



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

C Collection *Certifications*

Préparation à la Certification **LPIC-1**

LINUX

*Seconde
Edition*

EXAMENS LPI 101 et LPI 102

**47 travaux pratiques
536 questions-réponses**

OFFERT :
UN EXAMEN BLANC en ligne  
avec réponses commentées et détaillées



Sébastien ROHAUT

Préparation à la Certification **LPIC-1**

LINUX

*Seconde
Edition*

EXAMENS LPI 101 et LPI 102

**47 travaux pratiques
536 questions-réponses**

OFFERT :

UN EXAMEN BLANC en ligne  

avec réponses commentées et détaillées

This One



Z5S1-Y5E-ABFU

Éléments sous droits d'auteur

Toutes les marques citées ont été déposées par leur éditeur respectif.

La loi du 11 mars 1957 n'autorisant aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les "copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective", et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, "toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayant cause, est illicite" (alinéa 1er de l'article 40).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contre-façon sanctionnée par les articles 425 et suivants du code pénal.

Copyright - Editions ENI - Août 2009
ISBN : 978-2-7460-5009-9
Imprimé en France

Editions ENI

ZAC du Moulin Neuf
Rue Benjamin Franklin

44800 SAINT HERBLAIN CEDEX

Tél. : 02.51.80.15.15
Fax : 02.51.80.15.16

e-mail : editions@ediENI.com
<http://www.editions-eni.com>

Auteur : Sébastien ROHAUT
Collection dirigée par Joëlle MUSSET

Introduction

A. Pourquoi se certifier ?	26
B. Les certifications LPI	26
C. La certification LPIC-1	27
1. Les objectifs	27
2. Passer les examens	28
a. Inscription	28
b. Coût	28
c. Centre d'examen	28
d. Déroulement	29
e. Réussite	29
D. Contenu du livre	29

Chapitre 1**Présentation de Linux**

A. Bienvenue dans le monde Unix	32
1. Un nouveau monde	32
2. Histoire des ordinateurs	32
a. Complexité des ordinateurs	32
b. L'intelligence	33
3. Le système d'exploitation	33
4. Le système Unix, une brève histoire	35
a. De MULTICS à UNIX	35
b. Le langage C	37
c. Les licences et l'avènement de BSD et System V	38
d. La guerre des Unix	38
e. La standardisation	39
f. Unix est un standard	39
g. Unix sur les ordinateurs personnels	39
B. Le logiciel libre	40
1. Les origines du logiciel libre	40
2. Le projet GNU et la FSF	41
3. L'Open Source	42
4. GNU/Linux	42
a. Linus Torvalds	42
b. L'accident	43
c. La première version officielle	43
d. Le succès communautaire	43
e. Les années 1994-1997	44
f. À partir de 1998 : l'explosion	44
g. Aujourd'hui et demain	44

C. Quel matériel pour Linux ?	45
1. L'architecture	45
2. Compatibilité du matériel	46
D. Choisir une distribution	47
1. Debian	47
a. Ubuntu	48
b. Red Hat et Fedora	48
c. Mandriva (ex-Mandrake)	49
d. openSUSE	49
e. Les autres	50
2. Les LiveCD	51
E. Obtenir de l'aide	51
1. L'aide propre aux commandes	51
2. L'aide interne au shell	52
3. Le manuel en ligne	52
a. Accès	52
b. Structure d'une page	52
c. Navigation	53
d. Les sections	53
e. Rechercher par correspondance	54
4. Rechercher de l'aide sur Internet	55
F. Validation des acquis : questions/réponses.	55
G. Travaux pratiques	61
1. Histoire d'Unix et de Linux	61
2. Distributions	62
3. Aide et documentation	62

Chapitre 2

Installation de Linux et des logiciels

A. Installer une Debian	66
1. Support d'installation	66
2. Boot sur le support	66
3. Choix des langues et pays	67
4. Paramètres du réseau	68
5. Partitionner les disques	69
6. Comptes root et utilisateurs	71
7. Installation	71
8. Configuration des packages	73
9. Fin d'installation et redémarrage	73
B. Installation de openSUSE	74
1. Support d'installation	74

2. Boot sur le support	74
3. Choix de base	76
4. Type de bureau	77
5. Paramètres d'installation	78
6. Partitionnement	79
7. Installation des logiciels.	80
8. Installation en cours	81
9. Configuration du réseau	82
10. Mise à jour du système	84
11. Gestion des utilisateurs	85
12. Configuration du matériel	85
C. Red Hat Package Manager	87
1. Notion de package	87
2. Le gestionnaire RPM	87
3. Installation, mise à jour et suppression	88
4. Cas du noyau	88
5. Requêtes RPM	89
6. Vérification des packages.	90
7. Les dépendances	91
8. Mises à jour automatisées.	91
D. YUM.	91
1. Configuration des dépôts	91
2. Utilisation des dépôts	92
a. Rafraîchir le cache	92
b. Lister les packages.	92
c. Installer des packages	93
d. Mises à jour	94
e. Rechercher un package	94
f. Supprimer un package	95
E. Debian Package	95
1. dpkg : le gestionnaire de paquets Debian.	95
2. Installation, mise à jour et suppression	96
3. Requêtes dpkg.	97
a. Lister les paquets	97
b. Trouver un paquet contenant un fichier.	98
c. Lister le contenu d'un paquet	98
4. Convertir des packages	99
5. L'outil dselect	100
F. Gestionnaire APT.	100
1. Principe	100
2. Les dépôts	101
a. Configuration.	101

b. Mise à jour de la base	102
3. Mise à jour de la distribution	103
4. Rechercher et installer un package individuel.	105
5. Client graphique	106
G. Installer depuis les sources	106
1. Obtenir les sources	106
2. Pré-requis et dépendances	107
3. Exemple d'installation	107
4. Désinstallation	112
5. Les bases du Makefile.	113
a. Bases	113
b. Makefile intermédiaire	114
c. Un peu plus complexe	115
H. Gérer les bibliothèques partagées	117
1. Principe	117
2. Lieu de stockage	117
3. Quelles bibliothèques liées ?	118
4. Configurer le cache de l'éditeur de liens	119
I. Validation des acquis : questions/réponses	121
J. Travaux pratiques	129
1. Schéma de partitionnement	129
2. Gestion des RPM	129
3. Gestion de DPKG et APT	131
4. Les sources	132
5. Bibliothèques partagées	133

Chapitre 3**Le shell et les commandes GNU**

A. Le shell bash	136
1. Rôle	136
2. Bash : le shell par défaut.	137
a. Un shell puissant et libre	137
b. L'invite de commande	137
3. Utiliser le shell	138
a. La saisie	138
b. Syntaxe générale des commandes.	138
c. Premier exemple concret avec cal	138
d. Chaîner les commandes	139
e. Afficher du texte avec echo	140
f. Commandes internes et externes	140
g. Quelques raccourcis utiles	141

4. Rappel de l'historique	141
B. La gestion des fichiers	142
1. Le système de fichiers	142
2. Les divers types de fichiers	143
a. Les fichiers ordinaires ou réguliers	143
b. Les catalogues	143
c. Les fichiers spéciaux	143
3. Nomenclature des fichiers	143
4. Les chemins	144
a. Structure et nom de chemin	144
b. Répertoire personnel	144
c. Chemin relatif	145
d. Le tilde	145
e. cd	145
5. Les commandes de base	146
a. Lister les fichiers et les répertoires	146
b. Gérer les fichiers et les répertoires	147
c. Wildcards : caractères de substitution	150
d. Verrouillage de caractères	151
C. Rechercher des fichiers	152
1. Considérations générales	152
2. Critères de recherche	152
a. -name	152
b. -type	153
c. -user et -group	153
d. -size	153
e. -atime, -mtime et -ctime	154
f. -perm	154
g. -links et -inum	155
3. Commandes	155
a. -ls	155
b. -exec	155
c. -ok	156
4. Critères AND / OR / NOT	156
5. Retrouver des exécutables	156
a. whereis	156
b. which	157
6. locate	157
D. L'éditeur vi	158
1. Présentation	158
2. Fonctionnement	158

3. Les commandes	159
a. La saisie	159
b. Quitter et sauver	159
c. Déplacement	159
d. La correction	160
e. Recherche dans le texte	160
f. Commandes de remplacement	161
g. Copier-Coller	161
h. Substitution	161
i. Autres	161
E. Redirections	162
1. Principe	162
2. En sortie	162
3. En entrée	163
4. Document en ligne	163
5. Les canaux standards	163
6. Ouverture de canaux	164
7. Filtre : définition	164
8. Pipelines / tubes	164
F. Les filtres et utilitaires	165
1. Extraction des noms et chemins	165
2. Recherche de lignes	165
a. grep	165
b. egrep	166
c. fgrep	166
3. sed	167
4. Colonnes et champs	167
a. Colonnes	167
b. Champs	168
5. Décompte de lignes	169
6. Tri de lignes	169
7. Suppression des doublons	171
8. Jointure de deux fichiers	171
a. Sur des champs communs	171
b. Ligne à ligne	171
9. Découpage d'un fichier en morceaux	172
a. Découper	172
b. Reconstruire	172
10. Remplacement de caractères	173
a. Liste de caractères	173
b. Tabulations et espaces	174

11. Visualisation de texte	174
a. En pleine page	174
b. Début d'un fichier.	175
c. Fin et attente de fichier	175
12. Duplication du canal de sortie standard	175
13. Comparaison de fichiers	176
a. diff	176
b. cmp	177
14. Délai d'attente	178
G. Les processus	178
1. Définition et environnement.	178
2. États d'un processus	178
3. Lancement en tâche de fond.	179
4. Backgroung, foreground, jobs	179
5. Liste des processus	180
6. Arrêt d'un processus / signaux	181
7. nohup	182
8. nice et renice	182
9. time	183
H. Plus loin avec le bash	183
1. Alias	183
2. Groupement de commandes	184
3. Liaison et exécution conditionnelle.	184
I. Les variables	185
1. Nomenclature	185
2. Déclaration et affectation	185
3. Accès et affichage.	185
4. Suppression et protection	186
5. Export	187
6. Accolades	187
7. Accolades et remplacement conditionnel	187
8. Variables système.	188
9. Variables spéciales.	189
10. Longueur d'une chaîne	189
11. Tableaux et champs.	189
12. Variables typées	190
J. Configuration de bash	190
1. Fichiers de configuration	190
a. Shell de connexion.	191
b. Shell simple	191
c. Mode Bourne shell	191
d. Mode non interactif	191

2. Commandes set	191
K. Programmation shell	192
1. Structure et exécution d'un script	192
2. Arguments d'un script	192
a. Paramètres de position	192
b. Redéfinition des paramètres	193
c. Réorganisation des paramètres	194
d. Sortie de script	194
3. Environnement du processus	194
4. Substitution de commande	196
5. Tests de conditions	196
a. Tests sur une chaîne	196
b. Tests sur les valeurs numériques	197
c. Tests sur les fichiers	197
d. Tests combinés par des critères ET, OU, NON	198
e. Syntaxe allégée	198
6. if ... then ... else	198
7. Choix multiples case	199
8. Saisie de l'utilisateur	200
9. Les boucles	201
a. Boucle for	201
b. Boucle while	203
c. Boucle until	204
d. true et false	204
e. break et continue	204
f. Boucle select	205
10. Les fonctions	206
11. Calculs et expressions	206
a. expr	206
b. Calculs avec bash	207
12. Une variable dans une autre variable	208
13. Traitement des signaux	208
14. Commande « : »	209
L. SQL	209
1. Présentation	209
2. Requêtes de sélection	210
a. Select	210
b. Distinct	210
c. Where	210
3. Les expressions et les fonctions	211
4. La clause ORDER BY	211

5. La clause GROUP BY	211
6. Les jointures	212
7. Un Select dans un Select	212
8. Les insertions	213
9. Modifications	213
10. Suppression	213
M. Validation des acquis : questions-réponses.	213
N. Travaux pratiques	225
1. Gestion des fichiers	225
2. Rechercher des fichiers	226
3. Les redirections	227
4. Les filtres et utilitaires	228
5. Les processus	228
6. Programmation Shell Niveau 1	229
7. Fonction Shell	230

Chapitre 4

Les disques et le système de fichiers

A. Représentation des disques	232
1. Nomenclature	232
a. IDE	232
b. SCSI, SATA, USB, FIREWIRE, etc..	233
2. Cas spéciaux	233
B. Manipulations de bas niveau	233
1. Informations	233
2. Modification des valeurs	235
C. Choisir un système de fichiers	236
1. Principe	236
a. Définition	236
b. Représentation	236
c. Les métadonnées	236
d. Les noms des fichiers	237
e. Le journal	237
2. Les filesystems sous Linux	237
a. ext2	237
b. ext3	237
c. reiserfs	237
d. xfs	238
e. vfat	238

D. Partitionnement	239
1. Découpage logique	239
2. Organisation d'un disque.	239
a. MBR	239
b. Les partitions	239
c. EBR	240
d. PBR	240
e. Types de partitions	240
3. Manipuler les partitions	242
a. Outils disponibles	242
b. Manipuler les partitions	242
E. Manipuler les systèmes de fichiers	246
1. Définitions de base	246
a. Bloc	246
b. Superbloc	246
c. Table d'inodes	246
d. Tables catalogues	248
e. Hard link	248
2. Créer un système de fichiers	248
a. mkfs, syntaxe générale	248
b. Un premier exemple en ext2	249
c. ext2 et ext3	250
d. xfs	252
e. vfat	252
F. Accéder aux systèmes de fichiers	253
1. mount	253
a. Montage par périphérique	253
b. Options de montage	255
c. umount	256
d. /etc/fstab	256
e. Cas des CD et images ISO	258
G. Contrôler le système de fichiers	259
1. Statistiques d'occupation	259
a. Par système de fichiers	259
b. Par arborescence	260
2. Vérifier, régler et réparer	260
a. fsck	260
b. badblocks	261
c. dumpe2fs	261
d. tune2fs	263

H. Le swap	264
1. Pourquoi créer un swap ?	264
2. Taille optimale	264
3. Créer une partition de swap	264
4. Activer et désactiver le swap	264
a. Activation dynamique	264
b. Dans /etc/fstab	265
5. En cas d'urgence : fichier de swap	265
6. État de la mémoire	266
a. free	266
b. Mémoire réservée	266
c. meminfo	266
I. Les quotas disques	267
1. Définitions	267
2. Mise en place	268
J. Les droits d'accès	269
1. Les droits de base	269
a. Droits et utilisateurs	269
b. Signification	270
2. Modification des droits	270
a. Par symboles	270
b. Par base 8	271
3. Masque des droits	271
a. Restreindre des droits automatiquement	271
b. Calcul de masque	272
4. Changer de propriétaire et de groupe	272
5. Droits d'accès étendus	273
a. SUID et SGID	273
b. Real / effectif	273
c. Sticky bit	274
d. Droit s et répertoire	274
K. Validation des acquis : questions/réponses	275
L. Travaux pratiques	285
1. Les disques et partitions	285
2. Cr éation d'un syst ème de fichiers	285
3. Acc ès et montage du syst ème de fichiers	286
4. Statistiques et entretien du syst ème de fichiers	286
5. Swap et m émoire	287
6. Quotas	287
7. Les droits	288

Chapitre 5

Démarrage de Linux, services, noyau et périphériques

A. Processus de démarrage	290
1. Le BIOS	290
a. Rôle	290
b. Réglages basiques.	291
2. Le chargeur de démarrage	292
3. GRUB	292
a. Configuration.	292
b. Installation	293
c. Démarrage et édition.	293
4. Initialisation du noyau	293
B. init	294
1. Rôle.	294
2. Niveaux d'exécution	295
3. /etc/inittab	295
4. Changement de niveau	297
5. Paramétrage système de base	297
6. Niveaux d'exécution System V	298
a. rc	298
7. Gestion des niveaux et des services.	298
a. Services dans init.d.	298
b. Contrôle manuel des services.	299
c. Modification des niveaux d'exécution	300
8. Consoles virtuelles.	302
9. Les logins.	302
10. Arrêt	303
C. Consulter les traces du système.	304
1. dmesg.	304
2. /var/log/messages	306
D. Services et modules noyau	307
1. Présentation	307
2. uname	307
3. Gestion des modules	308
a. lsmod	309
b. depmod	309
c. modinfo	310
d. insmod	311
e. rmmod	312
f. modprobe	312
g. modprobe.conf.	313

4. Chargement des modules au boot	314
a. initrd	314
b. Red Hat /etc/rc.modules	315
c. openSUSE : /etc/sysconfig/kernel	315
d. Debian : /etc/modules	316
5. Paramètres dynamiques	316
a. /proc et /sys	316
b. sysctl	317
E. Compiler un noyau	318
1. Obtenir les sources	318
a. Sources officielles	318
b. Sources de la distribution	319
2. Les outils nécessaires	319
3. Configuration	320
a. Le .config	320
b. Récupérer la configuration du noyau	320
c. make oldconfig	321
d. make menuconfig	323
e. make xconfig	323
f. Quelques choix d'optimisation	324
4. Compilation	326
5. Installation	327
6. Test	328
7. Autres options	328
F. Les fichiers périphériques	328
1. Introduction	328
2. Fichiers spéciaux	329
3. Créer un fichier spécial	330
4. Connaître son matériel	331
a. Bus PCI	331
b. Bus USB	332
c. Ressources matérielles	333
d. Autres outils	335
5. Le support de l'USB et du hotplug	338
a. Les modules	338
b. Chargement	339
c. hotplug, usbmgr	339
d. udev	339
G. Validations des acquis : questions/réponses	341
H. Travaux pratiques	352
1. GRUB et le processus de boot	352
2. Init et runlevel	352

3. Noyau et modules	353
4. Recompilation du noyau	354
5. Les périphériques et le matériel	354

Chapitre 6**Les tâches administratives**

A. Administration des utilisateurs	358
1. Principe	358
a. Identification et authentification	358
b. Les utilisateurs	358
c. Les groupes	359
d. Les mots de passe	359
2. Les fichiers	360
a. /etc/passwd	360
b. /etc/group	360
c. /etc/shadow	360
d. /etc/gshadow	361
3. Gestion des utilisateurs	361
a. Ajout	361
b. Sécurité des mots de passe	362
c. Modification	365
d. Suppression	365
4. Gestion des groupes	365
a. Ajout	365
b. Modification	365
c. Suppression	366
5. Commandes additionnelles	366
a. Conversion des fichiers	366
b. Vérifier la cohérence	366
c. Vérifier les connexions	367
d. Actions de l'utilisateur	367
6. Configuration avancée	369
a. /etc/default/useradd	369
b. /etc/default/passwd	370
c. /etc/default/su	370
d. /etc/login.defs	370
7. Notifications à l'utilisateur	373
a. /etc/issue	373
b. /etc/issue.net	373
c. /etc/motd	373
8. L'environnement utilisateur	373
a. /etc/skel	373

b. Scripts de configuration	373
c. Groupes privés et setgid.	374
9. Aperçu de PAM	374
B. L'impression	376
1. Principe	376
2. System V	377
3. BSD	377
4. CUPS	378
a. Présentation.	378
b. Ajout d'une imprimante	378
C. Automatisation	385
1. Avec cron	385
a. Présentation.	385
b. Formalisme	385
c. Exemples	385
d. crontab système	385
e. Contrôle d'accès	386
f. Crontab en mode graphique	386
2. Avec at	387
a. Présentation.	387
b. Formalisme	387
c. Contrôle des tâches.	388
d. Contrôle d'accès	389
D. Les traces (logs) du système	389
1. Principe	389
2. Les messages	390
3. Configuration de syslog.	390
4. Les fichiers de traces	392
E. Archivage et backup	392
1. Les outils de sauvegarde.	392
a. Commandes, plans, scripts	392
b. Voici quelques autres commandes.	393
2. Tar.	393
a. Archiver.	393
b. Lister	394
c. Restauration	394
d. Autres paramètres	394
3. cpio	395
a. Archiver.	395
b. Lister	396
c. Restaurer.	396
4. dd.	397

F. L'horloge	398
1. Connaitre l'heure	398
a. date	398
b. hwclock	399
2. Modifier l'horloge matérielle	399
a. Via date	399
b. Via hwclock	399
3. NTP	400
a. Principe	400
b. Client ntp	400
G. Les paramètres régionaux	401
1. i18n et l10n	401
2. Réglages locaux	401
a. Outils de la distribution	401
b. Variables d'environnement	402
3. Codage des caractères	403
4. Fuseaux horaires	404
H. Validation des acquis : questions/réponses	404
I. Travaux pratiques	412
1. Gestion des utilisateurs	412
2. L'impression	413
3. Automatisation des tâches	413
4. Les traces du système	414
5. Archivage	414

Chapitre 7

Le réseau

A. TCP/IP	416
1. Bases	416
2. Adressage	417
a. Classes	417
b. Sous-réseaux	418
c. Routage	418
3. Configuration	419
a. Cas des distributions de type Red Hat/Fedora	419
b. Machines de type Debian	420
c. Routage	421
4. Outils réseaux	422
a. FTP	422
b. Telnet	423
c. Ping	424
d. Traceroute	425

e. Whois	425
f. Netstat	426
g. IPTraf	428
5. Fichiers généraux	429
a. /etc/resolv.conf	429
b. /etc/hosts et /etc/networks	430
c. /etc/nsswitch.conf	430
d. /etc/services	430
e. /etc/protocols	431
B. Services réseaux xinetd	432
1. Présentation	432
2. Configuration	432
3. Démarrage et arrêt des services	433
C. Connexion PPP	434
1. Choix et réglage du modem	434
a. Le cas des Winmodems	434
b. Les fichiers périphériques	434
c. Régler le port série	434
d. Les commandes AT	435
2. PPP	435
3. Connexion via la console	436
a. À la main	436
b. Par les fichiers	436
c. Connexion	437
d. Connexion par front-end	438
D. OpenSSH	439
1. Présentation	439
2. Configuration	439
3. Utilisation	439
4. Clés et connexion automatique	440
a. Côté client	440
b. Côté serveur	440
E. Monter un serveur DHCP	441
1. Présentation	441
2. Serveur dhcpd	441
a. Démarrage	441
3. Informations de base	441
4. Côté client	442
F. Serveur DNS	443
1. Présentation	443
2. Lancement	443

3. Configuration de Bind	443
a. Configuration générale	444
b. Section globale	444
c. Section de zones	445
d. Zone de résolution	445
e. Zone de résolution inverse	445
f. Exemple	446
g. Zones spéciales	446
4. Fichiers de zones	446
a. Définitions	446
b. Zone	447
c. Zone de résolution inverse	449
5. Diagnostic des problèmes de configuration	449
a. Interrogation dig et host	450
G. Courrier électronique	452
1. Principe	452
2. postfix	453
a. Configuration simple	453
b. Alias d'utilisateurs	453
c. Test	453
3. POP et IMAP	454
H. Service HTTP Apache	454
1. Présentation	454
2. Arrêt/Relance	455
3. Configuration	455
4. Directives générales	455
5. Gestion des performances	456
6. Les répertoires, alias et emplacements	456
a. Directory	456
b. Alias	457
7. Hôtes virtuels	457
I. Partage de fichiers	458
1. NFS	458
a. Lancement	458
b. Partage côté serveur	458
c. Montage côté client	459
J. FTP	460
K. Partages Windows avec Samba	460
1. Présentation	460
2. Configuration	461
3. Partage de fichiers	461
4. Partage des imprimantes	462

5. Méthodes d'authentification	462
6. Correspondance des noms et des mots de passe	463
7. Clients SAMBA	463
a. En ligne	463
b. Montage	463
L. Validation des acquis : questions/réponses	463
M. Travaux pratiques	476
1. Configuration TCP/IP de Linux	476
2. Quelques commandes réseau	477
3. Le resolver	478
4. Services réseau	479
5. Partages de fichiers	480

Chapitre 8

La sécurité

A. Bases de sécurité	482
1. Sécurité informatique	482
2. Contrôler les droits d'endossement	483
3. Vérifier les packages	484
4. Politique de mot de passe	484
5. Interdire les connexions	486
a. /bin/false	486
b. /etc/nologin	486
c. /etc/securetty	487
6. Tester les mots de passe	487
7. Rechercher des rootkits	488
a. Principe du rootkit	488
b. chkrootkit	489
8. Les virus	490
9. Les limites de l'utilisateur	492
10. Les droits SUDO	493
a. Donner des privilèges étendus	493
b. Syntaxe de /etc/sudoers	494
11. Audit plus complet	495
12. Les bulletins de sécurité	496
a. CERT : Computer Emergency Response Team	496
b. Bugtraq	497
c. Les bulletins des distributions	498
d. Les correctifs	499
B. Sécurité des services et du réseau	499
1. Vérifier les ports ouverts	499
a. Les sockets	499

b. Informations depuis netstat	500
c. L'outil nmap	501
2. Supprimer les services inutiles	503
a. Généralités	503
b. Services standalone	503
c. Services xinetd	503
3. Les tcp_wrappers	504
4. Netfilter	505
a. Présentation	505
b. Vie d'un paquet	505
c. Principe des règles	506
d. Cibles de règles	507
e. Premier exemple	507
f. Opérations de base	507
g. Critères de correspondance	508
h. Sauver ses réglages	509
5. GPG	509
a. Un clone de PGP	509
b. Générer les clés	509
c. Générer une clé de révocation	511
d. Gérer le trousseau	513
e. Exporter la clé publique	513
f. Importer une clé	514
g. Signer une clé	515
h. Signer et chiffrer	517
C. Validation des acquis : questions/réponses	519
D. Travaux pratiques	523
1. Contrôle des fichiers	523
2. Sécurité des utilisateurs	524
3. Sécurité générale du système	525
4. Sécurité réseau	525

Chapitre 9 X Window

A. Comment fonctionne un environnement graphique ?	528
1. X Window System	528
a. Un modèle client/serveur	528
b. Le gestionnaire de fenêtres	529
c. Les widgets et les toolkits	531
d. Les bureaux virtuels	533
2. Les environnements de bureau	533

B. Xorg	535
1. Présentation	535
2. Installation	536
3. Configuration	537
a. Via la distribution	537
b. Xorgcfg	538
c. Xorgconfig	539
4. Structure de xorg.conf.	540
a. Découpage	540
b. Valeurs booléennes	540
c. Section InputDevice	540
d. Section Monitor	541
e. Section Modes.	542
f. Section Device	543
g. Section Screen.	544
h. Section ServerLayout	545
i. Section Files.	546
j. Section Modules	546
k. Section ServerFlags	547
l. Section Extensions	548
5. xorg.conf complet	548
6. Tester et lancer X	550
a. Vérifier la configuration	550
b. Les traces	551
c. Tester le serveur	553
C. Le Display Manager	553
1. Principe	553
2. XDM	554
a. Configuration générale.	554
b. Setup : Xsetup	555
c. Chooser : RunChooser	557
d. Startup : Xstartup	557
e. Session : Xsession	557
f. Reset : Xreset.	558
g. Resources : Xresources	558
h. Servers : Xservers	558
i. AccessFile : Xaccess et XDMCP	558
3. Gdm et kdm	559
4. xdm, gdm ou kdm au boot	562
a. inittab	562
b. Service	563
c. /etc/sysconfig	563

D. Window Manager et environnement personnel	564
1. Via le display manager	564
2. Startx	564
3. Les terminaux	565
4. Les gestionnaires de fenêtres	566
a. twm	566
b. IceWM	566
c. Fvwm	567
d. CDE	567
e. WindowMaker	567
f. Enlightenment	568
g. Xfce	568
h. KDE et GNOME	568
i. Les autres	569
5. Exporter ses fenêtres	569
6. Les ressources d'une application X	570
a. Modifier l'apparence d'un programme	570
b. Éditer les ressources	571
c. xrdb	575
E. Accessibilité	576
1. Assistance au clavier et à la souris	576
2. Assistance visuelle et auditive	579
F. Validation des acquis : questions/réponses	579
G. Travaux pratiques	587
1. Comprendre X Window	587
2. Configuration simple de Xorg	587
3. Configuration simple du Display Manager	588
4.Modifier les ressources d'une application X	589

Chapitre 10

Partitionnement avancé : RAID et LVM

A. Partitionnement avancé RAID	592
1. Définitions	592
2. Précautions et considérations d'usage	593
a. Disque de secours	593
b. Disque défectueux	593
c. Boot	593
d. Swap	593
e. Périphériques	593
f. IDE	593
g. Hot Swap	594

3. RAID avec mdadm	594
a. Préparation	594
b. Création	594
c. Sauver la configuration.	596
4. État du RAID.	596
5. Simuler une panne	597
6. Remplacer un disque	598
7. Arrêt et relance manuels.	599
B. Initiation au LVM.	599
1. Principe	599
2. Les volumes physiques	600
a. Créer un volume physique	600
b. Voir les volumes physiques	601
3. Les groupes de volumes	601
a. Créer un groupe de volumes	601
b. Propriétés d'un VG	602
4. Les volumes logiques	603
a. Créer un volume logique	603
b. Propriétés d'un volume logique	604
c. Accès au volume logique	605
5. Agrandissements et réductions	605
a. Les groupes de volumes.	605
b. Agrandir un volume logique	607
c. Réduire un volume logique.	609
d. Déplacer le contenu d'un volume physique.	611
e. Réduire un groupe de volumes.	613
6. Supprimer un groupe de volumes	613
a. Étapes	613
b. Supprimer un volume logique.	613
c. Retirer tous les volumes physiques.	613
d. Détruire un groupe de volumes.	614
e. Supprimer un volume physique.	614
7. Commandes supplémentaires	614
C. Validation des acquis : questions/réponses.	614
D. Travaux pratiques	618
10.1 Gérer un RAID1	618
10.2 Manipuler un LVM	620

Annexes

Annexe	623
Tableau des objectifs	625
Index.....	627

Introduction

A. Pourquoi se certifier ?	26
B. Les certifications LPI	26
C. La certification LPIC-1	27
D. Contenu du livre	29

A. Pourquoi se certifier ?

Si vous recherchez un emploi dans l'informatique, et notamment dans le développement système, l'exploitation, la production, l'administration et l'ingénierie système, Linux est quasiment incontournable. Vous devez connaître Linux, une ligne correspondante doit apparaître dans votre CV.

Vous êtes nombreux à avoir appris Linux soit chez vous, soit à l'école. Pour certains vous êtes des passionnés, et pour la plupart vous vous êtes formés en autodidacte, soit à la maison, soit par votre expérience en entreprise. Vous formez un socle important et la plupart des entreprises se basent sur cette expérience pour vous embaucher.

Mais cela ne suffit parfois plus. Il est courant de rencontrer, de plus en plus, des annonces indiquant qu'une certification de tel ou tel niveau serait un plus, voire un pré-requis indispensable.

Linux s'est professionnalisé, et comme dans beaucoup de domaines une formation devient indispensable. Certains organismes proposent donc de certifier que vous avez certaines compétences, par niveau. Le cas classique est un éditeur qui va certifier que vous avez tel ou tel niveau sur son logiciel ou système. Il vous délivre un titre, un certificat, qui atteste de la validité de vos compétences.

Les certifications ne sont pas diplômantes, elles n'ont donc pas de valeur en tant que diplômes, mais elles sont reconnues et acceptées comme gages de compétences professionnelles par les entreprises.

La nature même de Linux, la multiplicité des distributions, font qu'il n'est pas simple de fournir un programme de certification commun à l'ensemble. Aussi des éditeurs proposent des cycles de formations certifiantes, comme Redhat ou Novell.

B. Les certifications LPI

Le **Linux Professional Institute, LPI**, est une organisation à but non lucratif (une NPO : *Non Profit Organization*) créée en décembre 1999 au Canada. Son but est de promouvoir les technologies Linux, du Libre et de l'Open Source au travers de programmes et d'exams de certification de grande qualité. Les programmes et exams des certifications sont **indépendants** : ils ne couvrent pas une distribution particulière (sauf s'il s'agit d'une certification spécialisée comme pour Ubuntu par exemple) mais l'ensemble des techniques communes à toutes les distributions Linux, et les points essentiels selon les types de distributions (par exemple, l'utilisation aussi bien de dkpg que de rpm).

Le LPI a été rejoint dans son initiative par de grands groupes dont IBM, Novell, SGI, NEC, et appuyé par plusieurs éditeurs de distributions et magazines spécialisés.

Dans ce but, le LPI a développé une série de trois certifications :

- **LPI-1** : niveau **Junior**, composée de deux exams (LPI 101 et LPI 102), qui certifie que la personne peut :
 - Travailler en ligne avec les commandes Linux ; accomplir les tâches faciles de maintenance : aide aux utilisateurs, ajout d'utilisateurs à un système étendu, sauvegarde et restauration, arrêt et réinitialisation (reboot).
 - Installer et configurer un poste de travail (incluant X) et le connecter à un LAN (réseau local), ou à un PC autonome, par un modem relié à Internet.
- **LPI-2** : niveau **Intermédiaire**, composée de deux exams (LPI 201 et LPI 202), qui certifie que la personne peut :
 - Administrer une implantation de petite et moyenne envergure.
 - Concevoir, installer, maintenir, sécuriser, et dépanner un petit réseau : connexion Internet (pare-feu, proxy, mail), gestion de serveurs Internet (serveur Web, serveur FTP), travail en équipe.

- **LPIC-3** : niveau **Senior**, composée de trois examens (LPI 301, LPI 302 et LPI 303), qui certifie que la personne peut :
 - Définir et administrer l'accès individualisé à plusieurs services (LDAP, PAM, et technologies associées).
 - Concevoir et administrer l'intégration multiservices d'un réseau mixte, voire multisites (Samba, NFS, et technologies associées).
 - Gérer la complexité (automatisation, sécurité, équipes, etc.).

Selon le LPI, une certification LPIC est :

- **Professionnelle** : établie par une communauté de professionnels de Linux, des volontaires, des professeurs et fournisseurs de services informatique.
- **Un challenge** : si vous ne connaissez pas tous les sujets, vous ne l'obtiendrez pas.
- **Accessible** : disponible dans des milliers de centres d'examen partout dans le monde et lors de grands événements.
- **De grande qualité** : elle suit les conseils et les recommandations d'experts Linux renommés et emploie une méthodologie d'examen de type psychométrique reconnue dans les milieux scientifiques et industriels.
- **Neutre** : elle ne s'appuie sur aucune distribution spécifique et demande des connaissances qui s'appliquent sur n'importe quel système Linux standard.
- **Pertinente** : les questions sélectionnées sont testées et choisies de manière précise pour déterminer lesquelles sont pertinentes en environnement réel.
- **Indépendante** vis-à-vis des organismes de formation : il n'y a pas un guide unique ou un cours unique de formation, tout type de méthode de formation est encouragé.
- **Supportée et sponsorisée** par de très grandes entreprises spécialisées ou non dans Linux.
- **Une fierté** : une certification respectée de la communauté Linux et dont elle est fière.

C. La certification LPIC-1

1. Les objectifs

Ce livre a pour objectif de vous fournir tous les éléments nécessaires au passage de la certification LPIC-1. Cette certification nécessite le passage de deux examens : le LPI 101 et le LPI 102. Vous devez obtenir ces deux examens pour être certifié LPIC-1.



La grille des objectifs en fin d'ouvrage reprend tous les objectifs de chacun des examens et vous indique où trouver les points correspondants dans le livre, ainsi que les exercices associés.

Voici les objectifs principaux de l'examen LPI 101 :

- 101 : Architecture et matériel ;
- 102 : Installation de Linux et gestion des packages ;
- 103 : Les commandes GNU et Unix ;
- 104 : Périphériques, système de fichiers, arborescence.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Chapitre 1

A. Bienvenue dans le monde Unix	32
B. Le logiciel libre.	40
C. Quel matériel pour Linux ?	45
D. Choisir une distribution	47
E. Obtenir de l'aide	51
F. Validation des acquis : questions/réponses.	55
G. Travaux pratiques	61

Pré-requis

- ☒ Accès à une console et à un navigateur pour obtenir de l'aide.

Objectifs

À la fin de ce chapitre, vous serez en mesure :

- ☒ De connaître l'histoire d'Unix.
- ☒ De connaître l'histoire de Linux.
- ☒ De connaître les grands principes du logiciel libre.
- ☒ De vérifier la compatibilité matérielle.
- ☒ De choisir une distribution.
- ☒ D'obtenir de l'aide sur Internet.
- ☒ D'obtenir de l'aide depuis Linux.

A. Bienvenue dans le monde Unix

1. Un nouveau monde

Linux n'est plus un simple effet de mode et d'annonce. Depuis ses tout premiers développements en 1991 et jusqu'à aujourd'hui Linux ne cesse d'évoluer, de changer. Le monde de l'informatique est vivant. S'il n'évolue pas, il végète. Avec Linux, des millions de personnes ont trouvé enfin ce qu'elles cherchaient.

Linux n'est pas plus compliqué à utiliser que n'importe quel autre système. Le frein au développement de Linux auprès du plus grand nombre n'est pas lié à un quelconque niveau de difficulté. L'expérience acquise auprès de nombreux utilisateurs débutants ou confirmés, des groupes d'utilisateurs Linux et des acteurs professionnels montre qu'il s'agit surtout d'un problème lié aux habitudes des gens, accoutumés des années durant à un système d'exploitation unique. En effet, ces habitudes doivent parfois être quelque peu modifiées pour s'adapter à un environnement Linux, tout comme conduire une voiture familiale ne fait pas de vous un as de la conduite sportive en Ferrari.

2. Histoire des ordinateurs

a. Complexité des ordinateurs

Un ordinateur est une machine électronique extrêmement complexe. Si le principe même de l'ordinateur tel que nous le connaissons n'a pas changé depuis l'époque de Alan Turing ou de Conrad Suze et date du début des années 1940 et même d'avant (machine de Charles Babbage), les évolutions technologiques et la miniaturisation ont permis de créer des machines de plus en plus puissantes tout en étant de plus en plus petites. Des premiers ordinateurs électromécaniques composés de milliers de lampes à vide et dont la programmation se faisait en branchant des câbles, à l'ordinateur moderne d'aujourd'hui, la complexité du matériel a été croissante. Entre l'époque où un ordinateur occupait tout un étage et où le circuit d'eau permettant de le refroidir (certains étaient même refroidis avec des pains de glace) chauffait tout un immeuble, et aujourd'hui, où il suffit d'aller faire les courses au supermarché du quartier pour acheter un ordinateur, on pourrait logiquement croire que la simplicité d'utilisation a suivi la même courbe de croissance, l'ordinateur devenant aussi simple à utiliser que votre lecteur DVD de salon.

Ce n'est malheureusement pas le cas. Alors qu'un ordinateur est tellement polyvalent doit-on lui demander d'être aussi simple à utiliser qu'une machine basique ? Doit-on forcément connaître les

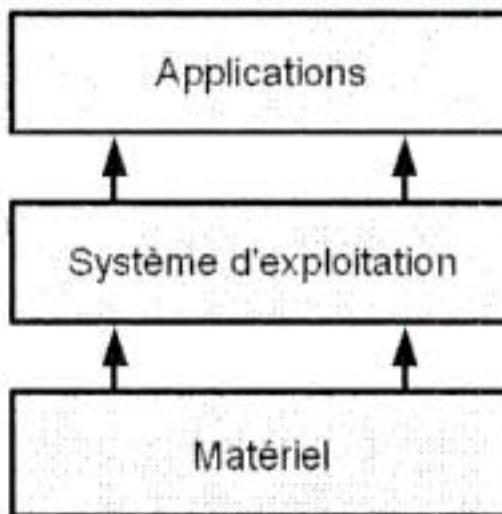
méandres techniques de son ordinateur pour pouvoir l'utiliser ? La réponse est généralement non. Cependant il y a quelques notions et bases élémentaires à retenir et à respecter.

b. L'intelligence

Un ordinateur ne sait rien faire tout seul. Il n'y a rien de plus stupide qu'un ordinateur, il faut toujours lui dire ce qu'il doit faire. L'ordinateur est moins intelligent que le moindre insecte. Ce qui le rend « intelligent » au sens puissance de calcul c'est vous et les programmes que vous lui faites exécuter. Bien qu'inventé par des humains et pour des humains, l'ordinateur ne comprend pas votre langage. Il parle le binaire, assemblage de zéro (0) et de un (1). Ces valeurs assemblées les unes aux autres forment des mots et des données pour l'ordinateur. Le binaire devient un langage appelé le langage machine. Les microprocesseurs utilisent un langage appelé l'assembleur où chaque instruction dispose d'un équivalent en binaire. À l'aide de ce langage assembleur, les informaticiens vont créer divers produits dont un appelé le compilateur, un traducteur de langage dit de haut niveau et compréhensible cette fois par un plus grand nombre d'informaticiens.

3. Le système d'exploitation

Entre le moment où vous appuyez sur le bouton d'allumage de votre ordinateur et celui où vous pouvez enfin travailler et utiliser vos logiciels il se passe un certain temps durant lequel des programmes sont chargés dans la mémoire de votre ordinateur. Le but de ces programmes est de vous simplifier la vie en rendant les choses plus simples et pas seulement pour l'utilisateur mais aussi pour l'informaticien. Ces programmes forment un ensemble appelé le système d'exploitation. Comme son nom l'indique, le rôle du système d'exploitation est d'exploiter l'ordinateur le plus souvent à votre place, ou plutôt le système d'exploitation vous fournit toute la base nécessaire pour exploiter du mieux possible les ressources de votre ordinateur.



Principe du système d'exploitation

■ Un système d'exploitation est un programme ou un ensemble de programmes assurant la gestion de l'ordinateur et des périphériques. Il sert d'interface entre le matériel (hardware) et le logiciel (software). C'est un ensemble de programmes très complexes dont le but est de rendre plus simples les programmes et l'utilisation de l'ordinateur.

Le système d'exploitation propose aux programmeurs une interface de programmation d'applications appelée **API**, *Application Programming Interface*. Tous les programmeurs utilisent les mêmes fonctions dans leurs programmes ce qui simplifie fortement le travail. Ils peuvent se concentrer sur le but de leur programme (créer un traitement de texte par exemple) sans avoir sans arrêt à écrire des morceaux de programmes pour gérer le disque dur, l'imprimante ou comment accéder au clavier. C'est le rôle du système d'exploitation de gérer :

- la mémoire,
- les accès aux périphériques,
- les données sur les disques,
- les programmes,

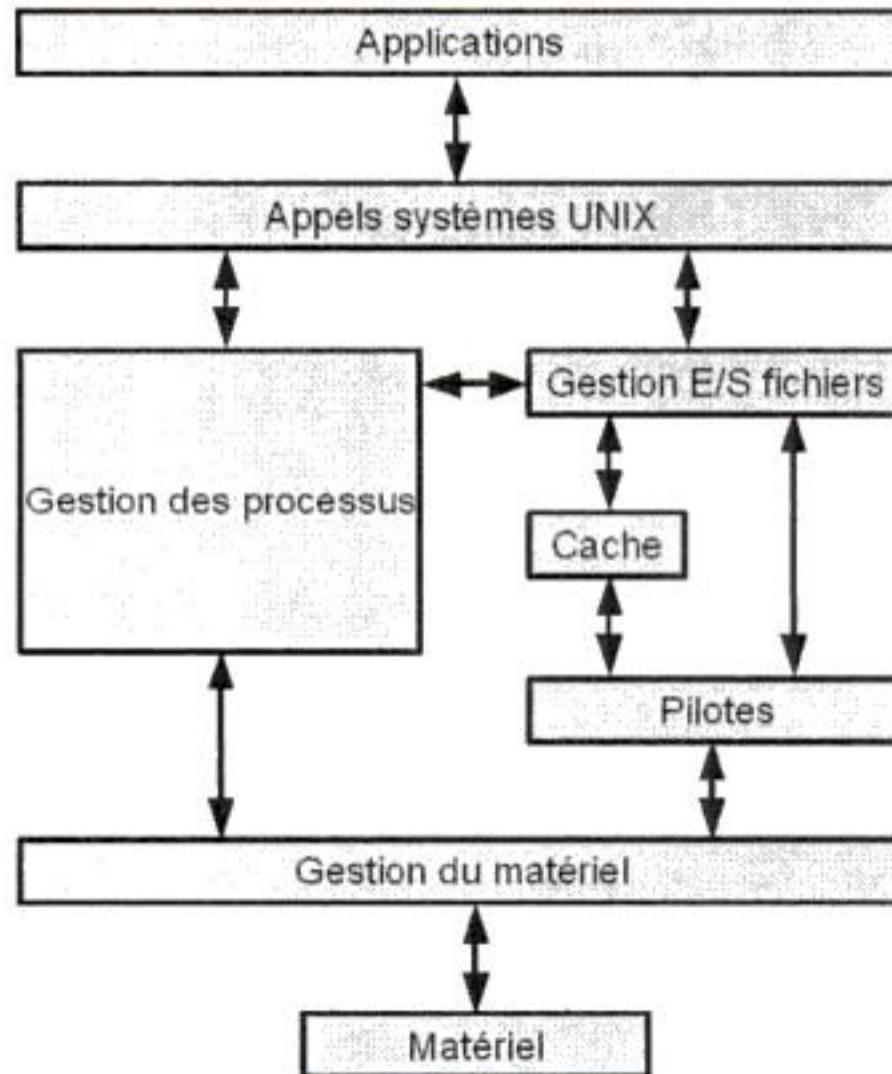
- la sécurité,
- la collecte des informations.

Il manque l'interface graphique. Dans un produit comme Microsoft Windows l'interface graphique est incluse au sein même du système d'exploitation. Il est d'ailleurs impossible de travailler sans, le moindre réglage se fait depuis une boîte de dialogue. Les utilisateurs ont de ce fait tendance à intégrer l'interface graphique comme composant de tout système d'exploitation. Historiquement l'interface graphique ne fait pas partie du système d'exploitation. Elle vient en complément. Vous ne trouverez aucun livre sur la théorie des systèmes d'exploitation traitant des interfaces graphiques. Quel est l'intérêt, sauf à ajouter de la lourdeur et occuper de précieuses ressources de la machine, d'avoir une interface graphique pour faire fonctionner un serveur Internet ? Linux propose des interfaces, mais ce sont des programmes comme les autres.

L'interface graphique n'est pas un composant du système d'exploitation Linux qui n'en a pas besoin pour fonctionner correctement. C'est un ensemble de plusieurs programmes classiques exécutés au-dessus du système d'exploitation, qu'il utilise.

Linux est un système d'exploitation de type Unix. Il existe des dizaines de systèmes d'exploitation dans cette famille. Unix est un système d'exploitation de la famille des systèmes multitâches et multi-utilisateurs :

- **Multitâche** : le système gère l'exécution simultanée de plusieurs programmes appelés des processus (Note : un vrai multitâche nécessite d'avoir plusieurs microprocesseurs ou équivalents – Hyper Threading par exemple).
- **Multi-utilisateurs** : le système permet l'existence de plusieurs utilisateurs différents sur une même machine, connectés ou non (un utilisateur peut faire tourner un programme sans être connecté, comme par exemple un serveur Internet).



Le schéma précédent est une synthèse simplifiée de la structure interne d'un système d'exploitation Unix. En bas se trouve votre matériel, en haut les programmes que vous faites fonctionner sur votre machine.

Entre les deux les divers composants du système assurent son bon fonctionnement :

- Les **appels systèmes** sont utilisés par les programmes pour communiquer avec le système d'exploitation Unix.
- La **gestion des processus** s'occupe de la commutation des tâches et de leur priorité. Ce composant s'occupe donc du multitâche.
- La **gestion des entrées et des sorties fichiers** s'occupe aussi bien de la lecture et de l'écriture des données sur vos disques durs mais aussi sur vos périphériques (carte son, imprimante, etc.)
- Certaines informations peuvent être placées dans une zone mémoire tampon appelée **cache**. Plutôt que d'écrire des données directement sur le disque dur (ce qui est lent), Unix va les écrire dans une zone mémoire puis ensuite les écrire sur le disque après quelques secondes. Ainsi, la relecture de ces données est plus rapide car elles sont déjà en mémoire et le logiciel ne perd pas de temps à attendre la fin de l'écriture des données.
- Les **pilotes** ont pour rôle de gérer au plus bas niveau le matériel ou les structures logiques du matériel (par exemple les données d'une partition).

► Une application bien programmée sur un système d'exploitation bien programmé ne peut pas court-circuiter ce schéma : elle ne « discute » jamais avec le matériel et passe obligatoirement par les API fournies.

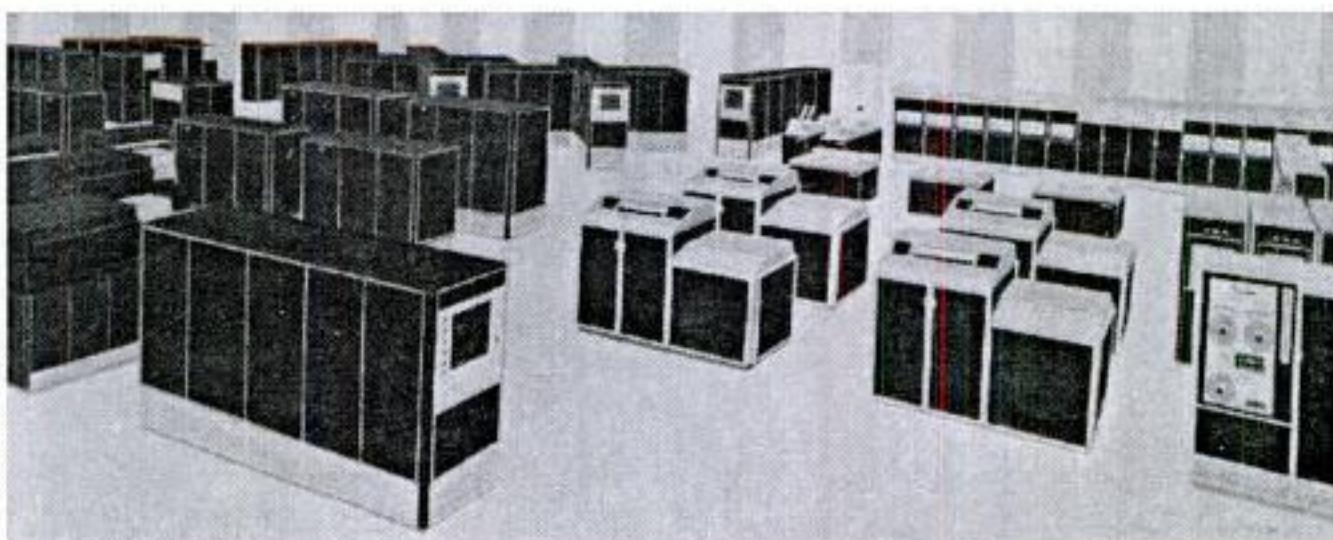
4. Le système Unix, une brève histoire

a. De MULTICS à UNIX

L'histoire d'Unix débute en **1964** quand le MIT, le laboratoire Bell Labs de AT&T et la General Electric commencent à développer le projet expérimental **MULTICS** (*Multiplexed Information and Computing Service*). Le projet Multics répond à de nouveaux besoins :

- pouvoir être utilisé par plusieurs personnes à la fois,
- pouvoir lancer des traitements en tâche de fond,
- une gestion accrue de la sécurité.

Multics était développé sur un gros système GE-645 de General Electric, équipé de deux processeurs sachant traiter chacun 435 000 instructions par seconde, trois unités de mémoire de 1 Mo chacune et 136 Mo de stockage. Il a fonctionné au MIT jusqu'en 1988, 82 sites en ont disposé et un maximum de 200 utilisateurs a pu y travailler simultanément chez General Motors. La dernière installation Multics à avoir été désactivée est celle de la Défense Canadienne le 30 octobre 2000.



Le GE-645

Dès le début pourtant, si Multics a vite atteint le degré de stabilité suffisant pour passer en production, il s'est révélé avoir des performances moindres que celles attendues. En 1969 Bell Labs se retire du projet pour se tourner vers le développement d'un autre système appelé GECOS.

Ken Thompson, développeur chez Bell, continue cependant à travailler sur le GE-645 et écrit un jeu appelé Space Travel. Tournant avec Multics il se révèle très lent et coûteux à faire tourner en temps partagé (avant le multitâche, le temps de la machine était découpé en tranches et chaque tranche d'utilisation était décomptée et facturée).



Ken Thompson

Ken réécrit alors le jeu en assembleur pour le mini ordinateur DEC PDP-7. Il est aidé de **Dennis Ritchie** qui lui aussi vient de Bell Labs. Cette expérience combinée avec celle de la conception de Multics pousse les deux hommes et leur équipe à créer un nouveau système d'exploitation pour le PDP-7. **Rudd Canaday**, encore de Bell Labs, était justement en train de développer un nouveau système de fichier qu'il voit comme un système d'exploitation. De là vient le fait que Unix est un système orienté fichier, où tout (ou presque) est fichier. Ils y rajoutèrent un interpréteur de commandes et quelques utilitaires. Ils nommèrent le système **UNICS** (*Uniplexed Information and Computing System*), selon une idée de **Brian Kernighan**. Le projet pouvait déjà gérer, dès le début, deux utilisateurs en même temps en vrai multitâche.



Le DEC PDP-7

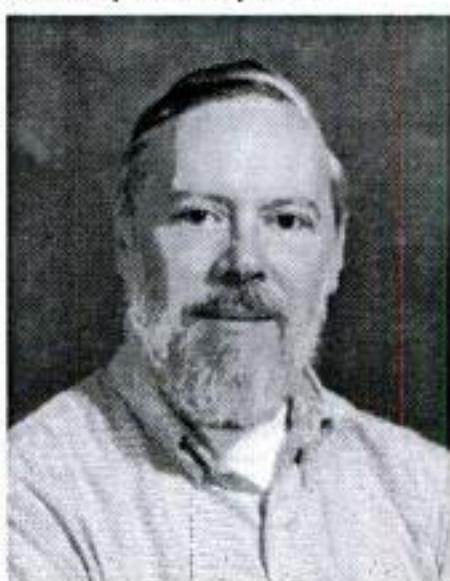
L'origine du mot UNICS est le sujet de nombreuses légendes, qui ont probablement toutes leur part de vérité. UNICS est une dérision de MULTICS dont l'architecture était sujette à de nombreuses critiques à l'époque : « MULTICS (multiple) faisait la même chose de plusieurs façons alors que UNICS (unique) faisait chaque chose de la même façon ». L'autre fait est qu'en anglais UNICS se prononce comme « eunuchs » c'est-à-dire eunuque, un système Multics « castré ».

UNICS reprend les concepts essentiels développés pour MULTICS en les améliorant. Notamment ses concepteurs proposent un tout nouveau système de communication entre les programmes où un premier programme peut renvoyer ses données à un autre programme. Rapidement le CS est remplacé par un X, une lettre de moins pour la même présentation. La légende **UNIX** venait de naître.

Tout aurait pu s'arrêter là car les équipes travaillaient sans aucun financement, Bell Labs s'étant en principe totalement désengagé de Multics et de ses successeurs. Pour continuer les travaux, Thompson et Ritchie proposent à Bell Labs l'ajout d'un traitement de texte à Unix pour le PDP-11/20. Bell acceptant, la machine est mise à disposition et l'équipe obtient un financement et un support officiels. L'outil *runoff* (qui deviendra roff puis troff) et l'éditeur *ed* sont développés et pour la première fois en **1970** la dénomination **Unix Operating System** est utilisée. Bell utilise alors Unix comme un système de traitement de texte pour la rédaction de ses brevets. Le premier manuel de programmation Unix date du 3 novembre 1971.

b. Le langage C

Un nouveau problème apparaît rapidement. Développé en assembleur et donc en langage machine, Unix doit être en partie réécrit pour chaque nouveau modèle d'ordinateur DEC. Or le langage machine est un art difficile. La question de la portabilité se pose alors. Dès 1970, Thompson se penche sur le problème. Il pense tout d'abord à développer Unix en langage TMG puis Fortran. Trouvant le langage incomplet il s'associe avec Dennis Ritchie pour créer le **langage B**, issu du langage **BCPL**. Là encore ça ne convient pas (problème avec le typage des variables et les nombres réels). Ritchie part du langage B et développe le **New B** qu'il appelle logiquement le **langage C**. Le langage C est transformé en langage machine une fois passé par une étape de compilation. L'écriture des programmes est plus rapide.



Dennis Ritchie

Unix est réécrit en langage C à partir de **1973**. Pour passer Unix d'une machine à une autre il suffit qu'un compilateur C soit disponible sur la nouvelle machine. Il est beaucoup plus simple et rapide d'écrire un compilateur C (lui-même écrit en grande partie en C) que de réécrire tout un système d'exploitation en assembleur. Seules les parties très proches de l'architecture matérielle de la machine sont écrites en langage machine. Unix devient portable et son développement s'accélère.

c. Les licences et l'avènement de BSD et System V

Un premier événement majeur va alors contribuer à la large diffusion d'Unix (le mot large prend une signification particulière lorsqu'on parle de quelques dizaines de copies). AT&T, dont dépend Bell Labs, a fait l'objet en 1956 d'un décret antitrust lui interdisant de commercialiser d'autres produits que ceux situés au cœur de son métier : les télécommunications. Il ne peut pas vendre Unix. AT&T (qui n'en voit même pas l'avenir commercial) décide en 1974 de diffuser le système UNIX complet à des fins éducatives auprès des universités et des entreprises sous une licence peu restrictive. Seul le code source (*le programme sous forme de texte compréhensible et pas encore compilé*) du noyau en assembleur n'est pas officiellement diffusé ou via des moyens détournés. Les versions les plus diffusées sont la sixième en 1975 et la septième en 1978. Unix v7 est la première version à avoir été spécifiquement retravaillée afin d'être portée sur d'autres machines que les PDP, notamment sur le VAX 11/780. La v7 est considérée comme la dernière version entièrement commune à tous les Unix suivants.

Le second événement majeur se produit à ce moment. Alors que Unix va fêter ses dix ans et que les universités américaines contribuent fortement à sa diffusion et son amélioration, AT&T rend la licence d'Unix plus restrictive. La branche commerciale d'Unix est en effet autorisée à vendre des licences du code source. Les tarifs prohibitifs forcent les universités à continuer pour le meilleur et pour le pire leurs développements à partir des développements antérieurs à cette nouvelle licence. L'une de ces universités est celle de Californie, appelée **Berkeley**. Berkeley est le plus gros contributeur à Unix sur lequel elle a commencé à travailler dès 1974. La version 1 de **BSD** (*Berkeley Software Distribution*) est basée sur Unix v6 en 1977 et est appelée **1BSD**. La version 2BSD basée sur Unix v7 date de 1978.

À partir de là deux écoles vont s'affronter. La première, en théorie officielle, est celle de AT&T qui va continuer à développer les versions 8, 9 et 10 durant les années 1980 dans des buts de recherche. Dans le même temps, elle développe un Unix entièrement commercial appelé **Unix System III** et le vend dès 1982. En 1983 AT&T développe et vend les premières versions Unix System V. La dernière version, **Unix System V release 4.2** date de 1993. On note cette version d'Unix par l'abréviation SVR4. Son code source est disponible sous licence. Un organisme peut en acheter une et développer sa propre version commerciale.

Durant ce temps, l'université de Berkeley ne chôme pas et continue le développement de BSD comme alternative sous licence Open Source de Unix System III et V dont elle n'a plus le droit d'utiliser les sources. C'est dans BSD que va être implémenté pour la première fois le protocole **TCP/IP**, base de l'Internet moderne, grâce au financement du ministère américain de la défense. La dernière version officielle de BSD est **4.4BSD** en juin 1994.

d. La guerre des Unix

La période allant du milieu des années 1980 à 1994 n'est pas de tout repos. Les effets de la séparation de Unix en deux branches ont été désastreux et ont failli causer sa perte. Les deux camps (AT&T avec son System V et Berkeley avec son BSD) ne s'entendent pas sur un standard commun. L'effet, outre les multiples procès (jusqu'en 1993) sur l'utilisation du nom et des outils dérivés d'Unix, est que de multiples versions d'Unix commerciales et surtout incompatibles entre elles ont poussé comme de la mauvaise herbe. C'est de cette époque que datent les grands noms des clones Unix dont **Solaris**, **AIX**, **OSF1 / Digital Unix / True64**, **Xenix**, **HP-UX**, **IRIX**, **Ultrix**, **Unixware**, **A/UX**, tous souvent incompatibles avec le voisin mais clamant haut et fort leur appartenance à Unix. Cette guerre des Unix est réellement connue comme la période sombre des « *Unix wars* ». Personne n'arrivant à se mettre d'accord sur une base et un standard communs. L'effet direct de cette guerre a été la création d'une niche dans le marché des systèmes d'exploitation dans laquelle la société Microsoft s'est largement engouffrée avec son système d'exploitation **Windows NT** (qui, peu de monde le sait, est aussi dérivé d'Unix).

En 1984 un groupe d'éditeurs d'Unix commerciaux tente une première standardisation en créant X/Open Standards afin de diffuser un document appelé **X/Open Portability Guide** décrivant un standard ouvert (accessible à tous) pour Unix. Ce comité aboutit en 1987 quand Sun Microsystems

et AT&T décident de travailler sur un Unix uniifié, fusion de BSD et de System V. Le résultat est en fait System V Release 4.

La jalousie est un vilain défaut. La concurrence accuse Sun de vouloir devenir le maître du jeu et fonde *Open Software Foundation* soit OSF en 1988. OSF se veut lui aussi LE standard ouvert Unix, sauf qu'il se base ouvertement sur BSD. Ses spécifications sont connues en 1990.

En réponse, AT&T et un nouveau groupe créent *Unix International* en 1989 dans une énième tentative d'unification. Sans plus de succès. Devant cet imbroglio AT&T décide de se débarrasser d'Unix dont elle est toujours officiellement propriétaire et crée pour cela une société appelée *Unix System Laboratories* en 1992. Tous les droits d'Unix sont transférés à USL.

e. La standardisation

Alors que la situation semble bloquée, un nouvel acteur apparaît et va réussir là où les autres ont échoué. La société **Novell** rachète USL l'année de sa création et devient propriétaire de SVR4.2. En 1993 Novell cède la marque Unix à X/Open. Unix International disparaît en 1994 et OSF est restructurée. Enfin, en 1995 Novell cède la licence d'exploitation du code source d'Unix à la société **SCO Santa Cruz Operations** (qui deviendra Caldera puis de nouveau SCO). La même année X/Open et OSF fusionnent définitivement et deviennent **The Open Group**.

Il n'existe qu'un seul organisme de standard Unix. Unix est de ce fait un standard ouvert : ses spécifications sont connues et chaque éditeur de système Unix commercial ou gratuit désirant assurer une compatibilité avec l'ensemble des Unix doit implémenter ce standard. Chaque éditeur est cependant libre de programmer ce standard comme il le souhaite, une même fonction pouvant être écrite de plusieurs manières. Les dégâts des *Unix Wars* ont été nombreux permettant l'émergence de nouveaux systèmes d'exploitation comme Windows NT de Microsoft.

f. Unix est un standard

Pour s'assurer que tous les Unix suivent les mêmes recommandations, The Open Group diffuse des normes (*Single Unix Specification*, *Unix95*, *Unix98*, *Linux Standard Base*, etc.) et peut faire passer des certifications. Ces normes s'appuient aussi en partie sur celles définies depuis 1988 par l'**IEEE Institute of Electrical and Electronics Engineers** (que l'on prononce I3E) et notamment **IEEE 1003** aussi appelée **POSIX** (*Portable Operating System Interface*). Le X est un héritage et une reconnaissance du travail effectué sur Unix. IEEE 1003 est composée de 15 documents regroupant par thèmes tout ce qu'un Unix doit contenir (commandes de base, interpréteur de commandes, interfaces utilisateur, fonctions de programmation, etc.) pour être conforme au standard POSIX. POSIX n'est pas limité à Unix. Windows NT est conforme à POSIX pour certains de ses composants. POSIX n'est pas un standard ouvert. Les spécifications de The Open Group sont ouvertes et accessibles à tous et les éditeurs préfèrent s'y référer.

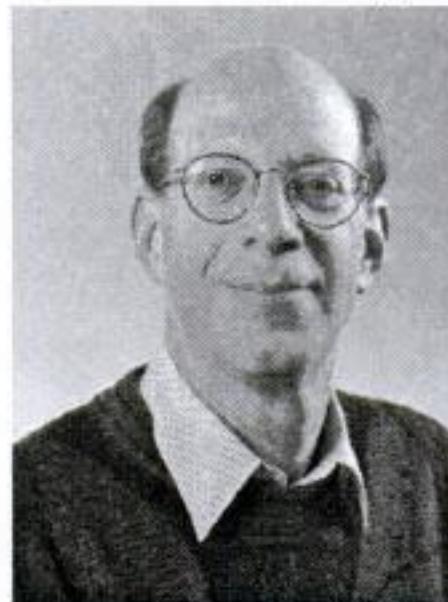
Pour être utilisé dans certaines administrations américaines, un système d'exploitation Unix doit être conforme au standard POSIX. De ce fait quand Linux a dû être utilisé, le gouvernement de Bill Clinton a entièrement fait financer la certification **PCTS** (*Posix Conformance Test Suite*) par le Trésor américain.

Les dernières versions officielles des versions BSD et System V datent de 1994. Les Unix conçus à partir de 1995 implémentent les recommandations de The Open Group. Cependant historiquement quelques versions continuent à être « orientées » plutôt BSD, ou plutôt System V concernant leur configuration ou parfois les deux, comme Linux (et selon la distribution).

g. Unix sur les ordinateurs personnels

Le premier Unix pour ordinateur personnel, au sens ordinateur de type IBM PC est **Xenix**. Il est issu de Unix v7 et est sorti en 1983 sur PC (des versions ont été disponibles plus tôt sur d'autres architectures matérielles). C'est la société Microsoft qui a effectué le portage de Xenix, au prix de nombreuses modifications. La version 2 de Xenix date de 1985 et est basée sur Unix System V. Lorsque IBM démarre le développement de OS/2 en association avec Microsoft, ce dernier

transfère les droits de Xenix à SCO en 1987. La version 2.3.1 de cette même année supporte le 386, le SCSI et TCP/IP. Xenix devient SCO Unix en 1989 puis disparaît au profit de SVR4.



Andrew Tanenbaum

L'américain **Andrew Stuart Tanenbaum** (surnommé Andy) est chercheur et enseignant en informatique et actuellement à la tête de l'Université libre d'Amsterdam. Il est aussi l'auteur d'ouvrages de références en informatique sur la théorie des systèmes d'exploitation. En 1987, dans un but pédagogique, il conçoit et écrit le système d'exploitation Minix. Il utilise 20 Mo d'espace disque et ne nécessite que peu de ressources étant parfaitement à l'aise avec 2 Mo de mémoire vive. Minix aura une grande importance pour Linux. Minix existe toujours et la version 3 est sortie en octobre 2005.

De nombreux dérivés de BSD ont été portés sur PC. Le premier est **386BSD** en octobre 1989 et dérive de 4BSD. S'il existe encore c'est son successeur **NetBSD** qui est le plus connu, dérivant lui-même de 4.3BSD et de 4.4BSD. NetBSD est le système d'exploitation qui a été le plus porté sur d'autres architectures matérielles. **OpenBSD** est dérivé de 4.4BSD et très orienté sécurité. En huit ans une seule faille de sécurité a pu être exploitée. **FreeBSD** est aussi dérivé de 4.4BSD et est issu directement de l'époque des procès entre BSDI et AT&T. De ce fait, FreeBSD est entièrement libre et ouvert.

Solaris, l'Unix de Sun Microsystems est disponible depuis plusieurs années sur PC et la version OpenSolaris (version 10) est Open Source.

Enfin, **Linux** est probablement l'Unix libre le plus connu et répandu sur le PC. L'histoire de sa création mérite bien quelques détails supplémentaires.

B. Le logiciel libre

1. Les origines du logiciel libre

Unix est le parfait exemple du travail qui peut être effectué quand toutes les énergies sont canalisées à la recherche d'un idéal technologique. Quand AT&T diffuse presque librement en 1974 le code source du système d'exploitation auprès des universités parce que, parmi ses raisons, il ne voit pas d'avenir économique pour son produit, il ne semble pas se douter de l'engouement des étudiants, des professeurs et des chercheurs en informatique. Cette première communauté va passer beaucoup de temps à modifier et à améliorer le produit, remontant toutes les nouveautés à AT&T pour une intégration dans le produit officiel. Au contraire lors du changement de licence de 1978 l'énergie de la communauté a été canalisée vers le projet universitaire BSD délaissant l'Unix commercial de AT&T. Notez que les plus grandes avancées eurent lieu avec l'Unix de Berkeley.

Les premiers ordinateurs étaient essentiellement des outils de recherche pour les universitaires (et aussi des monstres de calcul pour des besoins militaires). Dans les laboratoires de recherche, les logiciels circulaient comme les idées : librement. Il n'y avait rien de plus banal qu'un logiciel

développé par une équipe de programmeurs ou de chercheurs soit diffusé à d'autres équipes d'autres universités et partout où il y en avait besoin. Il n'y avait rien de plus normal que ce logiciel soit modifié par une autre équipe, et ainsi de suite. Aujourd'hui encore quand un illustre mathématicien démontre un théorème difficile il diffuse le résultat de ses travaux dans des ouvrages spécialisés dans le but de faire avancer la science. Tout le monde y a accès.

Mais l'informatique n'a pas suivi le même chemin. Bien que science, le fruit des recherches en informatique ne s'est pas restreint au monde des universitaires. Rapidement les entreprises ont pu voir l'immense intérêt d'automatiser certaines de leurs tâches comme la comptabilité, la paie, etc. Avec l'achat des premiers gros ordinateurs de gestion il fallait des programmes. Ces programmes ont commencé à être protégés comme des secrets industriels et une nouvelle branche commerciale est née : l'édition de logiciels. Une fois arrivée dans le monde des affaires, l'informatique est devenue très rapidement beaucoup moins libre. On s'est mis à parler de licences, de taxes et de redevances, de droit d'auteur (qui n'empêche pas d'autoriser selon le cas la copie), de limitation des droits, d'interdiction de copier, etc.

2. Le projet GNU et la FSF

Richard Stallman n'a probablement pas été le premier à déplorer ce fait mais a décidé de réagir. Informaticien au laboratoire d'intelligence artificielle au MIT à la fin des années 1970, il utilise une imprimante qui tombe souvent en panne. Comme ses collègues et lui disposent du code source du pilote (programme de gestion) de l'imprimante ils l'ont modifié pour qu'un signal leur soit envoyé à chaque panne. Quand le laboratoire achète un nouveau modèle de Xerox plus fiable, le pilote pour leur système d'exploitation n'est pas livré. Désirant l'adapter à ses besoins, Richard Stallman fait appel à un autre laboratoire qui dispose du code source mais qui refuse de le lui fournir : Xerox l'interdit. Ainsi l'imprimante ne marchera jamais, et Stallman est tellement choqué de cette réaction qu'il décide d'œuvrer dans la défense et la diffusion du logiciel libre en réaction au monde fermé du logiciel propriétaire.



Richard Stallman

Stallman décide en 1983 d'écrire un nouveau système d'exploitation entièrement libre d'accès, d'utilisation, de modification et de redistribution. Basé sur Unix il le nomme **GNU** (*Gnu's Not Unix*). Les acronymes récursifs sont très à la mode chez les informaticiens. On trouve l'annonce du projet et des motivations de Stallman sur <http://www.gnu.org/gnu/initial-announcement.html>. Pour son système il a besoin d'un noyau (le cœur du système d'exploitation) et d'outils (pour gérer les fichiers par exemples). Ce n'est pas un coup d'essai pour Stallman qui a déjà écrit un grand éditeur de texte appelé **Emacs**. Les premiers développements vont très vite et les outils sont très nombreux et souvent de meilleure qualité que ceux du commerce. Par contre la conception d'un noyau Unix est beaucoup plus complexe et nécessite une phase théorique importante. Le projet **HURD** (*Hird of Unix Replacing Daemons*) est lancé. Il n'a toujours pas abouti.

La bataille n'est pas que technique, elle est aussi politique, philosophique, commerciale et juridique. Pour défendre le logiciel libre Stallman crée la **FSF** (*Free Software Foundation*) en 1985 qui diffuse les idées du logiciel libre. Parmi ses premiers travaux figure la rédaction (avec l'aide d'avocats) d'une licence spéciale pour ces logiciels appelée la **GPL** (*General Public License*). Un logiciel libre garantit quatre libertés :

- **Liberté 0** : la liberté d'utiliser un logiciel quel que soit l'usage que vous en faites.
- **Liberté 1** : la liberté d'étudier le fonctionnement du programme et de l'adapter à votre besoin.
- **Liberté 2** : la liberté de redistribuer des copies afin d'aider votre voisin (au sens large du terme).
- **Liberté 3** : la liberté d'améliorer le programme et de diffuser les améliorations au public à fin d'en faire bénéficier l'ensemble de la communauté.

Les libertés 1 et 3 nécessitent d'avoir obligatoirement accès au code source du programme. La liberté 3 définit la notion de communauté autour du logiciel libre.

Remarquez que le mot « gratuit » n'est indiqué nulle part. En anglais « free » signifie tant libre que gratuit. Le logiciel libre est à prendre dans le sens de « liberté » et pas gratuit (Free as a speech et non pas Free as a beer comme disent les anglais). Il est tout à fait possible et même parfois conseillé de commerçer avec le logiciel libre. Mais comme les libertés 2 et 3 autorisent la diffusion du logiciel, il est toujours possible d'en récupérer une copie gratuitement et ce tout à fait légalement. La gratuité est un effet de la liberté telle que définie pour le logiciel libre.

Les travaux de HURD avancent peu ou mal. Ses développeurs ont pris le pari de développer un micro-noyau : les composants de base du système d'exploitation sont « éclatés » en plusieurs sous-unités indépendantes mais devant communiquer ensemble. Le choix théorique est excellent mais l'implémentation technique est très difficile. GNU ne dispose pas de noyau. C'est Linux qui va faire aboutir le projet en 1992 quand il passe sous licence GPL.

3. L'Open Source

Et l'**Open Source** ? L'expression est apparue en 1998 quand Netscape Communicator est devenu un logiciel libre. L'expression **Open Source** (source ouverte) était utilisée dans les slogans pour associer libre et diffusion du code source et faire comprendre et admettre les logiciels libres auprès des entreprises. Le but était de faire abstraction des apports fondamentaux du libre pour se concentrer uniquement sur les avantages techniques et économiques de ce nouveau modèle. Avec le temps, l'expression a été reprise dans tous les sens par les médias et les entreprises, et sa définition a été largement entachée. On a parlé de « Open Source limité » en proposant l'accès aux sources mais sans droit de modification ou de redistribution. Or, le logiciel libre ne souffre d aucun aménagement. Il est libre ou n'est pas.

Si vous voulez être certain que le programme que vous utilisez est libre, vérifiez le nom de la licence et rendez-vous sur le site de **OSI « Open Source Initiative » <http://www.opensource.org> qui en recense la majorité des plus connues. C'est une initiative de Eric S. Raymond (ESR) grand hacker (spécialiste de très haut niveau) et l'un des grands noms de l'Open Source. Parfois en conflit avec Richard Stallman, leurs deux visions (techniques pour ESR, philosophiques pour Stallman) sont pourtant complémentaires.**

4. GNU/Linux

a. Linus Torvalds

L'histoire de Linux commence quand Linus Torvalds, jeune étudiant finlandais à l'université de Helsinki âgé de 21 ans, acquiert en 1991 un ordinateur à base de 386 pour remplacer son Sinclair QL qui commence à montrer ses limites. Le 386 est un microprocesseur 32 bits génial qui gère, entre autres, la mémoire virtuelle et la commutation des tâches. Mais le gros problème est qu'un PC est livré avec MS-DOS, un système d'exploitation loin d'être optimal et surtout n'exploitant aucune possibilité de ce processeur. Linus eut alors l'idée d'installer un autre système appelé Minix, un petit Unix simple et gratuit développé par le célèbre Andrew Tanenbaum, qui permet d'exploiter

son beau PC tout neuf acheté à crédit. Linus se met à travailler et à développer dessus. Son but est d'apprendre le fonctionnement du 386, notamment la commutation des tâches en langage assembleur. Il commence à travailler sur un projet assez simple : un émulateur de terminal, entièrement en assembleur, pour se connecter au serveur de son université.



Linus Torvalds

b. L'accident

Oui mais voilà qu'un jour, suite à une mauvaise manipulation, il efface par accident les premiers secteurs d'amorce de la partition de son disque dur contenant Minix, effaçant ainsi son principal outil de développement. Il ne reste que deux solutions : soit tout réinstaller, soit partir de son existant et l'étoffer de manière à le rendre autonome. Bien entendu, l'environnement de développement est réinstallé, mais Linus décide d'améliorer son projet, en lui rajoutant le nécessaire : code de base, pilote rudimentaire de disque dur, passage au langage C, etc. Le 25 août 1991, la version 0.01 est prête et diffusée dans la plus grande indifférence ou presque. Pour les outils, rien de plus simple, le projet GNU initié par Richard Stallman dispose déjà de tout le nécessaire. Linux sera le noyau qui manque au système d'exploitation GNU.

c. La première version officielle

Le but de Linux est de faire quelque chose qui dépasserait Minix. Par la première version diffusée, il faut que le shell (interpréteur de commandes) et gcc (compilateur C) soient utilisables. C'est le cas pour la version 0.02 annoncée le 5 octobre 1991 sur le groupe comp.os.minix :

« Vous regrettiez les beaux jours de Minix-1.1, époque bénie où les hommes étaient dignes de ce nom et écrivaient leurs propres pilotes de périphériques ? Vous cherchez à vous investir dans un projet original et vous vous languissez d'un système modifiable à votre convenance ? Vous êtes frustré que tout fonctionne sous Minix ? Vous regrettiez les nuits blanches passées à tenter d'implanter un programme récalcitrant ? Si tel est le cas, lisez ce qui suit : Comme signalé il y a un mois, je travaille actuellement sur une version libre d'un système analogue à Minix pour ordinateur AT-386. Ce système est à présent utilisable (mais peut-être ne vous conviendra-t-il pas, tout dépend de ce que vous recherchez) et je compte en diffuser les sources. Il s'agit pour l'instant de la version 0.02, capable néanmoins d'exécuter bash, gcc, gnu-make, gnu-sed, compress, etc. »

d. Le succès communautaire

À partir de ce moment le succès, grâce à la diffusion par Internet, est au rendez-vous et les contributions commencent à arriver. Une communauté Linux se forme. La version 0.03 arrive, puis la 0.10. En 1992, Linux peut enfin faire fonctionner l'interface graphique X11. De ce fait, Linux pense qu'il est temps d'accélérer le mouvement et nomme la version suivante 0.99, pensant toucher au but. Ce fut, rétrospectivement, une erreur. En effet, il a fallu attendre 18 mois que la version 0.99pl114 soit finalisée et qu'enfin la version 1.0 sorte en janvier 1994. Entre les premiers

balbutiements et la version 1.0 il y a eu quelques rebondissements, notamment le conflit « technique » entre la conception monolithique de Linux et la vision micro-noyau de Andrew Tanenbaum, ce dernier engageant les hostilités avec la fameuse phrase : « Linux est obsolète ».

e. Les années 1994-1997

Ces années voient l'apparition des grandes distributions Linux que vous connaissez encore aujourd'hui : Red Hat, Debian, Suse, Slackware. Mandrake est arrivé un peu plus tard. Durant ces années, Linux ne cesse de s'améliorer, avec l'arrivée notable de la modularité et de la version 2.0. C'est surtout durant ces années que Linux sort du petit monde des hackers et se fait connaître en entreprise. Les projets foisonnent, et déjà l'idée d'améliorer le système et de l'ouvrir au monde du bureau (desktop) fait son bout de chemin avec le début du développement de produits comme Gnome ou KDE.

La mascotte de Linux appelée **Tux** date de 1996 et a été créé par Larry Ewing à l'aide du logiciel libre GIMP. Tux (apocope de *Tuxedo et Torvalds UniX*) n'est pas un pingouin mais un manchot pygmée. Le fait est que le mot anglais penguin désigne dans cette langue aussi bien le véritable pingouin (*razorbill*) que le manchot, d'où une certaine confusion.



Tux, la mascotte de Linux

f. À partir de 1998 : l'explosion

On ne sait pas si c'est dû à un ras le bol général des utilisateurs, mais l'année 1998 est celle d'annonces spectaculaires. Le monde de l'informatique réalise enfin que Linux n'est pas qu'un joujou pour étudiant bidouilleur. En janvier 1998, Netscape annonce que son produit passe en Open Source. Il en sortira Mozilla, Firefox et Thunderbird. Les instituts de formation ajoutent Linux à leur catalogue. En juillet 1998, Oracle et Informix sont portés. En septembre, IBM porte DB2 et Sybase fait de même. Linus Torvalds fait la une de « *Forbes* ». KDE et Gnome arrivent en version 1.0. En bourse, les cours montent, les sociétés Linux voient le jour. C'est le succès.

Janvier 1999, c'est l'arrivée de Linux 2.2 et la continuité du succès, qui commence à faire réagir Microsoft. C'est David contre Goliath. C'est toujours le cas. On aurait pu croire que l'explosion de la bulle Internet en bourse en 2000 allait tout faire capoter. Vous constatez que non. Linux n'est pas un colosse aux pieds d'argile. Ses pieds, c'est la communauté, inébranlable. Le noyau 2.4 sort le 4 janvier 2001. Le noyau 2.6 sort le 18 décembre 2003.

g. Aujourd'hui et demain

Aujourd'hui Linux est reconnu comme un système d'exploitation stable, robuste et performant. Il est utilisé dans plus du tiers des serveurs dans le monde et dans la moitié des serveurs web. Il a conquis le monde de l'entreprise, le monde universitaire. Il a surtout su conserver son indépendance, garantie par la communauté et le nombre de contributeurs, face aux géants de l'informatique. La prochaine grosse cible de Linux, c'est le poste de travail, et pourquoi pas, l'usage familial en remplacement de Windows. Il reste encore un peu de chemin, mais nombreux sont ceux qui ont déjà franchi le pas.

C. Quel matériel pour Linux ?

1. L'architecture

Linux existe pour au moins trois architectures matérielles courantes :

- **x86** pour les ordinateurs dont les processeurs sont du type Intel (du 386 au Pentium 4) ou AMD (Athlon, Duron, Sempron) 32 bits. Cette version fonctionne aussi sur les machines à base de processeurs 64 bits.
- **x86_64** pour les ordinateurs dont les processeurs sont du type Intel (Pentium 4 à partir des séries 600, Xeon, Dual Core/Quad Core) ou AMD (Athlon 64, Sempron 64, Opteron) 64 bits. Cette version ne marche pas sur les processeurs 32 bits.
- **ppc** pour les ordinateurs dont les processeurs sont de type PowerPC c'est-à-dire les anciens ordinateurs de marque Apple. Cette version ne s'installera pas sur les dernières machines Apple basées sur un processeur de marque Intel.

► Certains pilotes matériels ou applications sont encore peu ou mal adaptés à la version 64 bits. Si vous constatez des dysfonctionnements gênants, pensez à installer la version 32 bits qui devrait résoudre vos problèmes. N'oubliez pas que Linux est le premier système d'exploitation offrant le support complet des processeurs 64 bits et que ces problèmes sont le reflet de la jeunesse de cette version.

Configuration matérielle de base

Linux supporte théoriquement tous les types de processeurs depuis la version 386, et peut fonctionner avec seulement quelques Mo de mémoire. La distribution Polux Linux fonctionne sur un 386 avec 4 Mo de mémoire. La distribution Damn Small Linux fonctionne avec un 486, 16 Mo de mémoire et utilise 50 Mo d'espace disque. On trouve même des distributions sur une ou deux disquettes démarrant avec 2 Mo de mémoire.

N'espérez cependant pas travailler correctement avec une version moderne de Linux et son environnement bureautique graphique dans ces conditions pseudo-préhistoriques. Les pré-requis suivants doivent être respectés :

- **Un processeur** (ou plus) de type Intel Pentium et supérieur ou un équivalent de marque AMD.
- **Au moins 128 Mo de mémoire**, mais 256 Mo ou plus apportent une réel confort d'utilisation. Pensez plutôt à disposer de 512 Mo voire 1 Go pour une utilisation optimale. Au prix de la mémoire ce n'est pas un luxe. Dans le cadre d'une installation minimale en mode texte, 64 Mo suffisent.
- 500 Mo d'espace disque pour une installation minimale (sans interface graphique et seulement les outils de base), mais 2,5 Go pour une installation standard, auquel il faut rajouter l'espace pour les données de l'utilisateur et la partition d'échange.
- Une carte graphique même ancienne compatible avec la norme Vesa, acceptant de préférence le 1024x768 en 65 356 couleurs pour l'environnement graphique, et sans aucune importance en mode texte.

► Ce sont des pré-requis de base. Si la fréquence d'horloge de votre processeur joue principalement pour la vitesse d'exécution de vos applications, elle peut être fortement bridée par le manque de mémoire ou un disque dur trop lent. La quantité de mémoire est un facteur important de confort. Plus il y en a mieux c'est : plusieurs programmes pourront fonctionner en même temps, la partition d'échange ne sera pas sollicitée et le système pourra utiliser plus de mémoire tampon pour accélérer les accès aux disques et périphériques. Si vous disposez de 256 Mo ou moins, envisagez de passer à 512 Mo. La différence est flagrante.

Les performances globales restent acceptables sur un Pentium II 300 avec 256 Mo pour une utilisation bureautique ou Internet simples. Les performances s'écroulent lors du lancement simultané de plusieurs programmes. Sur un simple AMD Duron 800 avec 512 Mo, les performances sont excellentes pour la plupart des usages classiques.

2. Compatibilité du matériel

Avant d'installer Linux vérifiez si votre matériel est correctement pris en charge par Linux. Établissez une liste des composants de votre ordinateur et de vos différents périphériques. Le support du matériel est régulièrement mis en avant par les débutants lorsque l'installation échoue. Telle carte graphique, telle imprimante, tel scanner ne fonctionnent pas correctement ou pas du tout. Bien que Linux supporte la plupart des composants des ordinateurs récents, l'achat d'un ordinateur dernier cri n'est pas une garantie de bon fonctionnement.

Dans la liste, le plus important n'est pas la marque et le nom du modèle commercial mais le composant, la puce principale, appelée « *chipset* », du produit. Dans le cas du Wi-Fi, peu importe que la carte soit une Palmnet BZ46G. Mais si vous savez qu'elle est construite autour d'une puce Centrino (Intel 2200 par exemple), alors vous trouverez vite qu'elle fonctionne avec Linux. Les produits de certains constructeurs doivent être évités car leurs matériels ne disposent pas de pilotes permettant de les utiliser. La quasi-totalité du matériel d'impression proposé par Hewlett Packard fonctionne parfaitement avec Linux alors qu'il faut fuir les imprimantes à jet d'encre Lexmark (attention : ce n'est pas la qualité du produit qui est mise en cause, mais son support sous Linux).

Sauf à disposer d'une machine très ancienne, toutes les cartes graphiques fonctionneront. Dans tous les cas Linux propose un pilote générique appelé « *vesa* » qui, s'il n'offre pas les meilleures performances, permet d'utiliser toutes les cartes compatibles avec ce standard vieux de plus de dix ans. Certains constructeurs proposent des pilotes très performants. Les dernières cartes des constructeurs Nvidia et ATI sont supportées avec des pilotes 3D offrant les mêmes performances qu'avec les autres systèmes d'exploitation. Le système graphique de Linux supporte par défaut un grand nombre de carte, y compris avec l'accélération 3D. Les meilleures cartes graphiques pour Linux restent les cartes à base de composants NVIDIA et Intel.

Les cartes son intégrées aux cartes mère respectent un standard de facto (AC97) qui est supporté par Linux. Les cartes son intégrées sur les cartes mère sont rarement des composants haut de gamme. Une simple carte Live coûtant moins de 30 euros est bien plus performante. Certains modèles spécifiques de cartes son peuvent poser des problèmes.

Le Wi-Fi devrait fonctionner soit avec un pilote natif pour votre matériel, soit à l'aide d'un outil particulier appelé Ndiswrapper qui permet d'utiliser les pilotes de Windows pour Linux. Selon votre choix de distribution de petits composants appelés firmwares et nécessaires à la carte Wi-Fi ne sont pas fournis par défaut et doivent être récupérés à part, soit depuis le système de mise à jour, soit sur un support (le fameux add-on) supplémentaire, soit chez le constructeur de la carte. Le Bluetooth est parfaitement reconnu et supporté.

Différents sites disposent de bases de données de matériels compatibles pour vous renseigner rapidement. Les moteurs de recherche restent votre meilleure source. À titre indicatif, voici une liste de sites qui vous aideront dans vos recherches :

- Liste de compatibilité Novell : http://cdb.novell.com/index.php?LANG=en_UK
- Liste de compatibilité openSUSE : <http://en.opensuse.org/HCL>
- Imprimantes : <http://www.linuxprinting.org>
- Scanners : <http://sane-project.org/>
- Périphériques USB en général : <http://www.qbik.ch/usb/devices/>
- Cartes son : <http://www.alsa-project.org/>
- Les cartes Wi-Fi : http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/
- Modems internes ou externes de type Windomem : <http://linmodems.org/>
- Webcams : <http://www.linux.com/howtos/Webcam-HOWTO/hardware.shtml>

Malgré toutes les bonnes volontés du monde il arrive parfois qu'un matériel ne fonctionne absolument pas avec Linux. À qui la faute ? Les pilotes de périphériques sont très souvent écrits par des développeurs n'ayant même pas accès aux spécifications du matériel et qui font tout par ingénierie inverse, c'est-à-dire en tentant de reproduire le fonctionnement du périphérique depuis son résultat. C'est très long. Certains constructeurs jouent le jeu. À défaut de fournir un vrai pilote ils diffusent auprès des développeurs une documentation technique. D'autres fournissent eux-mêmes un pilote au code fermé pour une distribution donnée ou que vous devez adapter vous-même à chaque nouvelle version de Linux. Dans ce cas l'avenir du pilote n'est plus garanti (ce qui s'est par exemple passé avec les cartes à base de chipset graphique Kyo II). Aussi avant de critiquer Linux si votre matériel ne marche pas critiquez en premier le constructeur de celui-ci.

D. Choisir une distribution

1. Debian



Le projet Debian a été fondé en 1993 par Ian Murdock à une époque où l'idée même de distribution Linux en était encore à ses balbutiements. Le nom Debian provient de Debra (la femme de Murdock) et Ian. Debian a longtemps été la seule distribution entièrement et uniquement composée de logiciels libres et Open Source ce qui lui vaut toujours le nom officiel de Debian GNU/Linux. Debian a aussi été supporté quelques temps officiellement par la FSF comme distribution Linux de référence. Les avantages de Debian sont nombreux :

- un nombre gigantesque de packages qui se chiffre en milliers,
- un logiciel d'installation appelé APT très pratique et performant,
- une distribution 100% open source,
- une stabilité à toute épreuve pour un environnement de production.

Ces avantages entraînent aussi quelques inconvénients :

- des packages souvent anciens,
- des mises à jour de la distribution irrégulières et trop espacées,
- des risques liés à la multiplication des paquets et des dépendances,
- une installation et une configuration compliquées.

Tous ces inconvénients ne sont pas forcément des défauts. Faut-il préférer une version ancienne mais totalement exempte de bugs ou la toute dernière version d'un produit dont la fiabilité n'a pas été pleinement éprouvée ?

Tous ces éléments font de Debian une distribution idéale pour les informaticiens, les ingénieurs et administrateurs système et réseau, les environnements de production en entreprise, les puristes du libre, les amateurs éclairés qui n'ont pas peur de mettre les mains dans le cambouis. Quant aux débutants ils passeront probablement leur chemin au début sauf à vouloir apprendre sur le tas.

a. Ubuntu



Le milliardaire sud-africain Mark Shuttleworth, principalement connu du monde entier pour avoir été l'un des premiers touristes de l'espace, mais aussi des informaticiens pour avoir fait fortune en revendant sa société Thawte spécialisée dans la sécurité à Verisign, est un vrai informaticien qui a contribué au projet Debian. Devant les quelques inconvénients de la distribution il crée la distribution Ubuntu Linux en 2005 avec un budget initial de 10 millions de dollars pour rémunérer les développeurs. Le mot Ubuntu est un mot du langage africain bantou signifiant « humanité aux autres » ou encore « je suis ce que je suis grâce à ce que nous sommes tous ». Cette définition reflète ce qu'est la distribution : un dérivé de Debian dont le but est de fournir des logiciels plus récents et très fortement axés sur la convivialité et l'ergonomie à l'aide du support du plus grand nombre :

- une distribution issue de Debian,
- une compatibilité avec les packages de Debian,
- un système d'installation très simple,
- une sortie tous les 6 à 8 mois,
- un environnement graphique agréable.

Cette distribution est idéale pour les étudiants, cependant la tentation est très forte de revenir au fonctionnement d'une distribution Debian, les deux étant compatibles.

b. Red Hat et Fedora



Logo Red Hat

S'il y a bien une société commerciale dans le monde Linux qui a marqué et qui continue à marquer son époque, c'est bien la société Red Hat. Fondée en 1995 par Robert Young et Marc Ewing, elle édite la célèbre distribution éponyme dont la première version officielle date de 1994 (la société a été fondée après la sortie de la distribution). Le système de package RPM est apparu avec la version 2.0. Les distributions Red Hat ont très fortement marqué les esprits car elles sont restées la référence pendant presque dix ans. Chaque version était innovante tant dans l'intégration des logiciels que dans son installateur (appelé anaconda) et ses outils de configuration.

Cependant en 2003 la version 9.0 est la dernière destinée officiellement au grand public. Les versions suivantes ont été confiées au projet communautaire **Fedora** qui continue tous les six mois à sortir une nouvelle version. Red Hat se concentre maintenant sur le monde de l'entreprise avec des distributions commerciales appelées **RHEL** (*Red Hat Enterprise Linux*) :

- des versions professionnelles destinées aux entreprises,
- des solutions du poste de travail au plus gros serveur,
- des architectures matérielles nombreuses,
- un support commercial,

- des mises à jour assurées pendant sept ans,
- 100% libre.

Vous vous doutez bien que même si l'installation d'une version RHEL AS (*Advanced Server*) est possible sur un PC de bureau elle n'a pas forcément d'intérêt pour un poste de travail ou un débutant. Bien que libre (ses sources sont intégralement disponibles librement) son coût avec le support est très élevé. Cependant si l'installation ne vous fait pas peur la distribution **CentOS** (*Community Enterprise Operating System*) est une copie exacte et téléchargeable de RHEL dont toute trace des noms et visuels Red Hat a été supprimée.



Logo Fedora

Quant au projet Fedora, il suit un cycle de développement rapide et reste destiné au grand public. Son installation est simple. Cependant l'ensemble manque un peu de cohérence (par exemple l'outil de partitionnement des disques n'est accessible que durant l'installation) ce qui en fait une distribution idéale pour tous ceux qui, amateurs éclairés, souhaitent rentrer un peu plus dans le détail.

c. Mandriva (ex-Mandrake)



Mandriva Linux (ex-Mandrake) est une distribution dérivée et longtemps entièrement compatible avec la distribution Red Hat. Elle a été créée par Gaël Duval afin d'intégrer à la distribution l'environnement de bureau graphique KDE contrairement à Red Hat qui intégrait l'environnement GNOME. Pendant plusieurs années Mandrake a été la distribution phare en forte compétition avec Red Hat. Mandrake était en effet (et est toujours) plus conviviale. Son processus d'installation est un modèle du genre et son utilisation des plus simples. Renommée Mandriva suite au rachat de la société Connectiva, la distribution est pourtant en perte d'audience depuis quelques temps. Les raisons sont multiples mais fortement liées aux aléas de la société Mandriva. Une gestion difficile suite à une mauvaise orientation dans les années 2000-2001 (le e-Learning et l'expérience américaine des Start-up) a failli une première fois conduire à sa perte et a provoqué un redressement judiciaire dont la société a réussi à sortir avec brio, pour rencontrer de nouveau quelques temps plus tard des problèmes. L'introduction sur le marché boursier n'a pas donné les résultats espérés. Souffrant d'une image trop grand public, les solutions professionnelles n'arrivent pas à s'imposer. Enfin la distribution grand public si elle reste toujours au top techniquement souffre parfois de quelques problèmes d'instabilité.

Mandriva continue cependant d'innover fortement, notamment dans le poste de travail nomade avec des distributions clé en main bootables depuis des clés USB, et c'est généralement plus par habitude et ouïe-dire qu'elle est bien souvent automatiquement conseillée aux débutants.

d. openSUSE

Se prononçant *sousse*, **openSUSE** est une distribution d'origine allemande datant de 1992. Le nom de l'entreprise lui-même était un hommage au célèbre **Konrad Zuse** l'inventeur des ordinateurs modernes. La distribution est originellement basée sur la distribution Slackware. En 1996 SuSE se rapproche d'une distribution française appelée **Jurix** créée par Florian La Roche qui est utilisée comme base à la place de Slackware. Cette même année le développement de l'outil YaST est démarré et



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

2. Les LiveCD

Le LiveCD est une catégorie surprenante. Vous êtes certainement très nombreux à vouloir essayer Linux pour voir à quoi ça ressemble ou pour vérifier s'il fonctionne correctement avec votre matériel. Plutôt que de l'installer sur votre disque dur (si cette étape vous fait peur le chapitre Le shell et les commandes GNU de cet ouvrage vous propose un guide pas à pas pour installer votre Linux) pensez d'abord à tester Linux sans l'installer. Le LiveCD sert principalement à ça : c'est une installation complète de Linux qui est fortement compressée et qui tient sur un seul cd ou dvd (dans ce cas on parle de liveDVD).

Pour utiliser un liveCD c'est très simple : insérez le CD ou le DVD dans votre lecteur et redémarrez votre ordinateur en ayant bien vérifié dans la configuration de votre machine (le setup du BIOS, voire le mode d'emploi de votre ordinateur) que votre lecteur CD ou DVD est le premier à démarrer. Après quelques secondes (ou minutes parfois) de chargement, voici que le bureau apparaît et tous les programmes les plus connus sont accessibles alors que strictement rien n'est installé sur votre disque dur. Le LiveCD le plus connu actuellement est **Knoppix**. Il est basé sur une distribution Debian, et qui plus est, s'il vous plaît un installateur est prévu pour le copier sur votre disque dur. Chaque nouvelle version de SUSE Linux arrive avec un LiveCD pour tester les dernières nouveautés sans l'installer.

E. Obtenir de l'aide

1. L'aide propre aux commandes

Il n'est pas possible de connaître par cœur tous les paramètres et arguments d'une commande. Linux propose heureusement au moins deux mécanismes pour connaître ceux qui sont supportés par une commande. La plupart du temps, le paramètre **--help** affiche l'aide incluse directement au sein du programme appelé. Parfois l'aide apportée est suffisante pour trouver ce que vous cherchez. C'est le cas avec la commande **date** dont la sortie est volontairement tronquée ici car elle prendrait deux pages.

```
$ date --help
Usage: date [OPTION]... [+FORMAT]
      ou: date [-u|--utc|--universal] [MMJJhhmm[([CC]AA) [.ss]]]
Afficher la date courante selon le FORMAT spécifié ou
initialiser la date du système.

      -d, --date=CHAÎNE          afficher la date selon la description donnée
par la CHAÎNE,
                               excluant le mot réservé « now »
      -f, --file=FICHIER         identique à --date pour chaque ligne du
                               FICHIER de dates
      -r, --reference=FILE       display the last modification time of FILE
      -R, --rfc-2822              output date and time in RFC 2822 format.
...

```

Il peut cependant arriver que l'aide soit trop concise ou manque d'explications, ou bien soit totalement absente. Dans ce cas **--help** est considéré comme un paramètre invalide et vous risquez d'obtenir un message d'erreur et/ou une ligne d'informations.

```
$ cal --help
cal: option invalide -- -
usage: cal [-13smjyV] [[mois] année]
```

La dernière ligne n'explique pas la syntaxe des paramètres.

2. L'aide interne au shell

Les commandes internes n'acceptent pas de paramètre `--help`, mais pour ces commandes l'interpréteur de commandes propose une commande **help**. Utilisée seule elle vous fournit la liste des commandes internes. Si vous lui passez comme paramètre le nom d'une commande interne, l'aide de celle-ci est affichée. C'est ainsi que vous pouvez apprendre que `pwd` admet deux paramètres optionnels.

```
$ help pwd
pwd: pwd [-LP]
      Print the current working directory. With the -P option, pwd
      prints the physical directory, without any symbolic links; the -L
      option makes pwd follow symbolic links.
```

3. Le manuel en ligne

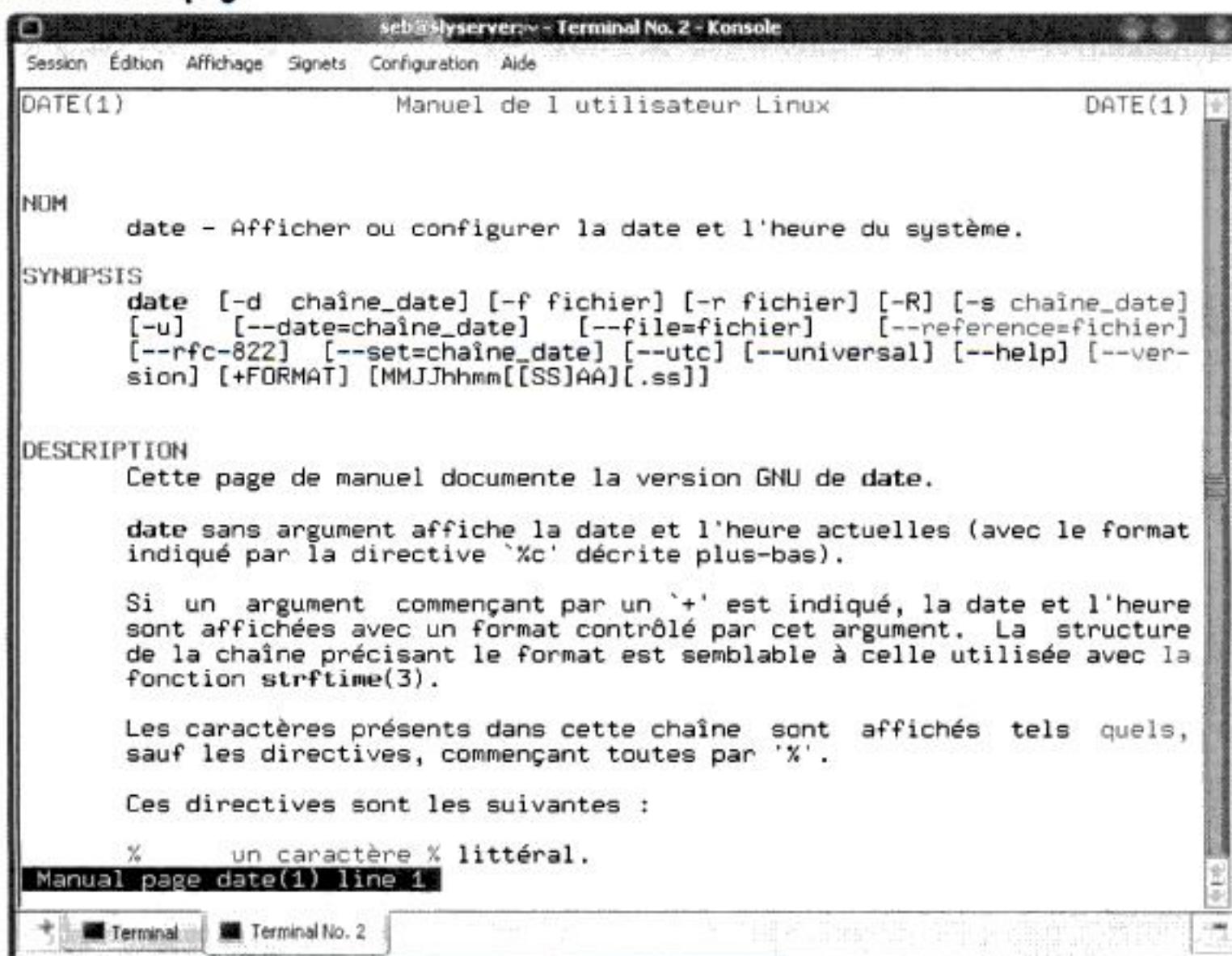
a. Accès

Quand les deux mécanismes d'aide précédents se révèlent être insuffisants, il est très probable que l'aide recherchée se situe au sein du manuel Unix. Ce manuel est standard sur tous les Unix dont Linux, et quel que soit le shell puisqu'il s'agit d'une commande externe.

Le manuel est accessible depuis la commande **man**. Vous pouvez faire un essai simple avec la commande **date** : \$ man date

Le mode d'emploi de la commande en paramètre de **man** est affiché.

b. Structure d'une page



Le manuel en ligne

Une page du manuel est composée de plusieurs sections dont celles-ci, sachant qu'elles ne sont pas forcément toutes présentes :

- **Nom** : nom et rôle de la commande.
- **Synopsis** : syntaxe générale, paramètres et arguments acceptés.
- **Description** : mode d'emploi détaillé du fonctionnement de la commande et des arguments principaux.
- **Options** : description détaillée de chaque paramètre possible, généralement sous forme de liste.
- **Exemples** : le manuel peut fournir des exemples concrets d'utilisation de la commande.
- **Environnement** : le fonctionnement de la commande peut réagir différemment si des variables du shell sont positionnées à certaines valeurs.
- **Conformité** : la commande est conforme à des recommandations ou normes (par exemple POSIX).
- **Bogues** : la commande peut parfois rencontrer des dysfonctionnements dans des cas ponctuels qui peuvent être énumérés à cet endroit.
- **Diagnostics/retour** : la commande, selon son résultat, peut retourner des codes d'erreurs significatifs dont la valeur permet de déterminer le type de problème (fichier en argument absent, etc.).
- **Voir aussi** : liste des commandes liées au programme qui peuvent intéresser l'utilisateur.

c. Navigation

Vous naviguez dans l'aide très simplement :

- La barre d'espace défile une page complète.
- La touche [Entrée] défile ligne par ligne.
- Les touches [Haut] et [Bas] défilent d'une ligne vers le haut ou vers le bas.
- Les touches [Pageup] et [Pagedown] défilent d'une demi-page vers le haut ou vers le bas.
- Les touches [Début] et [Fin] font exactement ce qu'on attend d'elles.
- La touche / permet une recherche. /toto recherche toto. Dans ce cas la touche n cherche l'occurrence suivante, tandis que [Shift] n (N) recherche la précédente.
- La touche Q quitte l'aide et revient au shell.

d. Les sections

Le manuel Linux ne fait pas que référencer les commandes classiques. C'est un manuel bien plus complet que ça. Les commandes simples, celles d'administration, les fichiers de configuration, les périphériques, les appels systèmes, les fonctions de programmation de divers langages, et bien d'autres choses encore, peuvent y être référencés. C'est pourquoi le manuel est composé de plusieurs sections distinctes.

Section	Contenu
1	Instructions exécutables ou commandes du shell
2	Appels système (API du noyau...)
3	Appels des bibliothèques (fonctions C...)
4	Fichiers spéciaux (contenu de /dev comme sd, hd, pts, etc.)
5	Format des fichiers (/etc/passwd, /etc/hosts, etc.)
6	Les jeux, économiseurs d'écran, gadgets, etc.
7	Divers, commandes non standard, ne trouvant pas place ailleurs
8	Commandes d'administration du système Linux
9	Sous-programmes du noyau (souvent vide)

Il arrive parfois que l'appel au manuel pour une commande ne retourne pas la page du manuel concernée. C'est que man recherche par défaut la première occurrence dans l'ordre des sections.

Si vous recherchez de l'aide sur le format du fichier des mots de passe, vous tomberez tout d'abord sur l'aide de la commande `passwd`. Regardez l'en-tête de la page. Le numéro de la section est indiqué juste après le nom de la commande entre parenthèses. La commande `man` a trouvé une occurrence de `passwd` dans la section 1 et affiche la page du manuel associée.

```
$ man passwd
PASSWD(1)           Manuel de l'utilisateur Linux
NOM
    passwd - mettre à jour les marques d'authentification d'un
utilisateur.
...
```

Vous pouvez demander à `man` de rechercher le manuel concerné dans une section spécifique en indiquant son numéro juste avant le nom de la commande. Pour accéder au manuel du fichier `passwd`, faites comme ceci.

```
$ man 5 passwd
PASSWD(5)           Manuel de l administrateur Linux
NOM
    passwd - Fichier des mots de passe.
...
```

e. Rechercher par correspondance

Si vous avez un doute sur la commande à utiliser, ou que vous ayez perdu son nom, ou encore que vous vouliez connaître toutes les commandes liées à un mot, alors utilisez le paramètre `-k` de `man`:

```
$ man -k passwd
/etc/rpasswd.conf (5) [rpasswd.conf] - configuration file for remote
password update client
chpasswd (8)          - change user passwords in batch
Crypt::SmbHash (3pm) - Perl-only implementation of lanman and nt md4
hash functions, for use in Samba style smbpasswd entries
fgetpwent_r (3)       - get passwd file entry reentrantly
getpwent_r (3)       - get passwd file entry reentrantly
gpasswd (1)          - change group password
ldappasswd (1)        - change the password of an LDAP entry
lppasswd (1)          - add, change, or delete digest passwords.
makepasswd (1)        - generate and/or encrypt passwords
mkpasswd (1)          - Overfeatured front end to crypt(3)
pam_localuser (8)     - require users to be listed in /etc/passwd
pam_rpasswd (8)       - PAM module to change remote password
passwd (1)            - change user password
passwd (lssl)         - compute password hashes
passwd (5)            - password file
passwd2des (3)        - RFS password encryption
rpasswd (1)           - change user password on remote server
rpasswd.conf (5)      - configuration file for remote password update
client
rpasswdd (8)          - remote password update daemon
saslpasswd2 (8)        - set a user's sasl password
smbpasswd (5)          - The Samba encrypted password file
smbpasswd (8)          - change a user's SMB password
vncpasswd (1)          - set passwords for VNC server
yppasswd (1)          - change your password in the NIS database
```

4. Rechercher de l'aide sur Internet

Comme indiqué au début du chapitre, une communauté existe autour de Linux et du logiciel libre, et les éditeurs de distributions fournissent de la documentation et du support. De ce fait, vous disposez de beaucoup de moyens pour obtenir de l'aide notamment sur Internet :

- la documentation de l'éditeur,
- les sites communautaires (FAQ, forum),
- les newsgroups,
- le projet de documentation libre (HOWTOs).
- etc.

Pensez tout d'abord à la documentation des éditeurs :

- Red Hat : <http://www.redhat.com/support>
- Debian : <http://www.debian.org/doc/>
- openSUSE : <http://en.opensuse.org/Documentation>
- Ubuntu : <https://help.ubuntu.com/>
- Mandriva : <http://club.mandriva.com/xwiki/bin/view/KB/OfficialDocumentation>
- Fedora : <http://docs.fedoraproject.org>

Sur chacun de ces sites, vous trouverez aussi très probablement :

- une base de connaissance,
- un Wiki,
- un forum,
- des rapports de bugs.

Il est impossible de lister tous les sites communautaires, mais en voici quelques-uns :

- LinuxFr : <http://linuxfr.org>
- Freshmeat : <http://freshmeat.net>
- Slashdot : <http://slashdot.org>
- Planet Libre : <http://www.planet-libre.org>
- Forum Fedora : <http://forums.fedora-fr.org/>
- Forum Mandriva : <http://forum.mandriva.com/index.php?op=Fr>
- Forum Debian : <http://forum.debian-fr.org/>
- Forum Ubuntu : <http://forum.ubuntu-fr.org>
- Forum openSUSE Alionet : <http://alionet.org>

Parmi les sites de documentation :

- Lea Linux : <http://lea-linux.org>
- The Linux Documentation Project : <http://tldp.org>
- LinuxDocs : <http://linuxdocs.org>

F. Validation des acquis : questions/réponses

Si l'état de vos connaissances sur ce chapitre vous semble suffisant, répondez aux questions ci-après.

Questions

Le système d'exploitation Unix

- 1** Quel est le seul langage directement compréhensible par un ordinateur ?
- 2** Quel est de manière synthétique le rôle d'un système d'exploitation ?
- 3** L'interface graphique fait-elle partie du système d'exploitation UNIX ?
- 4** Que signifie multitâche :
A - Un programme peut effectuer plusieurs tâches en même temps.
B - Le système sait lancer plusieurs programmes les uns après les autres.
C - L'ordinateur sait faire plusieurs choses.
D - Le système d'exploitation sait exécuter plusieurs programmes simultanément.
- 5** Dans un système multi-utilisateurs :
A - Plusieurs personnes peuvent se connecter en même temps.
B - Les programmes peuvent tourner sous des noms d'utilisateurs différents.
C - Un utilisateur doit être connecté pour lancer des programmes.
D - Plusieurs utilisateurs peuvent se connecter mais pas en même temps.
- 6** À quoi servent les appels systèmes d'un noyau Unix ?
- 7** Comment la gestion des entrées et sorties permet-elle d'accéder aux périphériques ?
- 8** Quel système d'exploitation est à l'origine de Unix ?
A - Ultrix.
B - Minix.
C - Multics.
D - Xenix.
- 9** Citez au moins deux personnes ayant contribué à la création d'Unix.
- 10** Que signifie UNIX ?
- 11** Sur quelle machine UNIX est apparu en premier ?
- 12** Pourquoi dit-on que Unix est portable ?
A - Son code source est de petite taille.
B - Le système fonctionne sur des ordinateurs portables.
C - Son code source est majoritairement écrit en langage C.
D - Unix n'est pas portable.
- 13** Que signifie BSD ?
- 14** Quelle est l'entreprise qui a la première écrit Unix ?
- 15** Quels sont les deux principaux standards Unix ?
- 16** Citez trois Unix propriétaires.
- 17** Quel a été le premier Unix à fonctionner sur PC ?
- 18** Quel est le rôle de The Open Group ?
- 19** Unix est-il standard ?



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

39 En tapant « man passwd » vous tombez sur le manuel que la commande passwd. Quelle commande permet d'accéder au manuel décrivant la syntaxe du fichier /etc/passwd ?

40 Sur quel site Internet Français pouvez-vous obtenir des informations sur l'actualité de Linux et des logiciels libres ?

Résultats

Référez-vous aux pages suivantes pour contrôler vos réponses. Pour chacune de vos bonnes réponses, comptez un point.

Nombre de points /40

Pour ce chapitre, votre score minimum doit être de 30 sur 40.

Repérez les points-clés qui vous ont posé problème et reprenez leur étude dans le chapitre avant de passer au chapitre suivant :

- Le système d'exploitation Unix.
- Le logiciel libre et Linux.
- Accès à l'aide.

Réponses

Le système d'exploitation Unix

1 Quel est le seul langage directement compréhensible par un ordinateur ?

Le langage binaire, propre à chaque microprocesseur.

2 Quel est de manière synthétique le rôle d'un système d'exploitation ?

Un système d'exploitation est l'interface entre le matériel et les logiciels.

3 L'interface graphique fait-elle partie du système d'exploitation UNIX ?

Non. L'environnement graphique n'est pas inclus dans le système. C'est un produit complémentaire fonctionnant en espace utilisateur.

4 Que signifie multitâche :

- A - Un programme peut effectuer plusieurs tâches en même temps.
- B - Le système sait lancer plusieurs programmes les uns après les autres.
- C - L'ordinateur sait faire plusieurs choses.
- D - Le système d'exploitation sait exécuter plusieurs programmes simultanément.

D.

5 Dans un système multi-utilisateurs :

- A - Plusieurs personnes peuvent se connecter en même temps.
- B - Les programmes peuvent tourner sous des noms d'utilisateurs différents.
- C - Un utilisateur doit être connecté pour lancer des programmes.
- D - Plusieurs utilisateurs peuvent se connecter mais pas en même temps.

A et B

6 À quoi servent les appels systèmes d'un noyau Unix ?

Ce sont les fonctions de programmation du noyau que les programmes utilisent pour accéder aux ressources de celui-ci et de l'ordinateur.

7 Comment la gestion des entrées et sorties permet-elle d'accéder aux périphériques ?

Les périphériques sont vus comme des fichiers, vous y accédez par des fichiers.

8 Quel système d'exploitation est à l'origine de Unix ?

- A - Ultrix.
 - B - Minix.
 - C - Multics.
 - D - Xenix.
- C

9 Citez au moins deux personnes ayant contribué à la création d'Unix.

2 parmi : Ken Thompson, Dennis Ritchie, Rudd Canaday, Brian Kernighan.

10 Que signifie UNIX ?

Uniplexed Information and Computing System.

11 Sur quelle machine UNIX est apparu en premier ?

Le DEC PDP-7.

12 Pourquoi dit-on que Unix est portable ?

- A - Son code source est de petite taille.
- B - Le système fonctionne sur des ordinateurs portables.
- C - Son code source est majoritairement écrit en langage C.
- D - Unix n'est pas portable.

C

13 Que signifie BSD ?

Berkeley Software Distribution.

14 Quelle est l'entreprise qui a la première écrit Unix ?

AT&T.

15 Quels sont les deux principaux standards Unix ?

BSD et System V.

16 Citez trois Unix propriétaires.

trois parmi : Solaris, AIX, HP-UX, True64, SCO, Qnx, etc.

17 Quel a été le premier Unix à fonctionner sur PC ?

Xenix.

18 Quel est le rôle de The Open Group ?

Il est dépositaire de la marque Unix et des recommandations de compatibilité.

19 Unix est-il standard ?

Oui, IEEE 1003 est un standard connu sous le nom de POSIX.

Le logiciel libre et Linux

20 Que signifie FSF ?

- A - Free Society for Freedom.
- B - Freeware Survey Front.
- C - Free Software Foundation.
- D - Foundation for Software to be Free.

C

21 Qui est Richard Stallman ?

Le président de la FSF et le fondateur du mouvement des logiciels libres.

22 Un logiciel libre peut-il être payant ?

Oui, seulement les sources doivent obligatoirement être diffusées gratuitement.

23 Est-il légal de graver et de donner une distribution comme Fedora à vos connaissances ?
Oui, la distribution est libre et les libertés offertes par la licence GPL vous l'autorise.

24 En quelle année le développement de Linux a-t-il commencé ?
En 1991.

25 Qui a commencé à écrire Linux ?
Linus Torvalds.

26 En quelle année est sortie la première version de Linux ?

- A - 1991
- B - 1992
- C - 1993
- D - 1994
- D*

27 Citez trois distributions Linux très connues.

Red Hat, Debian, openSUSE, Fedora, etc.

28 À partir de quel modèle de processeur Intel Linux fonctionne-t-il ?
Les modèles 80386 et au-delà.

29 Votre Quad Core 64 bits est-il supporté par Linux ?

Oui, Linux supporte les processeurs 64 bits et les multiprocesseurs.

30 À partir de quelle quantité de mémoire est-on à l'aise pour faire fonctionner Linux et son environnement graphique de manière optimale ?

256 Mo peuvent suffire, il est préférable pour l'utilisation de KDE ou Gnome de disposer d'au moins 512 Mo.

31 Comment savoir si votre matériel est compatible avec Linux ?

En vous rendant sur les listes de compatibilité de l'éditeur de la distribution, et/ou sur les divers sites communautaires, ou encore sur le site du constructeur.

32 Votre imprimante est une Lexmark 7500. Où savoir si un pilote existe pour Linux ?

Sur le site <http://www.linuxprinting.org>.

33 Quelle est la principale différence entre une distribution Debian et une distribution Red Hat ?

La distribution Debian est libre et communautaire, tandis que la Red Hat, bien que libre, est commerciale.

34 En tant que très grand débutant, quelles distributions vous conviendraient le mieux ?

- A - Fedora
- B - Ubuntu
- C - openSUSE
- D - Debian

B et C, car Debian est trop complexe pour un débutant et Fedora plairait plus aux amateurs éclairés.

35 Pourquoi, pour une entreprise, pourrait-il être plus sécurisant d'acheter des licences de distributions commerciales comme Red Hat ou Novell ?

Pour le support, le suivi des correctifs de bugs et de sécurité.

36 Comment essayer Linux sans l'installer ?

Avec un liveCD.

Accès à l'aide

37 La syntaxe de la commande **who** vous échappe. Pour obtenir de l'aide depuis cette commande vous devez taper :

- A - who --help
- B - who -h
- C - who help
- D - man who

A

38 Vous ne vous rappelez plus du nom exact d'une commande, mais seulement qu'elle contient « **passwd** » dans son nom. Pour obtenir la liste de l'ensemble des pages du manuels correspondant à ce mot, quelle commande devez-vous taper ?

man -k passwd

39 En tapant « **man passwd** » vous tombez sur le manuel que la commande **passwd**. Quelle commande permet d'accéder au manuel décrivant la syntaxe du fichier **/etc/passwd** ?

Le manuel est en section 5 : man 5 passwd.

40 Sur quel site Internet Français pouvez-vous obtenir des informations sur l'actualité de Linux et des logiciels libres ?

http://linuxfr.org.

G. Travaux pratiques

1. Histoire d'Unix et de Linux

1. Rendez-vous sur le site de Eric Levenez : <http://www.levenez.com/>. Eric Levenez tient à jour l'histoire de tous les systèmes Unix depuis leur origine. À l'aide des documents « Unix History » déterminez :

- a - La date exacte de Unix V1.
- b - La date exacte de 1BSD.
- c - La date exacte de Linux 1.0.

Réponses :

- a - Unix V1 est sorti le 3 novembre 1971.
- b - 1BSD est sorti le 9 mars 1978.
- c - Linux 1.0 est sorti le 14 mars 1994.

2. Rendez-vous sur le site de The Open Group. Tentez de trouver le documents décrivant le dernier standard Shell et Utilitaires.

- a - Quelles sont ses références ?
- b - Pouvez-vous y accéder gratuitement ?

c - D'après ce document, quels sont les seuls paramètres obligatoires de la commande **du** pour un Unix POSIX ?

Réponses :

a - Il porte la référence C048, se nomme « **Shell and Utilities, Issue 6, 2004 Edition [XCU]** », accessible depuis <http://www.opengroup.org/bookstore/catalog/c048.htm>. C'est un véritable standard IEEE Std 1003.1, 2004 Edition.

b - Vous pouvez accéder gratuitement à la version HTML en ligne du document. Rendez-vous sur <http://www.opengroup.org/onlinepubs/009695399/toc.htm> pour lire ce document. Il vous sera cependant demandé vos coordonnées (nom, prénom et mail).

c - D'après ce document, seules les options **-a**, **-H**, **-k**, **-L**, **-s** et **-x** sont obligatoires pour tout Unix Posix.

2. Distributions

1. Le site Distrowatch <http://distrowatch.com/> fournit des informations complètes sur toutes les distributions Linux connues et des statistiques. Rendez-vous sur ce site et répondez aux questions suivantes :

- a - Quelle est la spécificité de la distribution Gentoo ?
- b - Un débutant peut-il installer la distribution LFS ?
- c - Citez à la date de lecture, les cinq distributions en tête des statistiques.

Réponses :

a - Saisissez Gentoo dans le champ en haut à gauche. Le descriptif vous informe que Gentoo dispose d'un outil de supervision particulier appelé Portage. Si vous fouillez plus vous apprendrez que Gentoo accepte la compilation à la volée des packages pour s'adapter parfaitement à votre OS.

b - LFS n'est pas destinée aux débutants. Ce n'est pas une distribution mais une méthode pour créer votre propre distribution. Vous devez suivre un guide pour créer votre installation Linux depuis zéro (Linux From Scratch).

c - Au 26/06/2008, les distributions les plus en vue sont dans l'ordre Ubuntu, openSUSE, Fedora, Mint et PCLinuxOS. Cette liste est accessible à droite sur les statistiques des pages.

2. Rendez-vous sur le site officiel de openSUSE <http://www.opensuse.org/> et répondez aux questions suivantes.

- a - Quels sont les supports et méthodes possibles pour télécharger openSUSE ?
- b - La carte Creative Audigy2 est-elle supportée par la distribution openSUSE ?
- c - Qu'est-ce que Compiz Fusion ?

Réponses :

a - Cliquez sur « Get It » sur la page d'accueil. openSUSE peut être téléchargée sous forme de LiveCD, DVD ou d'installation via le réseau, pour les architectures 32 bits, 64 bits et PowerPC. Le téléchargement peut se faire via BitTorrent ou en transfert HTTP/FTP classique. Le tout est gratuit.

b - Dans le wiki du site tapez Audigy. Deux liens (au 26/06) vous sont proposés. Le mot HCL signifiant Hardware Compatibility List, cliquez sur le lien correspondant. Vous apprendrez ainsi que toutes les cartes son Audigy sont supportées.

c - Compiz Fusion (http://en.opensuse.org/Compiz_fusion) est un gestionnaire de fenêtres composite permettant d'obtenir de nombreuses fonctionnalités avancées pour le bureau et des effets 3D parfois aussi beaux qu'inutiles. Il peut être installé en quelques clics sous Linux.

3. Aide et documentation

1. On vous a parlé de la commande **dialog** mais vous ne comprenez pas son fonctionnement. Le but est d'arriver à exploiter celle-ci grâce à son aide.

a - Accédez à l'aide interne de dialog, notamment sur l'affichage d'un calendrier. Créez un calendrier du mois de juin 2008 avec un texte sur une ligne « choisissez la date » et une largeur automatique.

b - Le titre de l'écran est « choix de la date » et le titre de la boîte, « calendrier ». Mais l'aide interne n'est pas claire. Accédez au manuel de dialog. Quels sont les deux paramètres à passer à dialog ? Affichez la nouvelle boîte de dialogue.

c - Effectuez une recherche globale de dialog dans le manuel. Si vous trouvez la commande **Xdialog**, accédez à son aide. Quelle est la différence avec dialog ? Si vous le pouvez, affichez le même calendrier mais avec Xdialog.

Réponses :

a - L'option `--calendar` affiche un calendrier. Le premier paramètre est le texte « Sélectionnez la date ». Les deux suivants sont la hauteur et la largeur. 0 indique une taille automatique. Spécifiez ensuite la date au format « jour mois année ».

Le titre de la fenêtre est `--backtitle` tandis que le titre de la boîte est `--title`.

La commande est donc :

```
$ dialog --backtitle "Choix de la date" --title "Calendrier" --  
calendar "Sélectionnez la date" 0 0 01 06 2008
```

b - Accédez à l'aide avec « `man dialog` ». Dans les options communes (Common Options), le paramètre `--backtitle` permet de donner un titre au fond de l'écran, en haut à gauche. Le paramètre `--title` quant à lui donne un titre à la boîte de dialogue.

Quittez l'aide avec la touche Q. Tapez la commande suivante complète :

```
$ dialog --backtitle "Choix de la date" --title "Calendrier" --  
calendar "Sélectionnez la date" 0 0 01 06 2008
```

c - Faites un :

```
$ man -k dialog
```

Regardez les résultats en section 1. Si Xdialog est installé, son nom doit s'afficher. Accédez à l'aide de Xdialog :

```
$ man Xdialog
```

Vous remarquez que cette commande est quasi identique à `dialog`. Si vous êtes sous X, remplacez, dans le point précédent, `dialog` par `Xdialog` et validez :

```
$ Xdialog --backtitle "Choix de la date" --title "Calendrier" --  
calendar "Sélectionnez la date" 0 0 01 06 2008
```

2. Pour cette seconde partie, rendez-vous sur le site du projet de documentation Linux, « The Linux Documentation Project », <http://tldp.org/>. Ce site propose diverses documentations : FAQ, HOWTO, guides (de vrais livres), pages du manuel, le tout traduit ou non.

a - Un HOWTO est un document décrivant comment faire telle ou telle action, pas à pas. Allez voir le HOWTO en français sur le problème du Signal 11.

b - Dans ce HOWTO trouvez la cause principale de l'origine de ce message.

c - Comme dans ce document, vous soupçonnez une défaillance d'une barrette mémoire. Que vous conseille ce HOWTO ?

Réponses :

a - Cliquez à gauche sur Translated Howtos puis sur fr dans la liste du milieu. Choisissez le format souhaité : Texte plein (une grande page au format texte), HTML plein, HTML par chapitres ou PDF. Préférez pour le moment le format HTML sur une page. Choisissez ensuite dans la liste le sig11-HOWTO :

<http://www.ibiblio.org/pub/Linux/docs/HOWTO/translations/fr/html-1page/Sig11.html>.

b - Dès le début le HOWTO vous informe que la cause principale d'un plantage provoquant un message de type Signal 11 est un problème matériel.

c - Le HOWTO conseille d'effectuer divers tests, mais pour la mémoire, il vous parle d'un test appelé memtest86. Comme le HOWTO est ancien vous vous rendrez compte qu'il vaut mieux utiliser memtest+.

Chapitre 2

A. Installer une Debian	66
B. Installation de openSUSE	74
C. Red Hat Package Manager	87
D. YUM	91
E. Debian Package	95
F. Gestionnaire APT	100
G. Installer depuis les sources	106
H. Gérer les bibliothèques partagées	117
I. Validation des acquis : questions/réponses	121
J. Travaux pratiques	129

Pré-requis

- ☒ Avoir configuré le BIOS pour démarrer sur le lecteur CD.
- ☒ Disposer d'une source d'installation (CD, DVD, réseau).
- ☒ Savoir manipuler une console ou un terminal de base.

Objectifs

À la fin de ce chapitre, vous serez en mesure :

- ☒ D'installer une distribution Debian.
- ☒ D'installer une distribution openSUSE.
- ☒ De gérer le Red Hat Package Manager.
- ☒ De gérer les packages Debian.
- ☒ De gérer les gestionnaires APT et affiliés.
- ☒ D'installer un produit depuis les sources.
- ☒ De gérer les bibliothèques partagées.

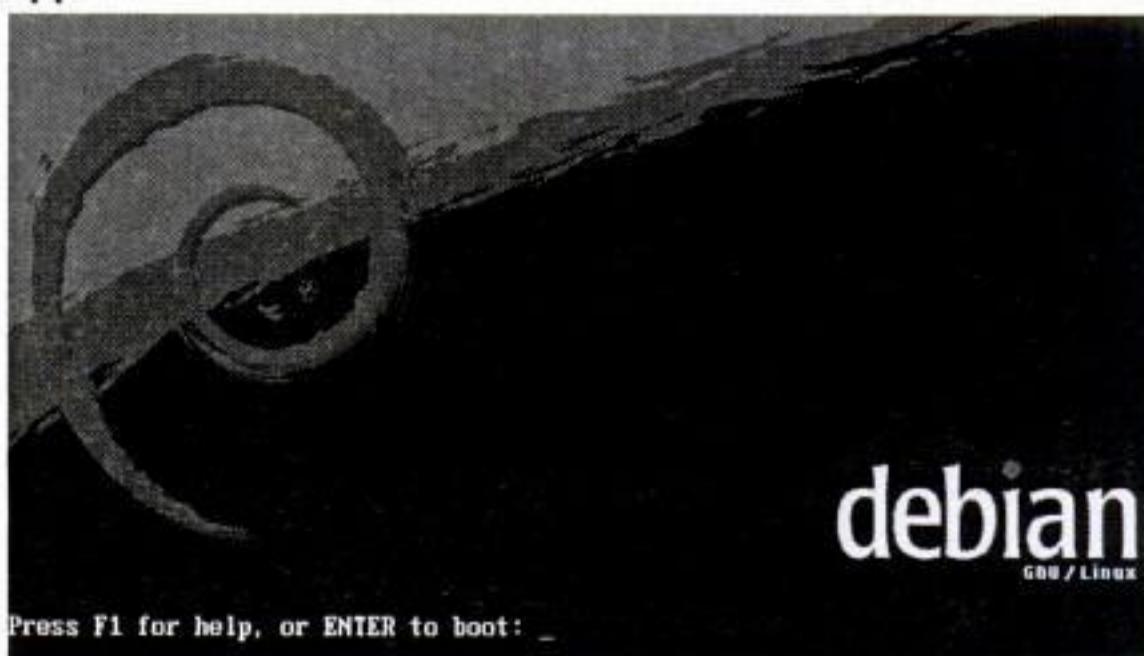
A. Installer une Debian

1. Support d'installation

Voici comment, étape par étape, installer une distribution Debian. Il s'agit de la dernière version stable, soit la 4.0r3 Etch, à l'écriture de ce livre. L'installation est effectuée en mode texte, ce qui est la meilleure méthode pour Debian (il existe un installateur graphique peu utilisé). Si vous souhaitez effectuer la même installation, vous pouvez récupérer l'image ISO correspondant à une installation via le réseau (Internet), appelée netinst et accessible via le site de Debian : http://cdimage.debian.org/debian-cd/4.0_r3/i386/iso-cd/debian-40r3-i386-netinst.iso

Gravez cette image comme CD ou clé USB. Pour les besoins de ce livre, Debian a été installée dans une machine virtuelle VMWare. VMWare Server est un produit qui n'est pas libre mais qui est gratuit.

2. Boot sur le support



Boot du support Debian

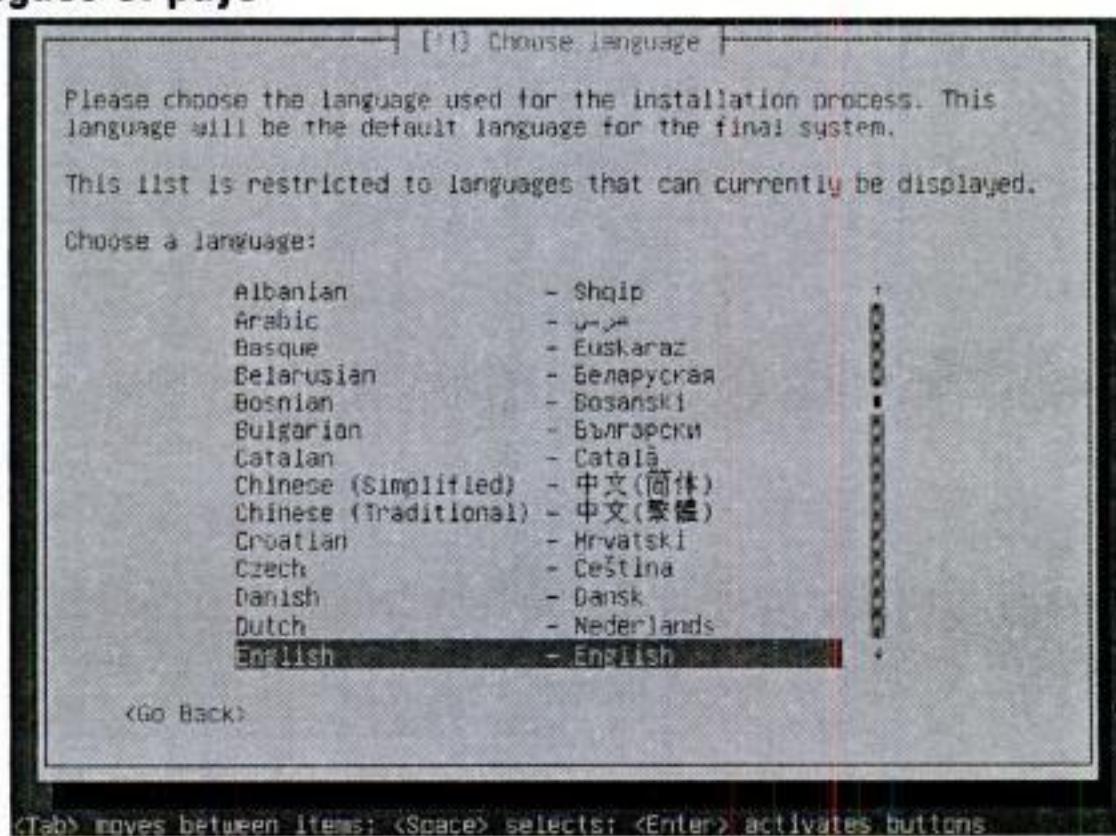
Configurez votre ordinateur pour qu'il démarre sur le support d'installation. Au moment du boot, vous avez accès à une ligne de commande permettant de lancer l'installation en appuyant sur [Entrée], ou les touches de [F1] à [F10] pour accéder à des écrans d'aide supplémentaires. Vous pouvez en effet passer des options en ligne de commande selon votre machine (de [F5] à [F10]) car dans certains cas ponctuels il peut être nécessaire de modifier des valeurs passées au noyau Linux pour un bon fonctionnement.



Options spéciales au boot

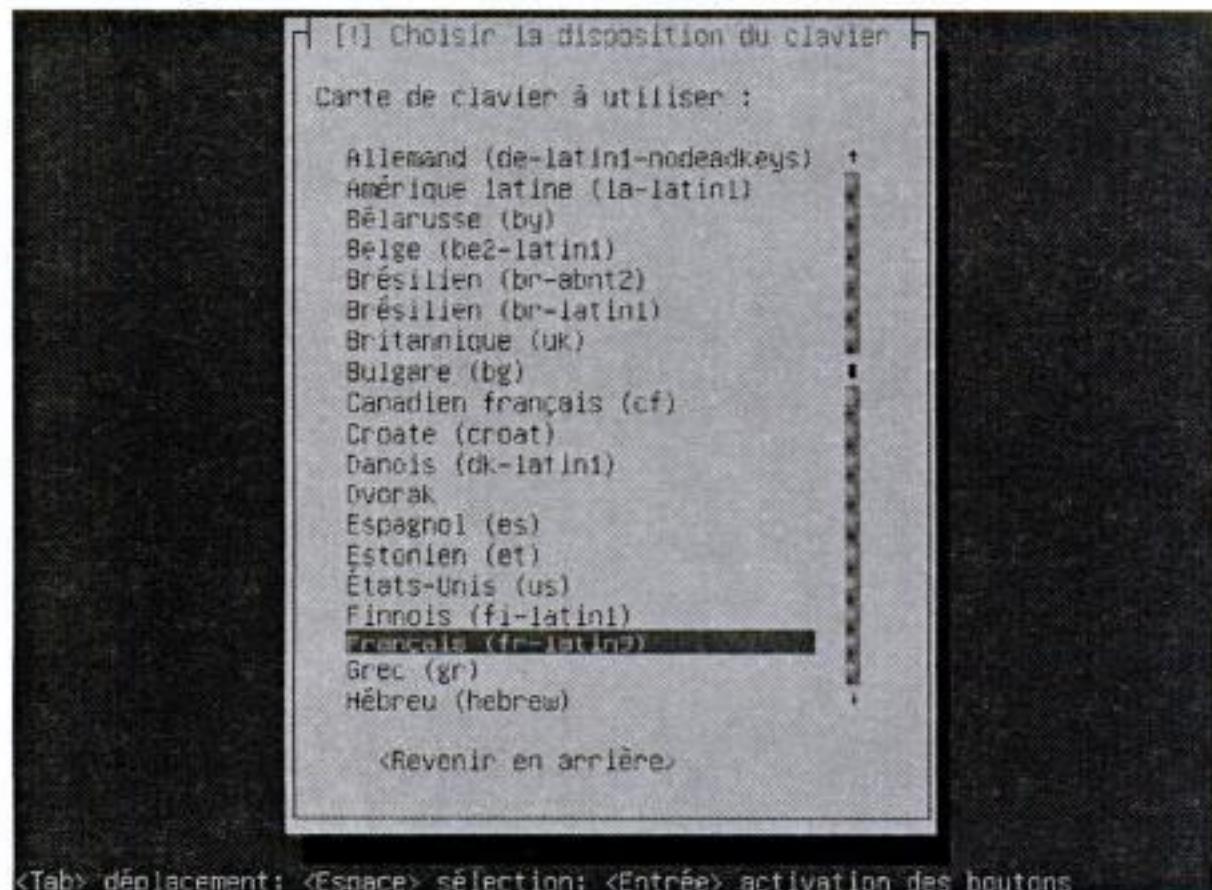
Appuyez sur la touche [Entrée] pour lancer l'installation. L'installateur lui-même fonctionne sous Linux, vous voyez défiler toutes les lignes du démarrage du noyau Linux. Enfin le premier écran de l'installateur est affiché.

3. Choix des langues et pays



Trois écrans vous permettent de choisir :

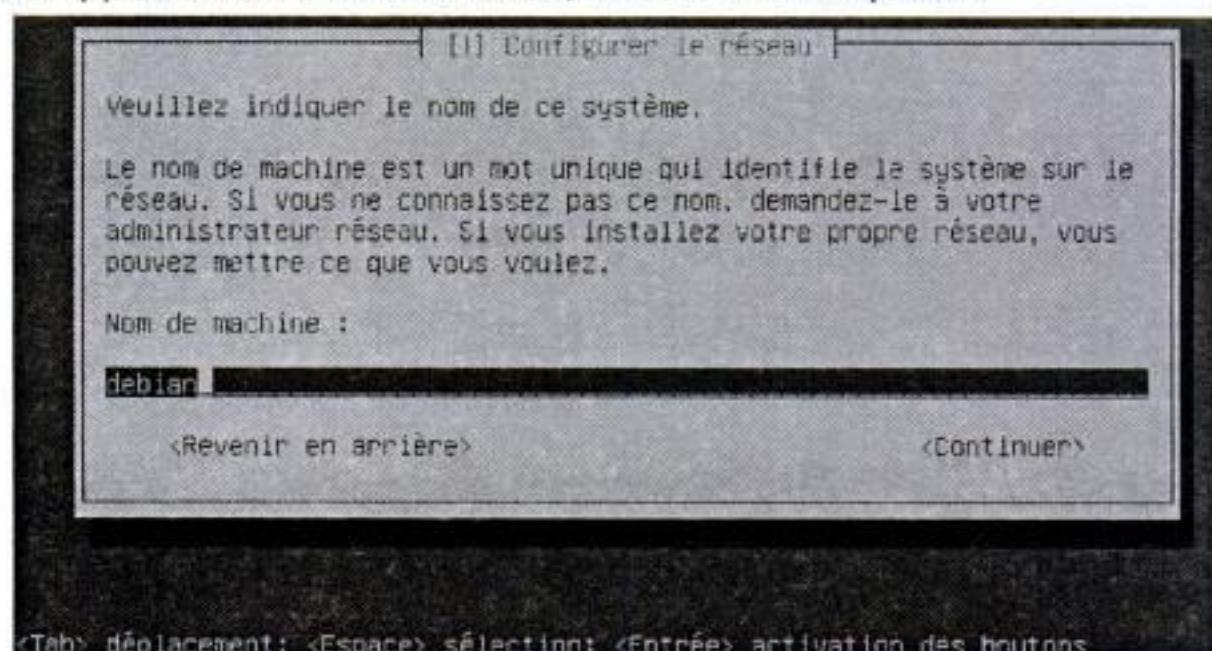
- La langue utilisée par le processus d'installation. Naviguez avec les flèches, appuyez sur [Entrée] pour continuer. Dans la suite, c'est le français qui est utilisé.
- Selon la langue initiale choisie, Debian vous demande ensuite dans quel pays vous vous situez. C'est utile car c'est ainsi que sont positionnées les variables locales : format de date, d'heure, encodage des caractères, formats numériques et monétaires, etc.
- Enfin, choisissez votre type de clavier. Pour la France, c'est fr-latin9.



4. Paramètres du réseau

Les trois étapes suivantes concernent les informations réseaux de base. Si l'installateur n'a pas réussi à configurer la carte réseau par DHCP, il vous demandera de saisir les informations de base : adresse IP, masque de sous réseau, passerelle par défaut, DNS.

Puis vous devez saisir un nom d'hôte (le nom de la machine sur le réseau) et le nom du domaine. Si votre machine n'appartient à aucun domaine, laissez le champ vide.



Nom d'hôte de la machine

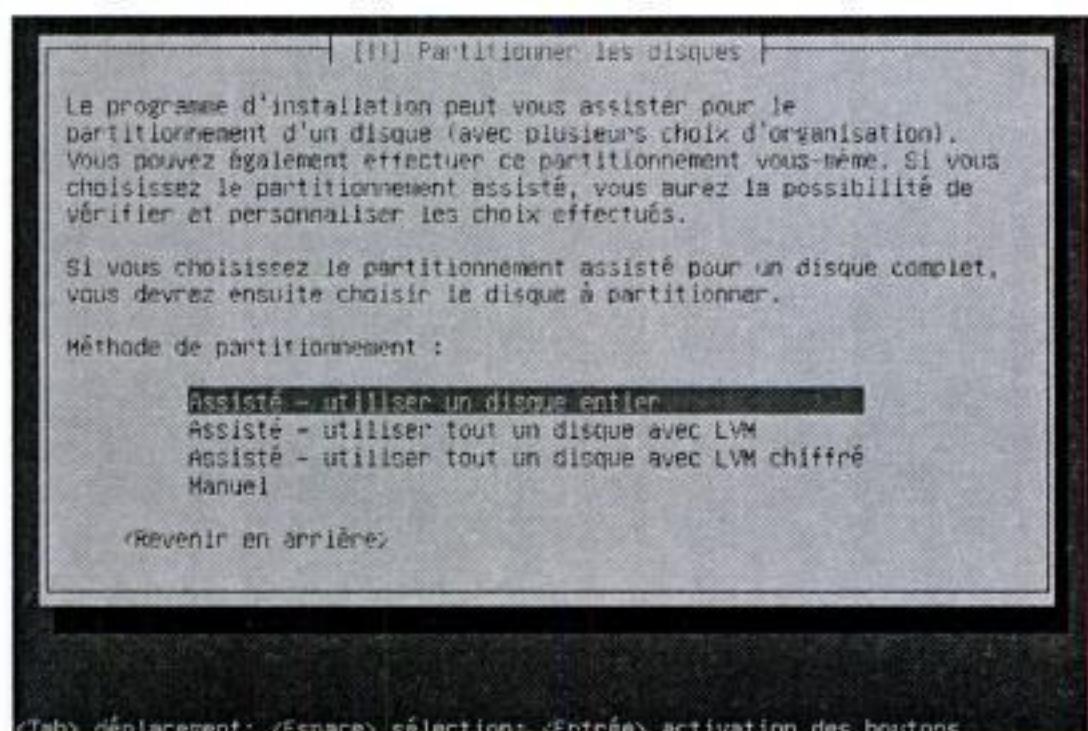
5. Partitionner les disques

De manière simpliste, vous avez le choix entre trois principales méthodes pour partitionner vos disques :

- Une méthode assistée (voire automatique) en utilisant le partitionnement classique (voir à ce propos le chapitre Les disques et le système de fichiers).
- Une méthode assistée proposant le LVM (*Logical Volume Manager*).
- Une méthode manuelle.

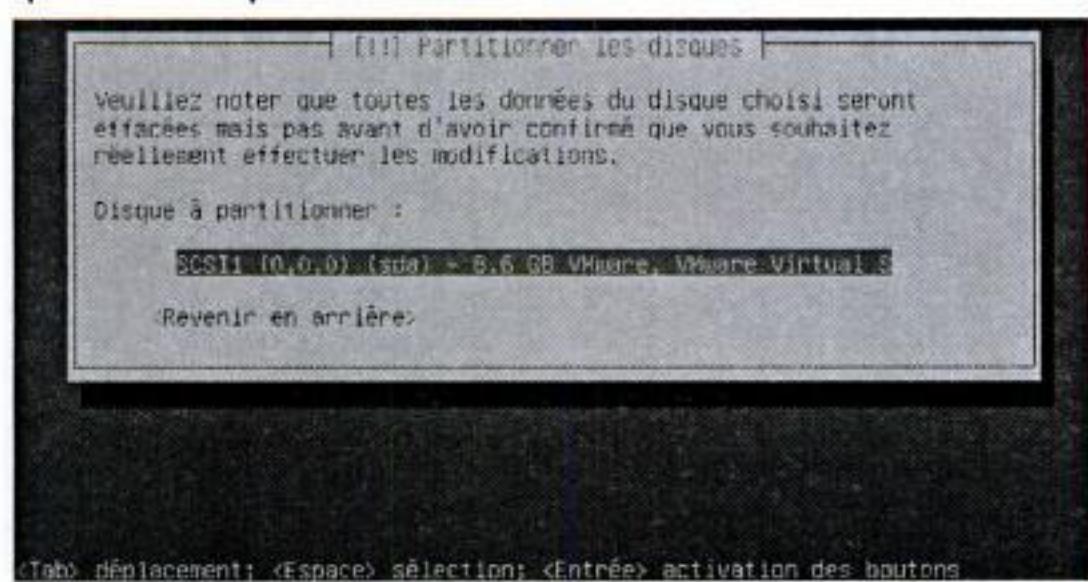
La méthode assistée classique dans le cas d'une nouvelle installation donne des bons résultats. Si vous réinstallez une machine, ou que vous installez Debian sur une machine disposant déjà de partitions contenant les dossiers personnels par exemple, passez par un partitionnement personnalisé. Le LVM consiste à regrouper des disques physiques ou partitions (appelés volumes physiques) en un seul grand espace (appelé groupe de volumes) dans lequel vous pouvez découper des espaces logiques à volonté (appelés volumes logiques), les agrandir, les réduire, etc.

Cependant vous devriez envisager la solution LVM dans le cadre d'un serveur d'entreprise ou si vous pensez rajouter à terme des disques dans votre machine pour rajouter de l'espace de stockage. Le LVM apporte une très grande souplesse.



Étape de partitionnement

Si vous avez choisi la première méthode, vous accédez à l'écran suivant. Pour les besoins du livre, un espace de 8 Go (environ) a été créé sous VMWare comme premier disque SCSI. C'est celui-ci qui va servir pour l'installation.

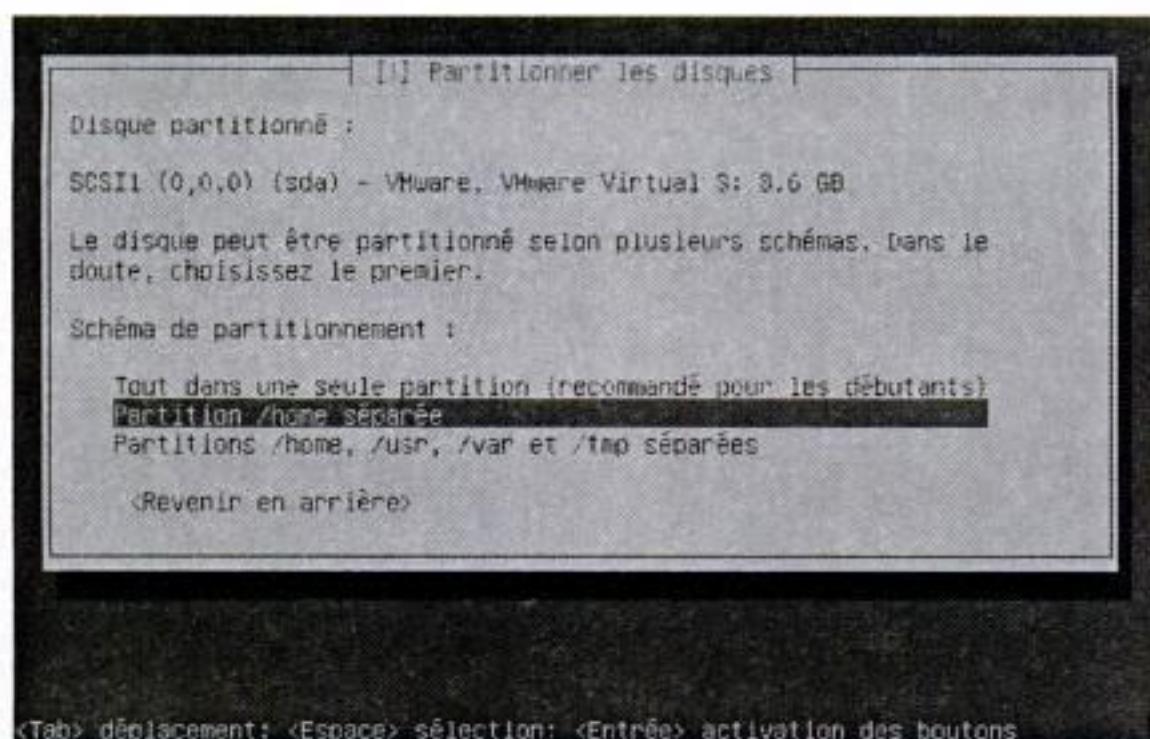


Choix du disque

L'étape suivante consiste à choisir le schéma de partitionnement :

- Soit une seule grosse partition dans laquelle vous mettez tout (système, programmes, données). Tout est mis dans la partition racine /.
- Soit deux partitions : une partition racine qui contiendra le système et tous ses composants (programmes, paramètres systèmes, etc.), et une partition qui va contenir les données des utilisateurs. Pour un poste de travail ou un PC personnel (à la maison), c'est la méthode la plus pertinente : elle permet de réinstaller facilement un autre système (mise à jour ou réinstallation complète) sans casser les données personnelles : la nouvelle distribution ainsi installée pourra réutiliser la partition montée sur /home et ainsi récupérer les données.
- La troisième méthode propose de créer cinq partitions différentes : la racine /, les données personnelles /home, les composants utilisateurs (programmes, bibliothèques, données partagées associées, etc.) /usr, le contenu variable /var et les fichiers temporaires /tmp. Ce choix est tout à fait pertinent sur un serveur. Les mails, informations DHCP, sites web, etc., sont souvent stockés dans /var. Les mises à jour des divers programmes (services) sont dans /usr, etc. Ce schéma de partitionnement est quasi-parfait : tout est indépendant. Il devient bien plus simple ensuite de changer de disque, de migrer les données, d'étendre la taille des volumes, etc., sans casser le reste.

Le meilleur choix pour l'installation de test est le second, la suite des opérations se base sur celui-ci.



Choix du schéma de partitionnement

S'agissant d'une méthode semi-automatisée, Debian vous indique ses choix pour chacune des partitions. Vous constatez la présence de trois partitions au lieu des deux proposées. Debian a analysé la taille mémoire de la machine et propose la création d'une zone de swap correspondant au meilleur choix possible.

Ne soyez pas non plus surpris par les numéros de partitions. Très souvent la seule partition primaire est la racine / tandis que toutes les autres sont des partitions logiques au sein d'une partition étendue, et Linux numérote les primaires de 1 à 4 (une partition étendue est une partition primaire) tandis que la numérotation des partitions logiques débute à 5. Pour une meilleure compréhension, veuillez vous reporter au chapitre Les disques et le système de fichiers.

Si ce schéma de partitionnement vous convient, validez. L'écran suivant vous donne un récapitulatif que vous devez de nouveau valider.

Attention ! Le partitionnement est suivi de l'écriture des systèmes de fichiers sur les partitions concernées. Cette opération est identique au formatage sous Windows. Cette opération est destructive.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



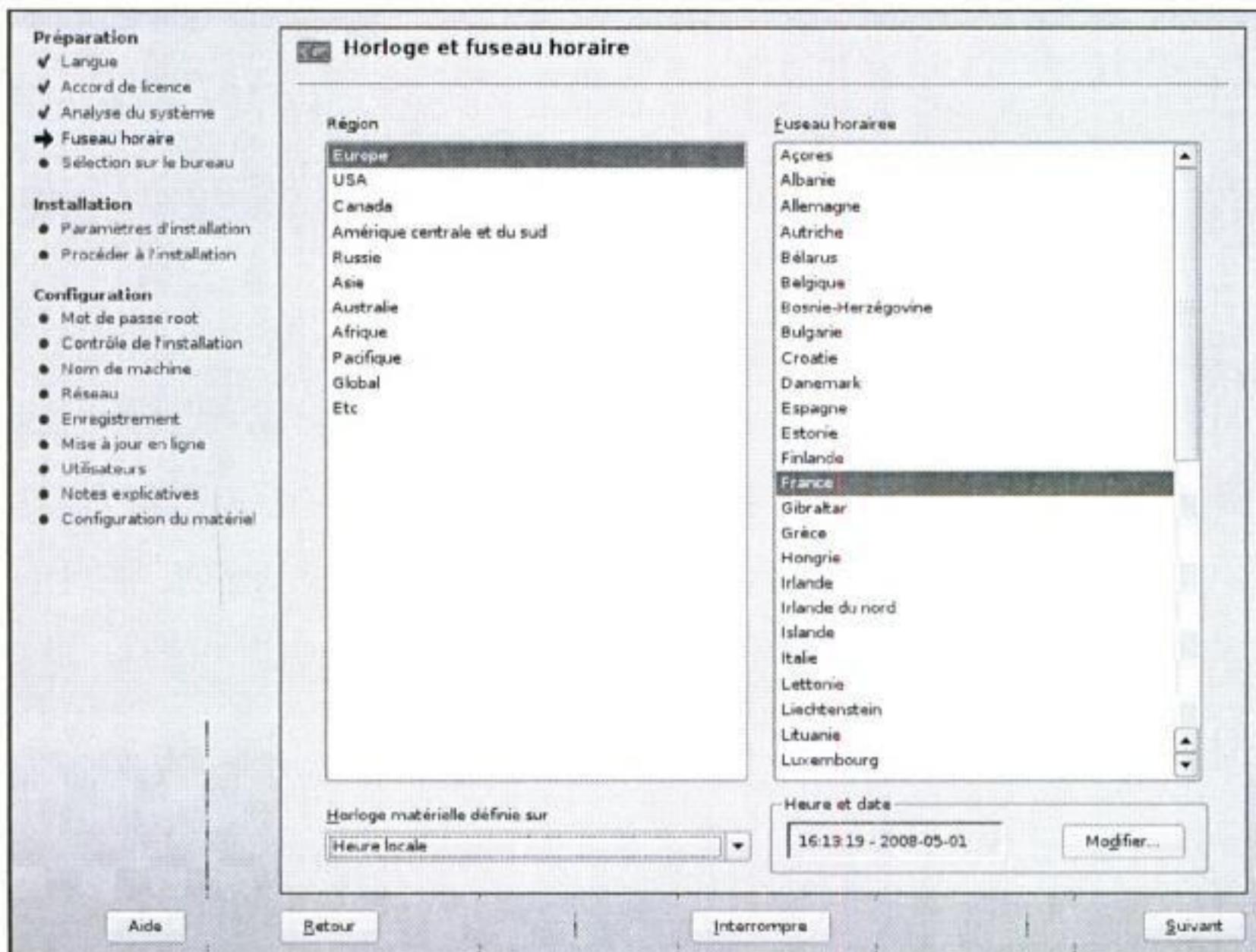
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



Cliquez sur **Suivant**.

4. Type de bureau

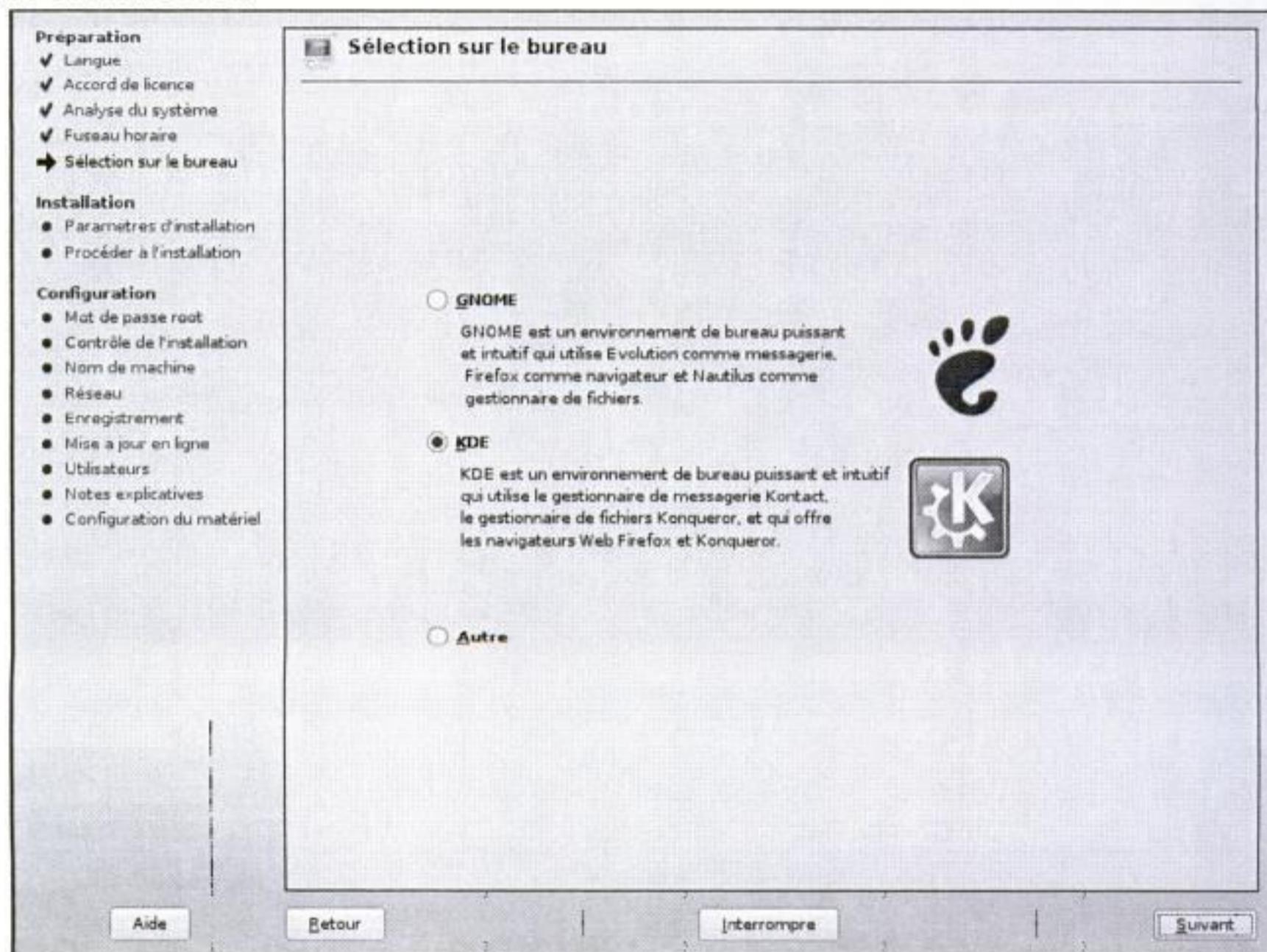
openSUSE n'est pas une distribution axée sur un type d'environnement bureautique par défaut (comme Debian et Ubuntu sous Gnome, Mandriva sous KDE, etc.) et supporte les deux environnements (voire plus avec XFCE) sur un pied d'égalité. Vous pouvez installer les deux. Cependant sur cette étape vous devez choisir l'environnement de bureau principal, celui qui sera utilisé par défaut lors des connexions.

Ce choix n'est pas définitif : dans les étapes suivantes vous pourrez choisir un autre environnement, et sur l'écran du gestionnaire de sessions vous pourrez aussi changer d'environnement (s'il est installé) à la volée.

Les deux environnements proposés par défaut sont Gnome et KDE :

- Gnome est l'environnement de bureau original du projet GNU basé sur les bibliothèques GTK et très évolué. Ses dernières versions ont largement compensé leur retard sur d'autres projets. Il est très simple, beaucoup d'options complexes étant masquées, laissant seuls paraître les choix évidents, permettant ainsi au débutant de ne pas se perdre. Il dispose d'outils puissants : le gestionnaire de fichiers Nautilus, la messagerie Evolution, le navigateur Firefox sont basés sur les mêmes widgets et donc parfaitement intégrés (tout au moins visuellement) à Gnome.
- KDE est historiquement le premier vrai environnement bureautique fonctionnel sous Linux. Il a évolué très vite et ses versions actuelles sont très intuitives et puissantes, proposant des fonctionnalités qu'on ne retrouve pas sur d'autres environnements, y compris sur d'autres OS propriétaires. Si vous venez de Windows et que vous êtes habitué à modifier votre environnement, KDE est pour vous.

Vous pouvez faire fonctionner des applications de n'importe quel environnement dans n'importe quel autre environnement.



Gnome ou KDE ?

Sélectionnez votre environnement, par exemple **KDE**, et cliquez sur **Suivant**.

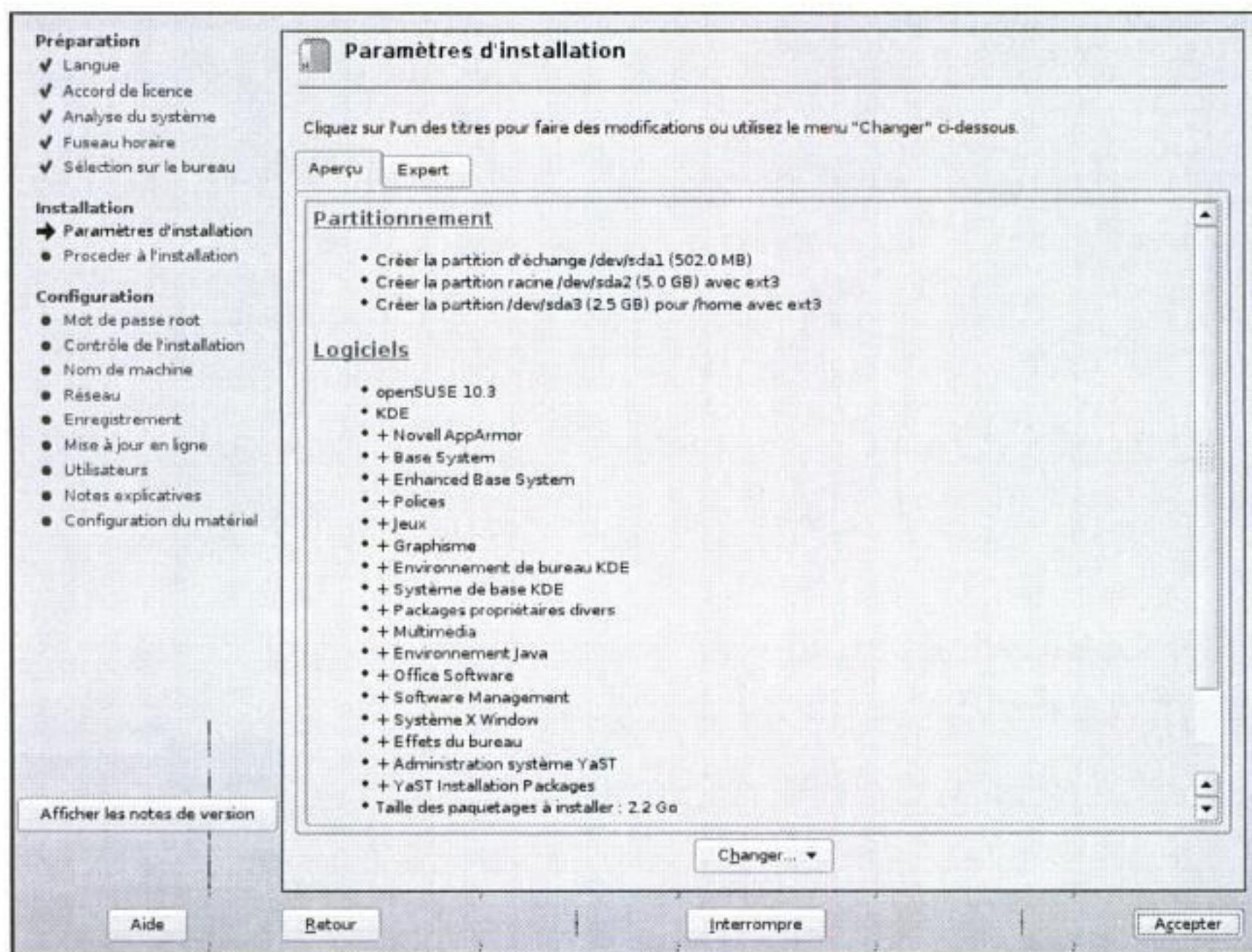
5. Paramètres d'installation

L'écran des paramètres d'installation est le plus important. C'est là notamment que vous avez accès à l'onglet du mode **Expert**, qui n'a d'expert que le nom puisqu'il étend simplement le nombre d'options proposées (bootloader, etc.). Restez sur l'onglet **Aperçu**.

Dans ce mode, vous disposez d'informations sur les deux plus importants choix d'installation qui sont le partitionnement des disques et le choix des packages logiciels. Le partitionnement a déjà été présenté lors de l'installation Debian. La logique de openSUSE est la même : il tente de déterminer les meilleurs choix possibles pour vous et celui-ci est généralement la création de trois partitions : la racine « / », les données personnelles « /home » et un espace de swap. C'est exactement la même chose que l'un des choix proposés par Debian et si vous comparez seule la taille varie, chacune des distributions disposant de ses propres méthodes de calcul.

Le choix des logiciels dépend en partie du choix du bureau. Comme vous avez sélectionné KDE, vous remarquez que les choix par défaut sont prédéfinis : environnement de bureau KDE, système de base KDE, etc.

Vous pouvez modifier ces choix, bien évidemment. En cliquant sur les titres des sections, vous accédez aux outils associés : partitionnement et gestion des logiciels.



6. Partitionnement

L'outil de partitionnement de openSUSE basé sur YaST est d'une très grande simplicité, pourtant appelé partitionnement en mode expert. Le cadre central vous donne la liste des disques (/dev/hdX, /dev/sdX) puis en dessous la liste des partitions associées (/dev/hdxy, /dev/sdxy), la taille, si elle sera formatée (F), le type, le point de montage, etc. Vous pouvez créer, modifier, supprimer et redimensionner les partitions. Le même outil permet de créer des volumes RAID, LVM ou même les deux en même temps, de créer des partitions chiffrées ou des points de montage NFS.

Il est possible de redimensionner tout type de partition, y compris les partitions NTFS. Dans ce cas, il est préférable de vous rendre au préalable sous Windows et de faire une défragmentation et une analyse complète du disque. Pour tout redimensionnement, cela implique qu'il reste de la place sur la partition pour déplacer les blocs correspondants.



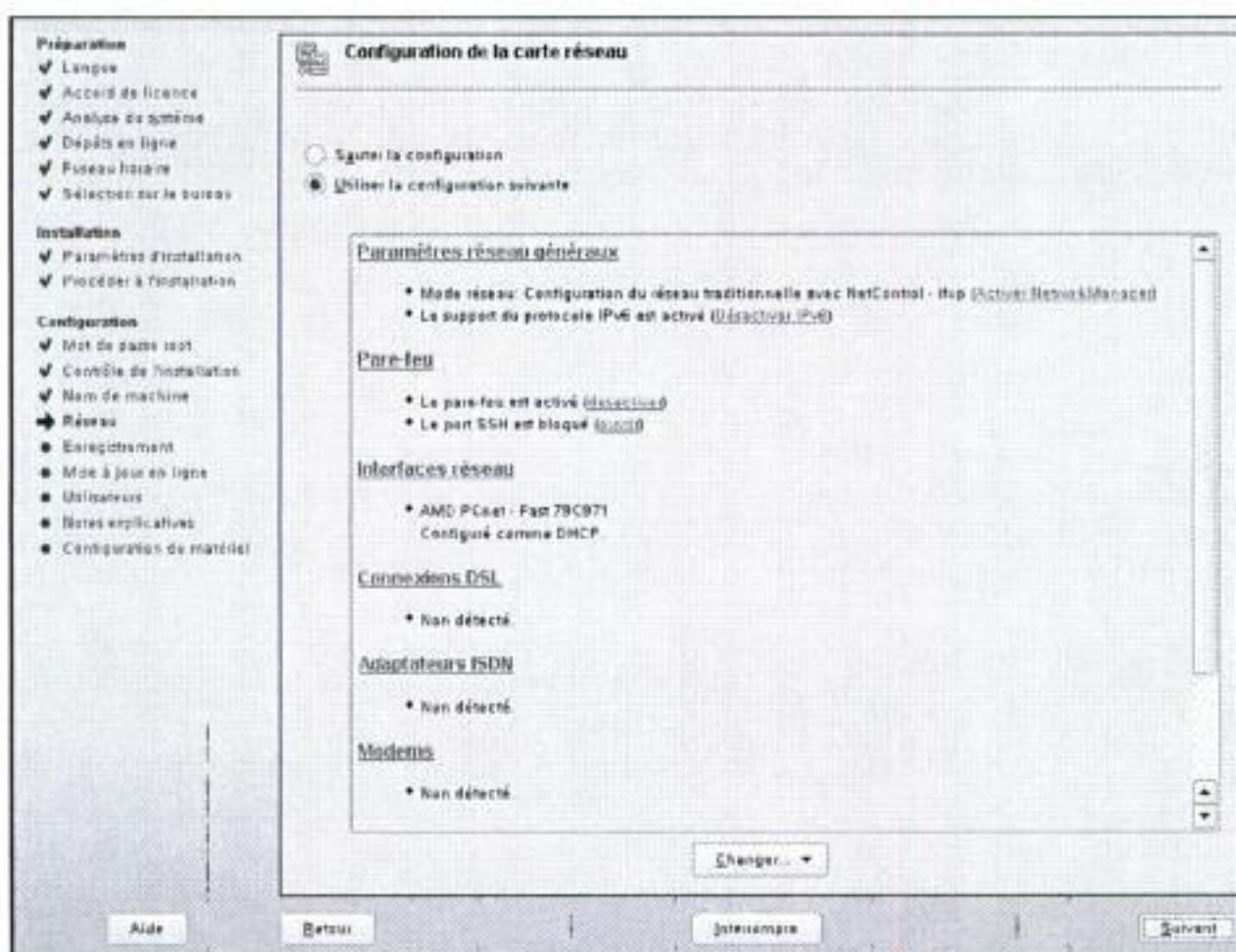
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

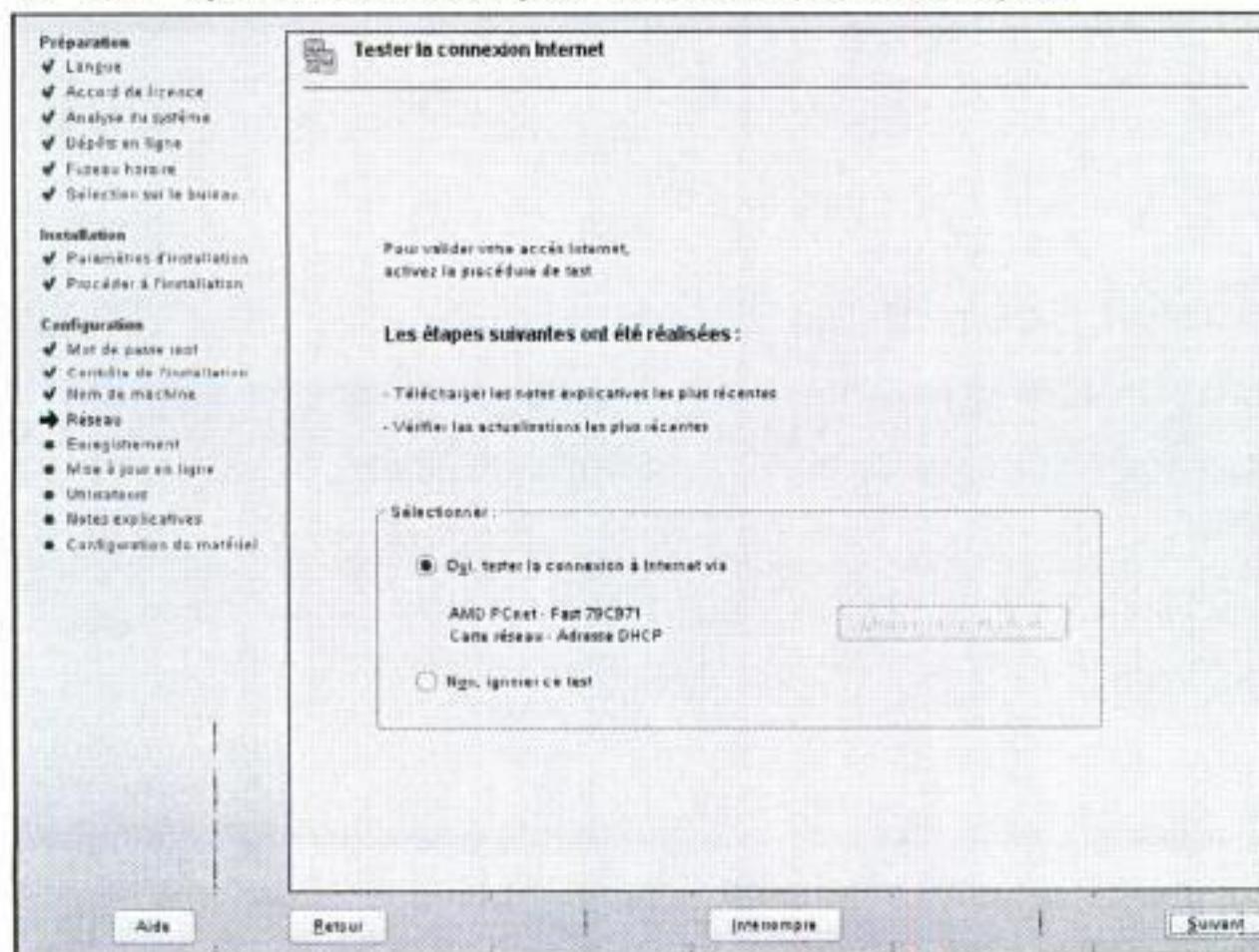


You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



Cliquez sur **Suivant** quand vos interfaces sont configurées. Le réseau est démarré. L'étape suivante consiste à tester votre connexion à Internet, étape presque indispensable notamment pour pouvoir accéder aux mises à jour. Si vous avez configuré plusieurs interfaces réseaux, vous pouvez sélectionner celles vous permettant d'accéder à Internet. Vous pouvez aussi ignorer le test.

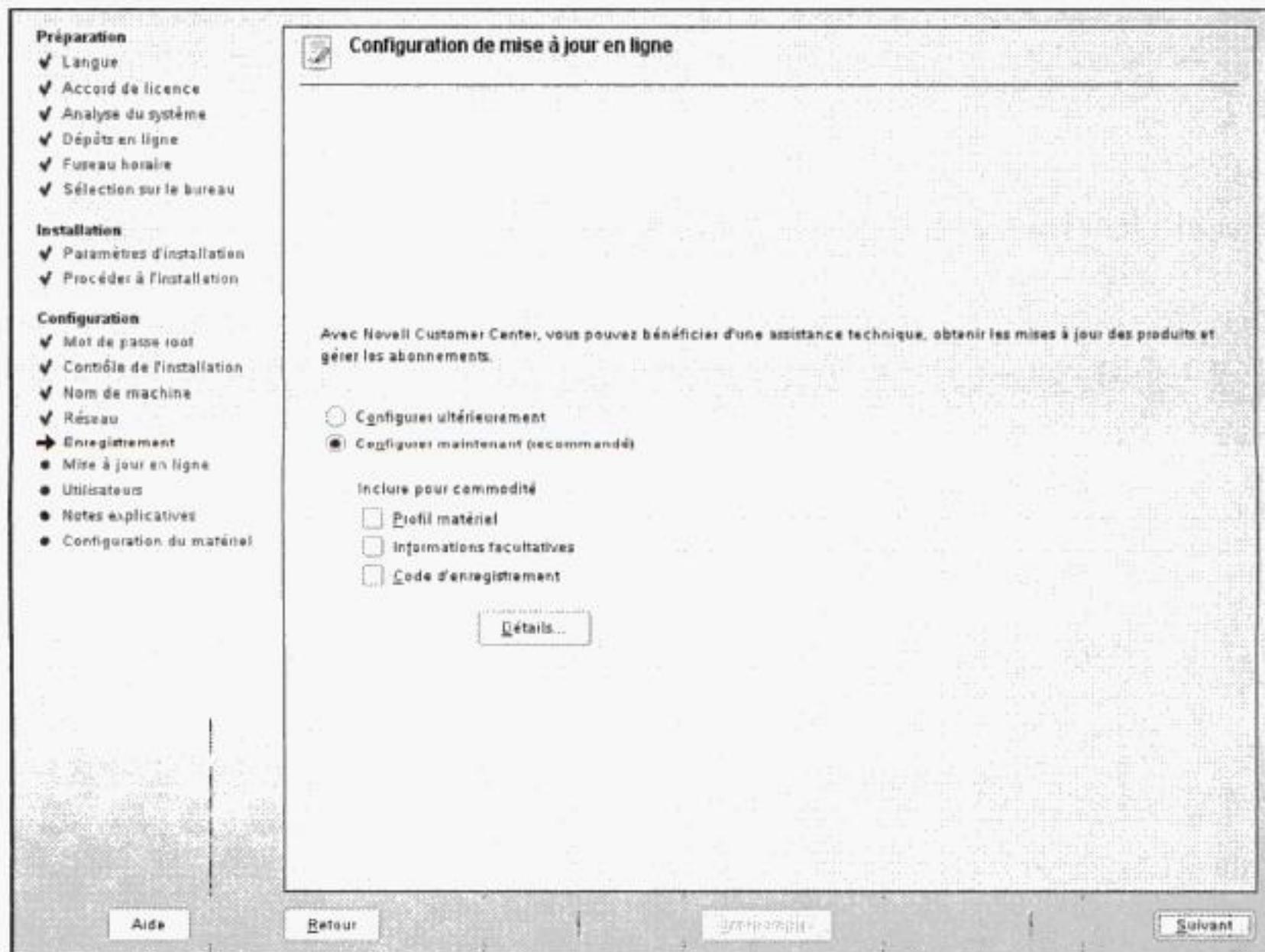
Si vous ignorez le test, ou qu'aucune connexion à Internet n'est disponible, l'étape suivante de mise à jour du système ne sera pas accessible. Vous devrez passer par le module d'installation de dépôt de mise à jour de YaST après l'installation pour accéder aux mises à jour.



10. Mise à jour du système

Si votre connexion Internet est correctement configurée, vous pouvez maintenant configurer votre source de mise à jour. Cliquez sur **Configurer maintenant**. YaST recherche un miroir sur Internet pour configurer une source.

Vous pouvez en profiter pour fournir auprès du support openSUSE des informations sur votre installation et sur votre machine et un éventuel code d'enregistrement pour le support (rarement utilisé avec les versions openSUSE, le support étant plutôt communautaire). Aucune information personnelle ne circule, et tout se fait sur votre autorisation.

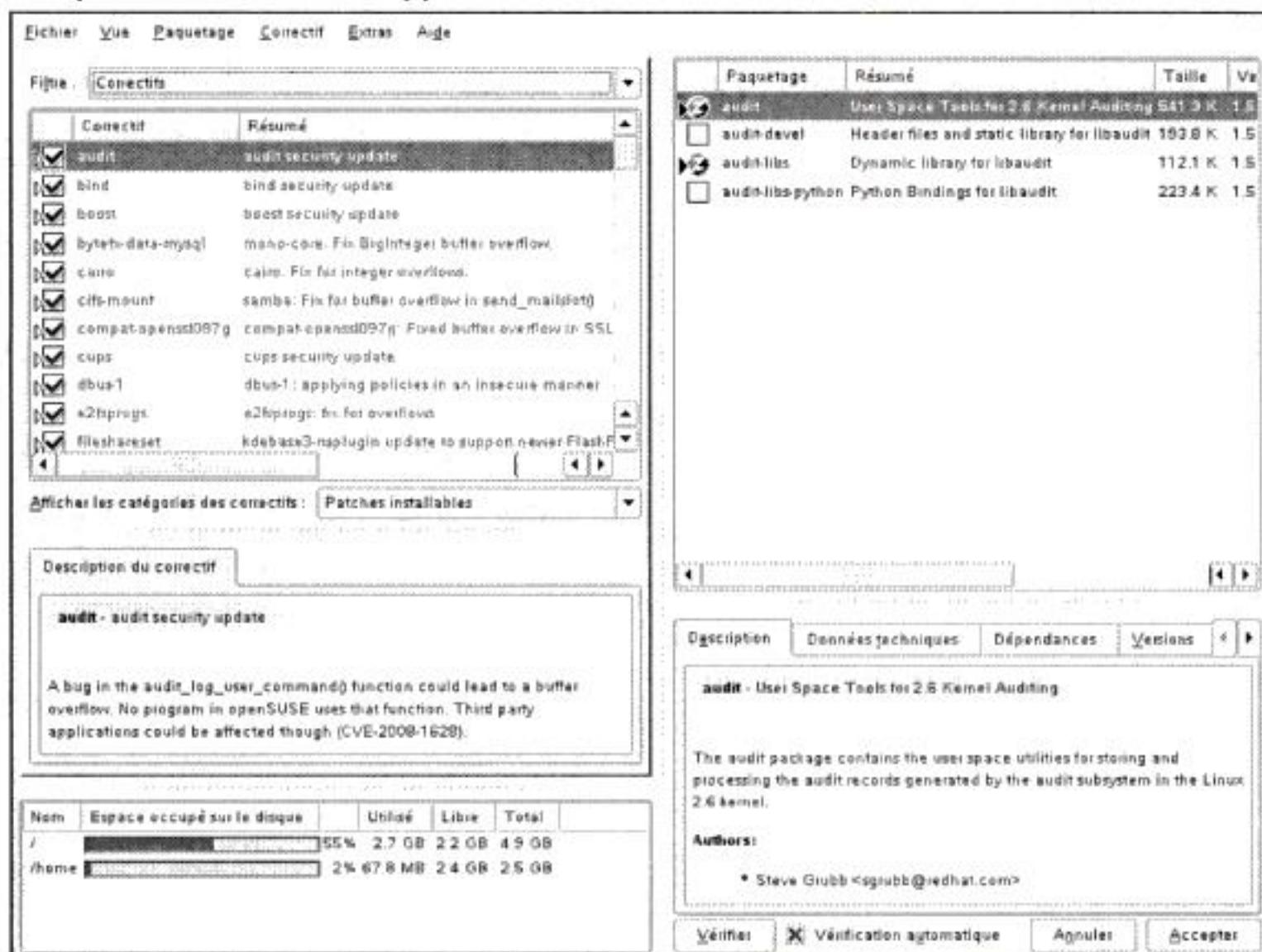


Si cette étape s'est déroulée correctement, vous pouvez demander à ce que la mise à jour de votre installation s'effectue dès à présent. Cette étape va télécharger les packages récents (mises à jour de sécurité et correction de bug notamment) et les installer.

Il se peut que le processus de mise à jour redémarre plusieurs fois. Il n'est pas étonnant de voir que les composants gérant les mises à jour (le backend zypp sous openSUSE) sont eux-mêmes mis à jour. Dans ce cas le processus de mise à jour redémarre pour réutiliser les nouveaux composants.

En principe une installation de openSUSE n'a pas besoin d'un redémarrage de l'ordinateur et s'effectue d'un trait : aucun reboot entre le démarrage sur le support et le premier login. Ceci s'appuie sur le fait que le noyau (et donc les pilotes) installé est le même que sur le support d'installation. Or lors d'une mise à jour, il se peut qu'un nouveau noyau, composant très critique, soit installé. Dans ce cas, le processus d'installation doit redémarrer l'ordinateur pour recharger et gérer les nouveaux modules du noyau (dont les pilotes) afin de pouvoir configurer le matériel par la suite.

L'installation des mises à jour utilise la même interface que l'installation de tout autre package. Les correctifs majeurs sont en rouge : ils corrigent un bug important présentant un risque de sécurité. Si vous descendez dans la liste, vous trouverez aussi des mises à jour qui ne sont pas critiques, ou qui n'ont pas forcément besoin d'être installées pour garantir le bon fonctionnement de Linux. C'est le cas par exemple de l'installation des polices (fontes) TrueType fournies par Microsoft. Elles apportent cependant un confort supplémentaire à l'utilisation.



Choix et installation des mises à jour

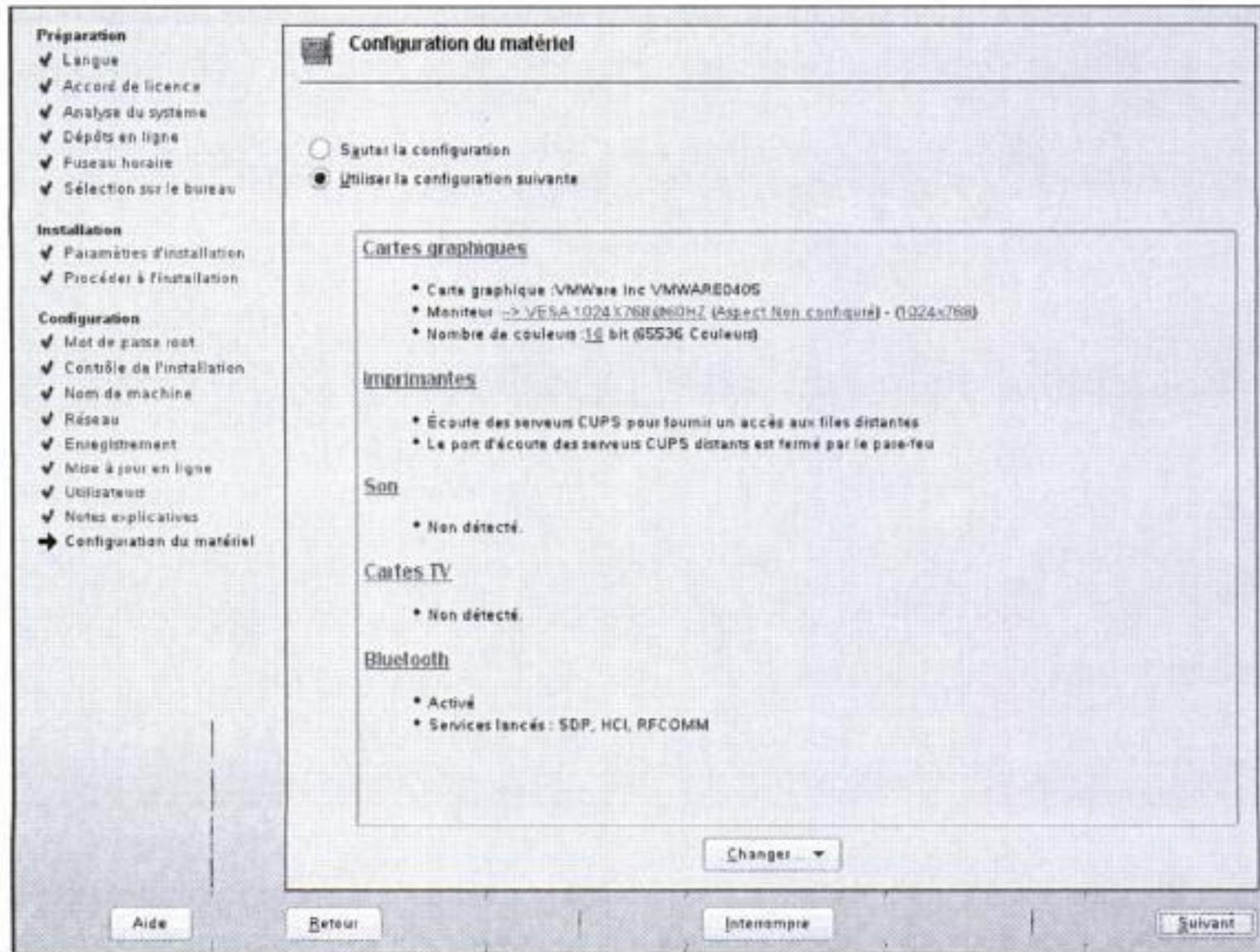
11. Gestion des utilisateurs

L'étape suivante consiste à rajouter des utilisateurs. Outre les utilisateurs locaux, vous pouvez vous raccorder à une base LDAP, un serveur NIS et même un domaine Windows pour l'authentification.

Vous pouvez créer un seul ou plusieurs utilisateurs en cliquant sur le bouton **Gestion utilisateurs**. Vous pouvez aussi faire en sorte d'activer l'autoconnexion de l'utilisateur afin d'éviter la saisie des logins et mots de passe. N'utilisez pas cette possibilité si plusieurs utilisateurs existent. De même ce n'est pas forcément une bonne idée de router les messages à destination de l'administrateur vers un compte utilisateur, sauf si celui-ci a comme vocation d'être utilisé comme passerelle vers les tâches d'administration.

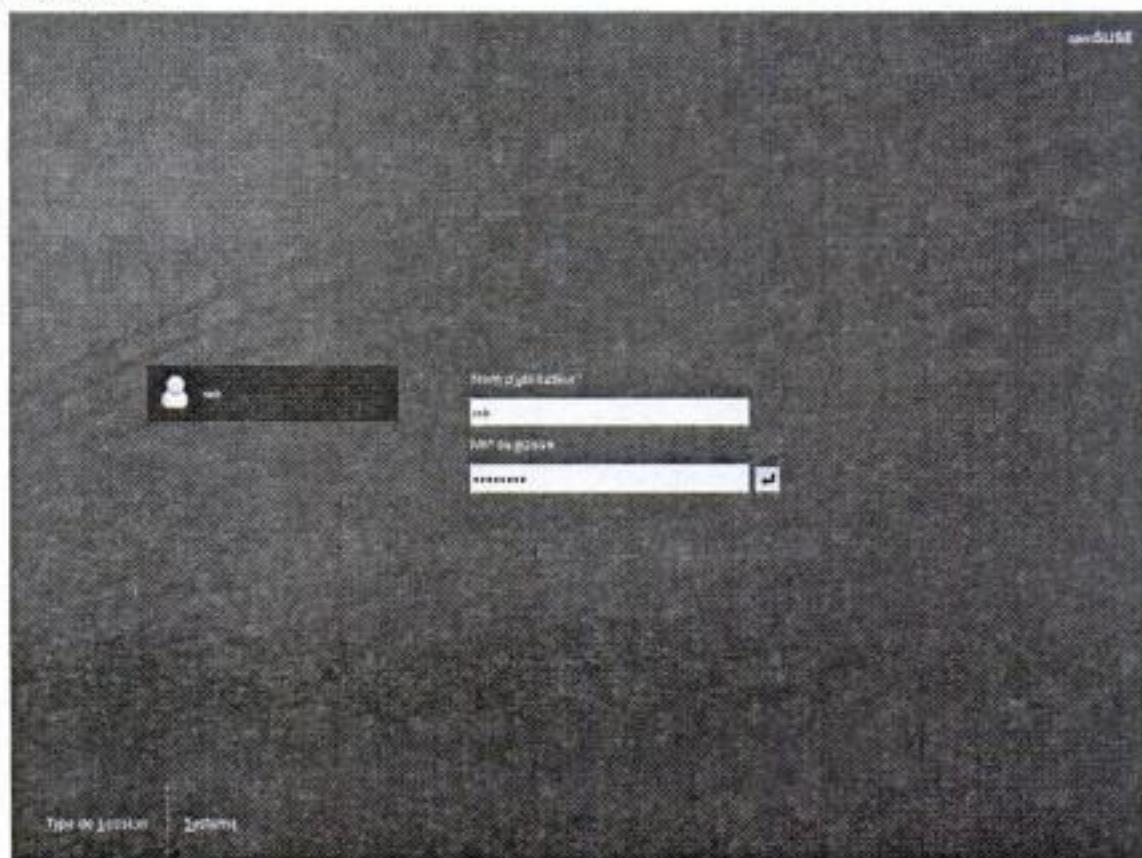
12. Configuration du matériel

La configuration du matériel est la dernière étape de l'installation de openSUSE. Cette distribution est très réputée pour la détection et le support du matériel qu'elle propose, dans la limite des pilotes disponibles. Il est fort probable que vos uniques actions consistent à modifier le moniteur ou plutôt la résolution par défaut, et peut-être (si elles sont reliées par réseau ou port parallèle) les imprimantes. Les scanners devront être configurés après la fin de l'installation.



Cliquez sur **Suivant**. La configuration du matériel est enregistrée. Un écran vous informe enfin de la fin de l'installation. Il n'y a pas de redémarrage : l'installateur se termine et il est possible de commencer directement à travailler.

La dernière capture montre le gestionnaire de sessions KDM dont le thème a été modifié aux couleurs de la distribution.





You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

La base de données dpkg est généralement placée dans `/var/lib/dpkg`. Les fichiers qui y sont présents sont au format texte. Cependant n'éditez pas les fichiers à la main. Le fichier `/var/lib/dpkg/status` contient l'intégralité des packages connus par dpkg avec leur état.

```
# grep ^Package: /var/lib/dpkg/status | grep glibc
Package: devhelp-book-glibc
Package: libg++2.8.1.3-glibc2.2
Package: glibc-doc
Package: libstdc++2.10-glibc2.2
```

Dpkg dispose d'une interface graphique, GDebi, qui permet d'éviter l'utilisation de la ligne de commande.

2. Installation, mise à jour et suppression

L'option `-i`, ou `--install`, installe le ou les packages passés comme argument.

```
# dpkg -i monpaquet.deb
```

Notez que comme rpm, dpkg ne gère pas seul les dépendances. S'il manque des dépendances, la commande vous en informera. Dans ce cas, vous devez installer les dépendances de la même manière avant d'installer votre package.

Vous pouvez demander l'installation de tous les packages présents au sein d'une arborescence avec le paramètre `-R`, pour récursif. Dans ce cas, indiquez comme argument un nom de répertoires : tous les packages présents dans le répertoire et ses sous-répertoires seront installés.

```
# dpkg -R folder
```

La mise à jour s'effectue de la même manière que l'installation, avec le `-i`. Si vous installez un package déjà présent, dpkg en effectue une mise à jour. Ainsi, une installation ou une mise à jour respectent la méthodologie suivante :

- Extraction des fichiers de contrôle du nouveau paquet.
- Quand une ancienne version du même paquet est déjà installée, exécution du script pré-suppression de l'ancien paquet.
- Lancement du script de préinstallation s'il est fourni par le paquet.
- Dépaquetage des nouveaux fichiers et sauvegarde des anciens pour pouvoir les restaurer en cas de problème.
- Si une ancienne version du paquet est déjà installée, exécution du script de post-suppression de l'ancien paquet.
- Configuration du paquet.

Il n'existe pas d'équivalence au mode `freshen` (`-F`) de rpm. Si vous souhaitez mettre à jour un package uniquement s'il est déjà installé, vous devez tout d'abord vérifier s'il est installé. Si un package est installé il commence par `ii` dans la liste :

```
# (dpkg -l zip | grep ^ii >/dev/null) && echo PRESENT || echo ABSENT
PRESENT
# (dpkg -l slapd | grep ^ii >/dev/null) && echo PRESENT || echo ABSENT
ABSENT
```

Pour mettre à jour un paquet seulement s'il est présent utilisez ce genre de ligne de commande, ou un équivalent de votre cru :

```
# (dpkg -l zip | grep ^ii >/dev/null) && dpkg -l zip.deb
```

Notez que le plus simple pour mettre à jour vos paquets déjà présents est de créer un dépôt APT contenant vos mises à jour et d'exécuter un apt-get upgrade depuis le client.

La suppression d'un package s'effectue avec le paramètre `-r` (en minuscule). Là encore, c'est à vous de gérer les dépendances.

La suppression d'un paquet effectue les étapes suivantes :

- Exécution du script de pré-suppression.
- Suppression des fichiers installés.
- Exécution du script de post-suppression.

```
# dpkg -r zip
```

Tout est supprimé sauf les fichiers de configuration et ce, afin d'éviter une reconfiguration de l'outil si vous le réinstallez. Pour tout supprimer, y compris ces fichiers, précisez le paramètre `-P` (purge).

```
# dpkg -P apache
```

Attention : ne confondez pas les paramètres `-r` et `-P` au risque d'un drame !

Si vous remplacez le nom du package par les paramètres `-a` ou `--pending` les packages non installés (non dépaquetés) mais présents dans les informations de la base pour être purgés ou supprimés, sont effacés.

L'utilisation des options `--force-all` et `--purge` permet de forcer la désinstallation du paquet et de supprimer les fichiers de configuration associés.

```
# dpkg --force-all --purge nom_du_paquet
```

3. Requêtes dpkg

a. Lister les paquets

Vous pouvez lister tous les packages Debian connus du système avec le paramètre `-l`.

```
# dpkg -l
...
ii  adduser           3.102
Add and remove users and groups
ii  alien              8.64
install non-native packages with dpkg
rc  amavisd-new       2.4.2-6.1
Interface between MTA and virus scanner/cont
ii  antlr               2.7.6-7
language tool for constructing recognizers,
rc  apache             1.3.34-4.1
versatile, high-performance HTTP server
ii  apache-common      1.3.34-4.1+etch1
support files for all Apache webservers
ii  apache2             2.2.3-4+etch4
Next generation, scalable, extendable web se
rc  apache2-common     2.0.54-5sarge2
next generation, scalable, extendable web se
ii  apache2-mpm-prefork 2.2.3-4+etch4
Traditional model for Apache HTTPD 2.1
ii  apache2-utils       2.2.3-4+etch4
utility programs for webservers
ii  apache2.2-common    2.2.3-4+etch4
Next generation, scalable, extendable web se
...
...
```

Vous pouvez indiquer un motif particulier :

```
# dpkg -l "apt*" | grep ^ii
ii apt 0.6.46.4-0.1 Advanced front-end for dpkg
ii apt-file 2.0.3-7 APT package searching utility -
command-line
ii apt-listchanges 2.72.5etch2 Display change history from .deb
archives

ii apt-rpm-repository 0.5.15lorg3.2-1 tools to create an APT RPM repository
ii apt-utils 0.6.46.4-0.1 APT utility programs
ii aptitude 0.4.4-4 terminal-based apt frontend
```

Astuce : si votre console est trop petite pour afficher les noms des packages (seconde colonne) vous pouvez ruser comme ceci :

```
# COLUMNS=160 dpkg -l "kernel*" | grep ^ii | awk '{print $2}'
kernel-image-2.6.7-1-686
kernel-image-2.6.8-1-686-smp
kernel-image-2.6.8-2-686
kernel-image-2.6.8-2-686-smp
```

Une autre méthode consiste à employer l'option **--get-selections** :

```
# dpkg --get-selections | grep kernel
fai-kernels install
kernel-image-2.6.7-1-686 install
kernel-image-2.6.8-1-686-smp install
kernel-image-2.6.8-2-686 install
kernel-image-2.6.8-2-686-smp install
linux-kernel-headers install
```

b. Trouver un paquet contenant un fichier

Le paramètre **-S** suivi du nom d'un fichier (son chemin) permet de retrouver le paquet d'origine.

```
# dpkg -S /usr/bin/basename
coreutils: /usr/bin/basename
```

c. Lister le contenu d'un paquet

Le paramètre **-L** liste le contenu du ou des paquets indiqués :

```
# dpkg -L coreutils | grep bin
/bin
/bin/mkdir
/bin/mv
/bin/true
/bin/mknod
/bin/sleep
/bin/touch
/bin/chgrp
/bin/uname
/bin/echo
/bin/sync
/bin/ln
/bin/date
/bin/dir
/bin/readlink
...
```

4. Convertir des packages

L'outil **alien** permet de convertir des packages RPM en DPKG et vice versa. Certains packages ne sont fournis que pour l'un ou l'autre des systèmes. C'est embêtant lorsqu'un produit n'est fourni que sous une forme et qu'il faut tout de même l'installer sur une autre plate-forme Linux.

Voici l'exemple d'un package, le client Networker, uniquement fourni pour Red Hat. Avec Alien, il est possible de le convertir au format dpkg.

Le paramètre par défaut **-d** convertit du rpm au dpkg :

```
# alien -d lgtocnt-7.4-1.i686.rpm
Warning: Skipping conversion of scripts in package lgtocnt: postinst postrm
preinst prerm
Warning: Use the --scripts parameter to include the scripts.
lgtocnt_7.4-2_i386.deb generated
```

Comme indiqué, la conversion par défaut va vérifier les dépendances, mais ne va pas inclure les scripts de pré-installation et de post-installation. Vous devez alors préciser le paramètre **--scripts**.

```
# alien --scripts -d lgtocnt-7.4-1.i686.rpm
lgtocnt_7.4-2_i386.deb generated
# ls -l *.deb
-rw-r--r-- 1 root root 29471546 2008-05-09 14:45 lgtocnt_7.4-2_i386.deb
```

Le résultat est le suivant :

```
# dpkg -I lgtocnt_7.4-2_i386.deb
nouveau paquet Debian, version 2.0.
taille 29471546 octets: archive de controle = 4498 octets.
    923 octets,   18 lignes      control
    3142 octets,   57 lignes      md5sums
    4014 octets,  148 lignes    * postinst          #!/bin/sh
    1362 octets,   35 lignes    * postrm           #!/bin/sh
    317 octets,   11 lignes    * preinst          #!/bin/sh
    1828 octets,   52 lignes    * prerm            #!/bin/sh
       61 octets,     3 lignes      shlibs
Package: lgtocnt
Version: 7.4-2
Section: alien
Priority: extra
Architecture: i386
Depends: libc6 (>= 2.3.6-6), libgl1-mesa-glx | libgl1, libice6 (>=
1:1.0.0), libncurses5 (>= 5.4-5), libsm6, libx11-6, libxext6,
libxmu6, libxp6, libxrender1, libxt6
Installed-Size: 71632
Maintainer: root <root@s64p17bib76.dsit.sncf.fr>
Description: NetWorker Client
  EMC NetWorker protects the critical business data of more than 10,000
...
  and the smallest satellite branch offices.

  (Converted from a rpm<+>package by alien version 8.64.)
```



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

3. Mise à jour de la distribution

Une fois les dépôts à jour, vous pouvez mettre à jour en une seule commande tous les packages installés sur votre distribution : APT vérifie si des packages plus récents sont disponibles dans les dépôts. Il se base pour cela sur la base de données locale. Si elle n'est pas à jour il est possible que certains des packages trop anciens ne soient plus présents.

Exécutez la commande **apt-get** avec l'option **upgrade**. APT vous informe que huit packages peuvent être mis à jour. Vous pouvez accepter ou refuser. Si vous acceptez, APT télécharge ces packages et leurs éventuelles dépendances, et les installe. Le processus peut être plus ou moins long selon le nombre de mises à jour et le type de support.

```
# apt-get upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Les paquets suivants seront mis à jour :
  cpio libgnutls13 libspeexl libssl0.9.8 linux-image-2.6.18-6-686
    openssh-client openssl rdesktop
8 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 21,3Mo dans les archives.
Après dépaquetage, 1876ko d'espace disque seront libérés.
Souhaitez-vous continuer [O/n] ? O
Réception de : 1 http://security.debian.org etch/updates/main linux-
image-2.6.18
-6-686 2.6.18.dfsg.1-18etch4 [16,3MB]
Réception de : 2 http://security.debian.org etch/updates/main cpio
2.6-18.1+etch
1 [132kB]
Réception de : 3 http://security.debian.org etch/updates/main lib-
gnutls13
1.4.4-
3+etch1 [282kB]
Réception de : 4 http://security.debian.org etch/updates/main libs-
s10.9.8
0.9.8c
-4etch3 [2717kB]
Réception de : 5 http://security.debian.org etch/updates/main opens-
sh-client
1:4
.3p2-9etch2 [660kB]
Réception de : 6 http://security.debian.org etch/updates/main libs-
peexl
1.1.12-3
etch1 [76,4kB]
Réception de : 7 http://security.debian.org etch/updates/main openssl
0.9.8c-4et
ch3 [1001kB]
Réception de : 8 http://security.debian.org etch/updates/main rdesktop
1.5.0-1et
ch2 [124kB]
21,3Mo réceptionnés en 2m51s (125ko/s)
Préconfiguration des paquets...
(Lecture de la base de données... 80122 fichiers et répertoires déjà
installés.)
...
Préparation du remplacement de cpio 2.6-18 (en utilisant
.../cpio_2.6-18.1+etch1
```

```

_i386.deb) ...
Dépaquetage de la mise à jour de cpio ...
Préparation du remplacement de libgnutls13 1.4.4-3 (en utilisant
.../libgnutls13
1.4.4-3+etch1_i386.deb) ...
Dépaquetage de la mise à jour de libgnutls13 ...
Préparation du remplacement de libssl0.9.8 0.9.8c-4etch1 (en utilisant
.../libssl
10.9.8_0.9.8c-4etch3_i386.deb) ...
Dépaquetage de la mise à jour de libssl0.9.8 ...
Préparation du remplacement de openssh-client 1:4.3p2-9 (en utilisant
.../openssh
h-client_1%3a4.3p2-9etch2_i386.deb) ...
Dépaquetage de la mise à jour de openssh-client ...
Préparation du remplacement de libspeex1 1.1.12-3 (en utilisant
.../libspeex1_1.
1.12-3etch1_i386.deb) ...
Dépaquetage de la mise à jour de libspeex1 ...
Préparation du remplacement de openssl 0.9.8c-4etch1 (en utilisant
.../openssl_0
.9.8c-4etch3_i386.deb) ...
Dépaquetage de la mise à jour de openssl ...
Préparation du remplacement de rdesktop 1.5.0-1etch1 (en utilisant
.../rdesktop_
1.5.0-1etch2_i386.deb) ...
Dépaquetage de la mise à jour de rdesktop ...
...
Paramétrage de cpio (2.6-18.1+etch1) ...
Paramétrage de libgnutls13 (1.4.4-3+etch1) ...
Paramétrage de libssl0.9.8 (0.9.8c-4etch3) ...
Checking for services that may need to be restarted...done.
Checking init scripts...
Restarting services possibly affected by the upgrade:
  exim4: stopping...starting...done.
Services restarted successfully.
...

```

Une autre possibilité est de faire une mise à jour profonde (appelée mise à jour distante). APT garde une certaine cohérence dans les packages lors de la mise à jour, notamment concernant la version de la distribution. Vous pouvez spécifier plusieurs distributions Debian dans vos dépôts. Mais même si une distribution est plus récente, un simple upgrade ne va pas transformer la vôtre en la toute dernière. Vous pouvez demander à APT de forcer la mise à jour vers la nouvelle distribution avec un **dist-upgrade**.

Pour les besoins de cet ouvrage, les dépôts de la version de test Debian appelée lenny ont été ajoutés :

```

## lenny
deb http://ftp.fr.debian.org/debian/ lenny main contrib non-free
deb-src http://ftp.fr.debian.org/debian/ lenny main contrib non-free
# security lenny
deb http://security.debian.org/ lenny/updates main contrib non-free
deb-src http://security.debian.org/ lenny/updates main contrib non-free

```

Effectuez une mise à jour de la base. Notez qu'il a fallu agrandir le cache de APT pour ceci et nettoyer la base :

```
# apt-get update
echo 'APT::Cache-Limit "141943904";' > /etc/apt/apt.conf.d/00Cache
# apt-get clean
# apt-get update
```

puis

```
$ apt-get dist-upgrade
Lecture des listes de paquets...
Construction de l'arbre des dépendances...
...
731 mis à jour, 228 nouvellement installés, 23 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 858Mo dans les archives.
Après dépaquetage, 623Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer [O/n] ? O
...
```

Bon courage !

4. Rechercher et installer un package individuel

La commande **apt-cache** permet de rechercher un package, par son nom ou son commentaire, au sein de la base de données locale APT.

```
# apt-cache search torrent
bittornado - bittorrent client with enhanced curses interface
bittornado-gui - bittorrent client with enhanced GUI interface
bittorrent - Scatter-gather network file transfer
bittorrent-gui - Scatter-gather network file transfer (GUI files)
cfv - versatile file checksum creator and verifier
qtorrent - BitTorrent client for QT 3.x
```

La commande **apt-get install xxx** installe le package **xxx** :

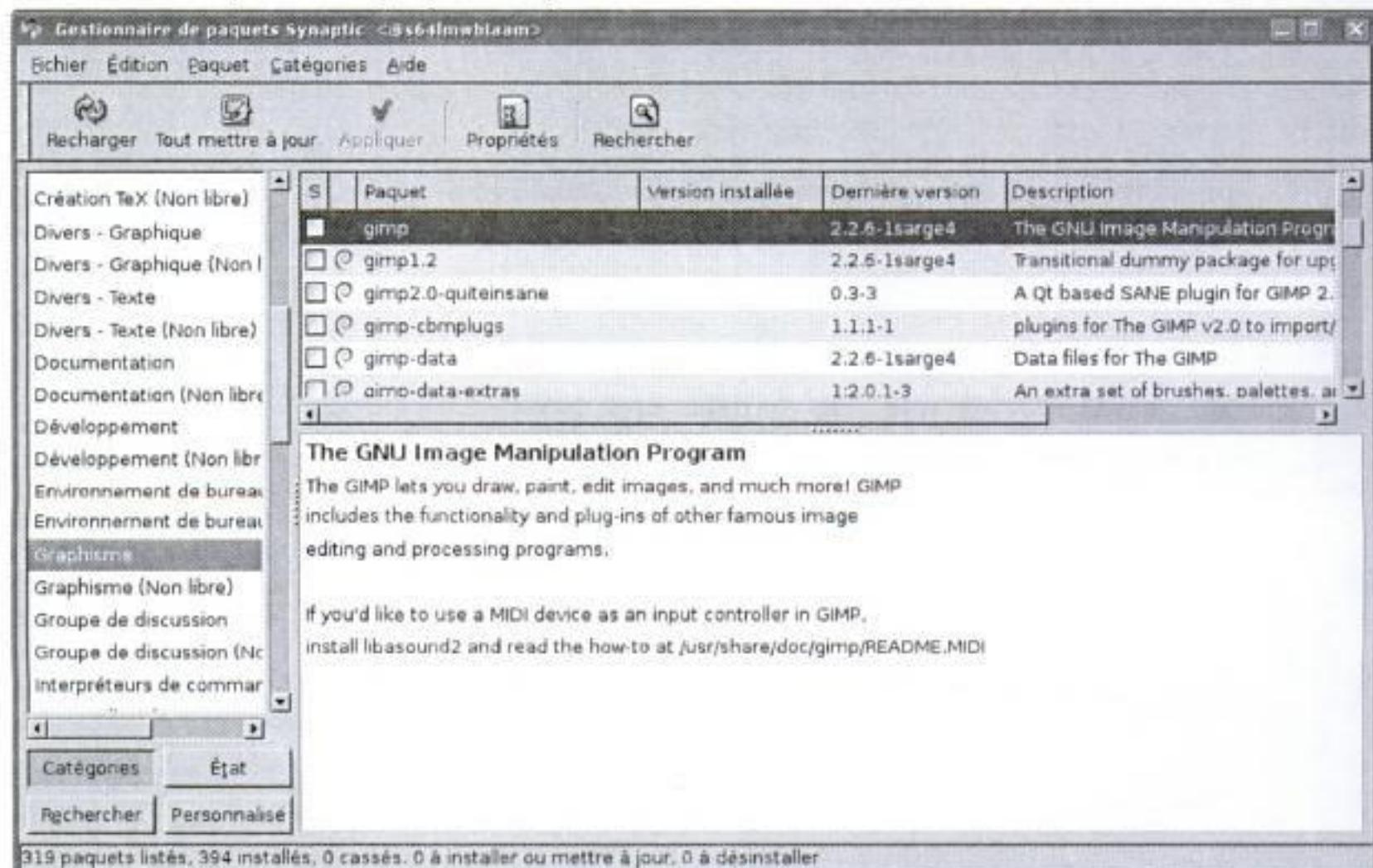
```
# apt-get install vim-gtk
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Les paquets supplémentaires suivants seront installés :
  vim vim-common
Paquets suggérés :
  ctags vim-doc vim-scripts cscope
Les NOUVEAUX paquets suivants seront installés :
  vim-gtk
Les paquets suivants seront mis à jour :
  vim vim-common
2 mis à jour, 1 nouvellement installés, 0 à enlever et 58 non mis à jour.
Inst vim [1:6.3-071+1sarge1] (1:6.3-071+1sarge3 Debian:3.1r8/oldstable) []
Inst vim-common [1:6.3-071+1sarge1] (1:6.3-071+1sarge3 Debian:3.1r8/oldstable)
Inst vim-gtk (1:6.3-071+1sarge3 Debian:3.1r8/oldstable)
Conf vim-common (1:6.3-071+1sarge3 Debian:3.1r8/oldstable)
Conf vim (1:6.3-071+1sarge3 Debian:3.1r8/oldstable)
Conf vim-gtk (1:6.3-071+1sarge3 Debian:3.1r8/oldstable)
```

Deux options méritent d'être retenues :

- le **-s** pour la simulation : APT indique ce qu'il devrait faire, mais ne le fait pas.
- le **-f** pour « fix-broken » : APT tente de réparer les problèmes de dépendances comme il le peut (ajout de packages).

5. Client graphique

L'outil synaptic est un front-end : une interface graphique qui fait appel aux fonctions de APT. Il permet toutes les opérations proposées par APT tout en étant très convivial.



Synaptic est un front-end à APT.

G. Installer depuis les sources

1. Obtenir les sources

Il n'est parfois pas possible d'obtenir un logiciel ou une bibliothèque depuis un package pour sa distribution. Dans ce cas, il reste la solution de compiler et d'installer soi-même le produit depuis les sources.

Cela est possible pour une majorité de produits sous Linux, grâce aux avantages des logiciels libres et de la licence GPL telle que définie au premier chapitre. Tout logiciel libre est fourni avec ses sources. Il est donc possible de reconstruire soi-même le logiciel en le recompilant.

Une archive source est souvent récupérée sur divers sites Internet comme par exemple SourceForge. C'est une archive bien souvent compressée au format tgz (archive tar compressée avec gzip) ou tar.bz2 (archive tar compressée au format bzip2). Elle contient :

- le code source sous forme de fichiers .c, .h, .cpp, etc., selon le langage ;
- parfois un fichier **Makefile** permettant d'automatiser la compilation du produit ;
- souvent un fichier **.configure** permettant de générer le fichier Makefile en fonction de votre installation et de diverses options.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Première règle : pour exécuter la règle `bonjour` il faut disposer des fichiers `bonjour.o` et `main.o`. Si on les a, il faut exécuter la commande `gcc -o bonjour bonjour.o main.o`.

Deuxième règle : pour exécuter la règle `bonjour.o` il faut disposer du fichier `bonjour.c`. S'il est présent alors la commande `gcc -o bonjour.o -c bonjour.c` est exécutée.

Troisième règle : pour exécuter la règle `main.o` il faut disposer des fichiers `main.c` et `bonjour.h`. S'ils sont présents alors la commande `gcc -o main.o -c main.c` est exécutée.

Les deux dernières règles permettent de résoudre la première. Si vous lancez la commande `make`, elle va déterminer quelles sont les règles applicables, dans quel ordre, et les appliquer, dans l'ordre des dépendances. Si les fichiers sont à jour, `make` ne les reconstruit pas sauf s'ils ont été modifiés.

```
$ rm -f *.o
$ make
gcc -o bonjour.o -c bonjour.c
gcc -o main.o -c main.c
gcc -o bonjour bonjour.o main.o
$ ./bonjour
Bonjour
```

b. Makefile intermédiaire

Le Makefile précédent fonctionne mais n'est pas optimal :

- Il ne permet pas de compiler plusieurs binaires.
- Il ne permet pas de nettoyer les fichiers temporaires (.o) après la compilation.
- Il ne permet pas de forcer la recompilation du projet.

L'ajout de nouvelles règles permet de pallier ces problèmes :

- `all` : génère n règles ;
- `clean` : nettoie les .o ;
- `mrproper` : appelle `clean` et supprime les binaires.

```
$ cat Makefile
all: bonjour

bonjour: bonjour.o main.o
        gcc -o bonjour bonjour.o main.o

bonjour.o: bonjour.c
        gcc -o bonjour.o -c bonjour.c

main.o: main.c bonjour.h
        gcc -o main.o -c main.c

clean:
        rm -rf *.o

mrproper: clean
        rm -rf bonjour

$ make clean
rm -rf *.o
$ make mrproper
rm -rf *.o
rm -rf bonjour
$ make all
gcc -o bonjour.o -c bonjour.c
```

```
gcc -o main.o -c main.c
gcc -o bonjour bonjour.o main.o
```

c. Un peu plus complexe

Variables utilisateur

Pour finir cette petite présentation, vous pouvez définir des variables dans votre fichier, et utiliser des variables internes prédéfinies :

Le Makefile devient :

```
$ cat Makefile
CC=gcc
CFLAGS=-W -Wall -ansi -pedantic
LDFLAGS=
EXEC=bonjour

all: $(EXEC)

bonjour: bonjour.o main.o
        gcc -o bonjour bonjour.o main.o $(LDFLAGS)

bonjour.o: bonjour.c
        gcc -o bonjour.o -c bonjour.c $(CFLAGS)

main.o: main.c bonjour.h
        gcc -o main.o -c main.c $(CFLAGS)

clean:
        rm -rf *.o

mrproper: clean
        rm -rf $(EXEC)
```

Variables internes

Parmi les variables internes :

- \$@ : nom de la cible.
- \$< : nom de la première dépendance.
- \$^ : liste des dépendances.
- \$? : dépendances plus récentes que la cible.
- \$* : nom du fichier sans le suffixe.

Le Makefile devient :

```
$ cat Makefile
CC=gcc
CFLAGS=-W -Wall -ansi -pedantic
LDFLAGS=
EXEC=bonjour

all: $(EXEC)

bonjour: bonjour.o main.o
        gcc -o $@ $^ $(LDFLAGS)
```

```

bonjour.o: bonjour.c
    gcc -o $@ -c $ $(CFLAGS)

main.o: main.c bonjour.h
    gcc -o $@ -c $ $(CFLAGS)

clean:
    rm -rf *.o

mrproper: clean
    rm -rf $(EXEC)

```

Règles d'inférence

Il existe des règles prédéfinies, à base de raccourcis, qui permettent de générer des cibles en fonction du nom du fichier C et objet : `%.o : %.c`. La tentation est grande de créer une règle unique pour `main.o` et `bonjour.o` :

```

%.o: %.c
    gcc -o $@ -c $ $(CFLAGS)

```

Cette règle est correcte mais il manque la dépendance du header `bonjour.h` : si celui-ci est modifié le projet n'est plus compilé. Il faut rajouter une règle spécifique :

```
main.o : bonjour.h
```

Le Makefile devient :

```

$ cat Makefile
CC=gcc
CFLAGS=-W -Wall -ansi -pedantic
LDFLAGS=
EXEC=bonjour

all: $(EXEC)

bonjour: bonjour.o main.o
    gcc -o $@ $^ $(LDFLAGS)

%.o: %.c
    gcc -o $@ -c $ $(CFLAGS)

main.o: bonjour.h

clean:
    rm -rf *.o

mrproper: clean
    rm -rf $(EXEC)

```

H. Gérer les bibliothèques partagées

1. Principe

Une bibliothèque partagée est un fichier particulier qui contient une liste de fonctions, ou API, accessible à tout programme en ayant besoin sans avoir à les réécrire. À l'opposé de la bibliothèque statique, le programme accède dynamiquement aux fonctions qui sont placées dans un fichier à part. N programmes différents peuvent accéder aux fonctions proposées par la bibliothèque. Les bibliothèques regroupent des fonctions propres à un domaine ou un ensemble de domaines cohérents : traitement d'images, du son, de l'accès à une base de données, etc.

Un ensemble de fonctions proposées par une ou plusieurs bibliothèques partagées forme une **API**, *Application Programming Interface*, et sont parfois regroupées au sein d'un framework offrant une solution complète pour un domaine donné.

Un lien est établi entre le programme et une bibliothèque partagée lors de l'étape de l'édition des liens par l'éditeur de liens **ld**, lui-même appelé par le compilateur **gcc** avec l'option **-l<lib>**.

Une autre possibilité pour un programme est d'utiliser la fonction C **dlopen** qui ouvre une bibliothèque dynamique comme un fichier et qui accède aux fonctions qui y sont contenues avec des pointeurs de fonctions.

Si un programme dépend d'une bibliothèque partagée et que celle-ci est absente, le programme ne pourra plus fonctionner.

Sous Linux (et Unix en général) les bibliothèques partagées sont appelées des **Shared Objects** (so) dans le sens où il s'agit de fichiers objets sans bloc d'instruction **main**. Ils portent le suffixe **.so**.

Une bibliothèque peut disposer de plusieurs versions, pouvant être ou non compatibles, et la version peut être précisée lors de l'édition des liens, avec une version par défaut possible.

2. Lieu de stockage

Les bibliothèques partagées sont par convention placées dans des répertoires appelés lib :

- **/lib** : bibliothèques systèmes de base, vitales ;
- **/usr/lib** : bibliothèques utilisateur de base, non nécessaires au boot ;
- **/usr/local/lib** : bibliothèques locales aux produits pour la machine ;
- **/usr/X11R6/lib** : bibliothèques de l'environnement X Window ;
- **/opt/kde3/lib** : bibliothèques de KDE...

```
$ ls -l /lib
total 6024
...
-rwxr-xr-x 1 root root 114636 oct 23 2007 ld-2.6.1.so
lrwxrwxrwx 1 root root      11 oct  5 2007 ld-linux.so.2 -> ld-2.6.1.so
lrwxrwxrwx 1 root root      13 oct  5 2007 ld-lsb.so.2 -> ld-linux.so.2
lrwxrwxrwx 1 root root      13 oct  5 2007 ld-lsb.so.3 -> ld-linux.so.2
lrwxrwxrwx 1 root root      15 oct  5 2007 libacl.so.1 -> libacl.so.1.1.0
-rwxr-xr-x 1 root root  27864 sep 22 2007 libacl.so.1.1.0
lrwxrwxrwx 1 root root      15 oct  5 2007 libaio.so.1 -> libaio.so.1.0.1
-rwxr-xr-x 1 root root    5248 sep 21 2007 libaio.so.1.0.1
-rwxr-xr-x 1 root root 10256 oct 23 2007 libanl-2.6.1.so
lrwxrwxrwx 1 root root      15 oct  5 2007 libanl.so.1 -> libanl-2.6.1.so
lrwxrwxrwx 1 root root      20 oct  5 2007 libapparmor.so.1 ->
libapparmor.so.1.0.2
-rwxr-xr-x 1 root root 30404 sep 22 2007 libapparmor.so.1.0.2
lrwxrwxrwx 1 root root      16 oct  5 2007 libattr.so.1 ->
libattr.so.1.1.0
```



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

J. Travaux pratiques

1. Schéma de partitionnement

Ce TP a pour but de déterminer le meilleur schéma de partitionnement possible, ce qui représente la plus grande difficulté pour un débutant. Le TP convient pour toutes les distributions. Vous disposez sur votre PC personnel d'un disque de 160 Go dont 40 sont déjà occupés par un autre système. Votre machine dispose de 2 Go de mémoire vive. Il vous reste 120 Go d'espace disque. Comment pouvez-vous les répartir, sachant que vous voulez séparer vos données du système ?

1. Quelle doit être la taille de la partition d'échange SWAP ?

La partition d'échange est utilisée lorsque Linux ne dispose plus d'assez de place en mémoire vive pour traiter toutes ses données. Les données sont déplacées en mémoire virtuelle sur cette partition d'échange pour libérer plus ou moins temporairement de la mémoire pour d'autres données.

Votre système disposant de 2 Go de mémoire, vous pouvez ne prévoir que 2 Go de swap. Il reste alors 118 Go d'espace disque pour la suite.

2. Quelle place réserver au système / ?

Même si vous deviez installer tous les produits présents sur un DVD d'installation, le total n'atteindrait pas 10 Go. Mais deux choses doivent attirer votre attention : vous pouvez rajouter des produits issus d'autres sources (nouveaux dépôts, installation manuelle, etc.) par la suite, et les répertoires /var et /tmp peuvent être amenés à grossir. Disposant d'assez de place, pourquoi ne pas réserver 20 ou 30 Go à la racine ? Partez par exemple sur 20 Go.

Il reste 98 Go.

3. Quelle place réserver au /home ?

La partition qui contient /home est celle qui contient vos données, ou celle des autres utilisateurs. C'est elle qui occupe le plus de place, entre les photos, la musique, les films, les documents de travail, etc. Réservez les 98 Go restants. Le disque est entièrement partitionné.

4. Est-il utile de créer une partition étendue ?

Vous avez trois partitions à créer sur le disque, en plus de celle qui existe déjà soit en tout quatre partitions. C'est le nombre exact pour prévoir quatre partitions primaires. Mais pensez que vous pouvez avoir besoin de réduire, supprimer ou recréer des partitions. Dans ce cas, la limite est déjà atteinte. Soyez prévoyant et créez une partition étendue où créer des partitions logiques.

5. Quel est le schéma final du disque ?

Partition primaire 1 : l'OS déjà présent, 40 Go.

Partition étendue :

- Partition logique 1 : /, 20 Go.
- Partition logique 2 : /home, 98 Go.
- Partition logique 3 : swap, 2 Go.

Les 160 Go sont tous occupés.

2. Gestion des RPM

Le but de ce TP est de travailler sur la base RPM des packages déjà installés sur votre poste et d'en installer de nouveaux. Le poste, ou une machine virtuelle, doit disposer d'une distribution basée sur RPM (Red Hat, Fedora, Mandriva, openSUSE, etc.).

1. Déterminez le nombre de packages RPM actuellement installés sur votre poste de travail.

Réponse :

Lancez la commande suivante :

```
$ rpm -qa | wc -l
```

Sur le poste de l'auteur elle retourne : 1684 !

- 2.** Vérifiez que le package coreutils est bien présent sur votre système, puis déterminez à quoi il sert à l'aide de sa description. Pouvez-vous faire en sorte de n'obtenir que la description et rien d'autre ? Lisez la page du manuel pour en savoir plus.

Dans un premier temps interrogez la base RPM sur ce package pour en obtenir les informations :

```
$ rpm -qi coreutils
```

En cas d'erreur, le package n'est sûrement pas installé. Sinon, lisez le contenu du champ Description.

Dans un second temps, lisez la section du manuel de rpm consacrée au format de sortie. Le paramètre -q accepte un format de sortie que vous pouvez formater avec --queryformat. Le format se spécifie ainsi : %{CHAMP} avec le champ en majuscules :

```
$ rpm -q --queryformat=%{DESCRIPTION}
```

- 3.** Essayez de supprimer le package coreutils. Pouvez-vous fournir la liste des dépendances qui vous en empêche ?

Tentez en tant que root :

```
# rpm -e coreutils
```

Vous obtenez la liste de tous les packages qui empêchent sa désinstallation : plusieurs centaines ! Notez l'existence du paramètre -R qui affiche de quoi dépend le package lui-même, et le --provides qui fournit le nom des éléments fournis par le package.

```
$ rpm -q --provides coreutils
fileutils
sh-utils
stat
textutils
coreutils = 6.9-43
```

- 4.** Téléchargez le package RPM de tuxpaint présent sur <http://www.tuxpaint.org/download/linux-rpm/>, choisissez la dernière version. Indiquée pour Fedora, elle fonctionne aussi sur Mandriva ou openSUSE. Installez ce package en indiquant le nom du package et une barre de progression.

Installez le package avec les paramètres -i, -v et -h :

```
# rpm -ivh tuxpaint.xxxxxxx.rpm
```

- 5.** Si le package était déjà installé, comment auriez-vous pu le mettre à jour ? Sachant qu'il est déjà installé maintenant, tentez de mettre à jour ce package de manière inconditionnelle. Dans quel cas cela peut-il être nécessaire ? Enfin, supprimez-le.

Vous pouvez mettre à jour le package avec les paramètres -U ou -F. Notez que vous auriez pu installer le package directement avec -U :

```
# rpm -Uvh tuxpaint.xxxxxxx.rpm
```

Simplement si le package est déjà installé dans la même version cela ne marche pas. Vous pourriez avoir besoin de le faire si des fichiers de ce package ont été supprimés : leur suppression, même complète, ne supprime pas le rpm de la base locale. Indiquez l'option --force.

```
# rpm -Uvh --force tuxpaint.xxxxxxx.rpm
```

3. Gestion de DPKG et APT

Le but de ce TP est de travailler sur la base dpkg des packages déjà installés sur votre poste, d'en installer de nouveaux et d'utiliser APT. Le poste, ou une machine virtuelle, doit disposer d'une distribution de type Debian ou Ubuntu.

1. Répondez aux questions 1 à 4 du TP précédent, mais avec les commandes et packages DPKG équivalents : coreutils est présent sous le même nom et tuxpaint dispose de packages Debian sur <http://www.tuxpaint.org/download/linux-debian/> (il est possible que vous deviez installer les packages additionnels fournis).

a - La liste des packages installés doit être filtrée. Par défaut dpkg fournit la liste de tous les paquets connus, dont ceux installés. Ils commencent par « ii » :

```
$ dpkg -l | grep ^ii | wc -l
```

Sur la machine de test de l'auteur il y a 695 packages installés.

b - L'option -l de dpkg peut prendre un filtre comme paramètre :

```
$ dpkg -l "coreutils"
```

Il est possible que vous trouviez deux packages de ce nom, aussi vous devrez soit lire les résultats, soit rechercher une correspondance exacte :

```
$ dpkg -l coreutils
```

Pour obtenir les détails du package déjà installé, il vous faut aller dans le manuel qui vous informe qu'il est possible d'utiliser la commande **dpkg-query** et le paramètre -W :

```
$ dpkg-query -W coreutils
```

Mais il manque la description. Le manuel de dpkg-query fournit une information supplémentaire : vous pouvez modifier le format de sortie avec le -f :

```
$ dpkg-query -W -f='${Description}' coreutils
```

c - Pour supprimer un package Debian, utilisez l'option -r :

```
# dpkg -r coreutils
```

Vous allez obtenir des erreurs :

```
dpkg : erreur de traitement de coreutils (--remove) :
  C'est un paquet indispensable - il ne doit pas être supprimé.
Des erreurs ont été rencontrées pendant l'exécution :
  coreutils
```

Pour savoir ce que fournit coreutils, utilisez à nouveau dpkg-query :

```
# dpkg-query -W -f='${Provides}' coreutils
textutils, shellutils, fileutils
```

d - Pour installer un package Debian, utilisez le paramètre -i :

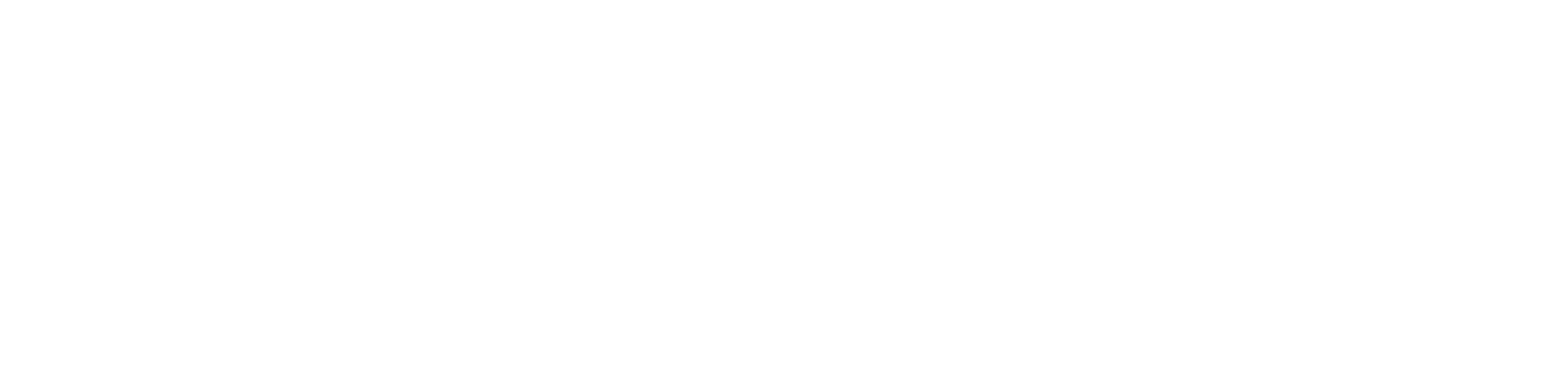
```
# dpkg -i tuxpaint.xxxxxxx.deb
```

Comme indiqué dans ce chapitre il n'y a pas de méthode directe équivalente à rpm pour la mise à jour d'un package. Si le package est déjà installé le -i va le mettre à jour. C'est à vous de vérifier avant si celui-ci est vraiment installé (voyez pour cela la réponse à la première question).

2. APT est un gestionnaire de meta-packages : il gère les dépendances à votre place et travaille sur des dépôts et non plus sur des packages individuels. Tuxpaint est présent dans le dépôt des backports. Le but est de gérer ce dépôt et les installations associées.

a - Rajoutez la ligne « deb <http://www.backports.org/debian> etch-backports main contrib non-free » dans le fichier des dépôts. Quelle est l'URL du dépôt contrib pour une architecture i386 ?

b - Mettez à jour la base de données locale de APT.



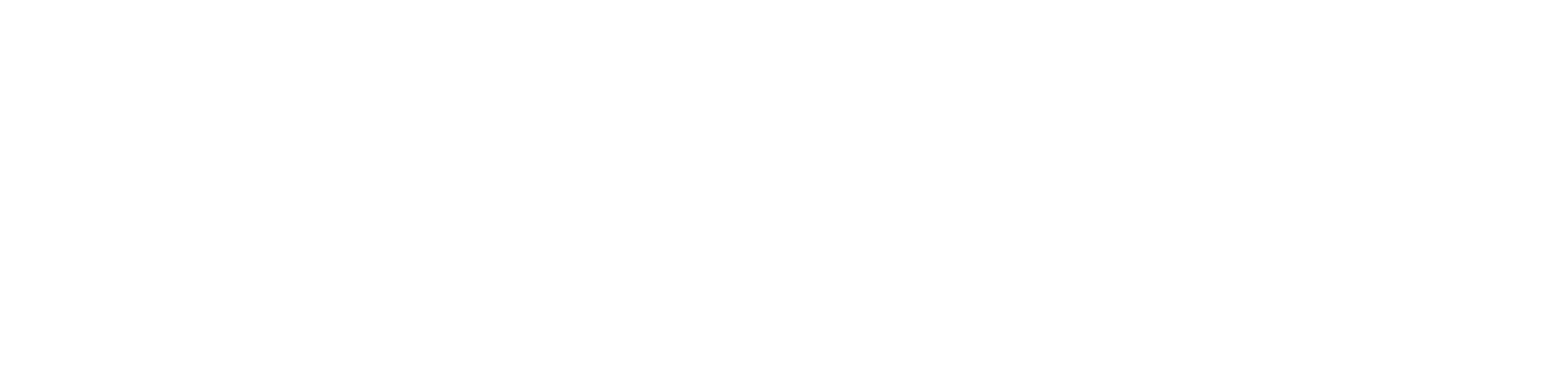
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



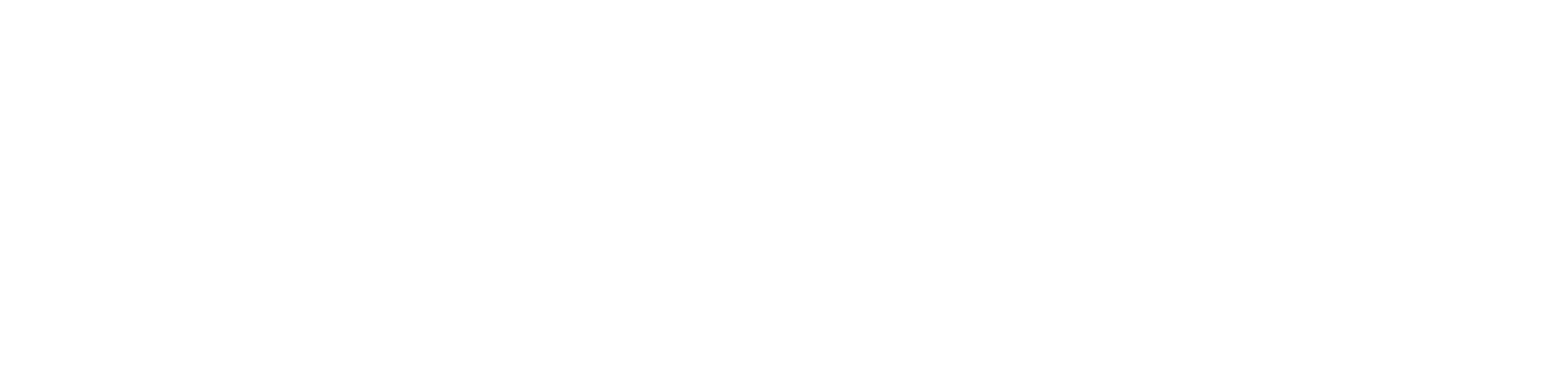
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

dur	70giga	150	30
disque	zip	12	30
disque	souple	10	30
ecran	15	150	20
ecran	17	300	20
ecran	19	500	20
ecran	21	500	20
clavier	105	45	30
clavier	115	55	30

Le fichier liste est l'original. Dans liste2, la deuxième ligne a été modifiée, une ligne écran a été ajoutée et les deux dernières lignes ont été supprimées.

```
$ diff liste liste2
2c2
< souris          optique 30      15
---
> souris          boutons 30      15
9a10
> ecran 21        500      20
12,13d12
< carte son        45      30
< carte video      145     30
```

- 2c2 : les lignes 2 de liste et liste2 doivent être échangées (elles doivent concorder soit en optique, soit en boutons).
- 9a10 : après la ligne 9 de liste (écran 19) il faut ajouter la ligne 10 (écran 21) de liste2.
- 12,13d12 : les lignes 12 et 13 de liste (carte son et vidéo) doivent être supprimés car elles n'existent pas après la ligne 12 de liste2.

b. cmp

La commande **cmp** compare les fichiers caractère par caractère. Par défaut la commande s'arrête dès la première différence rencontrée et indique la position de l'erreur.

`cmp [-l] [-s] fic1 fic2`

Le paramètre **-l** détaille toutes les différences en trois colonnes. La première colonne représente le numéro de caractère, la deuxième la valeur octale ASCII du caractère concerné de fic1 et la troisième la valeur octale ASCII du caractère concerné de fic2.

L'option **-s** retourne uniquement le code d'erreur (non visible), accessible par echo \$?.

```
$ cmp liste liste2
liste liste2 differ: char 38, line 2
$ cmp -l liste liste2
 38 157 142
 39 160 157
 40 164 165
 41 151 164
 42 161 157
 43 165 156
 44 145 163
182 143 145
183 154 143
...
```

14. Délai d'attente

La commande **sleep** permet d'attendre le nombre de secondes indiqués. Le script est interrompu durant ce temps. Le nombre de secondes est un entier compris entre 0 et 4 milliards (136 ans).

```
$ sleep 10
```

G. Les processus

1. Définition et environnement

Un **processus** représente à la fois un programme en cours d'exécution et tout son environnement d'exécution (mémoire, état, identification, propriétaire, père...).

Voici une liste des données d'identification d'un processus :

- **Un numéro de processus unique PID** (Process ID) : chaque processus Unix est numéroté afin de pouvoir être différencié des autres. Le premier processus lancé par le système est 1 et il s'agit d'un processus appelé généralement init. On utilise le PID quand on travaille avec un processus. Lancer 10 fois le même programme (même nom) produit 10 PID différents.
- **Un numéro de processus parent PPID** (Parent Process ID) : chaque processus peut lui-même lancer d'autres processus, des processus enfants (child process). Chaque enfant reçoit parmi les informations le PID du processus père qui l'a lancé. Tous les processus ont un PPID sauf le processus 0 qui est un pseudo-processus représentant le démarrage du système (créé le 1 init).
- **Un numéro d'utilisateur et un numéro de groupe** : correspond à l'UID et au GID de l'utilisateur qui a lancé le processus. C'est nécessaire pour que le système sache si le processus a le droit d'accéder à certaines ressources ou non. Les processus enfants héritent de ces informations. Dans certains cas (que nous verrons plus tard) on peut modifier cet état.
- **Durée de traitement et priorité** : la durée de traitement correspond au temps d'exécution écoulé depuis le dernier réveil du processus. Dans un environnement multitâche, le temps d'exécution est partagé entre les divers processus, et tous ne possèdent pas la même priorité. Les processus de plus haute priorité sont traités en premier. Lorsqu'un processus est inactif, sa priorité augmente afin d'avoir une chance d'être exécuté. Lorsqu'il est actif, sa priorité baisse afin de laisser sa place à un autre. C'est l'ordonnanceur de tâches du système qui gère les priorités et les temps d'exécution.
- **Répertoire de travail actif** : à son lancement, le répertoire courant (celui depuis lequel le processus a été lancé) est transmis au processus. C'est ce répertoire qui servira de base pour les chemins relatifs.
- **Fichiers ouverts** : table des descripteurs des fichiers ouverts. Par défaut au début seuls trois sont présents : 0, 1 et 2 (les canaux standards). À chaque ouverture de fichier ou de nouveau canal, la table se remplit. À la fermeture du processus, les descripteurs sont fermés (en principe).
- On trouve d'autres informations comme la taille de la mémoire allouée, la date de lancement du processus, le terminal d'attachement, l'état du processus, les UID effectif et réel ainsi que les GID effectif et réel.

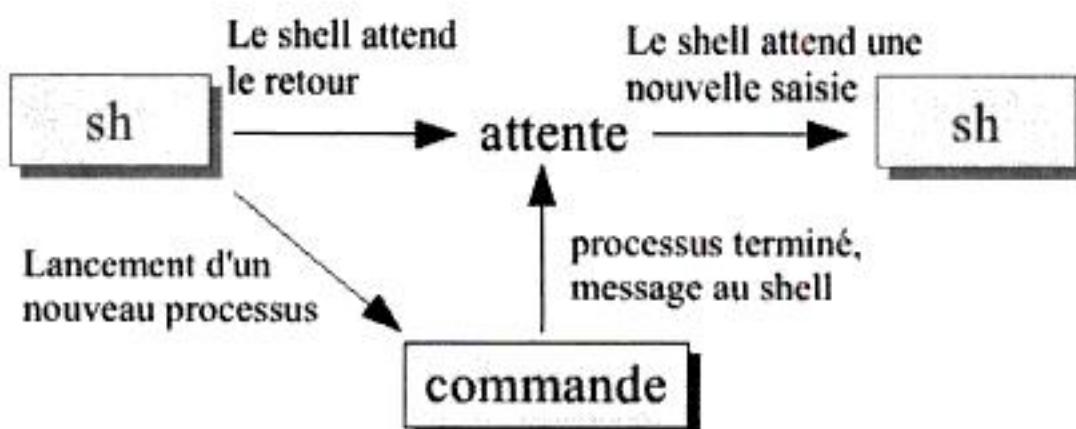
2. États d'un processus

Durant sa vie (temps entre le lancement et la sortie) un processus peut passer par divers états ou **process state** :

- exécution en mode utilisateur (**user mode**) ;
- exécution en mode noyau (**kernel mode**) ;
- en attente E/S (**waiting**) ;
- endormi (**sleeping**) ;
- prêt à l'exécution (**runnable**) ;
- endormi dans le swap (**mémoire virtuelle**) ;

- nouveau processus ;
- fin de processus (**zombie**).

3. Lancement en tâche de fond



En suivant ce que vous avez vu avant, le système étant multitâches un certain nombre de processus tournent déjà sur la machine sans que vous les voyiez. De même le shell que vous utilisez est lui-même un processus. Quand une commande est saisie, le shell crée un nouveau processus pour l'exécuter, ce processus devient un processus enfant du shell. Jusqu'à présent il fallait attendre la fin de l'exécution d'une commande pour en saisir une autre (sauf en utilisant « ; » pour chaîner les commandes).

Rien n'empêche le shell d'attendre le message du processus terminé pour rendre la main : de ce fait la commande une fois lancée, le shell peut autoriser la saisie d'une nouvelle commande sans attendre la fin de l'exécution de la commande précédente. Pour cela il suffit de saisir, après avoir tapé la commande, le **ET Commercial** « & ». Dans ce cas, le shell et la commande lancée fonctionneront en parallèle.

```
$ ls -R / > ls.txt 2>/dev/null &
[1] 21976
$
[1] Done
$ ls
fic1      fic3      liste      ls.txt      rep1      users
fic2      fic4      liste2     mypass      toto.tar.gz
```

Juste après la saisie un chiffre apparaît, il est à retenir car il s'agit du PID du nouveau processus lancé. Après une autre saisie une ligne Done indique que le traitement est terminé. La valeur [1] est propre à un shell particulier (ksh).

Quelques remarques sur l'utilisation du lancement en tâche de fond :

- Le processus lancé ne devrait pas attendre de saisie au risque de confusion entre cette commande et le shell lui-même.
- Le processus lancé ne devrait pas afficher de résultats sur l'écran au risque d'avoir des affichages en conflit avec celui du shell (par exemple apparition d'une ligne en milieu de saisie).
- Enfin, quand on quitte le shell, on quitte aussi tous ses fils : dans ce cas ne pas quitter le shell pendant un traitement important.

4. Background, foreground, jobs

Vous pouvez récupérer la main sous le shell si vous avez lancé un processus au premier plan. Vous pouvez le stopper temporairement en tapant [Ctrl] Z.

```
$ sleep 100
<CTRL+Z>
[1]+  Stopped                  sleep 100
```

Le processus est stoppé : son exécution est suspendue jusqu'à ce que vous le replaciez au premier plan avec la commande **fg** (*foreground*) :

```
$ fg
Sleep 100
```

Quand vous lancez une commande, vous avez remarqué le premier nombre entre crochets, c'est le numéro de job. Vous pouvez en obtenir la liste avec la commande **jobs**.

```
$ jobs
[1]- Stopped sleep 100
[2]+ Stopped sleep 100
```

Les commandes **bg** et **fg** permettent d'agir sur ces jobs en prenant comme paramètre leur numéro. La commande **bg** est exécutée sur un job stoppé pour le relancer en arrière-plan. Le job 2 est relancé en arrière-plan :

```
$ bg 2
[2]+ sleep 100 &
$
[2]+ Done sleep 100
```

5. Liste des processus

La commande **ps** (*process status*) permet d'avoir des informations sur les processus en cours. Lancée seule, elle n'affiche que les processus en cours lancés par l'utilisateur et depuis la console actuelle.

```
$ ps
 PID TTY      TIME CMD
 4334 pts/1    00:00:00 bash
 5017 pts/1    00:00:00 ps
```

Pour avoir plus d'informations, utilisez le paramètre **-f**.

```
ps -f
UID      PID  PPID  C STIME TTY      TIME CMD
seb      4334 24449  0 09:46 pts/1    00:00:00 /bin/bash
seb      5024  4334  0 10:51 pts/1    00:00:00 ps -f
```

Le paramètre **-e** donne des informations sur tous les processus en cours.

```
$ ps -ef
UID      PID  PPID  C STIME TTY      TIME CMD
...
seb      26431     1  0 Mar04 ?
seb      26436 26322  0 Mar04 ?
seb      26438     1  0 Mar04 ?
seb      26439 26424  0 Mar04 ?
seb      26441     1  0 Mar04 ?
seb      26443     1  0 Mar04 ?
seb      26453     1  0 Mar04 ?
seb      26454 26424  0 Mar04 ?
seb      26465 26424  0 Mar04 ?
seb      26467     1  0 Mar04 ?
seb      26474     1  0 Mar04 ?
seb      26485     1  0 Mar04 ?
...
00:00:30 kded [kdeinit]
00:00:03 kwrapper ksmserver
00:00:00 ksmserver [kdeinit]
00:00:50 kwin [kdeinit]
00:00:28 kdesktop [kdeinit]
00:03:40 kicker [kdeinit]
00:00:00 kerry [kdeinit]
00:00:01 evolution
00:00:11 kde-window-decorator
00:00:02 gconfd-2 12
00:00:01 knotify [kdeinit]
00:03:06 beagled
```



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

communs d'un Unix à l'autre, les numéros ne le sont pas forcément. L'option **-l** permet d'obtenir la liste des signaux.

Signal	Rôle
1 (SIGHUP)	Hang Up, est envoyé par le père à tous ses enfants lorsqu'il se termine.
2 (SIGINT)	Interruption du processus demandé (touche [Suppr], [Ctrl] C).
3 (SIGQUIT)	Idem SIGINT mais génération d'un Core Dump (fichier de déboggage).
9 (SIGKILL)	Signal ne pouvant être ignoré, force le processus à finir 'brutalement'.
15 (SIGTERM)	Signal envoyé par défaut par la commande kill . Demande au processus de se terminer normalement.

```
$ sleep 100&
[1] 5187
$ kill 5187
$
[1]+  Complété                 sleep 100
$ sleep 100&
[1] 5194
$ kill -9 5194
[1]+  Processus arrêté         sleep 100
```

7. nohup

Quand le shell est quitté (`exit`, `[Ctrl] D...`) le signal 1 **SIGHUP** est envoyé aux enfants pour qu'ils se terminent aussi. Lorsqu'un traitement long est lancé en tâche de fond et que l'utilisateur veut quitter le shell, ce traitement sera alors arrêté et il faudra tout recommencer. Le moyen d'éviter cela est de lancer le traitement (processus) avec la commande **nohup**. Dans ce cas le processus lancé ne réagira plus au signal **SIGHUP**, et donc le shell pourra être quitté, la commande continuera son exécution.

Par défaut les canaux de sortie et d'erreur standards sont redirigés vers un fichier `nohup.out`, sauf si la redirection est explicitement précisée.

```
$ nohup ls -lR / &
10206
$ Sending output to nohup.out
```

☞ Quand un fils se termine, il envoie le signal **SIGCHLD** à son père. Sauf cas prévu (le père se détache du fils) le père doit obtenir autant de **SIGCHLD** qu'il a eu de fils ou émis de **SIGHUP**. Si le père se termine avant que les fils se terminent ceux-ci deviennent des zombis : le signal **SIGCHLD** n'est pas reçu... Le processus fils est bien terminé, il est mort, il ne consomme aucune ressource. Il ne peut donc être tué (puisque il est mort) mais continue à occuper une entrée dans la table des processus. C'est `init` qui le récupère, et `init` étant toujours en attente, le zombie peut finir par disparaître.

8. nice et renice

La commande **nice** permet de lancer une commande avec une priorité plus faible, afin de permettre éventuellement à d'autres processus de tourner plus rapidement.

```
nice [-valeur] commande [arguments]
```

Une valeur positive causera une baisse de priorité, une négative l'augmentation de la priorité (si autorisé). La valeur doit être comprise entre -20 et 20. Plus la valeur est élevée et plus le traitement est ralenti.

```
$ nice -10 ls -lR / >liste 2>/dev/null&
10884
$ ps -1
  F S      UID      PID    PPID     C PRI  NI ADDR SZ WCHAN TTY
TIME CMD
80808001 U N+    75 10884 10822 28.5 54 10          0 848K aa3b3a9c tttyp4
0:03.32 ls
```

La commande **renice** fonctionne un peu comme nice mais elle permet de modifier la priorité en fonction d'un utilisateur, d'un groupe ou d'un PID. La commande visée doit donc déjà tourner.

renice [-n prio] [-p] [-g] [-u] ID

La priorité doit être comprise entre -20 et 20. L'utilisateur standard ne peut utiliser que les valeurs entre 0 et 20 permettant de baisser la priorité. L'option -p précise un PID, -g un GID et -u un UID.

9. time

La commande **time** mesure les durées d'exécution d'une commande, idéal pour connaître les temps de traitement, et retourne trois valeurs :

- **real** : durée totale d'exécution de la commande ;
- **user** : durée du temps CPU nécessaire pour exécuter le programme ;
- **system** : durée du temps CPU nécessaire pour exécuter les commandes liées à l'OS (appels système au sein d'un programme).

Le résultat est sorti par le canal d'erreur standard 2. On peut avoir une indication de la charge de la machine par le calcul real / (user+system). Si le résultat est inférieur à 10, la machine dispose de bonnes performances, au-delà de 20 la charge de la machine est trop lourde (performances basses).

```
$ time ls -lR /home
...
real    4.8
user    0.2
sys     0.5
```

H. Plus loin avec le bash

1. Alias

Un alias est un raccourci d'une commande avec d'éventuels paramètres. Il se définit avec la commande **alias**. Utilisée seule elle liste les alias disponibles.

```
$ alias
alias ..='cd ..'
alias ...='cd .../...'
alias cd..='cd ..'
alias dir='ls -l'
alias l='ls -alF'
alias la='ls -la'
alias ll='ls -l'
alias ls='ls $LS_OPTIONS'
alias ls-l='ls -l'
alias md='mkdir -p'
alias o='less'
alias rd='rmdir'
...
```

Vous pouvez créer vos propres alias.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

```
SSH_AGENT_PID=26377
HOSTNAME=p64p17bicb3
DM_CONTROL=/var/run/xdmctl
XKEYSYMDB=/usr/share/X11/XKeysymDB
HOST=p64p17bicb3
SHELL=/bin/bash
TERM=xterm
PROFILEREAD=true
HISTSIZE=1000
...
```

Une variable peut contenir des caractères spéciaux, le principal étant l'espace. L'exemple suivant ne fonctionne pas :

```
$ c=Salut les copains
les: not found
$ echo $c
```

Pour cela il faut soit verrouiller les caractères spéciaux un par un, soit les mettre entre guillemets ou apostrophes.

```
c=Salut\ les\ Copains # Solution lourde
c="Salut les copains" # Solution correcte
c='Salut les copains' # Solution correcte
```

La principale différence entre les guillemets et les apostrophes est l'interprétation des variables et des substitutions. " et ' se verrouillent mutuellement.

```
$ a=Jules
$ b=Cesar
$ c="$a $b a conquis la Gaule"
$ d='$a $b a conquis la Gaule'
$ echo $c
Jules Cesar a conquis la Gaule
$ echo $d
$a $b a conquis la Gaule
$ echo "Linux c'est top"
Linux c'est top
$ echo 'Linux "trop bien"'
Linux "trop bien"
```

4. Suppression et protection

Vous supprimez une variable avec la commande **unset**. Vous protégez une variable en écriture et contre sa suppression avec la commande **readonly**. Une variable en lecture seule, même vide, est figée. Il n'existe aucun moyen de la replacer en écriture et de la supprimer, sauf en quittant le shell.

```
$ a=Jules
$ b=Cesar
$ echo $a $b
Jules Cesar
$ unset b
$ echo $a $b
Jules
$ readonly a
$ a=Neron
a: is read only
$ unset a
a: is read only
```

5. Export

Par défaut une variable n'est accessible que depuis le shell où elle a été définie. La variable **a** est déclarée sous l'invite du shell courant puis est affichée par un script lancé depuis ce même shell. Ce dernier ne connaît pas la variable **a** : rien ne s'affiche.

```
$ a=Jules
$ echo 'echo "a=$a"' > voir_a.sh
$ chmod u+x voir_a.sh
$ ./voir_a.sh
a=
```

La commande **export** permet d'exporter une variable de manière à ce que son contenu soit visible par les scripts et autres sous-shells. Les variables exportées peuvent être modifiées dans le script, mais ces modifications ne s'appliquent qu'au script ou au sous-shell. Cette fois le premier script peut accéder à la variable **a** exportée. Mais les modifications restent locales au script. Une fois celui-ci terminé la modification disparaît.

```
$ export a
$ ./voir_a.sh
a=Jules
$ echo 'a=Neron ; echo "a=$a"' > voir_a.sh
$ ./voir_a.sh
a=Jules
a=Neron
$ echo $a
Jules
```

6. Accolades

Les accolades de base {} permettent d'identifier le nom d'une variable. Imaginez la variable **fichier** contenant le nom de fichier 'liste'. Vous voulez copier **liste1** en **liste2**.

```
$ fichier=liste
$ cp $fichier1 $fichier2
cp: opérande fichier manquant
Pour en savoir davantage, faites: « cp --help ».
```

Cela ne fonctionne pas car ce n'est pas \$fichier qui est interprété mais \$fichier1 et \$fichier2 qui n'existent pas.

```
$ cp ${fichier}2 ${fichier}1
```

Dans ce cas, cette ligne équivaut à :

```
$ cp liste2 liste1
```

7. Accolades et remplacement conditionnel

Les accolades disposent d'une syntaxe particulière.

```
{variable:Remplacement}
```

Selon la valeur ou la présence de la variable, il est possible de remplacer sa valeur par une autre.

Remplacement	Signification
{x:-texte}	Si la variable x est vide ou inexistante, le texte prendra sa place. Sinon c'est le contenu de la variable qui prévaudra.
{x:=texte}	Si la variable x est vide ou inexistante, le texte prendra sa place et deviendra la valeur de la variable.
{x:+texte}	Si la variable x est définie et non vide, le texte prendra sa place. Dans le cas contraire une chaîne vide prend sa place.

Remplacement	Signification
{x:?texte}	Si la variable x est vide ou inexiste, le script est interrompu et le message texte s'affiche. Si texte est absent un message d'erreur standard est affiché.
\$ echo \$nom \$ echo \${nom:-Jean} Jean \$ echo \$nom \$ echo \${nom:=Jean} Jean \$ echo \$nom Jean \$ echo \${nom:+Valeur définie} Valeur définie \$ unset nom \$ echo \${nom:?Variable absente ou non définie} nom: Variable absente ou non définie \$ nom=Jean \$ echo \${nom:?Variable absente ou non définie} Jean	

8. Variables système

En plus des variables que l'utilisateur peut définir lui-même, le shell est lancé avec un certain nombre de variables prédéfinies utiles pour certaines commandes et accessibles par l'utilisateur. Le contenu de ces variables système peut être modifié mais il faut alors faire attention car certaines ont une incidence directe sur le comportement du système.

Variable	Contenu
HOME	Chemin d'accès du répertoire utilisateur. Répertoire par défaut en cas d'utilisation de CD.
PATH	Liste de répertoires, séparés par des : où le shell va rechercher les commandes externes et autres scripts et binaires. La recherche se fait dans l'ordre des répertoires saisis.
PS1	Prompt String 1, chaîne représentant le prompt standard affiché à l'écran par le shell en attente de saisie de commande.
PS2	Prompt String 2, chaîne représentant un prompt secondaire au cas où la saisie doit être complétée.
IFS	Internal Field Separator, liste des caractères séparant les mots dans une ligne de commande. Par défaut il s'agit de l'espace, de la tabulation et du saut de ligne.
MAIL	Chemin et fichier contenant les messages de l'utilisateur.
SHELL	Chemin complet du shell en cours d'exécution.
LANG	Définition de la langue à utiliser ainsi que du jeu de caractères.
USER	Nom de l'utilisateur en cours.
LOGNAME	Nom du login utilisé lors de la connexion.
HISTFILE	Nom du fichier historique, généralement \$HOME/.sh_history.
HISTSIZE	Taille en nombre de lignes de l'historique.
OLDPWD	Chemin d'accès du répertoire accédé précédemment.
PS3	Définit l'invite de saisie pour un select.

Variable	Contenu
PWD	Chemin d'accès courant.
RANDOM	Génère et contient un nombre aléatoire entre 0 et 32767.

9. Variables spéciales

Il s'agit de variables accessibles uniquement en lecture et dont le contenu est généralement contextuel.

Variable	Contenu
\$?	Code retour de la dernière commande exécutée.
\$\$	PID du shell actif.
\$!	PID du dernier processus lancé en arrière-plan.
\$-	Les options du shell.

```
$ echo $$
23496
$ grep memoire liste
$ echo $?
1
$ grep souris liste
souris optique 30      15
$ echo $?
0
$ ls -lR >toto.txt 2<&1 &
26675
$ echo $!
26675
```

10. Longueur d'une chaîne

Il est possible d'obtenir la longueur d'une chaîne avec le caractère #.

```
$ a=Jules
$ echo "Longueur de $a : ${#a}"
Longueur de Jules : 5
```

11. Tableaux et champs

Deux moyens sont disponibles pour déclarer un tableau, l'un avec l'utilisation des crochets [], l'autre avec la création globale. Le premier élément est 0 le dernier 1023. Pour accéder au contenu du tableau il faut mettre la variable ET l'élément entre accolades {}.

```
$ Nom[0]="Jules"
$ Nom[1]="Romain"
$ Nom[2]="Francois"
$ echo ${Nom[1]}
Romain
```

ou :

```
$ Nom=(Jules Romain Francois)
$ echo ${nom[2]}
Francois
```

Pour lister tous les éléments :

```
$ echo ${Nom[*]}
Jules Romain Francois
```

Pour connaître le nombre d'éléments :

```
$ echo ${#Nom[*]}
3
```

Si l'index est une variable, on ne met pas le \$ devant celui-ci :

```
$ idx=0
$ echo ${Nom[idx]}
Jules
```

12. Variables typées

Les variables peuvent être typées en entier (integer) avec la commande **typeset -i** le permet. L'avantage est qu'il devient possible d'effectuer des calculs et des comparaisons sans passer par expr. La commande **let** ou **((...))** permet des calculs sur variables.

Opérateur	Rôle
+ - * /	Opérations simples
%	Modulo
< > <= >=	Comparaisons, 1 si vraie, 0 si faux
== !=	Égal ou différent
&&	Comparaisons liées par un opérateur logique
& ^	Logique binaire AND OR XOR

```
$ typeset -i resultat
$ resultat=6*7
$ echo $resultat
42
$ resultat=Erreur
ksh: Erreur: bad number
$ resultat=resultat*3
126
$ typeset -i add
$ add=5
$ let resultat=add+5 resultat=resultat*add
$ echo $resultat
50
```

J. Configuration de bash

1. Fichiers de configuration

Le shell bash peut être lancé dans plusieurs modes :

- le shell interactif de connexion (login shell) ;
- le shell interactif simple ;
- le shell non interactif ;
- le mode sh ;
- etc.

Selon son mode de démarrage, le shell va chercher et exécuter divers scripts et fichiers de configuration. Un fichier de configuration est un script shell, une séquence de commandes individuelles ayant pour but de configurer l'environnement de l'utilisateur.

a. Shell de connexion

Le shell de connexion est lancé après la saisie du login et du mot de passe sur la console. C'est celui précisé à la fin de chaque ligne de /etc/passwd. Dans ce mode, le shell cherche à exécuter, dans cet ordre et s'ils sont présents :

- /etc/profile
- ~/.bash_profile
- ~/.bash_login
- ~/.profile

À la déconnexion, il tente d'exécuter :

- - ~/.bash_logout

b. Shell simple

Le shell interactif simple correspond à l'exécution du bash dans une fenêtre (xterm, konsole), une console ou à la main (taper bash dans une console). Dans ce cas seul le fichier suivant sera exécuté s'il existe :

- ~/.bashrc

Notez que dans beaucoup de distributions Linux, le .bashrc est appelé soit par .bash_profile, soit par /etc/profile, et que la configuration est donc placée dans .bash_profile qui sera alors toujours appelé.

c. Mode Bourne shell

Lorsque bash est lancé en mode Bourne Shell via la commande **sh**, en shell de connexion ou non, il tente d'exécuter les fichiers dans cet ordre :

- /etc/profile
- ~/.profile

d. Mode non interactif

Le shell peut être lancé en mode non interactif. C'est généralement le cas lorsque vous exécutez un script. Dans ce cas il n'y a aucun script exécuté par défaut au démarrage sauf si vous précisez une variable appelée **BASH_ENV** qui contient le chemin d'un script. Dans ce cas bash charge et exécute ce fichier avant le début de l'exécution du script ou de la commande.

2. Commandes set

Le shell dispose de dizaines d'options, dont la plupart peuvent être paramétrées à l'aide de la commande **set**. Celles qui suivent ne sont qu'une simple sélection. Le **-** avant l'option permet de passer celle-ci à On. Un **+** passe l'option à off.

Option	Résultat
-a / -o allexport	Toutes les variables seront automatiquement exportées.
-u / -o nounset	Par défaut le shell traite les variables inexistantes comme des chaînes vides. Cette option produit une erreur si la variable n'existe pas.
-x / -o xtrace	Affiche toutes les commandes au fur et à mesure de leur exécution : idéal en début de script pour débogage.
-o vi	Manipulation de la ligne de commande avec la syntaxe de vi.
-o emacs	Manipulation de la ligne de commande avec la syntaxe de emacs.
-C / -o noclobber	Interdit les redirections en sortie si le fichier existe déjà.
history	Autorise la gestion de l'historique.

Le manuel du shell vous fournira toutes les options possibles.

K. Programmation shell

1. Structure et exécution d'un script

Le shell n'est pas qu'un simple interpréteur de commandes, mais dispose d'un véritable langage de programmation avec notamment une gestion des variables, des tests et des boucles, des opérations sur les variables, des fonctions...

Toutes les instructions et commandes sont regroupées au sein d'un script. Lors de son exécution, chaque ligne sera lue une à une et exécutée. Une ligne peut se composer de commandes internes ou externes, de commentaires ou être vide. Plusieurs instructions par lignes sont possibles, séparées par le ; ou liées conditionnellement par && ou ||. Le ; est l'équivalent d'un saut de ligne.

Par convention les noms des shell scripts se terminent généralement (pas obligatoirement) par « .sh » pour le Bourne Shell et le Bourne Again Shell, par « .ksh » pour le Korn Shell et par « .csh » pour le C Shell.

Pour rendre un script exécutable directement :

```
$ chmod u+x monscript
```

Pour l'exécuter :

```
$ ./monscript
```

Pour éviter le ./ :

```
$ PATH=$PATH:.
$ monscript
```

Notez que le point est placé en dernier dans le PATH. Le mettre en premier peut présenter un risque pour la sécurité : une nouvelle commande ls modifiée est placée dans votre répertoire.

Quand un script est lancé, un nouveau shell fils est créé qui va exécuter chacune des commandes. Si c'est une commande interne, elle est directement exécutée par le nouveau shell. Si c'est une commande externe, dans le cas d'un binaire un nouveau fils sera créé pour l'exécuter, dans le cas d'un shell script un nouveau shell fils est lancé pour lire ce nouveau shell ligne à ligne.

Une ligne de commentaire commence toujours par le caractère #. Un commentaire peut être placé en fin d'une ligne comportant déjà des commandes.

```
# La ligne suivante effectue un ls
ls # La ligne en question
```

La première ligne a une importance particulière car elle permet de préciser quel shell va exécuter le script :

```
#!/bin/bash
#!/bin/ksh
```

Dans le premier cas c'est un script Bourne Again, dans l'autre un script Korn.

2. Arguments d'un script

a. Paramètres de position

Les paramètres de position sont aussi des variables spéciales utilisées lors d'un passage de paramètres à un script.

Variable	Contenu
\$0	Nom de la commande (du script).
\$1-9	\$1,\$2,\$3... Les neuf premiers paramètres passés au script.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

```

echo "Parametres : 1=$1 2=$2 3=$3 4=$4"
echo "Liste : $*"

$ ./param2.sh riri fifi loulou donald picsou
Avant :
Nombre de parametres : 5
Parametres : 1=riri 2=fifi 3=loulou 4=donald
Liste : riri fifi loulou donald picsou
apres set alfred oscar romeo zoulou
Nombre de parametres : 4
Parametres : 1=alfred 2=oscar 3=romeo 4=zoulou
Liste : alfred oscar romeo zoulou

```

c. Réorganisation des paramètres

La commande **shift** est la dernière commande permettant de modifier la structure des paramètres de position. Un simple appel décale tous les paramètres d'une position en supprimant le premier : \$2 devient \$1, \$3 devient \$2 et ainsi de suite. Le \$1 original disparaît. \$# , \$* et \$@ sont redéfinis en conséquence.

La commande **shift** suivie d'une valeur n effectue un décalage de n éléments. Ainsi avec shift 4, \$5 devient \$1, \$6 devient \$2, ...

```

$ cat param3.sh
#!/bin/bash
set alfred oscar romeo zoulou
echo "set alfred oscar romeo zoulou"
echo "Nombre de parametres : $#"
echo "Parametres : 1=$1 2=$2 3=$3 4=$4"
echo "Liste : $*"
shift
echo "Après un shift"
echo "Nombre de parametres : $#"
echo "Parametres : 1=$1 2=$2 3=$3 4=$4"
echo "Liste : $*"

$ ./param3.sh
set alfred oscar romeo zoulou
Nombre de parametres : 4
Parametres : 1=alfred 2=oscar 3=romeo 4=zoulou
Liste : alfred oscar romeo zoulou
Après un shift
Nombre de parametres : 3
Parametres : 1=oscar 2=romeo 3=zoulou 4=
Liste : oscar romeo zoulou

```

d. Sortie de script

La commande **exit** permet de mettre fin à un script. Par défaut la valeur renvoyée est 0 (pas d'erreur) mais n'importe quelle autre valeur de 0 à 255 peut être précisée. Vous récupérez la valeur de sortie par la variable \$?.

```
$ exit 1
```

3. Environnement du processus

En principe seules les variables exportées sont accessibles par un processus fils. Si vous souhaitez visualiser l'environnement lié à un fils (dans un script par exemple) utilisez la commande **env**.

```
$ env
LESSKEY=/etc/lesskey.bin
NNTPSERVER=news
INFODIR=/usr/local/info:/usr/share/info:/usr/info
MANPATH=/usr/local/man:/usr/share/man
KDE_MULTIHEAD=false
SSH_AGENT_PID=28012
HOSTNAME=slyserver
DM_CONTROL=/var/run/xdmctl
XKEYSYMDB=/usr/share/X11/XKeysymDB
HOST=p64p17bicb3
SHELL=/bin/bash
TERM=xterm
PROFILEREAD=true
HISTSIZE=1000
XDM_MANAGED=/var/run/xdmctl/xdmctl-
:0,maysd,mayfn,sched,rsvd,method=classic
XDG_SESSION_COOKIE=16af07a56781b4689718210047060300-
1211264847.394692-546885666
TMPDIR=/tmp
GTK2_RC_FILES=/etc/gtk-2.0/gtkrc:/usr/share/themes//QtCurve/gtk-
2.0/gtkrc:/home/seb/.gtkrc-2.0-qtengine:/home/seb/.gtkrc-
2.0:/home/seb/.kde/share/config/gtkrc-2.0
KDE_NO_IPV6=1
GTK_RC_FILES=/etc/gtk/gtkrc:/home/seb/.gtkrc:/home/seb/.kde/share/co
nfig/gtkrc
GS_LIB=/home/seb/.fonts
WINDOWID=71303176
MORE=-sl
QTDIR=/usr/lib/qt3
XSESSION_IS_UP=yes
KDE_FULL_SESSION=true
GROFF_NO_SGR=yes
JRE_HOME=/usr/lib/jvm/jre
USER=seb
...
```

La commande **env** permet de redéfinir aussi l'environnement du processus à lancer. Cela peut être utile lorsque le script doit accéder à une variable qui n'est pas présente dans l'environnement du père, ou qu'on ne souhaite pas exporter. La syntaxe est :

```
env var1=valeur var2=valeur ... commande
```

Dans le cas de bash, **env** n'est pas indispensable.

```
var1=valeur var2=valeur ... commande
```

Si la première option est le signe - alors c'est tout l'environnement existant qui est supprimé pour être remplacé par les nouvelles variables et valeurs.

```
$ unset a
$ ./voir_a.sh
a=
$ env a=jojo ./voir_a.sh
a=jojo
$ echo a=$a
a=
```

4. Substitution de commande

Le mécanisme de substitution permet de placer le résultat de commandes simples ou complexes dans une variable. Vous placez les commandes à exécuter entre des accents graves « ` » ([Alt Gr] 7).

```
$ mon_unix=`uname`
$ echo ${mon_unix}
Linux
$ machine=`uname -a | cut -d" " -f5`
echo $machine
SMP
```

Attention, seul le canal de sortie standard est affecté à la variable. Le canal d'erreur standard sort toujours vers l'écran dans ce cas.

Les accents graves ne sont pas toujours idéaux pour ces manipulations. En effet si vous travaillez à plusieurs niveaux, vous devez verrouiller ceux qui sont à l'intérieur des premiers niveaux. Aussi le bash permet d'utiliser à la place la syntaxe \${...} qui n'a pas ce problème.

```
$ mon_unix=$(uname)
$ echo ${mon_unix}
Linux
$ machine=$(uname -a | cut -d" " -f5)
echo $machine
SMP
```

Ne confondez pas les accolades et les parenthèses. Les premières isolent les variables, les secondes effectuent la substitution des commandes.

5. Tests de conditions

La commande **test** permet d'effectuer des tests de conditions. Le résultat est récupérable par la variable \$? (code retour). Si ce résultat est 0 alors la condition est réalisée.

a. Tests sur une chaîne

- **test -z "variable"** : zero, retour OK si la variable est vide (ex : test -z "\$a").
- **test -n "variable"** : non zero, retour OK si la variable n'est pas vide (texte quelconque).
- **test "variable" = chaîne** : OK si les deux chaînes sont identiques.
- **test "variable" != chaîne** : OK si les deux chaînes sont différentes.

```
$ a=
$ test -z "$a" ; echo $?
0
$ test -n "$a" ; echo $?
1
$ a=Jules
$ test "$a" = Jules ; echo $?
0
```

Attention à bien placer vos variables contenant du texte entre guillemets. Dans le cas contraire un bug se produira si la variable est vide :

```
$ a=
$ b=toto
$ [ $a = $b ] && echo "ok"
bash: [: =: unary operator expected
```

Alors que :

```
[ "$a" = "$b" ] && echo "ok"
```

ne produit pas d'erreur.

b. Tests sur les valeurs numériques

Les chaînes à comparer sont converties en valeurs numériques. Bash ne gère que des valeurs entières. La syntaxe est `test valeur1 option valeur2` et les options sont les suivantes :

Option	Rôle
-eq	Equal : égal
-ne	Not Equal : différent
-lt	Less than : inférieur
-gt	Greater than : supérieur
-le	Less or equal : inférieur ou égal
-ge	Greater or equal : supérieur ou égal

```
$ a=10
$ b=20
$ test "$a" -ne "$b" ; echo $?
0
$ test "$a" -ge "$b" ; echo $?
1
$ test "$a" -lt "$b" && echo "$a est inferieur a $b"
10 est inferieur a 20
```

c. Tests sur les fichiers

La syntaxe est `test option nom_fichier` et les options sont les suivantes :

Option	Rôle
-f	Fichier normal.
-d	Un répertoire.
-c	Fichier en mode caractère.
-b	Fichier en mode bloc.
-p	Tube nommé (named pipe).
-r	Autorisation en lecture.
-w	Autorisation en écriture.
-x	Autorisation en exécution.
-s	Fichier non vide (au moins un caractère).
-e	Le fichier existe.
-L	Le fichier est un lien symbolique .
-u	Le fichier existe, SUID-Bit positionné.
-g	Le fichier existe SGID-Bit positionné.

```
$ ls -l
-rw-r--r-- 1 seb users 1392 Aug 14 15:55 dump.log
lrwxrwxrwx 1 seb users 4 Aug 14 15:21 lien_fic1 -> fic1
lrwxrwxrwx 1 seb users 4 Aug 14 15:21 lien_fic2 -> fic2
-rw-r--r-- 1 seb users 234 Aug 16 12:20 liste1
-rw-r--r-- 1 seb users 234 Aug 13 10:06 liste2
-rwxr--r-- 1 seb users 288 Aug 19 09:05 param.sh
```

```
-rwxr--r-- 1 seb users 430 Aug 19 09:09 param2.sh
-rwxr--r-- 1 seb users 292 Aug 19 10:57 param3.sh
drwxr-xr-x 2 seb users 8192 Aug 19 12:09 repl
-rw-r--r-- 1 seb users 1496 Aug 14 16:12 resultat.txt
-rw-r--r-- 1 seb users 1715 Aug 16 14:55 toto.txt
-rwxr--r-- 1 seb users 12 Aug 16 12:07 voir_a.sh
$ test -f lien_fic1 ; echo $?
1
$ test -x dump.log ; echo $?
1
$ test -d repl ; echo $?
0
```

d. Tests combinés par des critères ET, OU, NON

Vous pouvez effectuer plusieurs tests avec une seule instruction. Les options de combinaison sont les mêmes que pour la commande `find`.

Critère	Action
<code>-a</code>	AND, ET logique
<code>-o</code>	OR, OU logique
<code>!</code>	NOT, NON logique
<code>(...)</code>	groupeement des combinaisons. Les parenthèses doivent être verrouillées <code>\(...\)</code> .

```
$ test -d "repl" -a -w "repl" && echo "repl: repertoire, droit en écriture"
repl: repertoire, droit en écriture
```

e. Syntaxe allégée

Le mot `test` peut être remplacé par les crochets ouverts et fermés `[...]`. Il faut respecter un espace avant et après les crochets.

```
$ [ "$a" -lt "$b" ] && echo "$a est inferieur a $b"
10 est inferieur a 20
```

Le bash (et le ksh) intègre une commande interne de test qui se substitue au binaire `test`. Dans la pratique, la commande interne est entièrement compatible avec la commande externe mais bien plus rapide car il n'y a pas de création de nouveau processus. Pour forcer l'utilisation de la commande interne, utilisez les doubles crochets `[[...]]`.

```
$ [[ "$a" -lt "$b" ]] && echo "$a est inferieur a $b"
10 est inferieur a 20
```

6. if ... then ... else

La structure `if then else fi` est une structure de contrôle conditionnelle.

```
if <commandes_condition>
then
    <commandes exécutées si condition réalisée>
else
    <commandes exécutées si dernière condition pas réalisée>
fi
```

Vous pouvez aussi préciser `elif`, en fait un `else if`. Si la dernière condition n'est pas réalisée, on en teste une nouvelle.

```
$ cat param4.sh
#!/bin/bash
if [ $# -ne 0 ]
then
    echo "$# parametres en ligne de commande"
else
    echo "Aucun parametre; set alfred oscar romeo zoulou"
    set alfred oscar romeo zoulou
fi

echo "Nombre de parametres : $#"
echo "Parametres : 1=$1 2=$2 3=$3 4=$4"
echo "Liste : $*"

$ ./param4.sh titi toto
2 parametres en ligne de commande
Nombre de parametres : 2
Parametres : 1=toto 2=titi 3= 4=
Liste : toto titi

$ ./param4.sh
Aucun parametre; set alfred oscar romeo zoulou
Nombre de parametres : 4
Parametres : 1=alfred 2=oscar 3=romeo 4=zoulou
Liste : alfred oscar romeo zoulou
```

7. Choix multiples case

La commande **case ... esac** permet de vérifier le contenu d'une variable ou d'un résultat de manière multiple.

```
case Valeur in
    Modele1) Commandes ;;
    Modele2) Commandes ;;
    *) action_defaut ;;
esac
```

Le modèle est soit un simple texte, soit composé de caractères spéciaux. Chaque bloc de commandes lié au modèle doit se terminer par deux points-virgules. Dès que le modèle est vérifié, le bloc de commandes correspondant est exécuté. L'étoile en dernière position (chaîne variable) est l'action par défaut si aucun critère n'est vérifié. Elle est facultative.

Caractère	Rôle
*	Chaîne variable (même vide)
?	Un seul caractère
[...]	Une plage de caractères
[!...]	Négation de la plage de caractères
	OU logique

```
$ cat casel.sh
#!/bin/bash
if [ $# -ne 0 ]
then
    echo "$# parametres en ligne de commande"
else
    echo "Aucun parametre; set alfred oscar romeo zoulou"
```

```

        exit 1
fi

case $1 in
  a*)
    echo "Commence par a"
    ;;
  b*)
    echo "Commence par b"
    ;;
  fic[123])
    echo "fic1 fic2 ou fic3"
    ;;
  *)
    echo "Commence par n'importe"
    ;;
esac

exit 0
$ ./casel.sh "au revoir"
Commence par a
$ ./casel.sh bonjour
Commence par b
$ ./casel.sh fic2
fic1 ou fic2 ou fic3
$ ./casel.sh erreur
Commence par n'importe

```

8. Saisie de l'utilisateur

La commande **read** permet à l'utilisateur de saisir une chaîne et de la placer dans une ou plusieurs variable. La saisie est validée par [Entrée].

```
read var1 [var2 ...]
```

Si plusieurs variables sont précisées, le premier mot ira dans var1, le second dans var2, et ainsi de suite. S'il y a moins de variables que de mots, tous les derniers mots vont dans la dernière variable.

```

$ cat read.sh
#!/bin/bash
echo "Continuer (O/N) ? \c"
read reponse
echo "reponse=$reponse"
case $reponse in
  O)
    echo "Oui, on continue"
    ;;
  N)
    echo "Non, on s'arrête"
    exit 0
    ;;
  *)
    echo "Erreur de saisie (O/N)"
    exit 1
    ;;
esac
echo "Vous avez continué. Tapez deux mots ou plus :\c"
read mot1 mot2

```

```

echo "mot1=$mot1\nmot2=$mot2"
exit 0
$ ./read.sh
Continuer (O/N) ? O
reponse=0
Oui, on continue
Vous avez continue. Tapez deux mots ou plus :salut les amis
mot1=salut
mot2=les amis

```

9. Les boucles

Elles permettent la répétition d'un bloc de commandes soit un nombre limité de fois, soit conditionnellement. Toutes les commandes à exécuter dans une boucle se placent entre les commandes **do** et **done**.

a. Boucle for

La boucle **for** ne se base pas sur une quelconque incrémentation de valeur mais sur une liste de valeurs, de fichiers ...

```

for var in liste
do
    commandes à exécuter
done

```

La liste représente un certain nombre d'éléments qui seront successivement attribués à var.

Avec une variable

```

$ cat for1.sh
#!/bin/bash
for params in $@
do
    echo "$params"
done
$ ./for1.sh test1 test2 test3
test1
test2
test3

```

Liste implicite

Si vous ne précisez aucune liste à **for**, alors c'est la liste des paramètres qui est implicite. Ainsi le script précédent aurait pu ressembler à :

```

for params
do
    echo "$params"
done

```

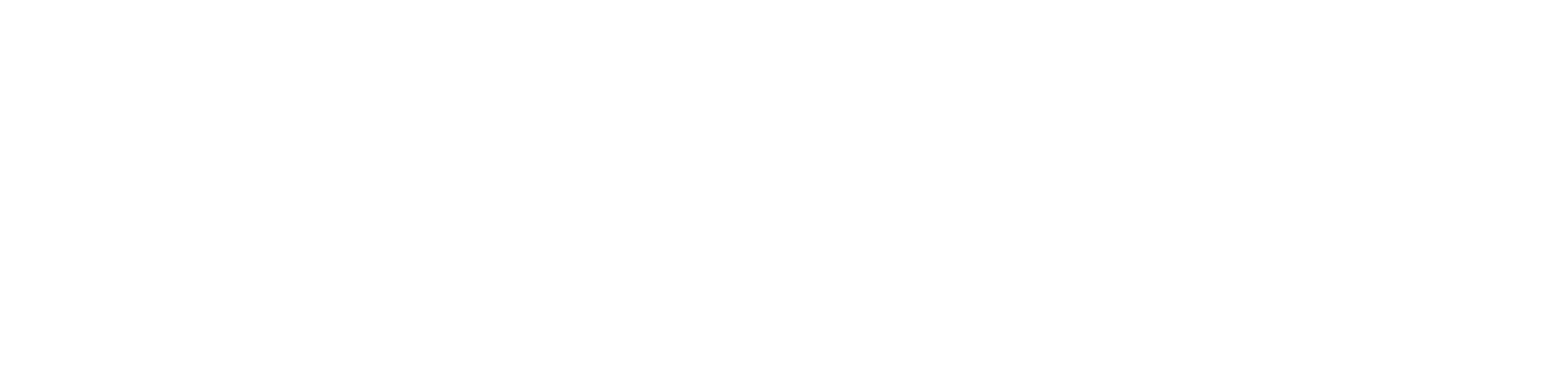
Avec une liste d'éléments explicite

Chaque élément situé après le « **in** » sera utilisé pour chaque itération de la boucle, l'un après l'autre.

```

$ cat for2.sh
#!/usr/bin/sh
for params in liste1 liste2

```



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Lecture d'un fichier ligne à ligne

```
#!/bin/bash
cat toto.txt | while read line
do
    echo "$line"
done
```

ou :

```
#!/bin/bash
while read line
do
    echo "$line"
done toto.txt
```

Il y a une énorme différence entre les deux versions. Dans la première, notez la présence du tube (pipe) : la boucle est exécutée dans un second processus. Aussi toute variable modifiée au sein de cette boucle perd sa valeur en sortie !

c. Boucle until

La commande **until** permet une boucle conditionnelle « jusqu'à ». Dès que la condition est réalisée, on sort de la boucle.

```
until condition
do
    commandes
done
```

ou :

```
until
bloc d'instructions formant la condition
do
    commandes
done
```

d. true et false

La commande **true** ne fait rien d'autre que de renvoyer 0. La commande **false** renvoie toujours 1. De cette manière il est possible de réaliser des boucles sans fin. La seule manière de sortir de ces boucles est un exit ou un break.

Par convention, tout programme qui ne retourne pas d'erreur retourne 0, tandis que tout programme retournant une erreur, ou un résultat à interpréter, retourne autre chose que 0. C'est l'inverse en logique booléenne.

```
while true
do
    commandes
    exit / break
done
```

e. break et continue

La commande **break** permet d'interrompre une boucle. Dans ce cas le script continue après la commande **done**. Elle peut prendre un argument numérique indiquant le nombre de boucles à sauter, dans le cas de boucles imbriquées (rapidement illisible).

```

while true
do
    echo "Chaine ? \c"
    read a
    if [ -z "$a" ]
    then
        break
    fi
done

```

La commande **continue** permet de relancer une boucle et d'effectuer un nouveau passage. Elle peut prendre un argument numérique indiquant le nombre de boucles à relancer (on remonte de n boucles). Le script redémarre à la commande **do**.

f. Boucle select

La commande **select** permet de créer des menus simples, avec sélection par numéro. La saisie s'effectue au clavier avec le prompt de la variable PS3. Si la valeur saisie est incorrecte, une boucle s'effectue et le menu s'affiche à nouveau. Pour sortir d'un select il faut utiliser un **break**.

```

select variable in liste_contenu
do
    traitement
done

```

Si `in liste_contenu` n'est pas précisé, ce sont les paramètres de position qui sont utilisés et affichés.

```

$ cat select.sh
#!/bin/bash
PS3="Votre choix :"
echo "Quelle donnee ?"
select reponse in Jules Romain Francois quitte
do
    if [[ "$reponse" = "quitte" ]]
    then
        break
    fi
    echo "Vous avez choisi $reponse"
done
echo "Au revoir."
exit 0

$ ./select.sh
Quelle donnee ?
1) Jules
2) Romain
3) Francois
4) quitte
Votre choix :1
Vous avez choisi Jules
Votre choix :2
Vous avez choisi Romain
Votre choix :3
Vous avez choisi Francois
Votre choix :4
Au revoir.

```

10. Les fonctions

Les fonctions sont des bouts de scripts nommés, directement appelés par leur nom, pouvant accepter des paramètres et retourner des valeurs. Les noms de fonctions suivent les mêmes règles que les variables sauf qu'elles ne peuvent pas être exportées.

```
nom_fonction ()  
{  
    commandes  
    return  
}
```

Les fonctions peuvent être soit tapées dans votre script courant, soit dans un autre fichier pouvant être inclus dans l'environnement. Pour cela saisissez :

```
. nomfic
```

Le point suivi d'un nom de fichier charge son contenu (fonctions et variables) dans l'environnement courant.

La commande **return** permet d'affecter une valeur de retour à une fonction. Il ne faut surtout pas utiliser la commande **exit** pour sortir d'une fonction, sinon on quitte le script.

```
$ cat fonction  
ll ()  
{  
    ls -l $@  
}  
li ()  
{  
    ls -i $@  
}  
$ . fonction  
$ li  
252 casel.sh      326 for4.sh      187 param.sh      897 resultat.txt  
568 dump.log      409 lien_fic1     272 param2.sh     991 toto.txt  
286 fonction       634 lien_fic2     260 param3.sh     716 voir_a.sh  
235 for1.sh        1020 liste       42 param4.sh     1008 while1.sh  
909 for2.sh        667 listel      304 read.sh  
789 for3.sh        1006 liste2      481 rep1
```

11. Calculs et expressions

a. expr

La commande **expr** permet d'effectuer des calculs sur des valeurs numériques, des comparaisons, et de la recherche dans des chaînes de texte.

Opérateur	Rôle
+	Addition.
-	Soustraction. L'étoile étant reconnue par le shell comme un wildcard, il faut la verrouiller avec un antislash : *.
*	Multiplication.
/	Division.
%	Modulo.
!=	Different. Affiche 1 si différent, 0 sinon.
=	Égal. Affiche 1 si égal, 0 sinon.
<	Inférieur. Affiche 1 si inférieur, 0 sinon.

Opérateur	Rôle
>	Supérieur. Affiche 1 si supérieur, 0 sinon.
<=	Inférieur ou égal. Affiche 1 si inférieur ou égal, 0 sinon.
>=	Supérieur ou égal. Affiche 1 si supérieur ou égal, 0 sinon.
:	Recherche dans une chaîne. Ex : expr Jules : J* retourne 1 car Jules commence par J. Syntaxe particulière : expr "Jules" : ".*" retourne la longueur de la chaîne.

```
$ expr 7 + 3
10
$ expr 7 \* 3
21
$ a=$(expr 13 - 10)
$ echo $a
3
$ cat expr1.sh
#!/bin/bash
cumul=0
compteur=0
nb_boucles=10
while [ "$compteur" -le "$nb_boucles" ]
do
    cumul=$(expr $cumul + $compteur)
    echo "$cumul=$cumul, boucle=$compteur"
    compteur=$(expr $compteur + 1)
done
$ ./expr1.sh
cumul=0, boucle=0
cumul=1, boucle=1
cumul=3, boucle=2
cumul=6, boucle=3
cumul=10, boucle=4
cumul=15, boucle=5
cumul=21, boucle=6
cumul=28, boucle=7
cumul=36, boucle=8
cumul=45, boucle=9
cumul=55, boucle=10
$ expr "Jules Cesar" : ".*"
11
```

b. Calculs avec bash

Le bash propose une forme simple de calculs sur les entiers, en plaçant l'opération entre `$((...))` :

```
$ a=1
$ a=$((a+1))
$ echo $a
2
$ b=2
$ a=$((a*b))
$ echo $a
4
```

Vous n'avez pas besoin de spécifier les \$ des noms des variables entre les doubles parenthèses.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Il peut être utile pour un administrateur de savoir créer des requêtes dans une base de données : certains outils se servent de bases SQL pour y stocker leurs données. Même dans le cas contraire, le SQL étant très utilisé et très pratique, il est utile de le connaître. Cette initiation vous montre comment effectuer les quatre requêtes de base : sélection, ajout, modification et suppression de données. Le lecteur avisé désirant en savoir plus sur le SQL et les bases de données pourra se reporter aux livres « MySQL 5 » et « SQL et Algèbre Relationnelle » aux éditions ENI.

Le modèle de base de données utilisé pour cette présentation est présent en annexe de ce livre.

2. Requêtes de sélection

a. Select

Les sélections avec l'instruction SELECT, sont les principales requêtes utilisées en SQL, permettant d'extraire des données d'une ou de plusieurs tables selon des critères donnés.

```
SELECT nom_champ1, nom_champ2, ...
FROM table;
```

Le résultat est retourné sous forme d'une table dont les en-têtes sont les noms des champs sélectionnés. Il est possible de les renommer avec AS. Par exemple :

```
SELECT nom, prenom, id AS identifiant FROM t_utilisateurs;
```

L'utilisation de l'étoile « * » comme champ permet de sélectionner tous les champs de la table. La requête suivante liste tout le contenu de la table :

```
SELECT * FROM t_utilisateurs;
```

b. Distinct

Si la requête retourne des lignes identiques, vous pouvez supprimer les doublons avec DISTINCT. Dans le cas suivant, il est logique de penser que plusieurs utilisateurs ont le même prénom. La clause DISTINCT va supprimer en sortie les doublons (ou plus).

```
SELECT DISTINCT prenom FROM t_utilisateurs;
```

c. Where

La clause WHERE spécifie les conditions de sélection des lignes.

```
SELECT nom_champ1, nom_champ2, ...
FROM table
WHERE condition;
```

Les conditions s'appliquent sur tous les champs de la table, y compris sur ceux qui ne sont pas dans le Select. Il est possible d'utiliser de nombreux opérateurs : =, >=, <=, >, <, (différent) et la négation avec le point d'exclamation : != (différent), !> (pas supérieur), !< (pas inférieur). Les critères de conditions peuvent être liés par les opérateurs logiques AND, OR et NOT. Si le critère est un texte, vous le mettez entre guillemets simples : 'titi'.

```
SELECT prenom FROM t_utilisateurs WHERE nom='ROHAUT';
```

Attributs de Where

Vous pouvez utiliser les attributs LIKE, BETWEEN ou IN dans une clause WHERE.

- LIKE compare la valeur d'un champ avec une valeur texte spécifiée avec d'éventuels caractères de remplacement : le « % » pour une chaîne quelconque et le « _ » pour un caractère quelconque.
- BETWEEN spécifie un intervalle de données.
- IN propose une liste d'éléments.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

*image
not
available*

*image
not
available*



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

*image
not
available*

*image
not
available*

*image
not
available*



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

*image
not
available*

*image
not
available*

*image
not
available*

72 Si le fichier fic existe et est lisible, listez son contenu. Sinon, créez-le.

`[[-e fic -a -r fic]] && cat fic || touch fic. Ou encore [-r fic] car de toute façon le test retourne une erreur si le fichier est absent.`

73 À l'aide d'un if, s'il n'y a aucun paramètre transmis au script, mettez-en deux prépositionnés : -l et -r.

`if [$# -eq 0] ; then set '-l' '-r' ; fi`

74 Un programme attend trois valeurs au clavier, les unes après les autres, séparées par des espaces, sur la même ligne. Quelle doit être la commande à utiliser ?

`read a b c`

75 Comment faire une boucle pour lister tous les paramètres ? Choisissez deux réponses.

A - for param ; do echo \$param ; done

B - while [\$# -ne 0]; do echo \$1; shift ; done

C - until [\$# -ne 0]; do shift; echo \$1; done

D - a=1; while [\$a -ne \$#]; do echo \${!a}; a=\$((a+1)); done

A et B.

76 Comment sortir d'une boucle sans fin ?

Avec un break.

77 Comment programmer une fonction abs qui retourne la valeur absolue d'un nombre ?

`abs() { v=$1 ; [$v -lt 0] && v=$((0-v)) ; echo $v ; }`

78 Comment empêcher l'interception du signal 15 dans un script ?

`trap '' 15`

N. Travaux pratiques

1. Gestion des fichiers

But : effectuer des manipulations de base sur le système de fichiers.

1. À partir de votre répertoire personnel créez la structure suivante, en utilisant une seule commande :

```
|----dossier1
|      |----dossier3
|----dossier2
|      |----dossier4
```

Utilisez la commande **mkdir** avec le paramètre **-p** :

`$ mkdir -p dossier1/dossier3 dossier2/dossier4`

2. Déplacez-vous dans le répertoire dossier1 avec un chemin absolu et créez le fichier fichier1 dans ce répertoire.

Le chemin absolu part de la racine sans aucun chemin relatif. Tapez :

`$ cd /home/user/dossier1`

Créez le fichier avec **touch** :

`$ touch fichier1`

3. Copiez fichier1 dans le répertoire dossier3 avec un chemin relatif.

Le chemin est relatif en fonction de l'emplacement actuel : .. remonte d'un niveau, et . définit l'emplacement courant.

```
$ pwd
/home/user/dossier1
```



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

*image
not
available*

*image
not
available*

5. Additionnez ces deux valeurs et placez le résultat dans result. Affichez result.

```
[ $# -ne 1 -o ! -f $1 ] && exit
result=0
while read ligne
do
    c1=$(echo $ligne | cut -d" " -f1)
    c2=$(echo $ligne | cut -d" " -f2)
    result=$((c1+c2))
    echo -e "$c1 + $c2 = $result"
done < $1
```

7. Fonction Shell

But : un nombre est premier seulement si ses seuls diviseurs sont 1 et lui-même.

Autrement dit, si on peut diviser un nombre par autre chose que par un et lui-même et que le résultat de cette division est un entier (ou que le reste de cette division est 0, ce qui revient au même) alors il n'est pas premier.

Le chiffre 1 n'est pas premier car il n'a pas deux diviseurs différents.

Le chiffre 2 est premier (2×1 donc deux diviseurs différents).

Aucun nombre pair (sauf le chiffre 2) n'est premier (car ils sont tous divisibles par 2).

Liste des dix premiers nombres premiers : 2 3 5 7 11 13 17 19 23 29.

Écrivez une fonction appelée **est_premier**, qui prend comme paramètre un nombre et qui renvoie le code retour 0 si le nombre est premier, 1 sinon.

```
est_premier()
{
    [ $1 -lt 2 ] && return 1
    [ $1 -eq 2 ] && return 0
    [ $(expr $1 % 2) -eq 0 ] && return 1
    i=3
    while [ $((i*i)) -le $1 ]
    do
        [ $(expr $1 % $i) -eq 0 ] && return 1
        i=$((i+2))
    done
    return 0
}
```



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



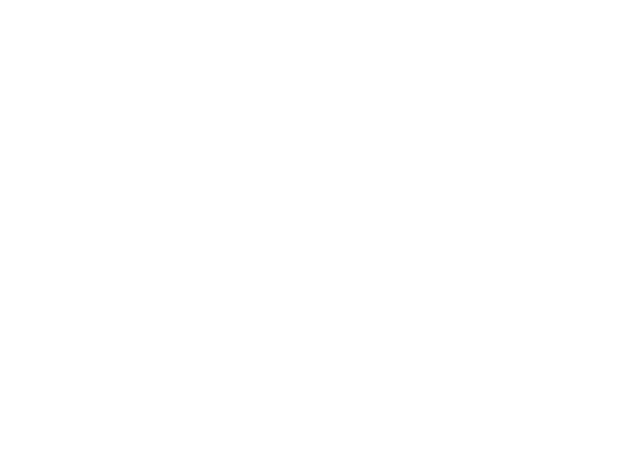
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



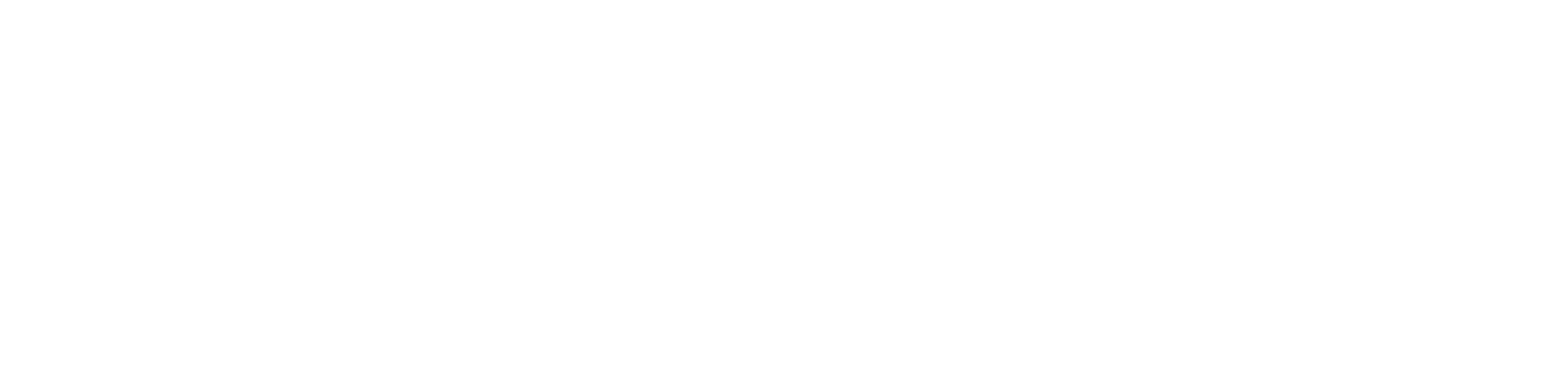
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

```

needs_recovery sparse_super large_file
Filesystem flags:           signed directory hash
Default mount options:     (none)
Filesystem state:          clean
Errors behavior:           Continue
Filesystem OS type:        Linux
Inode count:                66400
Block count:                132536
Reserved block count:      6626
Free blocks:                 123672
Free inodes:                  66358
First block:                   0
Block size:                    4096
Fragment size:                  4096
Reserved GDT blocks:         32
Blocks per group:            32768
Fragments per group:         32768
Inodes per group:             13280
Inode blocks per group:       415
Filesystem created:          Sat Feb 23 22:52:05 2008
Last mount time:              Thu Mar 20 19:13:51 2008
Last write time:              Thu Mar 20 19:13:51 2008
Mount count:                   1
Maximum mount count:         500
Last checked:                  Thu Mar 20 19:07:48 2008
Check interval:                5184000 (2 months)
Next check after:              Mon May 19 20:07:48 2008
Reserved blocks uid:          0 (user root)
Reserved blocks gid:          0 (group root)
First inode:                     11
Inode size:                      128
Journal inode:                   8
Default directory hash:        tea
Directory Hash Seed:           f1584155-5760-4445-8009-0444ffa81f91
Journal backup:                  inode blocks
Taille du journal:               16M

```

```

Groupe 0 : (Blocs 0-32767)
superbloc Primaire à 0, Descripteurs de groupes à 1-1
Blocs réservés GDT à 2-33
Bitmap de blocs à 34 (+34), Bitmap d'i-noeuds à 35 (+35)
Table d'i-noeuds à 36-450 (+36)
25696 blocs libres, 13256 i-noeuds libres, 2 répertoires
Blocs libres : 4559-5660, 6767-12287, 12289-14335, 15644-26721,
26820-32767
I-    noeuds libres : 14, 16-20, 23-24, 33-13280
...

```

Cette sortie est très longue (elle a pourtant été tronquée), mais vous donne tous les détails possibles sur le système de fichiers. Vous pouvez isoler uniquement l'en-tête (jusqu'à la taille du journal) avec le paramètre `-h`. Le mieux, si vous cherchez une information précise, est d'utiliser la commande `grep`.

```
# dumpe2fs -h /dev/sda2|grep -i "block size"
dumpe2fs 1.40.2 (12-Jul-2007)
Block size:          4096
```

d. tune2fs

La commande **tune2fs** permet de modifier certains paramètres d'un système de fichiers ext2 ou ext3. Vous avez déjà rencontré cette commande lorsqu'il a fallu convertir l'ext2 vers l'ext3, et vice versa. Voici quelques paramètres supportés par la commande :

Paramètre	Signification
-c n	Nombre de fois où le système de fichiers doit être monté avant d'être automatiquement vérifié. Par exemple si n vaut 30, au trentième montage fsck sera automatiquement lancé. Si n vaut 0 ou -1, la vérification est désactivée.
-i n	Intervalle de temps entre deux vérifications. L'unité par défaut est le jour. Les suffixes d (jours), w (semaines) ou m (mois) peuvent être accolés au nombre. -i 180d signifie que le système de fichiers sera contrôlé au bout de 180 jours.
-j	Rajoute un journal sur un système de fichiers ext2. Il est préférable de le faire système démonté. Dans le cas contraire un fichier caché .journal est rajouté à la racine du système, immuable (sauf si vous détruisez le journal), qui sera intégré au sein du système à la prochaine exécution de fsck.
-L	Modifie l'étiquette (label, nom de volume). L'étiquette ne doit pas dépasser 16 caractères ou elle sera tronquée.
-e err	Indique comment doit réagir le noyau si une erreur est détectée sur le système de fichiers lors du boot. La valeur par défaut est « continue ». Les autres valeurs possibles sont « panic » (blocage du noyau en mode kernel panic) et « remount-ro » : passage en lecture seule.
-m n	Le pourcentage n représente la taille réservée que s'octroient les processus lancés par root (et root lui-même) sur la taille totale. Sur un système de fichiers réservés aux utilisateurs, mettre 0 permet de remplir le système de fichiers jusqu'à 100%. Mais il est important sur la racine, ou /var, de réservé une zone pour que certains services comme syslogd puissent continuer à écrire les traces systèmes. Par défaut 5% d'espace est réservé.
-o [^]option	Ajoute ou supprime (avec ^) l'option indiquée par défaut au montage. Les options peuvent être par exemple acl ou user xattr.
-O [^]fonction	Ajoute ou supprime (avec ^) la fonction indiquée. La fonction la plus connue est « has_journal ». -O has_journal équivaut à -j. -O ^has_journal convertit ext3 en ext2.
-U UUID	Modifie la valeur de l'UUID à votre convenance (format hexadécimal). Il est aussi possible de le supprimer (clear), d'en générer un aléatoirement (random) ou d'en générer un en fonction de la date (time).
-s 0/1	Active ou désactive la « sparse super feature ». Sur des disques de grande taille, l'activation réduit le nombre de blocs de secours pour gagner de la place. Vous devez ensuite exécuter fsck.

```
# tune2fs -m 0 -s 1 -U random -e remount-ro -c 60 -I 180 /dev/sdb1
```

Notez que -c et -i vont de pair. Le premier terme échu est le premier qui effectue la vérification par fsck, puis les dates et compteurs sont remis à zéro. Les compteurs sont vérifiés lors du montage au démarrage (boot) du système. Si cela fait 300 jours que le système n'a pas rebooté et que le système de fichiers n'a pas été vérifié durant cet intervalle, il ne sera pas vérifié automatiquement durant ces 300 jours mais au prochain reboot. Certains serveurs ne rebootant que rarement (par exemple tous les deux ans) le système de fichiers n'est pas automatiquement vérifié durant tout ce temps...

H. Le swap

1. Pourquoi créer un swap ?

Dans un environnement 32 bits un processus peut théoriquement accéder à 4 Go d'espace mémoire. Il dispose de 4 Go de mémoire virtuelle, rien qu'à lui et à laquelle aucun autre processeur ne peut accéder. Dans la pratique il y a plusieurs freins à cette possibilité :

- L'espace mémoire adressable d'un processus est partagé entre zone de code et zone de données dont la taille peut varier selon le noyau utilisé.
- Les ordinateurs ne disposent pas tous de 4 Go de mémoire (bien qu'il soit courant de trouver des serveurs Linux disposant de 16, 32 ou même 64 Go de mémoire).
- Tous les processus doivent se partager la mémoire de l'ordinateur.

Que se passe-t-il si un processus n'a plus assez de mémoire pour traiter ses données ? Le système d'exploitation va décharger des segments de la mémoire physique dans une zone d'échange sur disque qui fera office de mémoire virtuelle tampon. Il y a donc un échange entre la mémoire physique et cette zone d'échange, appelé l'espace de swap. Ce processus permet d'utiliser plus de mémoire que l'ordinateur n'en dispose réellement, au prix d'un net ralentissement si le programme est très gourmand.

2. Taille optimale

Il n'y a pas de règles strictes sur la taille du swap. Cependant les quelques règles courantes suivantes sont valables dans la plupart des cas :

- Moins de 512 Mo de RAM : deux fois la RAM.
- 1 Go à 4 Go : la taille de la RAM.
- Plus de 4 Go : 4 Go, plus ou moins, selon l'utilisation des processus.

3. Créer une partition de swap

- Vous savez déjà créer une partition. Créez une partition avec fdisk de la taille souhaitée pour le swap, et donnez-lui le type 83.
- Synchronisez la table des partitions avec **partprobe**.
- Utilisez la commande **mkswap** pour préparer la partition à recevoir du swap.

```
# mkswap /dev/sda5
Initialisation de la version de l'espace de swap 1, taille =
2154983 kB
pas d'étiquette, UUID=c84714e6-c42c-44d4-9fe7-10dc6afac644
```

- Votre swap est prêt.

Il est possible d'attribuer une étiquette à la partition de swap avec le paramètre **-L**.

☞ Si vous créez plus de 1 ou 2 Go de swap, vous devriez penser à créer plusieurs partitions de swap sur des disques différents situés sur des contrôleurs matériels différents. Linux répartira la charge sur chacune de ces partitions, ce qui garantira des accès plus rapides.

4. Activer et désactiver le swap

a. Activation dynamique

Linux permet d'activer et de désactiver le swap, ou des morceaux de swap, directement sans avoir à redémarrer le système.

La commande **swapon** permet d'activer une partition de swap :

```
# swapon /dev/sda5
```

Le paramètre **-p** permet de modifier la priorité du swap. Plus la valeur, comprise entre 0 et 32767, est élevée, plus la priorité d'une zone de swap est élevée. Le système l'utilisera en priorité. Ce paramètre est utile si plusieurs partitions de swap existent sur des disques différents. Dans ce cas, privilégiez soit le disque le plus rapide, soit indiquez une priorité égale pour une meilleure répartition.

Comme avec mount, le paramètre **-L** permet d'activer une zone de swap grâce à son étiquette.

La commande **swapoff** désactive une zone de swap. Veillez à disposer de l'espace mémoire libre nécessaire, sinon la commande ne fonctionnera pas.

Le contenu de **/proc/swaps** reflète l'état actuel des zones de swap actives.

# cat /proc/swaps	Filename	Type	Size	Used	Priority
		partition			
	/dev/sda5		1461872	2012	-1

b. Dans /etc/fstab

Les zones de swap se placent dans le fichier **/etc/fstab**. Voici un exemple :

```
/dev/sda5 swap swap defaults 0 0
```

Les options **noauto** et **pri=X** peuvent être précisées. L'option **pri** permet de définir la priorité de la zone de swap.

Lors du démarrage, le système exécute **swapon -a** qui active toutes les partitions de swap présentes dans la fstab sauf si noauto est précisé.

Lors de l'arrêt, le système exécute **swapoff -a** qui désactive complètement le swap.

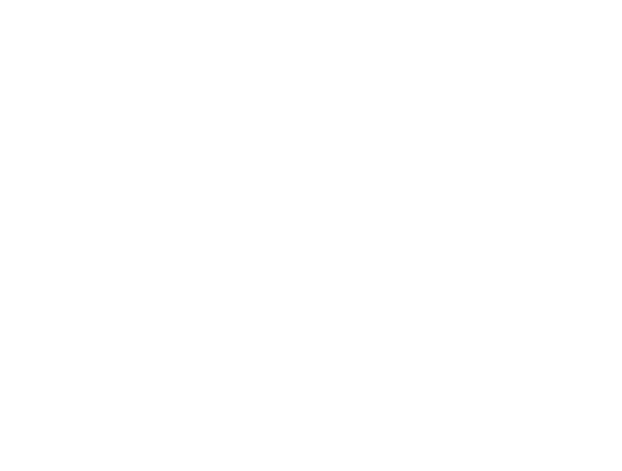
5. En cas d'urgence : fichier de swap

Que faire si vous manquez d'espace de swap et qu'il n'est plus possible de créer une nouvelle partition ? Linux sait utiliser un fichier d'échange (comme Windows, voire celui de Windows). S'il reste de la place sur un de vos systèmes de fichiers vous pouvez créer dessus un fichier d'échange d'une taille prédéfinie. Ce swap sera moins performant qu'une partition de swap (problème de fragmentation, temps d'accès au système de fichiers).

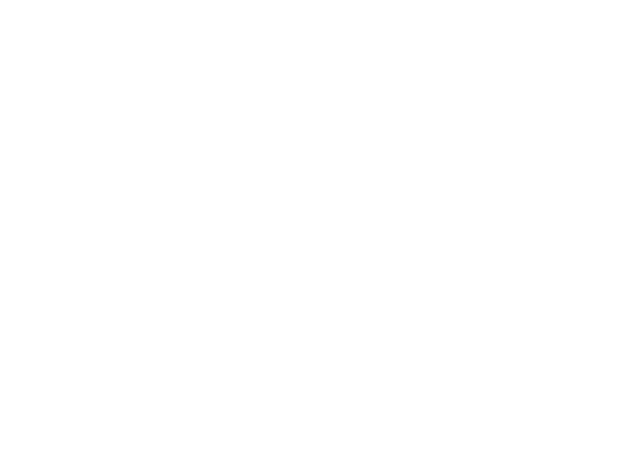
Voici les manipulations pour un petit swap de 32 Mo :

```
# free | grep Swap
Swap:      2104472      4344    2100128
# dd if=/dev/zero of=/swap bs=1024 count=32768
32768+0 enregistrements lus
32768+0 enregistrements écrits
33554432 bytes (34 MB) copied, 0,35697 s, 94,0 MB/s
slyserver:~ # mkswap /swap
Initialisation de la version de l'espace de swap 1, taille = 33550 kB
pas d'étiquette, UUID=b2e5e99e-09a1-4b2d-ac76-59f76526453a
slyserver:~ # chmod 600 /swap
slyserver:~ # sync
slyserver:~ # swapon -v /swap
swapon sur /swap
slyserver:~ # free | grep Swap
Swap:      2137232      4308    2132924
```

Modifiez éventuellement le fichier **/etc/fstab**, en espérant que le swap soit activé après le montage des systèmes de fichiers. Le swap est activé au boot, généralement après le montage de **/**. Mais s'il est ailleurs (autre point de montage) alors comme le swap est activé avant les autres points de montage, il en résultera une erreur. Il est donc préférable de créer le fichier dans le système de fichiers racine **/**.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



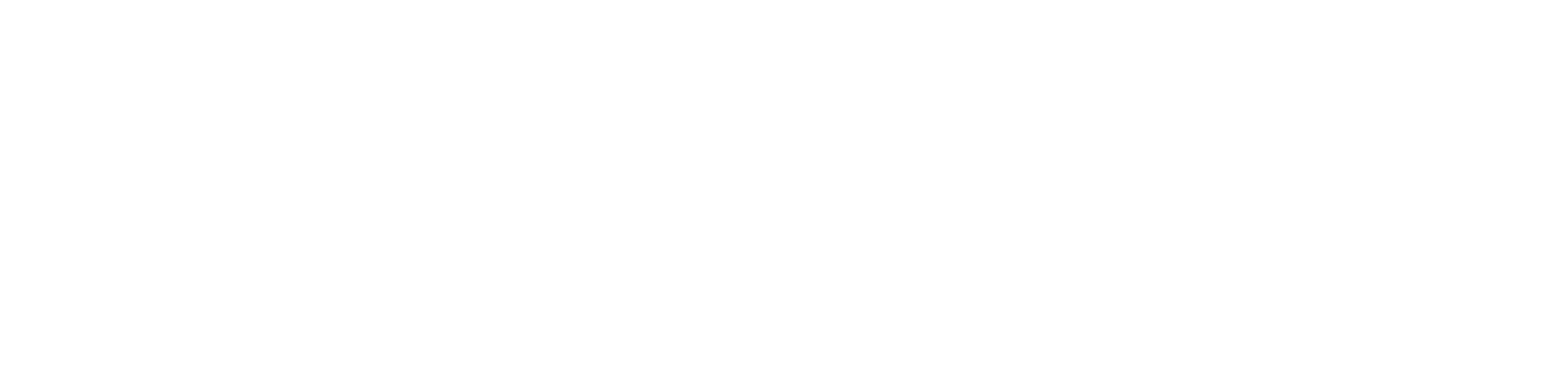
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



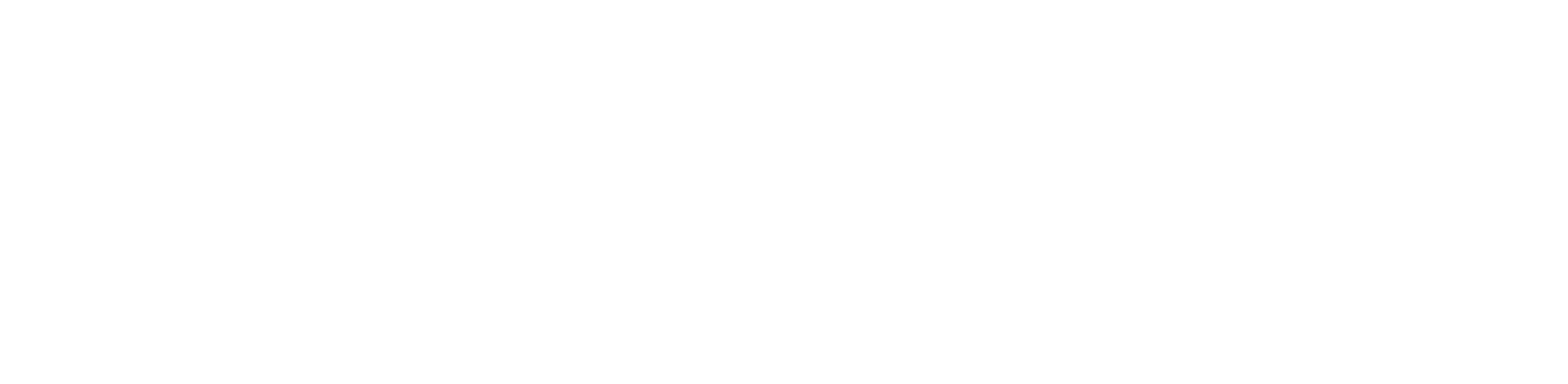
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

2. Niveaux d'exécution

Un niveau d'exécution, ou runlevel, est un état dans lequel se trouve Unix/Linux. Cet état est contrôlé par init. Chaque état dispose de sa propre configuration (soit par inittab, soit par des scripts appelés initscripts). Un niveau d'exécution peut par exemple être utilisé pour lancer Unix en mono-utilisateur, en multi-utilisateurs, avec ou sans réseau, avec ou sans mode graphique. Tous les niveaux sont personnalisables par l'administrateur. Ces niveaux sont généralement définis comme ceci par convention sur les distributions Red Hat/Fedora, Mandriva, openSUSE et associées :

Niveau	Effet
0	Halt : stoppe le système d'exploitation, éteint la machine.
1	Mode mono-utilisateur utilisé pour la maintenance, mode console.
2	Multi-utilisateur, sans réseau, console.
3	Multi-utilisateur, avec réseau, console.
4	Idem que le 3, laissé à la convenance de l'administrateur.
5	Multi-utilisateur, avec réseau, avec environnement graphique X Window.
6	Reboot : redémarrage de la machine.
S,s	Single user mode, le mode le plus bas en cas de soucis.

Les niveaux 7 à 9 sont parfaitement valides mais pas utilisés par défaut. Le niveau d'exécution par défaut est positionné dans `/etc/inittab` sur la ligne `initdefault`.

`id:5:initdefault:`

Remplacez 5 par le niveau souhaité au démarrage.

► Un gag, posé lors de certains exercices éliminatoires de certifications, consiste à créer une situation de panne où la valeur par défaut est positionnée à 0 ou 6. Dans le premier cas la machine s'éteint dès l'exécution de init, dans l'autre elle redémarre en boucle...

La distribution Debian (et les distributions qui en dérivent) considère aussi les niveaux 2 à 5 comme multi-utilisateur mais n'établit pas de différences entre ces niveaux, et démarre par défaut au niveau 2 où tout est lancé, y compris éventuellement l'interface graphique.

Comme chaque niveau peut être totalement modifié et reconfiguré, il est possible de tout redéfinir et donc de faire en sorte qu'une Debian démarre comme une Red Hat, ou qu'une Red Hat démarre comme une Debian. Pour des raisons de conformité et de support, considérez qu'il est important de rester conforme au « standard » de la distribution que vous utilisez.

3. /etc/inittab

Le comportement du processus init et des runlevels est défini dans le fichier `/etc/inittab`. La syntaxe d'une ligne est la suivante :

`Id:[niveaux]:action:commande`

Champ	Description
Id	Identifiant de ligne sur quatre caractères, juste indicatif (sous Linux avec getty/mingetty : numéro de terminal).
Niveaux	Indique si la commande doit être prise en compte pour le niveau demandé, c'est la liste des niveaux sans séparateur.
Action	Type d'action à effectuer selon les circonstances pour cette ligne.
Commande	La commande à exécuter avec ses paramètres et les redirections.

L'action est très importante car elle définit les activités de init lors du démarrage et du changement de niveau.

Voici les principales :

Action	Signification
initdefault	Définit le niveau par défaut lors du boot et du lancement d'init.
sysinit	Exécuté une seule et unique fois lors du démarrage du système.
boot	Idem mais après sysinit.
bootwait	Idem, mais init attend la fin de l'exécution de la commande avant de continuer à dérouler le fichier inittab.
off	La ligne est ignorée.
once	La commande est exécutée à chaque changement de niveau pour les niveaux spécifiés.
wait	Idem, mais init attend la fin de l'exécution avant de continuer.
respawn	La commande est lancée pour les niveaux concernés. Si le processus se termine, il est automatiquement relancé. C'est le cas pour les terminaux si un utilisateur s'en déconnecte.
powerwait	La commande est lancée si le serveur passe sur alimentation de secours (UPS).
powerfail	Idem, mais sans attente de la fin d'exécution de la commande.
powerokwait	La commande est lancée lorsque le courant est rétabli.
powerfailnow	Commande de dernier recours lorsque l'alimentation de secours est presque vide.
ctrlaltdel	Init reçoit un signal SIGINT issu d'une séquence [Alt][Ctrl][Suppr].

Voici un exemple issu d'une installation openSUSE 10.3 :

```
# Niveau d'exécution à 5 (multiuser graphique)
id:5:initdefault:

# Premier script execute au démarrage
si::bootwait:/etc/init.d/boot

# Gestion des services par niveau d'exécution
10:0:wait:/etc/init.d/rc 0
11:1:wait:/etc/init.d/rc 1
12:2:wait:/etc/init.d/rc 2
13:3:wait:/etc/init.d/rc 3
14:4:wait:/etc/init.d/rc 4
15:5:wait:/etc/init.d/rc 5
16:6:wait:/etc/init.d/rc 6

# Cas du mode single, console de secours pour root
ls:S:wait:/etc/init.d/rc S
~~:S:respawn:/sbin/sulogin

# Action sur Alt+Ctrl+Del
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now

# Que faire en cas de coupure de courant
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop

# Lancement des consoles virtuelles Alt+Fx
1:2345:respawn:/sbin/mingetty --noclear tty1
2:2345:respawn:/sbin/mingetty tty2
```

```
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

4. Changement de niveau

Vous pouvez changer de niveau à la volée après le démarrage de la machine avec la commande **/sbin/init** ou **/sbin/telinit**, cette dernière étant un simple lien symbolique vers init. La commande suivante passe au niveau 5.

```
# telinit 5
```

Les valeurs q, u ou -t peuvent être précisées :

- Q ou q : init relit le fichier **/etc/inittab**, s'il a été modifié, en corigeant ses tables internes.
- U ou u : init se relance sans relire inittab et sans changer de niveau. Si des services ont été rajoutés ou supprimés du niveau en cours, init prend en compte la modification.
- -t : quand init a terminé l'arrêt des services (ou plutôt quand le script rc l'a fait, voir un peu plus loin), init envoie le signal SIGTERM à tous les processus restants, leur demandant de se terminer proprement, attend le nombre de secondes spécifié (5 par défaut), puis envoie SIGKILL.

Le niveau d'exécution actuel est visible avec la commande **/sbin/runlevel**. La première valeur renvoyée est le niveau précédent le niveau actuel. Un N signifie qu'il n'y a pas de précédent niveau. La seconde valeur est le niveau actuel.

```
# runlevel
N 5
```

5. Paramétrage système de base

Quel que soit le niveau d'exécution précisé par défaut, init lance toujours la commande associée aux actions sysinit, bootwait ou boot de **/etc/inittab** lors du démarrage du système, l'action sysinit étant la première.

- Sous Red Hat : si::sysinit:/etc/rc.d/rc.sysinit
- Sous openSUSE : si::bootwait:/etc/init.d/boot
- Sous Debian : si::sysinit:/etc/init.d/rcS

Sous Red Hat, c'est un seul script monolithique qui s'occupe de toute la configuration de base. Sous Debian, le script appelle tous les scripts du niveau S (single). Sous openSUSE le script met en place le strict nécessaire puis exécute le contenu de **/etc/rc.d/boot.d** qui établit le reste de la configuration de base.

Dans tous les cas, les tâches suivantes sont exécutées à peu près dans cet ordre :

- Configuration des paramètres du noyau présents dans **/etc/sysctl.conf** (ex : IP Forwarding).
- Mise en place des fichiers périphériques (**/dev** via udev par exemple).
- Configuration de l'horloge du système.
- Chargement des tables de caractères du clavier.
- Activation des partitions d'échange SWAP.
- Définition du nom d'hôte.
- Contrôle et montage du système de fichiers racine (en lecture-écriture cette fois).
- Ajout des périphériques RAID et/ou LVM. Ceci peut déjà être mis en place lors du chargement de inittab.
- Activation des quotas de disque.
- Contrôle et montage des autres systèmes de fichiers.
- Nettoyage des verrous (stale locks) et des fichiers PID en cas d'arrêt brusque.

Il est possible avec certaines distributions de passer en mode interactif. Au début du boot, après le lancement d'init, il peut vous être demandé de taper sur la lettre i puis de répondre par oui ou par non aux différentes actions.

6. Niveaux d'exécution System V

a. rc

Le script `/etc/init.d/rc` prend comme paramètre le niveau d'exécution par défaut selon la ligne `initdefault` de `/etc/inittab` ou celui spécifié lors de l'appel manuel des commandes `init` ou `telinit`. Le script rc initialise le niveau d'exécution voulu et est responsable du démarrage et de l'arrêt des services associés quand le niveau d'exécution change.

```
11:1:wait:/etc/init.d/rc 1
12:2:wait:/etc/init.d/rc 2
13:3:wait:/etc/init.d/rc 3
14:4:wait:/etc/init.d/rc 4
15:5:wait:/etc/init.d/rc 5
16:6:wait:/etc/init.d/rc 6
```

Les services sont analysés à chaque niveau d'exécution. Lors du passage d'un niveau à un autre, et quel que soit l'ordre (du 2 au 5, du 5 au 3, etc.) le script rc compare les services qui doivent être arrêtés ou démarrés entre l'ancien et le nouveau niveau. Si un service est commun aux deux niveaux, il est maintenu. Si un nouveau service doit être lancé dans le nouveau niveau, il le lance. Si un service doit être arrêté car il est absent du nouveau niveau, il l'arrête.

Ce fonctionnement, standard à toutes les distributions Linux de type System V, n'est pas commun à tous les Unix. HP-UX (un Unix de HP) considère qu'il doit y avoir une progression constante dans les niveaux, passant successivement du 1 au 3 (1 puis 2 puis 3) et chargeant successivement les services. À l'arrêt il redescend jusqu'au niveau 0 en terminant successivement les services. La différence est de taille : il ne compare pas les niveaux et n'effectue pas d'arrêt/relance entre chaque niveau...

7. Gestion des niveaux et des services

a. Services dans init.d

Le niveau d'exécution définit les services à démarrer pour ce niveau. C'est le script rc qui charge les services. Les services sont contrôlés (démarrage, arrêt, relance, status, etc.) à l'aide de scripts présents dans `/etc/init.d`.

```
# cd /etc/init.d
# ls -l
-rwxr-xr-x 1 root root 1128 aou 9 2004 acpid
-rwxr-xr-x 1 root root 834 sep 28 2004 anacron
-rwxr-xr-x 1 root root 1429 jun 22 2004 apmd
-rwxr-xr-x 1 root root 1176 avr 14 2006 atd
-rwxr-xr-x 1 root root 2781 mar 5 2007 auditd
-rwxr-xr-x 1 root root 17058 sep 5 2007 autofs
-rwxr-xr-x 1 root root 1368 fev 2 2007 bluetooth
-rwxr-xr-x 1 root root 1355 mai 2 2006 cpuspeed
-rwxr-xr-x 1 root root 1904 jui 16 2007 crond
-rwxr-xr-x 1 root root 2312 oct 30 13:46 cups
...
```

Pour chaque niveau d'exécution n , il existe un répertoire `rcn.d` qui contient des liens symboliques (raccourcis) vers les services présents dans `/etc/init.d` à lancer ou arrêter. Ce répertoire peut être à différents endroits selon la distribution :

- Red Hat : `/etc/rc.d/rcn.d` avec des liens sur `/etc/rcn.d`



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



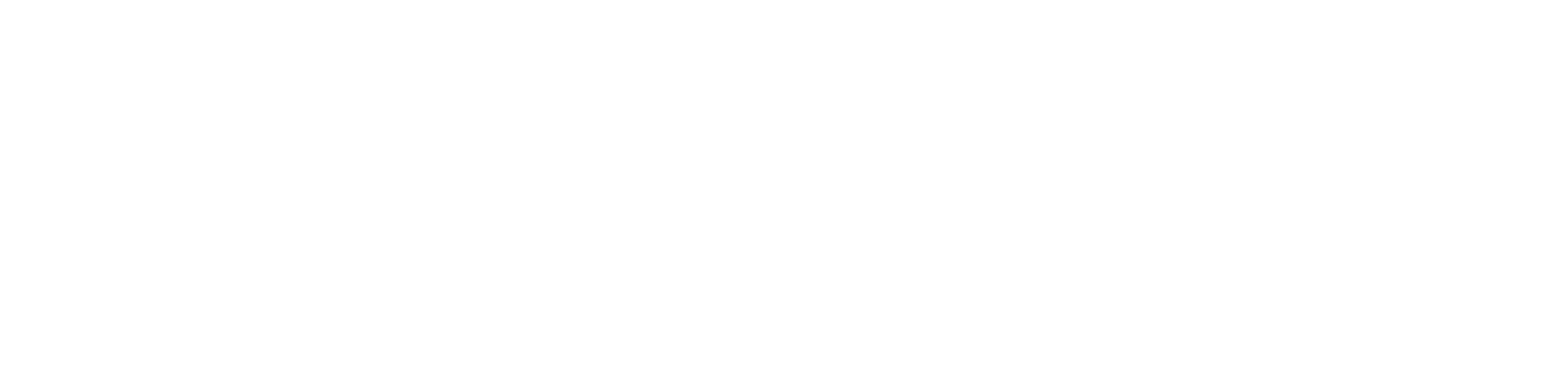
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



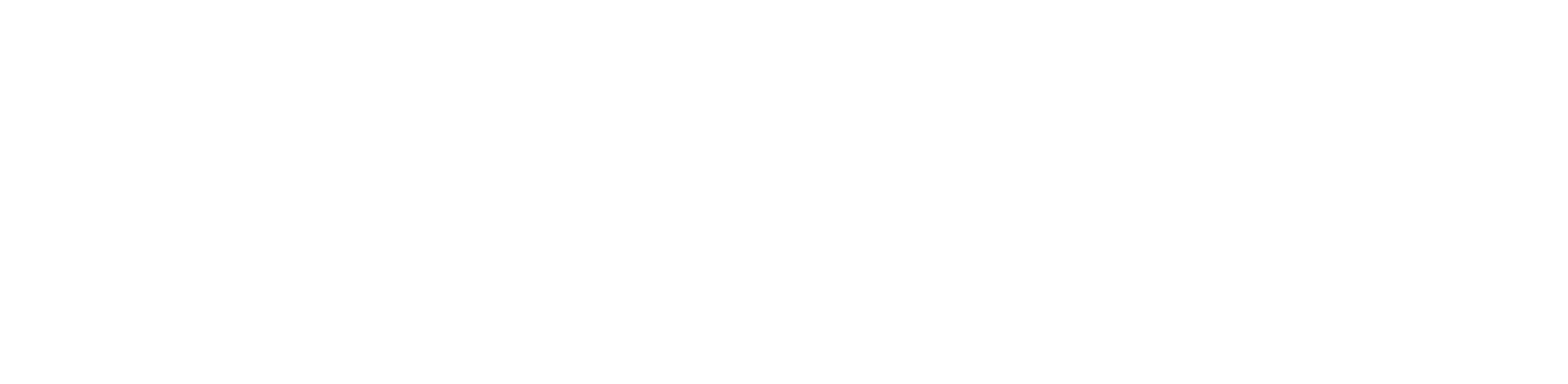
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



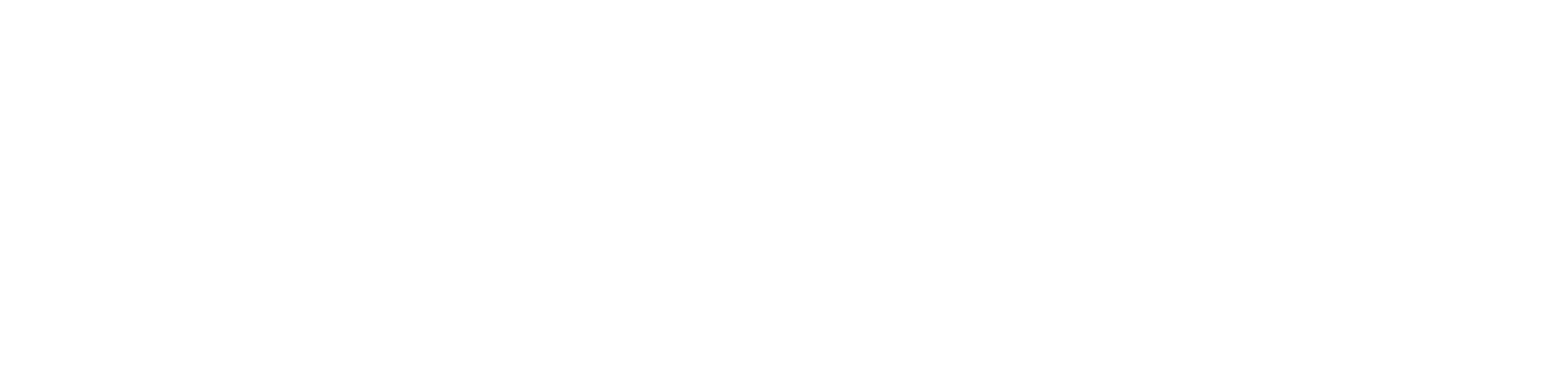
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



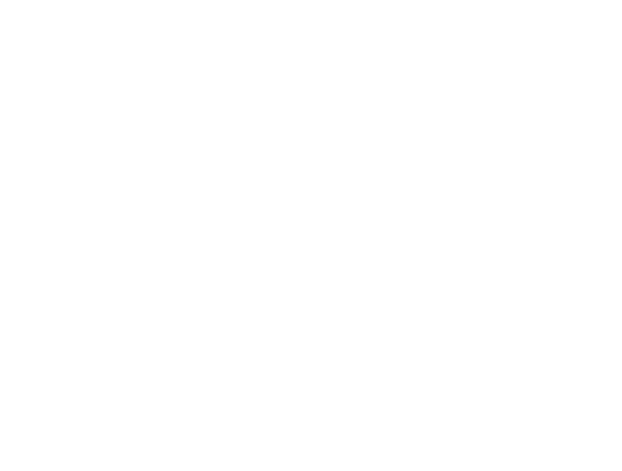
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



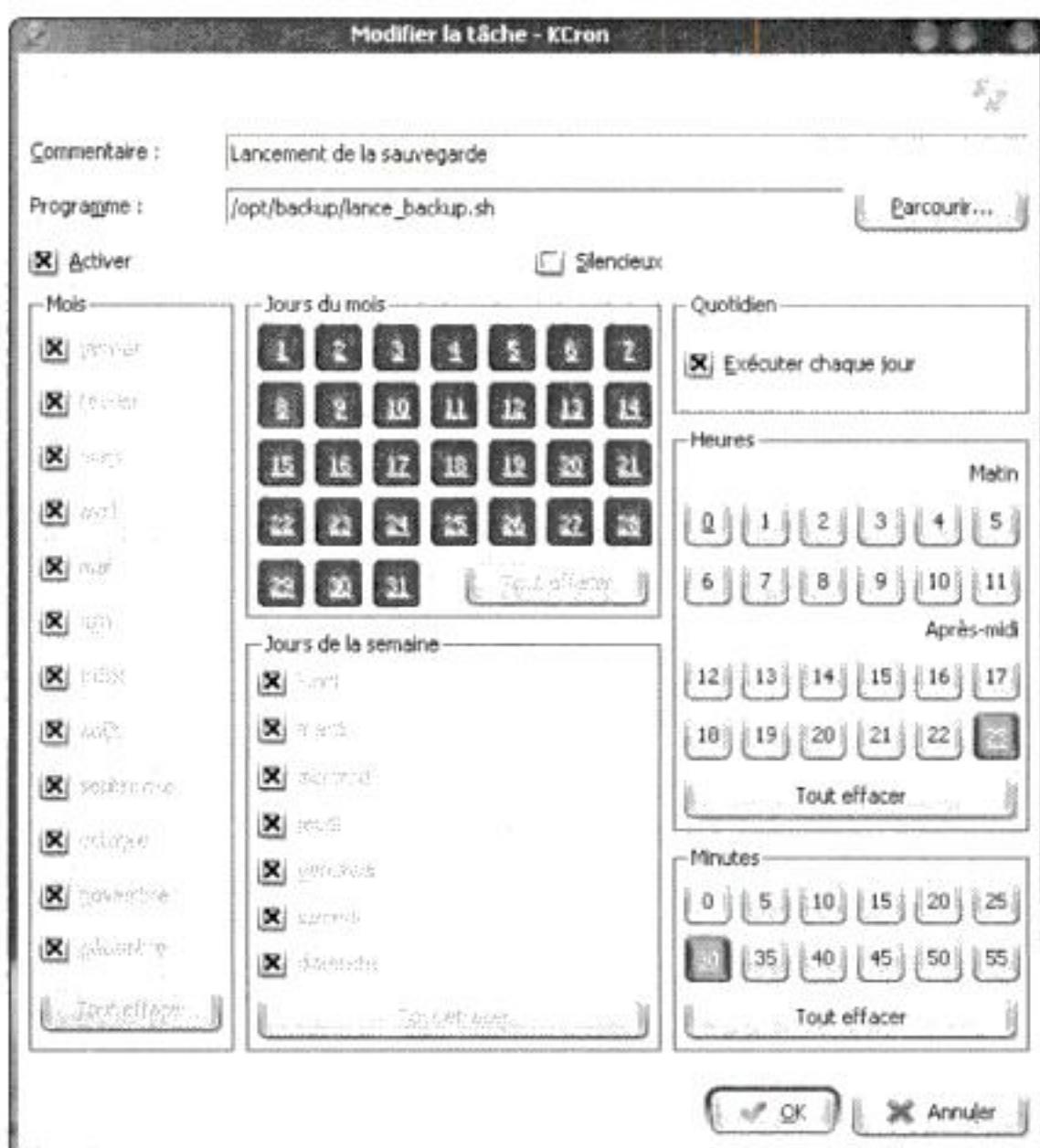
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



L'outil kcron sous KDE

L'exemple de la capture présente un enregistrement de la crontab d'un utilisateur en cours d'édition. La tâche `lance_backup.sh` est exécutée tous les jours à 23h30. Les lignes correspondantes dans la crontab (au format texte) sont les suivantes :

```
# Lancement de la sauvegarde
30 23 * * * /opt/backup/lance_backup.sh
```

2. Avec at

a. Présentation

La commande **at** et les commandes associées permettent une gestion des traitements batchs. Contrairement à la crontab les modifications sont volatiles : elles sont perdues lorsque la session est terminée. C'est à vous de placer la liste des commandes dans un éventuel fichier et de le charger au besoin via les scripts de votre profil.

Pour que at fonctionne le service **atd** (*at daemon*) doit fonctionner.

```
$ ps -ef | grep atd
at      7988  1 0 21:05 ?    00:00:00 /usr/sbin/atd
```

b. Formalisme

Pour simplifier, il y a deux moyens d'utiliser at :

- en lui passant de manière interactive une ligne de commande,
- en lui passant un fichier exécutable contenant les commandes à exécuter.

```
# ls -l /var/spool/atjobs/
-rwx----- 1 seb users 5620 mai 8 21:29 a000090133cf92
-rwx----- 1 seb users 5628 mai 8 21:30 a0000a0133d531
```

Si vous regardez le contenu de l'exécutable, vous voyez que votre commande n'est pas seule. Elle est située à la fin, mais le script positionne tout l'environnement lors de la création de l'entrée at.

```
#cat a0000a0133d531
#!/bin/sh
# atrun uid=1000 gid=100
# mail      seb 0
umask 22
LESSKEY=/etc/lesskey.bin; export LESSKEY
NNTPSERVER=news; export NNTPSERVER
INFODIR=/usr/local/info:/usr/share/info:/usr/info; export INFODIR
MANPATH=/usr/local/man:/usr/share/man:/opt/gnome/share/man; export
MANPATH
KDE_MULTIHEAD=false; export KDE_MULTIHEAD
... (environ 80 lignes) ...
cd /home/seb || {
    echo 'Execution directory inaccessible' >&2
    exit 1
}
echo salut !
```

La commande **atrm** permet de supprimer une tâche :

```
$ atrm 10
$ atrm 9
$ atq
```

d. Contrôle d'accès

Vous pouvez contrôler l'accès à la commande **at** par utilisateur avec les fichiers **/etc/at.allow** et **/etc/at.deny**.

- Si **at.allow** est présent, seuls les utilisateurs qui y sont explicitement indiqués peuvent utiliser **at**.
- Si **at.allow** est absent, **at** vérifie la présence d'un fichier **at.deny**. Tous les utilisateurs n'y étant pas sont autorisés à utiliser **at**. S'il est vide la commande **at** est autorisée pour tout le monde.
- Si les deux fichiers sont absents, seul root peut utiliser **at**.

D. Les traces (logs) du système

1. Principe

Lorsque le système démarre, fonctionne et effectue tout type d'action, ses actions et celles de la plupart de ses services sont tracées dans divers fichiers. Deux services sont spécialisés dans la réception des messages à écrire dans ces fichiers :

- **klogd**: *kernel log daemon*, chargé de la gestion des informations émises par le noyau.
- **syslogd**: *system log daemon*, chargé de la gestion des informations émises par tout type de service et éventuellement le noyau.

Certaines distributions utilisent maintenant **syslog-ng** dont les règles de traitement des messages, basées sur des expressions régulières, ont fortement évolué. Le principe reste cependant exactement le même.

Historiquement le service **syslogd** gérait aussi les messages émis par le noyau. Il en est toujours capable, mais la quantité de messages émis, les différents niveaux de严重性 et les nouvelles

méthodes d'accès aux messages du noyau font qu'il a semblé important et pertinent de séparer la gestion des messages du noyau de ceux émis par les services.

Les messages importants émis par un composant du système devraient passer par le service `syslogd`. Ceci n'empêche pas, au contraire, qu'un service puisse gérer ses propres traces dans ses propres fichiers. Les traces applicatives ne devraient pas être placées dans les traces de système. Les traces d'accès aux pages Web d'un serveur Apache n'ont rien à y faire. Par contre les traces de connexion au système (via la console, ssh, telnet, etc.) ont un intérêt important et doivent être présentes dans les fichiers logs du système.

Dans la suite de l'ouvrage, les traces seront appelées par leur nom d'usage courant : les **logs**.

2. Les messages

Le service `klogd` gère les messages émis par le noyau. Il dispose de deux sources d'accès aux messages :

- le système de fichiers virtuel `/proc`, utilisé par défaut s'il est présent, et notamment `/proc/kmsg` ;
- les appels systèmes via l'API du noyau, notamment `sys_syslog`, si `/proc` est absent ou si le paramètre `-s` a été passé à `klogd`.

Les messages du noyau ont des niveaux de priorité différents, étagés de 0 (haute priorité) à 7 (message de débogage) :

Niveau	Alias système	Signification
0	EMERG	Le système est inutilisable.
1	ALERT	Une action doit être prise immédiatement.
2	CRIT	Problème critique.
3	ERR	Erreur.
4	WARNING	Avertissement.
5	NOTICE	Normal mais nécessite une attention particulière.
6	INFO	Information standard.
7	DEBUG	Trace de débogage du noyau.

Le service `klogd` renvoie les messages de niveau 0 à 6 à `syslogd` qui redirigera ceux-ci dans les fichiers de logs et éventuellement sur les consoles concernées. Les informations de débogage de niveau 7 ne sont pas tracées par défaut.

Le service **syslogd** (ou `syslog-nd`) reçoit les messages issus des services mais aussi de `klogd`. Il les dispatche ensuite selon l'émetteur, la严重性, dans des fichiers, des consoles, sous forme de mails aux utilisateurs du système (root par exemple), etc.

Les actions les plus courantes sont l'écriture des logs dans des fichiers, la redirection de messages sur une console (la 10 ou la 12 bien souvent) ou l'envoi de messages à root.

3. Configuration de syslog

Le fichier de configuration `/etc/syslog.conf` permet de définir l'origine, l'importance et la destination de chaque message, sous forme de deux champs.

- L'origine définit en fait un ensemble de **systèmes** et de **sous-systèmes** (noyau, services). La liste, extensible, est composée à l'origine des éléments suivants. L'étoile définit l'ensemble des sous-systèmes.

Sous-système	Signification
<code>auth/authpriv</code>	Service de sécurité et d'authentification.
<code>cron</code>	Service cron.
<code>daemon</code>	Les démons du système.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



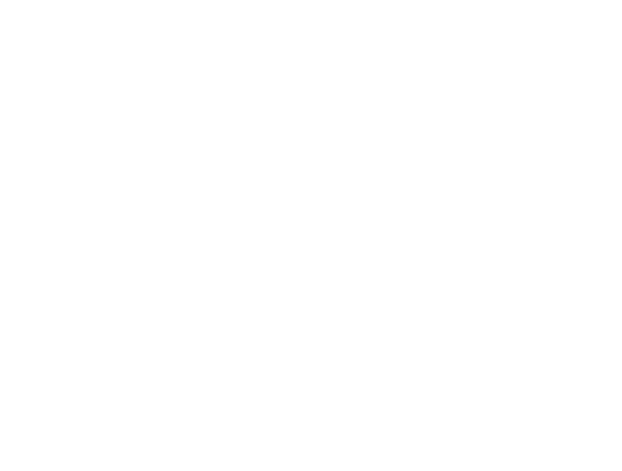
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

21 Quel fichier devez-vous modifier pour que tous les utilisateurs disposent à leur création du shell /bin/ksh ?
Le fichier /etc/default/useradd, ligne SHELL.

22 Quelle variable du fichier /etc/login.defs devez-vous modifier pour que tous les UID des utilisateurs démarrent à 1500 ?

UID_MIN doit passer à 1500.

23 Donnez l'ordre d'affichage des messages à la connexion locale :

- A - /etc/issue.net puis /etc/motd.
- B - /etc/motd puis /etc/issue.
- C - /etc/issue.net, /etc/issue puis /etc/motd.
- D - /etc/issue puis /etc/motd.

D. Il s'agit d'une connexion locale, donc /etc/issue.net n'est pas utilisé.

24 Quel est le répertoire par défaut qui contient le profil d'un utilisateur générique ?
/etc/skel ?

25 Quel fichier devez-vous créer pour interdire toute connexion sur un serveur Linux utilisant les PAM ?
/etc/nologin.

Impression

26 Quel est le nom du système d'impression généralement utilisé sous Linux ?

- A - BSD
- B - System V
- C - CUPS
- D - LPD

C : CUPS, Common Unix Printing System.

27 Avec les commandes System V, comment imprimer un fichier texte appelé fi.txt sur la file d'impression lj2100 ?
lpr -Plj2100 fic.txt

28 Que ce soit en System V ou BSD, quel est le nom du service d'impression ?

lpd.

29 Dans ce même cas, donnez le chemin du fichier de configuration des files d'impression.
/etc/printcap.

30 Quelles commandes System V et BSD fonctionnent avec CUPS ?

Toutes. CUPS garantit une compatibilité dans les deux cas.

31 Est-il conseillé de modifier le fichier /etc/cups/printers.conf à la main ?

Non, il est préférable de passer par l'interface HTML ou par un outil graphique.

32 Quelle est l'URL de configuration de CUPS sur la machine locale ?

http://localhost:631.

33 Est-il nécessaire de déclarer plusieurs files pour une imprimante si vous souhaitez modifier les options d'impression pour chacune des files ?

Non. C'était le cas avec LPD où chaque file avait ses propriétés et qu'il n'était pas possible facilement de changer de paramètres d'impression à volonté. C'est encore possible bien entendu mais les API de CUPS permettent la création de boîtes de dialogues de propriétés d'impression par file, comme sous Windows.

Automatisation

34 Quel est l'effet de la ligne crontab suivante * * * * * who > /tmp/test?

Toutes les minutes, le résultat de la commande who écrase l'ancien contenu de /tmp/test la liste des utilisateurs connectés localement.

35 Quelle ligne crontab écrire pour lancer la commande **checkall** du lundi au vendredi à 23h, seulement au mois de juin ?

- A - * 23 * 6 1,5 checkall
- B - 0 23 0 6 1-5 checkall
- C - 23 * 6 1-5 checkall
- D - 0 23 * 6 1-5 checkall

D. 0 23 : 23h00, * 6 1-5. *En juin, du lundi au vendredi.*

36 Donnez le chemin de la configuration crontab du système.

/etc/crontab.

37 Quelle commande permet d'éditer la crontab de l'utilisateur pierre ?

crontab -u pierre

38 Vous devez lancer exceptionnellement une tâche appelée **script.sh** ce soir à 21h00 depuis votre console. Quelle est la commande à taper ?

at -f script.sh 21:00

39 La tâche est annulée. Comment lister puis supprimer la tâche ?

atq pour lister, et atrm suivi du numéro de job pour le supprimer.

40 Comment interdire à un utilisateur d'utiliser la crontab ?

En rajoutant son login dans /etc/cron.deny.

Syslog

41 Quel est le nom du service qui trace les informations émises par le noyau ?

- A - syslogd.
- B - klogd.
- C - kerneld.
- D - dmesg.

B. C'est klogd qui trace les messages émis par le noyau.

42 syslogd peut-il traiter les messages émis par le noyau ?

Oui. Klogd reformate et renvoie les messages à syslogd.

43 Donnez le chemin du fichier de configuration de syslog.

/etc/syslog.conf.

44 syslog contient une ligne « *.* /var/log/messages ». Que contient /var/log/messages ?

/var/log/messages contient l'intégralité des messages émis par tous les composants (noyau, services et programmes) utilisant syslog.

45 Vous voulez que tous les messages liés à la sécurité des utilisateurs arrivent dans /var/log/secure. Quelle ligne devez-vous ajouter dans le fichier de configuration de syslog ?

auth.* /var/log/secure

Archivage

46 Est-il pertinent d'effectuer une sauvegarde totale tous les jours ?

En principe non. Vous pouvez étaler des sauvegardes complètes à intervalles réguliers puis effectuer une sauvegarde incrémentale entre les deux sauvegardes complètes.

47 Chez vous quelle est la méthode la plus fiable d'archivage ?

- A - Sur un CD/DVD.
- B - Sur un disque dur externe.
- C - Sur une clé USB.
- D - Sur une autre partition du disque dur.

A. Les autres supports ne sont pas pérennes, et en tout cas soumis à plus de risques.

```
# chage tom  
Changing agin information for tom  
    Minimum Password Age [7] : 10  
    Maximum Password Age [40] : 50  
    ...
```

6. Supprimez l'utilisateur tom. Lancez ensuite la commande **pwck**. Rectifiez le problème.

```
# userdel tom  
# pwck
```

La commande vous informe que le groupe cat contient un utilisateur tom qui n'existe plus. Vous devez donc retirer tom du groupe cat pour conserver la cohérence du système.

7. Prévenez vos utilisateurs que le système sera arrêté le 12 décembre à 17h00 pour maintenance. Vous pouvez utiliser le fichier **/etc/issue**, le fichier **/etc/motd** et le fichier **/etc/issue.net** pour y écrire votre texte.

2. L'impression

But : configurer les imprimantes, et imprimer des fichiers.

1. Vous avez le choix entre trois systèmes d'impression : BSD, System V et CUPS. Quel système devriez-vous utiliser ?

C'est CUPS car il est compatible avec les deux premiers.

2. Connectez-vous à l'interface d'administration de CUPS via un navigateur Web :
<http://localhost:631>

3. Sur l'une des imprimantes configurées, vérifiez l'état des impressions et imprimez une page de test.

4. Que se passe-t-il si vous tentez d'imprimer une image png ainsi : **lpr fichier.png** ?

Cela fonctionne parfaitement : le système d'impression CUPS dispose de filtres qui gèrent la plupart des documents en les transformant dans le langage d'impression de l'imprimante et notamment en postscript.

5. Annulez toutes les impressions en cours :

```
# lprm -
```

3 Automatisation des tâches

But : utiliser la crontab.

1. Exécutez la commande **ps -ef** pour l'utilisateur root toutes les 5 minutes et redirigez le résultat dans **/tmp/ps_result**, sans écraser les anciens :

```
# crontab -e root
```

Rajoutez la ligne suivante :

```
*/5 * * * * ps -ef >> /tmp/ps_result
```

2. Vérifiez la liste des tâches en crontab :

```
# crontab -l
```

3. Attendez cinq minutes et vérifiez le résultat dans **/tmp**.

4. Tous les fichiers placés dans **/etc/cron.d** sont interprétés par cron de manière automatique. Il est alors inutile, si la commande est permanente, de modifier la crontab de l'utilisateur root : rajoutez un fichier **/etc/cron.d/cron_ps** avec la ligne précédente dedans.

Chapitre 7

A. TCP/IP	416
B. Services réseaux xinetd	432
C. Connexion PPP	434
D. OpenSSH	439
E. Monter un serveur DHCP	441
F. Serveur DNS	443
G. Courrier électronique	452
H. Service HTTP Apache	454
I. Partage de fichiers	458
J. FTP	460
K. Partages Windows avec Samba	460
L. Validation des acquis : questions/réponses	463
M. Travaux pratiques	476

Pré-requis

- ☒ Avoir des connaissances de base sur le fonctionnement d'un réseau.
- ☒ Disposer d'une connexion Ethernet.
- ☒ Si possible, disposer d'une seconde machine, ou pouvoir accéder à d'autres machines du réseau.
- ☒ Avoir l'accès root.

Objectifs

À la fin de ce chapitre, vous serez en mesure :

- ☒ Rappels sur TCP/IP.
- ☒ De configurer les interfaces réseaux.
- ☒ De connaître les principales commandes.
- ☒ De paramétrier le resolver et les principaux fichiers.
- ☒ De mettre en place les services xinetd.
- ☒ D'établir une connexion PPP via un modem.
- ☒ D'utiliser les clients et serveurs SSH.
- ☒ De configurer et interroger un DNS.
- ☒ De configurer un serveur de courrier électronique basique.
- ☒ De configurer POP et IMAP.
- ☒ De créer un serveur FTP.
- ☒ De monter un serveur Apache basique.
- ☒ De partager vos fichiers avec NFS et Samba.

A. TCP/IP

1. Bases

L'origine de **TCP/IP** provient des recherches du **DARPA (Defense Advanced Research Project Agency)** qui débutent en 1970 et débouchent sur **ARPANET**. Dans les faits, le DARPA a financé l'université de Berkeley qui a intégré les protocoles de base de TCP/IP au sein de son système **UNIX BSD 4**.

TCP/IP s'est popularisé grâce à son interface générique de programmation d'échanges de données entre les machines d'un réseau, les primitives **sockets**, et l'intégration de protocoles applicatifs. Les protocoles de TCP/IP sont supervisés par l'**IAB (Internet Activities Board)** lui-même supervisant deux autres organismes :

- L'**IRTF (Internet Research Task Force)** qui est responsable du développement des protocoles.
- L'**IETF (Internet Engineering Task Force)** qui est responsable du réseau Internet.

Les adresses réseau sont distribuées par le **NIC (Network Information Center)** et en France l'**INRIA**. L'ensemble des protocoles de TCP/IP est décrit dans les documents **RFC (Request For Comments)** (voir le RFC 793).

- La couche inférieure est **IP (Internet Protocol)**.
- La couche de transport est **TCP (Transmission Control Protocol)** ou **UDP (User Datagram Protocol)**.

- Les couches supérieures sont les couches des protocoles applicatifs, par exemple :
 - **NFS (Network File System)** : partage de fichiers à distance.
 - **DNS (Domain Name System)** : association hôte->IP.
 - **FTP (File Transfer Protocol)** : transfert de fichiers.
 - **TELNET** : émulation d'un terminal de type texte...

La version du protocole IP représenté est la V4. Le futur, déjà présent, est le protocole IPV6. Compatible IPV4, il propose un adressage sur 128 bits (16 octets) permettant d'étendre les capacités du réseau notamment en matière de taille et d'adressage.

2. Adressage

a. Classes

Il est important de savoir avant l'installation dans quel type de réseau doit s'intégrer le nouveau serveur, TCP/IP bien sûr, mais il faut déjà lui réservé une adresse IP, un hostname (nom de machine réseau), connaître les diverses passerelles, le nom de domaine, la classe utilisée et le masque de sous-réseau netmask.

Voici un bref rappel sur les classes IP. Une adresse IP est définie sur 32 bits et représentée par quatre nombres séparés par des points : **n1.n2.n3.n4**. Cette adresse est constituée de deux parties qui définissent l'adresse réseau et l'hôte dans le réseau.

Distinguez, suivant les cas, quatre ou cinq classes d'adresses : A, B, C, D et E, mais seules les trois premières nous intéressent.

Légende : N et h sont des bits, N identifiant du réseau h identifiant de la machine.

Classe A : 0NNNNNNN hhhhhh hh hh hh hh soit 1.x.x.x à 126.x.x.x.
n1 est compris entre 1 et 126.
16777214 hôtes, 127 réseaux.

Classe B : 10NNNNNN NNNNNNNN hhhhhh hh hh hh hh soit de 128.0.x.x à 191.255.x.x.
n1 est compris entre 128 et 191.
65534 hôtes, 16382 réseaux.

Classe C : 110NNNNN NNNNNNNN NNNNNNNN hh hh hh hh soit de 192.0.0.x à 223.255.255.x.
n1 est compris entre 192 et 223.
254 hôtes, 2097150 réseaux.

Classe D : Commence par 1110, pour la multidiffusion IP.
Classe E : Commence par 1111, pour expérimentation.

Il existe des adresses d'hôtes qui ne peuvent pas être exploitées. Par exemple dans la classe C on ne peut avoir que 254 hôtes, alors que l'identifiant de la machine est codé sur 8 bits (donc 256 valeurs). C'est que l'adresse 0 représente l'adresse du réseau, et l'adresse 255 celle du **broadcast** (multidiffusion).

Notez que les adresses suivantes ne doivent pas être routées sur Internet et sont réservées aux réseaux locaux.

- 10.0.0.0 - 10.255.255.255 (10/8)
- 172.16.0.0 - 172.31.255.255 (172.16/12)
- 192.168.0.0 - 192.168.255.255 (192.168/16)

L'adresse 127.0.0.1 est l'adresse de loopback ou bouclage : elle représente la machine elle-même, ainsi que le sous-réseau 127.0.0.0/8.

b. Sous-réseaux

De plus, il est possible de découper ces réseaux en sous-réseaux à l'aide de masques permettant un découpage plus fin des adresses. Un **netmask** est un masque binaire qui permet de séparer immédiatement l'adresse du réseau et du sous-réseau de l'adresse de l'hôte dans l'adresse IP globale. Les masques prédéfinis sont :

- **Classe A** : 255.0.0.0
- **Classe B** : 255.255.0.0
- **Classe C** : 255.255.255.0

Pour communiquer directement entre eux les hôtes doivent appartenir à un même réseau ou sous-réseau. Calculer un sous-réseau est assez simple. Voici un exemple pour un réseau de classe C.

- Réseau : 192.168.1.0
- Adresse de réseau : 192.168.1.255
- Masque de réseau : 255.255.255.0

Calculer un masque de sous-réseau :

- Pour calculer le masque de sous-réseau, vous devez tout d'abord déterminer combien de machines vous souhaitez intégrer dans celui-ci. Un réseau de classe C permet d'intégrer 254 machines (0 et 255 étant réservés). Vous souhaitez créer des réseaux contenant 60 machines. Ajoutez 2 à cette valeur pour les adresses réservées (adresse du sous-réseau et adresse de broadcast) ce qui donne **62**.
- Une fois le nombre de machines déterminé, trouvez la puissance de deux exacte ou juste supérieure au nombre trouvé. 2 puissance 6 donne **64**.
- Écrivez le masque en binaire, placez tous les bits du masque de réseau de classe C à 1 et placez à 0 les 6 premiers bits du masque correspondant à la partie machine :
11111111 11111111 11111111 11000000
- Convertissez ce masque en décimal : **255.255.255.192**, et calculez l'ensemble des sous-réseaux possibles. Comme vous êtes dans un réseau de classe C, vous pouvez encore faire varier les deux derniers bits de la partie machine :
 - 00xxxxxx : 255.255.255.0
 - 01xxxxxx : 255.255.255.64
 - 10xxxxxx : 255.255.255.128
 - 11xxxxxx : 255.255.255.192
- Au final, vous obtenez quatre sous-réseaux de 62 machines, soit 248 machines. Vous tombez bien sur 256 si vous rajoutez les quatre adresses de broadcast et les quatre adresses de réseau.

c. Routage

Le masque de réseau permet de déterminer si une machine destinataire est sur le même réseau que vous ou non. Il faut indiquer le chemin que doivent prendre les paquets IP pour rejoindre leur destination. Si votre machine est un poste client disposant d'une seule carte réseau et que ce réseau ne comporte qu'un seul routeur (cas classique d'une connexion vers Internet) alors vous devez créer deux routes. La première est celle indiquant quelle carte réseau doivent emprunter les paquets pour accéder au reste du réseau (au sous-réseau), la seconde quelle route doivent emprunter les paquets pour sortir du réseau. Généralement, on parle de route par défaut quand un seul routeur est présent.

- Vers réseau1 -> utiliser interface réseau gauche.
- Vers réseau2 -> utiliser interface réseau droite.
- Vers autres -> utiliser interface réseau droite vers routeur1.



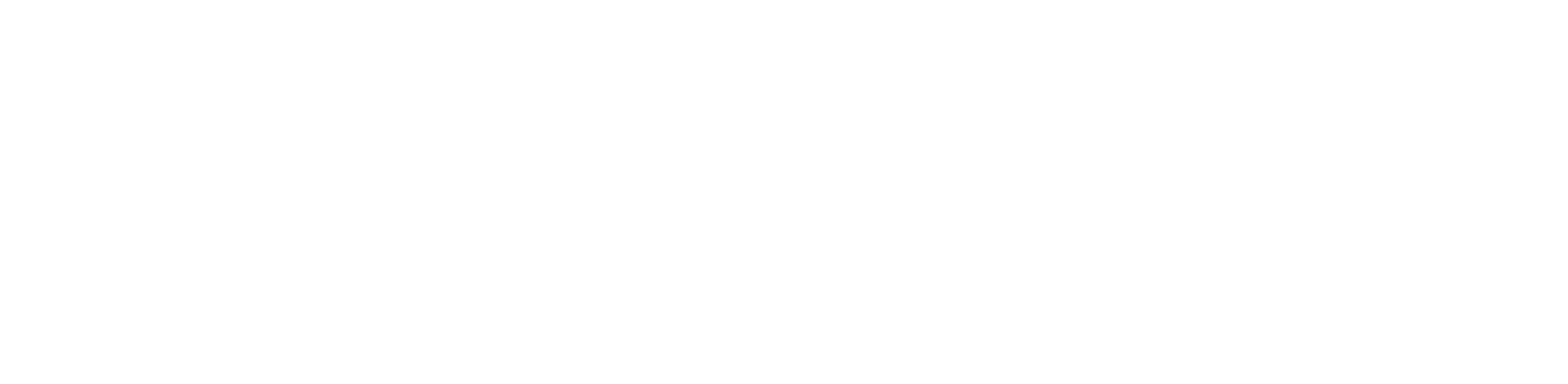
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

```
# netstat -ei
Table d'interfaces noyau
eth0    Lien encap:Ethernet HWaddr 00:XX:D3:XX:AA:XX
        inet adr:12.168.1.60 Bcast:192.168.1.255 Masque:255.255.255.0
              UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
              RX packets:2335314 errors:0 dropped:0 overruns:0 frame:0
              TX packets:678095 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 lg file transmission:1000
              RX bytes:1055212145 (1006.3 Mb) TX bytes:61264196 (58.4 Mb)
              Interruption:20 Adresse de base:0x8c00

lo     Lien encap:Boucle locale
        inet adr:127.0.0.1 Masque:255.0.0.0
              UP LOOPBACK RUNNING MTU:16436 Metric:1
              RX packets:1109 errors:0 dropped:0 overruns:0 frame:0
              TX packets:1109 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 lg file transmission:0
              RX bytes:60423 (59.0 Kb) TX bytes:60423 (59.0 Kb)
```

Le paramètre **-r** permet d'obtenir, comme route, les tables de routage. Ajoutez le paramètre **-n** pour indiquer les IPs à la place des noms.

```
# netstat -rn
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic   MSS Fenêtre irtt
Iface
192.168.211.0  0.0.0.0        255.255.255.0   U       0 0       0
vmnet8
192.168.1.0    0.0.0.0        255.255.255.0   U       0 0       0
eth0
172.16.248.0   0.0.0.0        255.255.255.0   U       0 0       0
vmnet1
169.254.0.0    0.0.0.0        255.255.0.0    U       0 0       0
eth0
127.0.0.0      0.0.0.0        255.0.0.0     U       0 0       0
lo
0.0.0.0         192.168.1.1    0.0.0.0        UG      0 0       0
eth0
```

Le paramètre **-a** permet de visualiser toutes les connexions, pour tous les protocoles, y compris les ports en écoute de la machine. La sortie est trop longue pour être placée dans ces pages.

```
# netstat -a | wc -l
495
```

Le paramètre **-A** permet de spécifier le protocole visible : **inet**, **unix**, **ipx**, **ax25**, **netrom** et **ddp**.

```
# netstat -a -A inet
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale           Adresse distante
Estat
tcp      0      0 localhost:716             *:*
LISTEN
tcp      0      0 *:sunrpc                *:*
LISTEN
tcp      0      0 localhost:ipp            *:*
LISTEN
tcp      0      0 localhost:smtp            *:*
```

```

LISTEN
tcp      0      0 localhost:hpssd          *:*
LISTEN
tcp      0      0 slyserver:41851        imap.free.fr:imap
ESTABLISHED
tcp      0      0 slyserver:41850        imap.free.fr:imap
ESTABLISHED
tcp      0      0 slyserver:54220        by1msg4176111.gate:msnp
ESTABLISHED
tcp      0      0 slyserver:34267        by2msg2105007.phx.:msnp
ESTABLISHED
tcp      0      0 slyserver:47990        by1msg4082314.phx.:msnp
ESTABLISHED
udp      0      0 *:filenet-tms         *:*
udp      0      0 *:mdns                *:*
udp      0      0 *:sunrpc              *:*
udp      0      0 *:ipp                 *:*
udp      0      0 172.16.248.1:ntp       *:*
udp      0      0 192.168.211.1:ntp       *:*
udp      0      0 slyserver:ntp          *:*
udp      0      0 localhost:ntp          <+> *:*
udp      0      0 *:ntp                 *:*
raw     0      0 *:icmp               *:*

```

7

Enfin le paramètre **-p** permet d'indiquer, quand c'est possible, le PID et le nom du processus.

```

# netstat -A inet -p
Connexions Internet actives (sans serveurs)
Proto Recv-Q Send-Q Adresse locale           Adresse distante
Proto Recv-Q Send-Q Adresse locale           Adresse distante
Etat      PID/Program name
Etat      PID/Program name
...
tcp      0      0 slyserver:54220        by1msg4176111.gate:msnp
ESTABLISHED 4041/kopete
tcp      0      0 slyserver:34267        by2msg2105007.phx.:msnp
ESTABLISHED 4041/kopete
tcp      0      0 slyserver:47990        by1msg4082314.phx.:msnp
ESTABLISHED 4041/kopete

```

g. IPtraf

La commande **iptraf** permet de visualiser en temps réel l'activité du réseau via un outil texte interactif ou non (ligne de commande). Les menus sont clairs. Vous vous y déplacez avec les touches fléchées et les divers raccourcis précisés.

La capture suivante montre l'affichage détaillé des statistiques de trafic de la carte eth0. Cet écran est accessible via la ligne de commande avec :

```
# iptraf -d eth0
```

```

compressnet      3/udp   # Compression Process
rje             5/tcp    # Remote Job Entry
rje             5/udp   # Remote Job Entry
echo            7/tcp    Echo
echo            7/udp   Echo
discard         9/tcp    # Discard
discard         9/udp   # Discard
systat          11/tcp   users      # Active Users
systat          11/udp   users      # Active Users
daytime         13/tcp   # Daytime (RFC 867)
daytime         13/udp   # Daytime (RFC 867)
netstat          15/tcp   # Unassigned [was netstat]
qotd            17/tcp   quote     # Quote of the Day
qotd            17/udp   quote     # Quote of the Day
msp              18/tcp   # Message Send Protocol
msp              18/udp   # Message Send Protocol
chargen          19/tcp   # Character Generator
chargen          19/udp   # Character Generator
ftp-data         20/tcp   # File Transfer [Default Data]
ftp-data         20/udp   # File Transfer [Default Data]
ftp              21/tcp   # File Transfer [Control]
fsp              21/udp   # File Transfer [Control]
ssh              22/tcp   # SSH Remote Login Protocol
ssh              22/udp   # SSH Remote Login Protocol
telnet           23/tcp   # Telnet
telnet           23/udp   # Telnet
...

```

e. /etc/protocols

Le fichier **/etc/protocols** contient la liste des protocoles connus par Unix.

#	Assigned Internet Protocol Numbers			
#	Decimal	Keyword	Protocol	References
#	-----	-----	-----	-----
#	protocol	num aliases	# comments	
hopopt	0	HOPOPT	# IPv6 Hop-by-Hop Option	[RFC1883]
icmp	1	ICMP	# Internet Control Message	[RFC792]
igmp	2	IGMP	# Internet Group Management	[RFC1112]
ggp	3	GGP	# Gateway-to-Gateway	[RFC823]
ip	4	IP	# IP in IP (encapsulation)	[RFC2003]
st	5	ST	# Stream	[RFC1190, RFC1819]
tcp	6	TCP	# Transmission Control	[RFC793]
cbt	7	CBT	# CBT	[Ballardie]
egp	8	EGP	# Exterior Gateway Protocol	[RFC888, DLM1]
igp	9	IGP	# any private interior gateway	[IANA]
bbn-rcc-mon	10	BBN-RCC-MON	# BBN RCC Monitoring	[SGC]
nvp-ii	11	NVP-II	# Network Voice Protocol	[RFC741, SC3]
pup	12	PUP	# PUP	[PUP, XEROX]
argus	13	ARGUS	# ARGUS	[RWS4]
emcon	14	EMCON	# EMCON	[BN7]
xnet	15	XNET	# Cross Net Debugger	[IEN158, JFH2]
chaos	16	CHAOS	# Chaos	[NC3]
udp	17	UDP	# User Datagram	[RFC768, JBP]
#				

B. Services réseaux xinetd

1. Présentation

Le démon **xinetd** est un « super-service » permettant de contrôler l'accès à un ensemble de services, **telnet** par exemple. Beaucoup de services réseaux peuvent être configurés pour fonctionner avec xinetd, comme les services **ftp**, **ssh**, **samba**, **rcp**, **http**, etc. Des options de configuration spécifiques peuvent être appliquées pour chaque service géré.

Lorsqu'un hôte client se connecte à un service réseau contrôlé par xinetd, xinetd reçoit la requête et vérifie tout d'abord les autorisations d'accès TCP (voir **tcp_wrappers** au prochain chapitre) puis les règles définies pour ce service (autorisations spécifiques, ressources allouées, etc.). Une instance du service est alors démarrée et lui cède la connexion. À partir de ce moment **xinetd** n'interfère plus dans la connexion entre le client et le serveur.

2. Configuration

Les fichiers de configuration sont :

- **/etc/xinetd.conf** : configuration globale
- **/etc/xinetd.d/*** : répertoire contenant les fichiers spécifiques aux services. Il existe un fichier par service, du même nom que celui précisé dans **/etc/services**.

```
$ ls -l /etc/xinetd.d
total 92
-rw-r--r-- 1 root root 313 sep 22 2007 chargen
-rw-r--r-- 1 root root 333 sep 22 2007 chargen-udp
-rw-r--r-- 1 root root 256 mar 20 22:11 cups-lpd
-rw-r--r-- 1 root root 409 nov 4 2005 cvs
-rw-r--r-- 1 root root 313 sep 22 2007 daytime
-rw-r--r-- 1 root root 333 sep 22 2007 daytime-udp
-rw-r--r-- 1 root root 313 sep 22 2007 discard
-rw-r--r-- 1 root root 332 sep 22 2007 discard-udp
-rw-r--r-- 1 root root 305 sep 22 2007 echo
-rw-r--r-- 1 root root 324 sep 22 2007 echo-udp
-rw-r--r-- 1 root root 492 sep 22 2007 netstat
-rw-r--r-- 1 root root 207 avr 23 19:04 rsync
-rw-r--r-- 1 root root 337 fév 17 14:22 sane-port
-rw-r--r-- 1 root root 332 sep 22 2007 servers
-rw-r--r-- 1 root root 334 sep 22 2007 services
-rw-r--r-- 1 root root 351 jun 21 2007 svnserve
-rw-r--r-- 1 root root 277 nov 8 2007 swat
-rw-r--r-- 1 root root 536 sep 21 2007 systat
-rw-r--r-- 1 root root 387 fév 4 10:11 tftp.rpmsave
-rw-r--r-- 1 root root 339 sep 22 2007 time
-rw-r--r-- 1 root root 333 sep 22 2007 time-udp
-rw-r--r-- 1 root root 2304 avr 4 11:39 vnc
-rw----- 1 root root 768 sep 22 2007 vsftpd
```

Contenu de xinetd.conf :

```
defaults
{
    instances          = 60
    log_type          = SYSLOG authpriv
    log_on_success    = HOST PID
    log_on_failure    = HOST
    cps               = 25 30
}
includedir /etc/xinetd.d
```

- **instances** : nombre maximal de requêtes qu'un service xinetd peut gérer à un instant donné.
- **log_type** : dans notre cas, les traces sont gérées par le démon **syslog** via **authpriv** et les traces sont placées dans **/var/log/secure**. **FILE /var/log/xinetd** aurait placé les traces dans **/var/log/xinetd**.
- **log_on_success** : xinetd va journaliser l'événement si la connexion au service réussit. Les informations tracées sont l'hôte (**HOST**) et le **PID** du processus serveur traitant la connexion.
- **log_on_failure** : idem mais pour les échecs. Il devient simple de savoir quels hôtes ont tenté de se connecter si par exemple la connexion n'est pas autorisée.
- **cps** : xinetd n'autorise que 25 connexions par secondes à un service. Si la limite est atteinte, xinetd attendra 30 secondes avant d'autoriser à nouveau les connexions.
- **includedir** : inclut les options des fichiers présents dans le répertoire indiqué.

Exemple /etc/xinetd.d/telnet :

```
# default: on
# description: The telnet server serves telnet sessions; it uses \
#               unencrypted username/password pairs for authentication.
service telnet
{
    disable = no
    flags      = REUSE
    socket_type = stream
    wait       = no
    user       = root
    server     = /usr/sbin/in.telnetd
    log_on_failure += USERID
}
```

La première ligne en commentaire, **default**, a une importance particulière. Elle n'est pas interprétée par xinetd mais par **ntsysv** ou **chkconfig** pour déterminer si le service est actif.

- **service** : nom du service qui correspond à un service défini dans **/etc/services**.
- **flags** : attributs pour la connexion. **REUSE** indique que la socket sera réutilisée pour une connexion telnet.
- **socket_type** : spécifie le type de socket. Généralement **stream** (tcp) ou **dgram** (udp). Une connexion directe IP se fait par **raw**.
- **wait** : indique si le serveur est single-threaded (yes) ou multi-threaded (no).
- **user** : sous quel compte utilisateur le service sera lancé.
- **server** : chemin de l'exécutable devant être lancé.
- **log_on_failure** : le **+=** indique qu'on rajoute l'option associée au fichier de trace en plus de celles par défaut. Ici : le login.
- **disable** : indique si le service est actif ou non.

Certaines options peuvent améliorer les conditions d'accès et la sécurité :

- **only_from** : permet l'accès uniquement aux hôtes spécifiés.
- **no_access** : empêche l'accès aux hôtes spécifiés (ex : 172.16.17.0/24).
- **access_times** : autorise l'accès uniquement sur une plage horaire donnée (ex : 09:00-18:30).

3. Démarrage et arrêt des services

On distingue deux cas.

Premier cas, le service **xinetd** est un service comme un autre dont le démarrage ou l'arrêt peut s'effectuer avec la commande **service** ou directement via l'exécution de **/etc/init.d/xinetd**.

```
# service xinetd start
```

Dans ce cas, la commande **chkconfig** (Red Hat, openSUSE) autorise ou non le lancement du service au démarrage pour chaque niveau d'exécution (runlevel).

```
# chkconfig --level 345 xinetd on
```

Second cas, comme xinetd gère plusieurs services, l'arrêt de xinetd arrête tous les services associés, et le démarrage de xinetd lance tous les services associés. Il n'est pas possible de choisir quels services de xinetd seront lancés dans tel ou tel niveau d'exécution. Mais vous pouvez choisir d'activer ou de désactiver simplement un service avec chkconfig.

```
# chkconfig telnet on
```

C. Connexion PPP

1. Choix et réglage du modem

a. Le cas des Winmodems

Tous les modems RTC (analogiques) se connectant sur un port série (externe), ou émulant un vrai port série (carte PCI ou via le port USB) sont entièrement supportés sous Linux.

Cependant, il existe une catégorie particulière de modems appelés les **winmodems**. Ils se présentent parfois comme des « vrais » modems (ils leur ressemblent parfois). D'une manière générale, fuyez ce type de modems. Cependant quelques modèles sont connus pour fonctionner sous Linux. Rendez-vous sur le site <http://linmodems.org/> pour obtenir des informations à ce sujet.

D'autres adaptateurs que les modems RTC sont reconnus par Linux comme des modems ; c'est le cas de quelques adaptateurs ADSL, mais aussi des téléphones portables reliés par une câble USB ou via une connexion Bluetooth.

b. Les fichiers périphériques

Les ports série de type RS232 se nomment **ttySn** :

- **/dev/ttys0** : premier port série.
- **/dev/ttys1** : second port série.
- etc.

Les ports série de type USB se nomment **ttyUSBn** : **/dev/ttys0**, et ainsi de suite.

Les ports de communication série via bluetooth se nomment **rfcommn** (pour radio frequency communication) : **/dev/rfcomm0**, et ainsi de suite.

Pour utiliser les ports série et établir une communication, vous devez pouvoir écrire sur les périphériques (pour envoyer les ordres) et donc soit avoir les droits correspondants, soit utiliser un programme SUID, ou encore disposer de règles udev adaptées.

c. Régler le port série

Les ports série se gèrent via la commande **setserial**.

La commande **setserial** permet d'interroger la configuration d'un port série avec le paramètre **-g**. Le port série **ttys0** est de type **16550A** (le plus rapide), utilise l'**IRQ 4** et le port d'adresse **0x03f8**.

```
# setserial -g /dev/ttys0  
/dev/ttys0, UART: 16550A, Port: 0x03f8, IRQ: 4
```

Les informations n'étant pas suffisantes, rajoutez le paramètre **-a**. Vous obtenez entre autres la vitesse de la ligne qui est de **115200** bits par seconde (le mot **baud** est à proscrire).

```
# setserial -a -g /dev/ttys0  
/dev/ttys0, Line 0, UART: 16550A, Port: 0x03f8, IRQ: 4  
      Baud_base: 115200, close_delay: 50, divisor: 0  
      closing_wait: 3000  
      Flags: spd_normal skip_test
```

Setserial permet aussi de configurer le port. En interrogeant le port, vous pouvez savoir quels paramètres ont permis son réglage avec le -G :

```
# setserial -G /dev/ttyS0  
/dev/ttyS0 uart 16550A port 0x03f8 irq 4 baud_base 115200 spd_normal  
skip_test
```

De là, il est possible d'extrapoler de nouvelles valeurs. Par exemple passez ainsi le port série à une vitesse de 57600 bps :

```
# setserial /dev/ttyS0 baud_base 57600  
# setserial -a -g /dev/ttyS0  
/dev/ttyS0, Line 0, UART: 16550A, Port: 0x03f8, IRQ: 4  
    Baud_base: 57600, close_delay: 50, divisor: 0  
    closing_wait: 3000  
    Flags: spd_normal skip_test
```

d. Les commandes AT

Les modems utilisent tous un jeu de commandes standard appelées **commandes AT**. Leur vrai nom est **Commandes Hayes**, du nom de la société les ayant inventées. AT signifie Attention. Le modem attend après ces premières lettres une suite permettant de le configurer, de numérotter, de raccrocher, etc. Le jeu est en principe standard mais la configuration varie d'un modèle à un autre. Si vous ne rentrez pas dans les détails, une configuration générique suffit et fonctionne pour la quasi-totalité des modems. Comme il n'est pas possible de décrire les commandes AT ici, vous en trouverez une liste sur le site 3com (qui a racheté US Robotics, le meilleur fabricant de modems) : <http://www.usr.com/support/3cp3056/3cp3056-french-ug/atcoms.htm>. En voici tout de même quelques-unes :

- Numéroter : ATDT0102030405
- Répondre : ATA
- Raccrocher : ATH

2. PPP

Le protocole **PPP** (*Point to Point Protocol*) permet de vous relier à une autre machine afin d'y accéder, ou à son réseau, ce qui est souvent le cas d'Internet. Aujourd'hui encore et malgré les nombreuses solutions proposées par le câble ou l'ADSL, certaines connexions se font encore via un modem RTC (modem classique connecté sur port USB, série ou interne) relié à une prise téléphonique classique.

L'établissement d'une liaison PPP nécessite :

- Un client disposant des outils ppp (pppd) et chat pour dialoguer avec le serveur.
- Un serveur disposant de pppd et des moyens de fournir une adresse IP (dhcp).
- Un modem.

La suite ne prend en considération que la partie cliente.

Vous devez connaître :

- Le port série sur lequel est branché votre modem : ttySX (série ou USB), ttyACMX (usb), rfcommX (bluetooth), etc.
- Le numéro d'appel de votre fournisseur d'accès Internet (FAI).
- Le nom d'utilisateur et le mot de passe chez votre FAI.
- L'adresse du serveur DNS de votre FAI.

☞ Le modem n'est pas forcément RTC. Un téléphone portable reconnu comme modem via le câble de connexion au PC ou depuis le protocole Bluetooth fait un excellent modem. Pour peu qu'il soit aux normes 3G ou Edge, les débits peuvent être très impressionnantes. Un grand nombre de connexions ADSL sont aussi effectuées via le protocole PPP. Dans ce cas la suite s'applique mais des modifications sont à prévoir.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

```

noauth

maxfail 10
lcp-echo-interval 5
lcp-echo-failure 12
holdoff 3
noaccomp noccp nobsdcomp nodeflate nopcomp novj novjccomp
lock
crtscts

```

Chaque ligne contient au moins une instruction dont voici les plus pertinentes :

- **/dev/modem** : le périphérique de connexion (le modem) ;
- **connect** : la chaîne de connexion envoyée au FAI ;
- **defaultroute** : la route par défaut est remplacée par celle fournie par le FAI ;
- **noipdefault** : le FAI fournit l'IP par son DHCP ;
- **usepeerdns** : récupère les informations DNS du FAI ;
- **115200** : la vitesse de communication du périphérique (elle sera négociée) ;
- **debug** : fournit le détail complet de la connexion ;
- **noauth** : ce n'est pas le script ppp qui établit l'authentification (voir ligne connect) ;
- **maxfail** : n tentatives de connexion avant d'abandonner ;
- **holdoff** : attente de n secondes entre deux connexions ;
- **lock** : permet l'accès exclusif au fichier périphérique ;
- **crtscts** : active le contrôle de flux matériel.

Soit le second fichier `/etc/ppp/peers/cnx1-chat` utilisé par la ligne **connex** du premier fichier `/etc/ppp/peers/cnx1` :

```

# cat /etc/ppp/peers/cnx1-chat
ABORT ERROR
ABORT "NO CARRIER"
ABORT BUSY "" ATZ
OK ATDT0102030405
CONNECT ""
login: "login"
word: "password"

```

Il ne s'agit que d'un exemple. Vous devez vérifier tant du côté de votre FAI que du côté de la documentation de votre modem quelles sont les bonnes commandes AT à passer (elles sont généralement standard).

c. Connexion

Initialisez la connexion :

```
# pppd call cnx1
```

Vous devriez voir les leds de votre modem clignoter, et si le haut-parleur est activé le bruit caractéristique se fait entendre. Si la connexion est établie, une nouvelle interface réseau apparaît : **ppp0**.

```

# ifconfig ppp0
ppp0      Link encap:Point-Point Protocol
          inet addr:10.xx.yy.zz  P-t-P:10.xx.yy.zz Mask:255.255.255.0
          UP POINTOPOINT RUNNING MTU:552 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0

```

d. Connexion par front-end

Plutôt que d'établir une connexion depuis les commandes en ligne et des fichiers de configuration, il existe plusieurs outils graphiques, front-ends à PPP, qui permettent à la fois de configurer le modem et d'établir une connexion PPP avec tous les réglages réseaux possibles.

KDE est un très bon exemple avec l'outil **kppp**. L'interface principale ressemble à ce que vous avez peut-être pu déjà connaître sous Windows :

- le choix d'un profil de connexion,
- un nom d'utilisateur,
- un mot de passe,
- et les boutons, dont celui permettant de se connecter.



kppp permet d'établir une connexion PPP

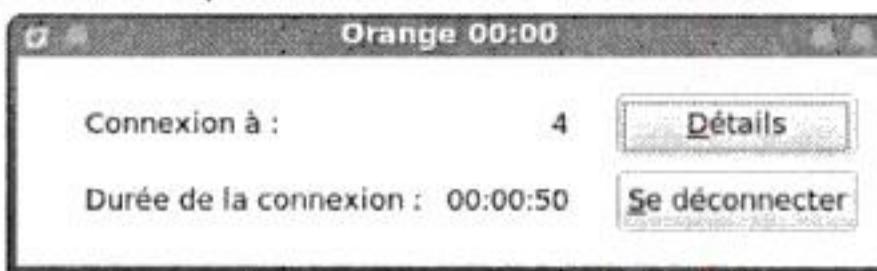
La configuration se fait en trois étapes :

- configuration du modem (la capture représente un téléphone portable 3G via une connexion bluetooth),
- configuration des informations de connexion : profil, numéro de téléphone, sauvegarde du mot de passe, coût, etc.,
- configuration des paramètres réseaux : réglages du DNS, du client DHCP, de la passerelle, etc.



Configuration du modem avec kppp

Enfin, une fois connecté une fenêtre d'état vous permet d'obtenir des informations détaillées sur la connexion, sa durée et vous offre la possibilité de vous déconnecter.



La liaison ppp est établie.

Les distributions sont souvent accompagnées d'outils de type NetworkManager qui permettent une connexion ethernet, Wi-Fi ou ppp à la volée en quelques clics. La distribution Mandriva mérite une mention spéciale : la configuration d'une connexion Internet via un mobile 3G au travers du protocole Bluetooth s'est fait en quelques secondes, l'outil Drakconf ayant tout détecté seul.

D. OpenSSH

1. Présentation

OpenSSH est un protocole de shell sécurisé, un mécanisme qui permet l'authentification sécurisée, l'exécution à distance et la connexion à distance. Il permet aussi le transport sécurisé du protocole X Window. En fait, il est capable d'encapsuler des protocoles non sécurisés en redirigeant les ports.

Les packages à utiliser pour un serveur sont **openssh**, **openssl** et **openssh-clients**. Pour X on rajoute les packages **openssh-askpass*** (il peut y en avoir plusieurs suivant l'environnement de bureau). La liste des packages à installer dépend de chaque distribution.

L'utilisation la plus commune reste l'accès distant sécurisé à une machine via le client ssh.

2. Configuration

La configuration est /etc/ssh/sshd_config. Quelques options sont éventuellement à modifier :

- **Port** : le numéro de port, par défaut 22 ;
- **Protocol** : fixé à 2,1 il autorise SSH1 et SSH2. On préférera SSH2 et donc on laissera la valeur 2 seule ;
- **ListenAddress** : par défaut ssh écoute sur toutes les IP du serveur. On peut autoriser uniquement l'écoute sur une interface donnée ;
- **PermitRootLogin** : ssh autorise les connexions de root. On peut placer la valeur à « no ». Dans ce cas, il faudra se connecter en simple utilisateur et passer par **su** ;
- **Banner** : chemin d'un fichier dont le contenu sera affiché aux utilisateurs lors de la connexion.

Ssh est un service System V à lancer avec service ou directement par /etc/init.d/sshd.

```
# service sshd start
```

3. Utilisation

La commande **ssh** permet d'établir une connexion.

```
$ ssh -l login host  
$ ssh login@host
```

L'option -X permet d'activer la redirection (forwarding) du protocole X Window.

```
$ ssh -X login@host
```

4. Clés et connexion automatique

Il est possible d'établir une connexion automatique vers une autre machine sans saisir de mot de passe. Pour cela, il est nécessaire depuis le compte utilisateur du client (la machine qui va se connecter) de générer une paire de clés, privée et publique. Aucune passphrase ne doit être saisie.

Du côté du serveur ssh, la clé publique du client doit être placée dans un fichier contenant les clés autorisées à se connecter dans le compte de destination.

a. Côté client

- Générez une clé au format RSA avec la commande **ssh-keygen** :

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/bean/.ssh/id_rsa):
Created directory '/home/bean/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/bean/.ssh/id_rsa.
Your public key has been saved in /home/bean/.ssh/id_rsa.pub.
The key fingerprint is:
f6:39:23:4e:fa:53:d0:4e:65:7f:3f:fd:a3:f4:8e:2a bean@p64p17bicb3
```

- Le répertoire de l'utilisateur contient maintenant un répertoire .ssh :

```
$ cd .ssh
$ ls
id_rsa  id_rsa.pub
```

- Le fichier **id_rsa.pub** contient la clé publique :

```
$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEArc/VskR9v708J2EDG1LM1Q6HmKJc
P2UenurnSr7rWTSZK5w9Hzn4DCz5iMzLAPc4659I0uKJbmF3vBXozIgLrCdCZCQE
hhPLwJVLXbGNc81MF742E/WqkkJ/uQYb31iPAU7Efosei+DVZ21No725XjiSCZ2q
zKKx7ZuNQEtxW0eVkwvlA0u7Hvrwn+FQksW3NXwTxwHhudSw7S6kIC3tyF5rkzf
vu7zQbOGDGGPiF3aOvd0oSBNgijTz+M0PaoXXI3brMd66WkGfSwf4ofYKNDCA/3T
Q4xU6WxkxqTBcsjEm1gIymFAyxDo+zzf63jxLG08Pp50DKf7DUqBx7+rjw==
bean@slyserver
```

b. Côté serveur

- Allez dans le répertoire .ssh du compte auquel vous souhaitez accéder sur le serveur (créez-le s'il n'existe pas) :
- ```
$ cd /home/seb/.ssh
```
- Éditez le fichier **authorized\_keys2** (créez-le s'il n'existe pas) et copiez-y sur une nouvelle ligne le contenu du fichier **id\_rsa.pub** du client. Sauvez.

```
$ echo "ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEArc/VskR9v708J2EDG1LM1Q6HmKJcP2Uenurn
Sr7rWTSZK5w9Hzn4DCz5iMzLAPc4659I0uKJbmF3vBXozIgLrCdCZCQEhhPLwJVL
XbGNc81MF742E/WqkkJ/uQYb31iPAU7Efosei+DVZ21No725XjiSCZ2qzKKx7ZuN
QEtxW0eVkwvlA0u7Hvrwn+FQksW3NXwTxwHhudSw7S6kIC3tyF5rkzfkuu7zQbOG
DGDPiF3aOvd0oSBNgijTz+M0PaoXXI3brMd66WkGfSwf4ofYKNDCA/3TQ4xU6Wxk
xqTBcsjEm1gIymFAyxDo+zzf63jxLG08Pp50DKf7DUqBx7+rjw== bean@slyser
ver" > authorized_keys2
```

- Tentez une connexion, le mot de passe n'est pas demandé :

```
$ ssh seb@slyserver
```

## F. Serveur DNS

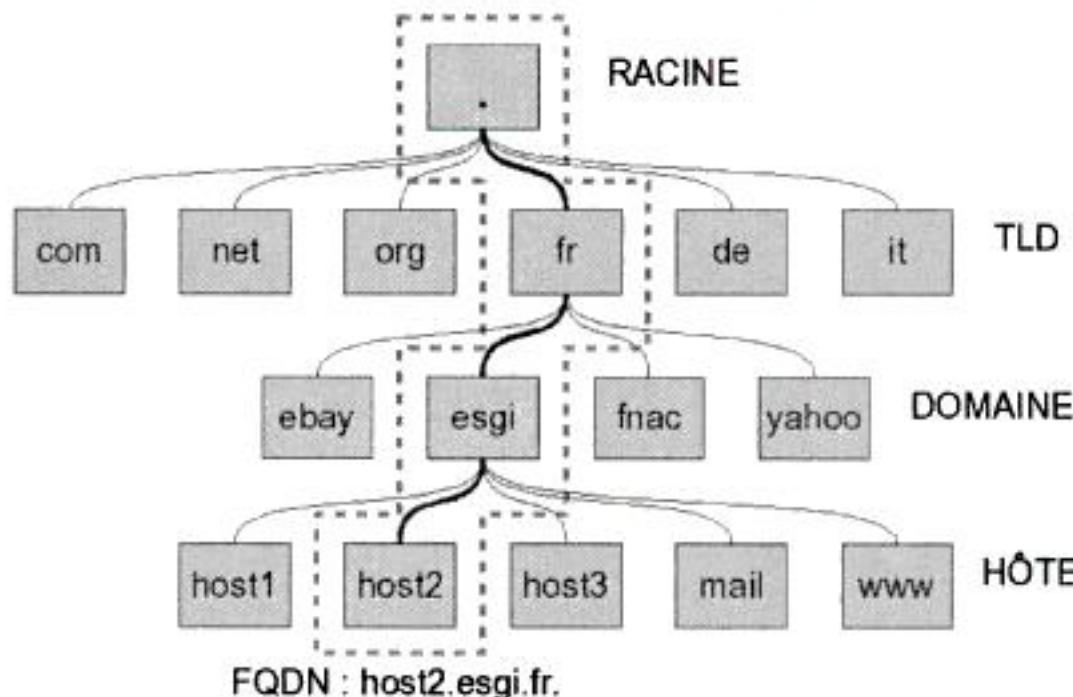
### 1. Présentation

Le Système de Noms de Domaines **DNS** (*Domain Name System*) transforme les noms d'hôte en adresses IP : c'est la **Résolution de nom**. Il transforme les adresses IP en noms d'hôte : c'est la **Résolution inverse**. Il permet de regrouper les machines par domaines de nom. Il fournit des informations de routage et de courrier électronique.

Le DNS permet de faire référence à des systèmes basés sur IP (les *hôtes*) à l'aide de noms conviviaux (les *noms de domaines*). L'intérêt d'un DNS est évident. Les noms de domaine sont plus simples à retenir, et si son adresse IP change l'utilisateur ne s'en rend même pas compte. On comprend que le DNS est un service clé critique pour Internet.

Les noms de domaine sont séparés par des points, chaque élément pouvant être composé de 63 caractères ; il ne peut y avoir qu'un maximum de 127 éléments et le nom complet ne doit pas dépasser 255 caractères. Le nom complet non abrégé est appelé **FQDN** (*Fully Qualified Domain Name*). Dans un FQDN, l'élément le plus à droite est appelé **TLD** (*Top Level Domain*), celui le plus à gauche représente l'hôte et donc l'adresse IP.

Le DNS contient une configuration spéciale pour les routeurs de courrier électronique (définitions MX) permettant une résolution inverse, un facteur de priorité et une tolérance de panne.



*Représentation d'une arborescence DNS*

Une zone est une partie d'un domaine gérée par un serveur particulier. Une zone peut gérer un ou plusieurs sous-domaines, et un sous-domaine peut être réparti en plusieurs zones. Une zone représente l'unité d'administration dont une personne peut être responsable.

### 2. Lancement

Le service s'appelle **named**.

```
service named start
```

Ou :

```
/etc/init.d/named start
```

### 3. Configuration de Bind

**Bind** (*Berkeley Internet Name Daemon*) est le serveur de noms le plus utilisé sur Internet. Bind 9 supporte l'IPv6, les noms de domaine unicode, le multithread et de nombreuses améliorations de sécurité.

### a. Configuration générale

La configuration globale de Bind est placée dans le fichier `/etc/named.conf`. La configuration détaillée des zones est placée dans `/var/lib/named`. `/etc/named.conf` est composé de deux parties. La première concerne la configuration globale des options de Bind. La seconde est la déclaration des zones pour les domaines individuels. Les commentaires commencent par un `#` ou `//`.

**Attention** il arrive parfois (notamment sur RHEL 4.x) que la configuration de Bind soit « chrootée » (déplacée dans une arborescence spécifique d'où le service ne peut sortir, le reste de l'arborescence lui étant inaccessible). Sur Centos et RHEL 4.x et supérieurs `named.conf` est dans `/var/named/chroot/etc/`. On peut modifier ce mode en modifiant le fichier de configuration `/etc/sysconfig/named`.

```
cat /etc/sysconfig/named
...
CHROOT=/var/named/chroot
...
```

Dans ce cas, tous les fichiers de configuration, y compris les zones, sont relatifs à ce chemin. Voici un fichier `named.conf` de base.

```
options {
 directory "/var/lib/named";
 forwarders { 10.0.0.1; };
 notify no;
};

zone "localhost" in {
 type master;
 file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" in {
 type master;
 file "127.0.0.zone";
};

zone "." in {
 type hint;
 file "root.hint";
};
```

### b. Section globale

La configuration globale est placée dans la section **options**. Voici un détail de quelques options importantes (le point-virgule doit être précisé) :

- **directory "filename";** : emplacement des fichiers contenant les données des zones.
- **forwarders { adresse-ip; };** : si le serveur bind ne peut résoudre lui-même la requête, elle est renvoyée à un serveur DNS extérieur, par exemple celui du fournisseur d'accès.
- **listen-on-port 53 {127.0.0.1; adresse-ip; };** : port d'écoute du DNS suivi des adresses d'écoute. On indique ici les adresses IP des interfaces réseau de la machine. Il ne faut pas oublier 127.0.0.1.
- **allow-query { 127.0.0.1; réseau; };** : machine(s) ou réseau(x) autorisés à utiliser le service DNS. Par exemple 192.168.1/24. Si la directive est absente, tout est autorisé.
- **allow-transfer { 192.168.1.2; };** : machine(s) ou réseau(x) autorisés à copier la base de données dans le cas d'une relation maître et esclave. Par défaut aucune copie n'est autorisée.
- **notify no;** : on notifie ou non les autres serveurs DNS d'un changement dans les zones ou d'un redémarrage du serveur.

### c. Section de zones

Pour chaque domaine ou sous-domaine, on définit deux sections **zone**. La première contient les informations de résolution de nom (nom vers IP) et la seconde les informations de résolution inverse (IP vers Nom). Dans chacun des cas, la zone peut être maître **Master** ou esclave **Slave** :

- **Master** : le serveur contient la totalité des enregistrements de la zone dans ses fichiers de zone. Lorsqu'il reçoit une requête, il cherche dans ses fichiers (ou dans son cache) la résolution de celle-ci.
- **Slave** : le serveur ne contient par défaut aucun enregistrement. Il se synchronise avec un serveur maître duquel il récupère toutes les informations de zone. Ces informations peuvent être placées dans un fichier. Dans ce cas l'esclave stocke une copie locale de la base. Lors de la synchronisation, le numéro de série de cette copie est comparé à celui du maître. Si les numéros sont différents, une nouvelle copie a lieu, sinon la précédente continue à être utilisée.

### d. Zone de résolution

Elle est généralement appelée **zone**. Pour chaque domaine ou sous-domaine, elle indique dans quel fichier sont placées les informations de la zone (c'est-à-dire et entre autres les adresses IP associées à chaque hôte), son type (maître ou esclave), si on autorise ou non la notification, l'adresse IP du serveur DNS maître dans le cas d'un esclave, etc.

Le nom de la zone est très important puisque c'est lui qui détermine le domaine de recherche. Quand le DNS reçoit une requête, il recherche dans toutes les zones une correspondance.

```
zone "domaine.org" {
 type "master";
 file "domaine.org.zone";
};
```

- **type** : master ou slave ;
- **file** : nom du fichier qui contient les informations de la zone. Il n'y a pas de règles précises de nommage mais pour des raisons de lisibilité il est conseillé de lui donner le même nom que la zone tant pour une zone master que pour une slave. Pour un master, c'est l'original éventuellement rempli par vos soins. Pour un slave, ce n'est pas obligatoire. S'il est présent, ce sera une copie du master, synchronisée.
- Dans le cas d'un Master, on peut rajouter **allow-transfer** (serveurs autorisés à dupliquer la zone) et **notify yes** (indique une mise à jour ou une relance pour les slaves).

En cas de Slave : on rajoute la directive **masters** pour indiquer à partir de quel serveur Master dupliquer.

### e. Zone de résolution inverse

Pour chaque réseau ou sous-réseau IP (ou plage d'adresses) on définit une zone de résolution inverse dont le fichier contient une association IP vers nom de machine. C'est en fait presque la même chose que la zone de résolution sauf que l'on doit respecter une convention de nommage :

- Le nom de la zone se termine toujours par une domaine spécial **.in-addr.arpa**.
- On doit tout d'abord déterminer quel réseau la zone doit couvrir (cas des sous-réseaux). Pour nous : un réseau de classe C 192.168.1.0 soit **192.168.1/24**.
- On inverse l'ordre des octets dans l'adresse : **1.168.192**.
- On ajoute **in-addr.arpa**. Notre nom de zone sera donc **1.168.192.in-addr.arpa**.
- Pour le reste, les mêmes remarques que pour la zone de résolution s'appliquent.

```
Zone "1.168.192.in-addr.arpa" {
 type master;
 file "192.168.1.zone";
};
```

### f. Exemple

Soit un domaine `domaine.org` sur un réseau de classe C 192.168.1.0. Soit deux serveurs DNS 192.168.1.1 Master et 192.168.1.2 Slave.

#### Sur le Master

```
zone "domaine.org" {
 type master;
 file "domaine.org.zone";
 allow-transfer { 192.168.1.2; } ;
 notify yes;
};

zone "1.168.192.in-addr.arpa" {
 type master;
 file "192.168.1.zone";
 allow-transfer { 192.168.1.2; } ;
 notify yes;
};
```

#### Sur le Slave

```
zone "domaine.org" {
 type slave;
 file "domaine.org.zone";
 masters { 192.168.1.1; };
};

zone "1.168.192.in-addr.arpa" {
 type slave;
 file "192.168.1.zone";
 masters { 192.168.1.1; } ;
};
```

### g. Zones spéciales

La zone racine « . » permet de spécifier les serveurs racines. Quand aucune des zones n'arrive à résoudre une requête, c'est la zone racine qui est utilisée par défaut et qui renvoie sur les serveurs racines.

La zone de loopback n'est pas nécessaire bien que utile. Elle fait office de **cache DNS**. Quand une requête arrive sur le serveur et qu'il ne possède pas l'information de résolution, il va la demander aux serveurs DNS racines qui redescendront l'information. Celle-ci est alors placée en cache. Du coup les accès suivants seront bien plus rapides !

## 4. Fichiers de zones

### a. Définitions

Les fichiers de zones utilisent plusieurs termes, caractères et abréviations spécifiques.

- **RR** : *Resource Record*. Nom d'un enregistrement DNS (les données du DNS).
- **SOA** : *Star Of Authority*. Permet de décrire la zone.
- **IN** : *the Internet*. Définit une classe d'enregistrement qui correspond aux données Internet (IP). C'est celle par défaut si elle n'est pas précisée pour les enregistrements.
- **A** : *Address*. Permet d'associer une adresse IP à un nom d'hôte. Pour Ipv6 c'est AAAA.
- **NS** : *Name Server*. Désigne un serveur DNS de la zone.
- **MX** : *Mail eXchanger*. Désigne un serveur de courrier électronique, avec un indicateur de priorité. Plus la valeur est faible, plus la priorité est élevée.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



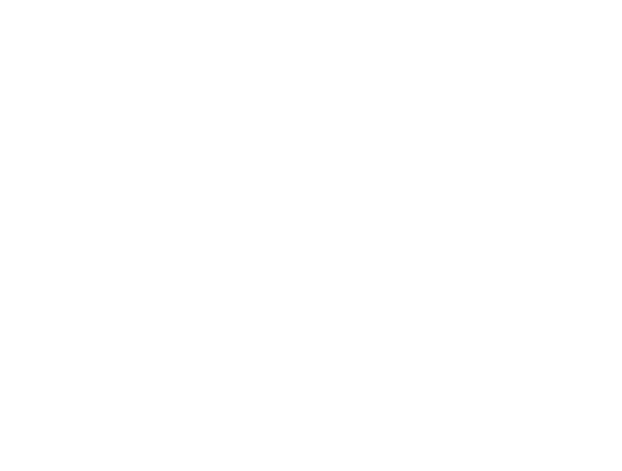
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



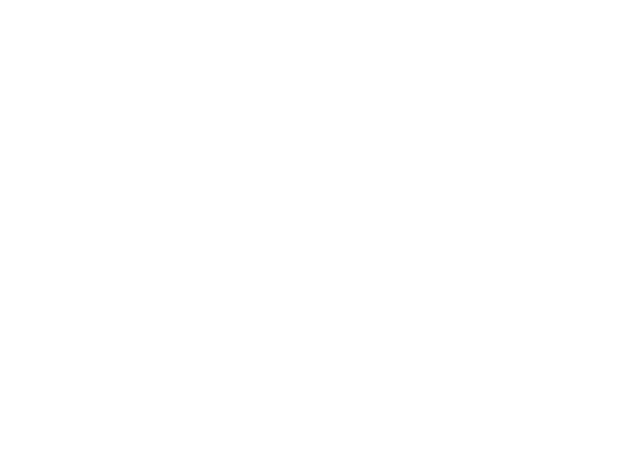
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



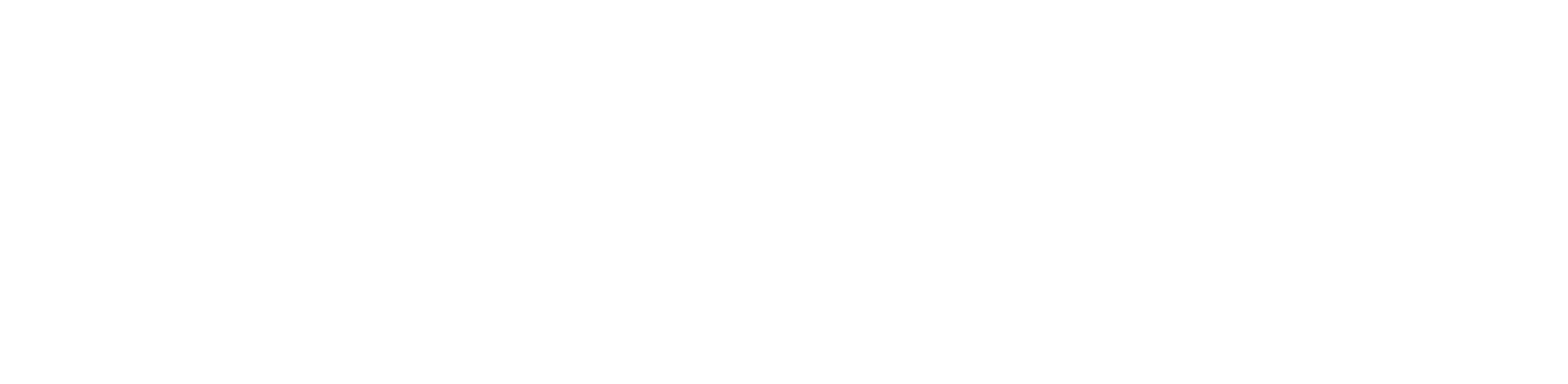
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



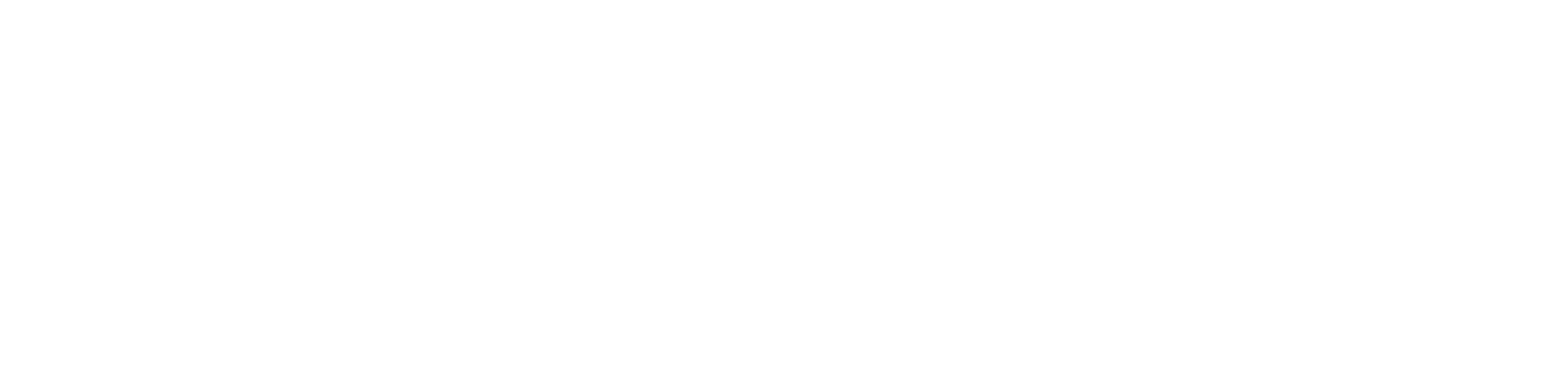
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



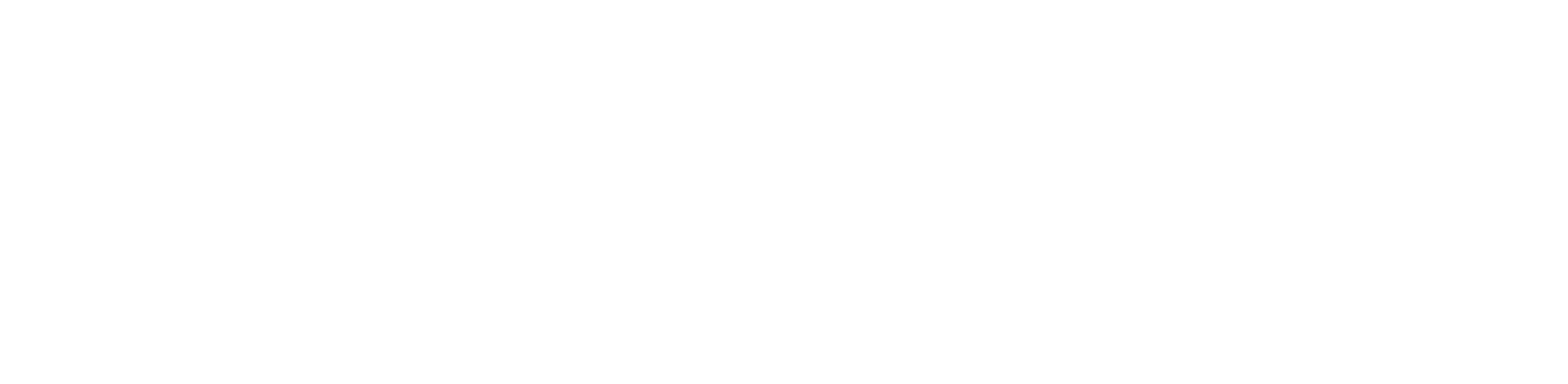
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



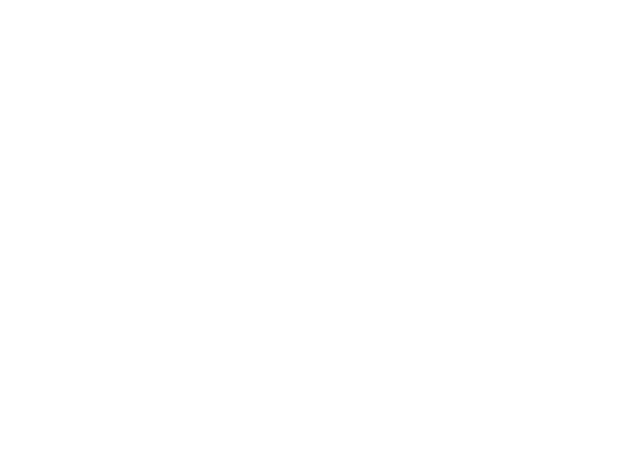
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



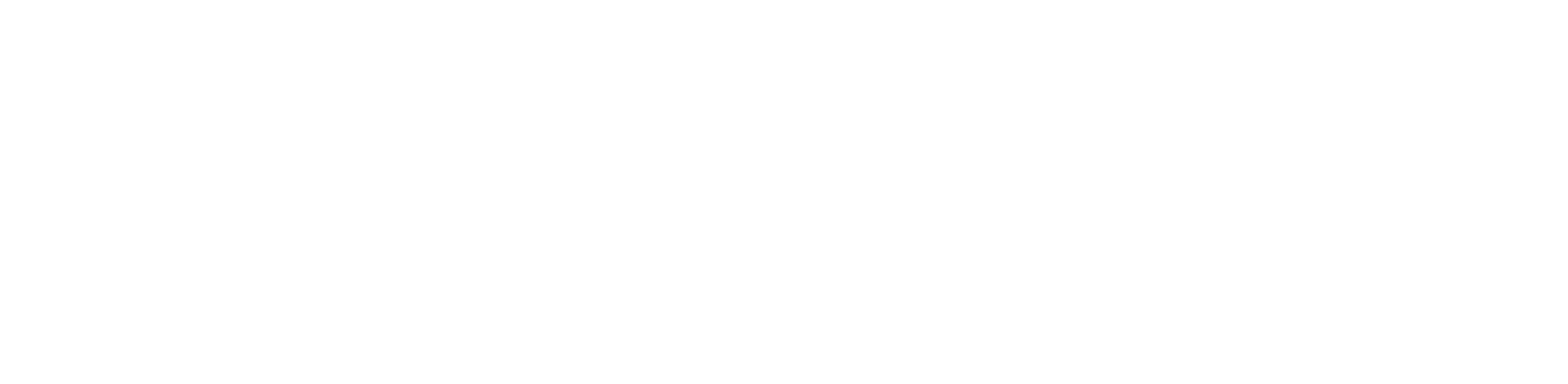
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



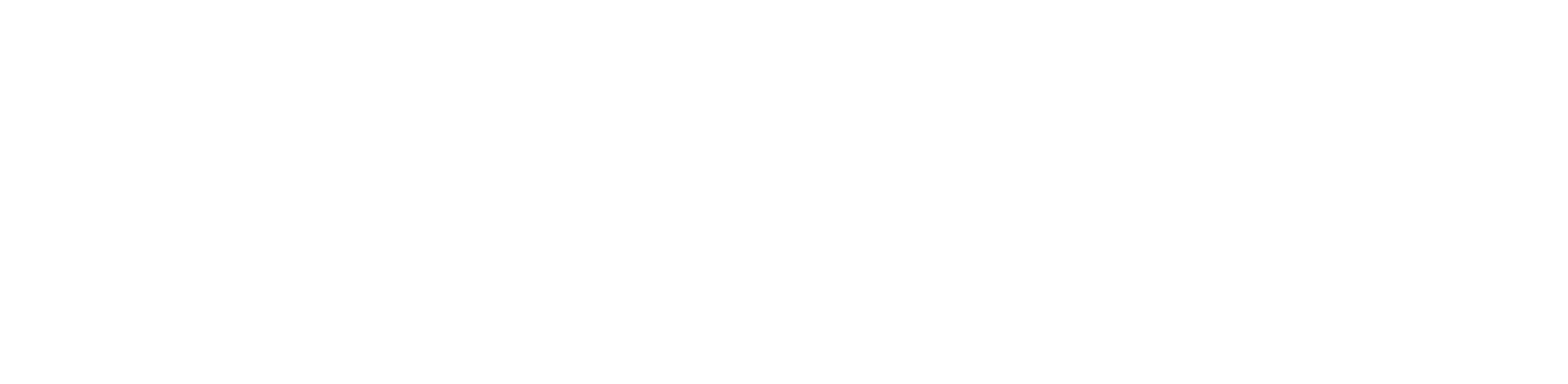
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



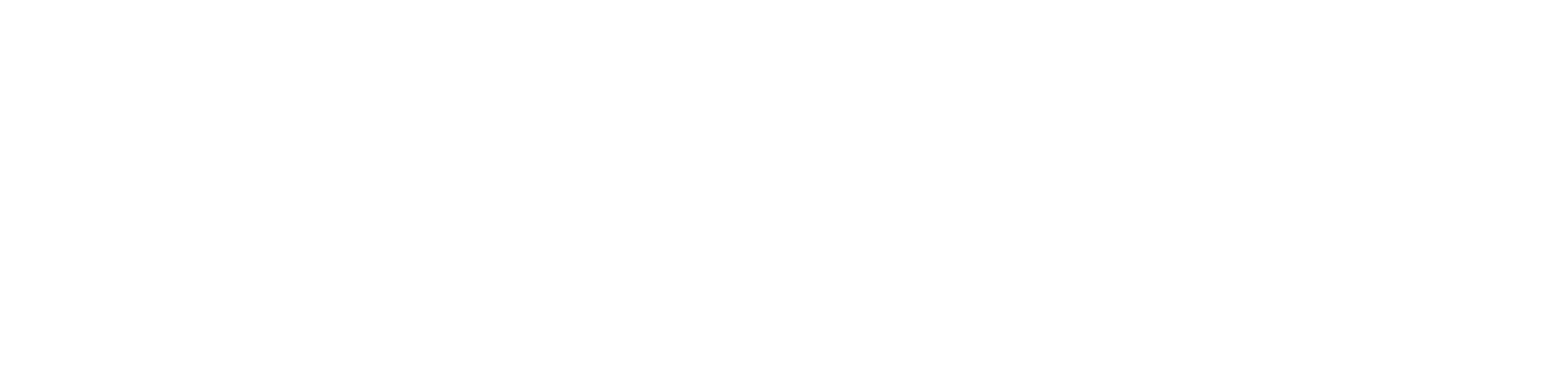
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



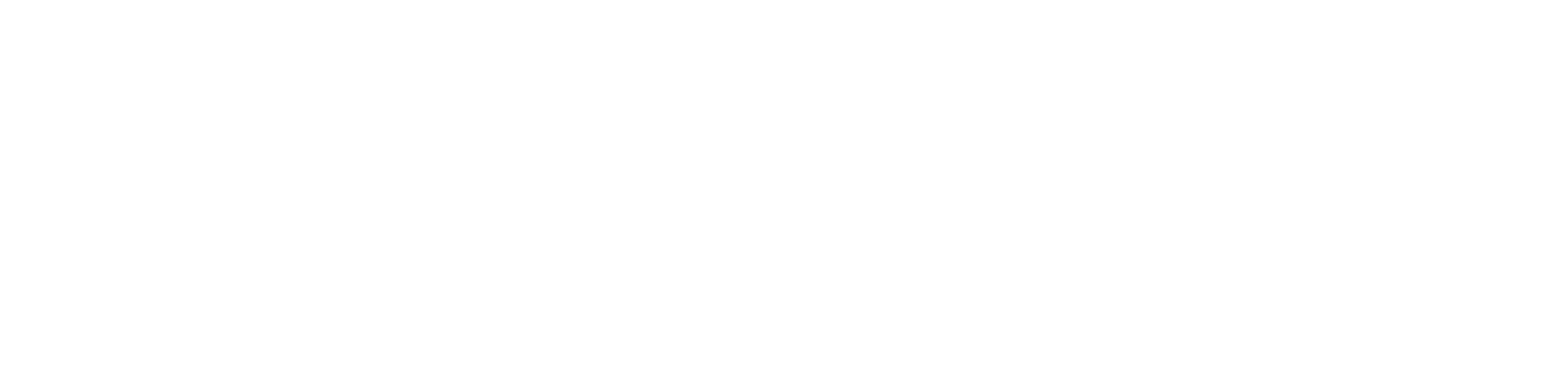
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



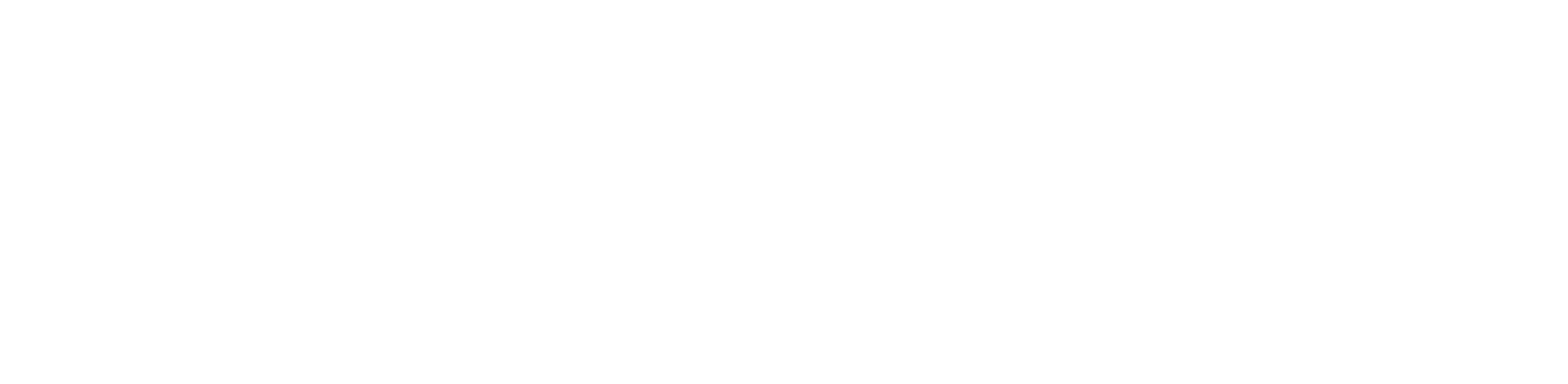
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



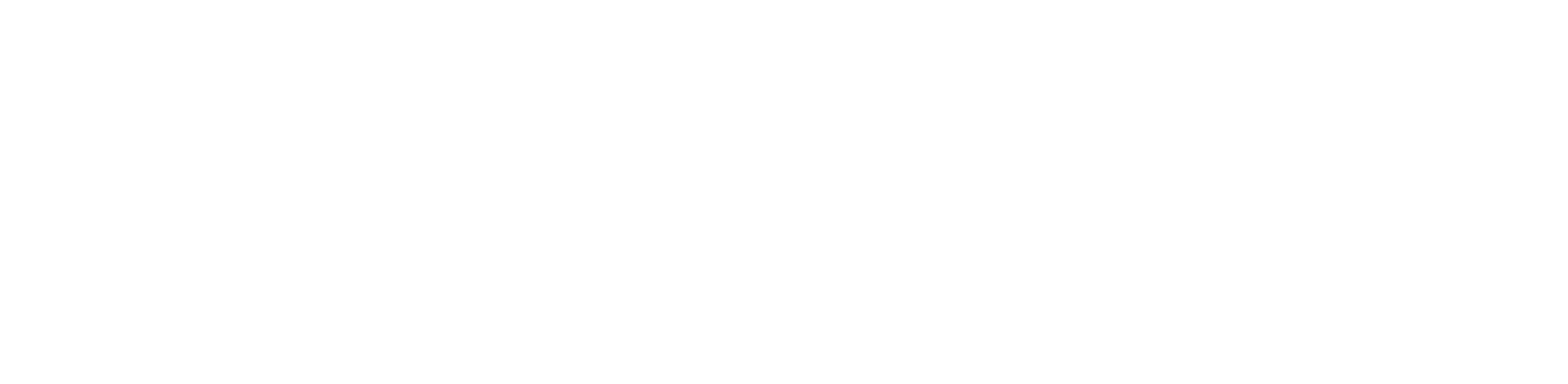
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Chaque objectif est pondéré par un poids (valeur indiquée entre parenthèses). Les poids s'étendent de **1** à **10** et indiquent l'importance relative de chaque objectif. Les objectifs avec les poids les plus élevés seront représentés dans l'examen par plus de questions.

| Objectifs de l'examen LPI 101                                           | Chapitre                   | Travaux pratiques |
|-------------------------------------------------------------------------|----------------------------|-------------------|
| <b>101 Matériel et architecture</b>                                     |                            |                   |
| 101.1 Réglages fondamentaux du BIOS (2)                                 | 5.A.1                      |                   |
| 101.2 Démarrer le système (3)                                           | 5.A                        | 5.1               |
| 101.3 Changer les niveaux, arrêter ou rebooter (3)                      | 5.B                        | 5.2               |
| <b>102 Installation de Linux et gestion des packages</b>                |                            |                   |
| 102.1 Préparer le disque dur (2)                                        | 2.A.5, 2.B.6, 4.A          | 2.1, 4.1          |
| 102.2 Installer un gestionnaire de boot (2)                             | 2.A.9, 5.A.3               | 5.1               |
| 102.3 Gérer les bibliothèques partagées (1)                             | 2.H                        | 2.5               |
| 102.4 Le gestionnaire de packages Debian et APT (3)                     | 2.E, 2.F                   | 2.3               |
| 102.5 Le gestionnaire de packages RPM et YUM (3)                        | 2.C, 2.D                   | 2.2               |
| <b>103 Les commandes GNU et Unix</b>                                    |                            |                   |
| 103.1 Utiliser la ligne de commande (4)                                 | 3.A                        |                   |
| 103.2 Traiter les flux texte avec les filtres (3)                       | 3.F                        | 3.4               |
| 103.3 Gestion de base des fichiers (3)                                  | 3.B                        | 3.1               |
| 103.4 Canaux, tubes et redirections (4)                                 | 3.E                        | 3.3               |
| 103.5 Créer, visualiser et tuer les processus (4)                       | 3.G                        | 3.5               |
| 103.6 Modifier la priorité des processus (2)                            | 3.G.8                      | 3.5               |
| 103.7 Rechercher du texte, expressions régulières (2)                   | 3.C                        | 3.4               |
| 103.8 Les bases de l'éditeur vi (3)                                     | 3.D                        |                   |
| <b>104 Périphériques de stockage, Système de fichiers, arborescence</b> |                            |                   |
| 104.1 Créer des partitions et des systèmes de fichiers (2)              | 4.B, 4.C, 4.D,<br>4.E, 4.H | 4.1, 4.2          |
| 104.2 Maintenir l'intégrité des systèmes de fichiers (2)                | 4.G                        | 4.4               |
| 104.3 Monter et démonter des systèmes de fichiers (3)                   | 4.F                        | 4.3               |
| 104.4 Appliquer des quotas (1)                                          | 4.I                        | 4.6               |
| 104.5Modifier les droits des fichiers (3)                               | 4.J.1 à 4.J.5              | 4.7               |
| 104.6Créer et modifier les liens (2)                                    | 4.E.1.e, 3.B.5.b           |                   |
| 104.7Rechercher des fichiers (2)                                        | 3.C                        | 3.2               |

| Objectifs de l'examen LPI 102                                           | Chapitre                      | Travaux pratiques             |
|-------------------------------------------------------------------------|-------------------------------|-------------------------------|
| <b>105 Shell, scripts, programmation et compilation</b>                 |                               |                               |
| 105.1 Modifier l'environnement du shell (4)                             | <a href="#">6.A.8, 3.J</a>    |                               |
| 105.2 Écrire et modifier des scripts (4)                                | <a href="#">3.K</a>           | <a href="#">3.6, 3.7</a>      |
| 105.3 Gestion des données avec SQL                                      | <a href="#">3.L</a>           |                               |
| <b>106 Environnements graphiques et de bureau</b>                       |                               |                               |
| 106.1 Installer et configurer X11 (2)                                   | <a href="#">9.B</a>           | <a href="#">9.2</a>           |
| 106.2 Configurer un gestionnaire d'affichage (2)                        | <a href="#">9.C</a>           | <a href="#">9.3</a>           |
| 106.3 Accessibilité (1)                                                 | <a href="#">9.E</a>           |                               |
| <b>107 Tâches administratives</b>                                       |                               |                               |
| 107.1 Gérer les groupes, les utilisateurs et répertoires personnels (5) | <a href="#">6.A</a>           | <a href="#">6.1</a>           |
| 107.2 Automatiser les tâches d'administration (4)                       | <a href="#">6.C</a>           | <a href="#">6.3</a>           |
| 107.3 Régionalisation et internationalisation (3)                       | <a href="#">6.G</a>           |                               |
| <b>108 Services essentiels du système</b>                               |                               |                               |
| 108.1 Maintenir le système à l'heure (3)                                | <a href="#">6.F</a>           |                               |
| 108.2 Configurer syslog (2)                                             | <a href="#">6.D</a>           | <a href="#">6.4</a>           |
| 108.3 Configuration basique du MTA (3)                                  | <a href="#">7.G</a>           | <a href="#">7.4</a>           |
| 108.4 Gérer les imprimantes et files d'impression (2)                   | <a href="#">6.B</a>           | <a href="#">6.2</a>           |
| <b>109 Les bases du réseau</b>                                          |                               |                               |
| 109.1 Les bases de TCP/IP (4)                                           | <a href="#">7.A.1, 7.A.2</a>  | <a href="#">7.1</a>           |
| 109.2 Configuration TCP/IP de Linux (4)                                 | <a href="#">7.A.3 à 7.A.5</a> | <a href="#">7.1, 7.2</a>      |
| 109.3 Résoudre les problèmes réseaux courants                           | <a href="#">7.A</a>           |                               |
| 109.4 Gestion des noms de domaines DNS de base (4)                      | <a href="#">7.F</a>           | <a href="#">7.3</a>           |
| <b>110 Sécurité</b>                                                     |                               |                               |
| 110.1 Tâches d'administration de sécurité (3)                           | <a href="#">8.A</a>           | <a href="#">8.1</a>           |
| 110.2 Sécurité de la machine locale (3)                                 | <a href="#">8.A, 8.B</a>      | <a href="#">8.1, 8.3, 8.4</a> |
| 110.3 Sécuriser les données en les chiffrant                            | <a href="#">7.D, 8.5</a>      | <a href="#">8.2</a>           |



./configure, 107  
 /etc/bashrc, 374  
 /etc/exports, 458  
 /etc/fstab, 459  
 /etc/group, 360  
 /etc/hosts, 430  
 /etc/localtime, 404  
 /etc/network/interfaces, 420  
 /etc/networks, 430  
 /etc/nologin, 486  
 /etc/nsswitch.conf, 430  
 /etc/passwd, 360  
 /etc/profile, 373  
 /etc/resolv.conf, 429  
 /etc/samba/smb.conf, 461  
 /etc/security, 487  
 /etc/security/limits.conf, 493  
 /etc/services, 430  
 /etc/shadow, 360  
 /etc/sudoers, 493  
 /etc/sysconfig/network-scripts/, 420  
 /etc/syslog.conf, 390  
 /etc/timezone, 404  
 /etc/xinetd.conf, 432  
 /etc/yum.conf, 91  
 /etc/yum.repos.d, 91  
 /proc, 316  
 /proc/mdstat, 596  
 /sys, 316



Accessibilité, 576  
 AccessX, 577  
 Accolades, 187  
 Aide, 51  
 Alias, 183  
 Apache, 454  
     hôtes virtuels, 457  
 API, 33  
 appres, 572  
 APT, 100  
     configuration, 101  
     dépôts, 101  
     mise à jour, 102  
 apt-cache, 105  
 apt-get, 102 - 103  
 Archivage, 392

Arrêt, 303  
 Assistance, 576  
 at, 387  
 AT (modems), 435  
 atq, 388  
 atm, 389  
 Authentification, 358  
 Automatisation, 385



badblocks, 261  
 banner, 175  
 basename, 165  
 bash, 136  
 Bash  
     configuration, 190  
     groupement, 184  
     liaison, 184  
 bg, 180  
 Bibliothèques partagées, 117  
     cache, 119  
     stockage, 117  
 Bind, 443  
 BIOS, 290  
 blockdev, 245  
 BOOTP, 441  
 Bourne Shell, 136  
 break, 204  
 britty, 579  
 Bulletins de sécurité, 496  
 Bureaux virtuels, 533



cal, 138  
 Calcul, 206  
 Canaux, 163  
     ouverture, 164  
 case, 199  
 cat, 174  
 cd, 145  
 CDE, 567  
 CERT, 496  
 chage, 364, 485  
 Chargeur de démarrage, 292  
 Chemin  
     absolu, 144  
     personnel, 144  
     relatif, 145

# C Collection *Certifications*

Les examens LPI 101 et LPI 102 sont les deux examens qui permettent d'obtenir la **certification LPIC-1 "Junior Level Linux Professionnal"**. Ce programme de certification du Linux Professional Institute est de plus en plus **reconnu par les recruteurs** qui voient dans cette certification un pré-requis à l'embauche ou à l'accession à un poste d'administrateur.

Les examens LPI 101 et 102 prouvent aux professionnels que vous maîtrisez les bases de l'administration système **Linux quelle que soit la distribution** : l'installation et la configuration complète d'un poste de travail et de tous les services associés, tant systèmes que réseaux.

Pour vous aider à préparer efficacement cette certification, ce livre couvre **tous les objectifs officiels de la dernière version de l'examen**, tant d'un point de vue théorique que d'un point de vue pratique. Il a été rédigé en français (il ne s'agit pas d'une traduction) par un formateur professionnel reconnu, également consultant, certifié Linux. Ainsi, les savoir-faire pédagogique et technique de l'auteur conduisent à une approche claire et visuelle, d'un très haut niveau technique.

Chapitre par chapitre, vous pourrez **valider vos acquis théoriques**, à l'aide d'un grand nombre de **questions-réponses (536 au total)** mettant en exergue aussi bien les éléments fondamentaux que les caractéristiques spécifiques aux concepts abordés.

Chaque chapitre s'achevant par des **travaux pratiques (47 au total)** vous aurez les moyens de mesurer votre autonomie. Ces manipulations concrètes, au-delà même des objectifs fixés par l'examen, vous permettront de vous forger une première expérience significative et d'acquérir de véritables compétences techniques sur des mises en situations réelles.

A cette maîtrise du produit et des concepts, s'ajoute la préparation spécifique à la certification : vous pourrez accéder **gratuitement à 1 examen blanc en ligne**, destiné à vous entraîner dans des conditions proches de celles de l'épreuve. Sur ce site, chaque question posée s'inscrit dans l'esprit de la certification et, pour chacune, les réponses sont suffisamment commentées pour contrôler et identifier vos ultimes lacunes. A vous de juger quand vous serez prêt pour l'examen final !

## Auteur : Sébastien ROHAUT

Sébastien Rohaut est Ingénieur Système en missions régulières pour de grands comptes. Il enseigne également Unix et PHP à des classes préparatoires et d'ingénieurs et préparent les étudiants aux certifications LPIC. Lui-même certifié Linux, il est fortement investi dans le monde des logiciels libres (fondateur et ancien président de Slyunix, association de promotion de Linux). Il écrit fréquemment dans la presse spécialisée (Planète Linux...) des articles destinés aux amateurs de Linux et des logiciels libres.

ISSN 1633-8693

ISBN : 978-2-7460-5009-9



9 782746 050099

Prix : 59 €

## EXAMENS LPI 101 et LPI 102

