées d'un tiret. Les fréquences fixes peuvent être spécifiées simplement avec une seule valeur. Il est possible de spécifier plusieurs fréquences fixes et plusieurs plages de fréquences en les séparant par des virgules. L'unité utilisée pour les fréquences horizontales est le kilohertz, et celle utilisée pour les fréquences verticales est le Hertz.

La deuxième partie de la section « **Monitor** » contient les lignes de mode, à raison d'une ligne de mode par mode graphique que le moniteur est capable d'afficher. En général, une ligne de mode ressemble à ceci :

```
Modeline "1024x768" 92.96 1024 1064 1240 1352 768 768 778 802
```

mais il existe cette une autre syntaxe, beaucoup plus claire, car elle différencie les différentes valeurs données dans la ligne précédente :

```
Mode "1024x768"  
    DotClock 92.96  
    HTimings 1024 1064 1240 1352  
    VTimings 768 768 778 802  
EndMode
```

La première valeur indiquée entre guillemets dans les lignes de mode est le nom du mode graphique. Comme on le verra plus tard, c'est ce nom qui est utilisé pour spécifier les modes graphiques utilisables dans la section « **Screen** ». La deuxième valeur est la fréquence de base des signaux émis par la carte graphique. Cette vitesse est exprimée en MHz et représente le nombre de pixels par seconde que le faisceau électronique du moniteur balaye. Les quatre valeurs suivantes fixent les paramètres horizontaux du mode graphique : résolution horizontale, pixel à partir duquel le signal de synchronisation pour le retour de balayage horizontal commence, pixel auquel ce signal s'arrête et longueur totale de la ligne en pixel, blanking compris. De même, les quatre dernières valeurs fixent les paramètres verticaux du mode graphique, à savoir résolution verticale, ligne de début et ligne de fin du signal de synchronisation pour le retour de balayage vertical, et nombre de lignes total du mode graphique. On notera que les informations concernant le signal de synchronisation du balayage horizontal sont toutes multiples de 8. Cette contrainte est imposée par le matériel de la plupart des cartes graphiques. D'autre part, l'unité utilisée pour ces valeurs est le pixel. Il faut donc se baser sur la fréquence de base indiquée au début de la ligne de mode pour les convertir en temps. Les informations concernant le signal de synchronisation vertical, quant à elles, sont exprimées en nombre de lignes. Elles peuvent ne pas être multiples de 8. La conversion en temps se calcule cette fois à partir de la fréquence de base et de la longueur totale d'une ligne horizontale pour ce mode graphique. Enfin, il est possible de rajouter des options complémentaires à la fin des lignes de mode pour préciser le fonctionnement du mode graphique, à l'aide du mot-clé « **Flags** ». L'option la plus courante étant sans aucun doute « **Interlace** », qui permet de créer des modes entrelacés. L'option « **Doublescan** » quant à elle permet de faire en sorte que chaque ligne est affichée deux fois de suite. Elle permet de créer des modes graphiques de faible résolution, qui utilisent des fréquences de balayage trop faibles pour les moniteurs actuels. Vous pourrez trouver les autres options dans la page de manuel xorg.conf.

La section « **Monitor** » se termine, comme toute section du fichier `xorg.conf`, par une ligne de fin de section :

```
EndSection
```

Comme vous vous en êtes aperçu, les données stockées dans les lignes de mode sont assez techniques. Quelques explications complémentaires ne seront donc pas de trop pour comprendre l'influence de ces paramètres et pour créer de nouvelles lignes de mode.

Pour commencer, la fréquence de base de la ligne de mode doit évidemment être gérée par la carte graphique. Vous pouvez déterminer la fréquence de base maximale en regardant les informations affichées par le serveur X adapté à cette carte lorsqu'il démarre. Cette information est toujours juste, même si les lignes de mode écrites dans votre fichier `xorg.conf` ne conviennent pas et que le serveur X ne démarre pas correctement. La ligne contenant cette fréquence maximale est semblable à celle-ci :

```
Maximum allowed dot-clock: 163.000 MHz
```

Cette fréquence de base peut être générée par la carte graphique, mais il n'est pas certain qu'elle soit supportée par le moniteur. En effet, les moniteurs ne sont pas capables de gérer les fréquences de base au-delà d'une certaine limite. La gamme de fréquences qu'ils acceptent est ce que l'on appelle leur bande passante. La fréquence de base doit donc être dans cette bande passante, c'est-à-dire qu'elle doit être inférieure à la fréquence maximale que le moniteur peut gérer. La bande passante de votre moniteur est normalement indiquée dans sa fiche de spécifications techniques. Si toutefois ce n'est pas le cas, vous pouvez vous faire une idée de la fréquence la plus élevée de cette bande passante en regardant la résolution maximale qu'il peut afficher et en consultant ce tableau :

Tableau 10-1. Fréquence maximale des moniteurs

Résolution	Fréquence maximale
640x480	25
800x600	36
1024x768	65
1280x1024	110
1600x1200	185

Les valeurs des fréquences indiquées dans ce tableau sont les plus petites valeurs constatées pour l'ensemble des moniteurs gérés par X.org. On peut donc supposer que votre moniteur gère ces fréquences, et certainement même de plus hautes.

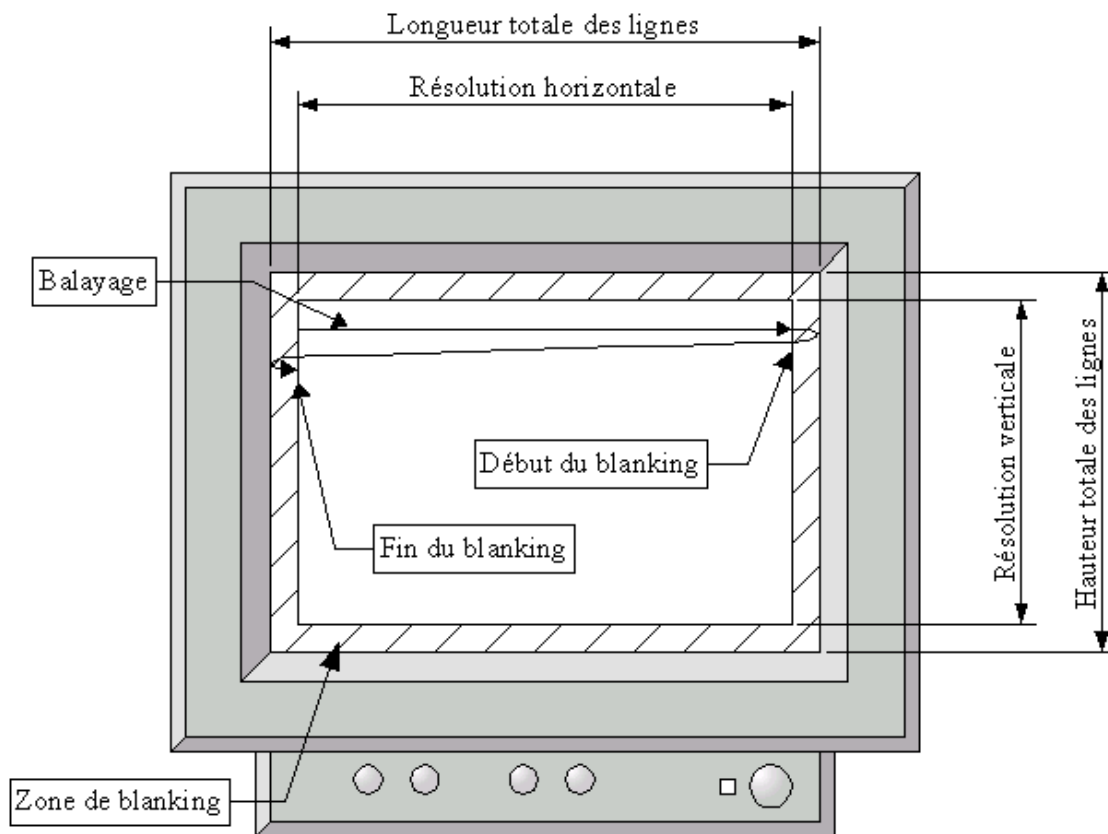
La résolution du mode graphique est donnée par la première des quatre valeurs des paramètres horizontaux et verticaux dans la ligne de mode. Ainsi, dans la ligne de mode exemple donnée ci-dessus, la résolution horizontale est de 1024 et la résolution verticale est de 768. Sachez que vous êtes parfaitement libre d'utiliser des résolutions non standards, en écrivant les lignes de mode correspondantes. Dans l'exemple donné ci-dessus, il s'agit simplement du mode graphique 1024x768. Notez, encore une fois, que la profondeur de couleur n'intervient absolument pas dans les paramètres du moniteur. Ils n'interviennent que dans les composants de la carte graphique qui sont chargés de générer les signaux vidéo pour le moniteur.

La quatrième valeur des paramètres horizontaux donne la longueur totale des lignes physiques dans ce mode. Cette longueur comprend bien entendu la partie visible de l'image, mais également la partie due au blanking horizontal. Rappelons que le blanking est la partie balayée par le faisceau alors qu'il est éteint, ce qui comprend la partie balayée pendant la durée du signal de synchronisation horizontal. Dans l'exemple donné ci-dessus, le blanking a une longueur égale à 1352-1024 pixels, soit 328 pixels. La quatrième valeur des paramètres verticaux fixe de même le nombre de lignes total du mode, en tenant compte des lignes du blanking vertical.

Comme vous pouvez le constater dans la ligne de mode exemple, le pixel auquel le signal de synchronisation commence n'est pas le dernier pixel visible. En effet, il y a 40 pixels d'écart entre le dernier pixel visible (en l'occurrence, le pixel 1024) et le pixel auquel le signal de synchronisation horizontal commence (le pixel 1064). Cet espace fait partie du blanking, il sert de marge pour éviter les perturbations dues au retour de balayage et pour positionner l'image sur l'écran. Cette zone noire se trouve au-delà du bord droit de l'image affichée. Elle est suivie par la zone qui serait balayée pendant la durée du retour de balayage si le faisceau poursuivait sa route au lieu de revenir sur le bord gauche. Cette zone se termine au pixel indiqué par la troisième valeur (1240 dans notre exemple). Ce pixel est donc toujours positionné sur le point le plus à droite que le moniteur peut afficher. Par conséquent, la taille de la partie du blanking situé à la droite de l'image est égale au numéro du pixel où le signal de synchronisation horizontal se termine moins le nombre de pixels affichés. Dans notre exemple, cette partie de blanking a une taille de 1240-1024 pixels, soit 216 pixels. La taille de la partie du blanking horizontal situé à la gauche de l'écran est donc logiquement égale à la longueur totale de la ligne (1352) moins le pixel où le signal de synchronisation se termine (1240), ce qui

donne 112 pixels. Vous voyez ainsi que l'on peut décaler l'image vers la gauche en augmentant les valeurs de départ et de fin du signal de synchronisation. Inversement, on peut décaler l'image vers la droite en diminuant ces valeurs.

Le même raisonnement peut être appliqué aux paramètres verticaux du mode graphique. Le blanking vertical comprend les lignes balayées avant et après le signal de synchronisation, ainsi que les lignes balayées pendant le temps où ce signal est généré. La dernière ligne de ce signal est, encore une fois, la ligne la plus basse que le moniteur peut afficher. On peut donc déplacer l'image vers le haut ou vers le bas en jouant sur les valeurs de départ et de fin du signal de synchronisation vertical dans les paramètres verticaux du mode graphique.



Paramètres des lignes de mode

En augmentant le nombre de pixels physiques pour chaque ligne et en conservant la même résolution, la largeur de l'image visible est réduite. En effet, la proportion de la partie visible de l'image sur la largeur totale de l'écran est d'autant plus faible que le nombre de pixels physiques des lignes est grand. Le même raisonnement peut être appliqué sur les paramètres verticaux. L'image peut donc être agrandie ou réduite horizontalement et verticalement en jouant sur la longueur totale des lignes et sur la hauteur totale de l'image. Si l'on veut que l'image reste centrée, il faut également décaler les points de début et de fin des signaux de synchronisation, pour que la proportion du blanking placée de part et d'autre de l'image reste constante.

Il faut bien comprendre que la modification des paramètres de synchronisation et de la longueur totale de la ligne est soumise à de fortes contraintes. Il faut s'assurer que les durées des signaux de synchronisation et du blanking restent dans les limites de ce que peut accepter le moniteur. De plus, le nombre de points balayés au total est évidemment le produit de la longueur totale des lignes et du nombre de lignes total du mode. À fréquence de base fixe, cela revient à définir le taux de rafraîchissement, puisque plus il y a de pixels à balayer dans chaque image, moins il est possible d'afficher d'images par seconde. Cela n'est pas gênant si vous pouvez augmenter la fréquence de base. Par contre, si vous avez atteint la limite de votre carte graphique, ou la limite de bande passante de votre moniteur, vous serez limité dans le taux de rafraîchissement si vous devez réduire les dimensions de l'image affichée.

Vous devez vous en douter à présent : l'écriture d'une ligne de mode est un sport très difficile. La technique à utiliser est itérative. Vous pouvez tourner en rond pendant un certain temps, si vous ne savez pas vous y prendre. La première des choses à faire est de choisir la dimension de l'image que vous désirez obtenir. Cette dimension va fixer la largeur

totale et le nombre de lignes total qui seront utilisés pour ce mode graphique. Vous pouvez toujours partir des lignes de mode utilisées pour les modes graphiques VESA pour un moniteur semblable au vôtre. Une fois que vous aurez ces valeurs, vous pourrez essayer de faire monter le taux de rafraîchissement en augmentant la fréquence de base. Vous aurez peut-être à redimensionner et à recentrer l'image sur le moniteur. Les quatre contraintes à respecter lorsque vous augmentez la fréquence de base sont bien entendu la durée du blanking horizontal et vertical, la durée des signaux de synchronisation et les plages de fréquences horizontales et verticales acceptées par votre moniteur. Il est probable que vous aurez à accroître l'écart entre les valeurs de début et de fin des signaux de synchronisation, car leur durée peut baisser dangereusement lorsque la fréquence de base augmente. Malheureusement, cela peut vous limiter dans la possibilité de centrer l'image sur votre moniteur. Une image dont les bords ne sont pas rectilignes ou pas nets est un signe indiquant que le signal de synchronisation n'est pas suffisamment long. Si vous ne vous en sortez pas, c'est que vous essayez d'atteindre un taux de rafraîchissement trop élevé.

Si vous désirez fabriquer un mode graphique de résolution non standard, vous ne trouverez pas forcément une ligne de mode pouvant servir de point de départ. Dans ce cas, il va vous falloir créer cette ligne de mode ex-nihilo. Vous trouverez dans le fichier de documentation VideoModes.doc de X.org les explications permettant de créer une ligne de mode correcte à partir des caractéristiques de votre moniteur. Pour les moniteurs multisynchrones, la technique suivante peut être utilisée.

La première étape est de trouver la bande passante de votre moniteur et les plages de fréquences horizontales et verticales utilisables par votre moniteur. Ces données sont très souvent fournies dans la documentation du moniteur. Par exemple, pour un moniteur Sony Multiscan 100ES, ces plages de fréquences sont les suivantes :

	Horizontales (kHz)	Verticales (Hz)
Fréquence minimale	30	50
Fréquence maximale	70	120

La deuxième étape est de choisir la résolution que l'on désire utiliser. Supposons que l'on veuille utiliser une résolution intermédiaire entre 1024x768 et 800x600 sur ce type d'écran 15 pouces. Prenons par exemple 904x678. Notez que la résolution horizontale est divisible par 8, et que la résolution verticale est égale aux trois quarts de cette dernière. Ce rapport permet simplement de conserver la proportion de 4/3 des écrans d'ordinateur. Pour pouvez cependant choisir un autre rapport si vous le désirez.

Vous devez ensuite déterminer les longueurs totales des lignes en pixels et le nombre total de lignes, blanking compris. C'est cette partie qui est la plus difficile, elle nécessite de connaître les durées minimales des signaux de synchronisation horizontaux et verticaux. Ces données sont hélas rarement fournies par les fabricants. Heureusement, on peut s'en passer en pratique, car tous les écrans cathodiques utilisent la même technologie. En général, la longueur totale d'une ligne est égale à la résolution horizontale divisée par 0,8, et le nombre total de lignes est égal à la résolution verticale augmentée de 5 %. Pour notre mode graphique, le calcul est donc le suivant :

```
longueur totale des lignes = 904 pixels / 0,8 = 1130 pixels
hauteur totale de l'image = 678 lignes * (1+5/100) =
678 lignes * 1,05 = 711,9 lignes.
```

La longueur totale des lignes n'est pas multiple de 8, nous devons donc l'arrondir à 1136. De même, la hauteur totale de l'image doit être arrondie à 712 lignes.

Nous disposons à présent des premières et dernières valeurs des paramètres horizontaux et verticaux de la ligne de mode. Les paramètres intermédiaires, qui fixent le début et la fin des signaux de retour de balayage, doivent être choisis de telle sorte que ces signaux commencent un peu après le début du blanking et se terminent un peu avant le début de la génération de l'image suivante. En pratique, on peut couramment considérer qu'une marge de 32 à 40 pixels est suffisante pour les signaux de synchronisation horizontaux, et qu'il faut 0 à 10 lignes pour les signaux de synchronisation verticaux. L'écart entre le début et la fin de ces signaux doit être suffisamment grand pour garantir leurs durées minimales exigées (mais a priori inconnues) par votre moniteur. Dans notre exemple, nous pouvons donc choisir les paramètres suivants :

```
HTimings 904 944 1096 1136
VTimings 678 683 707 712
```

Il reste à déterminer la fréquence de base à utiliser. Le but est bien entendu d'obtenir le taux de rafraîchissement le plus élevé possible, donc la fréquence de base la plus élevée, tout en restant dans les limites de la carte graphique et du moniteur. La contrainte la plus forte en général est la fréquence maximale de balayage horizontal.

Dans notre cas, l'écran est capable de travailler à 70 kHz au plus, c'est-à-dire d'afficher 70000 lignes par seconde. Comme notre mode graphique utilise 712 lignes physiques au total, cela nous donne un taux de rafraîchissement de 70000/712, soit environ 98 images par seconde. Bien entendu, il faut contrôler que cette fréquence est bien dans la plage de fréquences verticales du moniteur. C'est bien le cas ici, puisqu'il peut supporter des fréquences allant jusqu'à 120 Hz. En général, il faut prendre la plus petite de ces valeurs comme taux de rafraîchissement initial.

Une fois le taux de rafraîchissement déterminé, nous devons calculer la fréquence de base à utiliser. Pour afficher 98 images de 710 lignes de 1125 pixels par seconde, nous calculons cette fréquence de la manière suivante :

```
98 * 712 * 1136 = 79,26 MHz
```

Encore une fois, il faut contrôler que la carte graphique peut générer un signal de cette fréquence et que celle-ci est bien inférieure à la bande passante du moniteur. Dans le cas contraire, il faudrait encore une fois se limiter à la plus petite de ces valeurs. La plupart des cartes graphiques récentes peuvent monter au moins jusqu'à 150MHz, et l'écran Sony Multiscan 100ES a une bande passante bien supérieure à cette valeur. Nous pouvons donc conserver cette fréquence de base, et obtenons donc la ligne de mode suivante :

```
Mode "904x678"
  DotClock 79.26
  HTimings 904 944 1096 1136
  VTimings 678 683 707 712
EndMode
```

Cette ligne de mode a peu de chances d'être satisfaisante sans retouches, mais elle peut servir de point de départ pour la recherche de la ligne de mode idéale. En général, les paramètres étant choisis proches de l'optimum, l'image sera bien trop grande ou déformée sur les bords. Vous devrez donc faire des ajustements manuels pour vous rapprocher, petit à petit, d'une ligne de mode utilisable. Vous pourrez utiliser l'utilitaire xvidtune pour cela. Cet utilitaire est un utilitaire fourni avec X.org, qui permet de modifier les dimensions et la position de l'image interactivement. Cet utilitaire fonctionne directement sous XWindow, et permet donc d'avoir un aperçu de l'image avec les paramètres modifiés. Vous trouverez une description plus détaillée de la manière d'utiliser xvidtune dans la [Section 10.3.6](#).

Seul le redimensionnement de l'image modifie le nombre total de lignes et leur longueur totale. En particulier, la réduction de la taille de l'image accroît la taille totale des lignes et leur nombre, ce qui entraîne la baisse du taux de rafraîchissement. À chaque fois que vous réduisez l'image, vous pourrez recalculer le taux de rafraîchissement et la fréquence de base nécessaire pour atteindre ce taux, tout en veillant à ne dépasser ni les limites de votre carte graphique ni celles de votre moniteur.

Ainsi, après quelques modifications de la ligne de mode dans xvidtune et réajustements de la fréquence de base, la ligne de mode suivante peut être obtenue :

```
Mode "904x678"
  DotClock 79.82
  HTimings 904 920 1104 1144
  VTimings 678 679 710 712
EndMode
```

Cette ligne de mode permet d'obtenir un affichage correct en 904 par 678, avec un taux de rafraîchissement de 97.99 Hz, ce qui est plus qu'honorable. On constate que la première ligne de mode utilisait une durée de balayage vertical trop petite, ce qui provoquait des déformations sur les bords de l'image.

Si vous voulez approfondir le sujet, vous pouvez lire les formules mathématiques donnant les relations entre les différents paramètres techniques influant sur la génération des images dans l'[Annexe C](#). Ces formules sont données à titre indicatif, elles ne vous donneront pas les caractéristiques techniques de votre écran. Encore une fois, je ne saurais trop vous recommander d'utiliser les outils fournis avec votre distribution...

X-C-4-h - Sections « Modes »

Généralement, les lignes de mode sont définies pour un moniteur donné, car l'apparence de l'image dépend des caractéristiques physiques de chaque moniteur. Par conséquent, ces lignes sont définies la plupart du temps dans les sections « Monitor ». Cependant, il peut être utile de regrouper toutes les lignes de mode d'un type de moniteur donné dans une section à part, afin de structurer et d'organiser le fichier `xorg.conf`.

Pour cela, on écrira des sections « Modes », qui n'auront donc pas d'autre rôle que de contenir une ou plusieurs lignes de mode, et qui seront référencées dans les sections « Monitor » des moniteurs qui désirent utiliser ces lignes de mode. L'intérêt de regrouper toutes les lignes de mode dans une section « Modes » apparaît clairement dès que plusieurs sections « Monitor » utilisent les mêmes lignes de mode. Ce peut être le cas pour des moniteurs d'un même fabricant ou possédant des caractéristiques techniques similaires.

Le format des sections « Monitor » est très simple, puisqu'elles ne contiennent qu'un identificateur, introduit comme à l'accoutumée par le mot-clé « Identifier » suivi du nom de la section entre guillemets, et les lignes de mode elles-mêmes. La syntaxe de ces lignes est exactement la même que celle qui est utilisée dans la section « Monitor » et ne sera donc pas décrite plus en détail ici. Vous trouverez ci-dessous un exemple de section « Modes » :

```
# Exemple de section Modes :
Section "Modes"
    Identifier "Modes Standards"
    Modeline "640x480"      50.00 640 656 720 832 480 480 489 501
    Modeline "800x600"      72.80 800 816 928 1040 600 600 610 626
    Modeline "1024x768"     92.00 1024 1076 1252 1344 768 772 782 806
EndSection
```

X-C-4-i - Sections « Screen »

Les sections « Screen » permettent de regrouper les moniteurs et les cartes graphiques pour effectuer l'affichage des écrans d'un display, et d'indiquer les modes graphiques que l'on désire utiliser parmi tous les modes que le moniteur indiqué sait gérer, pour chaque profondeur de couleur.

Les sections « Screen » sont, comme les sections « Monitor », constituées de deux parties. La première partie spécifie les informations générales de l'écran, à savoir essentiellement la carte graphique et le moniteur à utiliser. La deuxième partie quant à elle permet de préciser, pour chaque profondeur de couleur utilisable avec cet écran, les résolutions des modes graphiques utilisables ainsi que les paramètres des résolutions virtuelles. Voici trouverez ci-dessous un exemple de section « Screen » :

```
Section "Screen"
    Identifier "Ecran 1"
    Device      "Carte 1"
    Monitor     "Moniteur 1"
    DefaultDepth 24
    SubSection "Display"
        Depth 24
        Modes "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth 16
        Modes "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth 8
        Modes "1024x768" "800x600" "640x480"
    EndSubSection
EndSection
```

L'option « Identifier » permet, comme d'habitude, de nommer la section « Screen » courante. Le nom doit être indiqué entre guillemets, à la suite du mot clé « Identifier ». C'est ce nom qui sera utilisé dans les sections « ServerLayout » pour référencer les écrans du display. Le mot-clé « Device » indique la section « Device » à utiliser pour obtenir les

informations sur la carte graphique qui gère l'affichage de cet écran. Il doit être suivi du nom qui a été spécifié après le mot-clef « Identifier » dans la section « Device » correspondante. De même, le mot-clef « Monitor » indique la section « Monitor » à utiliser pour obtenir les informations sur le moniteur courant. Enfin, le mot-clef « DefaultColorDepth » indique la profondeur de couleur utilisée par défaut au démarrage de XWindow. Comme nous le verrons plus loin, une autre profondeur de couleur que celle-ci pourra être utilisée en passant un argument sur la ligne de commande du serveur X.

Comme vous pouvez le voir, la section « Screen » contient une ou plusieurs sous-sections « Display ». Ce sont ces sous-sections qui indiquent les paramètres de l'écran spécifiques à chaque profondeur de couleur utilisable. Les sous-sections « Display » contiennent les informations suivantes :

- la profondeur de couleur considérée, exprimée en bits par pixels après le mot-clef « Depth » ;
- la liste des modes graphiques utilisables avec cette profondeur de couleur, référencés par le nom de leur ligne de mode, après le mot-clef « Modes ». Plusieurs modes peuvent être spécifiés, en donnant leurs noms respectifs, séparés par des espaces.
- des informations complémentaires concernant les résolutions virtuelles pour les modes dans lesquels la résolution logique de l'écran est supérieure à la résolution réellement utilisée. Ces paramètres indiquent la résolution logique de l'écran virtuel, ainsi que le positionnement de l'image affichée par l'écran physique dans cet écran virtuel. Vous trouverez de plus amples informations sur ces options dans la page de manuel xorg.conf.

X-C-4-j - Sections « ServerLayout »

Le dernier type de section, qui regroupe toutes les informations des sections vues précédemment, est le type des sections « ServerLayout ». Comme leur nom l'indique, les sections de ce type fournissent les informations concernant la disposition des écrans d'un display. Cependant, dans la plupart des cas, les displays n'ont qu'un seul écran, et il est évident que la disposition de cet écran est triviale dans ce cas.

Il peut exister plusieurs sections « ServerLayout » dans le fichier `xorg.conf`. En effet, ce fichier peut être utilisé par plusieurs serveurs X qui peuvent tourner simultanément sur la même machine (que l'on ait plusieurs cartes graphiques ou non). Chaque serveur X peut donc utiliser une section « ServerLayout ». La section utilisée par défaut est la première que le serveur X trouvera dans le fichier `xorg.conf`, mais il est possible de lui demander d'en utiliser une autre à l'aide d'une option en ligne de commande. Toutefois, encore une fois, la plupart des machines ne disposent que d'un seul écran, et même si l'on lance plusieurs serveurs X, il n'y a ici besoin que d'une seule section « ServerLayout ».

Les sections « ServerLayout » indiquent surtout au serveur X quels sont les périphériques d'entrée à utiliser (au moins un clavier et une souris) en plus de l'écran ou les écrans du display. Ce sont donc ces sections qui définissent complètement un display donné. L'exemple donné ci-dessous présente une section « ServerLayout » simple :

```
Section "ServerLayout"
    Identifier "Serveur 1"
    Screen "Ecran 1"
    InputDevice "Clavier 1" "CoreKeyboard"
    InputDevice "Souris 1" "CorePointer"
    Option "BlankTime" "5"
EndSection
```

Les sections « ServerLayout » disposent d'un nom qui permet de les identifier. Ce nom est le nom qui sera utilisé dans la ligne de commande du serveur X si plusieurs sections « ServerLayout » sont définies. Il doit être indiqué, comme d'habitude, à la suite du mot-clef « Identifier », entre guillemets.

Les sections « ServerLayout » doivent ensuite indiquer au moins un écran pour le display qu'elles décrivent. Les écrans sont introduits à l'aide du mot-clef « Screen », suivi de l'identificateur de la section « Screen » à utiliser. Si plusieurs écrans doivent être utilisés, il est possible de spécifier la position des écrans les uns par rapport aux autres à l'aide d'options complémentaires, qui sont spécifiées à la suite de l'identificateur de l'écran. Ces options permettent de positionner les écrans relativement les uns aux autres, ou de manière absolue en précisant leurs coordonnées dans le système de coordonnées de l'écran virtuel. Les options les plus simples à utiliser sont les options `RightOf`,

LeftOf, Above et Below, qui permettent d'indiquer respectivement que l'écran courant est situé à droite, à gauche, au-dessus ou en dessous de l'écran dont l'identificateur est spécifié juste après. Par exemple, si l'on veut rajouter un deuxième écran à la section précédente, et que cet écran doit être situé à droite du premier écran, on ajoutera simplement la ligne suivante :

```
Screen "Ecran 2" RightOf "Ecran 1"
```

La section « `ServerLayout` » doit ensuite contenir au moins une référence à un clavier et une référence à un périphérique de pointage. Ces références sont introduites à l'aide du mot-clef « `InputDevice` », suivi de l'identificateur de la section « `InputDevice` » à utiliser. Le serveur X doit interpréter les événements provenant des différents périphériques d'entrée correctement. Il faut en particulier lui indiquer quels sont les périphériques principaux à l'aide de l'option `CorePointer` pour la souris principale et l'option `CoreKeyboard` pour le clavier principal. Seuls les périphériques d'entrée principaux seront utilisés en tant que clavier et en tant que souris.

Enfin, comme vous pouvez le voir dans l'exemple donné ci-dessus, il est possible d'utiliser des options spécifiques pour chaque section « `ServerLayout` ». Ces options sont les mêmes que celles utilisées dans la section « `ServerFlags` ». Vous pouvez donc ici spécifier une autre valeur que celles indiquées dans cette section, afin de préciser le comportement d'un serveur spécifique. Dans l'exemple donné ci-dessus, l'économiseur d'écran a été réglé pour s'activer au bout de 5 minutes.

X-C-5 - Informations utilisées lors du démarrage de X.org

Les informations stockées dans le fichier de configuration `xorg.conf` sont utilisées par X.org pour déterminer la manière dont l'affichage doit se faire. Lorsque le serveur X démarre, il commence par rechercher la section « `ServerLayout` » qu'il doit utiliser pour déterminer les principales options du display. Par défaut, il utilisera systématiquement la première section « `ServerLayout` » du fichier `xorg.conf`. Cependant, il est possible de lui demander d'en utiliser une autre, à l'aide de l'option de ligne de commande `-layout`, suivie de l'identificateur de la section à utiliser.

Le serveur X initialise alors tous les écrans du display et prend en charge les périphériques d'entrée. Pour chaque écran, il choisit la profondeur de couleur indiquée dans la variable « `DefaultDepth` » de la section « `Screen` » et tente de l'utiliser. Cependant, il est possible de choisir une autre profondeur de couleur à l'aide de l'option `-depth`, suivie de la profondeur de couleur à utiliser, spécifiée en bits par pixels. Remarquez que dans le cas des configurations à plusieurs écrans, il n'est possible de spécifier qu'une seule profondeur de couleur. Il faut donc que toutes les sections « `Screen` » des écrans utilisés contiennent une sous-section « `Display` » pour la profondeur de couleur choisie.

Une fois la profondeur de couleur déterminée, le serveur X choisit pour chaque écran la première résolution indiquée dans la liste des modes de la sous-section « `Display` » dont la profondeur de couleur correspond à celle utilisée. Il tente alors de passer l'adaptateur graphique dans ce mode graphique, en utilisant pour cela la ligne de mode correspondante à cette résolution pour piloter le moniteur.

Il est possible, lorsque XWindow fonctionne, de changer cette résolution avec les combinaisons de touches `CTRL+ALT+PLUS` et `CTRL+ALT+MINUS` (PLUS et MINUS étant les touches + et - du pavé numérique). Ces deux combinaisons de touches permettent respectivement de passer d'une résolution à la suivante ou à la précédente dans la liste des modes spécifiés dans la sous-section « `Display` ». Pour chaque mode choisi, la ligne de mode correspondante à la résolution de ce mode est utilisée pour générer les signaux à destination du moniteur.

Note : La résolution virtuelle de l'écran n'est pas modifiée lorsqu'on change de résolution physique. Elle est toujours égale à la plus grande des résolutions disponibles dans la section « `Display` » utilisée. Seule la résolution physique de l'affichage est modifiée. Vous pourrez déplacer la zone d'affichage en déplaçant la souris, et faire défiler ainsi tout l'écran.

Il est impossible de changer la profondeur de couleur utilisée sans redémarrer le serveur X. Les seuls changements acceptés sont ceux concernant les résolutions graphiques listées dans la sous-section « `Display` » pour la profondeur de couleur courante.

X-C-6 - Utilisation de xvidtune

xvidtune est un petit utilitaire permettant de régler les paramètres d'affichage de chaque mode graphique. Grâce à lui, vous pourrez ajuster la taille de l'image horizontalement et verticalement, ainsi que sa position par rapport aux bords du moniteur. xvidtune se lance simplement, en tapant son nom en ligne de commande :

```
xvidtune
```

Dès qu'il démarre, il commence par afficher une fenêtre d'avertissement, pour vous prévenir qu'un usage par une personne non avertie peut provoquer de sérieux dommages au moniteur. En effet, il permet de modifier les paramètres de synchronisation horizontale et verticale du moniteur pour ajuster l'image. Si vous choisissez de mauvais paramètres, vous pouvez sortir des spécifications techniques de votre moniteur, ce qui pourrait fort bien l'endommager. Fort heureusement, la plupart des moniteurs modernes disposent de mécanismes de sécurité qui les empêchent de générer l'image lorsque ces paramètres sont incorrects. Par conséquent, il est assez rare d'avoir des problèmes avec cet utilitaire, d'autant plus qu'il effectue des contrôles avant toute tentative douteuse. Vous pouvez donc valider en toute confiance, mais soyez malgré tout averti des risques potentiels. Rappelons que le serveur X peut être arrêté à n'importe quel moment à l'aide de la séquence de touches **CTRL+ALT+BACKSPACE**.

L'écran de xvidtune se compose de quatre parties. La partie supérieure gauche permet de régler les paramètres horizontaux du mode graphique, à savoir la largeur de l'image et sa position par rapport aux bords verticaux du moniteur. La partie supérieure droite permet de régler les paramètres verticaux, à savoir la hauteur et la position par rapport aux bords horizontaux du moniteur. La partie inférieure droite contient les informations essentielles sur le mode vidéo courant. Enfin, la partie inférieure gauche contient les boutons permettant de choisir les actions à effectuer.

Les paramétrages horizontaux et verticaux du mode graphique peuvent être modifiés très simplement. Les boutons « **Left** » et « **Right** » permettent de centrer l'image horizontalement, et les boutons « **Wider** » et « **Narrower** » permettent d'ajuster sa largeur. De même, les boutons « **Up** » et « **Down** » influent sur le centrage vertical, et les boutons « **Shorter** » et « **Taller** » ajustent sa hauteur.

Vous pouvez tester les modifications apportées à tout instant à l'aide du bouton « **Test** ». Il est même recommandé d'essayer régulièrement les paramètres choisis et de procéder pas à pas. Lorsque les paramètres conviennent, ils peuvent être appliqués immédiatement à l'aide du bouton « **Apply** ». Si vous désirez générer une ligne mode valide pour la configuration courante, par exemple afin de modifier votre fichier de configuration `xorg.conf`, vous pouvez cliquer sur le bouton « **Show** ». La ligne de mode sera affichée sur le terminal à partir duquel vous avez lancé xvidtune. Vous pourrez alors la recopier dans le fichier `xorg.conf` à la place de la ligne de mode du mode graphique en cours d'utilisation.

Il est possible de changer le mode graphique courant en cliquant sur les boutons « **Next** » et « **Prev** ». Ils permettent de passer en revue tous les modes graphiques qui ont été enregistrés dans le fichier `xorg.conf` par `xorgconfig` ou `xorgcfg`. Une fois que vous aurez ajusté tous les modes graphiques et que vous les aurez enregistrés à l'aide du bouton « **Apply** », vous pourrez quitter xvidtune en cliquant sur le bouton « **Quit** ». Notez que le bouton « **Apply** » valide les choix en cours, mais ne les enregistre pas dans le fichier `xorg.conf`. Vous devez reporter manuellement dans le fichier `xorg.conf` les paramètres du mode courant, tels qu'ils sont présentés par la commande « **Show** ».

X-D - Utilisation du pilote frame buffer du noyau

Si par malheur votre carte graphique n'est gérée par aucun des pilotes de X.org (cas relativement exceptionnel), vous serez sans doute obligé d'utiliser le pilote `vesa`. Ce pilote permet d'utiliser toutes les cartes compatibles avec le standard VESA 2.0 (c'est-à-dire la plupart des cartes graphiques, mais il existe des exceptions notables).

Il existe une alternative à ce pilote, qui se base sur les fonctionnalités « **frame buffer** » du noyau de Linux. Cette fonctionnalité permet d'utiliser Linux complètement en mode graphique, en fournissant un accès linéaire direct à la mémoire vidéo de la carte graphique grâce au fichier spécial de périphérique `/dev/fb0`. Il existe un pilote X.org pour le frame buffer du noyau, qui permet donc de démarrer le serveur X en s'appuyant complètement sur le noyau.

L'avantage du frame buffer du noyau est que même les consoles en mode texte feront leur affichage en mode graphique (cela est aussi un inconvénient du point de vue des performances). En revanche, vous ne pourrez pas changer de résolution une fois que le système aura démarré.

Pour accéder à la mémoire vidéo, le noyau se base également sur l'interface de programmation du standard VESA 2.0, qui est gérée par le BIOS de la plupart des cartes graphiques récentes. Cela signifie également que vous ne disposerez pas d'accélération matérielle en général, sauf pour quelques cartes graphiques courantes reconnues par le noyau.

X-D-1 - Configuration du noyau et installation du pilote

La mise en œuvre du pilote pour le frame buffer se fait évidemment dans la configuration du noyau. Les options à activer sont toutes dans le menu « Console drivers ». En plus de l'option « VGA text console », vous devez impérativement activer « Video mode selection support ». Cette option vous permettra de choisir le mode VESA à utiliser lors du démarrage de l'ordinateur. Vous devrez également cocher l'option « Support for frame buffer devices (EXPERIMENTAL) » (cette option ne vous sera proposée que si vous avez validé l'option « Prompt for development and/or incomplete code/drivers » du menu « Code maturity level options »). Les options suivantes du gestionnaire du frame buffer du noyau devront également être activées :

- « VESA VGA graphics console » (cette option permet d'utiliser un mode graphique VESA indiqué au démarrage pour l'affichage de la console) ;
- « Advanced low level driver options (NEW) » (cette option vous permet de préciser la structure de la mémoire vidéo dans les différents modes graphiques utilisés avec le frame buffer) ;
- « 8 bpp packed pixels support », « 16 bpp packed pixels support », « 24 bpp packed pixels support » et « 32 bpp packed pixels support » (ces options correspondent aux différentes structures de la mémoire vidéo qui sont utilisées par les modes graphiques VESA) ;
- « Select compiled-in fonts (NEW) » (cette option vous permet de choisir les polices de caractères qui seront utilisées par la console) ;
- « VGA 8x8 font » et « VGA 8x16 font » (ces deux polices sont les polices standards utilisées par la console).

Il faut ensuite vérifier que le fichier spécial de périphérique `/dev/fb0` a été créé par le programme d'installation de votre distribution. Si ce n'est pas le cas, vous devez le créer à l'aide de la commande `mknod`. Le numéro de périphérique majeur de ce fichier est 29. Le numéro mineur à utiliser est le numéro du périphérique. Par exemple, le fichier spécial de périphérique `/dev/fb0` porte les numéros 29 et 0, le fichier `/dev/fb1` porte les numéros 29 et 1, etc. Ces fichiers sont tous de type caractère, la ligne de commande pour créer un de ces fichiers est donc la suivante :

```
mknod fbn c 29 n
```

où `n` est le numéro du fichier spécial de périphérique à créer.

Il est également recommandé de créer un lien symbolique `/dev/fb` vers `/dev/fb0` afin d'assurer la compatibilité avec de vieux programmes utilisant ce nom pour accéder au fichier spécial de périphérique du gestionnaire du frame buffer du noyau.

Une fois ces opérations réalisées, vous devez compiler le noyau et l'installer, en suivant la méthode décrite dans la partie décrivant la compilation du noyau. Lors du redémarrage du système, vous pourrez passer l'option suivante au noyau pour préciser le mode graphique VESA à utiliser :

```
vga=mode
```

où `mode` est le numéro du mode graphique désiré. Les numéros valides sont indiqués dans le tableau donné ci-dessous :

Tableau 10-2. Numéros des modes graphiques VESA

Couleurs	Résolution				
	640x480	800x600	1024x768	1280x1024	1600x1200
256	769	771	773	775	796
32768	784	787	790	793	797
65536	785	788	791	794	798
16,8M	786	789	792	795	799

Si tout se passe correctement, votre système devrait démarrer dans le mode graphique indiqué et afficher le logo de Linux (un pingouin nommé « Tux », pour ceux qui ne le sauraient pas encore). Lorsque vous aurez déterminé le mode graphique qui vous convient, vous pourrez modifier le fichier de configuration de Lilo et spécifier le numéro de ce mode dans la ligne « vga=... ». De cette manière, votre système redémarrera automatiquement dans ce mode graphique.

X-D-2 - Configuration du serveur X

Les manipulations précédentes n'ont pas grand intérêt si vous ne désirez travailler qu'avec la console. En effet, l'affichage en mode graphique est beaucoup plus lent que l'affichage en mode texte, et l'affichage du pingouin Tux au démarrage ne vous apportera pas grand-chose. C'est pour cela que l'étape suivante est normalement de configurer le serveur X de X.org pour le pilote frame buffer, afin d'utiliser l'environnement graphique XWindow et son système de fenêtrage.

La configuration du serveur X est élémentaire. Il faut avant tout s'assurer que l'on dispose bien du pilote permettant au serveur X d'utiliser l'interface /dev/fb0. Ce pilote se nomme fbdev, et utilise un autre module spécifique au système d'exploitation nommé fbdevhw. Il faut ensuite modifier ou créer le fichier xorg.conf pour utiliser ce pilote. Les seules sections à modifier pour utiliser le pilote frame buffer sont la section « Device » et la section « Screen ».

La section « Device » est réduite à sa plus simple expression, puisque tous les paramètres sont fixés par le mode VESA choisi au démarrage d'une part, et parce que le serveur X ne saurait pas les exploiter d'autre part. Il suffit donc d'indiquer que le pilote à utiliser est le pilote fbdev, et de donner l'adresse de la carte vidéo sur le bus à l'aide du mot-clef « BusID » :

```
Section "Device"
    Identifier "Carte 1"
    Driver "fbdev"
    BusID "PCI:1:5:0"
EndSection
```

Vous pourrez déterminer l'adresse de votre carte graphique à l'aide de la commande lspci, ou en demandant au serveur X de scanner les bus PCI en lui passant l'option -scanpci en paramètre :

```
X.org -scanpci
```

La section « Screen » est elle aussi très simplifiée, puisque le seul mode graphique utilisable est le mode choisi au démarrage de la machine. La liste des modes utilisables peut donc être franchement omise, ou se réduire à la valeur spéciale « default » :

```
Section "Screen"
    Device "Carte 1"
    Monitor "Moniteur 1"
    DefaultDepth 16
    SubSection "Display"
        Depth 16
        Modes "default"
    EndSubSection
EndSection
```

Notez qu'il est impératif que la profondeur de couleur de la sous-section « Display » soit la même que celle du mode VESA indiqué au démarrage. Prenez garde à ne pas utiliser une profondeur de couleur trop élevée, car cela

dégraderait encore un peu plus les performances. Par ailleurs, comme aucun mode n'est spécifié dans la section « Screen », les lignes de mode des sections « Monitor » sont à présent facultatives. Ces sections peuvent donc être simplifiées également.

Une fois ces modifications réalisées, vous devrez pouvoir démarrer XWindow simplement avec la commande `startx`. Vous disposerez alors de toutes les fonctionnalités de XWindow, avec des performances quelque peu inférieures à celles que vous auriez avec un serveur X adapté à votre carte graphique. Il est conseillé de suivre l'actualité de X.org afin de savoir si un tel serveur est en cours de développement et, si oui, d'en récupérer une version finale dès que possible.

X-E - Configuration des terminaux X

Nous avons vu dans le chapitre de configuration du système de base que la connexion des utilisateurs se faisait par l'intermédiaire d'un terminal. La plupart des terminaux sont en mode texte, mais il est également possible d'utiliser des terminaux graphiques sous XWindow. Les terminaux de ce type sont logiquement appelés « terminaux X ».

Si le niveau d'exécution par défaut de votre système est le niveau associé au démarrage sous XWindow (ce niveau est en général 3, 4 ou 5 selon les distributions), XWindow est lancé dès le démarrage de la machine et la demande de connexion se fait via un écran graphique. Dans ce cas, vous vous connecterez directement par l'intermédiaire de ce terminal X. Le principe de fonctionnement des terminaux X n'est pas exactement le même que celui que nous avons vu pour les terminaux virtuels (voir la [Section 6.8](#)). En effet, les terminaux X ne sont pas gérés par les processus `getty` et `login` classiques, mais par un programme spécifique nommé « `xdm` » (abréviation de l'anglais « X Display Manager »).

X-E-1 - Principe de fonctionnement de xdm

Contrairement à ce qu'il fait avec les processus `getty`, `init` ne lance qu'un script d'initialisation de XWindow lorsqu'il passe dans le niveau d'exécution correspondant. Ce script a pour tâche essentiellement le démarrage du démon `xdm`. C'est ce démon qui est en charge de gérer les connexions sur tous les serveurs X qu'il peut trouver, en proposant la fenêtre de login et en permettant ainsi à l'utilisateur de s'identifier et s'authentifier.

En général, `xdm` lance lui-même le serveur X de la machine locale, et surveille son exécution. Lorsque l'utilisateur se déconnecte, le serveur X se réinitialise, et `xdm` affiche une nouvelle fenêtre de connexion. Cependant, `xdm` est également capable d'utiliser des serveurs X distants (qui sont donc déjà lancés), afin de permettre la connexion d'utilisateurs sur des terminaux X déportés. Enfin, il est possible de le configurer pour qu'il se signale sur un réseau, afin que les serveurs X fonctionnant sur ce réseau puissent lui demander une connexion. Les serveurs X capables de se connecter à `xdm` de cette manière doivent comprendre le protocole XDMCP (abréviation de l'anglais « XDM Control Protocol »). Comme on le voit, la souplesse de ce mécanisme est tout simplement exceptionnelle.

Note : Pratiquement, le processus `xdm` se duplique pour chaque terminal X dont il gère la connexion. Ainsi, chaque processus `xdm` propose la connexion et surveille la déconnexion de l'utilisateur sur un terminal X donné. Tous les processus `xdm` sont lancés par le processus `xdm` initial qui a été lancé par les scripts de changement de niveau d'exécution. Ce mécanisme est donc complètement transparent pour l'utilisateur.

La plupart des environnements graphiques modernes (comme KDE et GNOME) fournissent leur propre gestionnaire de display. Ces gestionnaires sont toutefois compatibles avec `xdm`, et leur configuration se fait de la même manière.

X-E-2 - Configuration de xdm

Le comportement de `xdm` est complètement défini dans ses fichiers de configuration, qui sont en général placés dans le répertoire `/etc/X11/xdm/`. Ces fichiers de configuration décrivent chacun un des aspects du comportement de `xdm`. Le fichier de configuration principal est le fichier `xdm-config`, qui contient les références sur les autres fichiers de configuration. Notez que si vous utilisez l'environnement de bureau KDE et son gestionnaire de connexion `kdm`, les fichiers de configuration à utiliser sont ceux de `kdm`. Bien qu'ils soient strictement compatibles avec ceux de `xdm`,

ces fichiers se situent dans le répertoire de configuration de kdm. Ce répertoire est généralement le sous-répertoire `share/config/kdm/` du répertoire d'installation de KDE.

X-E-2-a - Serveurs X locaux

La définition des serveurs X qui n'utilisent pas le protocole XDMCP et que xdm doit prendre en charge directement est réalisée dans le fichier de configuration référencé par la ligne « `DisplayManager.servers` » du fichier `xdm-config`. Par défaut, ce fichier est le fichier `/etc/X11/xdm/Xservers`. Il contient une ligne par serveur, chaque ligne ayant la syntaxe suivante :

```
display type [commande]
```

où `display` est le nom du display à utiliser, `type` est le type de serveur X (serveur local ou distant), et `commande` est la commande à exécuter pour lancer le serveur s'il est local. Le display indiqué sera celui qui sera utilisé pour définir la variable d'environnement `DISPLAY`, et doit donc utiliser la syntaxe normale des displays. Les deux types de serveurs disponibles sont « `local` », pour les serveurs de la machine locale que xdm doit lancer lui-même, et « `foreign` », pour les serveurs distants sur lesquels xdm doit afficher la fenêtre de demande de connexion (ils doivent donc être déjà lancés). La ligne de commande à utiliser pour le lancement du serveur ne doit être spécifiée que pour les serveurs de type « `local` ».

Les serveurs X peuvent prendre un certain nombre de paramètres en ligne de commande pour leur démarrage. Vous en trouverez la liste complète dans la page de manuel `Xserver` et dans la page de manuel `X.org`. Les options les plus intéressantes dans le cadre du lancement du serveur X par xdm sont celles permettant de définir le display que le serveur aura en charge et le terminal virtuel qu'il devra utiliser pour effectuer son affichage. Notez bien que le display doit être le même que celui donné à xdm. Ces deux options sont passées directement à la suite du nom de l'exécutable du serveur X à lancer :

```
/usr/bin/X display vtNN
```

où `vtNN` est le nom du terminal virtuel à utiliser (`vt01` pour `/dev/tty01`, `vt02` pour `/dev/tty02`, etc.).

Ainsi, si l'on veut faire en sorte que xdm lance automatiquement deux sessions X sur les terminaux virtuels 11 et 12, en leur affectant respectivement les displays `:0.0` et `:1.0`, on devra placer ces deux lignes dans son fichier de configuration `Xservers` :

```
:0 local /usr/bin/X :0 vt11
:1 local /usr/bin/X :1 vt12
```

Note : La spécification du terminal virtuel à utiliser est facultative. Si aucun terminal n'est indiqué, le serveur X utilisera le premier terminal qu'il trouvera. Cependant, il est plus sûr de toujours indiquer le terminal à utiliser, car le noyau ne crée les terminaux qu'à la demande, lorsqu'un processus désire y accéder. Or les serveurs X ne cherchent pas à en créer de nouveau. Si tous les terminaux virtuels existants sont déjà utilisés, le serveur X tentera donc de s'en approprier un, sans se préoccuper du programme qui l'utilise à ce moment. Le fait d'indiquer manuellement le terminal à utiliser vous évitera donc des problèmes assez curieux, tels que ceux qui peuvent survenir si les terminaux utilisés par les serveurs X le sont déjà par un `getty` par exemple. De plus, cela permet également d'éviter, si on lance plusieurs serveurs X sur la même machine, qu'ils n'accèdent ensemble au même terminal virtuel lors de leur initialisation.

Il est recommandé par ailleurs de basculer sur chaque terminal virtuel sur lequel un serveur X est lancé avant de se connecter afin de les forcer à créer leur terminal. En effet, si vous ne procédez pas ainsi, il est possible que vous ayez le temps de vous connecter dans une session X avant que les autres terminaux X n'aient fini de s'initialiser. Lorsque ceux-ci démarreront, ils changeront le terminal virtuel courant et l'affichage de votre bureau risque d'être

corrompu. Si cela se produit, ne vous inquiétez pas : votre session X n'a pas planté. Assurez-vous seulement que vous vous trouvez bien sur le bon terminal virtuel, et faites en sorte que l'écran soit rafraîchi (par exemple, changez de bureau virtuel ou déplacez une fenêtre sur l'écran).

Remarquez également que les fichiers de configuration fournis avec X.org ne prévoient pas l'utilisation de plus de deux sessions X par défaut. Si d'aventure vous désirez en utiliser plus, vous devrez compléter le fichier de configuration `xdm-config` avec les lignes suivantes :

```
DisplayManager._2.authorize: true
DisplayManager._3.authorize: true
etc.
```

Chaque ligne indique que le display correspondant (:2 pour « DisplayManager._2.authorize », etc.) doit utiliser les mécanismes de sécurité de XWindow. Nous verrons ces mécanismes dans la section suivante.

Méfiez-vous enfin de la consommation mémoire requise par chaque session X. Les serveurs X sont des programmes gourmands en mémoire, car ils doivent manipuler des images. Les performances du système risquent donc de se dégrader sensiblement si vous lancez plusieurs serveurs X. N'oubliez pas qu'après tout, XWindow est un système de fenêtrage et que la plupart des gestionnaires de fenêtres X donnent la possibilité d'utiliser des bureaux virtuels. En pratique, deux sessions X devraient suffire, afin de se connecter sous deux utilisateurs différents (un utilisateur normal et l'utilisateur root par exemple).

Enfin, notez que le changement d'un terminal X à un autre terminal virtuel peut provoquer quelques problèmes en ce qui concerne l'état du clavier. En effet, l'affichage des diodes (Verrou numérique et Majuscule en particulier) n'est pas rétabli correctement, même si le clavier continue à fonctionner correctement. Cela peut surprendre quelque peu la première fois que l'on rencontre ce problème. Il suffit d'appuyer deux fois de suite sur les touches `Verr Num` et `Caps Lock` (c'est-à-dire la touche de verrouillage des majuscules) afin de rétablir l'état correct de ces diodes. Remarquez également que les serveurs X mémorisent l'état des terminaux virtuels lors de leur démarrage, et neutralisent à chaque basculement vers un terminal en mode texte les éventuels changements de police de caractères.

X-E-2-b - Serveurs X utilisant XDMCP

Comme nous l'avons dit plus haut, `xdm` est également capable de se signaler sur un réseau à l'aide du protocole XDMCP, afin de permettre aux serveurs X distants de réaliser une connexion sur la machine où il est lancé. De cette manière, le lancement d'un serveur X sur une machine du réseau permettra d'obtenir la fenêtre de connexion à votre machine Linux.

`xdm` est capable de répondre aux demandes de connexion directe des serveurs X distants (remarquez que dans ce cas, `xdm` est serveur de connexions pour les serveurs X des postes clients. Faites bien attention à la terminologie client/serveur ici !). Cela suppose que chaque poste client connaisse l'adresse des machines qui utilisent `xdm`. Cette information peut faire partie de la configuration du poste client, mais elle peut également être déterminée dynamiquement. Pour cela, le serveur X du poste client émet une requête XDMCP en mode broadcast sur le réseau (c'est-à-dire à destination de tout le monde). Chaque machine sur laquelle `xdm` fonctionne répondra à ce client, et le serveur X proposera ainsi la liste des machines sur lesquelles une connexion est réalisable via `xdm`. Ainsi, l'utilisateur du poste client pourra choisir le serveur sur lequel il désire se connecter, et le processus `xdm` de ce serveur lui enverra la fenêtre de login.

En réalité, tous les serveurs X ne sont pas capables de gérer les réponses reçues de plusieurs processus `xdm` provenant de plusieurs machines à la suite de l'envoi d'une requête en mode broadcast. Par ailleurs, il n'est pas toujours possible d'utiliser le mode broadcast, car les paquets de ce type peuvent très bien ne pas être routés lors de la traversée d'une passerelle. Par conséquent, `xdm` offre également la possibilité d'interroger les machines du réseau local en réponse à une requête de connexion indirecte. Le serveur X reçoit en réponse le résultat de la requête effectuée par broadcast, et l'utilisateur peut choisir la machine à laquelle il désire accéder.

La configuration de xdm pour le protocole XDMCP se fait dans le fichier identifié par la ligne « `DisplayManager.accessFile` » du fichier `xdm-config`. Par défaut, le fichier ainsi référencé est le fichier `Xaccess` du répertoire `/etc/X11/xdm/`. La syntaxe de ce fichier est très simple. Il contient des règles indiquant le comportement de xdm lorsqu'il reçoit une requête de connexion de la part d'un serveur X. Ces règles sont définies pour des groupes de machines. Les machines sont identifiées par leur nom ou leur adresse IP, plusieurs machines pouvant être spécifiées par une même règle grâce aux caractères génériques classiques '*' et '?'. Il est également possible d'exclure une machine ou un groupe de machines en préfixant le nom du caractère de négation '!'.

Il existe quatre types de lignes dans le fichier `Xaccess`. Le premier type permet simplement d'indiquer que les connexions directes ou en mode broadcast provenant d'une machine sont toutes acceptées. Ce sont les lignes les plus simples, puisqu'il suffit d'indiquer la machine ou le groupe de machines. Par exemple, la ligne suivante :

```
*.localnet.org
```

indique que les demandes de connexion provenant de toutes les machines du domaine « `localnet.org` » sont acceptées. Il est possible de faire en sorte que xdm ne réponde qu'aux demandes de connexion directes simplement en ajoutant le mot clé « `NOBROADCAST` » à la suite de la ligne. Ainsi, si la machine « `termX3.localnet.org` » doit pouvoir se connecter directement, mais ne doit pas recevoir de réponse de notre machine, on ajoutera la ligne suivante dans le fichier `Xaccess` :

```
termX3.localnet.org NOBROADCAST
```

Le deuxième type de ligne permet de spécifier le comportement de xdm pour les demandes de connexion indirectes. Ces lignes contiennent toujours le nom de la machine ou du groupe de machines, mais ce nom est suivi de la liste des machines auxquelles xdm doit faire suivre la demande de connexion du serveur X. Par exemple, supposons que la machine « `termX2.localnet.org` » ne soit pas capable d'effectuer des requêtes en mode broadcast, et que l'on veuille qu'elle puisse se connecter sur les serveurs « `s5.localnet.org` », « `s11.localnet.org` » et « `s13.localnet.org` ». On devra alors ajouter la ligne suivante dans le fichier `Xaccess` :

```
termX2.localnet.org s5.localnet.org s11.localnet.org \  
s13.localnet.org
```

Le troisième type de ligne permet de lancer un programme nommé « `chooser` » et qui va proposer la liste des serveurs à l'utilisateur du serveur X qui demande à se connecter. Ce programme est très utile pour les serveurs X qui ne sont pas capables de proposer ce choix à l'utilisateur. La syntaxe de ces lignes est très simple, puisqu'il suffit de faire précéder les noms des serveurs par le mot clé `CHOOSER`. Si l'on reprend l'exemple précédent, la ligne de configuration pour le terminal X « `termX2` » deviendrait :

```
termX2.localnet.org CHOOSER s5.localnet.org s11.localnet.org s13.localnet.org
```

La liste des serveurs qui suit peut être assez longue, et il peut être utile de faire en sorte que le programme `chooser` effectue une requête en mode broadcast pour le compte du serveur X qui n'en est pas capable. Il suffit pour cela de remplacer la liste des serveurs par le mot clé « `BROADCAST` ». Ainsi, la ligne suivante :

```
termX4.localnet.org CHOOSER BROADCAST
```

signifie simplement que lorsque le serveur X de la machine « `termX4.localnet.org` » se connecte au processus xdm local, celui-ci lance le programme `chooser`. Ce dernier lance une requête en broadcast sur le réseau afin de déterminer la liste des machines exécutant le processus xdm. Une fois qu'il a obtenu cette liste, il l'affiche sur le display géré par le serveur X qui s'est connecté. De cette manière, l'utilisateur peut choisir la machine sur laquelle il va se connecter. Le serveur X est mis en relation avec le processus xdm de cette machine, et la fenêtre de login est enfin proposée à l'utilisateur.

Le quatrième type de ligne permet de définir des « macros » de remplacement pour gérer les listes de noms de machines plus simplement. La définition d'une macro se fait avec la syntaxe suivante :

```
%nom liste
```

où nom est le nom de la macro à définir, et liste est la liste des machines qu'elle représente. Il est possible d'utiliser des macros dans les définitions de macros.

Note : Le protocole XDMCP utilise le port 117 du protocole UDP. Vous pourrez donc ajouter la ligne suivante dans votre fichier `/etc/services`, si elle ne s'y trouve pas déjà :

```
xdmcp 117/udp
```

Notez également que vous ne pourrez pas tester votre configuration en utilisant l'interface réseau loopback. En effet, xdm vérifie la validité des adresses sources des paquets qu'il reçoit, et il refuse par défaut tout paquet provenant de l'adresse 127.0.0.1. Si vous désirez malgré tout utiliser XDMCP sans réseau réel, vous pourrez utiliser l'interface réseau dummy. Cette interface peut être créée en activant l'option « Dummy net driver support » du menu « Network device support ».

X-E-2-c - Paramétrage du serveur X pour utiliser le protocole XDMCP

L'écriture du fichier `Xaccess` ne représente que la partie serveur du travail de configuration du protocole XDMCP. En effet, il faut encore indiquer aux serveurs X distants à quel processus xdm il doivent tenter de se connecter lors de leur démarrage. Bien entendu, cela dépend fortement du serveur X utilisé, qui peut être quelconque. Par exemple, le serveur X peut très bien fonctionner sur une machine Windows. Par conséquent, seule la syntaxe des serveurs de X.org sera décrite ici.

Les serveurs X de X.org utilisent des options en ligne de commande pour déterminer comment ils doivent se comporter à leur démarrage. Lorsqu'ils sont lancés directement par xdm (c'est-à-dire lorsqu'ils sont lancés en local, avec les options indiquées dans le fichier `Xservers`), ils n'ont pas besoin d'options spécifiques, car ils n'utilisent pas XDMCP dans ce cas. En revanche, si l'on désire les lancer manuellement (ou dans un script d'initialisation), on devra utiliser l'une des options suivantes :

- `-query` permet de demander une connexion directe sur une machine donnée. Le nom de la machine doit être spécifié à la suite de l'option ;
- `-indirect` permet de demander une connexion indirecte sur une machine donnée. Le nom de cette machine doit être spécifié à la suite de l'option. La liste des machines qui a été indiquée dans le fichier `Xaccess` sera retournée, ou bien le programme `chooser` sera lancé ;
- `-broadcast` permet de demander au serveur X d'effectuer une requête de connexion en broadcast sur le réseau.

Les serveurs de X.org ne sont pas capables de proposer une liste de machines à l'utilisateur. Ils prennent systématiquement la première machine qu'ils trouvent. Vous ne contrôlerez donc pas la machine sur laquelle la connexion sera faite si vous utilisez l'option `-broadcast`. C'est pour cela qu'il est recommandé d'utiliser le programme `chooser` dans le fichier de configuration `Xaccess` et de configurer les serveurs pour faire des demandes de connexion indirectes.

X-E-2-d - Fichiers d'initialisation de sessions

xdm fournit la possibilité d'effectuer des actions lorsque divers événements concernant la session XWindow en cours ont lieu. Par exemple, il est possible de réaliser un traitement particulier pour l'initialisation de l'environnement d'un serveur X, pour fixer l'environnement de l'utilisateur qui se connecte, et pour effectuer le ménage lorsqu'il ferme sa session. Toutes ces actions sont effectuées dans des scripts qui sont référencés par des lignes du fichier `xdm-config`.

La ligne « `DisplayManager.DISPLAY.setup` », où `DISPLAY` est le nom du display (`_0` pour `:0`, `_1` pour `:1`, etc.), référence un script qui est exécuté au nom de l'utilisateur root avant que la fenêtre de login ne soit présentée. Le

fichier par défaut utilisé par X.org est le fichier `/etc/X11/xdm/Xsetup`. C'est donc dans ce script que vous pourrez mettre des commandes d'initialisation spécifiques pour préparer le login.

De la même manière, la ligne « `DisplayManager.DISPLAY.startup` » référence le script `Xstartup` par défaut. Ce fichier est exécuté sous le compte `root` juste après l'authentification de l'utilisateur. C'est ici que l'on pourra par exemple enregistrer la connexion de l'utilisateur dans le système.

La ligne « `DisplayManager.DISPLAY.session` » référence quant à elle le script `Xsession`, qui est exécuté après `Xstartup`, mais au nom de l'utilisateur cette fois. Ce script prend généralement en charge le lancement des programmes nécessaires à la gestion de la session de l'utilisateur. Par exemple, il est possible d'y placer les commandes de lancement du gestionnaire de fenêtres ou du gestionnaire de bureau.

Enfin, la ligne « `DisplayManager.DISPLAY.reset` » référence le script `Xreset` du répertoire `/etc/X11/xdm/`, qui est exécuté au nom de l'utilisateur `root` lorsque la session X se termine. Vous pouvez donc exécuter les tâches nécessaires pour faire le ménage à cet endroit, par exemple le désenregistrement de la connexion de l'utilisateur dans le système.

Tous ces scripts sont lancés sans paramètres, sauf le script `Xsession`. Ce script peut en effet recevoir des paramètres fournis par le gestionnaire de connexions (`xdm`, `kdm`, `gdm` ou autre), qui permettent de représenter les choix que l'utilisateur a faits pour sa connexion dans la fenêtre de login. Le premier paramètre de ce script est normalement le nom du gestionnaire de fenêtres à utiliser. Par exemple, pour l'environnement de bureau KDE, le programme qui doit être exécuté est le script `startkde`, alors que pour l'environnement de bureau Gnome, il s'agit du programme `gnome-session`. Notez que le nom « `failsafe` » référence un mode dégradé, et n'est normalement utilisé que quand les autres gestionnaires de fenêtres disponibles ne se lancent pas correctement.

Les scripts lancés au nom de l'utilisateur `root` (`Xsetup` et `Xreset`) sont exécutés dans un environnement minimal. La valeur de la variable d'environnement `PATH` pour ces scripts est déterminée par la valeur de la ligne « `DisplayManager.DISPLAY.systemPath` » du fichier de configuration `xdm-config`. De même, le script `Xsession` est exécuté avec la variable d'environnement `PATH` fixée à la valeur spécifiée par la ligne « `DisplayManager.DISPLAY.userPath` » du fichier `xdm-config`. Si vos scripts de gestion des connexions ne fonctionnent pas correctement, c'est sans doute un problème lié à l'environnement.

En pratique, ces fichiers de scripts de gestion des connexions sont fournis par votre distribution et vous ne devriez donc pas à avoir à y toucher. Vous pouvez toutefois y faire quelques modifications si vous estimez que les actions effectuées ne vous conviennent pas. En particulier, certaines distributions n'enregistrent pas l'utilisateur qui commence une nouvelle session. Cette opération est nécessaire, parce que `xdm` n'est pas un processus de login classique et ne le fait pas automatiquement (il n'utilise pas de terminal pour faire la connexion). Vous aurez pour cela à utiliser l'utilitaire `sessreg` fourni avec `XWindow`. Cet utilitaire permet d'enregistrer un utilisateur avec l'option `-a` et de le supprimer avec l'option `-d`. Vous devrez donc typiquement placer une ligne telle que celle-ci :

```
sessreg -a -l $DISPLAY -x /etc/X11/xdm/Xservers $LOGNAME
```

dans le fichier `Xstartup`, afin d'enregistrer les utilisateurs qui commencent une session `XWindow`, et une autre ligne telle que celle-ci :

```
sessreg -d -l $DISPLAY -x /etc/X11/xdm/Xservers $LOGNAME
```

dans le fichier `Xreset`, afin de les supprimer lorsqu'ils terminent cette session.

X-E-3 - Paramétrage des terminaux X

La configuration des terminaux X comprend un certain nombre de points qui vont de l'ajustement du nombre de couleurs et de la résolution à la disposition du clavier, en passant par les paramètres de la souris et l'économiseur d'écran. Heureusement, nous avons déjà vu comment nous pouvions contrôler un grand nombre de ces paramètres. Par exemple, la résolution graphique, ainsi que le nombre de couleurs et la disposition du clavier, sont définis dans

le fichier de configuration général de X.org `xorg.conf`. Nous verrons donc ici comment modifier des paramètres qui tiennent plus de l'ergonomie que de la configuration de base.

X-E-3-a - La commande `xset`

Ces paramètres peuvent souvent être fixés lors du démarrage du serveur X, ou dynamiquement avec la commande `xset`. Les paramètres que l'on peut fixer avec `xset` sont très nombreux, et seuls les plus utiles seront décrits ici. Veuillez consulter la page de manuel `xset` pour plus de détails.

L'option `dpms` permet de fixer les temps d'attente avant la mise en veille du moniteur. Sachez que la mise en veille constitue de loin le meilleur économiseur d'écran, et que les économiseurs logiciels du type ballet de lignes sont très jolis, mais ne servent à rien. Pire, ils peuvent ralentir la machine, ce qui est inadmissible si vous l'utilisez en tant que serveur (heureusement, les économiseurs d'écran logiciels se lancent généralement avec des priorités très faibles, afin de ne pas perturber les autres processus).

Cette option prend de un à trois paramètres, qui représentent respectivement le temps avant le passage en mode économie d'énergie du moniteur, le temps avant le passage en mode veille, et le temps avant l'extinction complète du moniteur. Ces temps sont exprimés en secondes, la valeur nulle permettant de désactiver cette fonctionnalité. Ainsi, la commande suivante :

```
xset dpms 0 0 600
```

permet d'éteindre l'écran au bout de dix minutes d'inactivité.

Ces temps peuvent également être fixés dans le fichier de configuration `xorg.conf`, où ils sont exprimés en minutes, à l'aide des mots clés « `StandbyTime` », « `SuspendTime` » et « `OffTime` ». Ces mots clés doivent être placés dans la section « `ServerFlags` » du fichier de configuration. Cependant, les serveurs X n'activent ces fonctionnalités que pour les moniteurs capables de gérer le standard DPMS. Ces moniteurs doivent donc être signalés à l'aide de la ligne suivante :

```
Option "DPMS" "On"
```

dans la section « `Monitor` » qui les définit.

Note : Les valeurs des temps d'attente que vous pouvez fixer dans le fichier de configuration `xorg.conf` peuvent être écrasées par les valeurs définies dans la configuration de votre gestionnaire de bureau. Vous devrez donc vérifier ces paramètres si les modifications que vous faites dans le fichier `xorg.conf` ne sont pas prises en compte ou si l'économie d'énergie reste désactivée malgré la présence de l'option DPMS dans la section « `Monitor` » de votre moniteur.

Une autre option importante de la commande `xset` est l'option `fp`, qui permet d'ajouter et de supprimer des chemins de répertoires de polices de caractères. Pour ajouter un répertoire, il suffit d'utiliser l'option `+fp` et de faire suivre le chemin de ce répertoire. La suppression d'un répertoire se fait de la même manière, avec l'option `-fp`. Par exemple, pour ajouter le répertoire de polices `/usr/share/fonts/100dpi/` à la liste des répertoires de polices du serveur, il suffit de taper la commande suivante :

```
xset +fp /usr/share/fonts/100dpi
```

De plus, comme nous l'avons déjà vu plus haut, les chemins des répertoires de polices de caractères peuvent être fixés statiquement à l'aide du mot clé « `FontPath` » de la section « `Files` » du fichier de configuration `xorg.conf`.

Enfin, l'option `q` de `xset` vous permettra de visualiser l'ensemble des paramètres en cours d'utilisation.

X-E-3-b - Configuration de la disposition du clavier

Nous avons vu dans le chapitre de configuration du système de base le fonctionnement du clavier sous Linux. Les applications qui lisent les données provenant du pilote clavier de la console peuvent travailler au niveau ASCII, au niveau keycode ou au niveau scancode. Les serveurs X font partie des logiciels qui préfèrent la troisième solution, ce qui signifie qu'ils interprètent eux-mêmes le flux de scancodes provenant du clavier.

La raison de ce choix est que X.org est une implémentation portable du système XWindow pour PC. Cela signifie qu'il est prévu pour fonctionner sur plusieurs systèmes d'exploitation de type Unix et fonctionnant sur l'architecture compatible PC. Par conséquent, il ne pouvait pas se baser sur des keycodes, qui sont évidemment dépendants du système utilisé.

L'inconvénient en revanche est que cela impose que les serveurs X définissent un mécanisme parallèle de gestion des touches du clavier, qui permette de modifier la disposition des touches et de définir le type de clavier utilisé. Plusieurs mécanismes sont utilisables, cependant, il semble que ce qui est le plus utilisé actuellement est le mécanisme « Xkb » (abréviation de l'anglais « X KeyBoard »). Ce mécanisme permet de définir un grand nombre de données concernant le clavier, puisqu'outre l'emplacement des touches, il fournit la géométrie du clavier. Ainsi, toute application le désirant peut déterminer la forme du clavier et en dessiner une représentation fidèle.

Malheureusement, le protocole Xkb ne prévoit pas la possibilité de définir de nouvelles touches avec leurs scancodes. La gestion des codes reçus du clavier n'est en effet pas paramétrable. Par conséquent, il est impossible d'utiliser des claviers exotiques si ceux-ci ne sont pas reconnus par X.org.

Le protocole Xkb utilise les fichiers de configuration stockés dans le répertoire `/etc/X11/xkb/`. Ce répertoire contient un certain nombre de sous-répertoires, dont chacun est relatif à un des aspects de la configuration du clavier. Le rôle des principaux sous-répertoires est détaillé dans le tableau suivant :

Nom du répertoire	Fonction
<code>geometry/</code>	Contient les fichiers de description de la géométrie des claviers. Ces fichiers peuvent être utilisés par les applications pour obtenir une représentation graphique du clavier.
<code>keycodes/</code>	Contient les fichiers de définitions des « keycodes » X du clavier. Les keycodes X ont la même fonction que les keycodes de Linux : donner un nom unique à chaque touche. Cependant, ils diffèrent des keycodes de Linux en ce sens qu'ils ne sont pas définis à partir des scancodes du clavier, mais à partir de codes numériques générés par le serveur X. Ce mécanisme suppose que le serveur X connaisse tous les scancodes envoyés par les claviers. Comme il est impossible de déclarer de nouvelles séquences de scancodes au niveau du serveur X, on ne peut pas compléter les définitions de clavier existantes pour prendre en charge d'éventuelles nouvelles touches.
<code>symbols/</code>	Contient les définitions des plans de clavier ou, autrement dit, les associations

	entre les keycodes X et les symboles obtenus lors de l'appui sur les touches.
rules/	Contient les définitions des modèles de clavier de X.org. Ces modèles de clavier sont utilisés pour simplifier la définition des claviers dans le fichier de configuration xorg.conf.
keymaps/	Contient les définitions des principaux claviers connus. Une définition de clavier comprend entre autres la définition des keycodes, des touches modificatrices et des symboles affectés aux keycodes.

Il est probable que vous n'ayez pas à modifier un seul des fichiers de ces répertoires, car ils sont correctement définis par défaut. Le seul fichier qui peut être intéressant est le fichier `fr` du sous-répertoire `symbols/`. En effet, c'est dans ce fichier que vous trouverez la définition des symboles affectés à chaque touche pour un clavier français, en fonction des modificateurs (`ALT`, `CTRL`, etc.) actifs lors de l'appui sur la touche. Le principe de fonctionnement de ce fichier est semblable aux plans de claviers de Linux : pour chaque keycode, un ensemble de symboles peuvent être définis. Ces symboles sont indiqués entre accolades (caractères '[' et ']'), en deux jeux de symboles entourés de crochets (caractères '[' et ']'). Le premier jeu de symboles indique les symboles accessibles directement et en utilisant la touche majuscule, et le deuxième jeu définit les symboles accessibles avec la combinaison de la touche `AltGr`.

Par exemple, la touche du chiffre '3' au-dessus des lettres 'Z' et 'E' est définie comme suit dans le fichier de définition du clavier français (fichier `fr`) :

```
key <AE03> { [ quotedbl, 3 ],
              [ numbersign, sterling ] };
```

Cela signifie que la touche dont le keycode est « `<AE03>` » (c'est-à-dire celle la quatrième de la deuxième rangée de touches du clavier) génère les symboles « " » et « 3 » selon que la touche majuscule est utilisée ou non, et les symboles « # » et « £ » respectivement en minuscule ou en majuscule et avec la touche `AltGr` enfoncée.

Si vous regardez le fichier du clavier français, vous constaterez que toutes les touches ne sont pas définies. La raison de cela est que ces fichiers utilisent par défaut les affectations de touches du clavier américain standard. Par conséquent, seules les différences ont été redéfinies dans ce fichier.

Vous pourrez éventuellement modifier certaines affectations de touches si vous le désirez. Ce peut être nécessaire si vous désirez homogénéiser les combinaisons de touches entre la console Linux et XWindow. Par exemple, il peut être utile de redéfinir le comportement de la touche 'E' pour qu'elle renvoie le symbole Euro (caractère 'â, ¤') en combinaison avec la touche `AltGr`. Pour cela, vous pourrez utiliser la configuration suivante dans le fichier de clavier utilisé :

```
key <AD03> { [ e, E ],
              [ currency, E ] };
```

Note : Notez que la touche Euro est déjà accessible par défaut sur les claviers français avec la touche des monnaies (Livre Sterling, Dollar).

Une autre touche qui peut être redéfinie pour un clavier français est la touche 'O', afin de lui faire générer les symboles e dans l'o français ('œ' et 'œlig;') au lieu des symboles o barrés ('ø' et 'Ø') norvégiens, qui ne sont pas très utiles en français. Notez que les noms de symboles utilisés dans les tables d'encodage correspondent aux symboles de l'encodage ISO 8859-1, et que par conséquent, vous devrez utiliser les noms de symboles « `onehalf` » et « `onequarter` » pour représenter les caractères 'œ' et 'Ä'. En effet, ces symboles sont les symboles de l'encodage ISO 8859-1 qui ont le même code que les symboles 'œ' et 'Ä' dans l'encodage ISO 8859-15.

Sachez enfin que l'obtention de ces symboles est bien entendu assujettie à l'utilisation d'une police de caractères utilisant l'encodage ISO 8859-15.

Comme vous avez dû le constater, un certain nombre de plans de clavier sont définis en standard dans X.org. De même, plusieurs géométries et définitions de keycodes sont fournies dans les sous-répertoires `geometry/` et `keycodes/` du répertoire `/etc/X11/xkb/`. Le serveur X utilise les fichiers qui sont référencés dans la section « Keyboard » du fichier de configuration `xorg.conf`. Comme on l'a déjà vu plus haut dans ce chapitre, le mot clé « `XkbLayout` » permet de définir le plan de clavier à utiliser. Ainsi, si vous spécifiez le plan de clavier « `fr` » à la suite de ce mot clé, le fichier de définition des associations keycodes-symboles utilisé sera le fichier `fr` du répertoire `/etc/X11/xkb/symbols/`. Les autres options sont fixées par les mots clés « `XkbRules` » et « `XkbModel` ». Le premier mot clé indique quel est le fichier à utiliser pour définir les modèles de claviers de X.org. Ce fichier est localisé dans le sous-répertoire `rules/`, et fournit les valeurs par défaut à utiliser pour chaque modèle de clavier. Le deuxième mot clé indique le modèle de clavier à utiliser. Ainsi, il n'est nécessaire de spécifier que trois paramètres pour définir le clavier sous X.org : le fichier contenant la définition des modèles de clavier, le modèle lui-même et le fichier définissant la disposition des touches.

Note : Vous pouvez également définir tous les paramètres du clavier dans le fichier `xorg.conf`. Cette technique est plus compliquée et n'est pas nécessaire pour définir un clavier français, elle ne sera donc pas décrite ici. Vous trouverez plus de renseignements à ce sujet dans la page de manuel `xorg.conf`.

X-F - Paramétrage des applications et ressources X

Du fait que toutes les applications X font appel au serveur X pour effectuer leur affichage, et au gestionnaire de fenêtres pour définir l'apparence et la disposition de leurs fenêtres, elles sont soumises aux options générales de ces deux programmes. Elles peuvent cependant être paramétrées elles aussi, et leur comportement peut être personnalisé à souhait par chaque utilisateur.

Pour cela, XWindow fournit un mécanisme standard pour toutes les applications, afin qu'elles puissent gérer leurs options de configuration de manière uniforme. Ce mécanisme se base sur des paramètres gérés par le serveur X, que l'on appelle des ressources X. La plupart des applications utilisent ces ressources pour stocker leurs paramètres de configuration, et le fait que ce soit le serveur X qui centralise leur gestion assure une certaine cohérence entre les applications. En général, les ressources X décrivent essentiellement les aspects visuels des applications, par exemple les polices de caractères utilisées, les couleurs et l'épaisseur des traits. Cependant, les applications peuvent parfaitement utiliser les ressources pour enregistrer des paramètres qui leur sont propres. Le serveur X n'interprète en aucune manière les ressources utilisées par les applications, il ne fait que les mettre à disposition des applications.

Le serveur X gère les ressources dans une base de données qui est initialisée lors de son démarrage : la « X Ressource DataBase » (« `xrdb` » en abrégé). Cette base de données est initialisée lors du démarrage du serveur à partir de fichiers de configuration au format texte, que l'on peut donc éditer et modifier aisément. Le serveur X utilise deux jeux de fichiers lors de son initialisation : les fichiers de configuration par défaut des applications d'une part, et le fichier de préférences personnelles de chaque utilisateur d'autre part. Les fichiers de configuration des applications sont normalement placés dans le répertoire `/etc/X11/app-defaults/`. Les applications y copient leurs propres fichiers lors de leur installation. Le nom de chaque fichier correspond au nom de l'application à laquelle il appartient, ce qui permet de retrouver aisément le fichier de configuration d'une application donnée. Le fichier de préférences personnelles de chaque utilisateur, quant à lui, se place dans son répertoire racine et porte le nom `.Xresources`.

Les fichiers de ressources des applications sont lus en premier, en général dans les scripts d'initialisation de XWindow. En revanche, le fichier de préférences personnelles d'un utilisateur n'est lu que lors de l'ouverture d'une session X par cet utilisateur. Lorsqu'une ressource présente dans le fichier de configuration par défaut d'une application est redéfinie dans le fichier de préférences personnelles d'un utilisateur, la valeur utilisée est bien entendu celle de l'utilisateur. Ainsi, chacun peut redéfinir les valeurs par défaut des ressources de toutes les applications qu'il utilise. La plupart des applications peuvent également prendre des options en paramètres de ligne de commande, qui permettent de fixer les valeurs de certaines ressources. Ces options prévalent sur les valeurs définies dans les fichiers de configuration des applications et de préférences personnelles des utilisateurs.

Vous pourrez vous inspirer du contenu des fichiers de configuration des applications pour paramétrer vos applications. Pour cela, il vous suffira simplement de recopier les définitions des ressources qui vous intéressent dans votre fichier

.Xresources, et de tester le résultat avec différentes valeurs. L'affectation d'une valeur à une ressource se fait avec la syntaxe suivante :

```
ressource : valeur
```

où `ressource` est le nom de la ressource, et `valeur` est la valeur à lui affecter.

Les noms de ressources sont structurés de manière hiérarchique. Ils sont en effet constitués d'un certain nombre de composantes séparées par des points (caractère '.'). La première composante qualifie l'application elle-même, et porte son nom. Il est d'usage de mettre la première lettre de ce nom en majuscule, et si cette lettre est un 'X', de mettre également la deuxième lettre du nom en majuscule (beaucoup d'applications X ont en effet un nom commençant par un 'X'). Les composantes suivantes définissent un sous-ensemble de paramètres apparentés dans l'ensemble des paramètres déterminé par les composantes précédentes. Enfin, la dernière composante du nom de ressource constitue le nom du paramètre lui-même.

Par exemple, le nom de ressource suivant :

```
XApplication.mainWindow.background
```

qualifie la couleur d'arrière-plan (propriété « `background` ») de la fenêtre principale (propriété « `mainWindow` ») de l'application « `XApplication` ». Notez que les deux premières lettres du nom de la ressource sont en majuscules ici, car la première lettre du nom de l'application est elle-même un 'X' majuscule.

Note : Les fichiers de ressources utilisent des noms de couleurs prédéfinies pour définir les couleurs des différentes parties des applications. Vous pourrez trouver la liste de ces noms de couleurs ainsi que leurs définitions dans le fichier `/usr/share/X11/rgb.txt`. Vous pouvez également définir vos propres couleurs avec la syntaxe « `rgb:R/V/B` », où « `R` » représente la portion de rouge de la couleur, « `V` » représente la portion de vert, et « `B` » la portion de bleu.

Ainsi, l'ensemble des paramètres des applications X est organisé un peu comme sont organisés les fichiers dans une arborescence de répertoires. L'analogie ne s'arrête pas là : il est possible de caractériser un ensemble de ressources grâce à des caractères génériques. Par exemple, en utilisant une étoile (caractère '*') comme séparateur à la place du point dans un nom de ressource, toutes les ressources dont le nom comprend les deux composantes seront sélectionnées, que ces deux composantes soient adjacentes ou séparées d'autres composantes intermédiaires. Par exemple, le nom de ressource suivant :

```
XApplication*background
```

qualifie la couleur d'arrière-plan de toutes les fenêtres de l'application « `Xapplication` ».

De même, le caractère générique point d'interrogation (caractère '?') permet de remplacer une composante par un nom quelconque. Le nom de ressource suivant :

```
XApplication?.background
```

représente donc toutes les ressources comportant les composantes « `XApplication` » et « `background` », séparées par une composante de nom quelconque.

La structure des noms de ressources d'une application n'est pas due au hasard. Les paramètres sont regroupés soit par fonctionnalité, soit par thème, soit par appartenance à la même partie de l'application. En général, les noms de ressources utilisés par les applications sont simples à comprendre. Vous trouverez des exemples de noms de ressources dans les fichiers de ressources des applications.

Vous pouvez obtenir la liste exhaustive des ressources d'une application avec l'utilitaire `xrresconf`. En fait, cet utilitaire permet d'obtenir la liste des noms des ressources appartenant à une branche de l'arborescence des ressources. Cet utilitaire s'utilise selon la syntaxe suivante :

```
appres branche
```

où `branche` est le nom de la branche que l'on veut explorer.

Comme la base de données des ressources est initialisée au démarrage du serveur X et à l'ouverture de la session X, les modifications que vous pourrez apporter à votre fichier de ressources personnelles ne seront pas prises en compte immédiatement. Pour cela, vous devez demander la relecture de la base de données explicitement et relancer les applications concernées. Cette opération peut être effectuée à l'aide de l'outil `xrdb` (abréviation de l'anglais « X ressource DataBase »). En fait, cet outil permet d'effectuer diverses opérations sur la base de données des ressources.

L'option `-merge` est certainement celle que vous utiliserez le plus. Elle permet de mettre à jour les valeurs des ressources avec celles décrites dans un fichier. La syntaxe utilisée est la suivante :

```
xrdb -merge fichier
```

où `fichier` est le nom du fichier contenant la définition des ressources (généralement, il s'agit de votre fichier `.Xresources`).

L'option `-load` permet d'effectuer le même travail, mais la base de données est vidée au préalable. Cette option permet donc de réinitialiser complètement le contenu de la base de données avec le contenu du fichier passé en paramètre. Sa syntaxe est la même que celle de l'option `-merge`.

L'option `-remove` permet de vider la base de données et de supprimer toutes les ressources que vous auriez pu définir au préalable. Elle s'utilise selon la syntaxe suivante :

```
xrdb -remove
```

Enfin, l'option `-query` permet de lister l'ensemble des ressources existantes.

X-G - Gestion de la sécurité sous XWindow

XWindow n'effectue pas de contrôle d'accès sur les opérations demandées par un processus connecté à un serveur X. Cela signifie que tous les processus qui ont accès à un display y ont un accès total. Par exemple, un processus peut parfaitement faire une prise d'écran et récupérer tous les événements provenant du clavier. Il va de soi que ce genre de processus pose un problème de sécurité majeur, aussi XWindow considère-t-il que seuls les processus lancés sur la machine locale au nom de l'utilisateur courant ont le droit de se connecter au serveur X. Cela protège donc le display de l'utilisateur contre les malversations de personnes mal intentionnées.

Cependant, cette restriction est assez forte, parce qu'elle va vous empêcher de lancer un programme graphique sous un autre nom d'utilisateur. Par exemple, vous ne pourrez pas lancer un programme nécessitant les droits administrateur et dont le bit `setuid` n'est pas positionné. En effet, une fois que vous vous serez identifié en tant que `root` avec la commande `su`, les programmes que vous lancerez le seront au nom de l'utilisateur `root`, et le serveur X refusera la connexion. Un autre cas d'école où cette restriction peut être gênante est le lancement d'un programme graphique sur une machine distante. La situation est encore plus grave, car cette fois la demande de connexion au serveur X ne provient pas de la machine où il s'exécute, et n'a aucune chance d'aboutir.

XWindow fournit donc les outils nécessaires pour contrôler le niveau de sécurité en cours. Classiquement, le serveur X utilise deux techniques pour contrôler la validité des demandes de connexion des clients. Le premier mécanisme de contrôle d'accès est relativement grossier, puisqu'il se base simplement sur les adresses des machines. Le deuxième mécanisme a été introduit ultérieurement afin de permettre des contrôles plus fins. Les contrôles se font alors à l'aide d'un mécanisme de clés privées, que seuls les clients autorisés connaissent.

X-G-1 - La commande xhost

La commande `xhost` permet de fixer le niveau des contrôles d'accès basés sur les adresses de machines. Comme on l'a vu, le serveur X n'accepte les connexions que de la machine locale. Vous pouvez cependant lui indiquer d'accepter toutes les connexions provenant d'une autre machine avec une commande telle que celle-ci :

```
xhost +machine
```

où `machine` est le nom de la machine en qui l'on a confiance. Si aucun nom de machine n'est spécifié, tous les mécanismes de contrôle d'accès sont désactivés.

Inversement, la suppression d'une machine de la liste des machines autorisées est réalisée par la commande suivante :

```
xhost -machine
```

La commande `xhost -` (sans nom de machine) permet de réactiver les mécanismes de contrôle d'accès s'ils ont été désactivés par un `xhost +`.

Notez bien que donner le droit de connexion à une machine suppose que l'on fasse confiance à tous les utilisateurs de cette machine, car alors ils pourront tous se connecter sur votre serveur X local. De plus, rien ne vous garantit que la machine à qui vous donnez ces droits n'a pas été usurpée par celle d'un pirate (technique dite de l'« IP spoofing »). Ce n'est donc évidemment pas la solution recommandée, surtout si vous êtes connecté à Internet.

X-G-2 - La commande xauth

Le deuxième mécanisme de sécurité utilise une clé privée. Tous les clients qui désirent se connecter au serveur local doivent connaître cette clé, faute de quoi leur requête sera refusée. Ainsi, vous pouvez très simplement permettre l'utilisation de votre display à une personne donnée sur une machine donnée, simplement en lui communiquant la clé utilisée par le serveur X gérant votre display. Bien entendu, cette communication doit se faire de manière sûre.

Par défaut, les clés privées utilisées par les clients sont enregistrées dans le fichier `.Xauthority` de votre répertoire personnel. Ce fichier contient une clé privée pour chaque display auquel vous avez le droit de vous connecter. Les clients que vous lancez consultent donc ce fichier afin de déterminer la clé à utiliser, en fonction du display auquel ils désirent accéder. Une fois cette clé connue, ils peuvent s'authentifier auprès du serveur X gérant ce display, et ainsi obtenir une connexion. Notez bien que le fichier `.Xauthority` ne doit être accessible que par vous.

Si vous utilisez `xdm`, une nouvelle clé est automatiquement générée à chaque fois que vous vous connectez à un terminal X. `xdm` enregistre cette clé dans un fichier temporaire du répertoire référencé par le lien symbolique `/etc/X11/xdm/authdir` (qui référence généralement le répertoire `/var/lib/xdm/authdir/`), que seul l'utilisateur `root` peut accéder. Il communique ensuite ce fichier au serveur X local à l'aide de l'option `-auth` du serveur X pour que celui-ci puisse lire la clé à utiliser. Enfin, `xdm` enregistre cette clé dans votre fichier `.Xauthority`, pour que vous puissiez lancer des clients dans cette session X.

Comme vous pouvez le constater, ce mécanisme de sécurité est totalement transparent pour l'utilisateur. Cependant, il peut être nécessaire de manipuler le fichier `.Xauthority` pour lire les clés privées et les communiquer aux personnes de confiance qui doivent avoir accès à votre display. Ces manipulations peuvent être effectuées à l'aide de la commande `xauth`.

La clé associée à un display peut être obtenue avec la commande suivante :

```
xauth list display
```

où `display` est le nom du display auquel la clé donne accès. `xauth` affiche alors le display, le type d'authentification utilisé (MIT-MAGIC-COOKIE-1) et la clé privée.

Vous pouvez communiquer cette clé par les moyens que vous voulez à la personne devant accéder à votre display. Celle-ci pourra alors utiliser la commande suivante pour ajouter la clé à son fichier `.Xauthority` :

```
xauth add display . clé
```

où `display` est le display utilisant la clé `clé`. Le caractère `'` est une abréviation pour le type d'authentification « MIT-MAGIC-COOKIE-1 » (type d'authentification par défaut). Dès que la clé aura été ajoutée dans son fichier `.Xauthority`, cette personne aura accès à votre display.

Enfin, la commande suivante :

```
xauth remove display
```

permet de supprimer la clé utilisée pour le display `display`.

Note : Il est nécessaire d'utiliser une technique de communication sûre pour transmettre la clé, faute de quoi le mécanisme de sécurité de XWindow ne servirait à rien. L'utilisation d'OpenSSH est vivement recommandée.

X-H - Gestion des polices de caractères

La gestion des polices de caractères est relativement compliquée sous XWindow. En effet, elle est gérée par un protocole complexe, qui permet de décrire avec précision les diverses polices de caractères, quels que soient leur type et leur aspect. De plus, la gestion des polices nécessite de traiter correctement les symboles spécifiques à chaque pays, ce qui complique encore un peu plus les choses. Enfin, pour couronner le tout, XWindow n'utilise pas la notion de graphisme indépendant du périphérique, comme la GDI de Windows. Cela implique, hélas, qu'il ne se charge que de l'affichage et pas de l'impression. Chaque programme doit donc prendre en compte lui-même les problèmes d'impression, ce qui implique, en général, que les programmes soient capables d'imprimer sur des imprimantes PostScript. Par conséquent, la configuration des polices de caractères doit non seulement se faire pour XWindow, mais également pour chacun des programmes (ou, au moins, pour l'interpréteur GhostScript, utilisé pour l'impression avec des imprimantes non PostScript).

X-H-1 - Gestion des polices de caractères sous XWindow

Originellement, XWindow ne pouvait afficher que des polices de type bitmap, c'est-à-dire des polices de caractères définies par un dessin pour chaque caractère et pour un certain nombre de résolutions. Cette technique avait l'avantage d'être rapide à l'affichage, mais de relative mauvaise qualité lorsque les tailles demandées n'étaient pas exactement celles pour lesquelles la police avait été dessinée. En effet, la police devait alors être redimensionnée à partir de la taille la plus proche disponible. Ultérieurement, les polices Postscript sont apparues. Ces polices sont définies vectoriellement, c'est-à-dire par des formules mathématiques. La forme des caractères est ainsi calculée pour chaque dimension, ce qui permet d'avoir une qualité irréprochable. L'avantage des polices Postscript est qu'elles sont gérées de manière native par les imprimantes PostScript, et assurent une qualité d'impression optimale. En revanche, leur affichage sur les écrans n'est pas toujours correct, car les formules utilisées ont été conçues pour les périphériques disposant de grandes résolutions et ne donnent pas forcément un résultat esthétique pour les résolutions d'écrans. Enfin, les polices TrueType ont été inventées par Apple. Ces polices sont également vectorielles, mais disposent en plus de petites astuces permettant d'améliorer leur lisibilité sur les périphériques à faible résolution tels que les écrans. Microsoft a licencié la technologie TrueType et l'a intégrée à Windows par la suite.

Afin de décrire les polices le plus précisément possible, XWindow leur donne des noms relativement complexes. Ces noms constituent ce que l'on appelle la description logique des polices (« XLFD », qui est l'abréviation de l'anglais « X Logical Font Description »). Cette convention de dénomination spécifie que la description des polices doit être constituée de différents champs, séparés par des tirets ('-'). De plus, la description doit elle-même être précédée d'un tiret. Les différents champs utilisés dans la description logique des polices sont les suivants :

- le nom de l'éditeur de la police, ou le nom du type de la police ;
- le nom de la police (par exemple, « arial ») ;
- la graisse de la police (par exemple, « bold » pour gras, « medium » pour normal) ;
- l'inclinaison de la police (par exemple, 'r' pour roman, 'i' pour italique) ;
- la largeur de la police ;
- des options de styles avancées ;
- la taille de la police ;
- la taille des points ;
- la résolution horizontale de la police ;
- la résolution verticale ;
- le type d'espacement de la police (par exemple, 'c' pour constant, 'p' pour proportionnel) ;
- la largeur moyenne de la police ;
- le jeu de caractères de la police ;
- la page de codes de la police.

Vous pouvez consulter la documentation de XWindow pour une description plus détaillée de ces champs. Ces informations ne sont pas toutes supportées par les polices de caractères. Inversement, certaines polices peuvent correspondre à plusieurs descriptions (ne serait-ce que parce qu'elles disposent de plusieurs tailles).

Parmi les informations décrivant les polices se trouvent le jeu de caractères de la police et sa page de codes (rappelons que ces informations constituent ce que l'on appelle l'encodage de la police). Il peut y avoir plusieurs pages de codes pour un jeu de caractères, chacune représentant une manière de numéroté les différents caractères du jeu. Une police peut disposer de plusieurs jeux de caractères, mais en pratique ce n'est que rarement le cas. En revanche, la manière de numéroté les caractères (c'est-à-dire la page de codes) peut avoir une influence certaine.

Comme on le verra plus tard, le jeu de caractères le plus pratique pour les pays d'Europe de l'Ouest est le jeu ISO 8859 (jeu de caractères dit « latin »). Ce jeu de caractères dispose de la plupart des caractères utilisés en Europe. Pour les alphabets occidentaux, la page de codes la plus utilisée est la page de codes 1, ce qui fait que l'encodage des polices occidentales est ISO 8859-1. Cependant, quelques caractères ont été oubliés dans cette page de codes (notamment le o e dans l'o français ('œ')). Ces caractères sont pourtant disponibles dans certaines polices (en particulier, les polices Truetype provenant de Windows), mais ne sont malheureusement pas disponibles avec l'encodage ISO 8859-1. Pour y accéder, on est obligé d'utiliser un autre encodage, par exemple l'encodage ISO 8859-15. Cet encodage est quasiment identique à l'encodage ISO 8859-1, aux caractères additionnels près, qui ont été ajoutés pour les pays européens. Il est également possible d'utiliser la page de codes 1252 des polices de caractères de Windows. Cette page de codes correspond à l'encodage « windows-1252 », parfois également nommé « microsoft-ansi » ou encore « microsoft-cp1252 ». En résumé, le jeu de caractères et la page de codes permettent d'indiquer pour quel pays (ou quel alphabet) une police est destinée.

Note : Le problème des encodages est que seul l'encodage ISO 8859-1 est vraiment utilisé par la majorité des gens. Cela implique que les autres encodages risquent de ne pas être reconnus par tous les programmes. En particulier, il est assez difficile d'imprimer des textes encodés avec des encodages non standards.

La description logique des polices de caractères est très précise, puisqu'elle permet de spécifier l'origine de la police, son nom, les informations la concernant, les caractères qu'elle comprend et comment ils sont numérotés. Lorsqu'on choisit une police de caractères, on peut parfaitement ne préciser que certains critères (par exemple, l'encodage de la police). Dans ce cas, les champs constituant le nom de la police non spécifiés pourront être remplacés par des caractères génériques. Le caractère '*' permet ainsi de spécifier n'importe quelle valeur pour ce champ, et le caractère '?' permet de spécifier n'importe quel caractère dans un champ.

Le programme xfontsel permet de sélectionner les polices de caractères installées sur un système. Il peut être utile pour comprendre la signification des différents champs de la description logique des polices. Des exemples de descriptions logiques de polices sont donnés ci-dessous :

```
-adobe-helvetica-medium-r-*-*12-*-*-*-*iso8859-1
-*courier-bold-*normal-*10-*-*-*-*iso8859-15
-winfonds-arial-*i-normal-*16-*-*-*-*microsoft-cp1252
```


Les polices de caractères sont souvent regroupées dans des répertoires. Il peut exister plusieurs répertoires de polices sur un système, si bien qu'il est nécessaire d'indiquer à X.org dans quels répertoires il doit rechercher les polices de caractères utilisables. Les répertoires des polices utilisées par le serveur X sont indiqués dans le fichier `xorg.conf`, dans la section « Files ». Comme on le verra plus tard, cette section peut également contenir des références sur des serveurs de polices de caractères. Les serveurs de polices de caractères peuvent rechercher les polices dans des répertoires indiqués de différentes manières selon le serveur. Pour certains serveurs, les répertoires de polices sont indiqués dans un fichier de configuration, pour d'autres, ils sont indiqués en ligne de commande.

Dans le protocole X, les répertoires de polices doivent contenir un fichier `fonts.dir` donnant la liste des polices de ce répertoire. Ce fichier indique, sur sa première ligne, le nombre de polices installées dans ce répertoire, et, sur les lignes suivantes, l'association entre les fichiers de polices et leur description logique. Ce fichier est créé normalement par le programme `mkfontdir`. Ce programme génère le fichier `fonts.dir` à partir des informations contenues dans les fichiers des polices ou à partir du nom des fichiers des polices eux-mêmes. En général, les polices à taille variable ne contiennent pas ces informations dans le format standard des polices X11 classiques, et `mkfontdir` ne peut donc pas créer le fichier `fonts.dir` automatiquement. Pour ces polices, il faut créer un fichier `fonts.scale` contenant les mêmes informations que le fichier `fonts.dir`, et que `mkfontdir` utilisera pour créer ce dernier. La méthode pour créer le fichier `fonts.scale` dépend du type de police utilisée. Celle utilisée pour les polices TrueType sera décrite dans la [Section 10.8.2](#).

En général, les encodages les plus standards sont gérés directement par le serveur X ou par le serveur de polices. En particulier, l'encodage ISO 8859-1 est géré nativement. Il est toutefois possible de définir de nouveaux encodages dans des fichiers d'encodages. Pour que le serveur X ou le serveur de polices puisse utiliser les polices définies avec ces encodages, il faut que le répertoire d'installation des polices contienne un fichier `encodings.dir`. Ce fichier a la même structure que le fichier `fonts.dir`, c'est-à-dire qu'il contient le nombre des encodages sur la première ligne et, sur chaque ligne suivante, le nom de l'encodage et le nom d'un fichier contenant la définition d'un encodage, séparés par un espace. De cette manière, le serveur X et le serveur de polices sont capables de réaliser l'association entre le nom de l'encodage utilisé dans la description logique de polices et le fichier de définition de cet encodage. La méthode permettant de créer le fichier `encodings.dir` sera décrite plus loin dans la section traitant de l'[installation des polices TrueType](#).

L'écriture des fichiers de définition d'encodages de polices est une tâche assez compliquée, qui nécessite de bien connaître la numérotation des caractères de chaque type de fichier de police. De manière très simplifiée, on peut dire que les fichiers de définition d'encodage font l'association entre le numéro de chaque caractère dans la police et une numérotation standardisée des caractères (XWindow utilise l'encodage Unicode) de cette police. Cela permet de manipuler les textes avec la numérotation standard, et d'utiliser des polices qui ne définissent pas tous les caractères de cette numérotation ou qui ne les numérotent pas de la même manière. Heureusement, les encodages les plus courants ont déjà été écrits et il est fort peu probable que vous ayez à vous intéresser à ce problème. La structure des fichiers d'encodages ne sera donc pas décrite plus en détail dans ce document.

X-H-2 - Installation des polices TrueType

L'installation des polices TrueType sous XWindow ne pose désormais plus de problèmes, puisqu'il suffit de définir le fichier `fonts.dir` dans le répertoire d'installation de ces polices. Cependant, un certain nombre d'opérations supplémentaires devront être réalisées pour permettre l'impression des documents utilisant les polices TrueType. Ce paragraphe détaille la manière de déclarer les polices TrueType au niveau du serveur X, et présente les opérations nécessaires à leur impression pour quelques logiciels courants.

Généralement, les fichiers de polices sont placés dans des sous-répertoires du répertoire `/usr/share/fonts/`. Ces sous-répertoires permettent de classer les fichiers de polices par type et par taille. Dans la suite de ce document, il sera supposé que les polices TrueType sont toutes situées dans le sous-répertoire `TTF/`.

X-H-2-a - Configuration du serveur X

Pour permettre au serveur X d'utiliser les polices TrueType, il faut simplement écrire le fichier `fonts.dir` de chaque répertoire de polices. Rappelons que ce fichier contient la liste des polices du répertoire dans lequel il se trouve, et

est utilisé à la fois par le serveur X et par les serveurs de polices de caractères. La notion de serveur de polices sera vue en détail dans la [Section 10.8.3](#).

Normalement, le fichier `fonts.dir` est généré par le programme `mkfontdir`. Ce programme utilise les informations contenues dans les fichiers de polices ou dans le nom de ces fichiers pour le générer. Malheureusement, les polices TrueType, comme la plupart des polices à taille variable, ne contiennent pas ces informations dans un format compréhensible par `mkfontdir`, ni dans leurs fichiers, ni dans leurs noms. Celui-ci ne peut donc pas créer le fichier `fonts.dir` directement. Dans ce cas, `mkfontdir` utilise le fichier de définition des polices à taille variable `fonts.scale`. Ce fichier doit être créé soit manuellement, soit à l'aide de l'utilitaire `mkfontscale`. Celui-ci doit être appelé dans le répertoire contenant les polices de caractères, et génère un fichier `fonts.scale` directement utilisable par `mkfontdir`.

Une fois le fichier `fonts.scale` créé, il ne reste plus qu'à appeler `mkfontdir` avec la ligne de commande suivante :

```
mkfontdir -e /usr/share/fonts/encodings -e /usr/share/fonts/encodings/large
```

Cette commande aura pour effet de créer le fichier `fonts.dir` à partir du fichier `fonts.scale`, et de créer les fichiers d'encodage `encodings.dir` associés (en fait, il s'agit d'une simple recopie). `mkfontdir` utilise les fichiers de définition d'encodages standards placés dans les répertoires indiqués avec l'option `-e`. Dans la commande précédente, les encodages standards supportés par XWindow du répertoire `/usr/share/fonts/encodings/` sont utilisés.

Note : L'encodage Microsoft cp1252 correspond à l'encodage « ANSI » utilisé par Microsoft Windows. C'est un encodage non standard, qui ressemble fortement à l'encodage ISO 8859-1 spécifié par l'ISO. Ce dernier souffre malheureusement de l'absence de certains caractères français (comme le o e dans l'o). Si vous voulez rester compatible avec les normes ISO et accéder malgré tout à ces caractères, vous pouvez utiliser l'encodage ISO 8859-15. Vous accéderez ainsi à tous les caractères utilisés en Europe, mais vous ne pourrez plus utiliser les caractères additionnels définis par Microsoft.

Il faut noter que la plupart des polices sont défectueuses et indiquent une page de codes erronée dans leur fichier. Cela explique pourquoi quelques descriptions logiques de polices dans le fichier `fonts.scale` généré automatiquement par `mkfontscale` peuvent être fausses. Pour ces polices, il faudra corriger l'encodage manuellement en modifiant le fichier `fonts.scale` avant d'exécuter la commande `mkfontdir`.

X-H-2-b - Configuration des polices TrueType pour l'impression

La plupart des programmes modernes, comme la suite bureautique OpenOffice et les programmes du gestionnaire de bureau KDE par exemple, prennent en charge complètement les polices de caractères TrueType tant pour l'affichage que pour l'impression. En général, ils impriment dans des fichiers au format PostScript, et embarquent la définition des polices de caractères TrueType dans ces fichiers, qui sont utilisables de manière indépendante. Vous n'aurez donc certainement rien à faire pour pouvoir utiliser les polices TrueType à l'impression .

Cependant, les plus anciens programmes se contentent de référencer les polices utilisées dans les fichiers d'impression, et il faut configurer le sous-système d'impression pour qu'il puisse reconnaître ces polices et les imprimer correctement. D'une manière générale, deux cas peuvent se présenter :

- soit on dispose d'une imprimante PostScript, auquel cas il faut convertir les polices TrueType en polices Adobe Type 42, qui constituent une encapsulation des polices TrueType en PostScript. Cette conversion a l'avantage de ne pas provoquer de perte de qualité, puisque la police TrueType est utilisée telle quelle ;
- soit on ne dispose pas d'imprimante PostScript, auquel cas on utilise nécessairement un interpréteur PostScript. Cet interpréteur est souvent GhostScript, car il s'agit encore une fois d'un logiciel libre. Quel que soit l'interpréteur PostScript utilisé, il faut soit le configurer pour qu'il puisse utiliser les polices TrueType, soit que les applications fournissent la définition des polices TrueType qu'elles utilisent dans les fichiers PostScript qu'elles génèrent lors de l'impression. Cette dernière solution est celle utilisée par les logiciels récents (comme les applications de KDE ou OpenOffice), ce qui fait qu'en général, les polices TrueType sont prises en charge sans configuration additionnelle. Nous verrons plus loin comment configurer les versions de GhostScript ultérieures à la 7.00 pour qu'elles puissent prendre en compte les polices TrueType pour les autres applications.

Dans les deux cas, il se peut qu'il faille configurer le logiciel utilisé pour qu'il puisse utiliser les polices TrueType de concert avec le sous-système d'impression.

Les paragraphes suivants décrivent les procédures à suivre pour imprimer les polices TrueType en général. La configuration des logiciels ne sera pas abordée, consultez leur aide ou recherchez des renseignements sur Internet quant à la procédure à suivre.

X-H-2-b-i - Conversion des polices TrueType en polices Adobe de Type 42

La conversion des polices TrueType en polices Adobe de Type 42 est une opération nécessaire si l'on utilise directement une imprimante PostScript ou si l'on utilise un interpréteur PostScript incapable de gérer directement les polices TrueType. De manière générale, il est recommandé d'utiliser GhostScript même si l'on possède une imprimante PostScript, car la configuration de l'impression se fait de la même manière que les autres utilisateurs de Linux, et il est donc plus facile de trouver de l'aide sur Internet.

La conversion en polices Adobe de Type 42 peut être réalisée avec le programme `ttfps`, disponible sur Internet sous le nom `ttfps.tar.gz`. Cet outil est tout à fait correct, mais il souffre d'un grave défaut : il ne permet pas de choisir l'encodage de la police PostScript qu'il génère. Plus grave encore, il utilise systématiquement l'encodage standard Adobe, qui ne dispose pas de la plupart des lettres accentuées françaises (c'est aussi la raison pour laquelle il est recommandé d'utiliser GhostScript). Vous pouvez bien entendu modifier le code source si vous vous en sentez le courage.

Vous devrez compiler `ttfps` pour pouvoir l'utiliser. Lorsque vous aurez extrait les fichiers sources de l'archive, vous aurez à éditer le fichier `Makefile` pour :

- définir la variable d'environnement `CC` avec pour valeur le nom du compilateur que vous utilisez (en l'occurrence, `gcc`) :

```
CC=gcc
```

- choisir entre les deux jeux d'options de compilation `CFLAGS`. Vous ne devez en choisir qu'un seul, il faut obligatoirement commenter l'autre avec un caractère dièse (`#`). Le choix dépend de l'architecture de votre machine. Si vous utilisez un PC, vous devrez choisir l'option contenant l'option `-DSMALLENDIAN`.

Une fois ces modifications faites, vous pourrez compiler `ttfps` avec la simple commande suivante :

```
make
```

L'utilisation de `ttfps` est très simple. Pour convertir une police TrueType en police Adobe de Type 42, il suffit d'utiliser la ligne de commande suivante :

```
ttfps [-a fichier.afm] fichier.ttf fichier.pfa
```

où `fichier.afm` est le nom du fichier de définition des dimensions de la police PostScript, `fichier.ttf` est le nom du fichier de police TrueType à encapsuler, et `fichier.pfa` est le nom de la police PostScript résultante. La génération du fichier `.afm` est facultative.

X-H-2-b-ii - Installation des polices TrueType pour GhostScript

GhostScript est un interpréteur PostScript capable d'imprimer les polices TrueType. Sa configuration est très simple, puisqu'il suffit de lui donner le nom de police que les fichiers PostScript utilisent pour la référencer et le nom du fichier de police correspondant. Cette association est définie dans le fichier `Fontmap` de GhostScript, que vous pourrez trouver dans son répertoire d'installation (normalement, GhostScript se trouve dans le répertoire `/usr/share/ghostscript/version/`, où `version` est le numéro de la version installée).

Les informations que vous devez ajouter dans ce fichier sont similaires à celles stockées dans le fichier `fonts.dir` du répertoire de polices. Pour chaque police, il faut rajouter une ligne de la forme suivante :

```
(nom) (fichier) ;
```

où `nom` est le nom de la police d'imprimante que les programmes utilisent dans les fichiers PostScript qu'ils génèrent, et `fichier` est le chemin absolu du fichier de la police TrueType.

En pratique, le nom de la police d'imprimante peut être choisi librement. On veillera toutefois à ne pas utiliser des noms de polices contenant des espaces, car certains programmes peuvent avoir du mal à manipuler de tels noms. Il faudra bien utiliser ce nom lors de la déclaration des polices d'imprimante dans les logiciels capables d'imprimer en PostScript. Si l'on ne peut pas définir ce nom au niveau des logiciels, il faudra ajouter un alias dans le fichier `Fontmap` de GhostScript pour chaque nom de police utilisé par les logiciels. Il n'existe pas de règle générale permettant de déterminer les noms de polices utilisés par les programmes. Le plus simple est peut-être dans ce cas de déterminer les noms utilisés en imprimant un document dans un fichier PostScript et en regardant dans le fichier les instructions de changement de police de caractères.

X-H-3 - Configuration d'un serveur de polices

XWindow étant un système de fenêtrage fonctionnant en réseau, il propose des services additionnels sur le réseau en plus de l'affichage. Parmi ces services, on notera la possibilité de mettre en place des serveurs de polices de caractères. Ces serveurs permettent à des clients situés sur d'autres machines d'accéder à la définition des polices de caractères de la machine locale.

Le serveur de police fourni par XWindow se nomme « `xfs` » (abréviation de l'anglais « X Font Server »). Ce serveur utilise un fichier de configuration qui permet de lui spécifier les options avec lesquelles il doit démarrer. Dans la suite de ce document, le nom supposé de ce fichier est `/usr/lib/X11/fs/config`. Un fichier de configuration typique est donné ci-dessous :

```
# Exemple de fichier de configuration du serveur de police :

# Limite à 10 le nombre de clients connectés à ce serveur :
client-limit = 10

# Demande le démarrage d'un autre serveur quand la limite
# précédente est atteinte :
clone-self = on

# Répertoire des polices de caractères :
catalogue = /usr/share/fonts/TTF

# Fixe la taille par défaut (en dixièmes de points) :
default-point-size = 120

# Résolution par défaut (100 x 100 et 75 x 75) :
default-resolutions = 100,100,75,75

# N'utilise pas le mécanisme des traces du système :
use-syslog = off

# Utilise le fichier d'erreurs suivant à la place :
error-file = /root/xfs.errors
```

Lorsque vous aurez créé votre fichier de configuration, il ne vous restera plus qu'à tester si tout fonctionne bien. Il vous faut pour cela lancer le serveur de polices avec la ligne de commande suivante :

```
xfs -port 7100 -config /etc/xfs.conf &
```

et essayer de lui demander la liste des polices qu'il peut fournir :

```
fsfonts -server localhost:7100
```

Si cette dernière commande échoue, il se peut que le chemin indiqué pour les répertoires de polices dans le fichier `/usr/lib/X11/fs/config` ne soit pas correct, ou que le fichier `fonts.dir` n'existe pas ou ne soit pas correct dans un des répertoires de polices.

Si vous le désirez, vous pouvez faire en sorte que le serveur de polices soit démarré automatiquement au lancement de X. Vous pourrez pour cela créer un script de lancement du serveur de polices, que vous placerez dans le répertoire `/etc/rc.d/` (ou le répertoire `/sbin/init.d/`, selon votre distribution) pour lancer et arrêter le serveur de polices en suivant le mécanisme des niveaux d'exécution. Vous trouverez ci-dessous un exemple de script de lancement du serveur de polices permettant d'utiliser le fichier de configuration précédent :

```
#!/bin/bash
#
# /etc/rc.d/xfs
#
# Fichier de lancement du serveur de polices.
#

PORT=7100
FILE=/usr/lib/X11/fs/config
PRGM=/usr/bin/xfs

[ -f $PRGM ] || exit 0

# Analyse les paramètres :
case "$1" in
    start)
        # Vérifie si xfs est déjà lancé :
        if [ -f /var/lock/subsys/xfs ]; then
            echo -n "Redémarrage du serveur de polices"
            killall -e -TERM $PRGM
            rm -f /var/lock/subsys/xfs
        fi
        $PRGM -port $PORT -config $FILE &
        touch /var/lock/subsys/xfs
        ;;
    stop)
        if [ ! -f /var/lock/subsys/xfs ]; then
            echo -n "xfs n'est pas lancé"
        else
            echo -n "Arrêt du serveur de polices"
            killall -e -TERM $PRGM
            rm -f /var/lock/subsys/xfs
        fi
        ;;
    restart)
        echo -n "Relecture du fichier de configuration"
        if [ -f /var/lock/subsys/xfs ]; then
            killall -e -USR1 $PRGM
        fi
        ;;
    *)
        echo "Usage: $0 {start|stop|restart}"
        exit 1
    esac
exit 0
```

Comme X11 est démarré classiquement dans le niveau d'exécution 3 ou 4, vous devrez créer les liens symboliques vers ce fichier pour le lancement et l'arrêt du serveur de polices dans le répertoire `rc3.d/` ou `rc4.d/`. Attention, rappelez-vous que les noms de ces liens indiquent le moment où le script est exécuté, aussi bien pour l'entrée que pour la sortie du niveau d'exécution. Vous devrez impérativement faire en sorte que ce script soit appelé avant que XWindow ne démarre si vous désirez utiliser le serveur de polices dans les sessions X locales.

L'utilisation du serveur de polices est très simple. Il suffit d'y accéder comme à un répertoire de polices de caractères normal, en utilisant la syntaxe suivante :

```
tcp/machine:port
```

où `machine` est la machine sur laquelle le serveur de polices à accéder est lancé, et `port` est le port que ce serveur écoute. Par exemple, pour ajouter les polices d'un serveur de polices locales à la liste des polices du serveur X, il suffit de taper la commande suivante :

```
xset +fp tcp/127.0.0.1:7100
```

Vous pourrez alors vérifier que ces polices sont bien disponibles avec le programme `xfontsel`.

Si vous le désirez, vous pouvez ajouter automatiquement les polices d'un serveur de polices en ajoutant la ligne suivante dans la section « `Files` » dans le fichier de configuration `xorg.conf` du serveur :

```
FontPath "tcp/127.0.0.1:7100"
```

Cette ligne permet d'indiquer au serveur X qu'il trouvera des polices de caractères au niveau du serveur de polices de la machine locale. Bien entendu, le serveur de polices doit impérativement être lancé avant le serveur X si une telle ligne est placée dans le fichier de configuration `xorg.conf`.

X-I - Problèmes classiques rencontrés

Il est possible que vous rencontriez quelques problèmes au lancement de XWindow après avoir modifié sa configuration. Ces problèmes peuvent aller du blocage complet de la machine à l'impossibilité de lancer XWindow.

D'une manière générale, il est recommandé de ne pas modifier la configuration dans le niveau d'exécution de XWindow (3, 4 ou 5 selon les distributions). En effet, dans ces niveaux, `xdm` relance automatiquement le serveur X dès qu'il détecte sa terminaison. Si le serveur X ne peut pas se lancer, votre ordinateur se bloquera dans une boucle infinie, avec en prime deux changements de mode graphique à chaque itération (ce qui peut endommager sérieusement votre moniteur). Si cela vous arrivait, vous n'auriez plus qu'à tenter le redémarrage de la machine en basculant rapidement sur un terminal (avec `CTRL+ALT+F1` suivi de `CTRL+ALT+DEL`. Faites-le très rapidement, avant que `xdm` n'ait le temps de tenter un redémarrage du serveur X !), et à redémarrer en indiquant le niveau d'exécution 2 à l'amorçage du noyau. Par conséquent, essayez vos changements de configuration dans le niveau d'exécution 2 si possible, et faites vos tests avec la simple commande `startx`.

Des erreurs un peu plus techniques peuvent provenir de la configuration réseau elle-même (n'oubliez pas que XWindow est un système graphique basé réseau). Les erreurs les plus effroyables sont sans doute les erreurs du type « `_X11TransSocketINETConnect: Can't connect: errno = n` », où `n` est un code d'erreur numérique. Typiquement, cette erreur signale un problème dans la configuration réseau. Les deux erreurs les plus classiques sont l'erreur 101, qui signale que la machine indiquée dans le `display` n'a pas pu être contactée, et l'erreur 111, qui signale que cette machine est bien accessible par le réseau, mais que le serveur X ne s'y trouve pas.

En général, vous devrez vérifier toute votre configuration réseau si vous rencontrez des erreurs 101, et le problème n'est certainement pas spécifique à XWindow. Vous pouvez également vérifier la validité de votre `display`, car il se peut qu'il désigne une machine inaccessible sur votre réseau (ou que la résolution de nom ait échoué pour cette machine).

Pour ce qui est de l'erreur 111, c'est beaucoup plus simple. Dans la grande majorité des cas, le serveur X désigné n'existe tout simplement pas. Il faut donc s'assurer que le serveur X en charge de gérer le `display` indiqué dans la variable d'environnement `DISPLAY` est bien lancé sur la machine désignée.

Enfin, lorsqu'une application cliente refuse de démarrer en affichant un message d'erreur « `Can't open display: xxx` », c'est tout simplement que la variable d'environnement `DISPLAY` n'a pas été précisée, et que l'option en ligne de commande `-display` n'a pas été utilisée non plus. L'application ne sait donc tout simplement pas à quel `display` se connecter. Si le `display` a bien été défini, mais que le message « `Can't open display:xxx` » est complété du message « `Connection to "xxx" refused by server` » ou « `Client is not authorized to connect to Server` », c'est que le client a été

lancé dans un autre compte utilisateur que le compte que vous utilisez couramment, et que ce compte ne dispose pas des droits nécessaires pour accéder à ce display. Vous devez dans ce cas donner l'accès à votre display au compte utilisateur sous lequel vous lancez le client.

XI - Conclusion

Vous avez pu voir dans ce document les différentes procédures à mettre en œuvre pour installer Linux. Ces procédures peuvent paraître compliquées, et en fait elles le sont effectivement. Cependant, il faut faire un choix entre fonctionnalité et simplicité. Linux choisit la voie la plus difficile : celle sur laquelle il faut être performant et fournir le plus de possibilités. La complexité qui en découle se voit immédiatement lors de son installation, mais elle se comprend, car chaque étape permet de le rendre à chaque fois plus puissant, en lui ajoutant une fonctionnalité de plus. Finalement, ce qui vous a motivé pendant toutes ces étapes, c'est sans aucun doute le désir de bénéficier de la stabilité et de la puissance de Linux. Vous ne serez pas déçu, et vous rendrez vite compte que Linux dépasse de loin les systèmes soi-disant plus ergonomiques sur ces deux points, dans une telle proportion que vous finirez par ne plus pouvoir les utiliser. Et si vous y êtes contraint, vous ne louerez plus leur facilité d'emploi, mais vous pestifierez bel et bien contre leur comportement aléatoire, leurs incohérences ou leurs plantages à répétition. En réalité, comme vous allez bientôt le découvrir, Linux est simple à utiliser pour les travaux du quotidien. De plus, comme il l'a déjà été dit au début de ce document, ce qui paraît compliqué au premier abord est souvent tout simplement inhabituel. C'est à l'usage que vous vous familiariserez avec les concepts Unix, et plus vous les utiliserez, plus vous les apprécierez. Aussi ne puis-je que vous souhaiter bonne continuation !

XII - Annexe A. Options de configuration du noyau

Les questions posées par le programme de configuration du noyau 2.6.20 sont récapitulées ci-dessous. Les réponses recommandées ont été choisies pour correspondre à la plupart des cas courants. Il ne s'agit pas des options recommandées pour installer un serveur ou pour une machine contenant des périphériques exotiques (carte vidéo, port infrarouge, etc.). Avec ce jeu d'options, un PC standard est supposé démarrer sans poser de problèmes. Vous devrez cependant certainement les adapter selon vos besoins. La plupart de ces options sont décrites plus en détail dans le chapitre traitant de la [configuration du matériel](#).

Je tiens à préciser que certaines de ces options m'ont laissé dubitatif, étant dans l'incapacité absolue de les comprendre et de les tester. Ces options sont en général les options concernant des fonctionnalités avancées ou des périphériques rarement utilisés.

XII-A - Menu « General setup »

Ce menu comprend les options générales de configuration de Linux.

L'option « **Prompt for development and/or incomplete code/drivers** », permet d'activer les options de configuration des fonctionnalités en cours de développement dans le reste du programme de configuration du noyau. Vous devez répondre 'Y' si des pilotes particuliers sont nécessaires. Si l'on répond 'N', les questions de configuration concernant ces fonctionnalités ne seront pas posées. Il est recommandé de répondre 'Y' à cette question, car nombre de fonctionnalités récentes sont marquées comme étant « expérimentales », bien qu'étant tout à fait stable.

L'option « **Local version - append to kernel release** » permet d'ajouter une chaîne de caractères spécifique au nom du noyau, afin de lui donner un nom spécifique et de le distinguer des noyaux génériques pour une installation particulière. Il est recommandé de laisser cette chaîne vide.

L'option « **Automatically append version information to the version string** » permet d'ajouter automatiquement des informations de version pour les versions de développement du noyau dans le numéro de version. Cette fonctionnalité n'est utilisée que par les développeurs du noyau et n'est pas utile pour les utilisateurs normaux. Toutefois, on peut laisser la valeur par défaut 'Y' sans crainte.

La question « Support for paging of anonymous memory » permet d'activer le mécanisme de pagination de la mémoire virtuelle de Linux, couramment appelé le « swap ». Comme cette fonctionnalité permet d'exploiter plus efficacement la mémoire vive pour les machines disposant d'une mémoire de taille raisonnable, il est recommandé d'activer cette option et de répondre 'Y' à cette question.

L'option « System V IPC » permet d'activer la gestion des communications interprocessus compatibles System V. C'est quasiment un standard sur tous les systèmes Unix, il faut donc répondre par 'Y'.

L'option « POSIX Message Queues » permet d'activer l'implémentation d'un certain type de communications interprocessus appelé les files de messages selon le standard POSIX. C'est une fonctionnalité standard qu'il est recommandé d'activer sur tous les systèmes Unix, il faut donc répondre par 'Y'.

L'option « BSD Process Accounting » permet d'activer les statistiques des applications utilisé sur les systèmes BSD. Ce mécanisme de surveillance peut être utilisé par quelques applications, aussi est-il recommandé de répondre par 'Y'.

L'option « BSD Process Accounting version 3 file format » permet d'utiliser le format de fichier version 3 des fichiers de traces pour la surveillance des applications. La version recommandée est 'N'.

L'option « Export task/process statistics through netlink (EXPERIMENTAL) » permet d'activer les statistiques sur les applications via le mécanisme de communication générique entre le noyau et les processus utilisateurs NETLINK. Contrairement au mécanisme de surveillance BSD, celui-ci permet d'obtenir les statistiques des applications sont disponibles par un envoi commande. Les informations sur le processus sont également fournies lors de la fin du processus. La réponse recommandée est 'N'.

L'option « Enable per-task delay accounting (EXPERIMENTAL) » permet d'activer la comptabilisation du temps passé par les processus dans les opérations bloquantes. Ces informations peuvent être utilisées pour un meilleur ordonnancement des tâches dans le système. La réponse recommandée est 'N'.

L'option « Enable extended accounting over taskstats » permet d'activer la collecte de statistiques complémentaires pour les transférer via l'interface taskstats. Cette option vous donnera l'accès à l'option « Enable per-task storage I/O accounting », qui permet de prendre en charge les statistiques sur les entrées/sorties de chaque processus. La réponse recommandée est 'N' pour ces deux options.

L'option « User Namespaces » permet d'activer une fonction du noyau destinée à certains logiciels de virtualisation, et dont le but est de modifier les informations sur la nature du système renvoyée aux programmes exécutés dans des serveurs virtuels. La réponse recommandée est 'N'.

L'option « Auditing support » permet de prendre en charge les fonctionnalités d'audit pour les autres sous-systèmes de sécurité de Linux tels que SELinux. Cette fonctionnalité est encore peu utilisée et la réponse recommandée est 'N'.

L'option « Enable system-call auditing support » permet d'activer les fonctionnalités d'audit des appels systèmes conjointement avec d'autres sous-systèmes de sécurité de Linux tels que SELinux. Cette fonctionnalité est encore peu utilisée et la réponse recommandée est 'N'.

L'option « Kernel .config support » permet d'inclure les options de configuration ainsi que la description complète de l'environnement de compilation du noyau dans le noyau lui-même. Cela permet de retrouver la configuration utilisée pour compiler le noyau, soit pour en régénérer un, soit pour mieux pouvoir déboguer le noyau. Cette option vous donnera accès à l'option « Enable access to .config through /proc/config.gz », qui permet de créer une entrée dans le système de fichiers virtuel /proc/ afin de récupérer ces informations. Ces options étant plus réservées aux développeurs du noyau qu'aux utilisateurs normaux, la réponse recommandée est 'N'.

L'option « Kernel log buffer size (16 => 64KB, 17 => 128KB) » permet de spécifier la taille du tampon circulaire utilisé pour l'affichage des messages de traces du noyau. Si cette taille est trop petite et que le noyau génère beaucoup de messages lors du démarrage ou lors du chargement du périphérique, certains messages peuvent être perdus. De ce fait, le débogage de la configuration peut en être compliqué, et cette option devient alors utile. La taille indiquée

est la puissance de deux à utiliser pour spécifier la taille effective du tampon. Les noyaux utilisant l'ACPI peuvent générer beaucoup de messages de traces lors de la détection du matériel, aussi est-il recommandé de fixer la taille de ce tampon à 15 au moins.

L'option « Cpuset support » permet d'activer les fonctionnalités de partitionnement du système en groupes de processeurs et de nœuds de mémoire sur les architectures multiprocesseurs NUMA. La réponse recommandée est 'N'.

L'option « Create deprecated sysfs files » permet de prendre en charge à des fins de compatibilité ascendante certains fichiers du système de fichiers virtuel /sys/ qui sont devenus obsolètes. Ces fichiers ne doivent plus être utilisés de nos jours, aussi est-il recommandé de répondre 'N' à cette question, sauf si votre distribution est antérieure à 2006.

L'option « Kernel->user space relay support (formerly relayfs) » permet de prendre en charge une interface spéciale exposée par certains systèmes de fichiers virtuels permettant de transférer rapidement des grandes quantités de données entre le noyau et l'espace utilisateur. La réponse recommandée est 'N'.

L'option « Initial RAM filesystem and RAM disk (initramfs/initrd) support » permet d'activer la gestion des systèmes de fichiers virtuels en mémoire que les gestionnaires de chargement peuvent utiliser pour charger le système de fichiers racine lors du démarrage sans disque. Cette fonctionnalité est utilisée par les programmes d'installation, les disquettes de récupération et les Live CD de démonstration. Cette option vous donnera accès à l'option « Initramfs source file(s) » et aux options « User ID to map to 0 (user root) » et « Group ID to map to 0 (group root) », qui permettent d'activer la prise en charge du système initrd et d'indiquer respectivement un identifiant d'utilisateur et un identifiant de groupe pour les fichiers qui devront être affectés au superutilisateur dans le système de fichiers racine du disque virtuel. Cette n'étant généralement utilisée que pour les programmes d'installation, vous pouvez répondre 'N' à cette question.

L'option « Optimize for size (Look out for broken compilers!) » permet d'optimiser la compilation du noyau pour réduire sa taille. Certaines versions de GCC génèrent du code faux avec ces options, auquel cas une mise à jour s'imposera. La réponse recommandée est 'N'.

L'option « Configure standard kernel features (for small systems) » permet d'accéder à un sous-menu contenant des options de fonctionnalités que l'on désactive généralement lorsque l'on veut faire un noyau de petite taille, par exemple pour un système embarqué. Les options de ce sous-menu sont décrites ci-dessous. La réponse recommandée est 'N'.

L'option « Choose SLAB allocator » permet de choisir l'allocateur mémoire utilisé en interne pour les opérations de gestion mémoire courantes dans le noyau. Deux algorithmes sont disponibles, l'algorithme classique (SLAB) et un nouvel algorithme plus performant (SLUB). Le nouvel algorithme est appelé à remplacer l'ancien, mais n'est pas forcément totalement stabilisé. Il est donc recommandé d'utiliser l'ancien algorithme pour quelque temps encore et la réponse recommandée est 'SLAB'.

XII-A-1 - Sous-menu « Configure standard kernel features (for small systems) »

L'option « Enable 16-bit UID system calls », permet de désactiver le support des appels systèmes utilisant les identifiants d'utilisateurs codés sur 16 bits, pour ne conserver que les appels systèmes 32 bits. La réponse recommandée est 'N', car des programmes peuvent encore utiliser des identifiants 16 bits.

L'option « Sysctl syscall support », permet d'activer l'appel système sysctl permettant de modifier les paramètres de configuration généraux du noyau. Cet appel système s'est avéré difficile à maintenir et a été remplacé par un accès plus simple à ces paramètres via le système de fichiers virtuel /proc/. Il est donc possible de désactiver cet appel système sur les systèmes les plus récents. Toutefois, on répond 'Y' pour assurer la compatibilité ascendante.

L'option « Load all symbols for debugging/kksymbols », permet de supprimer les informations symboliques de débogage du noyau pour le cas où celui-ci planerait. Cela permet de réduire la taille du noyau sensiblement. La réponse recommandée est 'N'. La sous-option « Include all symbols in kallsyms » permet d'ajouter toutes les informations symboliques de débogage au noyau. Ces informations ne sont pas utilisées par le noyau en général,

mais peuvent l'être par certains débogueurs. La réponse recommandée est donc 'N'. La sous-option « `Do an extra kallsyms pass` » est une option qui ne doit normalement jamais être activée. Elle permet de contourner un bogue de génération du noyau qui est en attente de correction. La réponse recommandée est 'N'.

L'option « `Support for hot-pluggable devices` » permet d'activer la gestion des périphériques connectables à chaud (c'est-à-dire pendant que le système fonctionne). Parmi ces périphériques, on rencontre couramment les cartes PCMCIA des portables, mais également les périphériques USB et FireWire, ainsi que les cartes PCI connectables à chaud. Cette option est en particulier nécessaire pour l'utilisation des cartes PCMCIA sur les portables. Elle vous donnera l'accès au menu « `PCCARD (PCMCIA/CardBus) support` », qui permet d'activer la gestion des cartes PCMCIA 32 bits (cela n'est pas nécessaire pour utiliser les cartes PCMCIA 16 bits), au menu « `PCI Hotplug Support` », qui permet de prendre en charge la gestion des cartes PCI connectables à chaud, et à l'option « `Hotplug firmware loading support` », qui permet le chargement des firmwares via une interface standard du noyau par les gestionnaires de périphériques. Cette option est facultative, mais fortement conseillée, pour l'utilisation des périphériques USB. Ces périphériques ne seront parfois pas configurés automatiquement lorsque vous les connecterez à chaud si vous ne l'activez pas. En fait, pour que la configuration des périphériques USB connectés à chaud fonctionne, vous devez également activer la gestion des modules du noyau, ainsi que les options « `Kernel module loader` » et « `Hotplug firmware loading support` ». Si cette option est activée, le noyau appellera alors le programme de configuration `udev`, ou, selon votre distribution, le programme `hotplug` pour charger le gestionnaire de périphériques approprié lorsqu'un périphérique sera connecté et pour les configurer. La réponse recommandée est 'Y'.

L'option « `Enable support for printk` » permet d'activer les traces du noyau. Ces traces sont parfois désactivées afin d'en réduire la taille. Cette fonctionnalité est utile pour les systèmes embarqués, mais en exploitation normale ces traces sont souvent d'une grande aide. Aussi la réponse recommandée est-elle 'Y'.

L'option « `BUG() support` » permet d'activer le support des vérifications des erreurs fatales ou irrécupérables dans le noyau. Ces vérifications sont parfois désactivées dans les systèmes embarqués, mais elles sont vivement recommandées pour les systèmes normaux. La réponse recommandée est 'Y'.

L'option « `Enable ELF core dumps` » permet de prendre en charge la fonctionnalité de vidage sur incident de la mémoire des processus dans les fichiers « `core` ». Supprimer cette fonctionnalité peut économiser un peu de mémoire. La réponse recommandée est 'Y'.

L'option « `Enable full-sized data structures for core` » permet d'autoriser l'utilisation des structures de données complètes des algorithmes du noyau. Certaines structures peuvent être réduites en taille, moyennant une perte de performances. Cela peut être utile pour les systèmes embarqués, mais pour les systèmes normaux, il faut répondre 'Y'.

L'option « `Enable futex support` » permet de réaliser des synchronisations rapides entre les différents processus utilisateurs. Cette fonctionnalité est nouvelle et n'est pas forcément utilisée par les applications des systèmes embarqués, auquel cas elle pourra être supprimée. Sur un système normal en revanche, cette option est intéressante, aussi la réponse recommandée est-elle 'Y'.

L'option « `Enable eventpoll support` » permet d'optimiser les opérations d'entrée/sortie des processus. Cette fonctionnalité étant récente, certains systèmes embarqués, qui ne l'utilisent pas, peuvent ne pas en avoir besoin. Un système classique en revanche peut en bénéficier, aussi la réponse recommandée est-elle 'Y'.

Les options « `Enable signalfd() system call` », « `Enable timerfd() system call` », et « `Enable ioperfd() system call` » permettent de prendre en charge des appels systèmes qui peuvent être utiles à des applications pour gérer les différents événements que le système d'exploitation peut produire de manière générique. Ces fonctionnalités sont très utiles et de nombreux programmes les utiliseront sous peu, aussi est-il recommandé de répondre 'Y' à ces questions.

L'option « `Use full shmем filesystem` » permet d'activer la gestion complète du système de fichiers virtuel `/dev/shm/`, qui est utilisé pour la gestion des segments de mémoire partagés. En désactivant cette option, ce système de fichiers est remplacé par un système de fichiers virtuel en mémoire plus léger, ce qui peut être utile pour des configurations sans swap. La réponse recommandée est 'Y'.

L'option « **Enable VM event counters for /proc/vmstat** » permet d'activer la prise en charge des statistiques concernant la gestion de la mémoire. Ces statistiques ne sont pas nécessaires pour le noyau, aussi est-il possible de ne pas les prendre en charge afin d'économiser de la place. La réponse recommandée est 'Y'.

L'option « **Enable SLUB debugging support** » permet d'activer la gestion des fonctions de débogage et de diagnostic du nouvel allocateur mémoire SLUB. La réponse recommandée est 'Y'.

XII-B - Menu « Loadable module support »

L'option « **Enable loadable module support** » permet l'emploi des modules par le noyau. Il est recommandé de répondre 'Y' à cette question.

L'option « **Module unloading** » permet de décharger des modules en cours d'exécution. Les modules ne seront déchargés que s'ils ne sont pas utilisés, toutefois, il est possible de forcer le déchargement d'un module même si le noyau considère qu'il est utilisé (ce qui est bien entendu extrêmement dangereux). Pour cela, il faut activer la sous-option « **Forced module unloading** ». Il est recommandé de répondre 'Y' à ces questions.

L'option « **Module versioning support (EXPERIMENTAL)** » permet d'enregistrer des informations de version dans les modules du noyau afin de détecter les conflits entre les modules compilés pour une autre version du noyau ou compilés par un autre compilateur que celui qui a compilé le noyau. Cette fonctionnalité peut permettre l'emploi des modules normalement inutilisables, mais elle ne garantit pas un fonctionnement correct pour autant. Il est donc recommandé de répondre par 'N' à cette question.

L'option « **Source checksum for all modules** » permet de demander le calcul d'une somme de contrôle sur les fichiers source des modules du noyau. Cela permet aux développeurs du noyau d'être certains que les sources de ces modules n'ont pas été modifiées et sont identiques aux leurs lorsqu'ils analysent un rapport d'anomalie. Cette fonctionnalité n'est normalement pas utile pour un utilisateur standard, aussi la réponse recommandée à cette question est-elle 'N'.

L'option « **Automatic kernel module loading** » donne la possibilité au noyau de charger lui-même les modules du noyau. Cette fonctionnalité est nécessaire pour la bonne marche du système, aussi est-il recommandé de répondre par 'Y' à cette question.

XII-C - Menu « Block layer »

L'option « **Enable the block layer** » permet de désactiver la prise en charge des périphériques de type bloc, ce qui peut être utile pour les noyaux destinés à des appareils embarqués. Cela supprime des fonctionnalités essentielles aux systèmes de fichiers, aussi la réponse recommandée est-elle 'N'.

L'option « **Support for Large Block Devices** » permet de prendre en charge les périphériques de type bloc de grandes dimensions (c'est-à-dire plus de deux téra-octets). La réponse recommandée est 'N'.

L'option « **Support for tracing block io actions** » permet de prendre en charge le suivi des opérations sur les périphériques de type bloc à des fins d'analyse. La réponse recommandée est 'N'.

L'option « **Support for Large Single Files** » permet de prendre en charge les fichiers de très grandes dimensions (c'est-à-dire de taille supérieure à deux téra-octets). La réponse recommandée est 'N'.

L'option « **Block layer SG support v4 (EXPERIMENTAL)** » permet de prendre en charge une nouvelle implémentation des périphériques SCSI générique, qui est utilisée par les périphériques SCSI, les graveurs et les disques SATA. Cette nouvelle implémentation permet de prendre en charge des opérations plus complexes que l'ancienne implémentation, mais elle est encore marquée expérimentale. Aussi la réponse recommandée est-elle 'N'.

XII-C-1 - Sous-menu « IO Schedulers »

L'option « Anticipatory I/O scheduler » active l'ordonnanceur d'entrées/sorties par défaut de Linux. Cet ordonnanceur peut effectuer les opérations de manière anticipée lorsqu'il en a la possibilité, afin d'avoir évité de les faire de manière bloquante lorsque le système sera peut-être plus occupé à des opérations plus importantes que les entrées/sorties. Cette politique permet donc d'augmenter les performances de manière générale, sauf dans de rares cas d'utilisation, essentiellement lors de l'utilisation de certaines bases de données. La réponse recommandée est 'Y'. Cet ordonnanceur est l'ordonnanceur par défaut de Linux.

L'option « Deadline I/O scheduler » active un ordonnanceur plus simplifié et plus compact, relativement correct et parfois plus rapide que l'ordonnanceur de l'option « Anticipatory I/O scheduler », en particulier avec certaines bases de données. Cette option peut donc être utile dans certaines applications serveur. La réponse recommandée est 'Y'. Il peut être sélectionné comme ordonnanceur par défaut du système au démarrage avec la ligne de commande « `elevator=deadline` » du noyau.

L'option « CFQ I/O scheduler » active un ordonnanceur qui répartit les requêtes d'entrée/sortie de manière équitable entre tous les processus du système. Cet ordonnanceur garantit donc une meilleure répartition des accès aux périphériques que les autres, ce qui le rend très adapté pour les systèmes où la réactivité des processus doit être maximale. C'est l'ordonnanceur de prédilection pour les ordinateurs personnels, la réponse recommandée est donc 'Y'. Il peut être sélectionné comme ordonnanceur par défaut du système au démarrage avec la ligne de commande « `elevator=cfq` » du noyau.

L'option « Default I/O scheduler » permet de sélectionner l'ordonnanceur par défaut utilisé par Linux. Si aucun des ordonnanceurs précédents n'est disponible, les opérations d'entrée/sortie se feront dans l'ordre direct des demandes des sous-systèmes utilisateurs (ordonnanceur « No-op »). L'ordonnanceur par défaut peut être modifié en exécution, pour chaque périphérique, via le système de fichier `/sys/`. La réponse recommandée est 'CFQ'.

XII-D - Menu « Processor type and features »

L'option « Symmetric multiprocessing support » permet d'activer le support de plusieurs processeurs. Cela comprend les ordinateurs multiprocesseurs, ainsi que les ordinateurs utilisant un processeur hyperthreadé ou à cœurs cœurs, aussi est-il recommandé de répondre 'Y' à cette question. En pratique, il est même conseillé d'activer cette option même sur les ordinateurs à un seul processeur effectif de nos jours.

L'option « Subarchitecture Type » permet d'accéder à un sous-menu pour indiquer le type de machine pour l'architecture de processeur pour lequel le noyau doit être défini. Pour la plupart des machines basées sur des processeurs x86 ou x86-64 d'AMD, le type d'architecture utilisé est « PC-Compatible », c'est donc l'option recommandée.

L'option « Paravirtualization support » permet d'activer la prise en charge de l'exécution du noyau au sein d'une machine virtuelle sur les processeurs disposant des fonctions de paravirtualisation. Cette option n'est utile que pour les systèmes virtualisés, qui sont ainsi modifiés afin de s'exécuter de manière plus performante dans une machine virtuelle. Toutefois, elle peut ralentir l'exécution du noyau en dehors de ces machines virtuelles. Cette option vous donnera accès aux options de configuration des deux principales technologies de virtualisation, à savoir Xen (option « Enable support for Xen hypervisor ») et VMWare (option `VMI Paravirt-ops support`). La réponse recommandée est donc 'N'.

L'option « Processor family » vous permet de spécifier le type de processeur sur lequel le noyau fonctionnera. Choisissez le processeur dont vous disposez.

L'option « Generic-x86-64 » permet d'activer la prise en charge du jeu d'instructions 64 bits commun à Intel et AMD, et donc de réaliser des noyaux portables. Cette option est destinée aux noyaux de distributions, qui doivent être génériques. Il est toutefois recommandé de choisir le type processeur effectivement utilisé.

L'option « **Generic x86 support** » est similaire à la précédente, mais pour les processeurs 32 bits. Elle permet d'activer la prise en charge d'optimisation pour certains processeurs x86, même si le noyau n'est pas compilé spécifiquement pour ces processeurs. Cette option est destinée aux noyaux de distributions, qui doivent être génériques, mais qui peuvent chercher à bénéficier de certaines fonctionnalités des processeurs récents. Il est donc recommandé de répondre 'N' à cette option et de choisir le type processeur adéquat dans l'option « **Processor family** ».

L'option « **HPET Timer Support** » permet d'activer la prise en charge des horloges temps-réel de nouvelle génération sur les systèmes les plus récents. En l'absence d'une telle horloge, l'horloge standard sera utilisée. Vous pouvez répondre 'Y' à cette question si vous disposez d'un ordinateur récent, autrement répondez 'N'.

L'option « **Provide RTC interrupt** » permet d'activer l'émulation de l'horloge temps réel RTC sur les machines 64 bits, qui utilisent normalement systématiquement les horloges temps réel HPET. La réponse recommandée est 'Y'.

L'option permet de sélectionner le nombre maximum de processeurs disponibles sur la machine. Cette option n'est accessible que si vous avez activé la prise en charge des processeurs multiples en répondant par 'Y' à l'option « **Symmetric multiprocessing support** ». Vous devez dans ce cas indiquer le nombre de processeurs réels ou virtuels dont votre machine dispose. Chaque processeur virtuel des processeurs hyperthreadés compte pour un processeur à part entière, ainsi que chaque cœur des processeurs multicœurs.

L'option « **SMT (Hyperthreading) scheduler support** » permet de prendre en charge un ordonnanceur de tâches spécifiques pour les processeurs hyperthreadés. Ces processeurs partageant les mêmes caches de mémoire, la migration des tâches d'un processeur virtuel à un autre peut en effet être optimisée. Cet ordonnanceur prend donc en compte les spécificités des processeurs hyperthreadés, tels que les derniers pentium IV. Répondez par 'Y' si vous disposez d'un tel processeur, et par 'N' dans le cas contraire.

De la même manière, l'option « **Multicore scheduler support** » permet de prendre en charge un ordonnanceur de tâches spécifiques pour les processeurs multicœurs. Répondez par 'Y' si vous disposez d'un tel processeur, et par 'N' dans le cas contraire.

L'option « **Preemption Model** » permet de spécifier si le noyau lui-même peut être préempté pendant les appels systèmes ou non. La préemption des appels systèmes rend le noyau lui-même multitâche, et permet donc d'augmenter la réactivité du système. Dans certaines circonstances, cela peut aussi accroître le taux d'utilisation du processeur. Toutefois, cela peut aussi diminuer légèrement la bande passante dans les opérations d'entrée / sortie. L'activation du multitâche dans le noyau est recommandée pour les noyaux destinés à des applications temps réel ou multimédia, et déconseillée pour les noyaux destinés aux serveurs. De plus, si vous décidez d'autoriser la prise en charge du multitâche au sein des appels systèmes, vous pourrez faire en sorte que celui-ci soit coopératif (le noyau lui-même vérifiera si d'autres tâches n'ont pas besoin d'être exécutées) ou préemptif (le noyau pourra être interrompu à n'importe quel moment), ce qui augmentera la réactivité du système. Le choix recommandé est « **Preemptible Kernel (Low-Latency Desktop)** », sauf si vous envisagez d'installer un serveur qui sera utilisé intensivement.

L'option « **Preempt The Big Kernel Lock** » permet d'autoriser les mécanismes de préemption même au sein de la section critique globale du noyau. Cette option permet d'améliorer encore plus la réactivité du système, et est recommandée pour les noyaux destinés à des machines de bureau. La réponse recommandée est 'Y'.

L'option « **Local APIC support on uniprocessors** » permet d'activer les contrôleurs d'interruptions programmables intégrés dans certains processeurs. Certaines machines disposent d'un tel processeur, si c'est votre cas, vous pouvez répondre par 'Y' à cette question. Cette option n'est disponible que si vous avez répondu 'N' à la question « **Symmetric multiprocessing support** ».

L'option « **IO-APIC support on uniprocessors** » permet d'activer la gestion des contrôleurs d'interruption programmables avancés. Ces contrôleurs sont utilisés sur les machines multiprocesseurs, mais certaines cartes mères monoprocesseurs les utilisent. Si c'est le cas de votre carte mère, vous pouvez répondre par 'Y' à cette question.

L'option « **Machine Check Exception** » permet d'activer la surveillance de l'état de l'ordinateur (élévation anormale de température, pannes, etc.) afin de contourner les problèmes matériels ou d'arrêter le système avant une destruction

de composants ou des pertes de données irrémédiables. L'option « **Check for non-fatal errors on AMD Athlon/Duron / Intel Pentium 4** » permet d'activer la surveillance des problèmes mineurs, qui ne nécessitent pas un arrêt de la machine, mais pour lesquels on désire malgré tout avoir une trace dans les traces du noyau. L'option « **check for P4 thermal throttling interrupt.** » permet de générer une trace sur les systèmes à base de Pentium IV lorsque ceux-ci surchauffent. Pour les machines 64 bits, l'option « **Intel MCE features** » et « **AMD MCE features** » permettent d'activer les fonctions de surveillance du système telles que la température du processeur et les erreurs du bus mémoire. La réponse recommandée est 'Y' pour ces options si votre système dispose d'un processeur de ce type.

L'option « **Enable VM86 support** » permet de prendre en charge le mode 8086 virtuel des processeurs x86 32 bits. Ce mode permet de faire fonctionner les programmes 16 bits, tels que les vieux programmes DOS ou certaines fonctions du BIOS qui n'existent qu'en mode 16 bits. Cette option est en particulier nécessaire pour permettre une exécution performante des émulateurs DOS tels que DOSEMU, ou pour changer de mode graphique sur certaines cartes graphiques. La réponse recommandée est 'Y'.

L'option « **Toshiba Laptop support** » permet d'activer la gestion d'énergie pour les portables de marque Toshiba. Vous pouvez répondre 'Y' à cette question si vous désirez utiliser votre noyau sur un portable Toshiba. La réponse recommandée est 'N'.

L'option « **Dell Laptop support** » permet d'activer la gestion d'énergie pour les portables de marque Dell. Vous pouvez répondre 'Y' à cette question si vous désirez utiliser votre noyau sur un portable Dell. La réponse recommandée est 'N'.

L'option « **Enable X86 board specific fixups for reboot** » permet d'activer un contournement à un bogue matériel et logiciel dans le BIOS présent sur certaines machines et empêchant le système de redémarrer correctement. La réponse recommandée est 'N'.

L'option « **/dev/cpu/microcode - Intel IA32 CPU microcode support** » permet d'activer la reprogrammation du microcode des processeurs Intel postérieurs aux Pentium Pro. À l'aide de cette fonctionnalité, vous pourrez mettre à jour le microcode de votre processeur (cette mise à jour n'est pas permanente, le microcode doit être rechargé à chaque démarrage de la machine). Notez que cette option impose d'utiliser le système de fichiers virtuel `/dev/` du noyau. Il est donc nécessaire d'activer l'option « **/dev filesystem support (EXPERIMENTAL)** » du menu « **File systems** ». En général, il n'est pas nécessaire de mettre à jour le microcode de son processeur, à moins que celui-ci ne contienne un bogue incontournable ou que vous n'ayez à optimiser spécialement votre machine pour une utilisation particulière. La réponse recommandée est donc 'N'.

L'option « **/dev/cpu/*/msr - Model-specific register support** » permet d'activer la gestion du registre MSR sur les processeurs x86. Il existe un registre de ce type pour chaque processeur, que l'on trouvera dans les sous-répertoires portant comme nom les numéros de chaque processeur. La réponse recommandée est 'N'.

L'option « **/dev/cpu/*/cpuid - CPU information support** » permet d'obtenir les informations fournies par l'instruction CPUID pour chaque processeur de la machine. Il existe un registre de ce type pour chaque processeur, que l'on trouvera dans les sous-répertoires portant comme nom les numéros de chaque processeur. La réponse recommandée est 'N'.

L'option « **High Memory Support** » vous permet d'indiquer le mode de gestion de la mémoire que le noyau utilisera. Pour la plupart des gens, dont les machines disposent de moins d'un Go de mémoire, la réponse recommandée est « **off** ». Si votre machine dispose d'un gigaoctet ou plus de mémoire, mais moins de quatre Go, vous pouvez changer le mode d'adressage du noyau pour gérer la totalité de votre mémoire, en choisissant l'option « **4GB** ». Les processeurs x86 ne permettent pas d'adresser plus de 4Go de mémoire simultanément en raison de leur bus d'adresses 32 bits. Cependant, ils peuvent gérer jusqu'à 64Go de mémoire par tranches de 4Go, à l'aide d'une extension spécifique aux processeurs de type Pentium Pro. Si vous disposez d'une telle machine, vous devez choisir l'option « **64GB** ».

L'option « **Memory Split** » permet de choisir le type de découpage de l'espace d'adressage des processus entre la plage d'adresses utilisable par les programmes et la plage d'adresse correspondante à une copie de l'espace d'adressage du noyau. Sous Linux, la mémoire du noyau est projetée systématiquement dans l'espace d'adressage de chaque programme utilisateur (mais n'est pas lisible pour des raisons de sécurité évidentes), afin de permettre au noyau d'accéder directement à la mémoire du programme lors des appels systèmes. Il est donc d'usage de réserver le

gigaoctet supérieur des quatre gigaoctets d'espace d'adressage des processeurs 32 bits pour y placer une projection de la mémoire du noyau. Cependant, si le système dispose de plus d'un gigaoctet de mémoire, le noyau ne pourra pas accéder directement à la totalité de la mémoire sans frais supplémentaires, car il devra modifier cette projection. Cette option permet donc de spécifier la manière dont ce découpage est réalisé. La valeur classique est « 3G/1G user/kernel split », ce qui est donc la valeur recommandée. Une variante est disponible pour les machines disposant exactement d'un gigaoctet de mémoire. Les autres options permettent d'accroître la taille de l'espace d'adressage du noyau, au détriment de l'espace d'adressage des processus. Utiliser ces options nécessite de bien connaître les processus utilisés, car réduire leur espace d'adressage peut conduire à des situations de fragmentation mémoire ou d'échecs de réservation de mémoire susceptibles de les empêcher de fonctionner correctement.

L'option « PAE (Physical Address Extension) Support » permet d'activer le mode d'adressage des pages étendu sur les processeurs x86. Ce mode est nécessaire pour que le bit NX soit utilisable (bit permettant d'éviter que les zones de données des programmes soient exécutables, et donc éviter toute une catégorie d'attaque des virus et des pirates sur des programmes bogués leur permettant de contrôler une partie de leur mémoire). Toutefois, ce mode étendu consomme plus de ressources dans le système. La réponse recommandée est 'N'. Si vous désirez absolument bénéficier de la protection du bit NX et d'une grande quantité de mémoire vive, il est préférable d'utiliser une distribution 64 bits, ce qui évite d'avoir recours à des contournements plus ou moins efficaces.

L'option « Non Uniform Memory Access (NUMA) Support » (également appelée « NUMA Memory Allocation and Scheduler Support (EXPERIMENTAL) » dans la configuration pour les processeurs 32 bits) permet de prendre en charge les ordinateurs disposant de plusieurs types de mémoire, situés sur les bus de différents processeurs. Ces mémoires ont la spécificité d'avoir un temps d'accès qui n'est pas uniforme et dépend du processeur qui y accède. Ainsi, dans ce modèle, il est important que les processus utilisent au maximum la mémoire la plus proche du processeur sur lequel ils s'exécutent, aussi des traitements particuliers doivent-ils être réalisés lors de l'allocation de mémoire et lors de l'ordonnancement. Les fonctionnalités NUMA pour les processeurs 32 bits sont extrêmement expérimentales et ne doivent jamais être utilisées. Sur les machines 64 bits, elles sont tout à fait utilisables, mais les machines de ce type sont encore relativement rares. De ce fait, la réponse recommandée à cette option est 'N'. Cette option vous donnera accès aux options « Old style AMD Opteron NUMA detection » et « ACPI NUMA detection », qui permet de choisir la méthode de détection des zones de mémoire non uniformes (directement via le bus mémoire ou par les informations de l'ACPI). Elle vous donnera également accès à l'option « NUMA emulation », qui permet de simuler une machine à mémoire non uniforme. Cette option est réservée aux développeurs du noyau, et la réponse recommandée est 'N'.

L'option « Memory model » permet de choisir le modèle utilisé par l'ordinateur pour gérer la mémoire physique. La plupart des ordinateurs ont un modèle mémoire continu et linéaire, ce qui correspond à l'option « Flat Memory » (valeur par défaut recommandée). D'autres ordinateurs peuvent utiliser des modèles où la mémoire est constituée de différentes zones disjointes, et nécessiter ainsi le choix « Sparse Memory ».

L'option « Allow for memory hot-add » permet d'activer la prise en charge de certains serveurs sur lesquels de la mémoire peut être ajoutée à chaud, sans éteindre la machine. La réponse recommandée est 'N'.

L'option « Page migration » permet d'activer la fonctionnalité de déplacement de pages mémoires d'un nœud NUMA vers un autre, afin de rapprocher au maximum les pages mémoires des processus qui les utilisent. La réponse recommandée est 'N'.

L'option « 64 bit Memory and IO resources (EXPERIMENTAL) », permet d'accéder aux ressources mémoires et d'entrée/sortie en 64 bits sur les machines 32 bits. La réponse recommandée est 'N'.

L'option « Allocate 3rd-level pagetables from highmem », permet de placer les structures de données du gestionnaire de mémoire virtuelle en mémoire haute. Ces structures ayant une taille proportionnelle à la quantité de mémoire effectivement installée, cette option permet d'éviter qu'elles ne saturant la mémoire basse. La réponse recommandée pour cette option est 'Y' si vous vous trouvez dans ce cas de configuration.

L'option « Math emulation » permet d'activer l'émulateur d'unité de calcul en virgule flottante, pour les processeurs qui n'en disposent pas (386 et 486 SX). Si vous avez un ordinateur récent, choisissez 'N'.

L'option « **MTRR (Memory Type Range Register) support** » permet d'activer le support des plages mémoires du processeur. Celui-ci peut permettre l'accélération des transferts de données dans les plages mémoires des périphériques, en particulier pour les cartes graphiques. Cette fonctionnalité n'est disponible que pour les processeurs de type Pentium Pro et postérieurs. Si vous avez un ordinateur récent, choisissez 'Y'.

L'option « **Boot from EFI support (EXPERIMENTAL)** » permet de prendre en charge les services systèmes EFI disponibles sur certaines machines lors du démarrage. L'option recommandée est 'N'.

L'option « **Enable kernel irq balancing** » permet de répartir le traitement des interruptions matérielles sur les différents processeurs dont la machine dispose. Cette option n'est disponible que si l'option « **Symmetric multiprocessing support** » a été activée. La réponse recommandée est 'Y'.

L'option « **Enable seccomp to safely compute untrusted bytecode** » permet d'activer le support de « bacs à sable » pour isoler des applications désirant utiliser du code non authentifié dans un système sécurisé. Cette option permet aux applications de restreindre les appels systèmes qu'elles peuvent réaliser, et empêcher ainsi qu'elles soient détournées de leur fonctionnalité initiale. La réponse recommandée est 'N'.

L'option « **Timer frequency** » permet de sélectionner la fréquence d'interruptions de l'horloge du noyau. Les commutations de tâche étant réalisées lors de ces interruptions, une fréquence élevée permet d'obtenir une forte réactivité du système. Toutefois, cette fréquence ne doit pas être trop élevée, afin d'éviter de passer trop de temps à exécuter l'ordonnanceur du système. La fréquence de 1000 Hz est donc la fréquence recommandée pour une machine de bureau, alors que les fréquences inférieures sont recommandées sur les machines disposant d'un grand nombre de processeurs ou devant être utilisées en tant que serveur. La fréquence de 300Hz est également intéressante pour les machines destinées au multimédia, parce que cette fréquence est divisible par les fréquences d'image des films au format PAL et NTSC. La réponse recommandée est « 300 Hz ».

L'option « **IOMMU support** » permet de désactiver le support des accès DMA pour les machines embarquées. Les accès DMA sont utilisés par un grand nombre de pilotes et sont importantes pour les performances, aussi la réponse recommandée est-elle 'Y'.

L'option « **IBM Calgary IOMMU support** » permet de prendre en charge les contrôleurs DMA Calgary d'IBM, qui permettent de faire fonctionner les périphériques PCI 32 bits sur des machines 64 bits disposant de plus de 3 Go de mémoire. Cette fonctionnalité permet également de diagnostiquer des bugs très graves de corruption de la mémoire que certains pilotes de périphériques mal programmés seraient susceptibles de faire. Le noyau fera normalement le bon choix dans l'utilisation de cette fonctionnalité, mais elle reste dédiée à certaines machines seulement. La réponse recommandée est donc 'N'.

L'option « **Should Calgary be enabled by default?** » permet d'activer le contrôleur mémoire Calgary par défaut au démarrage du système. La réponse recommandée est 'Y'.

L'option « **kexec system call (EXPERIMENTAL)** » permet d'activer le support du redémarrage du système directement dans un nouveau noyau, sans avoir à redémarrer complètement la machine. La réponse recommandée est 'N'.

L'option « **kernel crash dumps (EXPERIMENTAL)** » permet d'activer la génération automatique d'un rapport de plantage après chaque redémarrage par l'intermédiaire de l'appel système kexec. Cela permet d'obtenir des informations même après un plantage du noyau sévère. La réponse recommandée est 'N'.

L'option« **Physical address where the kernel is loaded** » permet de spécifier l'adresse de chargement du noyau au démarrage. Cela peut en particulier être utile pour s'assurer, dans le cas où la fonctionnalité de génération automatique de rapport de plantage est activée, que le noyau qui effectue le rapport n'est pas rechargé au même emplacement que celui qui a planté. Il est fortement recommandé de ne pas changer la valeur par défaut de cette option.

L'option« **Build a relocatable kernel** » permet de réaliser un noyau relogeable au-delà du premier mégaoctet de la mémoire. Le noyau ainsi généré est plus gros, mais les informations complémentaires ne sont pas conservées une fois qu'il est chargé en mémoire. La réponse recommandée est 'N'.

L'option « **Alignment value to which kernel should be aligned** » permet d'indiquer la valeur de l'alignement en mémoire pour le chargement du noyau. Le noyau s'arrange pour être exécuté à partir d'une adresse correctement alignée dans tous les cas, mais il est vivement recommandé de ne pas modifier la valeur par défaut de cette option.

L'option « **Support for suspend on SMP and hot-pluggable CPUs (EXPERIMENTAL)** » permet de prendre en charge le branchement à chaud de processeurs dans l'ordinateur. Cette fonctionnalité est un prérequis à la fonction de mise en veille sur disque sur les machines multiprocesseur. La réponse recommandée est 'N'.

L'option « **Enable -fstack-protector buffer overflow detection** » permet d'activer la prise en charge de la détection des débordements de tampons alloués localement sur la pile, qui constituent une catégorie de bugs extrêmement dangereux, car ils peuvent permettre à un pirate de prendre le contrôle du programme fautif. Cette fonctionnalité requiert un compilateur récent. La réponse recommandée est 'N'.

L'option « **Use stack-protector for all functions** » permet d'activer la détection des débordements de tampons pour toutes les fonctions du noyau, et non uniquement celles qui déclarent des tampons sur la pile. La réponse recommandée est 'N'.

L'option « **Function reordering** » permet, si l'on dispose d'outils de compilation relativement récents, de réarranger les fonctions de l'ensemble du noyau afin d'optimiser les accès mémoire pour ne pas polluer le cache du processeur destiné à optimiser les mécanismes de gestion de la mémoire virtuelle. La réponse recommandée est 'N'.

L'option « **Compat VDSO support** » permet de charger la bibliothèque dynamique de réalisation des appels systèmes à un emplacement fixe dans la mémoire des processus. Sous Linux, les appels systèmes sont réalisés via des fonctions placées automatiquement par le noyau dans l'espace d'adressage des processus lors de leur lancement. Ces fonctions sont utilisées par la bibliothèque C, et s'adaptent à la configuration du noyau et aux conventions d'appel qu'il utilise, rendant ainsi la bibliothèque C indépendante du noyau et les appels systèmes plus performants. Les noyaux récents positionnent ces fonctions aléatoirement dans l'espace d'adressage du noyau pour des raisons de sécurité, rendant ainsi difficilement réalisable le détournement des appels systèmes par un attaquant qui aurait réussi à prendre le contrôle d'un programme défaillant. Toutefois, seules les bibliothèques C 2.3.3 ou plus sont capables de gérer ce placement aléatoire des fonctions d'interfaçage avec le noyau. Cette option permet donc, pour les anciennes bibliothèques, d'utiliser l'ancienne politique de placement (en fin d'espace d'adressage du processus). Si vous utilisez un système récent, cela n'est pas nécessaire et la réponse recommandée est 'N'.

XII-D-1 - Sous-menu « Firmware Drivers »

Ce sous-menu permet d'activer l'accès à certaines fonctionnalités fournies par les firmwares de certaines machines. L'option « **BIOS Enhanced Disk Drive calls determine boot disk (EXPERIMENTAL)** » permet de récupérer dans le BIOS les informations sur le disque dur à partir duquel l'amorçage du système s'est fait.

L'option « **EFI Variable Support via sysfs** » permet d'accéder en lecture et en écriture aux variables d'environnement des systèmes amorçables par le mécanisme de boot avancé EFI. La réponse recommandée pour ces deux options est 'N'.

L'option « **Console device selection via EFI PCDP or HCDP table** » permet de définir la console de Linux comme étant la console principale définie par le BIOS EFI. La réponse recommandée pour ces deux options est 'N'.

L'option « **BIOS update support for DELL systems via sysfs** » permet d'activer la fonctionnalité de mise à jour du BIOS via le système de fichiers virtuel `/sys/` pour les ordinateurs DELL. La réponse recommandée est 'N'.

L'option « **Dell Systems Management Base Driver** » permet d'exposer une interface de gestion de l'ordinateur spécifique à certains ordinateurs DELL via le système de fichiers virtuel `/sys/`. La réponse recommandée est 'N'.

L'option « **Export DMI identification via sysfs to userspace** » permet d'exposer les informations DMI via le système de fichiers virtuel `/sys/`. La réponse recommandée est 'N'.

XII-E - Menu « Power management options (ACPI, APM) »

L'option « Power Management support » permet d'activer la gestion d'énergie sur votre machine. Il existe actuellement deux protocoles de gestion d'énergie sur les PC : le protocole APM (abréviation de l'anglais « Advanced Power Management », relativement ancien, mais fiable, et le protocole ACPI (abréviation de l'anglais « Advanced Configuration and Power Interface »), plus récent, mais pouvant poser quelques problèmes avec certains BIOS bogués. Ces deux protocoles peuvent être paramétrés respectivement avec les deux jeux d'options suivantes. La réponse recommandée est 'Y'.

L'option « Legacy Power Management API » permet de prendre en charge l'ancienne interface du noyau de gestion de l'énergie. Cela peut être nécessaire pour les anciens pilotes binaires ou qui n'ont pas été portés. La réponse recommandée est 'Y'.

L'option « Power Management Debug Support » permet d'activer les fonctionnalités de débogage du code de gestion de l'énergie de Linux. Cette option est essentiellement utile aux développeurs de Linux. La réponse recommandée est 'N'.

L'option « Keep console(s) enabled during suspend/resume » permet de ne pas suspendre le pilote de la console afin de pouvoir voir les messages de débogage des pilotes de périphériques lors de la suspension. Cette option peut être dangereuse et n'est utile que pour les développeurs du noyau, aussi la réponse recommandée est-elle 'N'.

L'option « Suspend/resume event tracing » permet de prendre en charge une fonction de mémorisation des dernières opérations de mise en veille dans l'horloge système, afin de pouvoir déboguer les pilotes de périphériques lorsque la machine n'est pas parvenue à se mettre en veille correctement. Cette fonctionnalité n'est utile que pour les développeurs du noyau, aussi la réponse recommandée est-elle 'N'.

L'option « Suspend to RAM and standby » permet de prendre en charge la mise en veille du système en mémoire. Cette forme de mise en veille est extrêmement rapide, mais nécessite une alimentation électrique pour conserver l'état du système, car celui-ci est stocké en mémoire. Cette technique fonctionne relativement bien pour les ordinateurs portables, mais certains pilotes de périphériques ne sont pas encore capables de redémarrer correctement à la sortie de la mise en veille. La réponse recommandée est 'N'.

L'option « Hibernation (aka 'suspend to disk') » permet de prendre en charge la suspension du système sur disque. Cette forme de mise en veille est beaucoup plus lente que la mise en veille en mémoire, car tout l'état du système doit être stocké sur disque. De plus, la sortie de la mise en veille nécessite un redémarrage complet du noyau, et le système est généralement plus lent après le réveil le temps que toutes les données stockées sur disque soient rechargées en mémoire. Toutefois, cette technique est beaucoup plus fiable et fonctionne même sur les machines dont le matériel ne supporte pas la suspension en mémoire. La réponse recommandée est 'N'.

L'option « Default resume partition » permet de spécifier la partition de swap utilisée par défaut par la fonctionnalité de suspension sur disque. Il faut indiquer ici une partition de swap de taille suffisante pour stocker l'état du système lors de la suspension. Cette option est facultative, la partition à utiliser au redémarrage pouvant être spécifiée par l'option resume en ligne de commande du noyau. Toutefois, il est plus simple de l'indiquer ici une bonne fois pour toutes. La réponse dépend bien entendu de votre plan de partitionnement.

L'option « Advanced Power Management Emulation » permet de demander au système de simuler la mise en veille selon l'ancienne gestion d'énergie APM. Cela comprend la réinitialisation de l'horloge système, ainsi que la notification des changements d'état de la batterie aux programmes de gestion de l'APM. La réponse recommandée est 'N'.

L'option « ACPI Support (Advanced Configuration and Power Interface) » du menu « ACPI (Advanced Configuration and Power Interface) Support » permet d'activer la gestion du protocole de gestion d'énergie ACPI. Cette interface permet au système d'exploitation de contrôler finement la consommation d'énergie du système, mais nécessite que votre carte mère le gère ainsi que tous les périphériques connectés dessus. C'est généralement le cas pour toutes les machines récentes, et quasiment certain pour les ordinateurs portables, mais si ce n'est pas le cas, les périphériques qui seront mis en veille ne se réinitialiseront pas correctement lors du réveil de l'ordinateur. Cela peut provoquer des plantages violents. La réponse recommandée est 'Y', à moins que vous ayez un doute sur la capacité de votre

matériel à supporter l'ACPI. Notez que certains BIOS fournissent une description erronée du matériel de l'ordinateur à Linux et n'ont été testés que pour Microsoft Windows. Ces informations sont généralement générées avec les outils Microsoft, qui ne respectent pas la norme ACPI, mais dont Windows tolère les erreurs. Intel, le principal promoteur de l'ACPI, et les développeurs de Linux, ont décidé de ne pas prendre en charge ces erreurs de configuration et indiquent qu'il est du ressort des fabricants de fournir des informations correctes. Toutefois, ils permettent aux utilisateurs de spécifier au niveau de la configuration une description alternative du matériel. Cela est réalisable grâce aux options « Include Custom DSDT » et « Custom DSDT Table file to include ». Ces options ne sont accessibles que si l'option « Select only drivers that don't need compile-time external firmware » du menu « Generic Driver Options » n'est pas cochée. Notez enfin que si vous choisissez d'activer le protocole de gestion d'énergie ACPI et le protocole APM, celui qui sera utilisé sera le premier chargé.

L'option « APM (Advanced Power Management) BIOS support » du menu du même nom permet d'activer la gestion d'énergie APM par l'intermédiaire du BIOS. Cette méthode de gestion d'énergie est beaucoup plus sûre que l'ACPI, car elle est plus ancienne et en général parfaitement supportée par le matériel actuel. Il est recommandé de répondre par 'Y' à cette question si vous ne pouvez pas utiliser l'ACPI. Notez toutefois que la possibilité d'éteindre l'ordinateur via les fonctionnalités APM n'est disponible que pour les ordinateurs disposant d'un boîtier au format ATX. De plus, ce n'est pas la gestion d'énergie APM qui gère l'arrêt des disques durs et la veille des moniteurs « Green », et il est possible d'avoir ces fonctionnalités même si l'on n'a pas activé la gestion d'énergie APM. Les options suivantes dépendent fortement de la configuration APM des machines. Les options de ce menu ne seront pas décrites plus en détail ici, car elles sont trop spécifiques à chaque modèle de machine. Pour la plupart des gens, il faut répondre 'N' à toutes ces questions.

L'option « CPU Frequency scaling » du menu du même nom permet d'activer la gestion des processeurs à fréquence variable. Ces processeurs se rencontrent généralement dans les portables, à cause du gain d'énergie obtenu en abaissant la fréquence du processeur lorsqu'il est peu utilisé. On les rencontre également sur les ordinateurs de bureau récents, car cela permet de diminuer la vitesse de rotation des ventilateurs et d'obtenir ainsi des machines plus silencieuses. Les options qui suivent permettent de spécifier l'interface logicielle fournie par le noyau pour contrôler la fréquence du processeur, ainsi que d'indiquer la nature du processeur utilisé et la politique de gestion de la fréquence du processeur. La réponse recommandée est 'Y', sauf si vous utilisez une machine relativement ancienne. Vous devrez alors choisir le pilote pour le type de processeur dont vous disposez, ainsi que le gestionnaire d'énergie à utiliser. Les plus classiques sont « 'performance' governor » (pour utiliser en permanence la fréquence la plus élevée), « 'powersave' governor » (pour utiliser en permanence la fréquence la plus faible), « 'ondemand' governor » (pour que le noyau adapte la fréquence à la charge courante du système) et « conservative » (pour que le noyau adapte la fréquence à la charge courante du système, mais en passant d'une fréquence à une autre de manière plus progressive et en cherchant à optimiser la consommation électrique). Notez que dans le cas de l'adaptation à la demande de la fréquence du processeur les processus placés en priorité faible ne seront pas comptabilisés comme des processus devant accroître la fréquence du processeur, sauf si le fichier de configuration dynamique /sys/devices/system/cpu/cpu0/cpufreq/ondemand/ignore_nice_load contient la valeur 0 (attention, ce paramètre est en logique inverse étant donné le nom de ce fichier).

L'option « CPU idle PM support » permet de prendre en charge la gestion d'énergie des processeurs au repos. Cette gestion se fait au travers de gestionnaires logiciels en fonction de la charge du système. Cette option vous donnera l'accès aux deux options « 'ladder' governor (NEW) » et « 'menu' governor (NEW) », qui utilisent respectivement l'algorithme des anciens pilotes ACPI et un algorithme plus récent dont le but est de minimiser la consommation d'énergie. La réponse recommandée est 'Y'.

XII-F - Menu « Bus options (PCI, PCMCIA, EISA, MCA, ISA) »

L'option « PCI support » permet d'activer le support des ordinateurs à base de bus PCI. Si vous avez un ordinateur récent, répondez par 'Y'.

L'option « PCI access mode » de sélectionner le composant qui initialisera les bus PCI. Si vous avez un ordinateur récent, avec un BIOS récent, choisissez « BIOS ». Sinon, choisissez « Direct ». Le choix « Any » permet de demander à Linux d'essayer l'initialisation par le BIOS, et de faire le travail lui-même si ce dernier est défaillant. Vous pouvez donc toujours choisir l'option « Any ».

L'option « **Support mmconfig PCI config space access** » n'est pas documentée et ne sera pas décrite plus en détail dans ce document. Cette option n'est disponible que pour les machines 64 bits. La réponse recommandée est 'Y'.

L'option « **PCI Express support** » permet d'activer le support du nouveau bus PCI Express. Si vous avez un ordinateur récent, répondez par 'Y'.

L'option « **PCI Express Hotplug driver** » permet d'activer le support des périphériques PCI Express connectables à chaud. Si vous avez une carte mère permettant ce type d'opération, répondez par 'Y'.

L'option « **Use polling mechanism for hot-plug events (for testing purpose)** » permet d'utiliser un mécanisme de scrutation des périphériques pour détecter les périphériques PCI Express. Cela n'est normalement pas nécessaire, aussi cette option est-elle réservée aux développeurs du noyau ou de matériel. La réponse recommandée est donc 'N'.

L'option « **Root Port Advanced Error Reporting support** » permet d'activer le support de la gestion des erreurs des périphériques PCI Express au niveau de la racine du bus PCI Express. La réponse recommandée est 'Y'.

L'option « **Message Signaled Interrupts (MSI and MSI-X)** » permet de prendre en charge un nouveau mode de gestion des interruptions matérielles. La réponse recommandée est 'N'.

L'option « **PCI Debugging** » permet d'activer les traces de débogage du sous-système PCI. Elles peuvent être utiles en cas de problème, mais ne sont normalement pas nécessaires. La réponse recommandée est 'N'.

L'option « **Interrupts on hypertransport devices** » permet d'autoriser les périphériques HyperTransport à utiliser les interruptions. La réponse recommandée est 'Y'.

L'option « **ISA support** » permet la prise en charge des bus ISA. Le bus ISA est un bus en voie de désuétude, mais la plupart des anciennes cartes mères en disposent encore. Vous devez activer cette option si votre ordinateur dispose encore d'un bus ISA. La réponse recommandée est 'Y'.

L'option « **EISA support** » permet d'activer la gestion des bus EISA. L'activation de cette fonctionnalité donnera l'accès à des options spécifiques au matériel, ainsi qu'à l'option « **EISA device name database** », qui permet d'obtenir des noms humainement lisibles des périphériques EISA pendant leur configuration, et que l'on activera si l'on dispose de tels périphériques. Ce bus a désormais complètement été remplacé par le bus PCI, aussi devriez-vous répondre par 'N', à moins que vous ne disposiez d'une machine très ancienne.

L'option « **MCA support** » permet d'activer la gestion des bus MCA (pour les PS/2 d'IBM). Si vous activez cette option, vous accéderez à l'option « **Legacy MCA API Support** », qui permet de prendre en charge l'API des anciens noyaux pour les périphériques MCA. Cette option n'est utile que pour charger des gestionnaires de périphériques MCA fournis sous forme de module et qui n'ont pas été portés pour le noyau 2.6. Si vous l'activez, cette option vous donnera accès à l'option « **Support for the mca entry in /proc** », qui vous permettra d'accéder au fichier de gestion du bus MCA /proc/mca. Les bus MCA étant en voie de désuétude, la réponse recommandée est 'N'.

L'option « **NatSemi SCx200 support** » permet de prendre en charge le processeur SCx200 de National Semiconductor. La réponse recommandée est 'N'. Si vous activez cette option, vous pourrez accéder à l'option « **NatSemi SCx200 27MHz High-Resolution Timer Support** », qui permet d'utiliser le timer haute résolution du processeur comme base de temps du système, ce qui est généralement plus fiable lors de la mise en veille du processeur. La réponse recommandée pour cette option est 'Y'.

XII-F-1 - Sous-menu « **PCCARD (PCMCIA/CardBus) support** »

Ce menu contient l'option « **PCCard (PCMCIA/CardBus) support** », qui permet de prendre en charge la gestion des cartes PCMCIA et CardBus. Cette option vous donnera accès aux options de configuration des contrôleurs PCMCIA les plus courants, ainsi qu'à l'option « **Enable PCMCIA debugging** », qui vous permettra d'activer les options de débogage des gestionnaires de périphériques PCMCIA. La réponse recommandée est 'Y' si vous disposez d'un portable, 'N' dans le cas contraire.

XII-F-2 - Sous-menu « PCI Hotplug Support »

Ce menu contient l'option « **Support for PCI Hotplug (EXPERIMENTAL)** », qui permet de prendre en charge les contrôleurs PCI capables de connecter et de déconnecter des cartes PCI à chaud. L'option « **Fake PCI Hotplug driver** », permet de simuler un périphérique PCI Hotplug et n'est utile que pour les développeurs noyau. Les options suivantes permettent de sélectionner les gestionnaires de périphériques pour les contrôleurs PCI en question. La réponse recommandée est 'N'.

XII-G - Menu « Executable file formats / Emulations »

L'option « **Kernel support for ELF binaries** » permet d'utiliser les programmes dont le format de fichier binaire est le format ELF. Ce format de fichier étant devenu un standard, il faut répondre par 'Y' à cette question.

L'option « **Kernel support for a.out and ECOFF binaries** » permet d'utiliser les programmes dont le format de fichier binaire est le format « a.out ». Ce format de fichier binaire est obsolète, et il est peu probable que vous rencontriez de vieilles applications qui n'ont pas été recompilées et qui l'utilisent encore. La réponse recommandée est donc 'N'.

L'option « **Kernel support for MISC binaries** » permet d'activer la gestion de formats de fichiers binaires enregistrables dans le noyau. On peut alors utiliser des chargeurs spécifiques directement au niveau du noyau, et ainsi utiliser ces fichiers binaires directement en ligne de commande. Il est recommandé de répondre par 'Y' à cette question.

L'option « **IA32 Emulation** » permet d'activer la prise en charge des processus 32 bits dans les machines 64 bits de type x86_64. La réponse recommandée est 'Y'.

L'option « **IA32 a.out support** » permet d'activer la prise en charge du format de fichier exécutable 32 bits a.out. Ce format est obsolète et la réponse recommandée est 'N'.

XII-H - Menu « Networking »

L'option « **Networking support** » permet d'activer la prise en charge des fonctionnalités réseau dans le noyau. Étant donné que les systèmes Unix sont profondément orientés réseau, il faut impérativement répondre 'Y' à cette question. Cela vous donnera accès à des options complémentaires permettant de configurer les fonctionnalités réseau prises en charge par le noyau, ainsi qu'au menu de sélection des différents gestionnaires de périphériques réseau.

XII-H-1 - Menu « Networking options »

L'option « **Packet socket** » permet d'autoriser la manipulation directe des trames réseau par des applications clientes. Cette fonctionnalité peut être utile pour certains utilitaires réseau, et en particulier elle est nécessaire pour les clients et les serveurs DHCP. L'option recommandée est donc 'Y'.

L'option « **Packet socket: mmaped IO** » n'est disponible que si l'option précédente a été activée. Elle permet d'utiliser un mécanisme de communication optimisé pour la communication des trames réseau aux applications clientes, basé sur des segments de mémoire partagée (ce qui évite une copie des données transférées). La réponse recommandée est 'Y'.

L'option « **Unix domain sockets** » permet d'activer les communications réseau basées sur la notion de socket Unix. Comme la plupart des programmes Unix utilisent ce paradigme de communication, il faut répondre 'Y' à cette question.

L'option « **Transformation user configuration interface** » permet d'activer l'interface de configuration des extensions de sécurité du protocole IP. Cette interface est utilisée par les outils de configuration d'IPSec, aussi est-il recommandé de répondre 'Y' à cette question.

L'option « Transformation sub policy support » permet de prendre en charge les politiques de sécurité secondaires pour les paquets. Cela permet d'appliquer plus d'une politique de sécurité à un même paquet en même temps, par exemple pour optimiser les traitements des paquets dont les informations ne sont pas conservées longtemps dans le noyau. La réponse recommandée est 'N'.

L'option « Transformation migrate database (EXPERIMENTAL) » permet de prendre en charge l'association entre l'emplacement des clients IPv6 mobiles et les politiques de sécurité. La réponse recommandée est 'N'.

L'option « PF_KEY sockets » permet de prendre en charge les communications réseau sécurisées par l'intermédiaire de la suite logicielle KAME. Cette option constitue une implémentation des extensions de sécurité IPSec du protocole IP. La réponse recommandée est 'Y'.

L'option « PF_KEY MIGRATE (EXPERIMENTAL) » permet de prendre en charge l'association entre l'emplacement des clients IPv6 mobiles et les politiques de sécurité maintenues au travers de l'interface PF_KEY. La réponse recommandée est 'N'.

L'option « TCP/IP networking » permet d'activer le protocole de communication réseau TCP/IP. Le système utilisant intensivement ce protocole de communication, il faut répondre 'Y' à cette question.

L'option « IP: multicasting » permet d'autoriser l'envoi des données à plusieurs ordinateurs en mode multicast (un paquet pour plusieurs destinations). Cette fonctionnalité permet de réduire le trafic réseau dans certaines applications, mais elle est très peu utilisée. La réponse recommandée est donc 'N'.

L'option « IP: advanced router » permet de configurer le système pour être un routeur (ordinateur qui transfère des informations d'un réseau à un autre). La réponse recommandée est 'N'.

L'option « Choose IP: FIB lookup algorithm (choose FIB_HASH if unsure) » permet de sélectionner un algorithme de recherche dans les tables de routage. L'algorithme FIB_HASH convient pour la majorité des utilisateurs, aussi est-il recommandé de le choisir.

L'option « IP: policy routing » permet d'activer le routage des paquets en fonction des adresses source en plus des adresses destination des paquets. La réponse recommandée est 'N'.

L'option « IP: equal cost multipath » permet de choisir une route possible parmi plusieurs routes pour chaque paquet transmis. Cette option vous donnera accès à l'option « IP: equal cost multipath with caching support (EXPERIMENTAL) », qui permet de mémoriser les routes dans des caches du système. Les algorithmes de sélection des routes multiples peuvent être choisis avec les sous-options suivantes. La réponse recommandée est 'N' pour ces deux options.

L'option « IP: verbose route monitoring » permet d'activer les traces du sous-système de routage. La réponse recommandée est 'N'.

L'option « IP: kernel level autoconfiguration » permet de réaliser la configuration du protocole réseau IP au niveau du noyau, lors de la phase de démarrage. Cette option est utilisée notamment lorsqu'on désire monter le système de fichiers racine par NFS. La réponse recommandée est 'N'.

L'option « IP: DHCP support » permet de demander au noyau de déterminer automatiquement l'adresse IP lors du démarrage grâce au protocole « DHCP ». Cette option n'est valide que lorsque l'option « IP: kernel level autoconfiguration » a été activée. La réponse recommandée est 'N'.

L'option « IP: BOOTP support » permet de demander au noyau de déterminer automatiquement l'adresse IP lors du démarrage grâce au protocole « BOOTP ». Cette option n'est valide que lorsque l'option « IP: kernel level autoconfiguration » a été activée. La réponse recommandée est 'N'.

L'option « IP: RARP support » permet de demander au noyau de déterminer automatiquement l'adresse IP lors du démarrage grâce au protocole « RARP ». Ce protocole est un protocole plus ancien que le protocole BOOTP, il est en passe de devenir obsolète. Cette option n'est valide que lorsque l'option « IP: kernel level configuration » a été activée. La réponse recommandée est 'N'.

L'option « IP: tunneling » permet d'activer l'encapsulation des paquets d'un protocole dans les paquets d'un autre protocole. La réponse recommandée est 'N'.

L'option « IP: GRE tunnels over IP » permet d'autoriser l'encapsulation des protocoles IPv4 et IPv6 avec la méthode « GRE » (abréviation de l'anglais « Generic Routing Encapsulation »). Cette méthode d'encapsulation est destinée aux routeurs Cisco. La réponse recommandée est 'N'.

L'option « IP: broadcast GRE over IP » permet de créer un réseau Ethernet virtuel sur IP par l'intermédiaire de la méthode d'encapsulation GRE, qui permet d'effectuer des broadcasts d'IP dans le réseau virtuel. La réponse recommandée est 'N'.

L'option « IP: multicast routing » permet de configurer le système pour le routage des paquets ayant plusieurs destinations (c'est-à-dire les paquets IP envoyés en multicast). La réponse recommandée est 'N'.

L'option « IP: PIM-SM version 1 support » permet d'activer la gestion du protocole de routage « PIM » des paquets envoyés en multicast. Ce protocole est géré par Cisco. La réponse recommandée est 'N'.

L'option « IP: PIM-SM version 2 support » permet d'activer la gestion de la version 2 du protocole de routage PIM. La réponse recommandée est 'N'.

L'option « IP: ARP daemon support EXPERIMENTAL » permet de limiter à 256 la table d'adresses physiques utilisées pour les requêtes ARP. Cette option est utile pour limiter la consommation mémoire du noyau dans les grands réseaux. Les requêtes ne pouvant être satisfaites directement sont transférées à un démon, repoussant ainsi le reste de la table hors de la mémoire du noyau. La réponse recommandée est 'N'.

L'option « IP: TCP syncookie support (disabled per default) » permet de protéger la machine d'une certaine forme d'attaque du protocole réseau TCP/IP visant à provoquer un déni de service. Le fait de répondre par 'Y' à cette question inclut le support de cette protection, mais ne l'active pas par défaut. L'activation doit se faire par configuration dynamique du noyau via `/proc/`. La réponse recommandée est 'N'.

L'option « IP: AH transformation » permet de prendre en charge l'extension AH pour le protocole IPv4. Cette extension permet de s'assurer que les en-têtes IP sont authentiques, c'est-à-dire que la machine qui prétend les avoir envoyés est bien la machine qui les a effectivement envoyés. Cette option nécessite les fonctionnalités de cryptographie du menu « Cryptographic options ». La réponse recommandée est 'Y'.

L'option « IP: ESP transformation » permet de prendre en charge l'extension ESP pour le protocole IPv4. Cette extension permet de s'assurer de la confidentialité des données transférées sur le réseau, en chiffrant les données transmises. Cette option va de pair avec l'option « IP: AH transformation » et ne sert à rien sans elle (communiquer de manière chiffrée avec un pirate au lieu du bon destinataire ne dérange pas vraiment celui-ci). Cette option nécessite les fonctionnalités de cryptographie du menu « Cryptographic options ». La réponse recommandée est 'Y'.

L'option « IP: IPComp transformation » permet de prendre en charge l'extension de compression de données du protocole IPv4. Cette option permet de compresser les données avant chiffrement, dans le cas où l'on utiliserait également l'extension ESP. En effet, les données chiffrées apparaissent généralement comme des données parfaitement aléatoires et ne peuvent donc pas être compressées, il est donc nécessaire d'effectuer cette compression avant chiffrement, ce que fait cette extension. La réponse recommandée est 'Y'.

L'option « IP: IPsec transport mode » permet d'activer le protocole IPsec en mode transport (c'est-à-dire le mode de transmission natif des paquets IPsec). La réponse recommandée est 'Y'.

L'option « IP: IPsec tunnel mode » permet d'activer le protocole IPsec en mode tunnel (c'est-à-dire le mode de transmission des paquets IPsec basé sur leur encapsulation dans le protocole IP). La réponse recommandée est 'Y'.

L'option « IP: IPsec BEET mode » permet d'activer le protocole IPsec en mode BEET (« Bound End-to-End Tunnel »). Ce mode de transmission est utilisé pour réaliser un tunnel point à point, sans être pénalisé par l'encapsulation des paquets. Les tunnels de ce type sont moins génériques que ceux que l'on obtient en mode tunnel, mais conviennent dans le cas de connexions point à point directes. La réponse recommandée est 'Y'.

L'option « INET: socket monitoring interface » permet d'activer la prise en charge de l'interface de surveillance des sockets, qui est utilisée par certains outils réseau. La réponse recommandée est 'Y'.

L'option « TCP: advanced congestion protocol » permet d'activer la prise en charge de modules de gestion des congestions réseau pour le protocole réseau. Cette option vous donnera accès aux différents modules des algorithmes de gestion des congestions réseau. L'algorithme utilisé par défaut convient généralement à la plupart des utilisateurs, aussi la réponse recommandée est-elle 'N'.

L'option « TCP: MD5 Signature Option support (RFC2385) » permet d'activer la prise en charge des signatures MD5 des données transmises par TCP pour en assurer l'authenticité. Cette fonctionnalité n'est utilisée que par les routeurs sur Internet pour échanger leurs informations. La réponse recommandée est donc 'N'.

L'option « The IPv6 protocol (EXPERIMENTAL) » permet d'activer la gestion du protocole IPv6. Ce protocole est appelé à remplacer la version actuelle du protocole IP, afin d'accroître le nombre d'adresses disponibles, d'améliorer la qualité de service et d'intégrer les fonctionnalités d'authentification et de confidentialité des données. Cette option vous donnera accès aux options de configuration complémentaires du protocole. La réponse recommandée est 'N'.

L'option « IPv6: Privacy Extensions (RFC 3041) support » permet de demander à Linux de changer régulièrement et automatiquement l'adresse IP de votre interface, afin de garantir votre anonymat sur le réseau. La réponse recommandée est 'N'.

Les options « IPv6: Router Preference (RFC 4191) support » et « IPv6: Route Information (RFC 4191) support » permettent d'activer la gestion d'une extension du protocole permettant aux machines de choisir le routeur le plus approprié lorsqu'elles sont placées dans un réseau disposant de plusieurs passerelles. La réponse recommandée est 'N'.

L'option « IPv6: Enable RFC 4429 Optimistic DAD (EXPERIMENTAL) », active l'algorithme de détection de conflits d'adresses IPv6 décrit dans la RFC 4429, qui permet une auto configuration plus rapide des interfaces IPv6. La réponse recommandée est 'N'.

Les options « IPv6: AH transformation », « IPv6: ESP transformation » et « IPv6: IPComp transformation » permettent de prendre en charge les extensions d'authentification, de confidentialité et de compression de données décrites ci-dessus pour le protocole IPv6. La réponse recommandée est 'Y'.

L'option « IPv6: Mobility » permet de prendre en charge la fonctionnalité de suivi des appareils mobiles dans un réseau IPv6. Grâce à cette option, un appareil mobile conserve son adresse IPv6, mais se voit attribuer une adresse d'attache qui indique l'hôte le plus proche par lequel on peut le joindre. La transmission de ces informations se fait via un protocole réseau complémentaire à IPv6. La réponse recommandée est 'Y' si vous utilisez le protocole IPv6 sur une station mobile, 'N' sinon.

Les options « IPv6: IPsec transport mode », « IPv6: IPsec tunnel mode » et « IPv6: IPsec BEET mode » sont équivalentes aux options correspondantes du protocole IPv4 et permettent de définir les modes de transport des paquets IPv6 pris en charge par le noyau. La réponse recommandée pour toutes ces options est 'Y'.

L'option « IPv6: MIPv6 route optimization mode » permet d'activer la fonctionnalité d'optimisation du routage pour les périphériques IPv6 mobiles. La réponse recommandée est 'N'.

Les options « IPv6: IPv6-in-IPv4 tunnel (SIT driver) » permettent d'activer les fonctionnalités d'encapsulation des paquets IPv6 dans un flux IPv4. Cette option permet donc de réaliser un tunnel entre deux réseaux IPv6 via un réseau qui ne gère que le protocole IPv4. La réponse recommandée est 'Y'.

Les options « IPv6: IPv6-in-IPv6 tunnel » permettent d'activer les fonctionnalités d'encapsulation des paquets IPv6 dans un flux lui-même IPv6. La réponse recommandée est 'Y'.

Les options « IPv6: Multiple Routing Tables » et « IPv6: source address based routing » permettent de prendre en charge les tables de routage multiples pour le protocole IPv6 et de gérer les conflits de route en se basant également sur l'adresse source et non uniquement sur l'adresse destination lors de la détermination de la route. La réponse recommandée est 'N' pour ces deux options.

L'option « NetLabel subsystem support » permet d'activer la prise en charge des protocoles de marquage des paquets tels que les protocoles CIPSO et RIPS0 afin de les classer et de leur appliquer une politique de sécurité. La réponse recommandée est 'N'.

L'option « Security Marking » permet d'activer une fonctionnalité de marquage des paquets afin de leur appliquer des contrôles de sécurité par la suite. La réponse recommandée est 'N'.

L'option « Network packet filtering framework (Netfilter) » permet d'activer les fonctions de filtrage des paquets réseau du noyau. Ces fonctions peuvent être utilisées pour plusieurs raisons, les plus courantes étant sans doute la réalisation de pare-feu et de partages de connexion à Internet. Pour information, un pare-feu est un programme qui filtre les informations en provenance et à destination du réseau, selon des règles de sécurité prédéfinies. Ces règles permettent d'effectuer le filtrage en fonction des adresses et du type des paquets émis et reçus. Les actions qui peuvent être prises sur les paquets ainsi filtrés peuvent être variées, allant de l'élimination du paquet à sa modification ou son transfert vers une autre adresse que celle vers laquelle il devait aller initialement. Cette fonctionnalité permet donc également de réaliser la translation d'adresse des paquets TCP/IP. Vous devez activer cette option si vous désirez réaliser un partage de connexion à Internet ou un pare-feu. Le détail de la configuration du sous-menu rendu accessible par cette option est décrit dans une section dédiée plus loin dans ce document. La réponse recommandée est 'Y'.

L'option « Asynchronous Transfer Mode (ATM) (EXPERIMENTAL) » permet d'activer la gestion des réseaux ATM. ATM est un type de réseau travaillant en mode connecté (c'est-à-dire qu'une connexion permanente est établie entre les deux machines), ce qui permet d'effectuer une négociation initiale des ressources à allouer à cette connexion. Les réseaux ATM sont donc relativement adaptés aux transferts de données temps réel, comme la voix ou la vidéo. Les paquets transférés sont tous de taille fixe, ce qui permet de simplifier leur traitement et d'obtenir des débits très grands. ATM est utilisé aussi bien pour les réseaux de grande échelle que pour les réseaux locaux. L'option « Classical IP over ATM » permet d'activer le support du protocole IP encapsulé dans un réseau virtuel ATM. Une alternative à cette option est l'option « LAN Emulation (LANE) support ». L'option « Do NOT send ICMP if no neighbour » permet d'éviter l'envoi de paquet ICMP signalant l'inaccessibilité d'une machine lorsque le noyau supprime temporairement la connexion à cette machine de ses tables internes pendant certaines opérations de maintenance. L'option « LAN Emulation (LANE) support » permet de simuler un réseau local classique sur un réseau ATM et éventuellement d'établir un pont entre ce réseau local virtuel et d'autres réseaux Ethernet réels. L'option « MultiProtocol Over ATM (MPOA) support » permet l'établissement de canaux virtuels ATM au travers des limites des réseaux afin d'optimiser le routage des données. L'option « RFC1483/2684 Bridged protocols » permet d'envoyer des paquets Ethernet sur un réseau virtuel ATM. Enfin, l'option « Per-VC IP filter kludge » permet de fermer un grand nombre de canaux virtuels plus efficacement lorsqu'ils sont utilisés uniquement pour des communications IP. La réponse recommandée à toutes ces questions est 'N'.

L'option « 802.1d Ethernet Bridging » permet de configurer le système comme un pont Ethernet (un pont permet de regrouper physiquement plusieurs réseaux en un seul réseau physique). La réponse recommandée est 'N'.

L'option « 802.1Q VLAN Support (EXPERIMENTAL) » permet de créer des interfaces réseau virtuelles 802.1Q sur une interface Ethernet normale. La réponse recommandée est 'N'.

L'option « DECnet support » permet d'activer la gestion des réseaux DECnet, initialement créés par Digital (et repris maintenant par Compaq). Cette option vous donnera accès aux sous-options de configuration des réseaux DECnet. L'option « DECnet: router support (EXPERIMENTAL) » permet d'activer les fonctionnalités de routage sur les réseaux DECnet. La réponse recommandée pour ces options est 'N'.

L'option « ANSI/IEEE 802.2 LLC type 2 Support » permet de prendre en charge un protocole d'encapsulation nommé LLC dans un autre protocole. La réponse recommandée est 'N'.

L'option « The IPX protocol » permet de prendre en charge le protocole réseau « IPX » de Novell. Cette option vous donnera accès à la sous-option « IPX: Full internal IPX network », qui permet de configurer le serveur pour qu'il apparaisse comme un réseau IPX à part entière, en lui assignant un numéro de réseau Novell. Toutes les requêtes seront alors redirigées vers des nœuds gérés en interne pour ce réseau virtuel. La réponse recommandée est 'N' pour ces deux options.

L'option « Appletalk protocol support » permet d'activer la gestion des réseaux Appletalk. Elle vous permettra d'accéder à des sous-options de configuration du protocole réseau Appletalk. L'option « Appletalk interfaces support » vous permettra d'accéder aux options de configuration des cartes Appletalk. La sous-option « AppleTalk-IP driver support » permet de se connecter à un réseau utilisant le protocole IP via un adaptateur réseau AppleTalk. Cette option vous permettra de réaliser des encapsulations de paquets IP dans les paquets Appletalk (sous-option « IP to Appletalk-IP Encapsulation support »), pour les machines Linux qui n'ont accès qu'à un réseau Appletalk, et inversement d'encapsuler les paquets Appletalk dans les paquets IP (sous-option « Appletalk-IP to IP Decapsulation support »), pour les machines Linux qui font office de passerelles vers Internet pour les machines d'un réseau Appletalk. La réponse recommandée pour toutes ces options est 'N'.

L'option « CCITT X.25 Packet Layer (EXPERIMENTAL) » permet d'activer la gestion du protocole de bas niveau pour X.25. La réponse recommandée est 'N'.

L'option « LAPB Data Link Driver (EXPERIMENTAL) » permet d'activer la gestion du protocole de communication de haut niveau de X.25. La réponse recommandée est 'N'.

L'option « Acorn Econet/AUN protocols (EXPERIMENTAL) » permet d'activer la gestion des réseaux Econet. Cette option vous donnera accès à la sous-option « AUN over UDP », qui permet d'encapsuler les paquets Econet dans UDP, et à la sous-option « Native Econet », qui permet de prendre en charge les cartes réseau Econet. La réponse recommandée est 'N' pour toutes ces questions.

L'option « WAN router » permet d'effectuer le routage sur un réseau « WAN » (abréviation de l'anglais « Wide Area Network »). La réponse recommandée est 'N'.

XII-H-1-a - Sous-menu « IP: Virtual Server Configuration »

L'option « IP virtual server support (EXPERIMENTAL) » permet de réaliser des serveurs virtuels à répartition de charge. Ces serveurs ont pour but de présenter une adresse IP unique aux clients, et de répartir les requêtes de ceux-ci équitablement sur plusieurs machines de traitement. Les options qui suivent permettent de configurer le gestionnaire de serveur virtuel. La réponse recommandée est 'N'.

L'option « IP virtual server debugging » permet d'obtenir des traces complémentaires concernant le serveur virtuel. La réponse recommandée est 'N'.

L'option « IPVS connection table size (the Nth power of 2) » permet de définir la taille (en termes de l'exposant d'une puissance de deux) de la table de répartition des requêtes au sein du serveur virtuel. La réponse recommandée est '12'.

Les options qui suivent permettent de réaliser la répartition de charge dynamique pour les différents protocoles supportés, ainsi que les options permettant de définir les algorithmes de répartition de charge eux-mêmes.

L'option « **FTP protocol helper** » permet de prendre en charge le protocole FTP, pour lequel un traitement spécifique doit être réalisé lors de la redirection vers les serveurs physiques. En effet, le protocole FTP stocke des adresses IP au sein de ses paquets, et celles-ci doivent également être transformées à la volée. La réponse recommandée est 'N'.

XII-H-1-b - Sous-menu « Network packet filtering (replace ipchains) »

L'option « **Network packet filtering debugging** » active la gestion des messages de débogage des fonctionnalités de filtrage des paquets. Ces messages peuvent être très utiles pour le débogage des règles de filtrage et de translations d'adresses du noyau, aussi la réponse recommandée est-elle 'Y'.

L'option « **Bridged IP/ARP packets filtering** » permet de prendre en charge le filtrage des paquets ARP et IP lorsque la machine est configurée pour faire office de pont (c'est-à-dire que l'option « **802.1d Ethernet Bridging** » doit avoir été activée dans le menu supérieur. Cette option est utile pour réaliser un pont filtrant. La réponse recommandée est 'N'.

XII-H-1-b-i - Sous-menu « Core Netfilter Configuration »

L'option « **Netfilter netlink interface** » permet d'activer la mise à disposition des processus utilisateurs des paquets filtrés par le noyau via l'interface Netlink. La réponse recommandée est 'Y'. Les options qui suivent permettent de spécifier les fonctionnalités fournies par cette nouvelle interface.

L'option « **Netfilter NFQUEUE over NFNETLINK interface** » active la prise en charge de la file NFNETLINK dans l'interface NetLink pour permettre aux processus utilisateurs d'accéder aux paquets manipulés par Netfilter. La réponse recommandée est 'Y'.

L'option « **Netfilter LOG over NFNETLINK interface** » active le transfert des messages de traces relatifs aux paquets filtrés par Netfilter au travers de la queue NFNETLINK de l'interface NetLink. La réponse recommandée est 'Y'.

L'option « **Netfilter connection tracking support** » active la prise en charge des fonctionnalités de suivi de connexions du module de filtrage Netfilter du noyau. Ces fonctionnalités sont nécessaires pour permettre le filtrage correct des connexions des divers protocoles au niveau du pare-feu, ainsi que pour permettre les parages de connexion à Internet. La réponse recommandée est donc 'Y'.

L'option « **Connection tracking flow accounting** » permet d'activer la prise en charge des statistiques des partages de connexion pour chaque flux réseau. Ces statistiques peuvent ensuite être utilisées pour définir des quotas dans iptables via le critère de sélection connbytes. La réponse recommandée est 'N'.

L'option « **Connection mark tracking support** » permet d'activer le critère de sélection connmark et la cible CONNMARK d'iptables. Cette fonctionnalité permet d'identifier tous les paquets d'une même connexion. La réponse recommandée est 'N'.

L'option « **Connection tracking security mark support** » permet d'activer le critère de sélection basé sur les marqueurs de sécurité définis via la cible SECMARK d'iptables. La réponse recommandée est 'N'.

L'option « **Connection tracking events** » permet de prendre en charge les mécanismes de notification des changements d'état des connexions. Ces notifications peuvent être utilisées par d'autres parties du noyau. La réponse recommandée est 'N'.

L'option « **SCTP protocol connection tracking support (EXPERIMENTAL)** » active la gestion du suivi des connexions SCTP utilisé pour la diffusion des flux multimédias sur Internet. Ces connexions nécessitent en effet un traitement particulier, et vous devrez activer cette option si vous voulez utiliser des connexions SCTP avec un partage de connexion à Internet. La réponse recommandée est 'N'.

L'option « **UDP-Lite protocol connection tracking support (EXPERIMENTAL)** » active la gestion du suivi des connexions UDP-Lite. La réponse recommandée est 'N'.

L'option « **Amanda backup protocol support** » active la gestion des connexions utilisant le protocole réseau du logiciel de sauvegarde Amanda. La réponse recommandée est 'N', sauf si bien sûr vous utilisez ce logiciel.

L'option « **FTP protocol support** » active la gestion du suivi des connexions FTP. Ces connexions nécessitent en effet un traitement particulier, et vous devrez activer cette option si vous voulez utiliser des connexions FTP avec un partage de connexion à Internet. La réponse recommandée est 'Y'.

L'option « **H.323 protocol support** » active la gestion du suivi des connexions du protocole H.323. H.323 est un protocole de contrôle servant au transport de la voix sur IP, en particulier par les téléphones Internet. Ce module prend également en charge les protocoles RAS, H.245 en tunnel, RTP/RTCP et les protocoles basés sur T.120 pour la vidéo, l'audio, le fax, etc. La réponse recommandée est 'Y'.

L'option « **IRC protocol support** » active la prise en charge des commandes de transfert de fichiers DCC du protocole IRC. Si vous utilisez IRC couramment, il est expressément recommandé de répondre 'Y' à cette question, faute de quoi vous ne pourrez pas échanger de fichiers avec vos interlocuteurs.

L'option « **NetBIOS name service protocol support (EXPERIMENTAL)** » permet de prendre en charge le filtrage des annonces de noms NetBIOS sur le réseau. La réponse recommandée est 'N'.

L'option « **PPtP protocol support** » active la gestion du suivi des connexions PPTP. La gestion de ce protocole par ce module n'est toutefois pas encore tout à fait complète. La réponse recommandée est donc 'N'.

L'option « **SANE protocol support** » active la gestion du suivi des connexions aux scanners gérés par SANE. La réponse recommandée est 'N'.

L'option « **SIP protocol support (EXPERIMENTAL)** » active la gestion du suivi des connexions du protocole SIP de téléphone sur Internet. La réponse recommandée est 'Y'.

L'option « **TFTP protocol support** » active la gestion du suivi des connexions TFTP. Si vous utilisez des clients TFTP au travers de votre partage de connexion à Internet, vous devez activer cette option. La réponse recommandée est 'Y'.

L'option « **Connection tracking netlink interface** » permet d'activer l'interface utilisateur Netlink pour le suivi de connexion. La réponse recommandée est 'N'.

L'option « **Netfilter Xtables support (required for ip_tables)** » active la gestion des tables de filtrage de Netfilter. Ces tables constituent un des composants essentiels de Netfilter, aussi la réponse recommandée est-elle 'Y'.

L'option « **"CLASSIFY" target support** » active la gestion de la cible CLASSIFY, qui permet de marquer les paquets avec une priorité utilisable avec les outils de gestion de trafic. La réponse recommandée est 'N'.

L'option « **"CONNMARK" target support** » active la gestion de la cible CONNMARK, qui permet de marquer les paquets d'une même connexion. La réponse recommandée est 'N'.

L'option « **"DSCP" target support** » active la gestion de la cible DSCP, qui permet de manipuler le champ DSCP des en-têtes IPv4 et IPv6. La réponse recommandée est 'N'.

L'option « **"MARK" target support** » active la gestion de la cible MARK, qui permet de marquer les paquets avec un traceur afin de pouvoir les identifier ultérieurement. Par exemple, il est possible de modifier le routage de certains paquets selon qu'ils sont marqués ou non. On peut ainsi réaliser des liaisons prioritaires pour certaines catégories de paquets. La réponse recommandée est 'N'.

L'option « **"NFQUEUE" Target Support** » active les fonctionnalités de transfert des paquets vers l'espace utilisateur par la nouvelle interface Netlink du noyau. La réponse recommandée est 'Y'.

L'option « "NFLOG" target support » active la gestion de la cible NLOG. Cette cible permet d'enregistrer des messages de débogage dans les fichiers de traces du système via une interface générique du noyau, qui elle-même pourra s'appuyer sur les cibles classiques LOG, ULOG ou nfnetlink_log. La réponse recommandée est 'Y'.

L'option « "NOTRACK" target support » permet de marquer les paquets comme ne devant pas être pris en compte par les mécanismes de suivi de connexion. La réponse recommandée est 'N'.

L'option « "TRACE" target support » permet de marquer les paquets pour que le noyau effectue une trace à chaque fois qu'il vérifie une des règles de Netfilter. La réponse recommandée est 'N'.

L'option « "SECMARK" target support » permet de marquer les paquets pour des vérifications de sécurité ultérieures. La réponse recommandée est 'N'.

L'option « "CONNSECMARK" target support » permet de marquer les paquets avec les attributs de sécurité de leur connexion et réciproquement. La réponse recommandée est 'N'.

L'option « "TCPMSS" target support » permet de marquer les paquets pour modifier le champ MSS définissant la taille maximum du paquet sur le lien physique. Cette option n'est normalement pas nécessaire, sauf si le protocole ICMP est filtré par un routeur bogué ou un fournisseur d'accès non respectueux des principes de fonctionnement fondamentaux d'Internet. La réponse recommandée est donc 'N'.

L'option « "comment" match support » permet de prendre en charge un mode de sélection virtuel des paquets, qui permet d'ajouter des commentaires dans les jeux de règles de filtrage du noyau. La réponse recommandée est 'N'.

L'option « "connbytes" per-connection counter match support » permet de sélectionner les paquets selon leur taille en octet ou leur nombre. La réponse recommandée est 'N'.

L'option « "connlimit" match support » permet de sélectionner les paquets en fonction du nombre de connexions maintenues en parallèle par l'adresse IP du client qui les a émis. La réponse recommandée est 'N'.

L'option « "connmark" connection mark match support » permet de sélectionner les paquets selon le numéro de la connexion à laquelle ils appartiennent et qui leur est attribué par la fonctionnalité de suivi des connexions. La réponse recommandée est 'N'.

L'option « "conntrack" connection tracking match support » permet de sélectionner les paquets selon les connexions réseau auxquels ils appartiennent. Par exemple, cette option permet de distinguer les paquets de différentes connexions virtuelles dans le cas du tunneling. La réponse recommandée est 'Y'.

L'option « "DCCP" protocol match support » permet de sélectionner les paquets qui utilisent le protocole DCCP. La réponse recommandée est 'N'.

L'option « "DSCP" match support » permet de sélectionner les paquets selon la valeur du champ DSCP de leur en-tête IPv4 ou IPv6. La réponse recommandée est 'N'.

L'option « "ESP" match support » permet de sélectionner les paquets qui utilisent le protocole ESP d'IPSec. La réponse recommandée est 'N'.

L'option « "helper" match support » active la gestion du critère de sélection des paquets participant à une connexion établie et suivie par l'une des fonctionnalités activées par l'option « Connection tracking (required for masq/NAT) ». La réponse recommandée est 'Y'.

L'option « "length" match support » active la gestion du critère de sélection basé sur la longueur des paquets. La réponse recommandée est 'N'.

L'option « `"limit" match support` » active la gestion de la limitation du nombre de fois par seconde qu'une règle peut être vérifiée par un paquet. Cette limitation est utile lorsqu'on enregistre des messages pour tous les paquets qui vérifient certaines règles, afin d'éviter l'engorgement des fichiers de traces du système. La réponse recommandée est 'Y'.

L'option « `"mac" address match support` » active la gestion du critère de sélection des paquets basé sur leur adresse Ethernet source. Cette règle n'est utilisable que pour les paquets provenant d'une interface réseau de type Ethernet. La réponse recommandée est 'N'.

L'option « `"mark" match support` » active la gestion du critère de sélection des paquets basé sur le champ MARK de leur en-tête. Ce champ peut être modifié par certaines règles des chaînes précédemment traversées par les paquets, afin de les marquer pour un traitement ultérieur. Cette option doit donc obligatoirement être activée si l'on désire détecter ces paquets pour effectuer ce traitement. La réponse recommandée est 'N'.

L'option « `IPsec "policy" match support` » active la gestion du critère de sélection des paquets basé sur la politique de sécurité du protocole IPsec. La réponse recommandée est 'N'.

L'option « `Multiple port match support` » permet d'utiliser des plages de valeurs pour les ports TCP et UDP dans les critères de sélection des règles pour ces deux protocoles. Sans cette option, les ports doivent être spécifiés un à un, ce qui peut rendre relativement peu pratique la définition de certaines règles. La réponse recommandée est 'Y'.

L'option « `"physdev" match support` » permet de sélectionner les paquets en fonction du port physique sur lequel ils arrivent ou doivent repartir dans le cas d'une configuration de la machine en tant que pont. Cette option n'est disponible que si la machine a été configurée pour faire office de pont entre deux réseaux. La réponse recommandée est 'N'.

L'option « `"pkttype" packet type match support` » vous permet de sélectionner les paquets en fonction de leur nature (broadcast, multicast, etc...). La réponse recommandée est 'N'.

L'option « `"quota" match support` » vous permet de sélectionner les paquets en fonction d'un quota de données transférées, calculé grâce à un compteur d'octet.. La réponse recommandée est 'N'.

L'option « `"realm" match support` » permet de sélectionner les paquets en fonction de la clef `realm` du code de routage du noyau. Cette option est très spécifique, aussi la réponse recommandée est-elle 'N'.

L'option « `"sctp" protocol match support` » permet de sélectionner les paquets qui utilisent le protocole SCTP. La réponse recommandée est 'N'.

L'option « `"state" match support` » permet de sélectionner les paquets selon leur rôle dans la gestion des connexions réseau. Par exemple, cette option permet de distinguer les paquets qui établissent une connexion réseau des autres paquets. La réponse recommandée est 'Y'.

L'option « `"statistic" match support` » permet de sélectionner les paquets périodiquement ou selon une propriété statistique de leur contenu. La réponse recommandée est 'N'.

L'option « `"string" match support` » permet de sélectionner les paquets en fonction d'un motif textuel prédéfini. La réponse recommandée est 'N'.

L'option « `"tcpmss" match support` » permet de prendre en compte le champ MSS des paquets de demandes de connexion dans les critères de sélection. Ce champ indique la taille maximale que les paquets de cette connexion devront utiliser pas la suite. La réponse recommandée est 'N'.

L'option « `"u32" match support` » permet d'extraire n'importe quelle valeur de 32 bits d'un paquet réseau pour établir les critères de sélection. La réponse recommandée est 'N'.

L'option « "hashlimit" match support » permet de sélectionner les paquets selon des limites statistiques calculées à l'aide d'une table de hachage sur leurs adresses. Les limites définies avec cette option peuvent être définies de manière plus souple que les limites définies avec la fonction de limitation classique. La réponse recommandée est 'Y'.

XII-H-1-b-ii - Sous-menu « IP: Netfilter Configuration »

L'option « IPv4 connection tracking support (required for NAT) » permet de prendre en charge le protocole IPv4 au niveau du gestionnaire de suivi de connexions. La réponse recommandée est 'Y'.

L'option « proc/sysctl compatibility with old connection tracking » permet de prendre en charge une ancienne interface de configuration du suivi de connexion qui était dépendante de la couche de niveau 3 dans la pile réseau. La réponse recommandée est 'Y'.

L'option « Userspace queueing via NETLINK (OBSOLETE) » permet de mettre à disposition de programmes clients les paquets traités par le code de filtrage, par l'intermédiaire de l'ancienne interface Netlink. Cette interface étant obsolète à présent, la réponse recommandée est 'N'.

L'option « IP tables support (required for filtering/masq/NAT) » permet d'activer la gestion des tables au sein du code de filtrage du noyau. Une table est en réalité un ensemble cohérent de fonctionnalités permettant d'appliquer des traitements aux paquets selon des règles organisées en groupes. Ces traitements peuvent intervenir à différents endroits dans la gestion des paquets par le code réseau du noyau. Les deux tables les plus importantes sont celles qui permettent de réaliser le filtrage des paquets et les translations d'adresses. Vous devez donc activer cette fonctionnalité si vous désirez réaliser un Firewall ou un partage de connexion à Internet. La réponse recommandée est 'Y'.

L'option « IP range match support » permet de prendre en charge les règles de sélection de paquets définies sur des plages d'adresses IP. La réponse recommandée est 'Y'.

L'option « TOS match support » active la gestion du critère de sélection des paquets basé sur le champ TOS de leur en-tête. Ce champ permet de définir le type de service des paquets, principalement afin de distinguer les paquets prioritaires des paquets normaux. La réponse recommandée est 'Y'.

L'option « recent match support » permet un critère de sélection basé sur des listes d'adresses IP avec lesquelles des communications se sont faites récemment. La réponse recommandée est 'N'.

L'option « ECN match support » active la gestion du critère de sélection des paquets basé sur le champ ECN de leur en-tête. La réponse recommandée est 'N'.

L'option « AH match support » active la gestion des critères de sélection basés sur les informations utilisées par le protocole d'authentification AH de l'extension IPSec du protocole IP. La réponse recommandée est 'N'.

L'option « TTL match support » permet d'utiliser le champ TTL des paquets comme critère de sélection. Ce champ permet de définir la durée de vie des paquets au travers des différents routeurs, et permet de limiter la propagation des paquets sur Internet à un certain nombre d'interconnexions seulement. La réponse recommandée est 'N'.

L'option « Owner match support (EXPERIMENTAL) » permet de sélectionner les paquets créés par les processus locaux en utilisant comme critère les identifiants de groupe, de processus et d'utilisateur de celui qui les a créés. Ces critères de sélection ne sont pas utilisables sur les paquets provenant de l'extérieur. La réponse recommandée est 'N'.

L'option « address type match support » permet de sélectionner les paquets en fonction de ce que le code de routage du noyau pense de la nature d'une adresse (diffusion, point à point, etc.). Cette option est très spécifique, aussi la réponse recommandée est-elle 'N'.

L'option « Packet filtering » active la gestion de la table filter, couramment utilisée pour filtrer les paquets autorisés et réaliser un Firewall. La réponse recommandée est 'Y'.

L'option « **REJECT target support** » active la gestion de la cible **REJECT** dans les règles des chaînes de la table **filter**. Cette cible se distingue de la cible **DROP**, gérée nativement par la table **filter**, par le fait qu'un message d'erreur est renvoyé à la machine source du paquet. La réponse recommandée est 'Y'.

L'option « **LOG target support** » active la gestion de la cible **LOG**. Cette cible permet d'enregistrer des messages de débogage dans les fichiers de traces du système. On veillera à activer également la gestion des limites sur les règles de filtrage afin d'éviter d'engorger les fichiers de traces si l'on désire utiliser cette option. La réponse recommandée est 'Y'.

L'option « **ULOG target support (OBSOLETE)** » active la gestion de la cible **ULOG**. Cette cible permet d'enregistrer des messages de débogage au niveau d'un processus de gestion de traces fonctionnant dans l'espace utilisateur. Cette fonctionnalité est obsolète et a été remplacée par la nouvelle interface **Netlink** du noyau, aussi la réponse recommandée est-elle 'N'.

L'option « **Full NAT** » active les fonctionnalités de translation d'adresses, au travers de la table **nat**. Cette option doit être activée si vous désirez réaliser un partage de connexion à Internet. Dans le cas contraire, la réponse recommandée est 'Y'.

L'option « **MASQUERADE target support** » active la gestion du masquering des paquets réseau. Vous devez activer cette option si vous désirez réaliser un partage de connexion à Internet. La réponse recommandée est 'Y'.

L'option « **REDIRECT target support** » active la gestion de la translation d'adresses destinations pour rediriger les paquets vers la machine locale. Cette option est très utilisée pour réaliser des proxies qui devront fonctionner de manière transparente pour les clients. La réponse recommandée est 'N'.

L'option « **NETMAP target support** » permet d'activer le support des translations d'adresses réseau. Cela permet de transférer tout le trafic dédié à un réseau vers un autre réseau disposant de la même topologie. La réponse recommandée est 'N'.

L'option « **SAME target support** » permet d'activer le support des translations d'adresses sources, avec un jeu d'adresses sources visibles des clients fixe. La réponse recommandée est 'N'.

L'option « **Basic SNMP-ALG support (EXPERIMENTAL)** » active la gestion de la translation d'adresses pour permettre l'accès et le contrôle à des machines situées sur le réseau et pour lesquelles un partage de connexion à Internet est réalisé. À moins que vous n'utilisiez le protocole **SNMP** pour administrer les machines de votre réseau, vous pouvez répondre 'N' à cette question.

L'option « **Packet mangling** » active la gestion de la table **mangle**. Cette table a pour but de donner les moyens d'effectuer diverses modifications des en-têtes des paquets, afin de les marquer pour un traitement ultérieur spécifique. La réponse recommandée est 'N'.

L'option « **TOS target support** » permet d'utiliser la cible **TOS** dans les règles des chaînes de la table **mangle**. Cette cible autorise la modification du champ « **Type Of Service** » des paquets, principalement dans le but de modifier leur priorité. La réponse recommandée est 'N'.

L'option « **ECN target support** » permet d'utiliser la cible **ECN** dans les règles des chaînes de la table **mangle**. Cette cible autorise la modification du champ « **ECN** » des paquets. La réponse recommandée est 'N'.

L'option « **TTL target support** » active la gestion de la cible **TTL**, qui permet de modifier la durée de vie des paquets lors des passages au travers des routeurs. La réponse recommandée est 'N'.

L'option « **CLUSTERIP target support (EXPERIMENTAL)** » permet de distribuer les connexions sur une grappe de serveur pour réaliser de la répartition de charge. La réponse recommandée est 'N'.

L'option « **raw table support (required for NOTRACK/TRACE)** » permet de prendre en charge le filtrage des paquets bruts au niveau des tables PREROUTING et OUTPUT. Cette option vous permettra d'activer la prise en charge de la cible NOTRACK.

L'option « **ARP tables support** » permet de permettre la prise en charge des paquets ARP au niveau du code de filtrage du noyau. La réponse recommandée est 'N'.

L'option « **ARP packet filtering** » permet de filtrer et sélectionner les paquets ARP au niveau du noyau, éventuellement pour les modifier. La réponse recommandée est 'N'.

L'option « **ARP payload mangling** » permet de modifier à la volée les informations circulant dans les paquets ARP, comme l'adresse source et l'adresse destination, ainsi que les adresses MAC. La réponse recommandée est 'N'.

XII-H-1-b-iii - Sous-menu « IPv6: Netfilter Configuration »

L'option « **IPv6 connection tracking support (EXPERIMENTAL)** » permet de prendre en charge le protocole IPv6 au niveau du nouveau gestionnaire de suivi de connexion, que l'on peut activer avec l'option « **Layer 3 Independent Connection tracking (EXPERIMENTAL)** ». La réponse recommandée est 'N'.

L'option « **IP6 Userspace queueing via NETLINK (OBSOLETE)** » permet d'activer la gestion des transferts des paquets IPv6 sur un processus utilisateur via l'ancienne interface NETLINK. Cette interface est à présent obsolète, aussi la réponse recommandée est-elle 'N'.

L'option « **IP6 tables support (required for filtering/masq/NAT)** » permet d'activer la gestion de Netfilter pour le protocole IPv6. La réponse recommandée est 'N'.

L'option « **Routing header match support (EXPERIMENTAL)** » permet d'activer le critère de sélection basé sur les informations de routage des paquets. La réponse recommandée est 'N'.

L'option « **Hop-by-hop and Dst opts header match (EXPERIMENTAL)** » permet d'activer le critère de sélection des paquets selon leurs options « hop by hop » et destination. La réponse recommandée est 'N'.

L'option « **Fragmentation header match support (EXPRERIMENTAL)** » permet d'activer le critère de sélection selon les en-têtes de gestion des paquets fragmentés. La réponse recommandée est 'N'.

L'option « **HL match support** » permet d'activer le critère de sélection basé sur le nombre maximum de sauts d'un paquet indiqué dans son en-tête. La réponse recommandée est 'N'.

L'option « **Owner match support (EXPERIMENTAL)** » permet de sélectionner les paquets créés par les processus locaux en utilisant comme critère les identifiants de groupe, de processus et d'utilisateur de celui qui les a créés. La réponse recommandée est 'N'.

L'option « **IPv6 Extensions Headers Match (EXPERIMENTAL)** » permet d'activer la gestion du critère de sélection des paquets utilisant des extensions IPv6. La réponse recommandée est 'N'.

L'option « **AH match support** » permet d'activer la sélection des paquets des connexions IPv6 utilisant le protocole d'authentification AH. La réponse recommandée est 'N'.

L'option « **MH match support** » permet d'activer la sélection des paquets des connexions IPv6 utilisant le protocole MH. La réponse recommandée est 'N'.

L'option « **UEI64 address check (EXPERIMENTAL)** » permet d'activer la vérification de la validité des adresses IPv6 sources. La réponse recommandée est 'N'.

L'option « **Packet filtering** » permet d'activer la gestion de la table **filter**, afin de réaliser par exemple un Firewall. La réponse recommandée est 'N'.

L'option « **LOG target support** » active la gestion de la cible **LOG**. Cette cible permet d'enregistrer des messages de débogage dans les fichiers de traces du système. On veillera à activer également la gestion des limites sur les règles de filtrage afin d'éviter d'engorger les fichiers de traces si l'on désire utiliser cette option. La réponse recommandée est 'N'.

L'option « **REJECT target support** » active la gestion de la cible **REJECT**. Cette cible permet de rejeter les paquets concernés et de prévenir le correspondant de ce rejet. La réponse recommandée est 'N'.

L'option « **Packet mangling** » permet d'activer la gestion de la table **mangle**, qui autorise la modification de certains champs des en-têtes des paquets IP. La réponse recommandée est 'N'.

L'option « **HL (hoplimit) target support** » permet de prendre en charge la gestion de la cible **HL**, qui permet de spécifier le nombre de routeurs qu'un paquet peut traverser. Mal utilisée, cette option peut générer des paquets dont la durée de vie est virtuellement infinie. La réponse recommandée est 'N'.

L'option « **raw table support (required for TRACE)** » permet de prendre en charge la gestion des filtres sur les paquets bruts, au niveau des tables **PREROUTING** et **OUTPUT**. La réponse recommandée est 'N'.

XII-H-1-b-iv - Sous-menu « DECnet: Netfilter Configuration »

L'option « **Routing message grabulator (for userland routing daemon)** » permet d'activer l'interface noyau permettant au démon de routage réseau pour les réseaux DECnet de fonctionner. La réponse recommandée est 'N'.

XII-H-1-b-v - Sous-menu « Bridge: Netfilter Configuration »

L'option « **Ethernet Bridge tables (ebtables) support** » permet d'activer les fonctionnalités de sélection des paquets Ethernet. Ces fonctionnalités permettent de réaliser des ponts filtrants entre deux réseaux Ethernet. Les options qui suivent permettent d'identifier les paquets Ethernet en fonction de leurs caractéristiques. La réponse recommandée est 'N'.

XII-H-1-c - Sous-menu « DCCP Configuration (EXPERIMENTAL) »

L'option « **The DCCP Protocol (EXPERIMENTAL)** » permet de prendre en charge le protocole de datagrammes avec gestion de congestion DCCP directement au niveau noyau. Cette option vous donnera accès à des sous-options permettant de configurer ce protocole. La réponse recommandée à cette question est 'N'.

XII-H-1-d - Sous-menu « SCTP Configuration (EXPERIMENTAL) »

L'option « **The SCTP Protocol (EXPERIMENTAL)** » permet de prendre en charge le protocole diffusion de flux multimédia (streaming) SCTP directement au niveau noyau. Cette option vous donnera accès à des sous-options permettant de configurer ce protocole. La réponse recommandée à cette question est 'N'.

XII-H-1-e - Sous-menu « TIPC Configuration (EXPERIMENTAL) »

L'option « **The TIPC Protocol (EXPERIMENTAL)** » permet de prendre en charge le protocole de communication inter processus utilisé dans certains clusters. Cette option vous donnera accès à des sous-options permettant de configurer ce protocole et la géométrie de votre cluster. La réponse recommandée pour un particulier est certainement 'N'.

XII-H-1-f - Sous-menu « QoS and/or fair queueing »

L'option « QoS and/or fair queueing » permet d'activer les options de configuration des algorithmes qui fixent les priorités sur les paquets à envoyer sur le réseau. La réponse recommandée est 'N'.

Les options qui suivent permettent de choisir les algorithmes à utiliser pour la détermination des priorités d'émission. Parmi ces algorithmes, on retrouve les algorithmes basés sur la notion de qualité de service, qui peuvent être activés grâce à l'option « QoS support ». Les options suivantes permettent de paramétrer les notions attachées à la qualité de service.

XII-H-1-g - Sous-menu « Network testing »

L'option « Packet Generator (USE WITH CAUTION) », permet de générer différents types de paquets et de les envoyer sur une interface réseau à des fins de tests. Cette opération peut être dangereuse pour la machine locale et les autres machines du réseau, aussi la réponse recommandée est-elle 'N'.

L'option « TCP connection probing », permet de capturer tous les changements d'état des connexions TCP lors de l'arrivée des paquets TCP. Cette option n'est utile que pour les développeurs du noyau, pour déboguer les modules de gestion des congestions TCP, aussi la réponse recommandée est-elle 'N'.

XII-H-2 - Menu « Amateur Radio support »

L'option « Amateur Radio support » permet d'activer la gestion des communications Radio de Linux. Les options qui suivent permettent de préciser les options du protocole réseau utilisé pour ces communications. La réponse recommandée est 'N'.

XII-H-2-a - Sous-menu « AX.25 network device drivers »

Ce menu permet de choisir les gestionnaires de périphériques bas niveau à utiliser pour le support des communications Radio. Il faut choisir le pilote correspondant à votre matériel.

XII-H-3 - Menu « IrDA (infrared) subsystem support »

L'option « IrDA (infrared) subsystem support » permet d'activer la gestion des périphériques infrarouges. La réponse recommandée est 'N'.

Il faut choisir le protocole de communication que vous désirez dans les options qui suivent. Les dernières options permettent de choisir les options de ces protocoles.

XII-H-3-a - Sous-menu « Infrared-port device drivers »

Ces options permettent de choisir les pilotes bas niveau pour les périphériques infrarouges. Vous devez choisir le pilote correspondant à votre matériel.

XII-H-4 - Menu « Bluetooth subsystem support »

L'option « Bluetooth subsystem support » permet d'activer la gestion des périphériques sans fil utilisant la norme Bluetooth. La réponse recommandée est 'N'.

L'option « **L2CAP protocol support** » active la prise en charge du protocole de communication L2CAP, qui permet de travailler en mode connecté ou non avec les périphériques Bluetooth. Ce protocole étant utilisé par la majorité des périphériques, il est recommandé de répondre 'Y' à cette question.

L'option « **SCO links support** » permet d'activer la gestion de la voix au travers du protocole Bluetooth. La réponse recommandée est 'N'.

L'option « **RFCOMM protocol support** » permet d'activer le support des communications connectées sur la technologie Bluetooth. L'option qui suit permet de permettre l'utilisation d'un terminal par ce protocole. La réponse recommandée est 'N'.

L'option « **BNEP protocol support** » permet de créer une interface réseau Ethernet virtuelle utilisant les fonctionnalités Bluetooth comme couche de transport. Cela permet de créer des réseaux privés Bluetooth. Les options qui suivent permettent de spécifier les fonctionnalités disponibles au niveau de l'interface implémentée par ce gestionnaire. La réponse recommandée est 'N'.

L'option « **CMTP protocol support** » permet d'activer le support de l'interface CAPI pour les périphériques ISDN Bluetooth. La réponse recommandée est 'N'.

L'option « **HIDP protocol support** » permet d'activer la prise en charge des périphériques d'interface homme-machine Bluetooth. La réponse recommandée est 'N'.

XII-H-4-a - Sous-menu « Bluetooth device drivers »

L'option « **HCI USB driver** » permet de prendre en charge les périphériques Bluetooth utilisant une interface USB. La réponse recommandée est 'N'.

L'option « **SCO (voice) support** » permet d'activer le support du protocole SCO sur les périphériques Bluetooth USB. La réponse recommandée est 'N'.

L'option « **HCI UART driver** » permet de prendre en charge les périphériques Bluetooth utilisant une interface série classique. La réponse recommandée est 'N'.

L'option « **UART (H4) protocol support** » permet de prendre en charge le protocole de communication série H4 utilisé par les périphériques Bluetooth et leur hôte. Ce protocole est utilisé par un grand nombre de périphériques. La réponse recommandée est 'N'.

L'option « **BCSP protocol support** » permet de prendre en charge le protocole de communication série BCSP utilisé par les périphériques Bluetooth et leur hôte. L'option « **Transmit CRC with every BCSP packet** » permet de demander à ce qu'un code de contrôle de redondance cyclique soit transmis avec chaque paquet du protocole. Cela réduit les performances, mais rend le protocole plus sûr. Les réponses recommandées sont 'N'.

L'option « **HCI BCM203x USB driver** » permet de prendre en charge les périphériques BCM203 d'HCI. La réponse recommandée est 'N'.

L'option « **HCI BPA10x USB driver** » permet de prendre en charge les périphériques USB BPA10x d'HCI. La réponse recommandée est 'N'.

L'option « **HCI BlueFRITZ! USB driver** » permet de prendre en charge les périphériques BlueFRITZ! d'HCI. La réponse recommandée est 'N'.

L'option « **HCI DTL1 (PC Card) driver** » permet de prendre en charge les cartes PCMCIA Bluetooth disposant de l'interface DTL1 de Nokia. La réponse recommandée est 'N'.

L'option « HCI BT3C (PC Card) driver » permet de prendre en charge les cartes PCMCIA Bluetooth disposant de l'interface BT3C de 3Com. La réponse recommandée est 'N'.

L'option « HCI BlueCard (PC Card) driver » permet de prendre en charge les cartes PCMCIA Bluetooth disposant de l'interface BlueCard de Anycor. La réponse recommandée est 'N'.

L'option « HCI UART (PC Card) driver » permet de prendre en charge les cartes PCMCIA Bluetooth disposant d'une interface UART. La réponse recommandée est 'N'.

L'option « HCI VHCI virtual HCI device driver » permet d'activer le périphérique HCI virtuel, qui vous permettra de simuler des périphériques Bluetooth. La réponse recommandée est 'N'.

XII-H-5 - Option « RxRPC session socket »

L'option « RxRPC session sockets » permet la prise en charge des sessions RxRPC, qui sont utilisées par le système de fichiers réseau AFS. La réponse recommandée est 'N'.

L'option « RxRPC dynamic debugging » permet de configurer le niveau de trace des liaisons RxRPC dynamiquement. La réponse recommandée est 'N'.

L'option « RxRPC Kerberos security » permet la prise en charge des connexions RxRPC protégées par Kerberos. La réponse recommandée est 'N'.

XII-H-6 - Menu « Wireless »

L'option « Improved wireless configuration API » permet de prendre en charge des extensions à l'interface de programmation des extensions pour les réseaux sans fil de Linux. Actuellement, Linux fournit deux bibliothèques génériques utilisables par les pilotes de périphériques des réseaux sans fil : la bibliothèque SoftMac, qui est la bibliothèque historique et que de nombreux pilotes utilisent encore, et la bibliothèque mac80211 (anciennement nommée DeviceScape, du nom de la société qui l'a développée et l'a offert aux développeurs du noyau), plus performante, mais pour laquelle les pilotes de périphériques n'ont pas encore été adaptés. L'interface de programmation étendue n'est utile que pour cette dernière bibliothèque, et elle est activée automatiquement si nécessaire. La réponse recommandée est 'Y'.

L'option « Wireless extensions » permet de prendre en charge l'interface de programmation des extensions pour les réseaux sans fil. Cette interface est utilisée par la plupart des pilotes de périphériques, aussi la réponse recommandée est-elle 'Y'.

L'option « Generic IEEE 802.11 Networking Stack (mac80211) » permet d'activer le support générique des réseaux 802.11 via la bibliothèque mac80211 développée par la société DeviceScape. Cette bibliothèque est utilisée par certains pilotes de périphériques et vous devrez l'activer si vous disposez d'un tel pilote. Toutefois, la plupart des pilotes n'ont pas encore été modifiés pour utiliser cette nouvelle bibliothèque, aussi pouvez-vous répondre par 'N' à cette question pour l'instant.

L'option « Enable LED triggers » permet de prendre en charge la gestion des diodes électroluminescentes présentes sur certains périphériques pour signaler les échanges sur le réseau sans fil. La réponse recommandée est 'Y'.

L'option « Export mac80211 internals in DebugFS » permet d'exposer des informations de débogage interne de la bibliothèque mac80211 au travers du système de fichiers virtuel DebugFS. La réponse recommandée est 'N'.

L'option « Enable debugging output » permet d'activer les traces de débogage de la bibliothèque mac80211. La réponse recommandée est 'N'.

L'option « **Verbose debugging output** » permet d'activer des traces de débogage complémentaires de la bibliothèque mac80211. La réponse recommandée est 'N'.

L'option « **Debug frame dumping** » permet d'activer des traces de débogage pour chacun des paquets 802.11 qui traverse la bibliothèque mac80211. La réponse recommandée est 'N'.

L'option « **TKIP debugging** » permet d'activer les traces de débogage de la bibliothèque mac80211 pour la gestion des clefs de chiffrement temporaires. La réponse recommandée est 'N'.

L'option « **Extra statistics for TX/RX debugging** » permet d'activer des statistiques sur les paquets 802.11 qui traversent la bibliothèque mac80211. La réponse recommandée est 'N'.

L'option « **Support for IBSS testing** » permet d'activer des fonctionnalités de test des réseaux Wi-Fi fonctionnant sans points d'accès (donc en mode ad-hoc). La réponse recommandée est 'N'.

L'option « **Verbose powersave mode debugging** » permet d'activer des traces de débogage complémentaires relatives à la gestion de l'énergie par la bibliothèque mac80211. La réponse recommandée est 'N'.

L'option « **Generic IEEE 802.11 Networking Stack** » permet d'activer le support générique des réseaux sans fil 802.11 par l'intermédiaire de la bibliothèque SoftMac utilisée historiquement dans Linux. Cette bibliothèque est en cours de remplacement par la bibliothèque mac80211 (alias DeviceScape), mais est encore utilisée par la plupart des pilotes de périphériques. La réponse recommandée est donc 'Y'.

L'option « **Enable full debugging output** » permet d'activer la génération des informations complémentaires de débogage de cette fonctionnalité. La réponse recommandée est 'N'.

Les options « **IEEE 802.11 WEP encryption (802.1x)** », « **IEEE 802.11i CCMP support** » et « **IEEE 802.11i TKIP encryption** » permettent de prendre en charge respectivement les algorithmes de chiffrement WEP, CCMP et TKIP. La réponse recommandée est 'Y'.

L'option « **Software MAC add-on to the IEEE 802.11 networking stack** » permet d'activer la prise en charge au niveau logiciel certaines tâches que le matériel ne gère pas toujours. Cette fonctionnalité peut être nécessaire pour gérer les fonctionnalités de chiffrement avec certains pilotes de périphériques Wi-Fi. La réponse recommandée est donc 'Y'. Si vous activez cette option, vous aurez la possibilité d'activer les traces de débogage de la pile SoftMac à l'aide de l'option « **Enable full debugging output** ». La réponse recommandée est 'N'.

XII-H-7 - Option « RF switch subsystem support »

L'option « **RF switch subsystem support** » permet la prise en charge des interrupteurs présents sur la plupart des adaptateurs sans fil et des portables et permettant de désactiver les communications radio. Certains de ces interrupteurs sont gérés directement par le matériel et ne nécessitent donc pas de ce pilote. La réponse recommandée est 'Y'.

L'option « **Input layer to RF switch connector** » permet de modifier l'activité des communications radio pour les périphériques pour lesquels cette activité est contrôlable logiciellement. Vous devrez utiliser une application complémentaire pour utiliser cette fonctionnalité, le noyau ne fournissant qu'une interface bas niveau avec ce pilote. La réponse recommandée est 'N'.

XII-H-8 - Option « Plan 9 Resource Sharing Support (9P2000) (Experimental) »

L'option « **Plan 9 Resource Sharing Support (9P2000) (Experimental)** » permet la prise en charge du protocole de partage de ressources 9P2000 développé pour le système Plan 9. La réponse recommandée est 'N'.

L'option « Debug information » permet d'activer des traces de débogage pour le sous-système Plan 9. La réponse recommandée est 'N'.

XII-I - Device Drivers

Ce menu contient l'ensemble des options relatives aux gestionnaires de périphériques pris en charge par Linux.

XII-I-1 - Menu « Generic Driver Options »

L'option « Select only drivers that don't need compile-time external firmware » permet de n'afficher que les options relatives aux gestionnaires de périphériques pour lesquels des fichiers sources complémentaires à ceux fournis avec le noyau ne sont pas nécessaires. La réponse recommandée est 'N'.

L'option « Prevent firmware from being built » permet de désactiver la compilation des firmware fournis avec le noyau. En général, le firmware ne doit être recompilé que lors d'une mise à jour de firmware et non du noyau, aussi cette option doit-elle généralement être activée. La réponse recommandée est donc 'Y'.

L'option « Userspace firmware loading support » permet d'inclure le support du chargement du firmware des périphériques à chaud, quand bien même aucun gestionnaire de périphérique présent dans le noyau ne le nécessiterait. Cela permet de disposer de cette fonctionnalité pour les gestionnaires de périphériques externes fournis sous forme de module. La réponse recommandée est 'Y'.

L'option « Driver Core verbose debug messages » permet d'activer les messages de trace du système de base des gestionnaires de périphériques. La réponse recommandée est 'N'.

L'option « Managed device resources verbose debug messages » permet d'activer les messages de trace des pilotes de périphériques dont la gestion des ressources est totalement prise en charge par le noyau. Ces pilotes de périphériques sont un nouveau type de pilotes, qui utilisent des fonctions avancées du noyau permettant de gérer automatiquement les ressources systèmes. La réponse recommandée est 'N'.

XII-I-2 - Menu « Connector - unified userspace <-> kernelspace linker »

L'option « Connector - unified userspace <-> kernelspace linker » permet d'activer une technique de communication générique entre le noyau et les processus utilisateurs, basée sur l'interface Netlink du noyau. La réponse recommandée est 'N'.

L'option « Report process events to userspace » permet d'ajouter un point de connexion permettant de remonter les événements concernant la vie des processus vers l'espace utilisateur. La réponse recommandée est 'N'.

XII-I-3 - Menu « Memory Technology Devices (MTD) »

Ce menu vous permet d'activer la gestion des puces de mémoire Flash ou autres périphériques de mémoire persistante. Ce genre de périphérique est généralement utilisé pour réaliser des systèmes de fichiers sur les systèmes embarqués, qui ne disposent pas nécessairement de disques magnétiques. L'option « Memory Technology Device (MTD) support » vous donnera accès aux choix des différents pilotes pour les matériels supportés. La réponse recommandée est 'N'. Les sous-menus de ce menu sont les menus « RAM/ROM/Flash chip drivers », « Mapping drivers for chip access », « Self-contained MTD device drivers », « NAND Flash Device Drivers », « OneNAND Flash Device Drivers » et « UBI - Unsorted block images ». Ce dernier menu permet de prendre en charge les fonctions d'agrégation de périphériques en mémoire Flash tout en prenant en compte la spécificité de ces mémoires (en particulier le faible nombre de réécritures possible avant destruction de la mémoire).

XII-I-4 - Menu « Parallel port support »

L'option « **Parallel port support** » permet d'activer la gestion du port parallèle au niveau du noyau. Il est recommandé de le faire sous la forme de module, car le port parallèle n'est pas souvent utilisé. De toute façon, il est recommandé d'activer cette fonctionnalité, en répondant par 'Y' ou par 'M'. Quelle que soit la réponse donnée, on pourra ainsi utiliser le port parallèle pour plusieurs fonctionnalités différentes (imprimante, lecteurs externes, etc.). Répondez par 'Y' ou par 'M'.

L'option « **PC-style hardware** » permet d'indiquer au noyau que le port parallèle est compatible PC. Si vous compilez le noyau pour un PC ou un Alpha, répondez par 'M' à cette question.

L'option « **MultiIO cards (parallel and serial)** » permet de prendre en charge les ports parallèles et séries additionnels fournis par des cartes PCI. La réponse recommandée est 'N'.

L'option « **Use FIFO/DMA if available (EXPERIMENTAL)** » permet de demander au noyau d'utiliser une interruption et un canal DMA pour accéder au port parallèle, si le chipset de la carte mère le supporte. Cela permet généralement d'accélérer les entrées / sorties sur le port parallèle, en évitant au noyau d'utiliser un mécanisme de consultation périodique de l'état du port afin de savoir s'il est capable d'accepter des données ou si des données doivent être lues. Notez que, par défaut, le noyau n'utilisera aucune ligne d'interruption pour accéder au port parallèle, et ce même si vous avez activé cette option. Pour changer le comportement par défaut, vous devrez passer le paramètre `parport` au noyau lors de son démarrage, suivi de l'adresse du port et de la ligne d'interruption à utiliser, séparés par une virgule. Si vous avez demandé la compilation du gestionnaire de ports parallèles sous forme de module, vous devrez spécifier ces options dans le fichier `/etc/modprobe.conf`. La réponse recommandée est 'Y'.

L'option « **SuperIO chipset support (EXPERIMENTAL)** » permet d'activer la gestion des chipset SuperIO présents sur certaines cartes mères. La réponse recommandée est 'N'.

L'option « **Support for PCMCIA management for PC-style ports** » permet d'activer la gestion des ports parallèles via les mécanismes de gestion des cartes PCMCIA. La réponse recommandée à cette question est 'N'.

L'option « **AX88796 Parallel Port** » permet d'activer la gestion des ports parallèles présents sur les cartes réseau AX88796. Ce pilote de périphérique est totalement indépendant du pilote de périphérique pour ces cartes réseau et peut donc être activé seul. La réponse recommandée est 'N'.

L'option « **IEEE 1284 transfer modes** » permet de paramétrer le gestionnaire de ports parallèles pour utiliser les communications bidirectionnelles du port. Cette option est utile si l'on utilise une imprimante capable d'indiquer son état à l'ordinateur. Notez que pour que cette fonctionnalité soit disponible, vous devez également paramétrer votre BIOS pour que le port parallèle soit en mode EPP ou ECP. La réponse recommandée est 'Y'.

XII-I-5 - Menu « Plug and Play support »

L'option « **Plug and Play support** » permet d'activer la gestion du plug and play au niveau du noyau. La réponse recommandée est 'Y'.

L'option « **PnP Debug Messages** » permet d'activer la gestion des traces du gestionnaire de périphériques Plug and Play. Cela permet de déboguer la configuration des périphériques Plug and Play plus facilement. La réponse recommandée est 'N'.

L'option « **ISA Plug and Play support** » permet d'activer les fonctionnalités Plug and Play pour les périphériques ISA. Il est ainsi possible d'éviter d'avoir à utiliser les outils `isapnp` et `pnpdump`, et d'éviter la compilation des gestionnaires de ces périphériques en modules. La configuration de ces périphériques ISA Plug and Play en est grandement simplifiée. Le bus ISA n'étant plus utilisé de nos jours, il est donc recommandé de répondre par 'N' à cette question.

L'option « **Plug and Play BIOS support (EXPERIMENTAL)** » permet d'activer la détection des fonctionnalités Plug and Play des cartes mères via le BIOS. Leur configuration est donc assurée de ne pas entrer en conflit avec celle des périphériques Plug and Play ISA présents. La réponse recommandée est 'N'. Si vous activez cette option, vous aurez accès à l'option « **Plug and Play BIOS /proc interface** », qui vous permettra d'accéder aux informations du BIOS Plug and Play via le système de fichiers virtuels /proc/.

L'option « **Plug and Play ACPI support (EXPERIMENTAL)** » permet d'activer la détection des fonctionnalités Plug and Play des cartes mères via l'interface ACPI. La plupart des machines récentes ne disposent plus de bus ISA, mais les périphériques classiques sont toujours présents. Ces périphériques sont généralement déclarés dans la configuration ACPI, et cette option permet d'obtenir leurs paramètres via cette interface. La réponse recommandée est 'Y'.

XII-I-6 - Menu « Block devices »

L'option « **Normal floppy disk support** » permet d'activer la gestion des lecteurs de disquettes sous Linux. Vous pouvez répondre par 'Y' à cette question si vous disposez d'un lecteur de disquettes.

L'option « **PS/2 ESDI hard disk support** » permet d'activer la gestion des disques ESDI sur les ordinateurs de type PS/2. Cette option n'est disponible que si vous avez activé la gestion du bus MCA à l'aide de l'option « **MCA support** » du menu « **General setup** ». La réponse recommandée est 'N'.

L'option « **XT hard disk support** » permet d'activer la gestion des disques durs XT. Ce sont de très vieux disques durs, que plus personne n'utilise. Répondez par 'N' à cette question.

L'option « **Parallel port IDE device support** » permet d'activer la gestion des périphériques IDE connectés sur port parallèle. Il est recommandé de placer ce pilote en module, car les lecteurs sur port parallèle ne sont pas toujours connectés à l'ordinateur. Répondez par 'M' à cette question.

L'option « **Parallel port IDE disks** » permet d'activer le support des disques IDE connectés sur port parallèle. Si vous disposez d'un tel disque, répondez par 'M' à cette question, et choisissez le protocole de communication sur port parallèle correspondant dans l'une des options suivantes. La réponse recommandée est 'N'.

L'option « **Parallel port ATAPI CD-ROMs** » permet d'activer la gestion des CD-ROM ATAPI connectés sur port parallèle. Si vous disposez d'un tel lecteur, répondez par 'M' à cette question, et choisissez le protocole de communication sur port parallèle correspondant dans l'une des options suivantes. La réponse recommandée est 'N'.

L'option « **Parallel port ATAPI disks** » permet d'activer la gestion des disques ATAPI connectés sur port parallèle. Si vous disposez d'un tel disque, répondez par 'M' à cette question, et choisissez le protocole de communication sur port parallèle correspondant dans l'une des options suivantes. La réponse recommandée est 'N'.

L'option « **Parallel port ATAPI tapes** » permet d'activer la gestion des lecteurs de bandes connectés sur port parallèle. Si vous disposez d'un tel lecteur, répondez par 'M' à cette question, et choisissez le protocole de communication sur port parallèle correspondant dans l'une des options suivantes. La réponse recommandée est 'N'.

L'option « **Parallel port generic ATAPI devices** » permet la gestion de périphériques ATAPI non standards connectés au port parallèle. Les logiciels utilisateur peuvent envoyer des commandes ATAPI spécifiques à ces périphériques par l'intermédiaire de ce gestionnaire. En particulier, les graveurs de CD connectés sur port parallèle utilisent cette fonctionnalité. Si vous disposez d'un graveur de CD sur port parallèle, répondez par 'M' à cette question, et choisissez le protocole de communication sur port parallèle correspondant dans l'une des options suivantes. La réponse recommandée est 'N'.

Les options qui suivent permettent de choisir les protocoles de communication sur port parallèle adaptés à votre matériel. Vous devez en choisir au moins un si vous comptez utiliser un périphérique IDE connecté sur le port parallèle. Les réponses recommandées sont 'N' pour les protocoles que votre matériel ne comprend pas.

L'option « Compaq SMART2 support » permet d'activer la gestion des cartes contrôleurs Smart Array de Compaq. À moins que vous ne disposiez d'une telle carte, la réponse recommandée est 'N'.

L'option « Compaq Smart Array 5xxx support » permet d'activer la gestion des cartes contrôleurs Smart Array 5xxx de Compaq. À moins que vous ne disposiez d'une telle carte, la réponse recommandée est 'N'. Si vous activez cette fonctionnalité, vous pourrez également activer l'option « SCSI tape drive support for Smart Array 5xxx », qui permet d'utiliser des périphériques SCSI connectés sur les contrôleurs Smart Array 5xxx. La réponse recommandée pour cette option est également 'N'.

L'option « Mylex DAC960/DAC1100 PCI RAID Controller support » permet d'activer la gestion des contrôleurs RAID Mylex. La réponse recommandée est 'N'.

L'option « Micro Memory MM5415 Battery Backed RAM support (EXPERIMENTAL) » permet d'activer la gestion des cartes mémoires alimentées par batterie MM5415. La réponse recommandée est 'N'.

L'option « Virtual block device » permet d'activer la gestion des périphériques de type bloc du système hôte dans les systèmes virtualisés par UML (User Mode Linux). Cela permet d'accélérer fortement les entrées/sorties dans ces systèmes. La réponse recommandée est donc 'Y'.

L'option « Always do synchronous disk IO for UBD » permet de faire en sorte que les opérations d'écritures sur les périphériques virtuels des systèmes virtualisés par UML sont toujours réalisées de manière synchrone au niveau du système hôte. Cela permet de faire fonctionner correctement les systèmes de fichiers journalisés pour ces systèmes. La réponse recommandée est 'Y'.

L'option « Loopback device support » permet d'utiliser des fichiers classiques pour y placer un système de fichiers. Cela est essentiellement utilisé pour créer des images de CD et les tester avant de les graver. Si vous disposez d'un graveur de CD-ROM, il est recommandé de répondre 'Y' à cette question. Sinon, répondez 'M', pour vous réserver la possibilité d'utiliser des systèmes de fichiers stockés dans des fichiers classiques.

L'option « Cryptoloop Support » permet d'activer le chiffrement des disques durs par les algorithmes de chiffrement de Linux, en passant par l'intermédiaire d'un système de fichiers loopback. Cette option nécessite d'inclure le support des algorithmes de chiffrement, accessibles dans le menu « Cryptographic options ». Cette option n'est pas cryptographiquement sûre, et ne permet pas de stocker des systèmes de fichiers journalisés de manière fiable. Aussi n'est-elle pas recommandée, et on lui préférera les fonctionnalités de gestion des volumes disponibles dans le menu « Multidevice support (RAID and LVM) ». La réponse recommandée pour cette option est donc 'N'.

L'option « Network block device support » permet d'accéder aux fichiers spéciaux de périphériques de type bloc d'un ordinateur distant par l'intermédiaire du réseau. Cette fonctionnalité est peu utilisée, la réponse recommandée est donc 'N'.

L'option « Promise SATA SX8 support » permet d'activer la gestion des contrôleurs Serial ATA SX8. Vous devez l'activer si vous en disposez d'un. La réponse recommandée est 'N'.

L'option « Low Performance USB Block driver » permet de prendre en charge certains périphériques de masse USB très lents. La réponse recommandée est 'N'.

L'option « RAM disk support » permet d'activer la gestion des disques virtuels (chargés en mémoire). Notez que cette option n'est pas nécessaire pour activer les systèmes de fichiers en mémoire (ramfs), car ceux-ci sont intégrés d'office dans le noyau. Elle ne sert qu'à créer des périphériques de type bloc afin de fixer la taille des disques virtuels (le système de fichiers ramfs ne nécessite pas ces périphériques et adapte dynamiquement sa taille en fonction des besoins). Ces disques virtuels sont généralement utilisés pour créer des disquettes de réparation, qui chargent les utilitaires systèmes sur un disque en mémoire. La réponse recommandée est donc 'N' pour un système normal, et 'Y' pour un système destiné à être placé sur une disquette de réparation.

L'option « Default number of RAM disks » permet de fixer le nombre de disques virtuels par défaut. La valeur recommandée est '16'.

L'option « **Default RAM disk size (kbytes)** » permet de fixer la taille par défaut des disques virtuels. La taille recommandée est de 4096 Ko.

L'option « **Default RAM disk block size (bytes)** » permet de fixer la taille par défaut des blocs sur les disques virtuels. La taille recommandée est 4096, mais la taille par défaut est de 1024 Ko pour des raisons de compatibilité avec le chargeur de système initrd.

L'option « **Packet writing on CD/DVD media** » permet d'activer l'écriture par paquet sur les supports CD et DVD réinscriptibles. Cette fonctionnalité permet d'utiliser ces supports directement comme des lecteurs amovibles classiques, sans avoir à utiliser de logiciel de gravage spécifique. La réponse recommandée est 'Y'. Les sous-options « **Free buffers for data gathering** » et « **Enable write caching** » de cette option permettent de spécifier le nombre de tampons d'entrée/sortie utilisés simultanément pour les écritures et si les écritures doivent être effectuées de manière asynchrone afin d'optimiser les performances. Dans ce dernier cas, les erreurs d'écritures ne sont pas encore gérées, aussi est-il préférable de répondre par 'N' à cette dernière question.

L'option « **ATA over Ethernet support** » permet de prendre en charge les périphériques de stockage ATA connectables via un port Ethernet. La réponse recommandée est 'N'.

L'option « **Xen virtual block device support** » permet de prendre en charge les périphériques de type bloc virtuels pour les systèmes virtualisés par Xen. Cela permet d'accroître les performances des entrées/sorties sur les périphériques de type bloc du système hôte. La réponse recommandée est 'Y'.

XII-I-7 - Sous-menu « Misc devices »

L'option « **Device driver for IBM RSA service processor** » active la prise en charge des coprocesseurs cryptographiques Condor d'IBM. La réponse recommandée est 'N', sauf si vous disposez d'un tel coprocesseur.

L'option « **Sensable PHANToM** » active la prise en charge des périphériques Sensable PHANTOM. La réponse recommandée est 'N', sauf si vous disposez d'un tel coprocesseur.

L'option « **EEPROM 93CX6 support** » active la prise en charge des puces EEPROM 93c46 et 93c66. La réponse recommandée est 'N', sauf si vous disposez d'un tel coprocesseur.

L'option « **SGL IOC4 Base IO support** » active la prise en charge des puces IOC4 existantes sur certaines cartes mères SGI. Ce pilote n'est pas utile en soit, mais est utilisé par d'autres pilotes de périphérique. La réponse recommandée est 'N', sauf si vous disposez d'une telle carte mère.

L'option « **TI Flash Media interface support** » active la prise en charge des adaptateurs Flash Media de Texas Instruments. Cette option vous donnera accès aux options permettant de sélectionner les pilotes de périphériques pour l'adaptateur dont vous disposez. La réponse recommandée est 'N', sauf si vous disposez d'un tel adaptateur.

L'option « **Asus Laptop Extras (EXPERIMENTAL)** » active la prise en charge de contrôle des réseaux Bluetooth et Wi-Fi des portables Asus, ainsi que le contrôle de la luminosité de l'écran LCD et des événements de gestion d'énergie. Ce pilote de périphérique fonctionne également sur certains ordinateurs de marque Medion, JVC ou Victor. La réponse recommandée est 'N', sauf si vous disposez d'un tel ordinateur.

L'option « **MSI Laptop Extras** » active la prise en charge de contrôle des réseaux Bluetooth et Wi-Fi des portables MSI, ainsi que le contrôle de la luminosité de l'écran LCD. La réponse recommandée est 'N', sauf si vous disposez d'un tel ordinateur.

L'option « **Sony Laptop Extras (EXPERIMENTAL)** » active la prise en charge de contrôle des réseaux Bluetooth des portables Sony, ainsi que le contrôle de la luminosité de l'écran LCD et des touches d'extinction de certains périphériques. Cette option vous donnera accès à l'option « **Sonypi compatibility** » permettant d'activer la compatibilité avec le pilote sonypi. La réponse recommandée est 'N', sauf si vous disposez d'un tel ordinateur.

L'option « **ThinkPad ACPI Laptop Extras** » active la prise en charge de contrôle des réseaux Bluetooth des portables ThinkPad d'IBM ou de Lenovo, ainsi que le contrôle de la luminosité de l'écran LCD et des touches de fonctions de ces ordinateurs. Cette option vous donnera accès aux options « **Verbose debug mode** », « **Legacy Docking Station Support** », « **Legacy Removable Bay Support** » et « **Enable input layer support by default** » qui permettent de spécifier le niveau de traces du pilote, d'activer les pilotes de gestion à présent obsolètes pour la détection de la base et des périphériques amovibles, et de rediriger les informations des touches de fonctions vers le pilote d'entrée classique de Linux. La réponse recommandée est 'N', sauf si vous disposez d'un tel ordinateur.

XII-I-8 - Menu « ATA/ATAPI/MFM/RLL support »

L'option « **ATA/ATAPI/MFM/RLL support** » permet d'activer ou de désactiver globalement le support des périphériques IDE. À moins que votre ordinateur ne contienne que des périphériques SCSI, vous devez activer cette option.

L'option « **Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support** » permet d'activer la gestion des disques IDE. À moins que votre ordinateur ne soit complètement SCSI, répondez par 'Y' ou 'M' à cette question. Si vous répondez par 'M', il faut que Linux soit installé sur un disque SCSI amorçable. Il est recommandé de répondre par 'Y' à cette question.

L'option « **Support for SATA (deprecated; conflicts with libata SATA driver)** » permet d'activer la prise en charge de quelques contrôleurs Serial ATA de première génération. Normalement, les disques Serial ATA doivent être utilisés via le gestionnaire de périphérique SCSI, aussi l'activation de cette option est-elle déconseillée. La réponse recommandée est 'N'.

L'option « **Use old disk-only driver on primary interface** » permet d'utiliser un vieux pilote pour les disques IDE connectés à la première interface IDE et qui poseraient quelques problèmes avec le pilote général. Seule la première interface sera concernée par cette option, les autres interfaces utiliseront le nouveau pilote. En général, la réponse à cette question est 'N'.

L'option « **Include IDE/ATA-2 DISK support** » permet d'utiliser les disques durs IDE et ATAPI. Si l'ordinateur possède un disque IDE, il est fortement recommandé de répondre 'Y' à cette question. On ne doit répondre 'N' que si l'ordinateur est complètement SCSI. De plus, on prendra garde au fait que l'on ne peut pas mettre les fonctionnalités nécessaires au démarrage de l'ordinateur dans des modules. Cela signifie que si le disque de démarrage est un disque IDE, il ne faut pas mettre cette fonctionnalité dans un module. Donc, pour la plupart des gens, il faut répondre 'Y' à cette question.

L'option « **Use multimode by default** » permet d'activer le mode multimode par défaut sur les disques IDE. Cette option n'est normalement pas nécessaire, ce mode étant automatiquement choisi par le noyau. Toutefois, elle peut être nécessaire dans de rares cas. La réponse recommandée est 'N'.

L'option « **PCMCIA IDE support** » permet d'activer la gestion des périphériques IDE connectés par un port PCMCIA sur les portables. Cette option n'est accessible que si vous avez activé la gestion des cartes PCMCIA avec l'option « **CardBus support** » du menu « **PCMCIA/CardBus support** ». La réponse recommandée est 'N'.

L'option « **Cardbus IDE support (Delkin/ASKA/Workbit)** » permet d'activer la gestion des adaptateurs IDE Cardbus Delkin, ASKA ou Workbit. Sur les portables. Cette option n'est accessible que si vous avez activé la gestion des cartes PCMCIA avec l'option « **CardBus support** » du menu « **PCMCIA/CardBus support** ». La réponse recommandée est 'N'.

L'option « **Include IDE/ATAPI CDROM support** » permet d'utiliser les CD-ROM IDE et ATAPI. La plupart des ordinateurs possèdent un lecteur de CD-ROM ATAPI actuellement, il est donc recommandé d'activer cette fonctionnalité. La réponse recommandée est 'Y'.

L'option « **Include IDE/ATAPI TAPE support (EXPERIMENTAL)** » permet d'inclure la gestion des lecteurs de bandes magnétiques IDE et ATAPI. Comme les lecteurs de disques amovibles IDE sont assez rares, la réponse recommandée est 'N'.

L'option « **Include IDE/ATAPI FLOPPY support** » permet d'inclure la gestion des lecteurs de disques amovibles IDE et ATAPI. C'est en particulier le cas pour les lecteurs LS120 et ZIP. Comme les lecteurs de disques amovibles IDE sont assez rares, la réponse recommandée est 'N'.

L'option « **SCSI emulation support** » permet d'émuler la présence d'un périphérique IDE par un périphérique SCSI en convertissant les requêtes SCSI en requêtes ATAPI. Cette fonctionnalité n'est utile que lorsqu'on doit utiliser des logiciels bas niveau qui ne peuvent travailler qu'avec des périphériques SCSI. Cette option était nécessaire pour utiliser les graveurs de CD IDE avec les anciennes versions du noyau, mais ce n'est plus le cas à présent. Cette option n'est donc plus utile de nos jours et la réponse recommandée est 'N'.

L'option « **IDE ACPI support** » permet d'activer les fonctions d'énergie ACPI pour les périphériques IDE. La réponse recommandée est 'Y'.

L'option « **IDE Taskfile Access** » permet d'activer l'envoi de commandes directement au niveau du contrôleur IDE. La réponse recommandée est 'N'.

L'option « **legacy /proc/ide/ support** » permet d'activer l'affichage des informations sur les bus IDE dans le système de fichiers virtuel /proc/. Ces informations sont à présent exposées au travers du système de fichiers virtuels /sys/, aussi cette option n'est-elle fournie qu'à titre de compatibilité pour les applications qui n'auraient pas encore été mises à jour. La réponse recommandée est 'Y'.

L'option « **generic/default IDE chipset support** » permet d'activer la prise en charge générique des périphériques ATAPI en cas d'absence d'un gestionnaire plus spécialisé. La réponse recommandée est 'N'.

L'option « **CMD640 chipset bugfix/support** » permet de contourner un bogue des chipsets IDE CMD640. Vous devez activer cette option si vous disposez d'un tel chipset. La réponse recommandée est 'Y' si vous avez un tel chipset, et 'N' sinon.

L'option « **CMD640 enhanced support** » permet l'autodétection des paramètres idéaux pour les chipsets IDE CMD640. Cette option n'est disponible que si vous avez activé le support des chipsets CMD640 dans la question précédente. La réponse recommandée est 'Y' si vous avez un tel chipset, et 'N' sinon.

L'option « **PNP EIDE support** » permet de prendre en charge la gestion des cartes ISA Plug and Play permettant d'utiliser des disques EIDE additionnels. Cette option peut être nécessaire si ces cartes doivent être initialisées avant de chercher à utiliser les disques qui y sont connectés. La réponse recommandée est 'N'.

L'option « **PCI IDE chipset support** » permet d'activer la gestion des périphériques IDE sur bus PCI. Vous devez répondre par 'Y' à cette question si vous disposez d'une carte mère PCI et de contrôleurs IDE, ce qui est normalement le cas pour tous les ordinateurs récents.

L'option « **Sharing PCI IDE interrupts support** » permet d'activer le partage des lignes d'interruptions utilisées par les contrôleurs IDE avec les cartes présentes sur le bus PCI. Cette fonctionnalité nécessite un support matériel particulier de la part du contrôleur IDE, support qui n'est présent que sur certaines cartes mères. En général, l'activation de cette option ne perturbe pas le fonctionnement du système sur les ordinateurs incapables d'effectuer le partage des lignes d'interruptions des contrôleurs IDE, mais la réponse recommandée reste 'N'.

L'option « **Boot off-board chipsets first support** » permet d'inverser la numérotation des disques IDE connectés aux contrôleurs de la carte mère et des disques IDE connectés aux contrôleurs additionnels que l'on peut avoir sur une carte fille. Il faut répondre par 'N' à cette question.

L'option « **Generic PCI IDE Chipset support** » permet d'activer la gestion des périphériques IDE génériques. La réponse recommandée à cette question est 'Y'.

L'option « **OPTi 82C621 chipset enhanced support (EXPERIMENTAL)** » permet de prendre en charge les chipsets OPTi 82C621. La réponse recommandée est 'N'.

L'option « RZ1000 chipset bugfix/support » permet de prendre en charge les chipsets RZ1000 et de corriger un de leur bogue. La réponse recommandée est 'Y' si vous disposez d'un tel chipset, et de 'N' sinon.

L'option « Generic PCI bus-master DMA support » permet d'activer la gestion des disques Ultra DMA. La réponse recommandée est 'Y' si vous disposez d'une carte mère PCI gérant l'Ultra DMA et de disques IDE.

L'option « Force enable legacy 2.0.X HOSTS to use DMA » permet d'activer une vieille fonctionnalité non documentée et ne sera pas décrite ici. L'option recommandée 'N'.

L'option « Use PCI DMA by default when available » permet d'activer la gestion de l'Ultra DMA au démarrage. La réponse recommandée est 'Y', sauf si l'on veut désactiver le support de l'Ultra DMA.

L'option « Enable DMA only for disks » permet de désactiver les transferts DMA pour les périphériques IDE qui ne sont pas des disques durs. Cette option peut être activée si des périphériques connectés ne gèrent pas correctement les transferts DMA. La réponse recommandée est 'Y'.

Il suit un certain nombre d'options qui permettent d'activer un support étendu pour différents chipsets. Il est recommandé de répondre par 'N' à ces questions, sauf à celles correspondant aux chipsets effectivement présents sur votre carte mère.

L'option « IGNORE word93 Validation BITS » permet d'éviter une incohérence dans les spécifications du protocole matériel ATAPI. Ces spécifications n'ont pas été claires à un endroit, et il existe maintenant des différences mineures entre les différents chipsets présents sur le marché. Cette option permet de désactiver cette fonctionnalité ambiguë. Bien qu'il ne soit pas dangereux de répondre par l'affirmative à cette question, la réponse recommandée reste 'N'.

L'option « Old hard disk (MFM/RLL/IDE) driver » permet de prendre en charge les disques durs via un pilote de disque alternatif, qui peut être nécessaire sur certains vieux systèmes dont le comportement n'est pas semblable à celui des ordinateurs récents au niveau du séquençage des opérations. La réponse recommandée est 'N'.

XII-I-9 - Menu « SCSI device support »

L'option « RAID Transport Class » permet de prendre en charge la technologie RAID pour les périphériques SCSI. La réponse recommandée est 'N'.

L'option « SCSI device support » permet d'activer la prise en charge des périphériques SCSI. Il faut répondre par 'Y' ou 'M' à cette question si vous disposez de tels périphériques. De plus, si le noyau se trouve sur un disque SCSI ou s'il a besoin de composants se trouvant sur le disque SCSI pour s'amorcer, il ne faut pas mettre en module cette fonctionnalité. Ceux qui n'ont pas de périphérique SCSI et ne comptent pas en utiliser peuvent répondre 'N' à cette question. Sachez toutefois que de nombreux périphériques USB apparaissent comme des disques SCSI une fois connectés à l'ordinateur. C'est en particulier le cas des clés USB et de nombreux appareils photo numériques. Il est donc recommandé de répondre 'Y' à cette question, même si l'on ne dispose pas de périphérique réellement SCSI.

L'option « SCSI target support » permet d'accéder aux pilotes de périphériques SCSI en mode target. La réponse recommandée est 'Y'.

L'option « legacy /proc/scsi/ support » permet de définir les entrées concernant le sous-système SCSI dans le système de fichiers virtuel /proc/. Ces informations sont à présent exposées au travers du système de fichiers virtuel /sys/, mais des applications peuvent ne pas avoir été mises à jour, aussi est-il recommandé de répondre 'Y' à cette question.

L'option « SCSI disk support » permet d'activer la gestion des disques durs SCSI. Il ne faut pas répondre par 'M' si votre système a besoin d'un disque SCSI pour démarrer. Cette option permet également de gérer les lecteurs ZIP connectés sur port parallèle, ainsi que les périphériques comme les clés USB et bon nombre d'appareils photo numériques, qui apparaissent comme des disques SCSI une fois connectés à l'ordinateur. La réponse recommandée est donc 'Y'.

L'option « **SCSI tape support** » permet d'activer la gestion des périphériques à bande SCSI. Notez que les lecteurs de bande OnStream SC-x0 ne sont pas pris en charge par ce pilote générique, mais peuvent être utilisés malgré tout en activant l'option suivante « **SCSI OnStream SC-x0 tape support** ». Le pilote alors utilisé nécessitera la présence des fichiers spéciaux de périphérique `/dev/osstX`, qui utilisent le numéro de code majeur 206. La réponse recommandée pour ces deux options est 'N'.

L'option « **SCSI CD-ROM support** » permet d'activer la gestion des CD-ROM SCSI. Vous devez activer cette fonctionnalité également si vous utilisez l'émulation SCSI pour un graveur de CD ATAPI. Si vous êtes dans l'un de deux cas, il est recommandé de répondre 'Y' à cette question. Dans tous les autres cas, la réponse recommandée est 'N'.

L'option « **Enable vendor-specific extensions (for SCSI CDROM)** » permet d'activer la gestion de CD-ROM SCSI disposant de commandes SCSI spécifiques. C'est notamment le cas pour les lecteurs de CD multisessions NEC/TOSHIBA et les graveurs de CD HP. Pour la plupart des utilisateurs, la réponse recommandée est 'N'.

L'option « **SCSI generic support** » permet d'activer la gestion des périphériques SCSI non standards. Pour ces périphériques, il faut utiliser un programme capable d'envoyer les commandes SCSI appropriées à votre matériel. Il faut activer cette fonctionnalité pour la plupart des périphériques SCSI qui ne sont ni des disques, ni des lecteurs de CD-ROM, ni des lecteurs de bandes. C'est en particulier le cas si vous utilisez un graveur de CD-ROM SCSI ou IDE via l'interface d'émulation SCSI du noyau. Dans ce cas, la réponse recommandée est 'Y'. Pour la plupart des utilisateurs, la réponse recommandée est toutefois 'N'.

L'option « **SCSI media changer support** » permet de prendre en charge les périphériques de changement de média que l'on peut trouver dans les bibliothèques de bandes magnétiques ou dans certains jukebox. La réponse recommandée est 'N'.

L'option « **Probe all LUNs on each SCSI device** » permet d'effectuer la détection de tous les numéros logiques d'unités SCSI de chaque périphérique. Comme la plupart des périphériques SCSI ne disposent que d'un seul numéro d'unité logique, la réponse recommandée est 'N'.

L'option « **Verbose SCSI error reporting (kernel size +=12K)** » permet d'utiliser un jeu de messages d'erreurs alternatif pour le SCSI. Ces messages sont plus lisibles, mais prennent plus de place dans le noyau. La réponse recommandée est 'N'.

L'option « **SCSI logging facility** » permet d'activer les traces du sous-système SCSI. La réponse recommandée est 'N'.

L'option « **Asynchronous SCSI scanning** » permet d'activer la fonctionnalité de détection asynchrone des périphériques SCSI. Les bus SCSI ont en effet toujours été relativement lents à initialiser en raison de la recherche des périphériques qui y sont connectés. Avec cette option, le démarrage du système peut être accéléré en réalisant cette détection en asynchrone. Toutefois, l'initialisation du système peut échouer si celle-ci nécessite un périphérique SCSI non encore détecté. Il est possible de réaliser une synchronisation dans les scripts de démarrage du système en chargeant le module `scsi_wait_scan`. Toutefois, la réponse recommandée est 'N'.

XII-I-9-a - Sous-menu « SCSI Transport »

Ce jeu d'options permet d'exporter des informations sur les communications du sous-système SCSI dans le système de fichiers virtuels `/sys/`. La réponse recommandée pour ces options est 'N'.

XII-I-9-b - Sous-menu « SCSI low-level drivers »

Ce jeu d'options permet de sélectionner les gestionnaires de périphériques bas niveau SCSI adaptés à votre matériel. Il faut connaître la marque et le modèle de votre adaptateur SCSI, ainsi que ses paramètres pour répondre à ces questions. Si vous utilisez l'émulation SCSI pour les périphériques ATAPI ou des périphériques USB qui apparaissent comme des disques SCSI une fois connectés à l'ordinateur, il n'est pas nécessaire de choisir un gestionnaire bas

niveau et vous pouvez désactiver toutes ces options. Il est recommandé de répondre 'Y' à cette option et de choisir le pilote adapté à votre matériel.

XII-I-9-c - Sous-menu « PCMCIA SCSI adapter support »

Ce jeu d'options permet la prise en charge des périphériques SCSI au format PCMCIA. Les options qui suivent correspondent aux gestionnaires des différents types de matériel SCSI au format PCMCIA que Linux est capable de prendre en charge. Vous devez donc choisir les pilotes correspondants à votre matériel.

XII-I-10 - Menu « Serial ATA (prod) and Parallel ATA (experimental) drivers »

L'option « Serial ATA (prod) and Parallel ATA (experimental) drivers » permet d'activer la prise en charge générique des périphériques ATA. Cette prise en charge comprend la gestion des périphériques Serial ATA, qui sont parfaitement supportés, et un nouveau jeu de pilotes pour les périphériques ATAPI classiques (dont la connectique, à savoir les nappes IDE classiques, est, à l'inverse des périphériques Serial ATA, de type parallèle). Si les pilotes de périphériques Serial ATA peuvent être utilisés sans problème, les pilotes Parallel ATA sont beaucoup plus récents et sont encore considérés comme expérimentaux. Il est par conséquent recommandé de continuer à utiliser les pilotes de périphériques classiques du menu « ATA/ATAPI/MFM/RLL support » pour l'instant. Cette option vous donnera accès aux pilotes de périphériques pour les différents matériels supportés par Linux. La réponse recommandée est donc 'Y' si votre ordinateur dispose d'une interface Serial ATA, et vous devrez choisir le type de chipset Serial ATA utilisé dans les options suivantes.

Notez que les périphériques Serial ATA et les périphériques Parallel ATA de ce menu exposent les périphériques sous la forme de périphériques SCSI. Les fichiers spéciaux de ces périphériques ont donc un nom de la forme `/dev/sdXX` et non de la forme `/dev/hdXX`. À terme, lorsque les nouveaux pilotes Parallel ATA seront finalisés, tous les périphériques apparaîtront comme des périphériques SCSI et la manière d'accéder aux périphériques de type bloc sera uniforme.

XII-I-11 - Menu « Multidevice support (RAID and LVM) »

L'option « Multiple devices driver support (RAID and LVM) » permet d'activer la gestion de la redondance de données RAID et des volumes logiques. La technologie RAID permet de réaliser des agrégats de disques, soit dans le but de simuler des disques de grande capacité, soit afin de stocker les données de manière redondante sur plusieurs disques afin d'obtenir une sécurité accrue de ces données. La technologie LVM permet quant à elle uniquement de regrouper plusieurs volumes physiques afin de simuler la présence d'un volume logique de très grande capacité. Cette technologie utilise un mécanisme de regroupement des périphériques de type bloc que l'on appelle « Device Mapper ». Ce mécanisme permet également de transformer les données écrites sur les volumes à la volée, et en particulier de les chiffrer avec les algorithmes cryptographiques du noyau. Vous pouvez répondre par 'Y' à cette question si vous avez besoin de l'une de ces fonctionnalités. Dans le cas contraire, répondez par 'N'. La réponse recommandée est 'N'.

L'option « RAID support » permet d'activer la prise en charge des technologies RAID au niveau logiciel. Cette option n'est pas nécessaire pour utiliser les technologies RAID avec un matériel spécifique. La réponse recommandée est 'N'.

L'option « Linear (append) mode » permet de concaténer plusieurs partitions pour ne former qu'une seule zone de données plus grande. Cette fonctionnalité n'assure pas la redondance des données. La réponse recommandée est 'N'.

L'option « RAID-0 (striping) mode » permet de répartir les données sur plusieurs partitions de manière équilibrée. Cela permet de simuler des disques de très grande capacité, et également d'augmenter les performances en minimisant les temps d'accès, si ces partitions sont sur des disques différents. Cette fonctionnalité n'assure cependant pas la redondance des données. La réponse recommandée est 'N'.

L'option « RAID-1 (mirroring) mode » permet de dupliquer les données sur plusieurs disques à la volée. Cette redondance des données permet d'assurer une grande sécurité. La réponse recommandée est 'N'.

L'option « RAID-10 (mirrored striping) mode (EXPERIMENTAL) » permet de dupliquer les données sur plusieurs disques à la volée, dont l'espace peut être combiné pour réaliser des partitions équilibrées plus grandes. Ce mode de fonctionnement est donc un mode hybride entre le mode RAID-0 et le mode RAID-1, et offre donc un peu plus de souplesse. La réponse recommandée est 'N'.

L'option « RAID-4/RAID-5/RAID-6 mode » permet d'activer le support logiciel RAID-4, RAID-5 ou RAID-6. Dans le mode de fonctionnement RAID-4, un des disques est utilisé pour contrôler la validité des données sur les autres disques. Dans le mode de fonctionnement RAID-5, ces données de contrôle sont réparties sur tous les disques, pour une capacité toujours diminuée de la capacité de l'un des disques. Le système est tolérant à la défaillance d'un seul disque. Dans le mode RAID-6, les données de contrôle sont doublées et consomment deux disques sur l'ensemble des disques utilisés, mais le système tolère la défaillance de deux disques simultanément. La réponse recommandée est 'N'.

L'option « Support adding drives to a raid-5 array (experimental) » permet d'activer la fonctionnalité d'ajout de disque à un système RAID-5. Cela nécessite la redistribution des données sur l'ensemble des disques. Cette fonctionnalité permet de le faire en exploitation. Elle reste toutefois expérimentale, aussi la réponse recommandée est-elle 'N'.

L'option « Mutipath I/O support » permet d'activer le support des périphériques capables d'être accédés par différents jeux de ports d'entrée / sortie. La réponse recommandée est 'N'.

L'option « Faulty test module for MD » permet d'activer une fonctionnalité de génération d'erreurs aléatoires lors des opérations de lecture sur les périphériques multiples. Cette option est utile aux développeurs du noyau pour tester les techniques de reconstitution des informations dans les fonctionnalités RAID. La réponse recommandée est 'N'.

L'option « Device mapper support » permet d'activer la gestion des volumes virtuels. Avec cette option, vous pourrez faire des agrégats de plusieurs disques, périphériques RAID ou périphériques loopback afin de simuler un périphérique de type bloc de très grande capacité. Notez que cette fonctionnalité ne permet pas d'accéder aux partitions de type agrégat de Windows 2000 et Windows XP. Pour ce type de partition, consultez le sous-menu « Partition types » du menu « File systems ». La réponse recommandée est 'Y'.

L'option « Device mapper debugging support » permet d'activer des messages de débogage complémentaires lors de l'utilisation des volumes virtuels. La réponse recommandée est 'N'.

L'option « Crypt target support » permet d'utiliser les volumes virtuels pour chiffrer les systèmes de fichiers qui y sont stockés. La réponse recommandée est 'N'.

L'option « Snapshot target (EXPERIMENTAL) » permet d'effectuer une photographie de l'état complet d'un volume virtuel à un instant donné, de manière atomique. La réponse recommandée est 'N'.

L'option « Mirror target (EXPERIMENTAL) » permet de répliquer un volume sur un autre volume afin d'en avoir en permanence une copie conforme. La réponse recommandée est 'N'.

L'option « Zero target (EXPERIMENTAL) » permet de créer un volume virtuel sur lequel toutes les écritures sont ignorées, et dont les données sont toujours nulles en lecture. Ce type de volume peut être utile dans certaines situations de restauration. La réponse recommandée est 'N'.

L'option « Multipath target (EXPERIMENTAL) » permet de donner aux gestionnaires de volume la possibilité d'accéder aux périphériques capables de gérer plusieurs jeux de ports d'entrée/sortie. Si vous activez cette option, vous pourrez accéder aux options des pilotes de périphériques pris en charge par Linux. La réponse recommandée est 'N'.

L'option « I/O delaying target (EXPERIMENTAL) » permet de prendre en charge un volume virtuel qui retarde les opérations d'entrée/sortie sur les périphériques physiques. Cette fonctionnalité est principalement utile pour réaliser des tests, la réponse recommandée est 'N'.

XII-I-12 - Menu « Fusion MPT device support »

Les options « Fusion MPT ScsiHost drivers for SPI » , « Fusion MPT ScsiHost drivers for FC » et « Fusion MPT ScsiHost drivers for SAS » permettent d'activer la prise en charge de l'architecture Fusion respectivement pour les adaptateurs SCSI parallèles, à fibre optique et SCSI série. Cette architecture permet de multiplexer ces périphériques pour atteindre des taux de transfert encore plus élevés. Ce gestionnaire utilise une carte PCI compatible Fusion MPT pour effectuer les opérations d'entrée/sortie. Vous pouvez répondre 'Y' à cette question si vous disposez d'une telle carte, et régler le paramètre du nombre maximum de paramètres des opérations d'entrées / sorties multiples du driver avec l'option suivante. La réponse recommandée étant 'N'.

L'option « Maximum number of scatter gather entries (16 - 128) » permet d'indiquer le nombre maximum de segments de données utilisables dans le cadre d'entrées/sorties de données non contigües. Il est recommandé de conserver la valeur par défaut inchangée (128).

L'option « Fusion MPT misc device (ioctl) driver » permet d'activer un moyen de contrôle supplémentaire sur les adaptateurs MPT par l'intermédiaire du fichier spécial de périphérique de type caractère /dev/mptctl, dont le code majeur est 10 et le code mineur est 240. La réponse recommandée est 'N'.

L'option « Fusion MPT LAN driver » permet de prendre en charge les réseaux IP sur les adaptateurs Fusion à fibre optique. La réponse recommandée est 'N'.

L'option « Fusion MPT logging facility » permet d'activer des traces de fonctionnement sur les périphériques Fusion. La réponse recommandée est 'N'.

XII-I-13 - Menu « IEEE 1394 (FireWire) support (EXPERIMENTAL) »

L'option « IEEE 1394 (FireWire) support - alternative stack, EXPERIMENTAL » permet d'activer la gestion des cartes FireWire. Ces cartes fournissent l'accès à un bus de données extrêmement rapide, que l'on utilise généralement pour connecter des périphériques exigeant une bande passante très élevée, comme les caméras vidéo par exemple. Le pilote rendu accessible par cette option est un nouveau pilote de périphérique, qui n'est pas forcément totalement stabilisé encore. De plus, il ne prend pas encore en charge l'ensemble des fonctionnalités de l'ancien pilote (en particulier l'interface réseau au travers de l'interface FireWire). La réponse recommandée est donc 'N'.

L'option « Support for OHCI FireWire host controllers » permet d'activer la gestion des cartes FireWire compatible OHCI dans le nouveau pilote FireWire. La réponse recommandée est 'Y'.

L'option « Support for storage devices (SBP-2 protocol driver) » permet d'activer la gestion des périphériques de stockage dans le nouveau pilote FireWire, tels que les disques durs connectés par une prise FireWire. La réponse recommandée est 'Y'.

L'option « IEEE 1394 (FireWire) support (EXPERIMENTAL) » permet d'activer la gestion des cartes FireWire. Ce pilote de périphériques est le pilote utilisé historiquement dans Linux, il est donc plus stable que le nouveau pilote, mais appelé à être remplacé par ce dernier. La réponse recommandée est 'Y'.

L'option « Excessive debugging output » est réservée pour les développeurs de gestionnaires de périphériques. Elle permet de stocker sur disque toutes les informations transitant sur le bus FireWire, ce qui sature généralement les disques durs extrêmement rapidement. Il faut répondre par 'N' à cette question.

L'option « **Texas Instruments PCILynx support** » permet d'activer la gestion des cartes FireWire PCILynx de Texas Instruments. Cette option n'est disponible que si l'option « **I2C support** » du menu « **I2C support** » a été activée. La réponse recommandée est 'N'.

L'option « **OHCI-1394 support** » active la prise en charge des contrôleurs IEEE 1394 respectant les spécifications OHCI (il s'agit d'un standard de communication pour les contrôleurs). Ce pilote n'a été testé qu'avec un contrôleur de Texas Instruments, mais ce contrôleur est l'un des plus utilisés du marché. La réponse recommandée est 'Y'.

L'option « **OHCI-1394 Video support** » permet d'utiliser la capture vidéo des cartes FireWire. La réponse recommandée est 'Y'.

L'option « **SBP-2 support (Harddisks, etc.)** » permet d'activer la gestion des périphériques SBP-2, qui comprennent les disques durs et les lecteurs de DVD connectés sur les bus FireWire. La réponse recommandée est 'Y'.

L'option « **Enable replacement for physical DMA in SBP2** » permet d'utiliser un code de compatibilité pour que les gestionnaires périphériques SBP-2 puissent utiliser les accès directs à la mémoire sur les contrôleurs qui ne sont pas capables de les réaliser. La réponse recommandée est 'N'.

L'option « **IP over 1394** » permet d'utiliser une interface IEEE1394 comme une interface Ethernet. La réponse recommandée est 'N'.

L'option « **OHCI-DV I/O support (deprecated)** » permet d'activer l'envoi et la réception d'images DV par un port IEEE1394 au travers d'une interface simplifiée. Toutefois, les programmes vidéo n'utilisent plus cette fonctionnalité de nos jours et accèdent directement au port 1394 en mode brut (voir option suivante). Cette option n'est donc plus nécessaire, et la réponse recommandée est 'N'.

L'option « **Raw IEEE 1394 I/O support** » permet aux programmes de communiquer directement avec le matériel IEEE 1394. C'est en général le mode de fonctionnement désiré, aussi la réponse recommandée à cette question est-elle 'Y'.

XII-I-14 - Menu « **I2O device support** »

L'option « **I2O support** » permet d'activer la gestion des cartes d'entrée/sortie I2O. Ces cartes prennent en charge la gestion des entrées/sorties de manière uniforme, pour tous les périphériques matériels, et permettent donc d'écrire un gestionnaire de périphériques uniforme au niveau du système d'exploitation pour toute une classe de périphérique. Ce gestionnaire, nommé « **OSM** » (abréviation de « **Operating System Module** »), communique avec des pilotes spécifiques à chaque type de matériel. Ces pilotes peuvent alors ne faire aucune hypothèse sur le système d'exploitation utilisé. Ainsi, il est possible d'écrire des gestionnaires de périphériques communs à tous les systèmes d'exploitation. La réponse recommandée est 'N'.

L'option « **Enable LCT notification** » permet de prendre en charge les notifications relatives aux canaux de communication logiques du protocole I2O. Cette fonctionnalité est normalement nécessaire, mais peut être désactivée, car certains contrôleurs de SUN ne la prennent pas en charge. La réponse recommandée est 'Y', sauf si vous êtes dans ce cas.

L'option « **Enable Adaptec extensions** » permet de prendre en charge les extensions nécessaires à l'utilisation des contrôleurs RAID de marque Adaptec. Cette option vous donnera également accès à l'option « **Enable 64-bit DMA** » permettant d'activer les accès directs à la mémoire en 64 bits sur ces adaptateurs. La réponse recommandée est 'Y' si vous disposez de ce type de périphérique.

L'option « **I2O Configuration support** » permet d'activer l'interface de configuration des périphériques I2O. Cette option est en particulier nécessaire pour configurer les périphériques RAID sur les contrôleurs de marque Adaptec. Cette option vous donnera également accès à l'option « **Enable IOCTLs (OBSOLETE)** », qui permet de prendre en charge des commandes de configuration des périphériques obsolètes, à titre de compatibilité. La réponse recommandée est 'Y' si vous disposez de ce type de périphérique.

L'option « I2O Bus Adapter OSM » permet de prendre en charge le gestionnaire de périphériques pour le bus I2O lui-même. Cela permet de demander au bus de lister les périphériques qui y sont connectés, en particulier pour détecter les nouveaux périphériques. La réponse recommandée est 'Y'.

L'option « I2O Block OSM » permet de prendre en charge le gestionnaire de périphériques pour les périphériques RAID. La réponse recommandée est 'Y'.

L'option « I2O SCSI OSM » permet de prendre en charge le gestionnaire de périphériques SCSI ou FibreChannel. Ce gestionnaire vous permettra également d'accéder de manière unitaire aux disques RAID. La réponse recommandée est 'N'.

L'option « I2O /proc support » permet d'exporter des informations relatives aux périphériques I2O dans le système de fichiers virtuel /proc/. La réponse recommandée est 'N'.

XII-I-15 - Configuration des interfaces réseau

L'option « Network device support » permet d'activer la gestion des diverses interfaces réseau que Linux peut prendre en charge. On notera que les choix faits pour les protocoles de communication dans le menu « Networking options » sont nécessaires (la plupart des composants des systèmes Unix utilisent les fonctions réseau du système pour communiquer), mais non suffisants pour accéder au monde extérieur. Si l'on n'active pas cette option, seuls les processus de la machine locale seront accessibles (par l'intermédiaire de l'interface réseau « loopback »). Pour accéder aux autres ordinateurs, que ce soit par une carte réseau, un câble parallèle ou série, ou par modem via un fournisseur d'accès à Internet, il faut en plus répondre 'Y' à cette question. La réponse recommandée est 'Y'.

L'option « Netdevice multiple hardware queue support » permet de prendre en charge plusieurs files de transmission pour les paquets sortants, sur les adaptateurs qui fournissent cette fonctionnalité. La réponse recommandée est 'N'.

L'option « Intermediate Functional Block support » permet d'activer la prise en charge d'une interface réseau virtuelle qui se positionne entre l'interface réseau physique et les clients afin de réaliser des statistiques sur le trafic réseau. La réponse recommandée est 'N'.

L'option « Dummy net driver support » permet d'activer la fonctionnalité de réseau virtuel. Cette fonctionnalité fournit une interface virtuelle sur un réseau ne contenant qu'une seule machine. Cette interface peut être utilisée pour faire croire à des programmes réseau que la machine est effectivement connectée à un réseau. La réponse recommandée est 'N'.

L'option « Bonding driver support » permet de réunir plusieurs connexions Ethernet entre deux machines pour simuler une connexion unique, dont la bande passante est la somme des bandes passantes des connexions ainsi regroupées. La réponse recommandée est 'N'.

L'option « MAC-VLAN support (EXPERIMENTAL) » permet de créer plusieurs interfaces virtuelles afin de rediriger les paquets destinés à une adresse physique vers une interface spécifique. La réponse recommandée est 'N'.

L'option « EQL (serial line load balancing) support » permet d'activer la gestion de connexions multiples sur plusieurs connexions série (par exemple par l'intermédiaire de deux modems). Cette fonctionnalité nécessite également le support de cette fonctionnalité du côté de la machine distante. La réponse recommandée est 'N'.

L'option « Universal TUN/TAP device driver support » permet d'activer la gestion d'interfaces réseau virtuelles tunX et tapX (où 'X' est le numéro de chaque interface), dont les données peuvent être lues et écrites directement par des applications normales, par l'intermédiaire de fichiers spéciaux de périphériques /dev/tunX et /dev/tapX. Les interfaces de type tun se comportent exactement comme des interfaces réseau point à point (elles ne permettent donc de communiquer qu'avec une seule autre machine), alors que les interfaces de type tap simulent le fonctionnement d'une carte Ethernet classique. Cette fonctionnalité permet donc aux programmes classiques d'accéder aux paquets routés vers ces interfaces, et de recevoir et d'envoyer des données brutes sur le réseau. Les interfaces tunX et tapX sont créées dynamiquement, lorsque les programmes qui désirent les utiliser s'enregistrent au niveau du système

en ouvrant le fichier spécial de périphérique `/dev/net/tun`, de type caractère et de codes majeur et mineur égaux respectivement à 10 et 200. La réponse recommandée est 'N'.

L'option « General Instruments Surfboard 1000 » permet de prendre en charge les cartes modem Surfboard 1000 utilisées pour accéder à Internet via les opérateurs de câble. La réponse recommandée est 'N'.

L'option « Xen network device frontend driver » permet de prendre en charge les interfaces réseau pour les machines virtuelles Xen. Ce pilote doit être activé pour les noyaux des systèmes fonctionnant dans les machines virtuelles Xen. La réponse recommandée est 'N'.

L'option « FDDI driver support » permet d'activer la gestion des cartes Ethernet FDDI. Les options qui suivent permettent de sélectionner les gestionnaires de périphériques pour les cartes FDDI gérées par Linux. La réponse recommandée est 'N'.

L'option « HIPPI driver support (EXPERIMENTAL) » permet la gestion des cartes HIPPI. Les options qui suivent permettent de sélectionner les gestionnaires de périphériques pour les cartes de ce type. La réponse recommandée est 'N'.

L'option « PLIP (parallel port) support » permet d'activer les connexions par port parallèle. Ce type de connexion peut être utilisé pour transférer des fichiers par un câble parallèle entre deux ordinateurs. La réponse recommandée est 'N'.

L'option « PPP (point-to-point protocol) support » permet d'activer la gestion des connexions PPP. Ces connexions sont utilisées pour accéder à Internet via un modem, ou pour établir une connexion de manière plus générale via un câble série ou tout autre moyen de communication existant. Il est conseillé d'activer le support de ce type de connexion, à moins que vous soyez sûr de ne jamais vous connecter à Internet. La réponse recommandée ici est 'Y'.

L'option « PPP multilink support (EXPERIMENTAL) » permet d'utiliser plusieurs lignes PPP pour augmenter la bande passante en les agrégeant. Par exemple, si vous disposez de plusieurs lignes téléphoniques, vous pouvez vous connecter plusieurs fois à votre fournisseur d'accès et accroître ainsi votre bande passante. Bien entendu, pour que cette technique fonctionne, il faut que le fournisseur d'accès l'autorise. La réponse recommandée est 'N'.

L'option « PPP filtering » permet d'autoriser le filtrage des paquets traversant les interfaces PPP existantes dans le système. Cette fonctionnalité peut être utile pour réaliser un firewall ou pour effectuer des actions spécifiques lorsque certains paquets sont reçus. La réponse recommandée est 'N'.

L'option « PPP support for async serial ports » permet de prendre en charge les communications PPP sur les câbles série classiques. C'est en général l'option qui convient lorsqu'on veut se connecter à Internet à l'aide d'un modem classique. Notez que cette option n'est pas utilisable pour les connexions à l'aide d'un modem Numéris. La réponse recommandée est 'Y'.

L'option « PPP support for sync tty ports » permet de prendre en charge les communications PPP avec les adaptateurs synchrones, tels que les cartes SyncLink. La réponse recommandée est 'N'.

L'option « PPP Deflate compression » permet de prendre en charge l'algorithme de compression Deflate (le même algorithme que celui utilisé par le programme gzip) pour compresser les données transmises sur les connexions PPP. La réponse recommandée est 'Y'.

L'option « PPP BSD-Compress compression » permet de prendre en charge l'algorithme de compression LZW pour compresser les données transmises sur les connexions PPP. Cet algorithme est soumis à un brevet de logiciel et son utilisation est supposée être soumise à paiement de droits. Cependant, les brevets de logiciel sont encore illégaux en Europe, et il est tout à fait légal de l'utiliser. Les taux de compression obtenus sont de toute manière inférieurs à ceux de la méthode Deflate, aussi la réponse recommandée est-elle 'N'.

L'option « **PPP MPPE compression (encryption) (EXPERIMENTAL)** » permet de prendre en charge l'algorithme de compression MPPE utilisé par le protocole de communication point à point de Microsoft. La réponse recommandée est 'N'.

L'option « **PPP over Ethernet (EXPERIMENTAL)** » permet de réaliser des connexions point à point sur un réseau de type Ethernet. Ce protocole est également souvent utilisé par les fournisseurs d'accès pour transférer les données via les modems ADSL. La réponse recommandée est donc 'Y'.

L'option « **PPP over ATM** » permet de réaliser des connexions point à point sur un réseau ATM. La réponse recommandée est 'N'.

L'option « **PPP over L2TP (EXPERIMENTAL)** » permet de réaliser des connexions point à point sur un réseau L2TP. La réponse recommandée est 'N'.

L'option « **SLIP (serial line) support** » permet de connecter deux ordinateurs par une ligne série ou par un modem. Les options suivantes permettent de fixer les paramètres des connexions SLIP. Le protocole SLIP est en train de tomber en désuétude et est remplacé par PPP. La réponse recommandée à cette question est donc 'N'.

L'option « **CSLIP compressed headers** » permet d'activer la compression des en-têtes IP pour le protocole SLIP. La réponse recommandée est 'N'.

L'option « **Keepalive and linefill** » permet d'activer la surveillance de la connexion SLIP. Cette option est utilisée sur les connexions SLIP passant par des lignes analogiques de mauvaise qualité. La réponse recommandée est 'N'.

L'option « **Six bit SLIP encapsulation** » permet de faire en sorte que le protocole SLIP n'envoie que des données codées sur 6 bits. Cela permet d'améliorer la qualité des transmissions sur les réseaux peu fiables, ou qui ne peuvent pas transférer plus de 7 bits de données. La réponse recommandée est 'N'.

L'option « **Fibre Channel driver support** » permet d'activer la gestion des adaptateurs Fiber Channel. L'option qui suit donne la possibilité de sélectionner le gestionnaire de périphériques pour les cartes Interphase à base de chipset Tachyon. La réponse recommandée est 'N'.

L'option « **Traffic Shaper (EXPERIMENTAL)** » permet d'activer la possibilité de contrôler le débit maximal de données à travers une interface réseau. La réponse recommandée est 'N'.

L'option « **Network console logging support (EXPERIMENTAL)** » permet d'activer la possibilité d'envoyer les messages générés par le noyau vers le réseau, plutôt que de les envoyer à la console. La réponse recommandée est 'N'.

L'option « **Netpoll traffic trapping** » n'est pas documentée et ne sera pas décrite plus en détail ici. La réponse recommandée est 'N'.

XII-I-15-a - Sous-menu « ARCnet devices »

L'option « **ARCnet support** » permet d'activer la gestion des cartes de type ARCnet. Les options qui suivent correspondent aux différents pilotes pour les différentes cartes de ce type. Si vous avez une carte de ce type, répondez par 'M' ou 'Y' à cette question et choisissez le pilote correspondant à votre matériel dans la suite du menu. La réponse recommandée est 'N'.

XII-I-15-b - Sous-menu « PHY device support »

L'option « **PHY Device support and infrastructure** » permet d'activer la des périphériques PHY auxquels les contrôleurs Ethernet sont souvent attachés. Ces périphériques permettent de déterminer l'état du lien physique (présence du câble, mode de communication, croisement des fils, vitesse, etc.). La réponse recommandée est 'Y'.

Les options qui suivent correspondent aux différents pilotes pour les périphériques PHY de certaines cartes réseau. Si vous avez une carte de ce type, répondez par 'M' ou 'Y' à la question correspondante. La réponse recommandée est 'N'.

XII-I-15-c - Sous-menu « Ethernet (10 or 100Mbit) »

L'option « Ethernet (10 or 100Mbit) » permet d'activer la gestion des cartes de type Ethernet. Les options qui suivent correspondent aux différents gestionnaires de périphériques pour les différentes cartes de ce type. Si vous avez une carte de ce type, répondez par 'M' ou 'Y' à cette question et choisissez le gestionnaire de périphériques correspondant à votre matériel dans la suite du menu. La réponse recommandée est 'N'. Certaines cartes utilisent une interface commune pour le transmetteur, dont on peut activer le support grâce à l'option « Generic Media Independent Interface device support », qu'il est donc conseillé d'activer. La plupart des cartes réseau bon marché sont compatibles NE2000, celles fournies sur les cartes mères sont toutefois souvent différentes. Il existe deux types de pilotes pour les cartes réseau, selon qu'elles sont ISA ou non. Les pilotes pour les cartes ISA peuvent être choisis en activant les options « Other ISA cards ». Les pilotes pour les cartes PCI peuvent être sélectionnés en activant les options « EISA, VLB, PCI and on board controllers ».

XII-I-15-d - Sous-menu « Ethernet (1000 Mbit) »

Les options disponibles dans ce menu permettent de choisir les gestionnaires de périphériques à utiliser pour les cartes Ethernet Gigabit. Choisissez l'option qui correspond à votre matériel si vous en disposez d'une.

XII-I-15-e - Sous-menu « Ethernet (10000 Mbit) »

Les options disponibles dans ce menu permettent de choisir les gestionnaires de périphériques à utiliser pour les cartes Ethernet 10 Gigabits. Choisissez l'option qui correspond à votre matériel si vous en disposez d'une.

XII-I-15-f - Sous-menu « Token Ring devices (depends on LLC=y) »

Ce menu permet d'activer les gestionnaires de périphériques pour les cartes Token Ring. L'option « Token Ring driver support » permet d'activer la gestion des réseaux Token Ring. Les options qui suivent permettent de choisir les gestionnaires de périphériques pour ces adaptateurs. La réponse recommandée est 'N'.

XII-I-15-g - Sous-menu « Wireless LAN »

Les options de ce menu permettent d'activer la gestion des connexions sans fil Wi-Fi. Les options qui suivent permettent de choisir les gestionnaires de périphériques disponibles. Cette option est également nécessaire pour les gestionnaires de périphériques Wi-Fi non intégrés au noyau. Certains périphériques nécessitent également les fonctionnalités de l'infrastructure Wi-Fi activable avec les options « Generic IEEE 802.11 Networking Stack » du menu « Networking ». La réponse recommandée est 'Y' si vous disposez d'une interface réseau sans fil, et 'N' dans le cas contraire.

XII-I-15-h - Sous-menu « USB Network Adapters »

Ce sous-menu permet de prendre en charge les périphériques réseau connectés via un port USB. En particulier, l'option « Multipurpose USB Networking Framework » donne accès aux options de configuration des fonctions de connexion entre deux ordinateurs ou entre un ordinateur et un appareil USB intelligent, en utilisant un câble USB de transfert de données. Les interfaces réseau seront nommées usb0, usb1, etc. Elle donne également accès à l'option « CDC Ethernet support (smart devices such as cable modems) », qui doit être utilisée pour accéder à la plupart des périphériques ADSL qui simulent une interface Ethernet via le port USB. La réponse recommandée est 'N'.

XII-I-15-i - Sous-menu « PCMCIA network device support »

L'option « PCMCIA network device support » permet d'activer la gestion des cartes réseau au format PCMCIA. Les options qui suivent correspondent aux divers gestionnaires de périphériques pris en charge par Linux. La réponse recommandée est 'N'.

XII-I-15-j - Sous-menu « Wan interfaces »

L'option « Wan interfaces support » permet de prendre en charge les gestionnaires de périphériques pour les adaptateurs WAN (« Wide Area Network »). Cette option vous donnera accès aux options permettant de sélectionner les gestionnaires de périphériques pour ces adaptateurs. La réponse recommandée est 'N'.

L'option « WAN router drivers » permet d'activer les options de menu suivantes, qui donnent la possibilité de faire la configuration des adaptateurs WAN. Cette option n'est disponible que si l'option « Wan router » du menu de configuration des options générales du réseau a été activée. La réponse recommandée est 'N'.

L'option « LAPB over Ethernet driver » permet d'activer un gestionnaire de périphériques pour une carte X.25 virtuelle. La réponse recommandée est 'N'.

L'option « X.25 async driver » permet d'activer la gestion du protocole X.25 sur une ligne série normale ou sur un modem. La réponse recommandée est 'N'.

XII-I-15-k - Sous-menu « ATM drivers »

L'option « Dummy ATM driver » permet de fournir un pilote de périphérique ATM virtuel. Ce type pilote n'est utile qu'à des fins de test et de développement. La réponse recommandée est 'N'.

L'option « ATM over TCP » permet d'encapsuler ATM sur un réseau TCP/IP. Ce type d'encapsulation n'est utile qu'à des fins de test et de développement. La réponse recommandée est 'N'.

L'option « Efficient Networks Speedstream 3010 » permet de prendre en charge les cartes ATM Speedstream 3010 d'Efficient Networks. La réponse recommandée est 'N'.

L'option « Efficient Networks ENI155P » active la gestion des cartes ATM ENI155P d'Efficient Networks et Power155 de SMC. Les options qui suivent permettent de configurer ce gestionnaire de périphériques. La réponse recommandée est 'N'.

L'option « Fujitsu FireStream (FS50/FS155) » active la gestion des cartes ATM FireStream de Fujitsu. La réponse recommandée est 'N'.

L'option « ZeitNet ZN1221/ZN1225 » active la gestion des cartes ATM ZN1221 et ZN1225 de ZeitNet. Les options qui suivent permettent de configurer ce gestionnaire de périphériques. La réponse recommandée est 'N'.

L'option « IDT 77201 (NICStAR) (ForeRunnerLE) » active la gestion des cartes ATM basées sur le chipset NICStAR 77201. Les options qui suivent permettent de configurer ce gestionnaire de périphériques. La réponse recommandée est 'N'.

L'option « IDT 77252 (NICStAR II) » active la gestion des cartes ATM basées sur le chipset NICStAR 77252. Les options qui suivent permettent de configurer ce gestionnaire de périphériques. La réponse recommandée est 'N'.

L'option « Madge Ambassador (Collage PCI 155 Server) » active la gestion des cartes ATM basées sur le chipset ATMizer. L'option qui suit permet d'activer les informations de débogage de ce gestionnaire de périphériques. La réponse recommandée est 'N'.

L'option « Madge Horizon [Ultra] (Collage PCI 25 and Collage PCI 155 Client) » active la gestion des cartes ATM basées sur le chipset Horizon. L'option qui suit permet d'activer les informations de débogage de ce gestionnaire de périphériques. La réponse recommandée est 'N'.

L'option « Interphase ATM PCI x575/x525/x531 » active la gestion des cartes ATM ChipSAR d'Interphase. L'option qui suit permet d'activer les informations de débogage de ce gestionnaire de périphériques. La réponse recommandée est 'N'.

L'option « FORE Systems 200E-series » active la gestion des cartes ATM FORE Systems 200E. Les options qui suivent permettent de configurer ce gestionnaire de périphériques. La réponse recommandée est 'N'.

L'option « FORERunner HE Series » active la gestion des cartes ATM ForeRunner HE. Les options qui suivent permettent de configurer ce gestionnaire de périphériques. La réponse recommandée est 'N'.

XII-I-16 - Menu « ISDN subsystem »

L'option « ISDN support » permet d'activer la gestion des interfaces « ISDN » (« Numéris » en France). Si vous disposez d'une telle interface, la réponse recommandée est 'M', sinon, c'est 'N'.

XII-I-16-a - Ancienne interface ISDN4Linux

Le sous-menu « Old ISDN4Linux » permet d'activer l'ancienne interface de programmation ISDN4Linux. Si vous y activez l'option « Old ISDN4Linux (obsolete) », vous aurez accès des des options complémentaires permettant de configurer cette interface. La réponse recommandée est 'N', car on utilisera de préférence la nouvelle interface à présent.

L'option « Support synchronous PPP » permet d'utiliser une version de PPP qui ne gère pas les synchronisations dans la communication entre les deux ordinateurs. Cette fonctionnalité n'est en effet plus nécessaire avec ISDN, puisque c'est un protocole numérique. La réponse recommandée est 'Y'.

L'option « Use VJ-compression with synchronous PPP » permet d'utiliser l'algorithme de Van Jacobson pour la compression des en-têtes IP dans le protocole PPP synchrone. La réponse recommandée est 'Y'.

L'option « Support generic MP (RFC 1717) » permet de regrouper plusieurs connexions ISDN avec le protocole PPP synchrone, et ce afin d'accroître la bande passante. La réponse recommandée est 'N'.

L'option « Filtering for synchronous PPP » active les fonctionnalités de filtrage des paquets pour les interfaces IPPP. La réponse recommandée est 'N'.

L'option « Support BSD compression » active la gestion de la compression de données avec l'algorithme BSD. La réponse recommandée est 'N'.

L'option « Support audio via ISDN » permet d'activer la gestion des commandes vocales dans l'émulateur de modem du sous-système ISDN. La réponse recommandée est 'N'.

L'option « Support AT-Fax Class 1 and 2 commands » permet d'activer la gestion des commandes vocales des FAX. La réponse recommandée est 'N'.

L'option « X.25 PLP on top of ISDN » permet de prendre en charge le protocole X.25 au-dessus d'ISDN. La réponse recommandée est 'N'.

L'option « isdnloop support » du sous-menu « ISDN feature submodules » permet d'activer la gestion d'une interface ISDN virtuelle, afin de tester la configuration sans effectuer d'appel réel. La réponse recommandée est 'N'. L'option

« `Support isdn diversion services` » quant à elle permet d'activer la gestion de services ISDN supplémentaires. La réponse recommandée est également 'N'.

XII-I-16-a-i - Gestionnaires de périphériques ISDN4Linux

Les sous-menus « `Passive cards` » et « `Active cards` » permettent d'inclure les drivers pour la plupart des cartes ISDN qui supportent l'ancienne interface de programmation ISDN de Linux. Ces options sont placées dans les sous-menus en fonction de la nature des cartes utilisées. Le sous-menu « `Passive cards` » permet d'accéder aux drivers pour les cartes ISDN passives, dont les cartes HiSax, AVM, Sedlbauer et ELSA. Le sous-menu « `Active cards` » permet quant à lui d'accéder aux cartes ISDN actives, dont les cartes Spellcaster, IBM, Eicon, Auverttech et Hypercope. Le sous-menu « `Siemens Gigaset` » donne accès aux pilotes pour les cartes Siemens Gigaset. Vous devez bien entendu choisir les options qui correspondent à votre matériel.

XII-I-16-b - Interface CAPI

La nouvelle interface ISDN pour Linux est basée sur le standard CAPI (« `Common ISDN Application Programming Interface` ») et peut être activée à l'aide de l'option « `CAPI2.0 support` ». On préférera cette interface à l'ancienne, aussi la réponse recommandée est-elle 'Y' si l'on désire utiliser les lignes ISDN sous Linux.

L'option « `Verbose reason code reporting (kernel size +=7K)` » permet d'obtenir des messages détaillés sur les raisons des déconnexions provoquées par le pilote CAPI. La réponse recommandée est 'N'.

L'option « `CAPI trace support` » permet d'obtenir des messages de trace détaillés sur le fonctionnement du pilote CAPI. La réponse recommandée est 'N'.

L'option « `CAPI2.0 Middleware support (EXPERIMENTAL)` » permet d'étendre l'interface CAPI pour permettre le transfert des connexions établies via l'interface CAPI vers un terminal classique. La réponse recommandée est 'N'.

L'option « `CAPI2.0 /dev/capi support` » permet de donner l'accès à l'interface CAPI 2.0 aux applications par l'intermédiaire d'un fichier spécial de périphérique `/dev/capi20`. L'utilisation de ce fichier spécial de périphérique nécessite l'installation d'une bibliothèque de fonctions dédiée. La réponse recommandée est 'N'.

L'option « `CAPI2.0 filesystem support` » permet d'activer le support d'un système de fichiers virtuel semblable au système de fichiers `/dev/pts/`, dans lequel les terminaux créés par les fonctionnalités CAPI apparaîtront. Cette option est nécessaire pour utiliser le démon `pppd` avec son module `pppdcaplugin`. La réponse recommandée est 'N'.

L'option « `CAPI2.0 capidrv interface support` » permet d'utiliser les cartes compatibles CAPI avec l'interface de programmation ISDN classique. La réponse recommandée est 'N'.

Les options qui suivent permettent de prendre en charge les cartes AVM et Eicon. La réponse recommandée est 'N'.

XII-I-17 - Menu « Telephony Support »

L'option « `Linux telephony support` » permet d'activer la gestion des cartes téléphoniques. Ces cartes permettent de transférer les communications téléphoniques sur des supports numériques, par exemple en encapsulant les informations sonores sur un réseau TCP/IP. Cette option n'a rien à voir avec un modem, qui fait exactement l'inverse (transfert des données informatiques sur un réseau téléphonique analogique). La réponse recommandée est 'N'.

L'option « `QuickNet Internet LineJack/PhoneJack support` » active la prise en charge des cartes téléphoniques Quicknet. Vous pouvez répondre 'Y' ou 'M' si vous possédez ce matériel, sinon la réponse recommandée est 'N'.

L'option « `QuickNet Internet LineJack/PhoneJack PCMCIA support` » active la prise en charge des cartes téléphoniques Quicknet connectées sur port PCMCIA. Vous pouvez répondre 'Y' ou 'M' si vous possédez ce matériel, sinon la réponse recommandée est 'N'.

XII-I-18 - Menu « Input device support »

Ce menu contient les options permettant de configurer les périphériques d'entrée de Linux, dont le clavier et la souris.

L'option « **Generic input layer (needed for keyboard, mouse...)** » permet d'activer la prise en charge des pilotes des périphériques d'entrée sous Linux. Cela comprend le clavier et les souris, aussi la réponse recommandée est-elle 'Y'.

L'option « **Support for memoryless force-feedback devices** » permet de prendre en charge les périphériques à rétroaction sans mémoire, tels que les joysticks WingMan Force 3D de Logitech ou FireStorm Dual Power 2 de ThrustMaster. La réponse recommandée est 'N'.

L'option « **Polled input device skeleton** » permet de prendre en charge les fonctions de scrutation périodiques des périphériques d'entrée auxquelles certains pilotes de périphériques peuvent avoir recours. Les pilotes du noyau activent automatiquement cette option s'ils en ont besoin, mais dans le cas contraire ces fonctionnalités ne sont pas obligatoirement présentes dans le noyau et donc ne sont pas à disposition des pilotes additionnels non intégrés au noyau. La réponse recommandée est donc 'Y' afin de fournir ces fonctionnalités aux pilotes externes éventuels, même si aucun pilote du noyau n'en a besoin.

L'option « **Mouse interface** » permet d'offrir un accès aux périphériques de type souris aux applications utilisateur, par l'intermédiaire de fichiers spéciaux de périphériques `/dev/input/mouseX` et `/dev/input/mice`. Cette option vous donnera également accès à la sous-option « **Provide legacy /dev/psaux device** », qui permet de prendre en charge le fichier spécial de périphérique `/dev/psaux` ordinairement utilisé pour les souris PS/2. La réponse recommandée est 'Y' pour ces deux questions.

Les options « **Horizontal screen resolution** » et « **Vertical screen resolution** » permettent de définir la résolution horizontale et verticale de l'écran, afin de permettre l'utilisation d'une tablette de digitalisation USB comme une souris. Ces deux données permettent de faire la conversion entre les coordonnées de la tablette et les coordonnées de l'écran. Si vous disposez d'une telle tablette, vous devez entrer ici la résolution du mode graphique que vous utilisez prioritairement. Dans les autres cas, les valeurs de ces options ne sont pas utilisées.

L'option « **Joystick interface** » active la prise en charge des joysticks et permet aux applications d'y accéder au travers des fichiers spéciaux de périphériques `/dev/input/jsN`, où N est le numéro du joystick, et dont les numéros de codes majeur et mineurs sont respectivement 13 et 0 à 31. La réponse recommandée est 'N'.

L'option « **Touchscreen interface** » active la prise en charge des périphériques de pointage absolus qui gèrent le protocole Touchscreen de Compaq. Cette option vous donnera également la possibilité de fixer la résolution du mode graphique que vous utilisez, pour permettre au gestionnaire de faire la correspondance entre la position sur le TouchScreen et les pixels de l'écran. La réponse recommandée est 'N'.

L'option « **Event interface support** » active la gestion des fichiers spéciaux `/dev/input/eventN` (où N est le numéro du périphérique), qui permettent de lire les événements provenant des périphériques d'entrée de manière générique. Ces fichiers peuvent être utilisés par plusieurs périphériques distincts, et permettent de connecter plusieurs périphériques d'entrée sur la même machine en les gérant de manière uniforme. La réponse recommandée est-elle 'Y'.

L'option « **Event debugging** » permet de générer des traces sur tous les événements qui se produisent au niveau des périphériques d'entrée. Ces traces ne sont utiles qu'aux développeurs des gestionnaires de périphériques d'entrée, et peuvent générer de grandes quantités d'informations dans les fichiers de traces. De plus, elles contiendront également les mots de passe et constituent donc un trou de sécurité important. La réponse recommandée est donc 'N'.

Les options qui suivent permettent d'activer des gestionnaires complémentaires pour les périphériques d'entrée. Vous trouverez en particulier les gestionnaires de périphériques pour les souris PS/2 et sérieelles, ainsi que les gestionnaires pour les Joysticks et les Touchscreen. Vous devez activer les options correspondantes à votre matériel. Notez également que la prise en charge des claviers et des souris USB doit se configurer au niveau du menu « **USB support** » et ne peut être faite ici, même si les options précédentes restent nécessaires.

L'option « **Miscellaneous devices** » permet d'accéder aux options de configuration de gestionnaires de périphériques complémentaires, tels que celui permettant de contrôler le haut-parleur du PC (option « **PC Speaker support** ») et celui qui permet de relayer la gestion des périphériques d'entrées à des programmes utilisateurs (option « **User level driver support** »). La réponse recommandée est 'Y' pour l'option « **PC Speaker support** », et 'N' pour l'option « **User level driver support** ».

XII-I-18-a - Sous-menu « **Hardware I/O ports** »

Ce sous-menu vous permettra de configurer les ports d'entrée/sortie matériel classiques.

L'option « **Serial I/O support** » permet de prendre en charge les périphériques d'entrée qui utilisent des communications avec l'ordinateur de type série. Cela comprend les claviers et les souris, aussi la réponse recommandée est-elle 'Y'.

L'option « **i8042 PC Keyboard controller** » permet de prendre en charge les contrôleurs de claviers standards. Si votre clavier n'est pas USB, vous devez activer cette option. La réponse recommandée est 'Y'.

L'option « **Serial port line discipline** » permet de prendre en charge les périphériques connectés sur un port série RS232. C'est en particulier le cas de vieilles souris sérielles et de joysticks. La réponse recommandée est 'N'.

L'option « **ct82c710 Aux port controller** » permet de prendre en charge les souris connectées au port « **QuickPort** » des portables TravelMate de Texas Instrument. La réponse recommandée est 'N'.

L'option « **Parallel port keyboard adapter** » permet de prendre en charge les claviers ou souris connectés sur un adaptateur pour port parallèle. La réponse recommandée est 'N'.

L'option « **PCI PS/2 keyboard and PS/2 mouse controller** » permet de prendre en charge les claviers et les souris PS/2 pour connecteurs PCI externes des stations d'attache des portables. Cette option n'est pas utile pour les ordinateurs normaux disposant de claviers ou de souris PS/2. La réponse recommandée est 'N'.

L'option « **PS/2 driver library** » permet d'inclure dans le noyau les fonctions de gestion du port PS/2, pour qu'elles soient accessibles d'un pilote externe de périphérique PS/2. La réponse recommandée est 'Y'.

L'option « **Raw access to serio ports** » permet l'accès brut aux périphériques auxiliaires connectés sur les ports PS/2 via des fichiers spéciaux de périphériques de numéro majeur égal à 10. Le premier de ces fichiers correspond au fichier spécial de périphérique `/dev/psaux` pour des raisons historiques. La réponse recommandée est 'N'.

L'option « **Gameport support** » active la gestion des ports pour Joystick classiques. Cette option vous donnera accès aux différents gestionnaires de périphériques pour ce type de port. La réponse recommandée est 'N'.

XII-I-19 - Menu « **Character devices** »

L'option « **Virtual terminal** » permet d'activer la gestion des terminaux virtuels multiples utilisés pour gérer plusieurs terminaux avec le clavier et l'écran local. Ces terminaux sont généralement accessibles avec les raccourcis CTRL+Alt+Fn. Ils sont très souvent utilisés, et il est vivement recommandé de répondre 'Y' à cette option. Si vous le faites, vous pourrez accéder à l'option « **Support for console on virtual terminal** », qui permet d'autoriser l'émission des messages de la console du noyau sur les terminaux virtuels. Encore une fois, cette fonctionnalité est très souvent utilisée, et il est recommandé de répondre 'Y'. L'option « **Support for binding and unbinding console drivers** » permet d'activer le support de la sélection de la console utilisée pour les terminaux virtuels, lorsque plusieurs consoles sont disponibles sur la machine. La réponse recommandée est 'N'.

L'option « **Non-standard serial port support** » permet d'activer les options correspondantes à des ports série non standards. Les sous-options de cette option permettent de configurer les gestionnaires de périphériques

correspondants. La réponse recommandée est 'N'. Les gestionnaires de périphériques pour les ports série standards sont accessibles via le sous-menu « Serial drivers », décrit plus bas.

L'option « Unix98 PTY support » permet d'activer les pseudoterminaux compatibles Unix 98. Les pseudoterminaux sont des composants logiciels permettant d'émuler des terminaux. Il y a deux composants par pseudoterminal : la partie maître est le programme qui cherche à accéder au pseudoterminal, et la partie esclave est le programme qui simule le terminal physique. Les composants maîtres peuvent ouvrir le pseudoterminal par l'intermédiaire des fichiers spéciaux `/dev/ptyxx`. Les composants esclaves doivent ouvrir le fichier spécial `/dev/ptmx` afin d'obtenir un numéro `n` de pseudoterminal à utiliser, puis ils ouvrent le fichier `/dev/pts/n`. Ce dernier fichier spécial est créé à la volée par le noyau, grâce à un système de fichiers virtuel. Pour que ce mécanisme fonctionne, il faut donc que le système de fichiers `/dev/pts/` ait été également activé dans les options de configuration des systèmes de fichiers, et qu'il soit monté (voir la configuration du fichier `/etc/fstab`). La réponse recommandée est 'Y'.

L'option « Legacy (BSD) PTY support » permet d'activer la gestion des pseudoterminaux classiques tels qu'ils sont utilisés sur les systèmes BSD. Ces pseudoterminaux sont simplement accédés via des fichiers spéciaux de périphériques `/dev/ttyxx`, à raison d'un fichier par pseudoterminal. Cette technique est obsolète depuis l'introduction des terminaux Unix98, que l'on peut activer avec l'option précédente. Toutefois, certains programmes peuvent ne pas avoir porté et ne pouvoir accéder que via cette ancienne interface. Dans ce cas, cette option devra être activée. Si vous répondez 'Y' à cette option, vous accéderez à l'option « Maximum number of legacy PTY in use », qui vous permettra de choisir le nombre de pseudoterminaux que le système gèrera. La valeur par défaut est 256. La réponse recommandée à cette option est 'N'.

L'option « Parallel printer support » permet d'activer la gestion des imprimantes connectées sur port parallèle. Le fait d'activer cette option n'empêche pas d'utiliser le port parallèle pour d'autres périphériques. Il est recommandé d'activer cette option sous forme de module parce qu'ainsi, les fonctionnalités d'impression ne seront chargées que lorsque cela sera nécessaire. La réponse recommandée est donc 'M'.

L'option « Support for console on line printer » permet de rediriger les messages destinés à la console sur une imprimante connectée au port parallèle. Ces messages sont alors imprimés au fil de l'eau. La réponse recommandée est 'N'.

L'option « Support for user-space parallel port device drivers » active la gestion du fichier spécial de périphérique `/dev/parport`, au travers duquel les programmes peuvent accéder de manière uniforme au port parallèle. Cette fonctionnalité n'est pas nécessaire pour utiliser une imprimante ou des périphériques ATAPI connectés sur port parallèle, aussi la réponse recommandée est-elle 'N'.

L'option « Texas Instruments parallel link cable support » permet de prendre en charge les connexions aux calculatrices Texas Instruments à l'aide d'un câble parallèle. La réponse recommandée est 'N'.

L'option « Xen Hypervisor Console support » permet de gérer la console virtuelle dans les machines virtuelles Xen. La réponse recommandée est 'N'.

L'option « Hardware Random Number Generator Core support » permet de prendre en charge les générateurs de nombres aléatoires matériels fournis sur les cartes mères récentes. Ces générateurs permettent d'obtenir des nombres aléatoires utilisés dans les algorithmes de chiffrement et de sécurité. Cette option vous donnera accès aux pilotes de périphériques pour les différents générateurs de nombres aléatoires supportés par Linux. La réponse recommandée est 'Y'.

L'option « `/dev/nvram` support » permet d'activer l'accès à la mémoire non volatile de l'horloge temps réel. Cette mémoire peut être accédée par l'intermédiaire du fichier spécial de périphérique `/dev/nvram`, de code majeur 10 et de code mineur 144. La réponse recommandée est 'Y'.

L'option « Enhanced Real Time Clock Support » permet d'activer l'accès aux compteurs de l'horloge temps réel par l'intermédiaire du fichier spécial de périphérique `/dev/rtc`, de code majeur 10 et de code mineur 135. Cette option doit être activée si vous disposez d'une machine multiprocesseur. La réponse recommandée est 'Y'.

L'option « **Generic /dev/rtc emulation** » permet de simuler l'interface /dev/rtc à l'horloge temps réel du système pour le cas où vous ne voudriez pas activer l'option précédente. L'option suivante permet de compléter l'émulation avec la prise en charge d'une opération complémentaire, qui peut être nécessaire pour certains programmes. La réponse recommandée est 'N'.

L'option « **Extended RTC operation** » permet de simuler une horloge temps réel plus évoluée que celle simulée par défaut grâce à l'option précédente. Certains programmes peuvent avoir besoin des fonctionnalités avancées fournies par ces horloges. La réponse recommandée est 'N'.

L'option « **Double Talk PC internal speech card support** » permet d'activer la gestion du synthétiseur de voix Double Talk. La réponse recommandée est 'N'.

L'option « **Siemens R3964 line discipline** » active la gestion des synchronisations entre les périphériques utilisant le protocole de communication R3964 de Siemens. La réponse recommandée est 'N'.

L'option « **Applicom intelligent fieldbus card support** » active la gestion des cartes Applicom « intelligent fieldbus ». La réponse recommandée est 'N'.

L'option « **Sony Vaio Programmable I/O Control Device support** » permet d'activer la gestion des contrôleurs d'entrée / sortie programmables des portables Sony Vaio. La réponse recommandée est 'N'.

L'option « **/dev/agpgart support (AGP support)** » active la gestion des transferts de données accélérés par le bus AGP pour les pilotes de cartes graphiques. Ces transferts peuvent être pilotés au travers du fichier spécial de périphérique /dev/agpgart, de code majeur 10 et de code mineur 175. Cette fonctionnalité n'est pas nécessaire pour le bon fonctionnement des cartes graphiques AGP, mais elle permet d'accroître les performances des pilotes 3D. Elle est également indispensable pour permettre l'accès au registre MTRR du processeur permettant de contrôler le bus AGP. La réponse recommandée est 'Y' si vous disposez d'une carte graphique AGP, et 'N' sinon. Les options qui suivent permettent de sélectionner le type de chipset utilisé par la carte mère, afin de déterminer la manière dont le port AGP fonctionne. Si vous avez activé cette fonctionnalité, vous devez choisir l'option correspondant à votre chipset également.

L'option « **Direct Rendering Manager (XFree86 4.1.0 and higher DRI support)** » permet d'activer le support de l'architecture DRI (abréviation de l'anglais « Direct Rendering Infrastructure ») de XFree86 ou de X.org. La version fournie avec le noyau 2.6 nécessite l'utilisation de XFree86 en version 4.1 ou plus, ou de X.org en version 6.7 ou plus. Vous devrez sélectionner le type de carte graphique installé sur le système à l'aide des options suivantes. La réponse recommandée est donc généralement 'N', sauf si vous disposez d'une des cartes graphiques listées. Dans ce cas, vous devriez certainement utiliser XFree86 4.4.0 ou X.org 6.7, afin de bénéficier des gestionnaires de périphériques les plus récents. Notez enfin que certaines cartes graphiques 3D, bien que parfaitement supportées sous Linux, n'utilisent pas les fonctionnalités DRI du noyau. C'est en particulier le cas pour toutes les cartes graphiques à base de puce NVidia, pour lesquelles il faut utiliser le module fourni avec le pilote de NVidia. Vous pouvez donc répondre 'N' à cette question si vous disposez d'une telle carte graphique.

L'option « **ACP Modem (Mwave) support** » permet d'activer la prise en charge des Winmodems ACP (également connu sous le nom Mwave). De tels modems sont intégrés en particulier dans les Thinkpad 600E, 600 et 770 d'IBM. La réponse recommandée est 'N'.

L'option « **NatSemi PC8736x GPIO Support** » permet d'autoriser l'accès depuis l'espace utilisateur aux ports d'entrée/sortie programmables des puces PC-8736x de National Semiconductor. La réponse recommandée est 'N'.

L'option « **NatSemi PC8736x GPIO Support** » permet d'inclure un module facilitant l'accès aux ports d'entrée/sortie des puces PC8736X depuis l'espace utilisateur. La réponse recommandée est 'N'.

L'option « **AMD CS5535/CS5536 GPIO (Geode Companion Device)** » permet d'autoriser l'accès depuis l'espace utilisateur aux ports d'entrée/sortie programmables des puces CS5535 et CS5536 Geode d'AMD. La réponse recommandée est 'N'.

L'option « RAW driver (/dev/raw/rawN) (OBSOLETE) » permet d'accéder directement, sans copie de données lors des opérations d'entrée/sortie, aux périphériques de type bloc. Cet accès se fait via des fichiers de périphériques spéciaux /dev/raw/ dont les numéros de périphériques majeurs et de mineurs sont identiques à ceux du fichier spécial de périphérique de type bloc ainsi manipulé. Cette option permet donc de réaliser des entrées / sorties de manière extrêmement performante. Comme cette fonctionnalité n'est utilisable que par des applications particulières (en général des grosses bases de données), la réponse recommandée est 'N'. De plus, cette option est obsolète, car il est à présent possible d'accéder aux périphériques de manière directe avec les interfaces de programmation standards du système.

L'option « Maximum number of RAW devices to support (1-8192) » permet d'indiquer le nombre maximum de périphériques bruts pris en charge par le noyau. La réponse recommandée est '256'.

L'option « HPET - High Precision Event Timer » permet d'activer la gestion des timers à haute précision de Linux. Ces timers seront accessibles via le fichier spécial de périphérique /dev/hpet. Cette option vous donnera également accès aux options « HPET Control RTC IRQ » et « Allow mmap of HPET », qui permettent respectivement de désactiver la gestion de l'interruption du timer système par le timer et de désactiver la projection en mémoire de ce fichier. Cette dernière option doit être désactivée pour certains types de matériel, car ceux-ci exposent trop d'informations à l'utilisateur via cette interface. La réponse recommandée est 'N'.

L'option « Hangcheck timer » permet d'activer un chien de garde permettant de déterminer si le système est bloqué depuis un temps trop long et, si tel est le cas, de le redémarrer ou afficher un message de trace. La réponse recommandée est 'N'.

L'option « Telecom clock driver for MPBL0010 ATCA SBC » permet d'activer une interface spécifique aux ordinateurs MPBL0010 ATCA permettant d'accéder à la configuration d'une horloge pour les télécommunications. La réponse recommandée est 'N'.

XII-I-19-a - Sous-menu « Serial drivers »

L'option « 8250/16550 and compatible serial support » du sous-menu « Serial drivers » permet d'activer la prise en charge des ports série standards. Vous pouvez vous en passer si vous n'avez ni modem ni souris série, et si vous ne désirez pas utiliser le port série du tout. Il est recommandé de répondre 'Y' à cette question.

L'option « Console on 8250/16550 and compatible serial port » permet d'activer la redirection de la console système sur un port série. En général, la console est redirigée vers le terminal principal, aussi la réponse recommandée à cette question est-elle 'N'.

L'option « 8250/16550 PCI device support (NEW) » permet de prendre en charge les ports série connectés aux bus PCI. La réponse recommandée est 'Y'.

L'option « 8250/16550 PNP device support (NEW) » permet de prendre en charge les ports série Plug and Play des PC standards. La réponse recommandée est 'Y'.

L'option « 8250/16550 PCMCIA device support » permet de prendre en charge les ports série connectés via un adaptateur PCMCIA. La réponse recommandée est 'N'.

L'option « Maximum number of non-legacy 8250/16550 serial ports » permet d'indiquer le nombre maximum de ports série additionnels que Linux doit prendre en charge. La réponse recommandée est '4'.

L'option « Number of 8250/16550 serial ports to register at runtime » permet d'indiquer le nombre de ports séries qui seront activés dans le noyau au démarrage. La réponse recommandée est '4'.

L'option « Extended 8250/16550 serial driver options » permet d'activer les options de configuration étendues pour les ports série classiques. La réponse recommandée est 'N'.

L'option « **Support more than 4 legacy serial ports** » permet d'indiquer que l'ordinateur dispose de plus de quatre ports série livrés en standard. La réponse recommandée est 'N'.

L'option « **Support for sharing serial interrupts** » permet d'activer la gestion des interruptions de plusieurs ports série d'une même carte par un seul canal d'interruption. Il faut que la carte série gère le partage des interruptions pour pouvoir utiliser cette option. La réponse recommandée est 'N'.

L'option « **Autodetect IRQ on standard ports (unsafe)** » permet de demander au noyau de tenter une détection automatique de la ligne d'interruption utilisée par les ports série. Cette fonctionnalité n'est pas très sûre et la réponse recommandée est 'N'.

L'option « **Support RSA serial ports** » n'est pas documentée et ne sera pas décrite ici.

L'option « **Support 8250-type ports on MCA buses** » permet de prendre en charge les ports série sur les bus MCA. La réponse recommandée est 'N'.

L'option « **Digi International NEO PCI Support** » permet de prendre en charge les cartes multiport de la série NEO de Digi International. La réponse recommandée est 'N'.

XII-I-19-b - Sous-menu « IPMI »

L'option « **IPMI top-level message handler** » permet de prendre en charge la surveillance des capteurs du système (température, voltage, etc.). Les options qui suivent permettent de configurer les fonctionnalités de ce driver. La réponse recommandée est 'N'.

XII-I-19-c - Sous-menu « Watchdog cards »

L'option « **Watchdog Timer Support** » permet d'activer la détection des blocages systèmes grâce à un chien de garde. Le chien de garde exige un accès en écriture sur le fichier spécial de périphérique `/dev/watchdog`, de code majeur 10 et de code mineur 130, au moins une fois par minute. Si cette condition n'est pas vérifiée, la machine est redémarrée automatiquement. La gestion de cette fonctionnalité peut être réalisée matériellement grâce à une carte spéciale, ou logiciellement. La réponse recommandée est 'N'.

L'option « **Disable watchdog shutdown on close** » permet de maintenir l'activité du chien de garde même si le processus qui le réveille régulièrement ferme le fichier `/dev/watchdog`. Ce n'est pas le cas si la réponse 'N' est donnée, dans ce cas, le chien de garde ne fonctionnera plus en cas d'arrêt du processus de surveillance. La réponse recommandée est 'N'.

L'option « **Software Watchdog** » permet d'activer la gestion du chien de garde logiciel. La réponse recommandée est 'N'.

Les options qui suivent permettent de sélectionner les gestionnaires de chien de garde pour les différents types de cartes mères disposant d'un chien de garde matériel. Vous devez activer l'option correspondant à votre matériel si vous désirez utiliser cette fonctionnalité.

XII-I-19-d - Sous-menu « PCMCIA character device support »

Les options de ce menu permettent d'activer la gestion des cartes PCMCIA de type série, comme les modems, les ports série et les cartes réseau intégrant un modem. La réponse recommandée à ces options est 'N'.

XII-I-19-e - Sous-menu « TPM Hardware support »

Les options de ce menu permettent d'activer la gestion des puces cryptographiques compatibles avec les spécifications TCG (ex TCGA). Ces puces ont pour but d'authentifier le matériel et de permettre la fourniture d'un environnement logiciel authentifié et reconnu par les éditeurs de contenu afin de garantir que les données protégées par des droits d'auteurs stricts ne pourront pas être lues en dehors du cadre imposé par l'éditeur. Cette technique suppose que les logiciels et les systèmes soient fermés, et ne peut donc pas utiliser sous Linux sans violer la licence GPL. De ce fait, l'accès à ces puces est inutile, et ces puces ne seront généralement pas utilisées pour les tâches pour lesquelles elles ont été introduites. La réponse recommandée à ces options est donc 'N'.

XII-I-20 - Sous-menu « I2C support »

L'option « I2C support » permet d'activer la gestion du protocole de communication I2C. Ce protocole de communication est utilisé par beaucoup de microcontrôleurs, et peut être nécessaire pour accéder à certaines fonctionnalités. En particulier, les cartes d'acquisition TV basées sur les puces électroniques Bt848 nécessitent cette fonctionnalité pour être utilisables. Les capteurs matériels permettant de déterminer l'état du système utilisent également cette interface. La réponse recommandée est 'N'.

L'option « I2C device interface » active la gestion des fichiers spéciaux de périphériques `/dev/i2c-*`. La réponse recommandée est 'N'.

L'option « I2C bit-banging interfaces » active la gestion des adaptateurs « bit-banging ». Cette option est nécessaire pour faire fonctionner les cartes d'acquisition TV basées sur les puces électroniques Bt848. La réponse recommandée est 'N', sauf si vous disposez d'une telle carte. Les options suivantes activent la gestion des périphériques utilisant cette interface. Il n'est pas nécessaire de les activer pour faire fonctionner les cartes d'acquisition TV à base de Bt848. La réponse recommandée pour ces options est 'N'.

L'option « I2C PCF 8584 interfaces » active la gestion des adaptateurs PCF. Les options suivantes permettent d'activer les périphériques utilisant cette interface. Les réponses recommandées à ces questions sont 'N'.

L'option « I2C PCA 9564 interfaces » active la gestion des adaptateurs PCA. Les options suivantes permettent d'activer les périphériques utilisant cette interface. Les réponses recommandées à ces questions sont 'N'.

Les sous-menus « I2C Hardware Bus support » et « Miscellaneous I2C Chip support » permettent de prendre en charge les divers capteurs I2C présents sur les systèmes modernes. Vous devez activer les options correspondantes à votre matériel si vous désirez utiliser cette fonctionnalité.

Les options « I2C Core debugging messages », « I2C Algorithm debugging messages », « I2C Bus debugging messages » et « I2C Chip debugging messages » permettent d'activer la génération de messages de trace complémentaires relatifs aux fonctionnalités I2C. La réponse recommandée pour ces questions est 'N'.

XII-I-21 - Sous-menu « SPI support »

L'option « SPI support » active la prise du protocole de communication synchrone SPI, couramment utilisé pour accéder à des capteurs et à des mémoires programmables et flash. Les options fournies permettent d'activer les différents types de puces ainsi accessibles. La réponse recommandée est 'N'.

XII-I-22 - Sous-menu « Dallas's 1-wire bus »

L'option « Dallas's 1-wire support » active la prise en charge de périphériques à une broche relativement lents tels que des capteurs de températures. Les options fournies permettent d'activer les différents types de périphériques. La réponse recommandée est 'N'.

XII-I-23 - Sous-menu « Power supply class support »

L'option « Power supply class support » active la prise en charge de la surveillance des sources d'énergie (batteries, alimentation, ports USB, etc.). Notez que ce pilote n'est généralement pas nécessaire pour obtenir les informations sur les batteries des ordinateurs portables, pour lesquelles l'ACPI est utilisé. Il s'agit ici de prendre en charge les batteries des périphériques spécifiques tels que les PDA par exemple. Si vous activez cette option, vous pourrez accéder aux options des pilotes de périphériques gérant les différentes sources d'énergie. La réponse recommandée est 'N'.

L'option « Power supply debug (NEW) » permet d'activer les traces de débogage du sous-système de gestion des sources d'énergie. La réponse recommandée est 'N'.

XII-I-24 - Sous-menu « Hardware Monitoring support »

L'option « Hardware Monitoring support » active la prise en charge de la surveillance du bon fonctionnement du matériel de l'ordinateur. La plupart des ordinateurs disposent à présent de capteurs pour mesurer la température, les tensions, la vitesse des ventilateurs, etc. Cette option vous donnera à la liste des pilotes pour les différentes cartes mères. Notez que l'accès à ces pilotes nécessite également que l'option « I2C support » soit activée. La réponse recommandée est 'N'.

L'option « Hardware Monitoring Chip debugging messages » permet d'activer les traces de débogage du sous-système I2C pour les pilotes de surveillance du matériel. La réponse recommandée est 'N'.

XII-I-25 - Sous-menu « Multifunction device drivers »

Ce menu donne accès aux options permettant d'activer la prise en charge de différents périphériques multifonctions. Ces périphériques fournissent des fonctionnalités diverses telles qu'un adaptateur USB pour périphérique USB, des ports série asynchrones, des fonctionnalités audio ou vidéo. La réponse recommandée est 'N'.

XII-I-26 - Menu « Multimedia devices »

L'option « Video For Linux » permet d'activer les options de gestion de la vidéo et de radio sous Linux. Il faut activer cette option pour accéder aux sous-menus suivants, qui permettent de choisir les pilotes adaptés au type de matériel vidéo installé et les options de configuration pour certains de ces pilotes. La réponse recommandée est 'N'. Si vous activez cette option, vous aurez la possibilité d'activer l'interface de programmation Video For Linux version 1 avec l'option « Enable Video For Linux API 1 (DEPRECATED) » ou, à défaut, une couche de compatibilité pour les pilotes de périphériques qui l'utilisent encore, via l'option « Enable Video For Linux API 1 compatible Layer ». Comme de nombreux périphériques utilisent encore cette version de l'interface Video For Linux, la réponse recommandée est 'Y'.

L'option « DVB For Linux » permet d'activer les options des gestionnaires de périphériques des adaptateurs TNT sous Linux. Vous devez activer le pilote correspondant à votre matériel. La réponse recommandée à ces questions est 'N'.

L'option « DAB driver » permet de prendre en charge les récepteurs audio digitaux. La réponse recommandée est 'N'.

XII-I-26-a - Sous-menu « Video Capture Adapters »

Les options de ce menu permettent de choisir les pilotes pour les différents types de cartes d'acquisition vidéo. Vous devez activer la fonctionnalité correspondant au matériel dont vous disposez. En général, il est recommandé d'utiliser les pilotes sous forme de modules, car certains d'entre eux ne sont pas capables de faire la distinction entre les différents modèles de cartes qui utilisent la même électronique, et il faut leur communiquer ces informations sous la forme de paramètres lors de leur chargement. C'est en particulier le cas pour les cartes vidéo basées sur la puce électronique Bt848 et les puces qui en sont dérivées.

XII-I-26-b - Sous-menu « Radio Adapters »

Les options proposées par ce menu vous permettront d'activer les pilotes pour les différentes cartes radio supportées par Linux. Vous devez activer le pilote correspondant à votre matériel, et éventuellement spécifier les paramètres matériels de cette carte. La réponse recommandée à ces questions est 'N'.

XII-I-27 - Menu « Graphics support »

L'option « Lowlevel video output switch controls » permet de prendre en charge le pilote bas niveau de gestion des sorties vidéo. La réponse recommandée est 'N'.

L'option « Support for frame buffer devices » permet d'activer le support des cartes graphiques par l'intermédiaire d'une interface unifiée et d'une zone mémoire nommée « frame buffer », accessible via les fichiers spéciaux `/dev/fb*`. Cette interface permet aux programmes d'accéder aux fonctionnalités des cartes graphiques de manière portable. Le support de cette fonctionnalité n'est en général pas nécessaire pour les PCs, puisque beaucoup de serveurs X ont été développés pour cette plate-forme et que, dans le pire des cas, un serveur X basé sur le standard VESA 2.0 est disponible. Cependant, pour les autres plates-formes, ou si vous disposez d'une carte graphique exotique, il peut être utile d'activer cette fonctionnalité et d'utiliser un serveur X basé sur cette interface. Les options qui suivent permettent d'inclure les pilotes pour différentes cartes graphiques. Vous trouverez de plus amples informations sur ces options dans la huitième partie de ce document. La réponse recommandée est 'N'.

L'option « Enable firmware EDID » permet d'accéder à l'identifiant de périphérique EDID remonté par le firmware pour l'écran. Dans le cas des PC, cet identifiant est remonté par le BIOS. Cette option peut être utile si un pilote framebuffer ne parvient pas à obtenir cet identifiant en interrogeant l'écran. Cette option n'est pas nécessaire pour l'utilisation de l'environnement graphique X11, donc la réponse recommandée est 'N'.

Le sous-menu « Console display driver support » permet de choisir les gestionnaires de périphériques pour la console de Linux, c'est-à-dire l'affichage en mode texte. Sur les PC, le gestionnaire le plus efficace et le plus performant est celui utilisé pour gérer les cartes VGA, aussi est-il systématiquement sélectionné. Il donne accès à l'option « Enable Scrollback Buffer in System RAM », qui permet de mémoriser les pages texte en mémoire vive et non dans la mémoire de la carte graphique, et donc d'avoir une mémoire plus importante pour faire un défilement arrière en mode console. L'option « Video mode selection support », quant à elle permet de choisir le mode texte à utiliser au démarrage de Linux. Ce mode peut être indiqué grâce à un paramètre passé au noyau lors du démarrage. La réponse recommandée pour ces deux options est 'Y'.

Il existe également un gestionnaire pour les cartes graphiques MDA (ce sont des cartes extrêmement vieilles), qui peut être activé à l'aide de l'option « MDA text console (dual-headed) (EXPERIMENTAL) ». Ce gestionnaire permet également de disposer d'un second affichage en mode texte, si l'on installe la carte MDA en parallèle de la carte VGA du PC. Cette option ne doit être choisie que si la carte MDA n'est pas la carte d'affichage principale. La réponse recommandée est 'N'.

Le sous-menu « Backlight & LCD device support » permet de configurer les pilotes de contrôle de la lumière de rétroéclairage des écrans à LCD de certaines machines, telles que les PDA. Les sous-options de ce menu permettent de sélectionner les fonctionnalités prises en charge par ces pilotes. La réponse recommandée est 'N'.

Le sous-menu « Display device support » permet de configurer les pilotes de contrôle de la puissance des périphériques d'affichage de certaines plateformes. La réponse recommandée est 'N'.

Le sous-menu « Console display driver support » permet également d'utiliser le gestionnaire de périphérique du frame buffer pour la console, si l'option « Support for frame buffer devices » a été activée. Pour cela, il suffit d'activer l'option « Framebuffer Console support ». Cette option donnera la possibilité d'inclure une police de caractères graphique pour la console, à l'aide de l'option « Select compiled-in fonts ». Encore une fois, ces options ne sont pas nécessaires pour gérer correctement la console sur les ordinateurs de type PC.

Le sous-menu « Bootup logo » permet d'utiliser un logo représentant Tux au démarrage, si la fonctionnalité de frame buffer a été activée. Pour cela, il faut activer l'option « Bootup logo » et choisir un des logos proposés parmi les options suivantes. Cette fonctionnalité étant absolument indispensable pour la bonne marche de votre système et la sécurité de vos données, vous devez impérativement l'activer et admirer le magnifique Tux !

XII-I-28 - Menu « Sound »

L'option « Sound card support » permet d'activer la gestion des cartes son. Si vous ne disposez pas de carte son, choisissez la réponse 'N', sinon, activez cette fonctionnalité. Elle vous permettra de choisir parmi les deux jeux de drivers pour les cartes son dont dispose Linux actuellement. Les pilotes les plus performants sont sans aucun doute les drivers ALSA (« Advanced Linux Sound Architecture », mais vous aurez peut-être à utiliser les drivers OSS (« Open Sound System ») pour certaines cartes son.

Les drivers ALSA seront choisis de préférence en activant l'option « Advanced Linux Sound Architecture ». Cependant, l'interface de programmation des drivers OSS étant très utilisée, on veillera à activer les options de compatibilité « OSS Mixer API », « OSS PCM (digital audio) API », et « OSS Sequencer API ».

L'option « Sequencer support » permet de prendre en charge le séquenceur ALSA pour la gestion des fichiers MIDI. Elle vous permettra d'accéder à l'option « Sequencer dummy client », qui permet de gérer un simple client transférant les événements MIDI d'une entrée vers une sortie. La réponse recommandée est 'Y' pour ces deux questions.

L'option « RTC Timer support » permet de faire en sorte que les gestionnaires de carte son ALSA utilisent l'horloge matérielle de l'ordinateur pour les opérations de séquençement demandant une grande précision. Il est recommandé d'activer cette fonctionnalité, ainsi que sa sous-option « Use RTC as default sequencer timer ».

L'option « Dynamic device file minor numbers » permet de faire en sorte que les numéros de périphériques mineurs des fichiers de périphériques des cartes son sont alloués dynamiquement. Cela permet d'avoir plus de huit cartes son connectées, mais requiert l'utilisation du gestionnaire de périphériques en mode utilisateur udev. La réponse recommandée est 'N'.

L'option « Support old ALSA API » permet de prendre en charge l'ancienne interface de programmation d'ALSA (interface utilisée jusqu'à la version 0.9.0 rc3 comprise). Cette interface peut encore être utilisée par certains programmes, aussi la réponse recommandée est-elle 'Y'.

L'option « Verbose procs contents » permet de peupler le système de fichiers virtuel /proc/ avec des entrées complémentaires permettant d'obtenir des informations sur les périphériques audio. Ces entrées ne sont essentiellement utiles que pour les développeurs du noyau, aussi la réponse recommandée est-elle 'N'.

L'option « Verbose printk » et les options de débogage introduites par l'option « Debug » permettent de faciliter le diagnostic et le débogage des gestionnaires ALSA. Il n'est pas nécessaire de les activer en temps normal, aussi la réponse recommandée est-elle 'N'.

L'option « Debug » permet d'inclure du code de débogage dans le sous-système ALSA. Elle vous donnera accès à des sous-options permettant de spécifier la portée des tests effectués par ce code. Ce code n'est réellement utile que pour les développeurs du noyau, pour déboguer les pilotes de périphérique ALSA. La réponse recommandée est 'N'.

Les options qui suivent permettent de sélectionner les gestionnaires de périphériques pour le matériel installé. Ces gestionnaires sont classés par type d'interface (périphériques virtuels ou logiciels, cartes ISA ou PCI, appareils connectés par un port USB ou PCMCIA) dans les sous-menus correspondants. Vous devez bien entendu activer la prise en charge des gestionnaires de périphériques pour le matériel disponible dans votre ordinateur. Notez que les périphériques proposés dépendent de la prise en charge des bus correspondants. En particulier, les périphériques audio USB ne seront accessibles que si vous activez le support de l'USB dans le menu « USB support » ainsi que le support des périphériques audio USB (option « USB Audio/MIDI driver »).

L'option « **Open Sound System (DEPRECATED)** » permet d'activer la prise en charge des anciens pilotes OSS. Elle donne accès à deux jeux de pilotes, dont les pilotes OSS eux-mêmes. Les différentes options proposées permettent de sélectionner les gestionnaires de périphériques pour les différents matériels supportés.

Notez que le gestionnaire de périphériques pour le mixer intégré des cartes d'acquisition TV basées sur la puce Bt848 se situe parmi les pilotes OSS (option « **TV card (bt848) mixer support** »). Vous devez donc l'intégrer sous forme de module si vous disposez d'une telle carte. Il n'est pas nécessaire si vous utilisez les pilotes ALSA.

XII-I-29 - Menu « HID Devices »

L'option « **Generic HID support** » permet de prendre en charge les fonctionnalités génériques de l'interface homme/machine du noyau, qui sont utilisées par les gestionnaires de périphériques d'entrée/sortie USB ou Bluetooth. La réponse recommandée est 'Y'.

L'option « **USB Human Interface Devices (full HID) support** » permet d'activer la prise en charge des périphériques d'entrée et de saisie tels que les claviers, souris et tablettes de digitalisation. Ce pilote gère complètement les périphériques d'entrée USB, et est incompatible avec les pilotes simplifiés pour le clavier et la souris, que l'on peut activer avec les options « **USB HIDBP Keyboard (simple boot) support** » et « **USB HIDBP Mouse (simple boot) support** » du sous-menu « **USB HID Boot Protocol drivers** ». La réponse recommandée est 'N'.

L'option « **Force feedback support (EXPERIMENTAL)** » permet d'activer la gestion des périphériques à retour de force tels que certains joysticks. Cette option vous donnera accès à des options complémentaires permettant de sélectionner les gestionnaires de périphériques appropriés. La réponse recommandée est 'N'.

L'option « **/dev/hiddev raw HID device support (EXPERIMENTAL)** » permet de prendre en charge les périphériques d'entrée USB qui ne sont pas des périphériques utilisés par l'utilisateur directement, mais plutôt pour signaler des événements complémentaires (comme des moniteurs d'alimentation par exemple). La communication avec ces périphériques se fera par l'intermédiaire de fichiers spéciaux de périphériques `/dev/usb/hiddevX` complémentaires, tous de code majeur 180 et de codes mineurs allant de 96 à 111. La réponse recommandée est 'N'.

L'option « **USB HIDBP Keyboard (simple boot) support** » permet d'activer un pilote simplifié pour les claviers USB. Ce pilote peut être utilisé pour alléger le noyau dans les systèmes embarqués, mais ne gère pas toutes les touches des claviers étendus. Cette option est incompatible avec l'option « **USB Human Interface Device (full HID) support** » et n'est présentée que si cette dernière est désactivée. La réponse recommandée est 'N'.

L'option « **USB HIDBP Mouse (simple boot) support** » permet d'activer un pilote simplifié pour les souris USB. Ce pilote peut être utilisé pour alléger le noyau dans les systèmes embarqués. Cette option est incompatible avec l'option « **USB Human Interface Device (full HID) support** » et n'est présentée que si cette dernière est désactivée. La réponse recommandée est 'N'.

XII-I-30 - Menu « USB support »

L'option « **Support for Host-side USB** » permet de prendre en charge les périphériques USB de votre ordinateur sous Linux. La réponse recommandée est 'Y'.

L'option « **USB verbose debug message** » permet de demander aux gestionnaires de périphériques USB d'envoyer des informations de débogage dans les fichiers de traces du système. La réponse recommandée est 'N'.

L'option « **USB device filesystem** » active la gestion des informations sur les ports USB dans le système de fichiers virtuels `/proc/`. Ces informations comprennent en particulier la liste des périphériques connectés sur le bus USB. La réponse recommandée est 'Y'.

L'option « **USB device class-devices (DEPRECATED)** » permet d'exposer les informations sur les périphériques USB selon un ancien formalisme désormais obsolète. Ce formalisme peut toutefois encore être utilisé par les utilitaires en

espace utilisateur tels que le sous-système de gestion des périphériques udev et la bibliothèque libusb. Il est donc conseillé d'activer cette option le temps que les distributions aient été mises à jour. La réponse recommandée est 'Y'.

L'option « **Dynamic USB minor allocation (EXPERIMENTAL)** » active l'allocation dynamique de numéro de périphérique mineur pour les périphériques USB. Cela permet de gérer un grand nombre de périphériques du même type. La réponse recommandée est 'N'.

L'option « **USB selective suspend/resume and wakeup (EXPERIMENTAL)** » active la gestion de l'alimentation pour les périphériques USB. Cela permet de suspendre et de redémarrer les périphériques USB individuellement. Cette fonctionnalité est expérimentale et peut ne pas fonctionner correctement pour l'instant. La réponse recommandée est 'N'.

L'option « **USB device persistence during system suspend (DANGEROUS)** » permet de faire en sorte que les périphériques USB attachés au système lors de sa mise en veille, qu'elle soit prolongée ou non, disposent des mêmes identifiants lors du réveil. Cette option est très dangereuse et ne doit pas être utilisée si les périphériques peuvent être interchangés pendant la veille de l'ordinateur. La réponse recommandée est 'N'.

Les options suivantes (« **EHCI HCD (USB 2.0) support** », « **ISP116X HCD support** », « **OHCI HCD support** », « **UHCI HCD (most Intel and VIA) support** », « **SL811HS HCD support** » et « **R8A66597 HCD support** » permettent d'inclure les pilotes pour les différents types de contrôleurs USB. La plupart des machines utilisent un contrôleur UHCI (abréviation de l'anglais « Universal Host Controller Interface ») ou un contrôleur OHCI (abréviation de « Open Host Controller Interface »). Les contrôleurs UHCI sont fabriqués par Intel et VIA essentiellement, alors que les contrôleurs OHCI sont présents sur les chipsets non-Intel (Compaq, SiS, Aladdin). Vous devez choisir le pilote approprié au chipset de votre carte mère. Notez que la prise en charge de l'USB 2.0 n'est pas exclusive de celle de l'USB classique, les contrôleurs USB 2.0 étant généralement fournis avec des contrôleurs USB auxiliaires permettant d'accéder aux périphériques USB 1.0. Les gestionnaires de périphérique pour les contrôleurs USB 2.0 ne prenant en charge que les contrôleurs USB 2.0, vous devrez également inclure le gestionnaire de périphériques adéquat pour vos périphériques USB classiques si vous désirez y accéder.

L'option « **USB Modem (CDC ACM) support** » permet de prendre en charge les modems analogiques et Numéris utilisant l'interface CDC ACM (abréviation de l'anglais « Communication Device Class Abstract Control Model ») connectés sur le port USB. La réponse recommandée est 'N'.

L'option « **USB Printer support** » active la gestion des imprimantes connectées sur le port USB. La réponse recommandée est 'N'.

L'option « **USB Mass Storage support** » active la gestion des périphériques USB de masse (lecteurs de CD, clefs USB, etc.) ainsi que de la plupart des appareils photo USB, dont les mémoires flash sont accessibles comme des périphériques de masse. La réponse recommandée est 'Y'.

L'option « **USB Mass Storage verbose debug** » permet de demander au gestionnaire de périphériques USB de masse de générer des messages de débogage dans les fichiers de traces du système. La réponse recommandée est 'N'.

Les options qui suivent permettent d'activer les gestionnaires de périphériques pour les périphériques de stockage supportés par Linux. Notez qu'il n'est pas nécessaire de disposer d'un gestionnaire de périphériques pour accéder aux clefs USB et à la plupart des appareils photo. La réponse recommandée pour ces options est donc 'N', sauf si vous disposez d'un de ces appareils.

Viennent ensuite les options permettant de sélectionner les gestionnaires de périphériques pour de nombreux périphériques tels que les appareils photo et les caméras USB, les clefs USB ou les adaptateurs réseau. Vous devez activer ces options pour les périphériques que vous possédez, ou les activer sous la forme de module du noyau. La réponse recommandée est 'M'.

L'option « **USB Monitor** » permet d'activer une fonctionnalité de surveillance et de capture des messages échangés sur le bus USB. Cette fonctionnalité n'est utile que pour les développeurs, afin de mettre au point les pilotes des périphériques USB. La réponse recommandée est 'N'.

L'option « **USS720 parport driver** » permet de prendre en charge les adaptateurs USB pour port parallèle basés sur la puce USS-720 de Lucent Technologies. La réponse recommandée est 'N'.

L'option « **USB Serial Converter support** » du sous-menu « **USB Serial Converter support** » active la prise en charge des périphériques USB basés sur une interface série classique. Elle donne accès à un certain nombre de gestionnaires de périphériques, pour la plupart des périphériques série connectables sur un port USB. La réponse recommandée est 'N'.

Un certain nombre d'options suivent pour prendre en charge des périphériques USB divers et variés (LED, thermomètres, dongle, afficheurs, lecteurs d'empreintes, etc.). La réponse recommandée est 'N'.

L'option « **USB DSL modem support** » permet d'activer la prise en charge des modems ADSL USB. Cette option vous donnera accès aux différents pilotes disponibles pour ces modems. Remarquez que cette option et ces pilotes ne sont pas nécessaires pour les modems ADSL qui simulent une interface Ethernet, et pour lesquels on utilisera plutôt l'option « **CDC Ethernet support (smart devices such as cable modems)** ». La réponse recommandée est 'N'.

L'option « **Support for USB Gadgets** » du sous-menu « **USB Gadget Support** » permet d'activer la prise en charge du port USB par Linux pour les périphériques (c'est-à-dire du point de vue opposé à celui de l'ordinateur auquel on connecte ces périphériques). Cette option n'est utile que pour ceux qui réalisent des périphériques USB avec un système Linux embarqué. La réponse recommandée est donc 'N'.

XII-I-31 - Menu « **MMC/SD Card support** »

L'option « **MMC support** permet de prendre en charge les cartes multimédias MMC. Elle vous donnera accès aux options de configuration de ces cartes. Notez que l'utilisation de ces cartes au travers de lecteurs de cartes connectés sur les ports USB ne nécessite pas d'activer les fonctionnalités de ce menu. Seules l'option « **USB Mass Storage support** » du menu « **USB support** » est utile dans ce cas. La réponse recommandée est 'N'.

XII-I-32 - Menu « **LED devices** »

L'option « **LED support** » permet de prendre en charge les diodes électroluminescentes complémentaires que l'on trouve sur certains ordinateurs. Cette option vous permettra de contrôler l'affichage de ces diodes au niveau de l'espace utilisateur, ainsi qu'au niveau du noyau au moyen de pilotes accessibles via les options suivantes de ce menu. La réponse recommandée est 'N'.

XII-I-33 - Menu « **InfiniBand support** »

L'option « **InfiniBand support** » permet de prendre en charge les périphériques InfiniBand. Elle vous donnera accès aux options de configuration de cette technologie. La réponse recommandée est 'N'.

XII-I-34 - Menu « **EDAC - error detection and reporting (RAS) (EXPERIMENTAL)** »

L'option « **EDAC core system error reporting (EXPERIMENTAL)** » vous permettra d'activer les fonctions de détection et de signalement des erreurs matérielles. Les gestionnaires de périphériques que vous trouverez dans ce menu correspondent aux différents périphériques capables de signaler les erreurs matérielles, essentiellement des chipsets pour les barrettes mémoire. La réponse recommandée est 'N'.

XII-I-35 - Menu « **Real Time Clock** »

L'option « **RTC class** » permet de configurer les horloges temps-réel supplémentaires que l'on trouve sur certaines machines. Les options suivantes permettent d'activer les interfaces utilisateurs utilisables pour ces horloges, ainsi que les gestionnaires de périphériques correspondant au matériel installé. La réponse recommandée est 'N'.

XII-I-36 - Menu « DMA Engine support »

L'option « **Support for DMA engines** » permet d'accéder aux gestionnaires de périphériques de type DMA. Ces périphériques peuvent prendre en charge certains transferts de données entre la mémoire et des périphériques, déchargeant ainsi le processeur central de cette tâche. Actuellement, la seule utilisation possible de cette technique est le transfert de données entre les cartes réseau et les couches réseau de Linux. Les options suivantes permettent donc de configurer ces fonctionnalités. La réponse recommandée est 'N'.

XII-I-37 - Menu « Auxiliary Display support »

L'option « **Auxiliary Display support** » permet d'accéder aux gestionnaires de périphériques d'afficheurs de type LCD. Ce menu vous donnera accès aux options des gestionnaires de périphérique des différents afficheurs pris en charge. La réponse recommandée est 'N'.

XII-I-38 - Menu « Virtualization »

L'option « **Kernel-based Virtual Machine (KVM) support** » permet de prendre en charge les fonctionnalités de paravirtualisation en tant que système d'exploitation hôte des systèmes virtualisés. Cette option permet aux logiciels de virtualisation d'accéder aux fonctions de virtualisation des processeurs AMD ou Intel récents. Les options suivantes permettent de choisir le type de processeur dont vous disposez : choisissez l'option « **KVM for Intel processors support** » si vous disposez d'un processeur Intel, ou l'option « **KVM for AMD processors support** » si vous disposez d'un processeur AMD. La réponse recommandée est 'N'.

XII-I-39 - Menu « Userspace I/O »

L'option « **Userspace I/O drivers** » permet d'autoriser la prise en charge des pilotes de périphériques fonctionnant en espace utilisateur. La plupart des pilotes de périphériques sont intégrés dans le noyau à l'exécution, directement ou en tant que module, mais il est également possible de faire fonctionner des pilotes de périphériques plus spécifiques dans un processus classique grâce à cette option. Peu de pilotes ont recours à cette technique toutefois, essentiellement pour des raisons de performances et des raisons historiques. Cette option vous donnera accès à d'autres options permettant de prendre en charge des pilotes de périphériques en espace utilisateur. La réponse recommandée est 'N'.

XII-I-40 - Menu « Linux hypervisor example code (NEW) »

L'option « **Linux hypervisor example code (NEW)** » permet d'activer une fonction de virtualisation spécifique à Linux, grâce à laquelle il est possible de faire fonctionner plusieurs noyaux sur Linux sur la même machine. Cette fonction n'est disponible que pour les processeurs de type x86, si le mode d'adressage étendu n'a pas été activé, mais elle ne nécessite pas les extensions au jeu d'instructions x86 introduites par Intel et AMD sur les processeurs récents. Il est également nécessaire que l'option « **Paravirtualization support (EXPERIMENTAL)** » du menu « **Processor type and features** » ait été activée pour pouvoir activer cette fonctionnalité. La réponse recommandée est 'N'.

XII-J - Menu « File systems »

Linux peut utiliser un grand nombre de systèmes de fichiers, certains lui étant natifs, d'autres provenant d'autres systèmes d'exploitation. Les options de ce menu permettent de prendre en charge ces systèmes de fichiers, et de configurer les fonctionnalités dont ils disposeront.

Parmi ces systèmes, certains gestionnaires sont encore en cours de développement et ne fonctionnent pas très bien. Cela signifie qu'il est très déconseillé de les utiliser pour écrire des données sur ces systèmes de fichiers : de très grosses pertes de données peuvent s'ensuivre. En revanche, les systèmes de fichiers qui sont complètement gérés peuvent être utilisés sans risque. Cependant, ils peuvent ne pas être assez riches pour stocker les informations de fichiers nécessaires à tout système Unix. Dans ce cas, le système de fichiers ne peut être utilisé que pour stocker des

données d'applications, pas le système lui-même. C'est en particulier le cas de la FAT. Les systèmes de fichiers les plus utilisés actuellement sous Linux sont EXT3 et ReiserFS, qui offrent à la fois la sécurité, les fonctionnalités et les performances. L'option « `Ext3 journalling file system support` » permet d'activer le système de fichiers EXT3, et l'option « `Reiserfs support` » le système de fichiers ReiserFS. D'autres systèmes de fichiers journalisés sont disponibles, et chacun a ses avantages et ses inconvénients. Le choix dépend donc ici du contexte d'utilisation du système de fichiers et relève souvent d'une affaire de goût. Prenez toutefois garde au fait que certains systèmes de fichiers ne sont pas capables de prendre en compte les blocs défectueux du support physique. Si cette situation n'est pas normale, elle n'en est pour autant pas rare, surtout sur les ordinateurs vendus pour le grand public actuellement. Aussi est-il recommandé d'utiliser les systèmes de fichiers EXT2 ou EXT3, qui sont finalisés depuis longtemps déjà.

Quoi qu'il en soit, il est fortement recommandé d'utiliser un système de fichiers journalisé, car cela ajoute une sécurité accrue sur les données. Il est de plus impératif de compiler le pilote du système de fichiers racine dans le noyau. Si cela n'est pas fait, le noyau ne pourra pas monter le système de fichiers racine et se terminera en affichant le message « `kernel panic` ». Il ne faut pas compiler ce système de fichiers en tant que module, pour les mêmes raisons.

Les systèmes de fichiers Linux peuvent parfois prendre en charge des fonctionnalités avancées telles que les attributs étendus et les listes de contrôle d'accès en marge des mécanismes de gestion des droits Unix classiques. Ces fonctionnalités sont généralement accessibles via les options « `Extended attributes` » et « `POSIX Access Control Lists` » des systèmes de fichiers en question. Ces fonctionnalités peuvent être utiles pour les serveurs de fichiers, mais ne sont en général pas traitées par tous les utilitaires systèmes. La réponse recommandée à ces options est donc 'N'.

Certains systèmes de fichiers nécessitent le support de systèmes de fichiers de base. En particulier, les systèmes de fichiers VFAT (qui gère la FAT32) et MSDOS fs (qui gère les partitions DOS 12 et 16 bits) exigent le support du système de fichiers DOS FAT en général. Pour utiliser ces systèmes de fichiers, on devra donc activer l'option « `DOS FAT fs support` ». De même, pour lire les CD-ROM au format Joliet (l'extension Microsoft au format ISO 9660 pour supporter les noms longs sous Windows), il faudra activer l'option « `ISO 9660 CDFROM file system support` ». De toute façon, il est fortement recommandé de gérer ce système de fichiers si l'on veut utiliser des CD-ROMs.

Des systèmes de fichiers relativement spécifiques sont également développés de manière externe au noyau, soit parce qu'ils ne sont pas très courants, soit parce qu'ils sont très complexes, soit parce qu'ils ont besoin de fonctionnalités avancées que le noyau ne fournit pas. Ces systèmes de fichiers sont implémentés sous la forme de programmes utilisateurs, qui communiquent avec le noyau par l'intermédiaire d'une interface spéciale nommée « `FUSE` » (abréviation de l'anglais « `Filesystem in USEr space` »). Cette fonctionnalité est utilisée par des programmes pour exposer des fonctionnalités avancées au travers de simples systèmes de fichiers, et peut donc être particulièrement pratique. Il est donc recommandé de l'activer, grâce à l'option « `Filesystem in Userspace support` ».

Certains systèmes de fichiers sont virtuels. Ils ne correspondent à aucun support physique, et leur arborescence est créée uniquement en mémoire, à la volée, par le noyau. Parmi ces systèmes de fichiers, certains sont essentiels et doivent toujours être activés. Il s'agit du système de fichiers `/proc/`, qui fournit des informations dynamiquement sur l'état du système, du système de fichiers `/dev/`, qui est un système de fichiers obsolète qui permet de gérer les fichiers spéciaux de périphériques à la volée, du système de fichiers `/dev/pts/`, qui permet de créer des fichiers spéciaux de périphériques à la demande pour les pseudoterminaux et du système de fichiers `/dev/shm/`, qui permet de gérer les segments de mémoire partagée POSIX. À ces quatre systèmes de fichiers s'ajoute également un système de fichiers uniquement mémoire (option « `HugeTLB file system support` »), dont le fonctionnement n'est toutefois pas encore documenté. Il est fortement recommandé d'activer la gestion des systèmes de fichiers `/proc/`, `/dev/pts/` et `/dev/shm/`, car ils sont utilisés par beaucoup de programmes. On préférera désormais l'utilisation de `udev` au système de fichiers `/dev/`, aussi est-il déconseillé d'activer ce système de fichiers à présent.

Parfois, les applications utilisateur peuvent chercher à savoir si un fichier ou un répertoire est modifié par une autre application. Le noyau peut les prévenir lorsque cela se produit, leur évitant ainsi d'avoir à regarder régulièrement les fichiers et les répertoires. Pour cela, il fournit une interface de notification, que les applications peuvent utiliser. Cette fonctionnalité est activable grâce à l'option « `Inotify file change notification support` », que l'on aura donc intérêt à activer.

La plupart des systèmes de fichiers récents supportent la fonctionnalité de quota, qui permet de limiter la quantité d'espace disque disponible pour chaque utilisateur. L'activation de ces fonctionnalités se fait au travers de l'option

« **Quota support** ». Vous pouvez activer ces options si vous le désirez, bien qu'elles ne soient pas d'une utilité fondamentale pour une machine classique.

Enfin, Linux dispose d'une fonctionnalité permettant de monter automatiquement les systèmes de fichiers nommée le « **Kernel automounter** ». Deux versions de cette fonctionnalité ont été développées, et sont accessibles via les options « **Kernel automounter support** » et « **Kernel automounter version 4 support (also supports v3)** ». Ces fonctionnalités sont utiles pour monter les systèmes de fichiers réseau ou les systèmes de fichiers amovibles à la demande. Il est donc recommandé d'activer l'une de ces fonctionnalités. Notez que la version 4 du protocole de montage automatique permet également de prendre en charge la version 3.

XII-J-1 - Sous-menu « CDROM/DVD Filesystems »

Ce sous-menu du menu « **File systems** » regroupe les systèmes de fichiers des CD-ROM et des périphériques similaires.

L'option « **ISO 9660 CDROM file system support** » permet de prendre en charge le système de fichiers de base de tous les CD-ROM. Il est fortement recommandé de l'activer, ainsi que l'option « **Microsoft Joliet CDROM extensions** », qui permettra la prise en charge des extensions Microsoft à ce système de fichiers (gestion des noms longs de Windows 95).

L'option « **Transparent decompression extension** » permet la lecture des systèmes de fichiers de CD-ROM compressés. La réponse recommandée est 'N'.

L'option « **UDF file system support** » permet de prendre en charge le système de fichiers UDF, qui est utilisé sur les CD et les DVD écrits « par paquets ». La réponse recommandée est 'Y'.

XII-J-2 - Sous-menu « DOS/FAT/NT Filesystems »

Ce sous-menu du menu « **File systems** » permet d'accéder aux options des systèmes de fichiers de Microsoft.

L'option « **MSDOS fs support** » vous permettra de lire les systèmes de fichiers basés sur le DOS. Il est recommandé d'activer ce système de fichiers.

L'option « **VFAT (Windows 95) fs support** » vous permettra de lire les systèmes de fichiers FAT32 et suivants, qui gèrent les noms de fichiers longs. Si vous activez cette option, vous devrez indiquer la page de code utilisée implicitement par le noyau pour effectuer les conversions des noms de fichiers longs en noms de fichiers courts, si elle n'est pas spécifiée lors de la commande de montage. Cela se fait avec l'option « **Default codepage for FAT** ». La valeur par défaut à utiliser pour les systèmes français est 850. De même, il est nécessaire de spécifier le jeu de caractères utilisé implicitement pour les conversions entre les noms de fichiers Unix et les noms de fichiers longs. Il est recommandé d'utiliser le jeu de caractères ISO8859-1, qui est un sous-ensemble de l'Unicode courant dans la plupart des pays occidentaux. Pour cela, il faut affecter la valeur `iso8859-1` à l'option « **Default iocharset for FAT** ». Il est recommandé d'activer ce système de fichiers.

L'option « **NTFS file system support** » vous permettra d'accéder aux systèmes de fichiers NTFS, utilisés par tous les systèmes NT (Windows NT, Windows 2000 et Windows XP). Les deux sous-options de ce système de fichiers permettent respectivement de le déboguer et d'autoriser l'écriture sur les systèmes de fichiers NTFS. Le support de l'écriture est encore incomplet, et il ne permet que d'écraser des données dans un fichier existant. En revanche, contrairement aux anciennes versions de ce système de fichiers, la fonctionnalité d'écriture peut à présent être utilisée sans risque. La réponse recommandée est 'Y', car il existe à présent un pilote en espace utilisateur (basé sur FUSE) qui prend en charge complètement l'écriture et qui est suffisamment efficace pour un usage courant.

XII-J-3 - Sous-menu « Pseudo filesystems »

Ce sous-menu du menu « File systems » vous donnera accès aux systèmes de fichiers virtuels de Linux. Ces systèmes de fichiers sont fondamentaux au bon fonctionnement du système, aussi les activera-t-on en général.

L'option « `/proc` file system support » permet de prendre en charge le système de fichiers `/proc/`, qui fournit bon nombre d'informations sur le système et qui permet de le paramétrer. La réponse recommandée est 'Y'.

L'option « Virtual memory file system support (former shm fs) » vous permettra d'activer la gestion des segments de mémoire partagée via un système de fichiers virtuel. Cette technique est utilisée par de nombreux programmes, aussi est-il recommandé d'activer cette option.

L'option « HugeTLB file system support » permet d'activer un système de fichiers permettant aux applications d'accéder à des pages mémoires de grande taille tout en permettant au système d'exploitation d'optimiser les entrées dans le cache processeur contenant les correspondances entre les adresses mémoires virtuelles des processus et les adresses mémoire physiques. Ce cache étant de taille relativement limitée, et étant utilisé en permanence, les programmes qui le peuvent ont intérêt à utiliser cette fonctionnalité. Elle est toutefois spécifique à un usage relativement particulier, aussi la réponse recommandée est-elle 'N'.

L'option « Userspace-driven configuration filesystem (EXPERIMENTAL) » permet de prendre en charge un système de fichiers virtuel complémentaire du système de fichiers `/sys/`, en ce sens qu'il permet d'accéder plus spécifiquement aux paramètres de configuration des objets du noyau. En effet, le système de fichiers `/sys/` est plus destiné à fournir une vue de la configuration matérielle et de sa structure qu'à permettre la configuration des différents périphériques. Ce système de fichiers doit normalement être monté dans le répertoire `/config/`. La réponse recommandée est 'N'.

XII-J-4 - Sous-menu « Miscellaneous filesystems »

Ce sous-menu du menu « File systems » vous permettra de prendre en charge divers systèmes de fichiers utilisés par d'autres systèmes d'exploitation. La plupart de ces systèmes de fichiers ne sont pas finalisés, et le support de l'écriture est partiel ou dangereux. La réponse recommandée aux options de ce sous-menu est 'N'.

XII-J-5 - Sous-menu « Network File Systems »

L'option « NFS file system support » permet d'activer le support NFS classique en tant que machine cliente. NFS est un système de fichiers réseau extrêmement répandu, même si sa version 3 est réputée pour les trous de sécurité. Cette option vous donnera accès aux options « Provide NFSv3 client support » et « Provide NFSv4 client support (EXPERIMENTAL) », qui permettent de choisir la version de NFS à utiliser. Elle donnera également accès à l'option « Allow direct I/O on NFS files (EXPERIMENTAL) », qui permet d'activer les accès directs aux fichiers (c'est-à-dire sans cache) des systèmes de fichiers NFS. La réponse recommandée est 'N'.

L'option « NFS server support » permet d'activer le support NFS en tant que machine serveur. Cette option vous donnera accès aux options « Provide NFSv3 server support » et « Provide NFSv4 server support (EXPERIMENTAL) », qui vous permettront de choisir la version du système de fichiers NFS à utiliser. Vous aurez également accès à l'option « Provide NFS server over TCP support (EXPERIMENTAL) », qui permet d'activer le support du protocole NFS encapsulé dans le protocole de communication TCP. La réponse recommandée est 'N'.

L'option « Root file system on NFS » permet de monter la partition racine du système directement à partir d'un serveur NFS au démarrage de la machine. Cette option permet de faire tourner Linux sans disque dur, ou de réaliser une installation réseau par exemple. Cette option n'est disponible que si vous avez activé l'option « NFS file system support » en dur dans le noyau (le support NFS ne pouvant en effet pas être chargé en tant que module avant d'avoir monté la partition racine). L'utilisation de cette option suppose également l'activation d'une des sous-options de l'option « IP: kernel level autoconfiguration ». La réponse recommandée est 'N'.

L'option « **Support for rpcbind versions 3 & 4 (EXPERIMENTAL)** » permet de prendre en charge les versions 3 et 4 du protocole rpcbind, et à défaut de pouvoir communiquer avec ces protocoles avec le serveur, de se rabattre sur le protocole version 2. La réponse recommandée est 'N'.

L'option « **Secure RPC: Kerberos V mechanism (EXPERIMENTAL)** » permet de chiffrer les communications effectuées par le système de fichiers NFS via le protocole Kerberos. La réponse recommandée est 'N'.

L'option « **Secure RPC: SPKM3 mechanism (EXPERIMENTAL)** » permet de chiffrer les communications effectuées par le système de fichiers NFS via le protocole SPKM3. La réponse recommandée est 'N'.

L'option « **SMB file system support (to mount WfW shares, etc.)** » permet d'accéder aux répertoires partagés par les machines fonctionnant sous Windows. Cette fonctionnalité n'est possible que si les postes Windows utilisent TCP/IP comme protocole de réseau de base, et non NetBIOS. La réponse recommandée est 'Y'.

Les « **Use a default NLS** » et « **Default Remote NLS Option** » permettent de spécifier une page de codes par défaut pour lire les noms de fichiers et de répertoires partagés par le serveur. La réponse recommandée est 'N'.

L'option « **CIFS support (advanced network filesystem for Samba, Window and other CIFS compliant servers)** » permet d'activer la nouvelle version du système de fichiers SMB, que les systèmes Windows NT et que le logiciel Samba 3 utilisent en particulier. La réponse recommandée est 'Y'.

Les options « **CIFS statistics** » et « **Extended statistics** » permettent de collecter des statistiques sur les accès réalisés par les clients CIFS sur les partages. Ces statistiques sont accessibles via le système de fichiers virtuel `/proc/`. La réponse recommandée est 'N'.

L'option « **Support legacy servers which use weaker LANMAN security** » permet de prendre en charge un ancien algorithme d'authentification qui a été abandonné en raison de faiblesses de sécurité évidentes. Cet algorithme n'est plus utilisé de nos jours, mais vous pouvez en avoir besoin si vous accédez à une vieille machine Windows. La réponse recommandée est 'N'.

L'option « **CIFS extended attributes** » permet de prendre en charge les attributs étendus sur les parties CIFS. Ces attributs ne sont pas gérés nativement par CIFS, et ne sont pas traités de la même manière sous Windows et Samba. La réponse recommandée est 'N'.

L'option « **CIFS POSIX Extensions** » permet de prendre en charge une extension du protocole CIFS pour gérer les extensions POSIX sur les serveurs de fichiers Samba. La réponse recommandée est 'N'.

L'option « **Enable additional CIFS debugging routines** » permet d'activer des vérifications complémentaires qui peuvent dégrader les performances du module CIFS. La réponse recommandée est 'N'.

L'option « **CIFS Experimental Features (EXPERIMENTAL)** » permet de prendre en charge les fonctionnalités expérimentales du module CIFS. En particulier, elle vous donnera accès à l'option « **Kerberos/SPNEGO advanced session setup (EXPERIMENTAL)** », qui permet d'utiliser un logiciel dédié afin de réaliser les authentifications Kerberos nécessaires pour accéder à certains serveurs de fichiers sécurisés. La réponse recommandée est 'N'.

L'option « **NCP file system support (to mount NetWare volumes)** » permet d'accéder aux volumes NetWare. Cette fonctionnalité nécessite la gestion d'IPX au niveau de la machine Linux. On notera qu'il est inutile d'activer cette fonctionnalité si l'on désire faire en sorte que la machine Linux soit serveur de fichiers Novell. Les options qui sont proposées si l'on active cette fonctionnalité permettent de paramétrer les accès aux volumes NetWare (sécurité, verrous sur les fichiers, droits d'accès...). La réponse recommandée est 'N'.

L'option « **Coda file system support (advanced network fs)** » permet d'activer le support du système de fichiers réseau Coda, qui donne accès aux périphériques par le réseau comme s'ils étaient branchés sur la machine locale. Cette option active les fonctionnalités clientes au niveau du noyau, mais il faut également des outils complémentaires pour implémenter les fonctions serveur. La réponse recommandée est 'N'.

L'option « **Andrew File System support (AFS) (EXPERIMENTAL)** » permet de prendre en charge le système de fichiers réseau AFS, en lecture seule uniquement pour l'instant. La réponse recommandée est 'N'.

L'option « **Plan 9 Resource Sharing Support (9P2000) (Experimental)** » permet de prendre en charge le protocole de partage de ressources 9P9000 du système de fichiers Pan 9. La réponse recommandée est 'N'.

XII-J-6 - Sous-menu « Partition Types »

Certains systèmes utilisent un format différent pour les tables de partitions que le format utilisé par le BIOS des PC. C'est en particulier le cas de Windows 2000 et Windows XP, lorsqu'ils réalisent un agrégat de partitions. Pour lire les données stockées sur les disques partitionnés par ces systèmes, vous devez activer une gestion spéciale des tables de partitions. Les options de ce menu permettent d'activer la gestion des tables de partitions de ces systèmes. La réponse recommandée à ces questions est 'N', sauf si vous désirez accéder aux données d'un disque partitionné avec le gestionnaire de volumes de Windows par exemple, auquel cas vous devrez également activer l'option « **Windows Logical Disk Manager (Dynamic Disk) support** ».

XII-J-7 - Sous-menu « Native Language Support »

Les options fournies ici permettent de choisir les pages de codes à utiliser pour les jeux de caractères dans les systèmes de fichiers. Ces pages de codes doivent être choisies en fonction de la langue d'installation du système. Les pages de codes recommandées pour un système français sont les suivantes :

- « **Codepage 437 (United States, Canada)** », indispensable en raison de la majorité des programmes provenant des États Unis ;
- « **Codepage 850 (Europe)** », pour les caractères accentués européens ;
- « **NLS ISO 8859-1 (Latin 1; Western European Languages)** » pour les caractères accentués européens sur les CDROM ;
- « **NLS ISO 8859-15 (Latin 9; Western European Languages with Euro)** » pour les caractères accentués européens sur les CDROM, y compris quelques caractères additionnels non gérés par la page de code ISO 8859-1 ;
- « **NLS UTF8** », pour les caractères encodés en ISO10686 (dont Unicode est un sous-ensemble). Cette option est nécessaire pour certains systèmes de fichiers qui travaillent en Unicode, comme NTFS par exemple.

Pour toutes les autres pages de codes, la réponse recommandée est 'N'.

L'option « **Default NLS Option** » permet de choisir la page de code par défaut à utiliser parmi celles qui ont été choisies. La réponse recommandée est « **iso8859-1** ».

XII-K - Menu « Instrumentation Support »

L'option « **Profiling support (EXPERIMENTAL)** » permet d'instrumenter le code du noyau afin de mesurer ses performances et les temps d'exécution de certaines de ces fonctions. Si vous activez cette option, vous pourrez choisir le système d'instrumentation à utiliser. Ces options sont réservées aux développeurs du noyau et ne seront donc pas décrites plus en détail ici. La réponse recommandée est 'N'.

L'option « **Kprobes (EXPERIMENTAL)** » permet d'activer la mise en place de points d'entrée génériques pour appeler une fonction spécifique en n'importe quelle adresse mémoire du noyau. Cette technique permet d'instrumenter le noyau de manière externe et de le déboguer de manière efficace et non intrusive. Cette option est réservée aux développeurs du noyau, aussi la réponse recommandée est-elle 'N'.

XII-L - Menu « Kernel hacking »

L'option « Show timing information on printk » permet d'ajouter une datation aux traces du noyau. La réponse recommandée est 'N'.

L'option « Enable `__must_check` logic » permet de marquer les fonctions qui retournent un résultat avec un attribut du compilateur GCC pour que les appels de ces fonctions qui ne vérifient pas le code de retour soient signalés par un avertissement. Cette option est donc une aide à la recherche de bogues pour les développeurs. Toutefois, elle n'est pas d'utilité pour les utilisateurs et la réponse recommandée est 'N'.

L'option « Magic SysRq key » permet d'activer la prise en charge de la prise de contrôle du noyau dans un mini débogueur intégré via la touche système. Cette fonctionnalité permet d'effectuer certaines opérations sur le noyau en fonctionnement via des raccourcis clavier. De ce fait, elle constitue un trou de sécurité et ne doit être utilisée que par les développeurs. La réponse recommandée est 'N'.

L'option « Enable unused/obsolete exported symbols » permet de conserver les fonctions et les variables non utilisées dans le noyau pour les exporter, dans l'éventualité qu'un pilote ou un module externe les utiliserait. Cependant, conserver ces symboles accroît la taille du noyau de manière inutile, et il peut être intéressant de les supprimer. La réponse recommandée est 'Y'.

L'option « Debug Filesystem » permet d'activer la prise en charge d'un système de fichiers spécial utilisé par les développeurs du noyau pour y stocker des fichiers de débogage. La réponse recommandée est 'N'.

L'option « Run 'make headers_check' when building vmlinux » permet de faire en sorte qu'à chaque compilation du noyau, des tests sont effectués sur les en-têtes du noyau pour vérifier que ceux-ci n'exportent pas de structures privées du noyau vers l'espace utilisateur, et que ces en-têtes sont donc utilisables pour compiler des programmes utilisateurs. Cette option n'est utile que pour les développeurs du noyau, aussi la réponse recommandée est-elle 'N'.

L'option « Kernel debugging » permet d'accéder à une série d'options relatives au débogage du noyau. Ces options ne sont pas réellement utilisées pour un utilisateur et ne sont présentes que pour les développeurs de gestionnaires de périphériques ou les développeurs du noyau. Toutefois, certaines options utiles de la configuration générale du noyau peuvent ne pas être accessibles si vous désactivez cette fonctionnalité, aussi la réponse recommandée est-elle 'Y'. Vous pourrez accéder à des options complémentaires si vous répondez par l'affirmative à cette question.

L'option « Debug shared IRQ handlers » permet d'activer des vérifications sur les pilotes de périphériques qui se partagent une même interruption matérielle. La réponse recommandée est-elle 'N'.

L'option « Detect Soft Lockups » permet d'activer la détection d'une catégorie de bogues du noyau, à cause desquels celui-ci peut boucler indéfiniment dans un traitement sans permettre aux autres parties du système de travailler. Cette option n'est utile que pour les développeurs du noyau, aussi la réponse recommandée est-elle 'N'.

L'option « Collect scheduler debugging info » permet de récupérer des informations de débogage sur l'ordonnanceur de tâches du système et de les exposer via le système de fichiers virtuel `/proc/`. La réponse recommandée est 'N'.

L'option « Collect scheduler statistics » permet d'accumuler des statistiques sur l'ordonnanceur du système afin de déterminer les temps de latence et d'optimiser certains processus. La réponse recommandée est 'N'.

L'option « Collect kernel timers statistics » permet d'accumuler des statistiques sur les horloges périodiques utilisées en interne dans le noyau et de les exposer via le système de fichiers virtuel `/proc/`. La réponse recommandée est 'N'.

L'option « Debug slab memory allocations » permet d'activer des contrôles plus stricts sur les mécanismes d'allocation et de libération de la mémoire dans le noyau. La réponse recommandée est 'N'.

L'option « Debug slab memory allocations » permet d'activer des contrôles plus stricts sur les mécanismes d'allocation et de libération de la mémoire dans le noyau. La réponse recommandée est 'N'.

L'option « **SLUB debugging on by default** » permet d'activer les vérifications de l'allocateur mémoire SLUB par défaut au démarrage de la machine. Ce comportement par défaut peut être modifié en utilisant une option du noyau en ligne de commande lors du démarrage. La réponse recommandée est 'N'.

L'option « **Debug preemptible kernel** » permet d'activer une option de débogage qui simule une machine multiprocesseur sur les noyaux préemptibles, afin de détecter les erreurs dans les portions de code préemptibles. La réponse recommandée est 'N'.

L'option « **RT Mutex debugging, deadlock detection** » permet de contrôler que les zones d'exclusion mutuelles sont correctement utilisées et de détecter et de signaler les cas d'interblocages immédiatement en cours d'exécution. La réponse recommandée est 'N'.

L'option « **Built-in scriptable tester for rt-mutexes** » permet d'inclure un outil de test complémentaire pour les zones d'exclusions mutuelles. La réponse recommandée est 'N'.

L'option « **Spinlock and rw-lock debugging: basic checks** » permet de contrôler que les sections critiques de type spinlock sont toujours bien initialisées et que les erreurs les plus classiques ne sont pas faites lors de leur utilisation. La réponse recommandée est 'N'.

L'option « **Mutex debugging: basic checks** » permet de contrôler que les zones d'exclusion mutuelles sont correctement utilisées. La réponse recommandée est 'N'.

L'option « **Semaphore debugging** » permet de contrôler que les sémaphores sont correctement utilisés. La réponse recommandée est 'N'.

L'option « **Lock debugging: detect incorrect freeing of live locks** » permet de contrôler que les primitives de synchronisation du noyau ne sont pas détruits de manière incorrecte lors qu'elles sont encore utilisées par un thread dans le noyau. La réponse recommandée est 'N'.

L'option « **Lock debugging: prove locking correctness** » permet de conserver en mémoire des informations sur l'utilisation des différentes primitives de synchronisation et sur le contexte de leur utilisation, afin de vérifier qu'aucun cas de blocage ne peut se produire théoriquement en fonction de modèles mathématiques de ces blocages. Cette fonctionnalité ralentit le noyau, car il doit effectuer des traitements complémentaires à chaque utilisation de ces primitives de synchronisation. La réponse recommandée est 'N'.

L'option « **Lock usage statistics** » permet de récolter des statistiques sur les utilisations des verrous dans le noyau afin de détecter les portions de code critiques qui ne peuvent pas être exécutées en parallèle. Cette option n'est utile que pour les développeurs, et la réponse recommandée est 'N'.

L'option « **Lock dependancy engine debugging** » permet d'activer un jeu de tests complémentaires dans la fonctionnalité de vérification de l'utilisation des primitives de synchronisation afin de vérifier son intégrité. Cette option n'est utile que pour les développeurs, et la réponse recommandée est 'N'.

L'option « **Spinlock debugging: sleep-inside-spinlock checking** » permet d'activer une alarme en cas d'appel d'une méthode pouvant provoquer la suspension du processus appelant au sein d'une section critique de type spinlock. Ces sections critiques ne permettent en effet pas aux threads appelant de rendre la main à l'ordonnanceur du noyau, ce qui constitue un bogue grave que cette option permet de détecter. La réponse recommandée est 'N'.

L'option « **Locking API boot-time self-tests** » permet de vérifier au démarrage de la machine que les techniques de débogage des primitives de synchronisation sont bien efficaces. La réponse recommandée est 'N'.

L'option « **kobject debugging** » permet d'activer la génération de traces complémentaires sur les objets systèmes. La réponse recommandée est 'N'.

L'option « **Highmem debugging** » permet d'activer la génération de traces complémentaires sur les accès à la mémoire haute pour les systèmes qui ont plus d'un gigaoctet de mémoire. La réponse recommandée est 'N'.

L'option « **Verbose BUG() reporting (add 70K)** » permet d'activer la détermination des fichiers et numéros de ligne pour l'affichage des piles d'appels lors de l'affichage d'un bug grave. Cette option permet d'aider les développeurs pour la mise au point du noyau, mais augmente sensiblement sa taille. La réponse recommandée est 'N'.

L'option « **Compile the kernel with debug info** » permet de compiler le noyau avec les informations symboliques de débogage. Ces informations sont très utiles pour les développeurs pour l'analyse des bogues. La réponse recommandée est 'N'.

L'option « **Debug VM** » permet d'activer la prise en charge de tests d'intégrité de la mémoire virtuelle plus poussés, mais qui ralentissent le système. La réponse recommandée est 'N'.

L'option « **Debug linked list manipulation** » permet d'activer la prise en charge de tests d'intégrité des structures de données de type liste chaînée et des fonctions de parcours de ces listes. La réponse recommandée est 'N'.

L'option « **Compile the kernel with frame pointers** » permet de désactiver une optimisation du compilateur dont le but est normalement de simplifier les appels de fonctions. Cette optimisation peut empêcher un débogage correct des piles d'appels du noyau, aussi les développeurs du noyau ils la désactivent-ils généralement à l'aide de cette option. La réponse recommandée est 'N'.

L'option « **Force gcc to inline functions marked 'inline'** » permet de demander au compilateur de toujours inclure les fonctions marquées inline en tant que telles, même si ses algorithmes d'optimisation lui indiquent que cela n'est pas nécessaire. Les versions récentes de GCC sont suffisamment efficaces pour toujours savoir comment se comporter, mais ce n'est pas le cas des anciennes versions, aussi cette option est elle fournie afin de contrecarrer ses éventuels mauvais choix. La réponse recommandée est 'N'.

L'option « **torture tests for RCU** » permet d'activer un module qui réalise des tests poussés sur le code de gestion des copies de page sur modification, dans le but de détecter des bogues éventuels. La réponse recommandée est 'N'.

L'option « **Linux Kernel Dump Test Tool Module** » permet d'activer un module dont le but est de planter le noyau. L'utilité de ce module est de tester l'infrastructure de génération de rapports de plantage du noyau. Il ne faut évidemment pas l'utiliser, et la réponse recommandée est 'N'.

L'option « **Fault-injection framework** » permet d'activer la possibilité de simuler une erreur à différents niveaux dans le noyau. Les options suivantes permettent de prendre en charge la génération d'erreurs fictives au niveau des fonctions de réservation de mémoire et de gestion des entrées/sorties. La réponse recommandée est 'N'.

L'option « **Early printk** » permet d'activer les fonctions d'affichage sur la console et le port série très tôt dans la phase de démarrage du noyau. Cette option est utile si le noyau plante au démarrage avant qu'il n'ait eu le temps d'initialiser la console. Toutefois, son implémentation n'est pas très élégante et cette option ne doit être activée que si nécessaire. La réponse recommandée est donc 'N'.

L'option « **Check for stack overflows** » permet de vérifier qu'aucun débordement de pile n'a lieu dans le noyau. La réponse recommandée est 'N'.

L'option « **Stack utilization instrumentation** » active une fonctionnalité permettant de connaître la taille maximum utilisée par tous les threads du noyau. La réponse recommandée est 'N'.

L'option « **Debug page memory allocations** » permet de supprimer les pages mémoires libérées de l'espace d'adressage du noyau, afin de détecter les accès aux pages mémoires non disponibles. Cette fonctionnalité facilite la détection de bogues relatifs à l'utilisation de la mémoire. La réponse recommandée est 'N'.

L'option « **Write protect kernel read-only data structures** » permet de marquer les pages mémoire des structures en lecture seule du noyau comme inaccessibles en écriture au niveau de l'unité de gestion de la mémoire virtuelle. Cela permet de détecter les écritures sur ces structures, qui ne doivent jamais être modifiées après leur initialisation. La réponse recommandée est 'N'.

L'option « **Enable IOMMU debugging** » permet d'activer des tests sur l'utilisation des ressources DMA par les pilotes de périphérique sur les architectures x86_64 même lorsque la machine dispose de moins de 4 Go de mémoire. Cela permet de diagnostiquer les problèmes d'accès direct à la mémoire des périphériques 32 bits, et de déboguer les pilotes qui supposent que l'espace d'adressage est limité à 32 bits seulement. La réponse recommandée est 'N'.

L'option « **IOMMU leak tracing** » permet de vérifier que les ressources DMA allouées par les pilotes de périphérique sont bien relâchées. Cela permet de déboguer les pilotes qui n'utilisent pas correctement les fonctions de gestion des accès directs à la mémoire. La réponse recommandée est 'N'.

L'option « **Use 4Kb for kernel stacks instead of 8Kb** » permet de réduire la taille de la pile d'exécution des threads au sein du noyau. Cette réduction permet d'exécuter plus de threads dans le système. La réponse recommandée est 'N'.

L'option « **Enable doublefault exception handler** » permet de supprimer le gestionnaire d'exceptions pour les exceptions processeur récursives qui, sans traitement particulier, provoquent un redémarrage. Supprimer ce gestionnaire peut faire gagner quatre kilooctets de mémoire, ce qui peut être intéressant pour les systèmes embarqués, mais cela induit un risque de redémarrages imprévus. Les cas de double faute sont toutefois relativement rares, aussi cette option est-elle proposée. Toutefois, il est recommandé de répondre 'Y' à cette question afin de laisser le noyau gérer ces situations exceptionnelles, mais qui peuvent se produire en fonctionnement normal.

XII-M - Menu « Security options »

L'option « **Enable access key retention support** » permet d'activer une fonctionnalité de stockage des clefs privées au sein du noyau et non au sein des processus. La réponse recommandée est « N ».

L'option « **Enable the /proc/keys file by which all keys may be viewed** » permet de prendre en charge le fichier spécial de périphérique /proc/keys grâce auquel les clefs privées peuvent être lues. Chaque processus ne voyant que ses propres clefs, le risque de diffusion des clefs privées est limité et ce fichier ne constitue pas un trou de sécurité. La réponse recommandée est toutefois 'N'.

L'option « **Enable different security models** » permet d'activer les modèles de sécurité autres que ceux fournis par défaut par Linux (basés essentiellement sur la notion d'utilisateur et de droits d'accès sur les fichiers). Cette option donne accès aux options de configuration de ces modèles de sécurité. La réponse recommandée est 'N'.

L'option « **Socket and Networking Security Hooks** » permet d'inclure des points d'entrée au niveau des sockets réseau utilisables par des modules de sécurité afin de définir une politique de sécurité plus précise au niveau du réseau. La réponse recommandée est 'N'.

L'option « **XFRM (IPSec) Networking Security Hooks** » permet d'inclure des points d'entrée au niveau des fonctions de traitement des paquets réseau par des modules de sécurité afin de définir une politique de sécurité définie au niveau de chaque paquet. Cette fonctionnalité n'est pleinement utilisable que pour les paquets des connexions IPSec. La réponse recommandée est 'N'.

L'option « **Default Linux Capabilities** » permet d'activer la gestion de sécurité au travers de la notion de capacité. Une capacité est un droit particulier sur le système que l'on peut donner explicitement à un processus. Les capacités permettent en particulier de restreindre les droits des processus tournant sous le compte root, alors qu'ils auraient été sans cela tout-puissants. La réponse recommandée est 'Y', car cette fonctionnalité est une fonctionnalité classique de Linux.

L'option « Root Plug Support » est un module de sécurité d'exemple. Il permet d'interdire l'exécution de tous les processus appartenant au groupe root si une clef USB n'est pas connectée au système. Ce module n'ayant pas d'utilité autre que pédagogique, la réponse recommandée pour cette option est 'N'.

L'option « NSA SELinux Support » permet de prendre en charge les extensions de sécurité ajoutées à Linux par la NSA (« National Security Agency » américaine). Cette option donnera accès à des options complémentaires de configuration de ce modèle de sécurité. La réponse recommandée est 'N'.

XII-N - Menu « Cryptographic options »

Ce menu comprend un certain nombre d'options permettant de prendre en charge au niveau du noyau les algorithmes de chiffrement les plus classiques. Certains de ces modules sont nécessaires pour les protocoles réseau sécurisés par chiffrement ou pour les systèmes de fichiers chiffrés. D'autres algorithmes, comme celui activé par l'option « Deflate compression algorithm », peuvent également être utilisés pour compresser des données. La réponse recommandée est 'N' pour ces options, sauf si vous désirez utiliser les fonctionnalités de compression ou de cryptage pour les protocoles réseau IPSec ou pour les systèmes de fichiers (voir aussi l'option « Crypt target support » du menu « Device mapper support »).

XII-O - Menu « Library routines »

Les options de ce menu permettent d'inclure dans le noyau différentes fonctions (notamment des fonctions de calcul de contrôles de redondance cycliques) même si aucune partie du noyau ne les utilise. Cela permet de les exporter pour des modules livrés par des tiers qui en auraient éventuellement besoin. La réponse recommandée est 'Y'.

XIII - Annexe B. Compilation et mise à jour des principaux composants du système

Les paragraphes suivants contiennent les remarques et les options à choisir pour compiler les principaux composants du système. La compilation de GCC et du noyau a déjà été vue et ne sera donc pas décrite ici. Les informations fournies ci-dessous ne sont pas très détaillées, car la plupart des gens n'en ont pas besoin. Elles relèvent plus de la création du système que de son installation, mais elles sont fournies, car elles peuvent être utiles pour les personnes désirant approfondir sérieusement le sujet et disposant d'un certain niveau en informatique.

XIII-A - Compilation de make 3.81

Le programme make est utilisé systématiquement lors des compilations. Certaines compilations nécessitent la dernière version de make, qui n'est pas forcément fournie avec votre distribution. Vous pouvez donc avoir besoin de le recompiler. La version courante de make est la 3.81, il est fortement recommandé de l'utiliser.

La compilation de make est tout à fait classique. Elle nécessite de :

- extraire les fichiers sources de leur archive dans un répertoire source `srcdir` ;
- créer un répertoire pour les objets et s'y placer ;
- taper :

```
CFLAGS=-O2 ../srcdir/configure --enable-shared --host=i686-pc-linux-gnu \
--prefix=/usr
```

- lancer make.

Lorsque la compilation sera terminée, vous pourrez tester le nouveau programme avant de l'installer. Pour cela, il suffit de taper la commande suivante :

```
make check
```

Le programme make ainsi configuré sera installé avec la simple commande suivante :

```
make install
```

Le répertoire d'installation est le répertoire /usr/bin si le préfixe utilisé est /usr/.

XIII-B - Compilation des binutils 2.17.0

L'archive de GCC ne fournit en soi que les compilateurs des langages supportés, ainsi que quelques outils complémentaires. Elle ne contient pas en revanche les outils de génération et de manipulation des fichiers binaires. Ces outils sont fournis dans un paquetage distinct, les binutils.

La qualité des outils de manipulation des fichiers binaires est un facteur important pour une génération correcte des programmes à partir de leurs sources. En clair, le compilateur ne fait qu'une partie du travail, le reste revient aux binutils. Il est donc nécessaire de disposer des dernières versions de ces outils, car certaines compilations de grande taille peuvent mal se passer avec les anciennes versions. C'est en particulier le cas de la glibc.

La version courante des binutils est la 2.17.0. Les sources de ce paquetage peuvent, comme à l'accoutumée, être récupérées sur le [site Internet du GNU](#). Leur compilation ne pose pas de problème, puisqu'elle utilise le très classique script de configuration de GNU. La configuration se fera donc avec la commande suivante :

```
CFLAGS=-O2 ./configure --prefix=/usr
```

La compilation et l'installation de ces outils pourront alors être faites simplement avec les deux commandes suivantes :

```
make  
make install
```

Comme d'habitude, vous pourrez tester que tout s'est bien passé à l'aide de la commande make check avant de lancer l'installation.

XIII-C - Compilation de la bibliothèque C 2.5.0

La bibliothèque C constitue l'ensemble des bibliothèques de fonctions que tous les programmes écrits en C utilisent pour accéder au système. Il s'agit donc d'un composant essentiel dans le système (qui, rappelons-le, est lui-même écrit en C). Recompiler la bibliothèque C est donc une opération très sensible. Vous pouvez donc casser complètement votre système en effectuant cette mise à jour. De plus, certaines distributions fournissent des versions personnalisées du noyau, de GCC et de la bibliothèque C et recompiler les versions officielles de ces composants est certainement le meilleur moyen pour ne plus pouvoir lancer le moindre programme avec ces distributions. Vous êtes prévenus.

La compilation de la bibliothèque C est une opération fastidieuse à plus d'un titre. Elle nécessite souvent de recompiler certains outils, et elle est longue et consomme énormément de ressources. Enfin, il faut parfois recompiler certaines parties du système après son installation, afin de résoudre des conflits de versions dus à la migration. La dernière version de la bibliothèque C GNU est la 2.3.6, elle peut être récupérée sur le [site projet du GNU](#).

Les prérequis sont les suivants :

- les sources du noyau Linux 2.6 doivent être disponibles dans le répertoire /usr/src/linux/ ;
- le compilateur GCC (version 3.4.6 ou plus) doit avoir été compilé et installé ;
- les outils de manipulation des fichiers binaires (binutils 2.17 ou plus) doivent avoir été compilés et installés ;
- le programme make utilisé doit être de version récente (3.80 ou plus).

Le processus de compilation est ensuite assez classique. Il suffit de :

- extraire les fichiers sources de leur archive dans un répertoire source `srcdir` ;
- créer un répertoire pour les objets en dehors du répertoire des sources de la bibliothèque et s'y placer ;
- taper :

```
CFLAGS=-O2 ../srcdir/configure --prefix=/usr --enable-shared \  
--enable-add-ons=libidn,nptl --with-tls
```

Les options de la commande de configuration permettent l'emploi des bibliothèques dynamiques et la compilation des modules additionnels pour le support du multithreading de nouvelle génération. Le répertoire d'installation sera `/usr/lib/` pour les bibliothèques, et `/usr/include/` pour les fichiers d'en-têtes.

La compilation peut ensuite être lancée avec la simple commande suivante :

```
make
```

Une fois la compilation effectuée, vous pouvez tester la nouvelle bibliothèque avant de l'installer. Pour cela, tapez la commande suivante :

```
make check
```

Il est vivement recommandé d'effectuer ce test. Si la moindre erreur apparaît pendant l'exécution du test, n'installez surtout pas la bibliothèque, vous détruiriez sans aucun doute votre système.

Note : En général, un échec de ce test provient souvent d'une mauvaise génération de la bibliothèque due à l'utilisation de programmes trop vieux sur votre système. Il est donc parfois nécessaire de mettre à jour d'autres programmes avant d'effectuer la mise à jour de la bibliothèque C. Par exemple, votre version de `make` peut être obsolète, ainsi que celle de `GCC` ou encore celle des outils GNU binutils (assembleur, éditeur de liens, archiveur, etc.). Les sources de tous ces programmes peuvent être trouvées sur le [site du GNU](http://www.gnu.org).

La compilation de la bibliothèque C, ainsi que celle des autres composants « lourds » du système comme le noyau, `GCC`, `XWindow`, `Gnome` ou `KDE`, peuvent stresser votre système d'une manière qu'il n'a jamais connu. Les calculs intensifs peuvent durer plusieurs heures, ce qui fait chauffer les composants de votre ordinateur. Il n'est pas rare de voir certains de ces composants défaillir dans ces circonstances, alors qu'ils se sont toujours comportés apparemment normalement jusqu'à présent. En particulier, vous pouvez avoir des problèmes de surchauffe du processeur et des corruptions de données dans les barrettes de mémoire. Les symptômes sont en général l'apparition d'une erreur de type « SIG 11 » (erreur de segmentation) sur des programmes très fiables, comme `gcc` ou le `shell`. Dans ce cas, il faut identifier et remplacer les composants défectueux (ce n'est pas une blague, ce problème m'est personnellement arrivé avec trois barrettes mémoires et un processeur en moins d'un an, et il est arrivé à bien d'autres personnes dans le monde). Il est facile de penser que, avec la baisse du prix des barrettes mémoires que l'on a vécu ces derniers temps, les fabricants sont tentés d'être un peu plus laxistes au niveau du contrôle qualité des composants en fin de chaîne. Le phénomène risque donc de devenir courant, et il peut parfaitement vous arriver. Bien entendu, il va de soi qu'il ne faut pas overclocker sa machine lorsqu'on se lance dans des opérations telles que celles-ci. L'overclocking est de toute manière une technique douteuse dont le but n'est que d'arriver plus vite à avoir des problèmes curieux, surtout sous Linux. Vous êtes avertis.

L'installation se fera enfin avec les deux commandes suivantes :

```
make install
```

Note : ATTENTION ! La technique utilisée par le fichier `Makefile` pour l'installation de la bibliothèque C ne permet pas de remplacer la bibliothèque C du système en cours d'exécution de manière fiable. En pratique, cette commande ne fonctionnera que pour une mise à jour ou une installation de bibliothèque C compatible avec celle en cours d'exécution (c'est-à-dire compilée avec les mêmes options). Le remplacement de la bibliothèque en cours d'exécution par une bibliothèque compilée de même version, mais incompatible

échouera et laissera le système dans un état pitoyable. Il est recommandé, dans ce cas, de générer un paquetage pour votre distribution et d'utiliser les scripts d'installation fournis avec elle.

La commande suivante doit être ensuite utilisée pour installer l'ensemble des locales prises en charge par la bibliothèque C :

```
make localedata/install-locales
```

La compilation des paramètres internationaux pour la France se fera alors avec la commande suivante :

```
localedef -i fr_FR -f ISO-8859-15 fr_FR
```

Cette commande permet de compiler les fichiers des paramètres d'affichage des monnaies et des dates, ainsi que le jeu de caractères utilisé pour comparer et trier les chaînes de caractères. Dans l'exemple donné ci-dessus, la langue choisie est le français (« fr ») tel qu'il est parlé en France (« FR »), avec le jeu de caractères ISO-8859-15. Les fichiers compilés sont placés dans le répertoire `/usr/lib/locale/`, à raison d'un sous-répertoire pour chaque locale configurée. Si ce répertoire n'existe pas sur votre machine, vous devrez le créer manuellement, car la commande `localedef` ne le fait pas automatiquement.

Une fois la bibliothèque C installée, il est nécessaire de mettre à jour les fichiers d'en-tête du noyau utilisés pour la compilation des autres programmes. En effet, les fichiers d'en-tête du noyau utilisés pour la compilation de la bibliothèque C ne sont peut-être pas les mêmes que ceux de l'ancienne bibliothèque, et il est nécessaire de les remplacer pour que la compilation des autres programmes avec la nouvelle bibliothèque se fasse de manière cohérente. Ces fichiers d'en-têtes sont normalement placés dans les répertoires `/usr/src/linux/include/linux/`, `/usr/src/linux/include/asm/` et `/usr/src/linux/include/asm-generic/`. Ces deux derniers sont des liens symboliques vers les fichiers d'en-tête de la plate-forme matérielle cible du noyau (généralement, `/usr/src/linux/include/asm-i386/` et `/usr/src/linux/include/asm-generic/`). Ils doivent donc être recopiés dans le répertoire des en-têtes du système, à savoir le répertoire `/usr/include/`. Cela pourra être réalisé avec les commandes suivantes :

```
cd /usr/include
rm -rf linux asm*
( cd /usr/src/linux/include ; tar cf - linux/* asm/* asm-generic/* ; ) | tar xvf -
```

Note : Vous aurez peut-être à recompiler GCC après installation de la nouvelle bibliothèque C, ainsi qu'un certain nombre d'autres programmes, car la compatibilité binaire d'une version à l'autre de la bibliothèque C est très relative. Si la version que vous utilisez est un simple correctif de bug cependant, la compatibilité sera sans doute totale, mais il faut savoir qu'à chaque version majeure un certain nombre de fonctionnalités sont modifiées et peuvent nécessiter des mises à jour en cascade.

À partir de la version 2.3 de la bibliothèque C et de la version 2.13 des binutils, une amélioration dans le chargement des bibliothèques dynamiques a été apportée. Cette amélioration permet de « précharger » ces bibliothèques et donc de gagner du temps au démarrage des programmes. Pour pouvoir bénéficier de ces améliorations, il faut installer un outil complémentaire nommé « prelink ». Les sources de cet outil pourront être trouvées sur le compte de son auteur, à l'adresse <ftp://people.redhat.com/jakub/prelink/> [<ftp://people.redhat.com/jakub/prelink/>]. Sa compilation requiert que la bibliothèque `libelf`, permettant de manipuler les fichiers binaires, soit installée. Les sources de cette bibliothèque peuvent également être récupérés sur Internet, à l'adresse <http://www.stud.uni-hannover.de/~michael/software/libelf-0.8.2.tar.gz>. L'installation de cette bibliothèque et de `prelink` se fait simplement à l'aide des commandes suivantes :

```
./configure --prefix=/usr
make
make install
```

Note : Pour pouvoir utiliser correctement prelink, vous devez créer un fichier /etc/prelink.conf contenant la liste de tous les répertoires des bibliothèques de votre système, à raison d'une ligne par répertoire. Le contenu de ce fichier est donc typiquement le suivant :

```
# Exemple de fichier /etc/prelink.conf :  
/lib  
/usr/lib  
/usr/local/lib  
/usr/lib/qt/lib  
/opt/kde/lib
```

De plus, sachez que le préchargement des bibliothèques ne peut fonctionner que si toutes les bibliothèques dont dépend un programme directement ou indirectement (c'est-à-dire par l'intermédiaire d'éventuelles autres bibliothèques) ont été générées avec les binutils 2.13 ou plus. Cela implique qu'un certain nombre de composants systèmes (pour ne pas dire tous) devront être recompilés au préalable. Une mise à jour de votre distribution semble donc être la solution la plus simple à ce stade.

XIII-D - Compilation de OpenSSL

Les communications sur Internet n'ont jamais été considérées comme quelque chose de sûr. C'est pour cela que bon nombre de programmes ont recours au chiffrement des données pour assurer des communications fiables. Ce chiffrement est une nécessité vitale si vous désirez acheter quelque chose sur Internet ou consulter vos comptes en ligne, tant pour assurer votre protection que celle des intervenants commerciaux. La plupart des sites commerciaux refuseront l'accès aux clients non sécurisés.

La plupart des distributions utilisent, pour assurer le chiffrement, une bibliothèque fournissant un accès standard à la plupart des algorithmes de chiffrement : OpenSSL. Vous pourrez trouver cette bibliothèque sur [Internet](#).

La compilation d'OpenSSL n'est pas compliquée, car un jeu de fichier makefile pour Linux est fourni avec les sources. Vous devez simplement taper la commande suivante pour les sélectionner :

```
./Configure linux-elf shared
```

Note : Par défaut, OpenSSL va s'installer dans le répertoire /usr/local/ssl/, y compris les fichiers d'en-tête. Il faut donc ajouter le chemin /usr/local/ssl/ dans les variables d'environnement C_INCLUDE_PATH et CPLUS_INCLUDE_PATH si on veut compiler des programmes qui utilisent SSL (c'est en particulier le cas de KDE). Une autre solution est de demander une configuration explicite et d'utiliser l'option --prefix pour indiquer le répertoire de base pour l'installation, et l'option --openssldir pour indiquer le répertoire d'installation des utilitaires :

```
./Configure linux-elf shared --prefix=/usr --openssldir=/etc/ssl
```

pour effectuer l'installation des bibliothèques et des en-têtes dans les répertoires /usr/lib et /usr/include respectivement, et de placer les clefs privées et les fichiers de configuration dans le répertoire /etc/ssl.

Une fois la configuration effectuée, il suffit de taper les commandes suivantes pour effectuer la compilation et l'installation :

```
make  
make install
```

Comme d'habitude, vous pouvez effectuer un test avec la commande `make test` avant de réaliser l'installation pour vérifier que tout est correct.

XIII-E - Compilation de MESA 6.5.2

MESA est une bibliothèque compatible avec le standard OpenGL introduit par Silicon Graphics pour uniformiser les fonctions 3D des systèmes d'exploitation. X.org utilise MESA pour fournir les fonctions graphiques 3D accélérées ou simulées (selon votre matériel). Vous devrez donc recompiler MESA si vous voulez compiler X.org, ou si vous voulez effectuer une mise à jour.

L'installation de MESA ne pose pas de problème particulier, si ce n'est qu'il faut récupérer également l'archive des fichiers de démonstration, car elle contient la bibliothèque complémentaire GLUT que bon nombre de programmes OpenGL utilisent. L'archive de MESA et l'archive de ses programmes d'exemples peuvent être récupérées toutes les deux sur le site [Web de Mesa](http://www.mesa3d.org). Ces archives portent respectivement les noms `MesaLib-6.5.2.tar.bz2` et `MesaDemos-6.5.2.tar.bz2`.

Lorsque vous aurez récupéré ces archives, vous devrez les décompresser toutes les deux dans le même répertoire. Vous disposerez alors d'un sous-répertoire `Mesa-6.5.2`, dans lequel se trouvent tous les fichiers sources.

La compilation de MESA se fait simplement avec en exécutant la commande suivante :

```
make système
```

où `système` est le nom du système cible pour lequel MESA doit être compilé. Vous obtiendrez la liste des systèmes utilisables si vous n'en spécifiez aucun. Une fois la compilation effectuée, l'installation se fait manuellement, par copie des fichiers. Ceci se réalise simplement avec les deux commandes suivantes :

```
cp -r include/GL /usr/include
cp -pd lib/* /usr/lib
```

Note : Il est nécessaire de compiler les programmes de test pour que la bibliothèque GLUT soit installée. Pour cela, il faut que l'archive `MesaDemos-6.5.2.tar.bz2` ait été décompactée avant la compilation de la bibliothèque MESA.

Il est recommandé de supprimer les fichiers de l'ancienne version de la bibliothèque MESA fournie avec X.org. Ces fichiers sont situés dans le répertoire `/usr/lib/`, et portent des noms du type `libGL*`.

Si vous avez installé des bibliothèques permettant de bénéficier de l'accélération 3D matérielle de votre carte graphique, vous devrez supprimer les bibliothèques installées par MESA, et au besoin réinstaller vos bibliothèques prenant en charge votre carte graphique. Dans le cas contraire, les programmes utilisant OpenGL seront considérablement ralentis, et vous ne pourrez certainement plus les utiliser. Ce genre de situation se produit par exemple sur les machines disposant de cartes graphiques à base de puces nVidia, pour lesquelles vous devez utiliser les pilotes fournis par nVidia afin de bénéficier des accélérations 3D par le matériel.

XIII-F - Compilation de X.org 7.3.0

La compilation de X est certainement l'une des opérations les plus difficiles à réaliser, car ce logiciel est constitué d'une myriade de petits composants interdépendants. Initialement, X.org était monolithique et donc relativement facile à compiler, mais il a été rendu modulaire afin de faciliter le développement en permettant la compilation et la distribution de modules relativement indépendants. De ce fait, si la compilation d'un seul module de X.org est à présent une opération relativement facile sur un système où il est déjà installé, la compilation complète de X.org initiale est devenue très compliquée, parce qu'il faut compiler tous les modules un à un en respectant un ordre bien précis.

La compilation de X.org est donc quasiment inaccessible pour un particulier. Il s'agit typiquement d'une tâche qui est dévolue aux distributions et aux intégrateurs. Ainsi, je vous invite à utiliser les paquetages sources ou les scripts de compilation fournis par votre distribution, et de ne pas vous lancer dans une compilation complètement manuelle. La suite de cette section décrit toutefois la procédure générale à suivre pour information.

X.org requiert que d'autres logiciels indépendants soient installés. Les dépendances les plus importantes à ce sujet sont les bibliothèques Freetype, FontConfig, Mesa et libdrm. La compilation de Mesa a été présentée dans la section précédente. Celle des autres bibliothèques est décrite ci-dessous.

La bibliothèque libdrm est une bibliothèque permettant d'accéder aux fonctionnalités graphiques des cartes devant s'exécuter en mode noyau. C'est donc une bibliothèque d'interfaçage avec les pilotes DRM fournis par le noyau. Les sources de cette bibliothèque peuvent être trouvées sur le site de [Freedesktop](https://freedesktop.org/). La compilation et l'installation se font simplement avec le triplet configure, make et make install. Veillez à indiquer les bonnes options au programme configure, en particulier au niveau du répertoire d'installation cible (option --prefix).

Freetype est une bibliothèque de gestion des polices de caractères vectorielles, capable entre autres d'afficher les polices de caractères TrueType. Les sources de cette bibliothèque sont disponibles sur le site de [Freetype](https://freetype.org/). L'installation se fait simplement avec le triplet configure, make et make install. Toutefois, les sources de Freetype n'activent pas un algorithme essentiel pour l'affichage correct des polices TrueType par défaut, pour des raisons de brevets de logiciel détenus par Apple. Tant que ces brevets seront illégaux en Europe (ce qui ne saurait durer étant donné le lobbying effectué par les grandes sociétés), il est possible d'utiliser cet algorithme. Aussi est-il recommandé de l'activer, afin d'avoir un affichage correct des polices TrueType (sans cet algorithme, autant le dire tout de suite, ces polices sont inutilisables et l'installation de Freetype est déconseillée). Pour cela, il faut décommenter la ligne

```
#define TT_CONFIG_OPTION_BYTECODE_INTERPRETER
```

du fichier `ftoption.h` du répertoire `include/freetype/config/` des sources de Freetype. Une fois cette option modifiée, la compilation et l'installation se fera avec les trois commandes suivantes :

```
./configure --prefix=/usr
make
make install
```

Fontconfig est une bibliothèque de gestion des polices de caractères au sens général. Elle est maintenue par le projet [Freedesktop](https://freedesktop.org/). Elle peut être installée également très simplement, avec les trois commandes suivantes :

```
./configure --prefix=/usr --sysconfdir=/etc --mandir=/usr/share/man
make
make install
```

Note : Remarquez que pour certaines versions de cette bibliothèque, la documentation ne compile pas. La commande make signale donc une erreur. On peut malgré tout réaliser l'installation sans problème, et cela ne doit donc pas vous arrêter.

Une fois ces bibliothèques installées, la compilation de X.org se déroule classiquement avec les commandes configure et make. La plupart des fichiers sources de la distribution X.org sont fournis par la fondation [X.org](https://www.x.org/). Vous devrez procéder comme suit pour chacun des modules à compiler :

- décompressez le fichier d'archive du module ;
- allez dans le répertoire des sources du module ;
- exécutez la commande configure avec les options adéquat (en particulier, l'option --prefix qui indique le répertoire cible où X.org sera installé, ainsi que les autres options définissant les répertoires des bibliothèques et des fichiers de configuration) ;
- compilez et installez le module avec les commandes make et make install.

Note : La compilation des documentations de X.org nécessite que vous disposiez des outils de génération de documentation SMGL. Ces outils comprennent OpenJade, un traducteur SGML utilisant des feuilles de styles DSSSL, et les feuilles de styles associées. Ces outils sont normalement fournis avec toute bonne distribution, leur installation ne sera donc pas traitée dans ce document. Notez toutefois qu'il faut s'assurer que des liens symboliques jade et nsgmls existent et référencent respectivement les programmes openjade et onsgmls pour que la compilation des documentations puisse bien se passer.

Comme on le voit, la principale difficulté dans la compilation de X.org réside dans la détermination de l'ordre de compilation des différents modules, ainsi que des options spécifiques éventuelles à fournir à la commande configure. Vous pourrez vous inspirer du script build-from-tarball.sh fourni par X.org sur son Wiki (section développeurs) pour la génération à partir des sources, afin de déterminer l'ordre de compilation des modules et les options à utiliser.

La procédure de génération détaillée est donnée ci-dessous à titre indicatif :

- installation des macros pour le programme aclocal (paquetage util-macos) ;
- installation de la documentation (paquetages xorg-`sgml-doctools` et `xorg-docs`) ;
- installation des définitions de protocoles X (paquetages `bigreqproto`, `compositeproto`, `damageproto`, `dmxproto`, `evieproto`, `fixesproto`, `fontcacheproto`, `fontproto`, `glproto`, `inputproto`, `kbproto`, `pmproto`, `printproto`, `randrproto`, `recordproto`, `renderproto`, `resourceproto`, `scnrsaverproto`, `trapproto`, `videoproto`, `windowwmpproto`, `x11proto`, `xcmiscproto`, `xextproto`, `xf86bigfontproto`, `xf86dgaproto`, `xf86driproto`, `xf86miscproto`, `xf86rushproto`, `xf86vidmodeproto`, `xineramaproto`, `xproto` et `xproxymanagementprotocol`) ;
- compilation et installation des bibliothèques X (paquetages `xtrans`, `libXau`, `libXdmcp`, `libX11`, `libXext`, `libAppleWM`, `libWindowsWM`, `libdmx`, `libfontenc`, `libFS`, `libICE`, `liblxxutil`, `libboldX`, `libSM`, `libXt`, `libXmu`, `libXpm`, `libXp`, `libXaw`, `libXfixes`, `libXcomposite`, `libXrender`, `libXdamage`, `libXcursor`, `libXevie`, `libXfont`, `libXfontcache`, `libXft`, `libXi`, `libXinerama`, `libxkbfile`, `libxkbui`, `libXprintUtil`, `libXprintAppUtil`, `libXrandr`, `libXres`, `libXScrnSaver`, `libXTrap`, `libXtst`, `libXv`, `libXvMC`, `libXxf86dga`, `libXxf86misc`, `libXxf86vm` et `libpciaccess`) ;
- installation des images fournies avec X (paquetage `xbitmaps`) ;
- compilation et installation des applications et utilitaires fournis avec X (paquetage `appres`, `bdf2pcpf`, `beforelight`, `bitmap`, `editres`, `fonttosfnt`, `fslsfonts`, `fstobdf`, `iceauth`, `ico`, `lbxproxy`, `listres`, `luit`, `mkfontdir`, `mkfontscale`, `oclock`, `proxymngr`, `rgb`, `randercheck`, `rstart`, `scripts`, `sessreg`, `setxkbmap`, `showfont`, `smproxy`, `twm`, `viewres`, `x11perf`, `xauth`, `xbiff`, `xcalc`, `xclipboard`, `xclock`, `xcmsdb`, `xconsole`, `xcursorgen`, `xdbedizzy`, `xditview`, `xdm`, `xdpyinfo`, `xdriinfo`, `xedit`, `xev`, `xeyes`, `xf86dga`, `xfindproxy`, `xfd`, `xfontsel`, `xfs`, `xfinfo`, `xfwp`, `xgamma`, `xgc`, `xhost`, `xinit`, `xkbcomp`, `xkbevd`, `xkbprint`, `xkbutils`, `xkill`, `xload`, `xlogo`, `xlsatoms`, `xlsclients`, `xlsfonts`, `xmag`, `xman`, `xmessage`, `xmh`, `xmodmap`, `xmore`, `xpshellworld`, `xplsprinters`, `xpr`, `xprehashprinterlist`, `xprop`, `xrandr`, `xrdb`, `xrefresh`, `xrx`, `xset`, `xsetmode`, `xsetpointer`, `xsetroot`, `xsm`, `xstdcmap`, `xtrap`, `xvidtune`, `xvinfo`, `xwd`, `xwininfo` et `xwud`) ;
- compilation et installation de l'exécutable principal du serveur X (paquetage `xorg-server`, à configurer avec l'option `-with-mesa-source` pour indiquer le répertoire des sources de la version de Mesa qui a été installée) ;
- compilation et installation des pilotes de périphériques d'entrée/sortie (paquetages `xf86-input-aiptek`, `xf86-input-evdev`, `xf86-input-ur98`, `xf86-input-vmmouse`, `xf86-input-acecad`, `xf86-input-calcomp`, `xf86-input-citron`, `xf86-input-digitaledge`, `xf86-input-dmc`, `xf86-input-dynapro`, `xf86-input-elo2300`, `xf86-input-elographics`, `xf86-input-fpit`, `xf86-input-hyperpen`, `xf86-input-jamstudio`, `xf86-input-joystick`, `xf86-input-keyboard`, `xf86-input-magellan`, `xf86-input-magictouch`, `xf86-input-microtouch`, `xf86-input-mouse`, `xf86-input-mutouch`, `xf86-input-palmax`, `xf86-input-penmount`, `xf86-input-spaceorb`, `xf86-input-summa`, `xf86-input-tek4957` et `xf86-input-void`) ;
- compilation et installation des pilotes de cartes graphiques (paquetages `xf86-video-sisusb`, `xf86-video-sunffb`, `xf86-video-v4l`, `xf86-video-apm`, `xf86-video-ark`, `xf86-video-ast`, `xf86-video-ati`, `xf86-video-chips`, `xf86-video-cirrus`, `xf86-video-cyrix`, `xf86-video-dummy`, `xf86-video-fbdev`, `xf86-video-glide`, `xf86-video-glint`, `xf86-video-i740`, `xf86-video-intel`, `xf86-video-via`, `xf86-video-i128`, `xf86-video-imstt`, `xf86-video-mga`, `xf86-video-neomagic`, `xf86-video-newport`, `xf86-video-nsc`, `xf86-video-nv`, `xf86-video-rendition`, `xf86-video-s3`, `xf86-video-s3virge`, `xf86-video-savage`, `xf86-video-siliconmotion`, `xf86-video-sis`, `xf86-video-sunbw2`, `xf86-video-suncg14`, `xf86-video-suncg3`, `xf86-video-suncg6`, `xf86-video-sunleo`, `xf86-video-suntcx`, `xf86-video-tdfx`, `xf86-video-tga`, `xf86-video-trident`, `xf86-video-tseng`, `xf86-video-vesa`, `xf86-video-vga`, `xf86-video-vmware`, `xf86-video-voodoo`, `xf86-video-xgi` et `xf86-video-xgixp`) ;
- installation des fichiers de données additionnels (paquetage `xcursor-themes`) ;
- compilation et installation des outils de gestion des polices de caractères et des fichiers de définition des encodages (paquetages `font-util` et `encodings`) ;

- installation des fichiers de polices de caractères (paquetages font-adobe-100dpi, font-adobe-75dpi, font-adobe-utopia-100dpi, font-adobe-utopia-75dpi, font-adobe-utopia-type1, font-arabic-misc, font-bh-100dpi, font-bh-75dpi, font-bh-lucidatypewriter-100dpi, font-bh-lucidatypewriter-75dpi, font-bh-ttf, font-bh-type1, font-bitstream-100dpi, font-bitstream-75dpi, font-bitstream-speedo, font-bitstream-type1, font-cronyx-cyrillic, font-cursor-misc, font-daewoo-misc, font-dec-misc, font-ibm-type1, font-isas-misc, font-jis-misc, font-micro-misc, font-misc-cyrillic, font-misc-ethiopic, font-misc-meltho, font-misc-misc, font-mutt-misc, font-schumacher-misc, font-screen-cyrillic, font-sony-misc, font-sun-misc, font-winitzki-cyrillic, font-xfree86-type1 et font-alias) ;
- compilation et installation des outils de développement (paquetages xorg-cf-files, imake, makedepend, gccmakedep et Indir).

XIII-G - Compilation de Lesstif 0.95.0

Lesstif est un clone libre de la bibliothèque de composants graphiques Motif, qui est l'une des bibliothèques les plus utilisées en programmation graphique sous X11, mais dont l'utilisation est soumise à une licence commerciale. Lesstif est donc à Motif ce que Linux est à Unix en quelque sorte. Étant donné le grand nombre de programmes qui utilisent Motif, il est nécessaire de l'installer sur votre machine. La dernière version actuellement disponible est la 0.93.41. Vous pourrez en trouver les sources sur le [site de Lesstif](#).

La compilation et l'installation de Lesstif s'effectuent classiquement, à l'aide des programmes configure et make. Vous aurez donc à exécuter les commandes suivantes dans le répertoire d'installation des sources de Lesstif :

```
./configure --prefix=/usr
make
make install
```

Ce jeu de commandes installera Lesstif dans le répertoire `/usr/lib/LessTif`. Bien entendu, vous êtes libre d'utiliser un autre emplacement, mais il vous faudra alors modifier les chemins de recherche des fichiers d'en-tête et des bibliothèques lorsque vous voudrez utiliser des programmes qui en ont besoin.

XIII-H - Compilation de KDE 3.5.7

KDE utilise XWindow et la bibliothèque Qt. Il faut donc disposer des bibliothèques et des fichiers d'en-tête de XWindow, et compiler la bibliothèque Qt au préalable pour compiler KDE. La version de Qt recommandée pour compiler KDE 3.5.7 est la version 3.3.8. Les paragraphes qui suivent supposent que vous disposez de cette version.

Si la bibliothèque Qt est fournie avec votre distribution, son répertoire d'installation sera `/usr/lib/`. Dans le cas contraire, elle sera placée dans `/usr/local/lib/`. La compilation se fait sur place, ce qui permet d'éviter d'avoir à l'installer. Elle se déroule de la manière suivante :

- il faut se placer dans le répertoire `/usr/lib/` :

```
cd /usr/lib
```

- les sources doivent ensuite être extraites de l'archive de la bibliothèque Qt :

```
tar xvfz qt-x11-free-3.3.8.tar.gz
```

- les sources sont extraites dans le répertoire `qt-x11-free-3.3.8/`. Il faut renommer ce répertoire en `qt/` (c'est-à-dire, supprimer les numéros de version) ou faire un lien symbolique `qt` vers ce répertoire (solution permettant de conserver les anciennes bibliothèques, pour les anciens programmes) ;
- il faut ensuite ajouter les variables d'environnement dans le `.profile` ou `.login` (selon le shell utilisé, pour bash, il faut modifier `.profile`) comme indiqué dans le fichier `INSTALL` du répertoire `qt`. Ces variables d'environnement sont utilisées par le processus de compilation. Vous devrez bien entendu corriger les chemins définis par défaut pour référencer le répertoire d'installation des sources de Qt ;
- il faut recharger le fichier `.profile` ou `.login` afin de définir ces variables ;
- lancer la commande `configure` avec les options suivantes :


```
./configure -shared -thread -no-g++-exceptions -no-stl -system-zlib
-qt-gif -system-libpng -system-libjpeg -plugin-imgfmt-mng -xft
```

- lancer la commande make pour construire la bibliothèque :

```
make
```

- lancer ldconfig.

Outre la bibliothèque Qt, KDE utilise également quelques bibliothèques utilitaires qu'il est fortement recommandé d'installer. Il s'agit principalement de la bibliothèque pcre (<http://www.pcre.org>), qui permet de gérer les expressions rationnelles, de la bibliothèque Xine (<http://xinehq.de>), qui permet de jouer des fichiers vidéo de manière indépendante de leur format, et d'un certain nombre de bibliothèques du gestionnaire de bureau Gnome (<http://www.gnome.org>) :

- la bibliothèque libaudiofile, qui permet de manipuler les fichiers sons ;
- les bibliothèques libxml et libxslt, qui permettent d'analyser les documents au format XML ;
- et des bibliothèques libart et ImLib, qui permettent de manipuler les différents formats d'image de manière uniforme.

Ces bibliothèques peuvent être compilées de manière classique, à l'aide du classique triplet configure, make et make install.

D'autres bibliothèques devront être installées pour bénéficier de fonctionnalités complémentaires. Cependant, l'installation de ces bibliothèques reste facultative et n'est pas nécessaire à la compilation complète de KDE. Certaines d'entre elles sont cependant très intéressantes, et leur installation devra être considérée en fonction de vos besoins. Les bibliothèques et utilitaires utilisés par les différents composants de KDE sont listés ci-dessous

- Le paquetage kdelib utilise la bibliothèque mDNS (disponible sur <http://www.opensource.apple.com/darwinsource/tarballs/apsl>) pour permettre l'utilisation automatique des appareils réseau compatible avec le protocole ZeroConf.
- Le paquetage kdatabase utilise la bibliothèque d'authentification simplifiée SASL (<http://asg.web.cmu.edu/sasl/index.html>) pour l'authentification des protocoles de consultation et d'envoi de mails sécurisés.
- Le paquetage kdegraphics ne prendra en charge les métainformations des fichiers PDF que si xpdf est installé (<http://www.foolabs.com/xpdf/download.html>). L'accès aux appareils photo numérique se fait via la bibliothèque gphoto (<http://www.gphoto.org/>). La prise en charge des fichiers SVG requiert quant à elle la bibliothèque de gestion de couleurs lcms (<http://www.littlecms.com>) et la bibliothèque fribidi, qui implémente l'algorithme de gestion des textes bidirectionnels d'Unicode (<http://fribidi.sf.net>). Enfin, la prise en charge des scanners requiert l'installation de Sane (<http://www.sane-project.org>).
- Outre la bibliothèque libxine pour la lecture des fichiers vidéo, le paquetage kdemultimedia requiert utilise les outils mpg123 et Lame (disponibles respectivement à <http://www.mpg123.de> et <http://lame.sourceforge.net>) pour permettre l'encodage et le décodage des fichiers MP3, ainsi que les bibliothèques Ogg Vorbis et FLAC (disponibles à <http://www.xiph.org/ogg/vorbis/download.html> et <http://flac.sourceforge.net>) pour la prise en charge du fichier compressé en Ogg Vorbis ou via FLAC. La bibliothèque tablib permettra également de lire et de modifier les métainformations ID3 des fichiers Audio qui le supportent (<http://ktown.kde.org/~wheeler/taglib>). Enfin, la bibliothèque libmusicbrainz (<http://www.musicbrainz.org/products/client/download.html>) vous permettra d'obtenir les noms des morceaux à partir de leurs identifiants, via les serveurs MusicBrainz.
- Le paquetage kdenetwork utilise les outils Wireless pour les réseaux sans-fil (http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html). L'utilisation des outils de déport d'affichage et de prise de contrôle à distance requiert l'installation de openssl, protocole d'exploration et de localisation des services réseau (<http://www.openssl.org>).
- Le paquetage kdepim vous permettra de manipuler votre téléphone portable via la bibliothèque libgnokii et ses gestionnaires de périphériques (<http://www.gnokii.org>). Les programmes de synchronisation avec les Palms requièrent quant à eux la bibliothèque libmal (<http://jasonday.home.att.net/code/libmal>).
- L'environnement de développement kdevelop pourra générer des documentations à l'aide de Doxygen si celui-ci est installé (<http://www.doxygen.org>), et des diagrammes de dépendances entre les composants d'un logiciel si Dot est présent (<http://www.graphviz.org>).

Une fois les bibliothèques utilitaires et la bibliothèque Qt compilées, on peut s'attaquer à KDE lui-même. La compilation de KDE suppose que les variables d'environnement de la bibliothèque Qt sont toujours définies, il ne faut donc pas les supprimer après avoir compilé celle-ci.

KDE se configure classiquement, avec le programme de configuration configure. Pour chaque composant de KDE, il faut procéder comme suit :

- extraire les fichiers sources de l'archive ;
- appeler configure avec les bonnes options ;
- lancer la compilation avec « make » ;
- installer les fichiers binaires avec « make install ».

Sur certaines distributions, KDE est installé dans le répertoire `/opt/kde/`. Sur d'autres, il est considéré comme faisant partie intégrante du système et est installé dans le répertoire `/usr/`. Il faut donc spécifier les bons préfixes lors de l'appel à configure. Par exemple, pour installer KDE dans le répertoire `/usr/`, il faut utiliser la commande suivante :

```
./configure --prefix=/usr
```

Comme d'habitude, il faut éventuellement indiquer le type de machine avec l'option `--host` et demander l'utilisation des bibliothèques dynamiques avec l'option `--enable-shared`.

L'ordre des opérations lors de la compilation de KDE est important, parce que certaines parties de KDE en utilisent d'autres. En particulier, il faut impérativement compiler et installer les bibliothèques en premier. Il est donc recommandé d'opérer dans l'ordre suivant :

- arts ;
- kdelib ;
- kdebindings ;
- kdatabase ;
- les autres composants de KDE, kdenetwork devant être compilé avant kdepim, et kdeaddons devant être compilé en dernier.

Note : Si l'on veut que la bibliothèque Xine puisse utiliser le serveur de son arts, il est nécessaire de la compiler après celui-ci. Toutefois, elle devra être compilée avant le paquetage kdemultimedia, qui l'utilise pour la lecture des fichiers vidéo).

Note : Contrairement aux autres parties du système, il faut compiler KDE dans le répertoire des sources. En effet, certaines bibliothèques fournies avec les sources sont nécessaires pour la compilation correcte de quelques modules non essentiels de KDE.

D'autre part, si KDE est déjà présent sur votre système, il faut le supprimer avant de le recompiler (ou au moins renommer son répertoire si vous n'êtes pas sûr de vous...). En effet, les fichiers d'en-tête et les bibliothèques de l'ancienne version peuvent faire échouer la compilation de la nouvelle version.

Si vous le supprimez complètement, il se réinstallera avec la configuration par défaut. Les scripts de configuration de votre distribution pourront alors très bien ne plus fonctionner sur la configuration de KDE. Dans ce cas, vous aurez certainement quelques problèmes avec les droits fixés par défaut sur certains programmes. En particulier, vous ne pourrez jouer des fichiers son que sous le compte root, car le serveur de son « artsd » de KDE et ses outils connexes ont besoin de privilèges pour utiliser les fichiers spéciaux de périphérique de la carte son. La manière la plus propre pour régler ce problème est de créer un groupe d'utilisateurs nommé `audio` et contenant tous les utilisateurs ayant le droit d'utiliser la carte son, et d'attribuer à ce groupe les fichiers spéciaux de périphérique `/dev/dsp`, `/dev/audio`, `/dev/mixer` et tous les fichiers spéciaux de périphériques du répertoire `/dev/snd/`. Les droits d'accès en lecture et écriture sur ces fichiers devront ensuite être donnés au propriétaire (root) et au groupe d'utilisateurs (`audio`). Normalement, les autres utilisateurs ne doivent avoir aucun droit sur ces fichiers.

Note : KDE 3.5.7 et Qt 3.3.8 sont capables d'utiliser les fonctionnalités les plus avancées des serveurs X. Ces fonctionnalités peuvent apporter un plus indéniable, et il est recommandé que XWindow soit correctement installé avant de lancer leur compilation. En particulier, il est recommandé d'avoir compilé et installé la bibliothèque FreeType 2, pour le support des polices TrueType et l'antialiasing des polices de caractères, ainsi que la bibliothèque OpenGL MESA.

Note : Enfin, sachez que KDE 3.5.7 nécessite la version 1.4.2 ou plus de l'environnement d'exécution java pour permettre l'utilisation de java et de JavaScript dans le navigateur Konqueror. Cet environnement peut être téléchargé gratuitement sur le [site de Sun](#). Les binaires fournis par Sun ne sont pas des logiciels libres, mais ce sont les seuls qui respectent réellement la norme Java (ce qui est logique, puisqu'elle est imposée par Sun). L'installation de cet environnement d'exécution se fait extrêmement simplement. Il est recommandé d'installer les binaires dans le répertoire `/usr/lib/jre.x.y.z` et de créer un lien symbolique `/usr/lib/java` vers ce répertoire, afin de permettre l'installation de plusieurs versions de l'environnement d'exécution java tout en sélectionnant la version x.y.z par défaut. Après installation, vous devrez ajouter le répertoire `/usr/lib/java/bin` aux répertoires de votre variable d'environnement PATH pour permettre l'utilisation de Java par les programmes tels que Konqueror. Enfin, le script de lancement de la machine virtuelle utilise les commandes `cut` et `head`, et suppose qu'elles sont installées dans le répertoire `/usr/bin/`, alors que la plupart des distributions les placent directement dans le répertoire `/bin/`. Vous aurez donc peut-être à faire des liens symboliques `cut` et `head` sur ces deux commandes dans le répertoire `/usr/bin/`.

Notez que le support de Java doit également être activé manuellement dans la configuration de Konqueror pour que vous puissiez utiliser toutes les fonctionnalités de Java.

XIII-I - Compilation de Gnome 2.20.0

Contrairement à KDE, Gnome utilise sa propre bibliothèque graphique, qui est elle-même basée sur XWindow. La compilation de Gnome requiert donc de compiler cette bibliothèque ainsi que les bibliothèques utilitaires qu'il utilise avant de compiler les applications.

Le nombre de modules constituant l'environnement Gnome est très important, essentiellement parce que cela permet des installations et des mises à jour partielles, mais cela induit une grande complexité dans les dépendances entre les modules et requiert un ordre de génération relativement strict. Malheureusement, tout comme pour X.org, cet ordre de compilation n'est pas ou très peu documenté, cette opération n'étant supposée être faite que par les distributions. De plus, les développeurs de Gnome ne s'imposent pas des contraintes d'intégration très strictes, et il n'est pas rare qu'une bibliothèque supposée être utilisée avec une version de Gnome ne compile pas, ne fonctionne pas correctement, ou ne soit tout simplement pas fournie. Il faut donc aller à la pêche pour retrouver les composants nécessaires et suffisants, dans la bonne version, avant de parvenir à compiler Gnome. La complexité est telle que certaines distributions ne fournissent plus que les bibliothèques de base et laissent de côté le reste de l'environnement.

Toutefois, les dernières versions de Gnome sont fournies avec un script de génération nommé `Garnome`. L'utilisation de ce script est recommandée, mais il ne fonctionnera que sur un système récent, en raison des dépendances fortes des bibliothèques de Gnome envers d'autres modules non fournis directement par le projet Gnome. En particulier, un certain nombre de ces modules proviennent du projet Free Desktop (<http://www.freedesktop.org>), mais cette règle n'est pas absolue. D'autres bibliothèques externes sont également requises, et ne sont pas listées dans les dépendances de Gnome.

XIII-I-1 - Récupération des sources

Les fichiers d'archives de Gnome 2.20.0 peuvent être téléchargés directement à partir du site du projet Gnome (<http://www.gnome.org>) ou à partir de l'un de ses miroirs. Certaines archives non listées dans la version courante doivent être récupérées dans les répertoires des anciennes versions de Gnome.

Les sources de Gnome sont réparties dans trois répertoires. Le répertoire `platform/` contient les sources des bibliothèques de base de Gnome, qu'il est conseillé d'installer même si on n'utilise pas l'environnement Gnome lui-même, car elles sont souvent utilisées par des programmes indépendants. Le répertoire `desktop/` contient les sources

des programmes de l'environnement de bureau Gnome lui-même. Le répertoire `bindings/` contient les sources des bibliothèques permettant de programmer et de commander les programmes Gnome dans différents langages de programmation. Enfin, le répertoire `admin/` contient des programmes complémentaires pour configurer et administrer Gnome dans le contexte d'un déploiement en entreprise.

XIII-I-2 - Compilation des bibliothèques générales

La première étape dans la compilation de Gnome est donc d'installer les bibliothèques de base fournies par `Freedesktop` ou des projets tiers, ainsi que les bibliothèques de base de Gnome lui-même. L'ordre d'installation recommandé est le suivant :

- `pkgconfig` (bibliothèque de gestions de paquets sources, disponible chez [Freedesktop](#)) ;
- `guile` (implémentation du langage Scheme, disponible sur [le site du projet GNU](#)) ;
- `autogen` (utilitaire d'automatisation de génération de code répétitif, disponible sur [le site du projet autogen](#)) ;
- `liboil` (bibliothèque d'opérations classiques optimisées pour les différents types de processeurs, disponible sur [le site de Freedesktop](#)) ;
- `fontconfig` (bibliothèque de gestion de la configuration des polices de caractères, disponible sur [le site de Freedesktop](#)) ;
- `libXft` (bibliothèque d'affichage des polices TrueType de X11, disponible sur [le site de Freedesktop](#)) ;
- `libgpg-error` (bibliothèque GNU de gestion des erreurs relatives au logiciel cryptographique GnuPG, disponible sur [le site du projet GNU](#)) ;
- `libgcrypt` (bibliothèque GNU de gestion des algorithmes cryptographiques du logiciel GnuPG, disponible sur [le site du projet GNU](#)) ;
- `libtasn1` (bibliothèque GNU de gestion des certificats en notation ASN.1, disponible sur [le site du projet GNU](#)) ;
- `opencdk` (bibliothèque GNU de gestion des messages du protocole cryptographique utilisé par le logiciel GnuPG, disponible sur [le site du projet GNU](#)) ;
- `gnutls` (bibliothèque GNU de gestion des communications chiffrées, disponible sur [le site du projet GNU](#)) ;
- `libxml2` (bibliothèque d'analyse des flux XML, disponible sur [le site de Gnome](#)) ;
- `libxslt` (bibliothèque de transformation des flux XML, disponible sur [le site de Gnome](#)) ;
- `gnome-common` (utilitaires divers nécessaires à la compilation des paquets Gnome, mais non requis à l'exécution, disponible sur [le site de Gnome](#)) ;
- `intltool` (bibliothèque d'aide à l'internationalisation des programmes, disponible sur [le site de Gnome](#)) ;
- `rarian` (utilitaires divers pour l'indexation de la documentation, disponible sur [le site de Gnome](#) [<ftp://ftp.gnome.org/pub/gnome/sources/rarian/0.6/rarian-0.6.0.tar.bz2>]) ;
- `gtk-doc` (utilitaires d'aide à la génération de la documentation du code source en langage C, disponible sur [le site de Gnome](#)) ;
- `glib` (bibliothèque utilitaire complémentaire à la bibliothèque C, disponible sur [le site de Gnome](#)) ;
- `shared-mime-info` (déclarations de type de fichiers MIME standards utilisables par les différents environnements de bureau, disponible sur [le site de Freedesktop](#)) ;
- `hicolor-icon-theme` (jeu d'icônes de base pour les différents environnements de bureau, disponible sur [le site de Freedesktop](#)) ;
- `cairo` (bibliothèque de rendu 2D vectoriel et disposant de plusieurs périphériques de sortie, disponible sur [le site de Freedesktop](#)) ;
- `pango` (bibliothèque d'internationalisation de gestion du texte, disponible sur [le site de Gnome](#)) ;
- `gnome-doc-utils` (utilitaires nécessaires pour la génération des documentations de Gnome, disponible sur [le site de Gnome](#)) ;
- `atk` (bibliothèque de gestion de l'accessibilité, disponible sur [le site de Gnome](#)) ;
- `gtk+` (bibliothèque graphique générale de Gnome et de nombreux autres programmes, disponible sur [le site de Gnome](#)) ;
- `dbus` (bibliothèque de communication interprocessus bas niveau utilisée par les composants systèmes, disponible sur [le site de Freedesktop](#)) ;
- `dbus-glib` (bibliothèque d'intégration de DBUS dans la bibliothèque générale GLib, disponible sur [le site de Gnome](#)) ;
- `libvolume_id` (bibliothèque Linux de lecture des informations des périphériques de stockage de masse, disponible sur [le site du noyau Linux](#)) ;
- `hal` (bibliothèque de gestion du matériel sous Linux, disponible sur [le site de Freedesktop](#)) ;
- `audiofile` (bibliothèque de gestion du son, disponible sur [le site de la bibliothèque audiofile](#)) ;

- **libart_lgpl** (bibliothèque graphique 2D, disponible sur [le site de Gnome](#) ;
- **libmusicbrainz** (bibliothèque d'identification des fichiers musicaux par consultation d'un serveur MusicBrainz, disponible sur [le site de la bibliothèque Musicbrainz](#) ;
- **libdaemon** (bibliothèque utilitaire pour réaliser des démons Unix, disponible sur [le site de la bibliothèque libdaemon](#) ;
- **libgtop** (bibliothèque portable de gestion des processus, disponible sur [le site de Gnome](#) ;
- **gamin** (bibliothèque de gestion des notifications de changements sur les systèmes de fichiers, disponible sur [le site de Gnome](#) ;
- **libIDL** (bibliothèque de gestion des fichiers de description d'interfaces CORBA pour l'environnement CORBA ORBit2, disponible sur [le site de Gnome](#) ;
- **ORBit2** (système à composants CORBA pour le langage C, disponible sur [le site de Gnome](#) ;
- **gconf** (bibliothèque de gestion de base de données de configuration Gnome, disponible sur [le site de Gnome](#) ;
- **poppler** (bibliothèque générique de manipulation et de rendu des fichiers PDF, disponible sur [le site de Freedesktop](#) ;
- **libglade** (bibliothèque de gestion d'interfaces graphiques Gnome basées sur des ressources XML, disponible sur [le site de Gnome](#) ;
- **firefox** (navigateur Internet Firefox, disponible sur [le site de la fondation Mozilla](#) ;
- **xdg-user-dirs** (bibliothèque de gestion des répertoires du bureau avec prise en charge de l'internationalisation, disponible sur [le site de Freedesktop](#) ;
- **icon-naming-utils** (utilitaires permettant d'installer les icônes de manière standard pour une utilisation correcte par les différents environnements de bureau, disponible sur [le site du projet Tango](#) ;
- **iso-codes** (listes des codes ISO de pays et de régions, disponibles sur le site des sources de la distribution Debian [<ftp://pkg-isocodes.alioth.debian.org/pub/pkg-isocodes/iso-codes-0.53.tar.bz2>] ;
- **desktop-file-utils** (utilitaires pour la manipulation des fichiers sur le bureau, disponible sur [le site de Freedesktop](#) ;
- **startup-notification** (bibliothèque de gestion du protocole de notification de démarrage des applications des environnements de bureau, disponible sur [le site de Freedesktop](#) ;
- **libcolorblind** (bibliothèque de gestion des couleurs contrastées pour les daltoniens, disponible sur [le site des sources de la distribution Debian](#) ;
- **enchant** (bibliothèque générique de gestion des correcteurs orthographiques, disponible sur [le site du projet AbiWord](#) ;
- **pwlib** (bibliothèque de fonctions générales portables sur les systèmes Unix et Windows, disponible sur [le site de Gnome](#) ;
- **opal** (bibliothèque d'abstraction pour la téléphonie sur Internet, disponible sur [le site de Gnome](#) ;

Toutes ces bibliothèques et ces programmes peuvent être installés simplement en exécutant avec les commandes classiques `configure`, `make` et `make install`. Il est possible d'installer ces bibliothèques dans un autre répertoire que le répertoire par défaut en spécifiant le répertoire destination à l'aide de l'option `--prefix` lors de la compilation de chaque module. Ces bibliothèques et outils étant utilisés par un grand nombre de programmes (et pas uniquement par Gnome), il est recommandé de les installer globalement dans le système. Dans le cas d'une installation non standard, il est impératif dans ce cas d'ajouter le répertoire des binaires de Gnome dans la variable d'environnement `PATH`, ainsi que le répertoire des bibliothèques de Gnome dans la variable d'environnement `LD_LIBRARY_PATH` pour que la compilation des autres modules qui les utilisent se déroule sans problème. Il est également nécessaire d'ajouter le répertoire de stockage des informations sur les paquetages sources dans la variable d'environnement `PKG_CONFIG_PATH`.

Note : Le navigateur Internet `firefox` ne fait pas à proprement parler partie de Gnome, et encore moins de ses bibliothèques. Toutefois, l'environnement de Gnome utilise certaines de ses bibliothèques d'une part, et il s'appuie sur la bibliothèque graphique `Gtk+` de Gnome. De ce fait, son installation est un prérequis et il est considéré ici comme une des dépendances de base de Gnome.

`DBUS` utilise un démon de communication qu'il est conseillé d'exécuter dans un compte utilisateur spécifique pour des raisons de sécurité. Par défaut, cet utilisateur est l'utilisateur `messagebus`, qui doit appartenir au groupe d'utilisateur du même nom. Enfin, il faut s'assurer que le démon de gestion du bus de communication est effectivement lancé avant la compilation des modules suivants. Des scripts de démarrage système sont fournis avec les sources de

DBUS pour les principales distributions, il vous faudra les installer à l'endroit prévu à cet effet et lancer le démon avant de poursuivre la compilation.

La bibliothèque DBUS a changé le nom d'une de ses fonctions récemment, ce qui empêche la compilation des outils de la bibliothèque HAL. Il faut donc corriger les noms de cette fonction dans la bibliothèque HAL. Ainsi, il vous faudra remplacer toutes les occurrences de « `dbus_connection_disconnect` » en « `dbus_connection_close` » dans les fichiers sources de HAL. Par ailleurs, la compilation de la documentation de HAL ne se fait pas correctement. Cela n'est toutefois pas gênant et n'empêche pas l'installation des binaires avec la commande `make install`.

Vous devrez peut-être indiquer à Gnome où se trouve la DTD XML de DocBook une fois les bibliothèques libxml et libxslt installées. Cela peut être réalisé à l'aide des commandes suivantes :

```
xmlcatalog --noout --create /etc/xml/catalog
xmlcatalog --noout --add public "-//OASIS//DTD DocBook XML V4.1.2//EN" \
/usr/share/sgml/docbook/xml-dtd-4.1.2/docbookx.dtd /etc/xml/catalog
```

La première commande permet de créer un fichier de configuration vierge pour l'enregistrement des DTDs au niveau de la bibliothèque libxml, et la deuxième permet d'ajouter la DTD DocBook dans ce fichier. Bien entendu, vous ne devez exécuter la première commande que si le fichier `/etc/xml/catalog` n'existe pas déjà. Le chemin indiqué pour le fichier `docbookx.dtd` donné ici l'est à titre d'exemple et doit bien entendu être modifié en fonction de l'emplacement où vous avez installé la DTD DocBook.

XIII-I-3 - Compilation des modules de base d'interfaçage avec les langages

Une fois les bibliothèques de base installées, il est recommandé d'installer les bibliothèques d'interfaçage avec les différents langages de programmation utilisés pour les autres modules de Gnome. En particulier, l'installation des bibliothèques d'interfaçage avec le langage Python et le langage C# (environnement Mono) sont fortement recommandés. Les bibliothèques pour s'interfacer avec le langage C++ sont nettement moins utilisées.

Les bibliothèques suivantes permettent essentiellement d'accéder aux fonctionnalités des bibliothèques de base avec le langage C++ :

- `libsigc++2` (bibliothèque de gestion des signaux génériques pour le langage C++, disponible sur [le site de Gnome](#) ;
- `glibmm` (bibliothèque d'interface avec la bibliothèque Glib pour le langage C++, disponible sur [le site de Gnome](#) ;
- `libxml++` (bibliothèque d'interface avec la bibliothèque libxml pour le langage C++, disponible sur [le site de Gnome](#) ;
- `cairomm` (bibliothèque d'interface avec la bibliothèque Cairo pour le langage C++, disponible sur [le site du projet Cairo](#) ;
- `gtkmm` (bibliothèque d'interface avec la bibliothèque GTK pour le langage C++, disponible sur [le site de Gnome](#) ;
- `libglademmm` (bibliothèque d'interface avec la bibliothèque libglade pour le langage C++, disponible sur [le site de Gnome](#) ;

Les bibliothèques suivantes permettent essentiellement d'accéder aux fonctionnalités des bibliothèques de base avec le langage python :

- `dbus-python` (bibliothèque d'interface avec DBUS pour le langage Python, disponible sur [le site de Freedesktop](#) ;
- `pycairo` (bibliothèque d'interface avec la bibliothèque Cairo pour le langage Python, disponible sur [le site du projet Cairo](#) ;
- `pygobject` (bibliothèque d'interface avec la bibliothèque Glib pour le langage Python, disponible sur [le site de Gnome](#) ;

- pygtk (bibliothèque d'interface avec la bibliothèque GTK pour le langage Python, disponible sur [le site de Gnome](#) ;
- pyorbit (bibliothèque d'interface avec la bibliothèque ORBit pour le langage Python, disponible sur [le site de Gnome](#) ;

La bibliothèque suivante permet d'accéder aux fonctionnalités de Gtk avec le langage C# sous l'environnement Mono :

- gtk-sharp (disponible sur [le site de Gnome](#)).

XIII-I-4 - Compilation des bibliothèques de Gnome

Les bibliothèques de cette section sont les bibliothèques de base de Gnome, qu'il faut impérativement installer pour compiler les modules de Gnome restants.

- avahi (bibliothèque de détection de services sur le réseau, disponible sur [le site de la bibliothèque Avahi](#) ;
- gnome-mime-data (déclarations de type de fichiers MIME complémentaires pour l'environnement de bureau Gnome, disponible sur [le site de Gnome](#) ;
- gnome-icon-theme (jeu d'icônes par défaut de Gnome, disponible sur [le site de Gnome](#) ;
- esound (serveur de son pour Gnome, disponible sur [le site de Gnome](#) ;
- gail (bibliothèque d'abstraction des bibliothèques de gestion de l'accessibilité, disponible sur [le site de Gnome](#) ;
- gnome-vfs (bibliothèque d'abstraction des différentes sources de données permettant une manipulation uniforme des flux de données, disponible sur [le site de Gnome](#) ;
- libbonobo (bibliothèque de gestion des documents composites, disponible sur [le site de Gnome](#) ;
- gnome-vfs-monikers (bibliothèque de gestion des noms de ressources abstraites pour les flux gnome-vfs, disponible sur [le site de Gnome](#) ;
- libgnome (bibliothèque de base de Gnome, disponible sur [le site de Gnome](#) ;
- libgnomecanvas (bibliothèque de gestion d'objets graphiques en 2D sur un canevas, disponible sur [le site de Gnome](#) ;
- libbonoboui (bibliothèque graphique de gestion des documents composites de Gnome, disponible sur [le site de Gnome](#) ;
- gnome-keyring (gestionnaire de mots de passe et de clefs secrètes de Gnome, disponible sur [le site de Gnome](#) ;
- libgnomeui (bibliothèque graphique de base de Gnome, disponible sur [le site de Gnome](#) ;
- libgnomecups (bibliothèque de gestion d'objets graphiques en 2D, disponible sur [le site de Gnome](#) ;
- libgnomeprint (bibliothèque de gestion des fonctions d'impression de Gnome, disponible sur [le site de Gnome](#) ;
- libgnomeprintui (bibliothèque graphique de gestion des fonctions d'impression de Gnome, disponible sur [le site de Gnome](#) ;
- libcroco (bibliothèque de gestion des feuilles de style CSS pour Gnome, disponible sur [le site de Gnome](#) ;
- libgsf (bibliothèque de lecture de fichiers structurés, tels que les fichiers OLE de Microsoft ou les fichiers .zip, disponible sur [le site de Gnome](#) ;
- libxklavier (bibliothèque Gnome pour la gestion du clavier sous X, disponible sur [le site de Gnome](#) ;
- at-spi (bibliothèque d'interface pour l'accessibilité sous Gnome, disponible sur [le site de Gnome](#) ;
- gtk-engines (collection de thèmes pour Gtk, disponible sur [le site de Gnome](#) ;
- gnome-themes (jeu de thèmes pour Gnome, disponible sur [le site de Gnome](#) ;
- gnome-desktop (bibliothèques et modules de base de Gnome, disponible sur [le site de Gnome](#) ;
- librsvg (bibliothèque graphique de rendu de fichiers SVG, disponible sur [le site de Gnome](#) ;
- gnome-menus (jeu de menus standards pour Gnome, disponible sur [le site de Gnome](#) ;
- eel (bibliothèque de fonctions générales complémentaire des bibliothèques de base de Gnome, disponible sur [le site de Gnome](#) ;
- system-tools-backends (bibliothèques de gestion des tâches d'administration du système, disponible sur [le site de Gnome](#) ;
- gstreamer (bibliothèque multimédia généraliste, disponible sur [le site de Gnome](#) ;
- gst-plugins-base (modules complémentaires pour gstreamer, disponible sur [le site de Gnome](#) ;
- gst-plugins-good (modules complémentaires pour gstreamer, disponible sur [le site de Gnome](#) ;

- gnome-backgrounds (jeu d'arrière-plans pour le bureau de Gnome, disponible sur [le site de Gnome](#) ;
- gnome-speech (bibliothèque Gnome de fonctions dédiées à la synthèse vocale pour la lecture de textes, disponible sur [le site de Gnome](#) ;
- libsoup (bibliothèque de gestion du protocole HTTP, disponible sur [le site de Gnome](#) ;
- gtkhtml (bibliothèque de rendu graphique de pages HTML de Gnome, disponible sur [le site de Gnome](#) ;
- gtksourceview (bibliothèque de visualisation et de mise en forme de fichiers sources de Gnome, disponible sur [le site de Gnome](#) ;
- gtksourceview-1.0 (ancienne bibliothèque de visualisation et de mise en forme de fichiers sources de Gnome, disponible sur [le site de Gnome](#) ;
- libwnck (bibliothèque de gestion des notifications d'événements relatifs à l'affichage des documents multipage, disponible sur [le site de Gnome](#) ;
- evolution-data-server (serveur de données personnelles et de gestion des agendas, disponible sur [le site de Gnome](#) ;
- gnome-panel (barre de tâches de Gnome, disponible sur [le site de Gnome](#) ;
- libgail-gnome (bibliothèque de gestion de l'accessibilité de Gnome, disponible sur [le site de Gnome](#) ;
- libgnomekbd (bibliothèque de configuration du clavier sous Gnome, disponible sur [le site de Gnome](#) ;
- liboobs (bibliothèque d'interface objet pour Gnome des utilitaires systèmes, disponible sur [le site de Gnome](#) ;

XIII-I-5 - Compilation de Gnome

La compilation et l'installation des composants principaux de Gnome se fait ensuite également à l'aide des commandes configure, make et make install. L'ordre proposé ci-dessous permet de faire cette installation en respectant les dépendances entre les composants, mais n'est pas structuré par fonctionnalité comme dans les sections précédentes.

Les modules constituant le corps du gestionnaire de bureau Gnome sont les suivants :

- evolution (logiciel de messagerie, de gestion de contact et d'agenda de Gnome, disponible sur [le site de Gnome](#) ;
- evolution-exchange (module de communication avec Microsoft Exchange pour evolution, disponible sur [le site de Gnome](#) ;
- evolution-webcal (utilitaire de gestion des calendriers Web pour evolution, disponible sur [le site de Gnome](#) ;
- metacity (gestionnaire de fenêtres de Gnome, disponible sur [le site de Gnome](#) ;
- gdm2 (gestionnaire de connexions de Gnome, disponible sur [le site de Gnome](#) ;
- fast-user-switch-applet (module de commutation d'utilisateur de Gnome, disponible sur [le site de Gnome](#) ;
- bug-buddy (outil de rapport de bogues de Gnome, disponible sur [le site de Gnome](#) ;
- dasher (utilitaire de saisie pour l'accessibilité, disponible sur [le site de Gnome](#) ;
- nautilus (gestionnaire de fichiers de Gnome, disponible sur [le site de Gnome](#) ;
- nautilus-cd-burner (module de gestion des graveurs de CD pour nautilus, disponible sur [le site de Gnome](#) ;
- gnome-control-center (panneau de contrôle de Gnome, disponible sur [le site de Gnome](#) ;
- alacarte (éditeur de menus de Gnome, disponible sur [le site de Gnome](#) ;
- gconf-editor (éditeur de configuration du système, disponible sur [le site de Gnome](#) ;
- pygtksourceview (bibliothèque d'interface avec la bibliothèque gtksourceview pour le langage Python, disponible sur [le site de Gnome](#) ;
- gedit (éditeur de texte de Gnome, disponible sur [le site de Gnome](#) ;
- gnome-media (suite de logiciels multimédia de Gnome, disponible sur [le site de Gnome](#) ;
- totem (programme de lecture multimédia de Gnome, disponible sur [le site de Gnome](#) ;
- gnome-python (bibliothèque d'interface pour Gnome pour le langage Python, disponible sur [le site de Gnome](#) ;
- gnome-python-desktop (interfaces pour quelques bibliothèques de Gnome pour le langage Python, disponible sur [le site de Gnome](#) ;
- deskbar-applet (applet de requêtes de recherches documentaires, disponible sur [le site de Gnome](#) ;
- ekiga (logiciel de communication sur Internet, anciennement appelé Gnomemeeting, disponible sur [le site de Gnome](#) ;
- eog (logiciel de visualisation d'images de Gnome, disponible sur [le site de Gnome](#) ;
- epiphany (navigateur Web de Gnome, disponible sur [le site de Gnome](#) ;

- evince (logiciel de visualisation de documents, disponible sur [le site de Gnome](#) ;
- file-roller (gestionnaire d'archives et de fichiers compressés de Gnome, disponible sur [le site de Gnome](#) ;
- gcalctool (calculatrice de Gnome, disponible sur [le site de Gnome](#) ;
- gucharmap (table de caractères de Gnome, disponible sur [le site de Gnome](#) ;
- gnome-applets (applets diverses pour la barre de tâches de Gnome, disponible sur [le site de Gnome](#) ;
- gnome-games (jeux de Gnome, disponible sur [le site de Gnome](#) ;
- gnome-keyring-manager (interface du gestionnaire de clefs et de mots de passe de Gnome, disponible sur [le site de Gnome](#) ;
- gnome-netstatus (applet permettant d'afficher l'état de la connexion réseau, disponible sur [le site de Gnome](#) ;
- gnome-nettool (suite de logiciels de gestion du réseau sous Gnome, disponible sur [le site de Gnome](#) ;
- gnome-mag (outil de grossissement de Gnome, disponible sur [le site de Gnome](#) ;
- gnome-power-manager (gestionnaire d'énergie des portables sous Gnome, disponible sur [le site de Gnome](#) ;
- gnome-screensaver (jeu d'économiseurs d'écran de Gnome, disponible sur [le site de Gnome](#) ;
- gnome-session (gestionnaire de sessions utilisateur de Gnome, disponible sur [le site de Gnome](#) ;
- gnome-system-monitor (gestionnaire de tâches de Gnome, disponible sur [le site de Gnome](#) ;
- gnome-system-tools (outils graphiques d'administration du système, disponible sur [le site de Gnome](#) ;
- vte (contrôle d'accès aux terminaux virtuels, disponible sur [le site de Gnome](#) ;
- gnome-terminal (émulateur de terminal de Gnome, disponible sur [le site de Gnome](#) ;
- gnome-user-docs (base de la documentation de Gnome, disponible sur [le site de Gnome](#) ;
- gnome-utils (utilitaires divers de Gnome, disponible sur [le site de Gnome](#) ;
- gnome-volume-manager (gestionnaires de périphériques de stockage de masse de Gnome, disponible sur [le site de Gnome](#) ;
- gok (clavier virtuel de Gnome, disponible sur [le site de Gnome](#) ;
- orca (logiciel de prise d'écran de Gnome, disponible sur [le site de Gnome](#) ;
- seahorse (frontal Gnome pour le logiciel cryptographique GnuPG, disponible sur [le site de Gnome](#) ;
- sound-juicer (logiciel d'extraction de pistes audio, disponible sur [le site de Gnome](#) ;
- tomboy (logiciel de prise de notes de Gnome, disponible sur [le site de Gnome](#) ;
- vino (serveur d'accès à distance compatible VNC pour Gnome, disponible sur [le site de Gnome](#) ;
- yelp (logiciel d'affichage de l'aide de Gnome, disponible sur [le site de Gnome](#) ;
- zenity (outil d'affichage de boîtes de dialogue scriptable en ligne de commande, disponible sur [le site de Gnome](#) ;
- xchat (logiciel de communication en temps réel multiplateforme, disponible sur [le site de Gnome](#) ;
- camorama (logiciel de visiophonie pour Gnome, disponible sur [le site de Gnome](#) ;

XIII-I-6 - Compilation des bindings de Gnome

Une fois les modules de Gnome générés, il est possible d'installer les bibliothèques complémentaires de liaisons avec les langages.

Les bibliothèques suivantes permettent d'accéder aux fonctionnalités des bibliothèques de Gnome avec le langage C++ :

- gconfmm (bibliothèque d'interface avec la bibliothèque GConf pour le langage C++, disponible sur [le site de Gnome](#) ;
- libgnomecanvasmm (bibliothèque d'interface avec la bibliothèque libgnomecanvas pour le langage C++, disponible sur [le site de Gnome](#) ;
- gnome-vfsmm (bibliothèque d'interface avec la bibliothèque gnome-vfs pour le langage C++, disponible sur [le site de Gnome](#) ;
- libgnomemm (bibliothèque d'interface pour Gnome pour le langage C++, disponible sur [le site de Gnome](#) ;
- libgnomeuimm (bibliothèque d'interface avec la bibliothèque libgnomeui pour le langage C++, disponible sur [le site de Gnome](#) ;

XIII-I-7 - Compilation des outils de développements complémentaires

L'environnement Gnome fournit des outils de développements complémentaires, qui permettent de modifier la configuration du bureau et de l'adapter à des besoins spécifiques. Ces outils sont accessibles en Une fois les modules de Gnome générés, il est possible d'installer les bibliothèques complémentaires de liaisons avec les langages.

Les modules de ces outils sont les suivants :

- pessulus (utilitaire de verrouillage de la configuration de Gnome, disponible sur [le site de Gnome](#) ;
- sabayon (utilitaire de création des profils utilisateurs pour Gnome, disponible sur [le site de Gnome](#) ;
- devhelp (utilitaire d'affichage de la documentation Gnome pour les développeurs, disponible sur [le site de Gnome](#) ;
- glade3 (utilitaire de création d'interfaces graphiques basée sur libglade pour Gnome, disponible sur [le site de Gnome](#) ;
- accerciser (utilitaire de manipulation des options d'accessibilité de Gnome, disponible sur [le site de Gnome](#) ;
- gnome-devel-docs (documentation pour les développeurs Gnome, disponible sur [le site de Gnome](#) ;
- gob2 (préprocesseur permettant de faciliter l'écriture d'objets Gtk+ en C, disponible sur [le site de Gnome](#) ;

XIII-J - Compilation de Samba 3.0.23

La compilation de Samba 3.0.23 ne pose désormais plus de problèmes particuliers. Cependant, le choix des répertoires par défaut utilisés par Samba n'est pas très judicieux sur une machine Linux et pourra être éventuellement modifié.

La configuration se fait classiquement avec configure :

```
CFLAGS="-O2 -pipe" ./configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var \
--with-smbmount --with-smbwrapper
```

L'option `--with-smbmount` permet de compiler également les utilitaires `smbmount`, `smbmnt` et `smbumount`, qui permettent de monter les volumes partagés comme des systèmes de fichiers classiques. De même, l'option `--with-smbwrapper` permet de créer une bibliothèque dynamique que les programmes peuvent utiliser pour accéder directement aux partages SMB, qu'ils trouveront dans un répertoire virtuel `/smb/`.

Une fois la configuration faite, il est recommandé de modifier les chemins déclarés dans le fichier `makefile`. En effet, ces chemins ne sont pas forcément ceux spécifiés au programme de configuration. Les variables suivantes devront donc éventuellement être redéfinies :

```
# Répertoire des fichiers de configuration de Samba :
CONFIGDIR = ${sysconfdir}

# Répertoire privé pour les fichiers de mots de passe :
PRIVATEDIR=/etc/samba
```

La compilation se fait simplement avec la commande suivante :

```
make
```

et l'installation avec :

```
make install
```

Notez cependant que les outils `smbmnt` et `smbumount` ne sont pas `setuid`. Leurs droits doivent donc être modifiés avec les deux commandes suivantes :

```
chmod +s /usr/bin/smbmnt
```

```
chmod +s /usr/bin/smbmount
```

XIV - Annexe C. Formulaire pour la création des lignes de mode de X.org

Les formules données dans cette annexe donnent les relations entre les différentes valeurs utilisées dans les lignes de mode de X.org. Elles peuvent vous aider à comprendre les relations entre les principaux paramètres de ces lignes.

Ces formules utilisent des variables qui représentent les valeurs des lignes de mode et les caractéristiques techniques de votre écran. Le tableau donné ci-dessous liste les variables utilisées :

Variable	Signification
DC	« Dot Clock », fréquence de base du balayage. Représente le nombre de pixels balayés par seconde et sert de base de temps pour tous les calculs.
BW	« Bandwidth », bande passante de fonctionnement du moniteur. Représente la limite supérieure de la fréquence de base du balayage que le moniteur peut accepter. Cette valeur fait partie des caractéristiques techniques du moniteur. Elle peut être extrapolée à partir de la résolution maximale que le moniteur peut afficher.
GCMF	« Graphic Card Maximum Frequency », la plus grande valeur de fréquence de balayage que la carte graphique peut générer. Cette valeur fait partie des caractéristiques techniques de la carte graphique.
MHF	« Max Horizontal Frequency », fréquence maximale du balayage horizontal. C'est la plus forte valeur de la plage de fréquences horizontales pour les moniteurs multisynchrones. Cette donnée fait partie des caractéristiques techniques du moniteur.
MVF	« Max Vertical Frequency », fréquence maximale du balayage vertical. C'est la plus forte valeur de la plage de fréquences verticales pour les moniteurs multisynchrones. Cette donnée fait partie des caractéristiques techniques du moniteur.
RR	« Refresh Rate », taux de rafraîchissement des images du mode graphique. C'est le nombre d'images générées par seconde pour ce mode graphique.
HR	« Horizontal Resolution », résolution horizontale du mode graphique. C'est le nombre de pixels visibles dans ce mode graphique.
HFL	« Horizontal Frame Length », longueur totale d'une ligne dans ce mode. C'est

	le nombre de pixels total du mode graphique, visibles ou non. Les pixels non visibles sont les pixels de la zone de blanking horizontal.
HST	« Horizontal Sync Time », durée minimale des signaux de synchronisation du balayage horizontal. Cette donnée fait partie des caractéristiques techniques du moniteur.
HSL	« Horizontal Sync Length », longueur des signaux de synchronisation du balayage horizontal, exprimée en pixels. Cette donnée peut être déduite directement de la durée des signaux et de la fréquence de base du balayage.
HBT	« Horizontal Blanking Time », durée minimale du blanking horizontal dans ce mode graphique. C'est la durée pendant laquelle le faisceau électronique doit être éteint. Cette donnée fait partie des caractéristiques techniques du moniteur.
HBL	« Horizontal Blanking Length », longueur totale du blanking en pixels. Cette donnée peut être déduite directement de la durée du blanking et de la fréquence de base du balayage.
VR	« Vertical Resolution », résolution verticale du mode graphique. C'est le nombre de lignes visibles dans ce mode graphique.
VFH	« Vertical Frame Height », hauteur totale d'une image dans ce mode. C'est le nombre de lignes total du mode graphique, lignes non visibles comprises. Les lignes non visibles sont les lignes de la zone de blanking vertical.
VST	« Vertical Sync Time », durée minimale des signaux de synchronisation du balayage vertical. Cette donnée fait partie des caractéristiques techniques du moniteur.
VSH	« Vertical Sync Height », nombre de lignes balayées pendant le signal de synchronisation du balayage vertical. Cette donnée peut être déduite de la durée des signaux de synchronisation du balayage vertical, de la longueur totale d'une ligne et de la fréquence de base du balayage.
VBT	« Vertical Blanking Time », durée minimale du blanking vertical dans ce mode graphique. C'est la durée pendant laquelle le faisceau électronique doit être éteint avant et après un retour de balayage vertical. Cette donnée fait

	partie des caractéristiques techniques du moniteur.
VBH	« Vertical Blanking Height », nombre total de lignes du blanking vertical. Cette donnée peut être déduite de la durée du blanking vertical, de la longueur totale d'une ligne et de la fréquence de base du balayage.

Certaines des variables mentionnées dans ce tableau peuvent être directement déduites des autres. C'est en particulier le cas de RR, HSL, HVL, VSH et VBH.

Le taux de rafraîchissement dépend bien entendu de la fréquence de base du balayage et du nombre total de pixels à balayer. Il peut donc être calculé directement avec la formule suivante :

$$RR = DC \cdot (HFL \cdot VFH)$$

La longueur des signaux de synchronisation horizontale et la longueur du blanking dépendent de la fréquence de base du balayage et de leurs durées respectives. Elles peuvent donc être calculées avec les formules suivantes :

$$\begin{aligned} HSL &= HST \cdot DC \\ HBL &= HBT \cdot DC \end{aligned}$$

Le même raisonnement peut être fait pour le nombre de lignes parcourues pendant le signal de synchronisation du balayage vertical et le blanking. Cependant, il ne faut pas utiliser la fréquence de base directement quand on calcule sur les lignes, car cette fréquence est exprimée en pixels par seconde et les lignes contiennent HFL pixels. Il faut donc diviser cette fréquence par HFL pour obtenir la fréquence en nombres de lignes par seconde. Les formules à utiliser seront donc les suivantes :

$$\begin{aligned} VSH &= VST \cdot (DC \cdot HFL) \\ VBH &= VBT \cdot (DC \cdot HFL) \end{aligned}$$

Par ailleurs, la longueur totale des lignes doit bien entendu être supérieure à la résolution plus la taille de la zone de blanking horizontal. Le même raisonnement peut être appliqué au nombre de lignes total, à la résolution verticale et au nombre de lignes utilisées pour le blanking. On disposera donc des relations suivantes :

$$\begin{aligned} HFL &\geq HR + HBL \\ VFH &\geq VR + VBH \end{aligned}$$

Le choix des valeurs des lignes de mode est soumis à un certain nombre de contraintes, qui expriment les limitations techniques du moniteur et de la carte graphique. Ces contraintes sont exprimées ci-dessous.

Il va de soi que les signaux de synchronisation de balayage doivent être compris dans le blanking. Les deux premières contraintes sont donc les suivantes :

$$\begin{aligned} HSL &\geq HBL \\ VSH &\geq VBH \end{aligned}$$

Par ailleurs, la fréquence de balayage de base doit bien entendu être tolérée à la fois par le moniteur et par la carte graphique. Elle doit donc vérifier les deux inégalités suivantes :

$$\begin{aligned} DC &\leq BP \\ DC &\leq GCMF \end{aligned}$$

Enfin, il faut que les fréquences de balayage horizontales et verticales utilisées soient inférieures aux fréquences de balayage maximales que le moniteur peut gérer. Cette règle n'est valide que pour les moniteurs multisynchrones. Pour

les moniteurs à fréquences fixes, il faut que les fréquences de balayage horizontale et verticale soient strictement égales à celles du moniteur.

La contrainte horizontale donne une troisième condition sur DC :

$$(DC \cdot HFL) \cdot MHF$$

Or :

$$(DC \cdot HFL) \cdot MHF = (DC \cdot (HR + HBL))$$

donc l'inégalité précédente est forcément vérifiée si on s'assure que :

$$DC \cdot (HR + HBL) \cdot MHF$$

soit :

$$\begin{aligned} DC \cdot MHF &\geq (HR + HBL) \\ DC \cdot MHF &\geq (HR + DC \cdot HBL) \end{aligned}$$

d'où la condition suivante :

$$DC \cdot (MHF \cdot HR) \geq (1 - MHF \cdot HBL)$$

(le dénominateur étant positif en pratique).

La contrainte verticale porte sur le taux de rafraîchissement, qui doit être inférieur à la fréquence verticale maximale. Elle permet d'obtenir une troisième condition sur DC :

$$RR = DC / (HFL \cdot VFH) \cdot MVF$$

soit :

$$DC \cdot MVF \cdot HFL \cdot VFH = MVF \cdot (HR + HBL) \cdot (VR + VBH) = MVF \cdot (HR + DC \cdot HBL) \cdot (VR + DC \cdot VBT)$$

ce qui donne l'inéquation suivante :

$$\begin{aligned} MVF \cdot HBL \cdot VBT \cdot DC &+ (MVF \cdot HR \cdot VBT + MVF \cdot VR \cdot HBL \\ &- 1) \cdot DC + \\ &MVF \cdot HR \cdot VR > 0 \end{aligned}$$

Cette inéquation impose une limite inférieure en dessous de laquelle DC peut se trouver, ou une limite supérieure au-delà de laquelle il peut se trouver. En pratique, la limite supérieure ne peut jamais être atteinte. Il faut donc que la fréquence de base soit inférieure à la première racine de l'équation associée à l'inéquation donnée ci-dessous. Cette équation est du second degré et sa résolution ne pose pas de problème particulier. Certains moniteurs ont un temps nul pour les signaux de synchronisation verticaux. Comme le second membre de cette équation est négatif en pratique, la limite à respecter pour la fréquence de balayage de base est exprimée dans ce cas par l'inéquation suivante :

$$DC \cdot (MVF \cdot HR \cdot VR) \geq (1 - MVF \cdot (HR \cdot VBT + VR \cdot HBL))$$

XV - Annexe D. GNU Free Documentation License

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document « free » in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of « copyleft », which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The « Document », below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as « you ».

A « Modified Version » of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A « Secondary Section » is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The « Invariant Sections » are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The « Cover Texts » are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A « Transparent » copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not « Transparent » is called « Opaque ».

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for

human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The « Title Page » means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, « Title Page » means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

- B List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- C State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D Preserve all the copyright notices of the Document.
- E Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H Include an unaltered copy of this License.
- I Preserve the section entitled « History », and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled « History » in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the « History » section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K In any section entitled « Acknowledgements » or « Dedications », preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M Delete any section entitled « Endorsements ». Such a section may not be included in the Modified Version.
- N Do not retitle any existing section as « Endorsements » or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled « Endorsements », provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or

publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled « History » in the various original documents, forming one section entitled « History »; likewise combine any sections entitled « Acknowledgements », and any sections entitled « Dedications ». You must delete all sections entitled « Endorsements. »

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an « aggregate », and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License « or any later version » applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the

Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

XVI - Annexe E. Licence de documentation libre GNU

Disclaimer

This is an unofficial translation of the GNU Free Documentation License into French. It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU FDL--only the original English text of the GNU FDL does that. However, we hope that this translation will help French speakers understand the GNU FDL better.

Ceci est une traduction française non officielle de la Licence de documentation libre GNU. Elle n'a pas été publiée par la Free Software Foundation, et ne fixe pas légalement les conditions de redistribution des documents qui l'utilisent -- seul le texte original en anglais le fait. Nous espérons toutefois que cette traduction aidera les francophones à mieux comprendre la FDL GNU.

Traduction française non officielle de la GFDL Version 1.1 (mars 2000)

Copyright original :

Copyright (C) 2000 Free Software Foundation, inc

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Pour la traduction :

Version 1.0 FR (Jean Luc Fortin, juillet 2000)

Version 1.1 FR (Christian Casteyde, mars 2001)

Version 1.1.1 FR (César Alexanian, mars 2001)

Version 1.1.2 FR (Christian Casteyde et César Alexanian, mars 2001)

Version 1.1.3 FR (Christian Casteyde, avril 2001)

Chacun est libre de copier et de distribuer des copies conformes de cette Licence, mais nul n'est autorisé à la modifier.

0 - PRÉAMBULE

L'objet de cette Licence est de rendre tout manuel, livre ou autre document écrit « libre » au sens de la liberté d'utilisation, à savoir : assurer à chacun la liberté effective de le copier ou de le redistribuer, avec ou sans modifications, commercialement ou non. En outre, cette Licence garantit à l'auteur et à l'éditeur la reconnaissance de leur travail, sans qu'ils soient pour autant considérés comme responsables des modifications réalisées par des tiers.

Cette Licence est une sorte de « copyleft », ce qui signifie que les travaux dérivés du document d'origine sont eux-mêmes « libres » selon les mêmes termes. Elle complète la Licence Publique Générale GNU, qui est également une Licence copyleft, conçue pour les logiciels libres.

Nous avons conçu cette Licence pour la documentation des logiciels libres, car les logiciels libres ont besoin d'une documentation elle-même libre : un logiciel libre doit être accompagné d'un manuel garantissant les mêmes libertés que celles accordées par le logiciel lui-même. Mais cette Licence n'est pas limitée aux seuls manuels des logiciels ; elle peut être utilisée pour tous les documents écrits, sans distinction particulière relative au sujet traité ou au mode

de publication. Nous recommandons l'usage de cette Licence principalement pour les travaux destinés à des fins d'enseignement ou devant servir de documents de référence.

1 - APPLICABILITÉ ET DÉFINITIONS

Cette Licence couvre tout manuel ou tout autre travail écrit contenant une notice de copyright autorisant la redistribution selon les termes de cette Licence. Le mot « Document » se réfère ci-après à un tel manuel ou travail. Toute personne en est par définition concessionnaire, et est référencée ci-après par le terme « Vous ».

Une « Version modifiée » du Document désigne tout travail en contenant la totalité ou seulement une portion de celui-ci, copiée mot pour mot, modifiée et/ou traduite dans une autre langue.

Une « Section secondaire » désigne une annexe au Document, ou toute information indiquant les rapports entre l'auteur ou l'éditeur et le sujet (ou tout autre sujet connexe) du document, sans toutefois être en rapport direct avec le sujet lui-même (par exemple, si le Document est un manuel de mathématiques, une Section secondaire ne traitera d'aucune notion mathématique). Cette section peut contenir des informations relatives à l'historique du Document, des sources documentaires, des dispositions légales, commerciales, philosophiques, ou des positions éthiques ou politiques susceptibles de concerner le sujet traité.

Les « Sections inaltérables » sont des sections secondaires considérées comme ne pouvant être modifiées et citées comme telles dans la notice légale qui place le Document sous cette Licence.

Les « Textes de couverture » sont les textes courts situés sur les pages de couverture avant et arrière du Document, et cités comme tels dans la mention légale de ce Document.

Le terme « Copie transparente » désigne une version numérique du Document représentée dans un format dont les spécifications sont publiquement disponibles et dont le contenu peut être visualisé et édité directement et immédiatement par un éditeur de texte quelconque, ou (pour les images composées de pixels) par un programme de traitement d'images quelconque, ou (pour les dessins) par un éditeur de dessins courant. Ce format doit être pouvoir être accepté directement ou être convertible facilement dans des formats utilisables directement par des logiciels de formatage de texte. Une copie publiée dans un quelconque format numérique ouvert mais dont la structure a été conçue dans le but exprès de prévenir les modifications ultérieures du Document ou dans le but d'en décourager les lecteurs n'est pas considérée comme une Copie Transparente. Une copie qui n'est pas « Transparente » est considérée, par opposition, comme « Opaque ».

Le format de fichier texte codé en ASCII générique et n'utilisant pas de balises, les formats de fichiers Texinfo ou LaTeX, les formats de fichiers SGML ou XML utilisant une DTD publiquement accessible, ainsi que les formats de fichiers HTML simple et standard, écrits de telle sorte qu'ils sont modifiables sans outil spécifique, sont des exemples de formats acceptables pour la réalisation de Copies Transparentes. Les formats suivants sont opaques : PostScript, PDF, formats de fichiers propriétaires qui ne peuvent être visualisés ou édités que par des traitements de textes propriétaires, SGML et XML utilisant des DTD et/ou des outils de formatage qui ne sont pas disponibles publiquement, et du code HTML généré par une machine à l'aide d'un traitement de texte quelconque et dans le seul but de la génération d'un format de sortie.

La « Page de titre » désigne, pour les ouvrages imprimés, la page de titre elle-même, ainsi que les pages supplémentaires nécessaires pour fournir clairement les informations dont cette Licence impose la présence sur la page de titre. Pour les travaux n'ayant pas de Page de titre comme décrit ci-dessus, la « Page de titre » désigne le texte qui s'apparente le plus au titre du document et situé avant le texte principal.

2 - COPIES CONFORMES

Vous pouvez copier et distribuer le Document sur tout type de support, commercialement ou non, à condition que cette Licence, la notice de copyright et la notice de la Licence indiquant que cette Licence s'applique à ce Document soient reproduites dans toutes les copies, et que vous n'y ajoutiez aucune condition restrictive supplémentaire. Vous ne pouvez pas utiliser un quelconque moyen technique visant à empêcher ou à contrôler la lecture ou la reproduction

ultérieure des copies que vous avez créées ou distribuées. Toutefois, vous pouvez solliciter une rétribution en échange des copies. Si vous distribuez une grande quantité de copies, référez-vous aux dispositions de la section 3.

Vous pouvez également prêter des copies, sous les mêmes conditions que celles précitées, et vous pouvez afficher publiquement des copies de ce Document.

3 - COPIES EN NOMBRE

Si vous publiez des copies imprimées de ce Document à plus de 100 exemplaires, et que la Licence du Document indique la présence de Textes de couverture, vous devez fournir une couverture pour chaque copie, qui présente les Textes de couverture des première et dernière pages de couverture du Document. Les première et dernière pages de couverture doivent également vous identifier clairement et sans ambiguïté comme étant l'éditeur de ces copies. La première page de couverture doit comporter le titre du Document en mots d'importance et de visibilité égales. Vous pouvez ajouter des informations complémentaires sur les pages de couverture. Les copies du Document dont seule la couverture a été modifiée peuvent être considérées comme des copies conformes, à condition que le titre du Document soit préservé et que les conditions indiquées précédemment soient respectées.

Si les textes devant se trouver sur la couverture sont trop importants pour y tenir de manière claire, vous pouvez ne placer que les premiers sur la première page et placer les suivants sur les pages consécutives.

Si vous publiez plus de 100 Copies opaques du Document, vous devez soit fournir une Copie transparente pour chaque Copie opaque, soit préciser ou fournir avec chaque Copie opaque une adresse réseau publiquement accessible d'une Copie transparente et complète du Document, sans aucun ajout ou modification, et à laquelle tout le monde peut accéder en téléchargement anonyme et sans frais, selon des protocoles réseau communs et standards. Si vous choisissez cette dernière option, vous devez prendre les dispositions nécessaires, dans la limite du raisonnable, afin de garantir l'accès non restrictif à la Copie transparente durant une année pleine après la diffusion publique de la dernière Copie opaque (directement ou via vos revendeurs).

Nous recommandons, mais ce n'est pas obligatoire, que vous contactiez l'auteur du Document suffisamment tôt avant toute publication d'un grand nombre de copies, afin de lui permettre de vous donner une version à jour du Document.

4 - MODIFICATIONS

Vous pouvez copier et distribuer une Version modifiée du Document en respectant les conditions des sections 2 et 3 précédentes, à condition de placer cette Version modifiée sous la présente Licence, dans laquelle le terme « Document » doit être remplacé par les termes « Version modifiée », donnant ainsi l'autorisation de redistribuer et de modifier cette Version modifiée à quiconque en possède une copie. De plus, vous devez effectuer les actions suivantes dans la Version modifiée :

- A Utiliser sur la Page de titre (et sur la page de couverture éventuellement présente) un titre distinct de celui du Document d'origine et de toutes ses versions antérieures (qui, si elles existent, doivent être mentionnées dans la section « Historique » du Document). Vous pouvez utiliser le même titre si l'éditeur d'origine vous en a donné expressément la permission.
- B Mentionner sur la Page de titre en tant qu'auteurs une ou plusieurs des personnes ou entités responsables des modifications de la Version modifiée, avec au moins les cinq principaux auteurs du Document (ou tous les auteurs s'il y en a moins de cinq).
- C Préciser sur la Page de titre le nom de l'éditeur de la Version modifiée, en tant qu'éditeur du Document.
- D Préserver intégralement toutes les notices de copyright du Document.
- E Ajouter une notice de copyright adjacente aux autres notices pour vos propres modifications.
- F Inclure immédiatement après les notices de copyright une notice donnant à quiconque l'autorisation d'utiliser la Version modifiée selon les termes de cette Licence, sous la forme présentée dans l'annexe indiquée ci-dessous.
- G Préserver dans cette notice la liste complète des Sections inaltérables et les Textes de couverture donnés avec la notice de la Licence du Document.
- H Inclure une copie non modifiée de cette Licence.

- I Préserver la section nommée « Historique » et son titre, et y ajouter une nouvelle entrée décrivant le titre, l'année, les nouveaux auteurs et l'éditeur de la Version modifiée, tels que décrits sur la Page de titre, ainsi qu'un descriptif des modifications apportées depuis la précédente version.
- J Conserver l'adresse réseau éventuellement indiquée dans le Document permettant à quiconque d'accéder à une Copie transparente du Document, ainsi que les adresses réseau indiquées dans le Document pour les versions précédentes sur lesquelles le Document se base. Ces liens peuvent être placés dans la section « Historique ». Vous pouvez ne pas conserver les liens pour un travail datant de plus de quatre ans avant la version courante, ou si l'éditeur d'origine vous en accorde la permission.
- K Si une section « Dédicaces » ou une section « Remerciements » sont présentes, les informations et les appréciations concernant les contributeurs et les personnes auxquelles s'adressent ces remerciements doivent être conservées, ainsi que le titre de ces sections.
- L Conserver sans modification les Sections inaltérables du Document, ni dans leurs textes ni dans leurs titres. Les numéros de sections ne sont pas considérés comme faisant partie du texte des sections.
- M Effacer toute section intitulée « Approbations ». Une telle section ne peut pas être incluse dans une Version modifiée.
- N Ne pas renommer une section existante sous le titre « Approbations » ou sous un autre titre entrant en conflit avec le titre d'une Section inaltérable.

Si la Version modifiée contient de nouvelles sections préliminaires ou de nouvelles annexes considérées comme des Sections secondaires, et que celles-ci ne contiennent aucun élément copié depuis le Document, vous pouvez à votre convenance en désigner une ou plusieurs comme étant des Sections inaltérables. Pour ce faire, ajoutez leurs titres dans la liste des Sections inaltérables au sein de la notice de Licence de la Version modifiée. Ces titres doivent être distincts des titres des autres sections.

Vous pouvez ajouter une section nommée « Approbations », à condition que ces approbations ne concernent que les modifications ayant donné naissance à la Version modifiée (par exemple, comptes-rendus de revue de document, ou acceptation du texte par une organisation le reconnaissant comme étant la définition d'un standard).

Vous pouvez ajouter un passage comprenant jusqu'à cinq mots en première page de couverture, et jusqu'à vingt-cinq mots en dernière page de couverture, à la liste des Textes de couverture de la Version modifiée. Il n'est autorisé d'ajouter qu'un seul passage en première et en dernière page de couverture par personne ou groupe de personnes ou organisation ayant contribué à la modification du Document. Si le Document comporte déjà un passage sur la même couverture, ajouté en votre nom ou au nom de l'organisation au nom de laquelle vous agissez, vous ne pouvez pas ajouter de passage supplémentaire ; mais vous pouvez remplacer un ancien passage si vous avez expressément obtenu l'autorisation de l'éditeur de celui-ci.

Cette Licence ne vous donne pas le droit d'utiliser le nom des auteurs et des éditeurs de ce Document à des fins publicitaires ou pour prétendre à l'approbation d'une Version modifiée.

5 - FUSION DE DOCUMENTS

Vous pouvez fusionner le Document avec d'autres documents soumis à cette Licence, suivant les spécifications de la section 4 pour les Versions modifiées, à condition d'inclure dans le document résultant toutes les Sections inaltérables des documents originaux sans modification, et de toutes les lister dans la liste des Sections inaltérables de la notice de Licence du document résultant de la fusion.

Le document résultant de la fusion n'a besoin que d'une seule copie de cette Licence, et les Sections inaltérables existant en multiples exemplaires peuvent être remplacées par une copie unique. S'il existe plusieurs Sections inaltérables portant le même nom, mais de contenu différent, rendez unique le titre de chaque section en ajoutant, à la fin de celui-ci, entre parenthèses, le nom de l'auteur ou de l'éditeur d'origine, ou, à défaut, un numéro unique. Les mêmes modifications doivent être réalisées dans la liste des Sections inaltérables de la notice de Licence du document final.

Dans le document résultant de la fusion, vous devez rassembler en une seule toutes les sections « Historique » des documents d'origine. De même, vous devez rassembler les sections « Remerciements » et « Dédicaces ». Vous devez supprimer toutes les sections « Approbations ».

6 - REGROUPEMENTS DE DOCUMENTS

Vous pouvez créer un regroupement de documents comprenant le Document et d'autres documents soumis à cette Licence, et remplacer les copies individuelles de cette Licence des différents documents par une unique copie incluse dans le regroupement de documents, à condition de respecter pour chacun de ces documents l'ensemble des règles de cette Licence concernant les copies conformes.

Vous pouvez extraire un document d'un tel regroupement, et le distribuer individuellement sous couvert de cette Licence, à condition d'y inclure une copie de cette Licence et de l'ensemble des règles concernant les copies conformes.

7 - AGRÉGATION AVEC DES TRAVAUX INDÉPENDANTS

La compilation du Document ou ses dérivés avec d'autres documents ou travaux séparés et indépendants sur un support de stockage ou sur un média de distribution quelconque ne représente pas une Version modifiée du Document tant qu'aucun copyright n'est déposé pour cette compilation. Une telle compilation est appelée « agrégat », et cette Licence ne s'applique pas aux autres travaux indépendants compilés avec le Document, s'ils ne sont pas eux-mêmes des travaux dérivés du Document.

Si les exigences de la section 3 concernant les Textes de couverture sont applicables à ces copies du Document, et si le Document représente un volume inférieur à un quart du volume total de l'agrégat, les Textes de couverture du Document peuvent être placés sur des pages de couverture qui n'encadrent que le Document au sein de l'agrégat. Dans le cas contraire, ils doivent apparaître sur les pages de couverture de l'agrégat complet.

8 - TRADUCTION

La traduction est considérée comme une forme de modification, vous pouvez donc distribuer les traductions du Document selon les termes de la section 4. Vous devez obtenir l'autorisation spéciale des auteurs des Sections inaltérables pour les remplacer par des traductions, mais vous pouvez inclure les traductions des Sections inaltérables en plus des textes originaux. Vous pouvez inclure une traduction de cette Licence à condition d'inclure également la version originale en anglais. En cas de contradiction entre la traduction et la version originale en anglais, c'est cette dernière qui prévaut.

9 - RÉVOCATION

Vous ne pouvez pas copier, modifier, sous-licencier ou distribuer le Document autrement que selon les termes de cette Licence. Tout autre acte de copie, modification, sous-licence ou distribution du Document est sans objet et vous prive automatiquement des droits que cette Licence vous accorde. En revanche, les personnes qui ont reçu de votre part des copies ou les droits sur le document sous couvert de cette Licence ne voient pas leurs droits caducs tant qu'elles en respectent les principes.

10 - RÉVISIONS FUTURES DE CETTE LICENCE

La Free Software Foundation peut publier de temps en temps de nouvelles versions révisées de cette Licence. Ces nouvelles versions seront semblables à la présente version dans l'esprit, mais pourront différer sur des points de détail en fonction de nouvelles questions ou problèmes. Voyez <http://www.gnu.org/copyleft/> pour plus de détails.

Chaque version de cette Licence est dotée d'un numéro de version distinct. Si un Document spécifie un numéro de version particulier de cette Licence, et porte la mention « ou toute autre version ultérieure », vous pouvez choisir de suivre les termes de la version spécifiée ou ceux de n'importe quelle version ultérieure publiée par la Free Software Foundation. Si aucun numéro de version n'est spécifié, vous pouvez choisir n'importe quelle version officielle publiée par la Free Software Foundation.

XVII - Remarques de l'auteur

XVII-A - Au sujet de ce document

Il se peut que certaines informations fournies dans ce livre soient spécifiques à ma configuration matérielle et logicielle. À titre indicatif, j'utilise une machine fonctionnant sous Linux Slackware 12.0, et j'ai contrôlé les informations sur une Debian Etch. En particulier, il est connu que la Slackware n'utilise pas, par défaut, la notion de niveaux d'exécution. En revanche, elle n'utilise que les versions officielles des logiciels, et sa configuration reste simple et compréhensible.

Je me suis toutefois efforcé de rendre ce document générique et indépendant de ma configuration. J'espère donc que la plupart des informations fournies ici s'appliqueront à la plupart des configurations matérielles et à toutes les distributions Linux, bien que je ne puisse pas le garantir. Les informations données dans ce livre permettront donc sans doute aux personnes qui n'ont jamais vu Linux de débroussailler un peu le terrain, et à celles qui utilisent déjà Linux de comprendre en profondeur comment leur système fonctionne.

Je remercie d'avance les gens qui pourront m'envoyer des remarques concernant les imprécisions, voire les horreurs et les âneries que j'aurais pu écrire. Plus je recevrai de critiques constructives et de propositions, plus ce document a de chances de s'améliorer. Cependant, si vous prenez le temps de m'envoyer les remarques et les erreurs que vous avez pu détecter, je vous saurais gré de vérifier au préalable qu'elles sont toujours d'actualité dans la dernière version de ce document, que vous pourrez trouver dans différents formats de fichiers sur [mon site Web](#).

Vous pouvez contribuer au maintien et au support de mes documents libres sur la page de support de mon site Web, à l'adresse <http://casteyde.christian.free.fr/support/support.html>. Si vous estimez que ce travail mérite salaire, n'hésitez pas à m'offrir un pot, c'est facile et sécurisé (n'envoyez pas de pizza par La Poste SVP).

XVII-B - Copyright

Copyright © 2004 Christian Casteyde

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts.

A copy of the license is included in the section entitled « GNU Free Documentation License ».

Permission vous est donnée de copier, distribuer et modifier ce document selon les termes de la licence GNU pour les documentations libres, version 1.1 ou toute autre version ultérieure publiée par la Free Software Foundation.

Une copie de cette licence est incluse dans l'annexe intitulée « Licence de Documentation Libre GNU ».

XVII-C - Commentez

N'hésitez pas à commenter cet article ! [Commentez](#)