Guide d'introduction à Debian GNU–Linux

John Goerzen et Ossama Othman
Traduction française de Patrice Karatchentzeff <p.karatchentzeff@free.fr>

Traduction du document du 2 octobre 2000

Résumé

Ceci est la version française du guide d'introduction à la distribution Debian GNU–Linux.

Copyright

Copyright © 1998, 1999, 2000 Software in the Public Interest, Inc

Il est autorisé de faire et de distribuer des copies modifiées du présent document sous les conditions de reproduction déclarées dans la « GNU General Public License » ainsi que dans tous les autres paragraphes où les conditions particulières de *copyright* s'appliquent, et que le résultat dérivé du précédent manuel dans son intégralité soit redistribué dans les mêmes conditions.

Il est autorisé de faire et de distribuer des copies traduites de ce manuel dans d'autres langues dans les conditions des versions modifiées. La « GNU General Public License » peut être incluse en version traduite approuvée par la *Free Software foundation* à la place de la version anglaise du document.

Vous pouvez comme vous le souhaîter distribuer des versions exactes ou modifiées du présent document en respectant les termes de la « GNU General Public License », à l'exception des sections clairement marquées avec un *copyright* séparé.

Table des matières

1 Remerciements **1**

2 Préface **3**

3 Introduction **5**

3.1 Qu'est–ce que Debian ? 5

3.1.1 Qui est l'auteur de Debian ? 6

3.2 Un système d'exploitation multi–tâches et multi–utilisateurs. 6

3.3 Qu'est–ce qu'un logiciel libre ? 7

3.4 Au sujet de ce livre 8

3.4.1 Comment lire ce livre 8

3.4.2 Conventions typographiques 9

4 Bien débuter **11**

4.1 Matériel supporté 11

4.1.1 Espace disque et de mémoire 11

4.2 Avant de commencer 12

4.2.1 Les informations nécessaires 12

4.3 Partitionner son disque dur 12

4.3.1 Travaux préliminaires 13

4.3.2 Planifier l'usage du système 14

4.3.3 Les limitations des disques PC 15

4.3.4 Nom de périphériques sous Linux 15

4.3.5 Schéma de partitionnement recommandé 16

4.3.6 Partitionnement avant installation 17

4.3.7 Les étapes d'installation de Debian 18
4.4 Choisir son support d'installation 19
4.4.1 Installation à partir du cédérom 19
4.4.2 Installation à partir de disquettes 19
4.5 Booter sur la procédure d'installation 21

5 Installation pas à pas **23**
5.1 Sélection d'un écran monochrome ou couleur 23
5.2 Menu principal d'installation de Debian GNU/Linux 23
5.3 Configurer le clavier 24
5.4 Dernière chance de sauvegarder vos données ! 24
5.5 Partitionner un disque dur 24
5.6 Initialiser et activer une partition de *swap* 25
5.7 Initialiser une partition Linux 26
5.7.1 Monter une partition précédemment initialisée. 27
5.8 Installation du système d'exploitation et ses modules 27
5.9 Configuration du support PCMCIA 27
5.10 Configuration des modules de pilotes de périphériques 28
5.10.1 Configuration du réseau 28
5.11 Installation du système de base. 28
5.11.1 Configuration du système de base 29
5.11.2 Faire que Linux soit bootable directement à partir du disque dur 29
5.12 Fabriquer une disquette de boot 30
5.13 Le moment de vérité 30
5.14 Ajouter un mot de passe root 30
5.15 Créer un compte utilisateur ordinaire 31
5.16 Support des mots de passe cachés. 31
5.17 Retirer le support PCMCIA 31
5.18 Sélection et installation de *profil* 31
5.19 Installation de paquets avec `dselect` 32
5.19.1 Introduction 32

5.19.2 Une fois que `dselect` est lancé . . . 33
5.19.3 Quelques remarques pour conclure . . . 39
5.20 Glossaire . . . 39

6 Se loguer **41**
6.1 Les premiers pas . . . 41
6.2 Historique des commandes et édition de la ligne de commande . . . 42
6.3 Travailler en tant que Root . . . 43
6.4 Consoles virtuelles . . . 44
6.5 Arrêter le système . . . 45

7 Les bases **47**
7.1 La ligne de commandes et les pages de `man` . . . 47
7.2 Description de la ligne de commande . . . 48
7.3 Fichiers et répertoires . . . 49
7.3.1 Utilisation des fichiers : un tutoriel . . . 50
7.3.2 Fichiers cachés et `ls -a` . . . 54
7.4 Processus . . . 54
7.5 Le shell . . . 55
7.6 Gérer les processus avec `bash` . . . 56
7.7 Quelques fonctionnalités de `bash` . . . 59
7.7.1 Le complètement par tabulation . . . 59
7.8 Protéger votre identité . . . 59

8 Utiliser le shell **61**
8.1 Les variables d'environnement . . . 61
8.2 Où les commandes résident : la variable *PATH* . . . 63
8.3 Fichiers de configuration . . . 64
8.3.1 Configuration système contre configuration personnelle . . . 64
8.4 Aliases . . . 65
8.5 Contrôler les entrées et les sorties . . . 65
8.5.1 `stdin`, `stdout`, tubes et redirection . . . 65
8.6 Expansion de nom de fichier . . . 67

9 Un peu plus sur les fichiers **69**

9.1 Permissions 69

9.1.1 Les droits de propriété d'un fichier 70

9.1.2 Mode 71

9.1.3 Les permissions en pratique 73

9.2 Présentation des fichiers et de leurs emplacements 74

9.3 La compression de fichier avec `gzip` 76

9.4 Rechercher des fichiers 77

9.5 Déterminer le contenu d'un fichier 78

9.6 Utiliser un gestionnaire de fichier 78

10 Travailler avec des fichiers textes **79**

10.1 Lire des fichiers textes 79

10.2 Les éditeurs de textes 80

10.3 Utiliser `ae` 80

11 Le système X Window **81**

11.1 Introduction à X 81

11.2 Lancer l'environnement X 82

11.3 Les opérations élémentaires sous X 83

11.3.1 La souris 83

11.3.2 Les clients X 83

11.3.3 Problèmes 83

11.3.4 Quitter l'environnement X 84

11.3.5 Personnaliser le lancement de X 84

12 Les systèmes de fichiers **87**

12.1 Concepts 87

12.2 `mount` et `/etc/fstab` 88

12.2.1 Monter un système de fichiers 88

12.2.2 Exemple : monter un cédérom 88

12.2.3 `/etc/fstab` : automatiser la procédure de montage. 90

12.2.4 Supports extractibles (disquettes, lecteur Zip, etc.) . . . 90
12.3 Les outils de sauvegarde . . . 91
12.3.1 `tar` . . . 91

13 Réseau **93**
13.1 `PPP` . . . 93
13.1.1 Introduction . . . 93
13.1.2 Préparation . . . 93
13.1.3 La voie facile : `wvdial` . . . 94
13.2 Ethernet . . . 95

14 Installer et désinstaller des logiciels **97**
14.1 À quoi sert un utilitaire de maintenance de paquet . . . 97
14.2 `dpkg` . . . 97
14.3 `dselect` . . . 98
14.4 Compiler un logiciel . . . 98

15 Pour en savoir plus... **99**
15.1 Les expressions rationnelles . . . 99
15.2 Notions avancées sur les fichiers . . . 101
15.2.1 La véritable nature des fichiers : les liens physiques et les i–noeuds . . . 101
15.2.2 Types de fichiers . . . 103
15.2.3 Le système de fichiers `proc` . . . 106
15.2.4 Copie à grande échelle . . . 107
15.3 Sécurité . . . 107
15.4 Le développement de logiciels avec Debian . . . 108

16 Lire la documentation et obtenir de l'aide **109**
16.1 Les sources de documentation . . . 109
16.2 Utiliser `info` . . . 110
16.3 HOWTO . . . 110
16.4 Aide personnalisée . . . 111
16.5 Extraire des informations du système . . . 112

17 Problèmes **113**

17.1 Difficultés classiques 113

17.1.1 Travailler avec des noms de fichiers étranges 113

17.1.2 Imprimer 114

17.1.3 Les problème sous X 114

17.2 Problème lors de la procédure de boot 114

18 Booter le système **117**

19 La Licence Publique Générale de GNU **119**

19.1 GNU GENERAL PUBLIC LICENSE 119

19.2 Comment appliquer ces dispositions à vos nouveaux programmes ? 125

Chapitre 1

Remerciements

De nombreuses personnes ont contribué à ce manuel. Nous aimerions toutes les remercier et nous allons essayer ici.

En espérant n'oublier personne parmi les nombreux collaborateurs à ce manuel, nous remercions : Havoc Pennington, Ardo van Rangelooij, Larry Greenfield, Thalia Hooker, Day Irmiter, James Treacy, Graig Sawyer, Oliver Elphick, Ivan E. Moore, Eric Fischer, Mike Touloumtzis et le *Projet de documentation de Linux* pour leur travail sur ce qu'est devenu le *Guide d'introduction à Debian GNU–Linux*.

Nous remercions Richard Stallman de la *Free Software Foundation* pour ses conseils et pour l'édition.

Nous disons un gros merci à Bruce Perens, Sven Rudolph, Igor Grobman, James Treacy, Adam Di Carlo, Tapio Lehtonen et Stéphane Bortzmeyer pour leurs travaux sur ce qui allait devenir une collection de documents d'installation.

Enfin, nous souhaitons vivement remercier les centaines de développeur Debian et que les milliers d'auteurs de logiciels libres qui nous ont donnés matière à penser et à écrire.

Chapitre 2

Préface

« La liberté est la plus radicale de toutes les idées. »

Cette citation de Nathaniel Branden décrit bien, à nos yeux, le mouvement du logiciel libre. En l'espace de quelques décennies, à peine, l'informatique a transformé la vie de tout un pan de l'humanité. Par notre engagement dans le mouvement du logiciel libre, nous accentuons ce changement en offrant au plus grand nombre d'individus et de communautés la possibilité d'accéder pleinement aux ressources logicielles qui, dès lors, ne sont plus la chasse gardée de certains éditeurs et de leurs développeurs. Composants essentiels de cet objectif, ce livre et ce cédérom contiennent un précieux lot de logiciels libres, plus d'un millier de paquets, dont le plus populaire des serveurs web. Vous pouvez utiliser ces logiciels pour n'importe quoi, du dessin à la gestion de bases de données SQL.

La révolution de Logiciel Libre a pris l'industrie par surprise. Linux, conçu à partir de zéro il y a moins de dix ans, est devenu le noyau favori des auteurs de logiciels libres. Les idées et les expériences accumulées depuis ont propulsé Linux et les outils GNU de la *Free Sofware Foundation* aux avant–postes partout dans le monde. Des systèmes libres, comme celui de Debian GNU–Linux, rassemblent des milliers d'applications. Et qu'on ne s'y trompe pas, leur stabilité et leur puissance dépassent celles de bien des systèmes vedettes de l'industrie traditionnelle.

De nos jours, le système GNU–Linux joue un rôle dominant dans les serveurs Internet, parmi les fournisseurs d'accès, dans les universités, parmi les passionnés et dans le monde de la recherche informatique. Le système Debian GNU–Linux a mis la puissance des logiciels libres au service de tous, des ordinateurs portables aux engins spatiaux.

De très nombreuses entreprises dans le monde ont connu les satisfactions et les avantages qui découlent de l'utilisation de logiciels libres. Cette puissance et cette stabilité impressionnantes de même que la possibilité de communiquer directement avec l'auteur des programmes que vous utilisez, la capacité de les modifier à volonté, si, bien sûr, vous avez appris à le faire, et le support technique convivial que vous pouvez obtenir en ligne, en rien comparable, se combinent pour transformer l'usage au quotidien d'un ordinateur en une expérience pleine de découvertes et, surprise ! d'émotions.

Entrer dans l'univers du logiciel libre, c'est sûrement la meilleure chose que vous ayez faite avec votre ordinateur ; le faire à partir d'une Debian GNU–Linux, c'est d'autant mieux : elle est puissante, robuste, très bien documentée et, en prime, ludique !

Bienvenue à la révolution !

– John Goerzen.

Chapitre 3

Introduction

Nous sommes heureux d'avoir la chance de vous présenter Debian. Avant d'entamer la discussion sur GNU–Linux, nous aimerions d'abord parler un peu de ce qu'est exactement Debian – ce qu'elle fait et qu'elle sont ses buts dans le monde vaste du Logiciel Libre.

3.1 Qu'est–ce que Debian ?

Debian est un système d'exploitation (OS pour *Operating System*) libre pour votre ordinateur. Un système d'exploitation est un rassemblement de programmes élémentaires et d'utilitaires qui font fonctionner votre ordinateur. Au coeur du système se trouve le noyau. Le noyau est le programme le plus fondamental de votre ordinateur. Il s'occupe de toutes les tâches élémentaires et permet de lancer les autres programmes. Debian utilise le noyau Linux, un ensemble de logiciels entièrement libres initié par Linus Torvalds et supportant des milliers de logiciels dans le monde. Une part importante des outils de base qui forment le système d'exploitation vient du projet GNU (*http ://www.gnu.org/*) et ces outils sont aussi libres.

Une autre facette d'un système d'exploitation est la partie logiciels applicatifs : ce sont les programmes qui vous aident dans votre travail, de l'édition de documents en passant par les jeux jusqu'à l'écriture d'autres programmes. Debian rassemble plus de 1500 paquets (ce sont des logiciels pré–empaquetés dans un format pratique pour l'installation sur votre machine) tous libres.

Le système Debian ressemble un peu à une pyramide. À la base se trouve le noyau Linux. Juste au–dessus se trouve les outils de base, la plupart en provenance du mouvement GNU. Ensuite, on trouve les logiciels applicatifs que vous utilisez sur votre ordinateur ; la plupart d'entre eux sont aussi d'origine GNU. Les développeurs Debian agissent comme des architectes et des coordinateurs : ils organisent avec soin le système en ajustant l'ensemble dans un système d'exploitation cohérent et stable : Debian GNU–Linux.

La philosophie d'un système GNU–Linux est de répartir ses fonctionnalités en un ensemble de petites tâches élémentaires. De cette façon, il est aisé d'ajouter une nouvelle fonctionnalité en combinant les petits bouts de programmes de nouvelle façon. Debian est un jeu de construction : vous pouvez construire toutes sortes de choses avec.

Lorsque vous utilisez votre système d'exploitation, vous désirez réduire la quantité de travail pour effectuer une tâche donnée. Debian vous fournit quantité d'outils pour vous aider mais seulement si vous connaissez ce que font ces outils. Perdre une heure à faire fonctionner quelque chose pour l'abandonner ensuite n'est pas très productif. Ce guide a pour but de vous apprendre ces outils de base qui sont le coeur de Debian : quels outils utilisés dans certains cas et comment les combiner.

3.1.1 Qui est l'auteur de Debian ?

Debian est un projet de développement à travers Internet basé sur le volontariat. Il y a plusieurs centaines de volontaires travaillant dessus. La plupart sont responsables d'un petit nombre de paquets de logiciel et connaissent très bien les logiciels qu'ils empaquètent.

Ces volontaires travaillent de concert en respectant un ensemble de directives dans la façon d'assembler un paquet. Ces directives sont développées en coopération sur Internet lors de discussions dans des listes de diffusion.

3.2 Un système d'exploitation multi–tâches et multi–utilisateurs.

Comme nous l'avons précédemment mentionné dans la section 3.1, l'architecture de Debian GNU–Linux vient directement de celle d'Unix. À la différence des systèmes d'exploitation comme le DOS, Windows et MacOS, on trouve habituellement le système GNU–Linux sur de grands serveurs et des systèmes multi–utilisateurs.

Ceci veut donc dire que Debian possède des avantages que les précédents systèmes n'ont pas. Il autorise un nombre important de personnes à utiliser le système en même temps, tant que chaque utilisateur possède son propre terminal. [1] Pour autoriser de nombreux utilisateurs à travailler simultanément, Debian doit gérer de nombreux programmes et applications en même temps. Cette faculté s'appelle le *multi–tâches*.

Une grande partie de la puissance (et de la complexité) des systèmes GNU–Linux proviennent de ces deux fonctionnalités. Par exemple, le système doit avoir une solution pour empêcher un utilisateur de détruire accidentellement les fichiers d'un autre utilisateur. Le système d'exploitation doit aussi coordonner les nombreux programmes tournant en même temps et s'assurer qu'ils n'utilisent pas la même ressource, comme le disque dur, ensemble.

Si vous gardez à l'esprit la raison pour laquelle Debian a été créée, de nombreux de ses aspects vont alors avoir plus de sens. Vous apprendrez à tirer parti de la puissance de ces fonctionnalités.

[1] Un terminal est un simple écran et clavier qui sont connectés à l'ordinateur à travers le réseau, par l'intermédiaire d'un modem, ou directement. Votre clavier et l'écran forment alors un terminal qui est directement rattaché à l'ordinateur. Ce terminal spécial est souvent appelé *console*

3.3 Qu'est–ce qu'un logiciel libre ?

Lorsque les développeurs et les utilisateurs de Debian parlent de logiciels libres, ils pensent à liberté plutôt qu'à gratuité. [2] Debian est libre dans le sens où vous êtes libre de la modifier et de la redistribuer, et vous aurez toujours accès au code source pour le faire. Les « Directives du Logiciel Libre de Debian » (*http ://www.debian.org/social_contract#guidelines*) décrivent avec plus de détails ce que Debian appelle exactement libre. La *Free Software Foundation*, (*http ://www.fsf.org*), à l'origine du projet GNU, est aussi une excellente source d'information. Vous pouvez trouver plus de détails sur le libre sur le site web de Debian (*http ://www.debian.org*). Un des travaux les plus connus en la matière est un essai de Richard Stallman « Why Software should be free[3] ? » (*http ://www.fsf.org/philosophy/shoudbefree.html*). Jetez–y donc un coup d'oeil pour connaître les raisons pour lesquelles nous supportons les logiciels libres comme nous le faisons. Récemment, des gens ont commencé à appeler les logiciels libres des logiciels *Open Source* (à code ouvert) ; ces deux termes sont interchangeables.

Il se peut que vous vous demandiez pourquoi certaines personnes prennent sur leur temps libre et passent des heures à écrire des logiciels et les empaqueter soigneusement juste pour les donner. Les réponses à cette question sont aussi variées que les contributeurs.

Beaucoup croient au partage de l'information et à la liberté de coopérer avec quelqu'un d'autre et ils pensent que le logiciel libre encourage cette pratique. Une longue tradition de mise en avant de ces valeurs, parfois appelée l'Éthique du Hacker[4], a commencé dans les années cinquante. Le Projet Debian GNU–Linux a été fondé sur les bases de l'éthique de liberté des logiciels libres, sur le partage et la coopération.

D'autres désirent apprendre davantage sur les ordinateurs. De plus en plus de gens recherchent des solutions pour éviter les prix sans cesse croissants des logiciels propriétaires. C'est une communauté grandissante d'utilisateurs qui ne cesse d'apprécier tous les logiciels libres qu'elle a reçu d'autres personnes.

De nombreuses personnes dans les universités créent des logiciels libre pour favoriser le résultat de leur recherche dans des domaines non exploités. Certaines entreprises aident à maintenir certains logiciels libres et ont ainsi leur mot à dire dans la direction du développement – il n'y a pas de solutions plus rapide pour obtenir une nouvelle fonctionnalité que de l'implémenter vous–même ou d'embaucher un consultant pour le faire. Les entreprises sont aussi intéressées par la plus grande souplesse et la possibilité de choisir entre plusieurs vendeurs pour le support.

Quelques autres personnes voient les logiciels libres comme un bien social qui démocratise l'accès à l'information et prévient d'une centralisation excessive de l'infrastructure de l'information mondial. Bien sûr, beaucoup d'entre nous trouve cela tout simplement sympa.

Debian est tant rattachée au logiciel libre que nous avons pensé qu'il serait bien de tout formaliser dans un document. Notre *Contrat Social* (*http ://www.debian.org/social_contract*) nous promet que Debian

[2] NDT : En anglais, le *free* de *free software* veut aussi bien dire libre que gratuit d'où une confusion souvent ignorée du profane entre logiciel libre et gratuit.

[3] Pourquoi les logiciels devraient être libres ?

[4] Notez bien que le terme « *hacker* » ne doit pas être confondu avec celui de « *cracker* ». Pour faire simple, un *hacker* est un être bienveillant tandis qu'un *cracker* est malveillant. Les films et les autres formes de média utilisent la plupart du temps à tord le terme *hacker* au lieu de celui de *cracker*

sera toujours composée à 100% de logiciels libres. Lorsque vous installez un paquet de la distribution *main* de Debian, vous pouvez être certain qu'il respecte nos Directives du Logiciel Libre.

Bien que Debian croit aux logiciels libres, il y a des cas où certaines personnes désirent ajouter des logiciels propriétaires sur leurs machines. Dans la mesure du possible, Debian le supporte ; bien que les logiciels propriétaires ne sont pas inclus dans la partie *main* de la distribution Debian, on peut parfois les trouver sur le site ftp dans le répertoire *non free*. Et il y a un nombre grossissant de paquets dont le but unique est d'installer des logiciels propriétaires que nous ne sommes pas autorisés nous–mêmes à distribuer !

Il est important de bien distinguer les logiciels *commerciaux* de ceux *propriétaires*. Les logiciels propriétaires ne sont pas des logiciels libres ; les logiciels commerciaux sont des logiciels vendus contre de l'argent. Debian autorise les logiciels commerciaux, mais pas les propriétaires, à faire partie de la distribution *main*. Rappelez–vous bien que l'expression *logiciel libre* ne se rapporte pas au prix ; il est parfaitement possible de vendre un logiciel libre. Pour un peu plus de clarification sur la terminologie employée, voir *http ://www.fsf.org/philosophy/categories.html*.

3.4 Au sujet de ce livre

Ce livre a été conçu pour les lecteurs fraîchement arrivés à Debian GNU–Linux. Il n'est pas nécessaire d'avoir des connaissances sur le système GNU–Linux ou tout autre système de type Unix mais il est nécessaire d'avoir quelques connaissances de base sur les ordinateurs et leurs composants. Il vous faut connaître les différentes parties d'un ordinateur et ce que l'on peut faire avec.

De façon générale, ce tutoriel essaie de vous aider à comprendre ce qui se passe à l'intérieur d'un système Debian. L'idée est de vous rendre apte à résoudre de nouveaux problèmes et de tirer le meilleur parti de votre ordinateur. Ainsi, vous rencontrerez plein de théories et de faits amusants dans les différents aspects des « How To[5] » de ce manuel.

Nous aimerions recevoir vos commentaires sur ce livre ! Vous pouvez joindre les auteurs debian–guide_fr–1.0_chap1.sgmlà *debian–guide@complete.org*. Nous sommes tout particulièrement intéressés par le fait de savoir comment ce livre a pu vous aider et comment le rendre meilleur. Si vous avez des commentaires ou bien que vous pensez que ce livre est la meilleure chose depuis l'apparition du pain brioché, alors, s'il vous plaît, envoyez–nous un courriel.

N'envoyez pas s'il vous plaît de questions techniques aux auteurs sur Debian car il y a plein de forum pour cela. Allez au chapitre 17 pour obtenir de l'information ou de l'aide. N'envoyez des courriels qu'au sujet du livre à proprement parlé à l'adresse ci–dessus.

3.4.1 Comment lire ce livre

La meilleure façon d'apprendre n'importe quel programme informatique est de l'utiliser. La plupart des gens pensent que se contenter de lire un livre sans utiliser le programme n'est pas bénéfique. La meilleur voie pour apprendre Unix et GNU–Linux est encore de les utiliser. Utilisez GNU–Linux pour tout ce que vous pouvez. Sentez vous libre d'expérimenter !

[5] Littéralement : Comment faire…

Debian n'est pas aussi intuitive que certains autres systèmes d'exploitation. Il vous sera certainement nécessaire de lire au moins quelques chapitres de ce livre. La puissance et la complexité de GNU–Linux le rendent difficile au premier abord mais bien plus rentable par la suite.

Pour apprendre, nous vous suggérons de lire un petit peu, et de jouer ensuite un petit peu. Continuez de bidouiller jusqu'à ce que vous soyez à l'aise avec les concepts, et alors seulement poursuivez la lecture du livre. Vous trouverez une variété importante de sujets traités parmi lesquels certains vous intéresseront. Après quelque temps, vous serez assez confiant pour utiliser des commandes sans savoir exactement ce qu'elles font. Ce sera une bonne chose.

Si jamais vous vous trompez en tapant une commande ou que ne savez pas comment sortir d'un programme, appuyez sur CTRL–c (la touche *Ctrl* et la lettre minuscule *c* doivent être pressées simultanément). Cela stoppe la plupart du temps le programme.

3.4.2 Conventions typographiques

Avant d'aller plus en avant, il est important de se familiariser avec les conventions typographiques utilisées dans ce livre.

Quand vous aurez à presser simultanément plusieurs touches, une notation comme CTRL–a sera utilisée. Cela signifie « pressez la touche *Ctrl* et pressez la touche de la lettre minuscule *a* ». Certains claviers ont à la fois la touche *Alt* et *Meta* ; la plupart des ordinateurs n'ont que la touche *Alt* mais celle–ci va simuler le comportement de la touche *Meta*. Donc, si vous n'avez pas de touche *Meta*, utilisez la touche *Alt* à la place.

Les touches comme *Alt* et *Meta* sont appelées des *modificateur* parce qu'elle change le mécanisme des touches standard comme la lettre *A*. Parfois, il est nécessaire d'avoir recours à plusieurs modificateurs simultanément. Par exemple, *Meta–Ctrl–a* signifie d'appuyer simultanément sur les touches *Meta*, *Ctrl* et *a* minuscule.

Quelques touches ont une notation spéciale. Par exemple, *Entrée* (*RET*, la touche avec une flèche descendante et allant à gauche), *Suppr* (*Suppr* ou quelque fois le *retour arrière* (la touche avec une flèche vers la gauche)), *Échap* (*Échappement* ou *Esc*). Ces touches portent généralement un nom bien explicite.

Les espaces utilisés dans les expressions signifient d'appuyer sur les touches séquentiellement. Par exemple, *CTRL–a x RET* signifie d'appuyer simultanément sur *Ctrl* et *a*, puis sur la lettre *x* et enfin sur la touche *Entrée*.

Dans certaines sections, le texte en *caractère gras* servira à marquer le texte de l'utilisateur, celui en *italique* des commentaires sur un point donné de la partie et tout autre type de texte servira au renvoi de la commande. Pour certaines commandes très courtes, vous pourrez rencontrer un autre format, en surimpression avec un `espacement fixe`.

Chapitre 4

Bien débuter

« Une journée de plusieurs centaines de kilomètres doit commencer par un petit pas » – Lao–Tseu

Maintenant que vous avez vu les idées et la philosophie se cachant derrière Linux et Debian, il est temps de l'installer sur votre matériel ! Nous allons commencer par parler des préparatifs à l'installation de Debian, notamment en partitionnant le disque et finir par comment démarrer l'installation de votre système.

4.1 Matériel supporté

Debian n'impose pas un matériel requis, si ce n'est celui supporté par le noyau Linux et les outils GNU.

Plutôt que de tenter de décrire toutes les différentes configurations matérielles qui sont supportées sur la plateforme PC, cette section contient des informations générales et des pointeurs où trouver de l'information supplémentaire.

Il y a deux excellents endroits pour vérifier des informations détaillées. La liste du matériel supporté par Debian (*http ://www.debian.org/releases/slink/i386/ch–hardware–req.en.html*) et le HOWTO du Projet de Documentation Linux *Compatibility–HOWTO* (*http ://metalba.unc.edu/LDP/HOWTO/Hardware–HOWTO.html*). Pour des informations sur le support des cartes vidéos, vous pouvez aussi jeter un coup d'oeil à *XFree86* (*http ://www.xfree86.org/*) sur leur site web.

4.1.1 Espace disque et de mémoire

Vous devez avoir au minimum 4 Mo de mémoire et 35 Mo d'espace disponible sur votre disque. Si vous désirez installer un nombre raisonnable de logiciels, avec en particulier le système X Window, et quelques programmes de développement ainsi que des bibliothèques, vous aurez certainement besoin d'au moins 300 Mo. Pour une installation complète, il vous faudra 800 Mo. Pour installer *tous* les paquets de Debian, il vous faudra probablement environ 2 Go. Actuellement, tout installer n'a pas de sens car certains paquets procurent le même service.

4.2 Avant de commencer

Avant de commencer, assurez–vous bien de sauvegarder toutes vos données présentes sur le système. La procédure d'installation peut détruire toutes les données d'un disque dur ! Les programmes utilisés lors de l'installation sont assez sûrs et nombreux ont des années d'existence et d'exploitation ; mais un faux mouvement de votre part peut vous coûter cher. Même après avoir sauvegardé vos données, réfléchissez bien lors de vos réponses et de vos actions. Deux minutes de réflexion peuvent vous épargner des heures de travail non nécessaire.

Debian s'efforce au mieux de partager la place avec un ou plusieurs autres systèmes d'exploitation. Si vous prévoyez cette option, assurez–vous d'avoir le cédérom original ou les disquettes pour pouvoir réinstaller les autres systèmes d'exploitation. Si vous repartitionnez votre disque de boot, il se peut que vous ayez à réinstaller le système d'amorçage[1] ou le système dans son intégralité.

4.2.1 Les informations nécessaires

Si votre ordinateur est connecté à un réseau vingt–quatre heures sur vingt–quatre (*i–e* par une connexion Ethernet ou similaire et non par une connexion `ppp`), vous devez demander à votre administrateur les informations suivantes :

- Votre nom d'hôte (« *hostname* ») ;
- Votre nom de domaine (« *domain name* ») ;
- L'adresse IP de votre ordinateur ;
- L'adresse IP de votre réseau ;
- Le masque réseau utilisé par votre réseau ;
- L'adresse de votre réseau de diffusion en usage sur votre réseau (« *broadcast* ») ;
- L'adresse IP de la passerelle de routage par défaut (« *gateway* »), si votre réseau en possède une ;
- Le système de votre réseau que vous devrez utiliser comme serveur DNS ;
- La façon dont vous vous connectez au réseau via *ethernet* ;
- Vérifiez si votre interface *Ethernet* est de type PCMCIA, et dans l'affirmative, relevez le type du contrôleur.

Si votre seule connexion réseau est celle du téléphone via `ppp`, ou tout autre type équivalent de connexion, il n'est pas nécessaire de s'occuper du paramètrage réseau avant la fin d'installation. Voir la partie 13.1 concernant le paramètrage d'une connexion `ppp` sous Debian.

4.3 Partitionner son disque dur

Avant d'installer la Debian sur votre ordinateur, il est généralement de bon ton de planifier à l'avance l'organisation de votre disque dur. Une partie de ce travail consiste à partitionner le disque.

[1] *boot loader* en anglais. C'est le responsable de la procédure de choix du système d'exploitation au démarrage.

4.3.1 Travaux préliminaires

Le partitionnement d'un disque consiste simplement au fait de le diviser en parties. Chaque partie sera indépendante des autres. C'est un peu comme ajouter des murs à l'intérieur d'une maison : après, décorer une pièce n'affecte pas l'aspect des autres pièces.

Si vous avez déjà un système d'exploitation sur votre disque (Windows 95, Windows NT, DOS, etc.) et que vous désirez installer Debian GNU–Linux au même endroit, il sera probablement nécessaire de repartitionner le disque. De manière générale, transformer une partition contenant déjà un système de fichier détruit toutes les informations le contenant. Ainsi, vous devez absolument faire une sauvegarde de toutes vos données avant tout repartitionnement. Pour reprendre l'analogie avec la maison, vous devrez probablement retirer tous les ornements d'un mur avant de le déplacer ou sinon, vous risqueriez des les endommager. Heureusement, il existe une possibilité pour certains utilisateurs ; voir partie 4.3.6 pour plus d'informations.

Dans le cas le plus simple, le système GNU–Linux demande au moins une partition pour fonctionner. Vous pouvez avoir une seule partition contenant tout le système d'exploitation et vos fichiers personnels. De nombreuses personnes optent pour un partitionnement plus important pour le système GNU–Linux. Il y a deux raisons de subdiviser son système avec un nombre plus important de petites partitions. La première est pour une raison de sécurité. Si quelque chose arrive et corrompt le système de fichiers, seule une partition est généralement affectée. Ainsi, il n'est simplement nécessaire que de remplacer la portion incriminée (à l'aide des sauvegardes que vous avez pris soin d'effectuer consciencieusement). Au minimum, vous allez devoir créer ce que l'on appelle généralement une *partition root* ou *partition racine*. Elle contient les composants essentiels du système. Si une autre partition est corrompue, vous pouvez toujours amorcer votre système pour essayer de le réparer. Cela peut vous épargner les ennuis d'une réinstallation complète du système à partir de zéro.

La seconde raison est généralement plus critique dans un environnement commercial, mais elle dépend énormément de la finalité du système. Supposons que quelque chose échappe à tout contrôle et qu'elle se mette à dévorer l'espace disque disponible. Si le programme en question a des privilège de super–utilisateur (*root*), le système lui réservant un pourcentage du disque non disponible pour les utilisateurs, vous pouvez tout à coup vous retrouver sans ressource disque. Ce n'est vraiment pas une bonne idée étant donné que le système en a besoin tout le temps (au moins au travers de la zone d'échange[2]) pour de nombreux usages. Par exemple, des courriels non sollicités, comme le *spam*, peuvent rapidement saturer une partition. En utilisant de nombreuses partitions, vous pouvez vous protéger de ce genre de problème. Pour garder l'exemple du courrier électronique, placer le répertoire */var/spool/mail* dans sa propre partition n'empêchera pas le système de continuer à fonctionner, même en cas de saturation par *spam*.

Une autre raison apparaît lorsque vous avez des disques IDE de grande capacité et que vous n'utilisez ni le mode LBA ni un pilote de surcharge.[3] Dans ce cas, il est nécessaire de placer la partition racine à l'intérieur des 1024 premiers cylindres de votre disque dur, ce qui représente généralement environ 524 méga–octets. Voir la partie 4.3.3 pour obtenir plus d'informations à ce sujet.

La plupart des gens pensent qu'une partition d'échange est aussi une nécessité absolue alors que ce n'est pas exact. Le *swap* est une zone de stockage du système d'exploitation qui l'utilise comme une *mémoire*

[2] plus communément appelée le *swap*.

[3] Voir le manuel de votre disque pour une description de ces fonctionnalités.

virtuelle en complément de la mémoire vive. La placer dans une partition séparée offre des performances accrues. Il est aussi possible de forcer Linux à utiliser un simple fichier comme zone d'échange mais ce n'est pas recommandé.

Le seul réel inconvénient à utiliser des partitions multiples est qu'il est généralement difficile d'appréhender à l'avance ses besoins. Si vous sous–dimensionnez vos partitions, soit vous aurez à réinstaller le système, soit vous aurez à jongler en déplaçant quantité de choses pour faire de la place dans la partie sous–taillée. D'un autre côté, si la partition est surdimensionnée, vous perdrez de la place qui pourrait être utilisée pour autre chose...

4.3.2 Planifier l'usage du système

Les besoins en place sur un disque et son schéma de partitionnement sont fortement dépendants du type d'installation que vous avez décidé de créer.

Pour vous faciliter la vie, Debian offre un certain nombre de *profils* par défaut dont certains sont énumérés un peu plus loin. Les profils sont de simples jeux de paquets pré–sélectionnés en vue d'une tâche particulière sur votre système. L'installation est facilitée car tous les paquets requis dans le profil voulu sont automatiquement marqués à installer. Chaque profil fournit la taille globale du système après installation. Même si vous n'utilisez pas ces profils, cette aparté est important pour planifier car il vous permet de vous donner une idée de la taille de votre ou vos partitions suivant vos besoins. La liste suivante donne quelques profils ainsi que la taille associée :

- `Server_std`
  Ceci est une petite configuration de serveur utile pour de très petites machines qui ne conviendraient plus à un usage courant pour un utilisateur. Elle possède par défaut un serveur FTP, un serveur web, DNS, NIS et POP. Cela occupe environ 50 Mo. Bien sûr, ceci correspond simplement à la taille des logiciels installés ; toutes les données que vous aurez à traiter sont à rajouter.
- `Dialup`
  Ceci est typique d'une configuration classique de bureau avec un système X Window, des applications graphiques, du son, de l'édition, etc. La taille de l'ensemble des paquets occupe près de 500 Mo.
- `Work_std`
  Cette configuration convient aux petites machines ne pouvant supporter X Window et ses applications graphiques. C'est aussi parfait pour un ordinateur portable ou mobile. Cela occupe environ 140 Mo. Il est possible aussi d'avoir une configuration sous X avec moins de 100 Mo.
- `Devel_comp`
  Ceci est un environnement de travail rassemblant tous les paquets de développement populaires comme PERL, C et C++. Cela demande environ 475 Mo. En rajoutant X et quelques autres paquets utiles, vous devriez tourner autour de 800 Mo pour ce type d'installation.

Garder en mémoire que ces tailles n'incluent pas toutes les données généralement présentes sur un disque, comme les fichiers des utilisateurs, les courriels et les données. Il est toujours préférable d'être généreux avec la taille de vos propres données et fichiers. De plus, le répertoire Debian */var* contient quantités d'informations du système. Les fichiers de gestionnaire d'installation des paquets peut facilement occuper 20 Mo de disque. En général, vous devriez allouer au moins 50 Mo pour le répertoire */var* car les journaux de bord[4]du système sont aussi stockés là.

[4]plus communément appelés les fichiers de *log*

4.3.3 Les limitations des disques PC

Les BIOS des PC ajoutent de nouvelles contraintes au partitionnement de disque. Il y a une limite au nombre de partitions *primaires* et *logique* qu'un disque peut contenir. En plus, il y a des limites à la taille de l'endroit où le BIOS va chercher ses informations d'amorçage. On peut trouver plein d'informations à ce sujet dans le mini–HOWTO Partition (*http ://metalab.unc.edu/LDP/HOWTO/mini/Partition.html*). Cette partie va inclure une bref résumé pour vous aider à affronter la plupart des situations.

Les partitions *primaires* sont les partitions originelles des disques durs de PC. On ne pouvait en créer plus de quatre. Pour s'affranchir de cette limitation, les partitions *étendues* ou *logiques* ont été inventées. En configurant une de vos partitions *primaires* en *étendue*, vous pouvez subdiviser la zone de cette partition en partitions logiques. Le nombre de partitions logiques que vous pouvez créer est beaucoup moins limité que le nombre de partitions primaires. Dans tous les cas, vous ne pouvez créer qu'une seule partition étendue par disque.

Linux limite le nombre de partitions par disque à quinze partitions pour les disques SCSI (trois partitions primaires utilisables et douze partitions logiques) et soixante trois partitions pour les disques IDE (trois partitions primaires et soixante partitions logiques).

La dernière chose qu'il est nécessaire de connaître au sujet du BIOS du PC est que votre partition racine – c'est–à–dire la partition contenant l'image du noyau – doit être incluse dans les 1024 premiers cylindres de votre disque. Puisque généralement votre partition racine est aussi votre partition d'amorçage, il faut vous assurer que votre partition racine se trouve bien à l'intérieur des 1024 premiers cylindres.

Si vous avez un disque de grande capacité, il se peut que vous utilisiez des techniques de translation de cylindres à l'intérieur de votre BIOS comme la translation LBA (On peut se référer pour plus d'informations au sujet des disques de grande capacité dans le *Large Disk mini–HOWTO* (*http ://metalab.unc.edu/LDP/HOWTO/mini/L*
Disk.html)). Si vous utilisez ce schéma de translation de cylindres, votre partition d'amorçage doit se trouver à l'intérieur de la représentation translatée des 1024 cylindres.

4.3.4 Nom de périphériques sous Linux

Les noms des disques et partitions sous Linux peuvent être différents que ceux utilisés par d'autres systèmes d'exploitation. Vous devez connaître ces noms que Linux utilise au moment de la création et du montage des partitions. On trouve un schéma élémentaire dans la table suivante.

Périphériques	Nom sous Linux
Premier lecteur de disquette	/dev/fd0
Second lecteur de disquette	/dev/fd1
Première partition sur /dev/hda	/dev/hda1
(typiquement C: dans d'autres OS)	
Cinquième partition sur /dev/hdc	/dev/hdc5
Seconde partition sur /dev/sdb	/dev/sdb2
Disque IDE sur la première nappe	/dev/hda
IDE en maître ou cédérom	

Disque IDE sur la première nappe IDE en esclave ou cédérom	/dev/hdb
Disque IDE sur la seconde nappe IDE en maître ou cédérom	/dev/hdc
Disque IDE sur la seconde nappe IDE en esclave ou cédérom	/dev/hdd
Premier disque SCSI	/dev/sda
Second disque SCSI et les autres	/dev/sbb et ainsi de suite
Premier port série (COM1 dans d'autres OS)	/dev/ttyS0
Second, troisième, etc. port série	/dev/ttyS1, /dev/ttyS2 etc.
Unités de sauvegarde SCSI (à rembobinage automatique)	/dev/st0, /dev/st1, etc.
Unités de sauvegarde SCSI (à rembobinage non automatique)	/dev/nst0, /dev/nst1, etc.
Cédéroms SCSI	/dev/scd0, /dev/scd1,

Les partitions de chaque disque sont représentées en ajoutant un chiffre au numéro de disque. Par exemple, les noms *hda1* et *hda2* représentent les première et seconde partitions du premier disque IDE de votre système. Linux représente les partitions primaires avec le nom du périphérique plus les nombres 1 à 4. Par exemple, la première partition primaire du premier disque IDE est */dev/hda1*. Les partitions logiques sont numérotées à partir de 5 donc la première partition logique est */dev/hda5*. Garder bien en mémoire que la partition étendue – c'est–à–dire la partition primaire contenant les partitions logiques – n'est pas utilisable en tant que tel. Ceci s'applique aussi bien aux disques SCSI qu'aux disques IDE.

Imaginons que vous ayez un système avec deux disques SCSI, un à l'adresse 2 et l'autre à l'adresse 4. Le premier disque, à l'adresse 2, est appelé *sda* tandis que le second est appelé *sdb*. Si le disque *sda* a trois partitions, elles seront appelées *sda1, sda2* et *sda3*. On applique le même raisonnement pour les partitions du disque *sdb*. Remarquez que si vous possédez deux adaptateurs pour bus SCSI (contrôleurs), l'ordre des périphériques peut porter à confusion. La meilleure solution dans ce cas est encore de regarder les messages de *boot*, en supposant que vous connaissez le modèle de vos périphériques.

4.3.5 Schéma de partitionnement recommandé

Comme décrit auparavant, il est recommandé d'avoir une petite partition racine séparée et une grosse pour */usr* s'il y a suffisamment de place. Pour la plupart des utilisateurs, ce partitionnement est très largement suffisant. Ceci est spécialement recommandé si vous avez un petit disque car la création de nombreuses partitions peut vous faire perdre de la place.

Dans certains cas, il se peut que vous ayez besoin d'une partition */usr/local* séparée si vous planifier l'installation de nombreux programmes non–inclus dans Debian. Si votre machine est un serveur de courriel, vous aurez besoin d'une partition séparée pour */var/spool/mail*. Mettre */tmp* sur sa propre partition de 20 à 32 Mo est aussi une bonne idée. Si vous préparez un serveur avec un grand nombre de comptes

d'utilisateurs, c'est généralement une bonne idée de préparer une grande partition séparée contenant */home* à des fins de sauvegarde des comptes des utilisateurs. De façon générale, le partitionnement varie énormément d'un ordinateur à l'autre et dépend essentiellement de son usage.

Pour des systèmes très complexes, vous devez aller jeter un coup d'oeil au *Multi Disks HOWTO* (*http ://metalab.unc.edu/LDP/*
Disk–HOWTO.html). Il contient de nombreuses informations techniques pour les gens désirant configurer des serveurs.

La taille de la partition de *swap* doit aussi être étudiée. Il y a beaucoup de points de vue à ce sujet. Une règle approximative qui ne fonctionne pas trop mal est d'avoir autant de *swap* que de mémoire vive, quoiqu'il ne soit pas nécessaire de dépasser 64 Mo pour la plupart des utilisateurs. Elle ne doit pas non plus être plus petite que 16 Mo non plus. Bien sûr, il y a des tas d'exceptions. Si vous essayez de résoudre simultanément 10 000 équations avec une machine dotée de 256 Mo de mémoire vive, vous pouvez avoir besoin d'un giga–octet (si ce n'est plus. . .) de *swap*.

Par exemple, considérons une machine dotée de 32 Mo de mémoire vive et d'un disque IDE de 1,7 Go sur */dev/hda*. Il y a une partition de 500 Mo pour un autre système d'exploitation sur */dev/hda1*. Une partition de *swap* de 32 Mo est utilisée en */dev/hda3* et le reste, environ 1,2 Go, sur */dev/hda2* est la partition Linux.

4.3.6 Partitionnement avant installation

Il existe deux moments différents pendant lesquels partitionner : avant ou après l'installation de Debian. Si votre ordinateur est entièrement consacré à Debian, vous pouvez partitionner pendant l'installation comme décrit dans la partie 5.5. Si vous avez une machine équipée de plus d'un système d'exploitation, vous devez généralement laisser l'autre système créer ses propres partitions.

Les paragraphes suivant contiennent des informations sur la manière de partitionner votre premier système d'exploitation avant l'installation de Debian. Remarquez que vous allez avoir besoin de savoir comment relier les noms des périphériques de votre système d'exploitation à ceux des partitions Linux.

Partitionner à partir de DOS ou Windows

Si vous manipuler des partitions FAT ou NTFS existantes, il est recommandé d'utiliser soit les outils natifs de Windows ou DOS, soit le schéma plus loin. Sinon, il n'est pas nécessaire de partitionner à partir de DOS ou Windows : les outils de partitionnement de Linux font généralement un travail de meilleure qualité.

Réduire une partition existante

Un des cas les plus classiques d'installation est de le faire sur un système contenant déjà le DOS (y compris Windows 3.1), Win32 (comme Windows 95, 98 ou NT) ou OS/2 et d'y ajouter Debian sur le même disque sans détruire l'ancien système. Comme décrit précédemment, réduire la taille d'une partition existante conduit presque immanquablement à la perte des données qu'elle contient, du moins sans prendre de précautions. La méthode que nous allons décrire ici, bien que ne garantissant pas de protéger vos données, marche très bien en pratique. Mais vous *devez faire une sauvegarde* par précaution.

Avant d'aller plus loin, vous devez savoir comment vous comptez diviser votre disque. La méthode décrite ici découpe seulement la partition initiale en deux morceaux. Un contiendra l'ancien système

d'exploitation. L'autre sera utilisé par Debian. Pendant l'installation de Debian, on vous laissera le choix de partitionner la partie Debian à votre convenance, pour le *swap* ou pour le système de fichiers.

L'idée est de rassembler l'ensemble des données de la partition à son début de façon à ce que l'on ne perde rien lors de la réduction. Il est important de déplacer les données par paquets aussi petits que possible de façon à réduire les chances qu'un fichier soit écrit en fin de partition et réduise ainsi la taille disponible que vous pouvez extraire de cette partition.

La première chose que vous devez faire est une copie de FIPS qui est disponible dans le répertoires *tools* de votre cédérom Debian. Cette disquette doit être bootable. Sous DOS, une disquette bootable peut être créée en utilisant la commande `sys a :` sur un support précédemment formaté ou bien `format a; /s` pour une disquette non formatée. Dézipez l'archive et copiez les fichiers RESTORRB.EXE, FIPS.EXE et ERRORS.TXT sur la disquette bootable. FIPS est fourni avec une très bonne documentation que l'on vous recommande de lire. Vous devez absolument la lire si vous utilisez une compression de disque ou un gestionnaire de disque. Créer la disquette et lisez la documentation *avant* de poursuivre.

L'étape suivante consiste à déplacer toutes les données au début de la partition. DEFRAG, qui est fourni en standard dans les versions de DOS 6.0 et supérieures, peut très bien effectuer cette tâche. Regardez la documentation de FIPS pour une liste d'outils analogues. Remarquez bien que si vous utilisez Windows 95 ou plus, vous devez lancer DEFRAG de cet environnement car le DOS ne connaît pas le système de fichier VFAT utilisé pour accepter les noms longs dans les versions 95 et ultérieures de Windows.

Après avoir défragmenté le disque (ce qui peut prendre un certain temps sur les disques de grande capacité), rebooter sur la disquette FIPS que vous avez créée. Taper tout simplement `a :\ fips` et suivez les instructions.

Remarquez au passage qu'il existe de nombreux autres gestionnaires de partitions que celui–ci, au cas où FIPS ne fonctionnerait pas bien pour vous.

4.3.7 Les étapes d'installation de Debian

Pour installer Debian la première fois, il vous faudra passer par les différentes étapes suivantes :

– Booter sur le système d'installation ;
– Configurer le système de base ;
– Installer le système de base ;
– Booter sur le système de base précédemment installé ;
– Installer le reste du système.

Booter sur le système d'installation, pour la première étape, s'effectue généralement avec la disquette de secours[5] ou sur le cédérom.

Aussitôt que vous avez booté le système Linux, le programme `dbootstrap` se lance et vous guide pour effectuer la seconde étape, la configuration du système de base. Cette étape est décrite en détail dans le chapitre suivant.

Le *système de base Debian* est un ensemble de paquets de base recquis pour faire tourner Debian avec un système minimal et de façon autonome. `dbootstrap` va vous l'installer à partir du cédérom, comme

[5] *Rescue floppy* dans la distribution Debian.

décrit dans la partie 5. À partir du moment où vous avez installé et configuré le système de base, votre machine est complètement autonome.

La dernière étape consiste à installer le reste du système Debian. Ceci va inclure les applications et les documents que vous utilisez vraiment sur votre ordinateur, comme le système X Window, des éditeurs, des interpréteurs de commandes (plus connus sous leur vocable anglaise *shell*) et des environnements de développements. Ceci se fait à partir du cédérom. À partir de ce point, vous aurez à utiliser les outils standard de gestion de paquets sous Debian comme `dselect`. Cette étape est décrite dans la partie 5.20.

4.4 Choisir son support d'installation

Il faut choisir dans un premier temps le support à partir duquel vous installerez le système. Ensuite, il faut choisir la méthode d'installation du système de base.

Pour booter sur la procédure d'installation, vous avez les choix suivants : un cédérom bootable, des disquettes ou un système de chargement au boot différent de celui fourni sous Linux.

Booter avec les cédéroms est bien entendu la manière la plus facile d'installer Debian. Mais toutes les machines ne supportent pas cette méthode et il se peut que vous ayez besoin d'utiliser les disquettes. Booter à partir de disquettes est supporté pour la plupart des plateformes. La façon de le faire est décrit à la partie 4.4.2.

4.4.1 Installation à partir du cédérom

Si votre machine supporte de booter sur un cédérom, vous n'avez pas besoin de disquette. Mettez le cédérom dans son lecteur, éteignez votre ordinateur et rallumez–le. Vous devriez voir un écran d'accueil avec un curseur de boot en bas. Vous pouvez passez alors directement à la partie 4.5.

Si votre ordinateur ne « voit » pas vos cédéroms Debian, la solution la plus simple est de se fabriquer deux disquettes de boot (description dans la partie suivante) et de les utiliser pour lancer Debian. Mais ne vous inquiétez pas : Debian en aura alors fini avec les deux disquettes et elle trouvera le cédérom sans problème.

4.4.2 Installation à partir de disquettes

Ce n'est pas difficile de booter à partir des disquettes. En effet, votre cédérom contient toutes les informations pour créer les disquettes nécessaires. Il vous sera demandé d'avoir deux disquettes. Nommez la première *Disquette d'installation–secours Debian 2.1* et la seconde *Disquette de modules–pilotes Debian 2.1*.

Créer des disquettes à partir des images de disquettes

Les images des disquettes sont des fichiers contenant l'ensemble complet des éléments de la disquette de boot sous forme binaire. Les images de disquette, comme *resc1440.bin*, ne peuvent être copiées

directement sur la disquette. Un programme spécial est utilisé pour recopier les images des fichiers au format binaire.

La première étape consiste à obtenir une invite DOS. Dans Windows 95, et supérieur, vous pouvez l'obtenir en double–cliquant sur l'icône MS–DOS ou en allant dans *Démarrer –> Programme –> MS–DOS*. Ensuite, insérer le cédérom Debian GNU–Linux dans son lecteur. Déplacez–vous alors dessus. Dans la plupart des cas, il s'agit de D :

```
C:\WINDOWS>D:
```

Maintenant, déplacez–vous dans le répertoire contenant les images des disquettes :

```
D:\> CD \DISTS\SLINK\MAIN\DISKS-I386\2.1.8-1999-02-22
```

Si vous obtenez une erreur, vérifiez bien ce que vous avez tapé. Si l'erreur persite, déplacez–vous manuellement dans `\DISTS\SLINK\MAIN\DISKS-I386` et taper alors `DIR` et allez alors dans le répertoire précité. Remarquez que les commandes ci–dessus, et peut–être celles données un peu plus loin, peuvent tenir sur une seule ligne sur votre écran même si elles tiennent sur plusieurs lignes ici.

Maintenant, vous êtes prêt à créer la première des deux disquettes. Lancer le programme `rawrite2` pour les écrire :

```
D:\DISTS\SLINK\MAIN\DISKS-I386\2.1.8-1999-02-22>rawrite2
RaWrite 2.0 - Write disk file ta raw floppy diskette[6]
```

`Rawrite2` lance ainsi ses messages de démarrage. Ensuite, il demande le nom du fichier et le lecteur de disquette :

```
Enter disk image source file name[7]: resc1440.bin
Enter target diskette drive[8]: a:
```

`Rawrite2` vous demande maintenant de bien vouloir insérer une disquette dans le lecteur de disquette. Faîtes–le et appuyez sur *Entrée*

```
Please insert a formatted diskette into drive A: and press -ENTER-
[9]:
```

À ce moment, `rawrite2` va créer la première des deux disquettes. Il vous faut répéter le processus pour la seconde disquette :

[6] Pour écrire un fichier du disque dur vers une disquette au format binaire.

[7] Entrer le nom du fichier image cible

[8] Entrer le nom du lecteur de disquette

[9] Veuillez insérer une disquette formatée dans le lecteur A et appuyez sur Entrée

```
D:\DISTS\SLINK\MAIN\DISKS-I386\2.1.8-1999-02-22>rawrite2
RaWrite 2.0 - Write disk file ta raw floppy diskette
Enter disk image source file name: drv1440.bin
Enter target diskette drive: a:
Please insert a formatted diskette into drive A: and press -ENTER-
:
```

À partir de maintenant, les disquettes sont prêtes : vous pouvez les utiliser pour booter.

Booter sur Debian

Vous êtes maintenant prêt à booter sur Debian. Éteignez votre système d'exploitation en cours et placez la disquette d'installation–secours dans le lecteur de disquette. Rallumez votre ordinateur. Vous devriez avoir un écran de bienvenue et un curseur de prompteur en bas.

4.5 Booter sur la procédure d'installation

Vous devriez avoir à ce stade un curseur de boot. Presser simplement sur *Entrée* à cet endroit.

Après avoir appuyé sur la touche *Entrée*, vous devriez voir le message `Loading...` et ensuite `Uncompressing Linux....` et une quantité d'informations partout sur l'écran, voire sur plusieurs, au sujet du matériel de votre système. Généralement, vous pouvez ignorer ces messages. Linux recherche un grand nombre de périphériques variés et vous indique ce qu'il trouve et ce qu'il ne trouve pas. Ne vous inquiétez pas de tous ces messages à ce moment là. Attendez simplement jusqu'à ce que vous aperceviez l'écran de choix en couleur. Si vous avez des problèmes, voir le chapitre 17.

Chapitre 5

Installation pas à pas

`dbootstrap` est le nom du programme qui est lancé après avoir booté sur la procédure d'installation. Il est responsable de la configuration initiale du système et de l'installation du système de base.

Le travail principal de `dbootstrap` et celui de la configuration de base du système sont de configurer certains éléments basiques de votre système. Par exemple, cela inclut votre adresse IP, votre nom de domaine et quelques autres paramètres de configuration réseau, en cas de besoin. Cela comprend aussi la configuration des modules du noyau qui sont les pilotes chargés par le noyau. Ces modules comprennent les pilotes de sauvegarde externe, les pilotes réseau, le support multilingues et plein d'autres périphériques. Configurer ces choses fondamentales est fait prioritairement car elles sont souvent nécessaires au bon fonctionnement ultérieur de la machine lors des étapes de configurations suivantes.

`dbootstrap` est une application simple orientée caractères. Elle est très facile à utiliser. Généralement, elle vous guide à travers chaque étape du processus d'installation de façon linéaire. Vous pouvez aussi revenir en arrière et répéter une étape si vous vous êtes trompé. La navigation à l'intérieur de `dbootstrap` s'effectue à l'aide des flèches du clavier, des touches *Entrée* et *Tab*.

5.1 Sélection d'un écran monochrome ou couleur

Une fois que le système a terminé de booter, `dbootstrap` est invoqué. La première chose que `dbootstrap` vous demande concerne votre écran. Vous devez voir une fenêtre de dialogue *Select Color or Monochrome display* (Sélectionner un écran couleur ou monochrome). Si votre moniteur est capable d'afficher de la couleur, appuyez sur *Entrée*. L'écran devrait alors passer du noir et blanc en couleur. Déplacez–vous à l'aide des flèches sur le bouton *Next* (suivant) et pressez alors de nouveau la touche *Entrée* pour continuer l'installation.

5.2 Menu principal d'installation de Debian GNU/Linux

Il se peut que vous aperceviez une boîte de dialogue disant *The installation program is determining the current state of your system and the next installation step should be performed* (Le programme

d'installation est en train de déterminer l'état actuel de votre système et l'étape suivante d'installation ne devrait pas tarder). C'est une phase pendant laquelle le programme d'installation cherche à détecter automatiquement ce que vous aurez probablement besoin de faire ensuite. Dans certains cas, il se peut que vous n'aperceviez même pas cette boîte de dialogue.

Pendant toute la procédure d'installation, vous aurez affaire au menu principal, intitulé *Debian GNU–Linux Installation Main Menu* (Menu principal d'installation de Debian GNU–Linux). Les choix situés en haut du menu changent au fur et à mesure pour indiquer votre progression dans l'installation du système. Phil Hughes écrivait dans le *Linux Journal* (*http ://www.linuxjournal.com*) que vous pouviez apprendre à un poulet à installer une Debian ! Il soulignait ainsi le fait que le menu de la procédure d'installation consistait essentiellement à appuyer sur la touche *Entrée*. Le premier choix du menu d'installation est l'action suivante que vous aurez à effectuer si le système ne détecte pas que vous l'avez déjà fait. Si vous sélectionnez *Next* (suivant) ici, l'étape suivante dans l'installation du système est sélectionnée.

5.3 Configurer le clavier

Assurez–vous que le bouton *Next* est mis en surbrillance et appuyez sur *Entrée* pour entrer dans le menu de configuration du clavier.

Déplacez le curseur lumineux vers le clavier sélectionné et appuyer sur *Entrée*. Utilisez les flèches pour vous déplacer. Dans de nombreux cas, vous n'aurez qu'à sélectionner *U.S. layout*[1].

5.4 Dernière chance de sauvegarder vos données !

Ne vous a–t–on pas déjà dit de sauvegarder vos disques ? Voici la première occasion de les détruire et la dernière chance de sauvegarder vos vieux systèmes. Si vous n'avez encore sauvegardé vos disques, sortez la disquette ou le cédérom de son lecteur, éteignez le système puis faîtes vos sauvegardes.

5.5 Partitionner un disque dur

Quelque soit le choix proposé par le menu de sélection *Next*, vous pouvez utiliser les flèches « haut » et « bas » pour sélectionner *Partition a Hard Disk* (Partitionner un disque dur). Allez–y et à partir de là, appuyez sur *Entrée*.

Le menu de *Partition a Hard Disk* vous présente une liste de type de disques que vous pouvez partitionner et lance une application de partitionnement appelée `cfdisk`. Vous devez créer au moins une partition *Linux native* (de type 83) et vous aurez probablement besoin d'une partition *Linux swap* (de type 82) comme expliqué plus loin dans cette partie.

Vous devez créer maintenant les partitions dont vous aurez besoin pour installer Debian. Par exemple, on supposera que vous partionnerez un disque dur vierge.

[1] NDT : surtout pas pour nous francophones car vous risqueriez de vous arracher les cheveux...Prenez bien soin de sélectionner un clavier conforme à vos origines (française, suisse, belge ou canadienne) si tel est le cas.

La partition de boot doit se trouver à l'intérieur des 1024 premiers cylindres de votre disque dur (voir paragraphe 4.3.3). En gardant cela à l'esprit, utilisez la flèche droite pour mettre en surbrillance le menu *New* (nouveau) et pressez alors sur *Entrée*. On vous demandera alors de créer soit une partition *primaire* soit une partition *logique*. Pour vous assurer que la partition contenant l'information de boot soit bien dans les 1024 premiers cylindres, créez en premier votre partition primaire. La partition primaire sera votre partition *Linux native*.

Mettez en surbrillance le menu principal et appuyez sur la touche *Entrée*. Vous devez ensuite entrer la largeur de votre future partition. Si vous ne savez pas quelle taille lui donner, allez voir la partie 4.3.2. Ne pas oublier de laisser une place suffisante pour la partition de *swap* (voire la partie 4.3.5). Entrez la taille de la partition que vous désirez et presser sur *Entrée*. Ensuite, il vous sera demandé si vous voulez la placer au début de l'espace disponible ou à la fin. Placez la au début pour être sûr d'être à l'intérieur des 1024 premiers cylindres. Mettez en surbrillance *Beginning* et tapez sur *Entrée*. À ce moment, vous devriez être retourné à l'écran principal. Remarquez bien que la partition que vous venez de créer est maintenant affichée. Par défaut, c'est une partition *Linux native*. Cette partition doit maintenant être rendue amorçable. Assurez–vous que le menu *Bootable* est en surbrillance et appuyez sur la touche *Entrée*. On doit maintenant voir apparaître la champ *Boot* dans cette partition dans la colonne *Flags* .

IMPRESSION D'ÉCRAN DE CFDISK

Avec la place restante, créez une nouvelle partition primaire. En utilisant la flèche de direction du clavier, mettez en surbrillance le champ *free space* (espace disponible) dans la liste des partitions. Mettez en surbrillance maintenant le champ *New* et faîtes de même que lorsque vous avez fabriqué la première partiton primaire. Remarquez qu'elle est aussi affichée comme une partition *Linux native*. Puisque vous allez la dédier au *swap*, il va falloir la signaler comme telle. Assurez–vous de bien l'avoir mis en surbrillance et ensuite appuyez sur la flèche gauche du clavier jusqu'à ce que le menu *Type* soit mis en surbrillance. Pressez alors sur la touche *Entrée*. On vous présente alors une liste de types de partitions supportées. Le type *Linux swap* devrait déjà être sélectionné. Si tel n'était pas le cas, entrez le numéro dans la liste correspondant au *swap* (82) et pressez sur *Entrée*. Votre partition de *swap* doit maintenant se présenter sous forme de *Linux swap* dans la colonne *FS Type* de l'écran principal.

Votre écran `cfdisk` devrait ressembler à celui de la figure 5.1. Les chiffres peuvent différer mais les colonnes *Flags* et *FS Type* doivent être identiques.

Jusqu'à maintenant, rien sur votre disque n'a été modifié. Si vous êtes content de votre partionnement, appuyez sur la flèche gauche jusqu'à *write* et appuyez alors sur la touche *Entrée*. Votre disque dur est maintenant partitionné. Quittez l'application `cfdisk` en sélectionnant le menu *Quit*. Une fois quitté `cfdsik`, vous devez vous retrouvez dans `dbootstrap`, l'application d'installation de Debian.

5.6 Initialiser et activer une partition de *swap*

Cela doit être le choix suivant après avoir créé une partition de disque (*initialize and Activate a Swap Partition*). Vous avez le choix d'initialiser et d'activer une nouvelle partition de *swap*, d'activer une partition déjà initialisée ou bien de faire sans partition de *swap*.

Une partition de *swap* est chaudement recommandée mais vous pouvez faire sans si vous insistez et que votre système a plus de 4 Mo de mémoire vive. Si vous désirez le faire, sélectionnez le choix *Do Without a Swap Partition* (Faire sans partition de *swap*) du menu et allez directement à la partie suivante.

Il est toujours possible de réinitialiser une partition de *swap*. Ainsi, sélectionnez *Initialize and Activate a Swap Partition* (Activer et initialiser une partition de *swap*) si vous n'êtes pas sûr de ce que vous faîtes. Dans un premier temps, le menu va vous proposer une boîte de dialogue dans laquelle vous pourrez lire *Please select the partition to activate as a swap device* (veuillez sélectionner la partition à activer comme *swap*). Le choix par défaut devrait être la partition de *swap* que vous venez de paramétrer ; si tel est le cas, appuyez simplement sur la touche *Entrée*.

Ensuite, on vous propose d'activer l'option de recherche des blocs non lisibles, causés par des défauts sur la surface des plateaux des disques sur l'intégralité de la partition. Ceci est très utile si vous avez des disques MFM, RLL ou de vieux SCSI. Dans tous les cas, cela ne peut pas faire de mal (bien que cela demande pas mal de temps). La plupart des disques modernes fonctionnant correctement n'ont pas besoin de cette étape car ils possèdent leur propre mécanisme interne pour s'affranchir des blocs défectueux.

Finalement, vous obtenez un message de confirmation car l'initialisation va détruire toutes les données présentes antérieurement sur la partition. Si tout est correct, sélectionnez *Yes*. L'écran s'allumera lorsque le programme d'initialisation tournera.

5.7 Initialiser une partition Linux

À cet endroit, le choix suivant devrait s'intituler *Initialize a Linux Partition* (Initialiser une partition Linux). Si ce n'est pas le cas, soit vous n'avez pas achevé correctement le processus de partitionnement, soit vous n'avez pas rempli un des menus concernant la partition de *swap*.

Vous pouvez initialiser une partition Linux, ou bien vous pouvez en monter[2] une précédemment initialisée. Remarquez bien que `dbootstrap` ne fera pas de mise à jour d'un vieux système sans le détruire. Si vous êtes en train de mettre à jour, Debian peut très bien le faire tout seul et il n'est pas nécessaire d'utiliser `dbootstrap`. Les *releases notes* de la Debian 2.1 contiennent les instructions de mise à jour[3].

Si vous utilisez de vieilles partitions non vides, c'est–à–dire que vous voulez juste nettoyer ce qu'il y a dessus, il suffit de les initialiser (ce qui écrase tous les fichiers). De plus, vous devez initialiser toutes les partitions que vous avez créées lors de l'étape de partitionnement. La seule raison pour laquelle vous pourriez omettre d'initialiser une partition à cette étape serait de monter une partition que vous auriez déjà faîte à l'aide des disquettes d'installation.

Sélectionnez le menu *Next* pour initialiser et monter la partition de disque « / ». La première partition que vous montez ou que vous initialisez sera montée comme « / » (prononcer *root*[4] (racine)). On vous proposera, à l'instar de l'initialisation du *swap* de vérifier les blocs défectueux. Cela ne mange pas de pain de le faire mais cela peut prendre dix minutes ou plus si vous possédez des disques de grande capacité.

Une fois la partition / de montée, le menu suivant devrait être *Install Operating system and modules* (Installer le système d'exploitation et ses modules) à moins que vous n'ayez pas terminé une des étapes

[2] francisation de *mount* qui est passé dans le langage courant. On trouve parfois *mounter* qui est assez laid il faut bien l'avouer :–)

[3] http ://www.debian.org/releases/slink/i386/releases–notes/ch–upgrading.en.html

[4] se prononce « route »…

précédentes. Vous pouvez utiliser les flèches de déplacement pour sélectionner les menus d'initialisation ou de montage de partitions de disques si vous avez d'autres partitions à fabriquer. Si vous avez créé des partitions séparées pour */var*, */usr* ou d'autres systèmes de fichiers, c'est le moment de les initialiser et de les monter.

5.7.1 Monter une partition précédemment initialisée.

Un pendant à l'étape *Initialise a Partition* (Initialiser une partition) est celle de *Mount a Previously–Initialised Partition* (Monter une partition précédemment initialisée). Utilisez la si vous recommencez une installation qui a été interrompue ou bien si vous montez des partitions déjà initialisées.

5.8 Installation du système d'exploitation et ses modules

Cela doit être l'étape suivante après avoir monté la partition racine, sauf si vous avez déjà effectué cette tâche précédemment avec `dbootstrap`. Vous devrez confirmer dans un premier temps que le périphérique que vous avez monté sur la partition racine est bien le bon. Par la suite, il vous sera proposé un ensemble de périphériques à partir desquels vous pourrez installer le noyau et les modules ; ce sera généralement soit un cédérom soit le premier lecteur de disquette.

Si vous faîtes l'installation à partir de disquettes, vous aurez à travailler avec la disquette de secours (*Rescue Floppy*) qui est déjà probablement dans le lecteur et ensuite avec la disquette de pilotes (*Drivers Floppy*).

5.9 Configuration du support PCMCIA

Il y a une autre étape *avant* la configuration des modules de pilotes de périphériques appelée « Configuration du support PCMCIA » (*Configure PCMCIA Support*). Elle est utilisé pour activer le support PCMCIA.

Si vous avez une extension PCMCIA mais que vous ne l'utilisez pas pour installer votre système Debian (c'est–à–dire installation avec une carte ethernet PCMCIA), il n'est pas nécessaire de configurer le support PCMCIA ici. Il est très facile de le configurer et de l'activer plus tard, à la fin de l'installation. Dans tous les cas, si vous devez installer quand même une extension PCMCIA pour le réseau, vous devez la choisir ici et son support sera configuré antérieurement à celui du réseau.

Si vous avez besoin de PCMCIA, sélectionnez l'autre choix en dessous de « Configuration des modules de pilotes de périphériques » (*Configure Device Driver Modules*). On vous demandera alors le nom du contrôleur PCMCIA de votre système. Dans la plupart des cas, ce sera `i82365`. Quelques fois, vous pourrez aussi rencontrer `tcic`. Votre vendeur–fournisseur d'ordinateur portable doit vous procurer l'information. Vous pouvez généralement laisser les autres champs d'options vides. Attention, certains matériels ont des besoins particuliers ; le *PCMCIA–HOWTO* (*http ://metalab.unc.edu/LDP/HOWTO/PCMCIA–HOWTO.html*) contient plein d'informations au cas où les options par défaut ne conviendraient pas.

Dans certains cas exceptionnels, il se peut que vous ayez à modifier le fichier */etc/pcmcia/config.opts*. Vous devez ouvrir votre seconde console virtuelle (*Alt–F2*) et éditer le fichier dedans pour reconfigurer

votre extension PCMCIA. Vous pouvez aussi forcer un rechargement des modules par l'intermédiaire de `insmod` et `rmmode`.

Une fois que votre interface PCMCIA est correctement configurée et installée, il vous faut passer à la configuration des pilotes de périphériques comme décrit dans le paragraphe suivant.

5.10 Configuration des modules de pilotes de périphériques

Sélectionnez le menu *Configure Device Driver Modules* (configuration des modules de pilotes de périphériques) et passez en revue les périphériques présents sur votre système. Configurez les pilotes de ces périphériques et ces derniers seront chargés lors des amorçages de votre système.

Il n'est pas nécessaire de tout configurer à cette étape ; ce qui est crucial est d'avoir configuré les périphériques requis pour l'installation d'un système de base.

Une fois le système installé, vous pouvez reconfigurer les modules à +tout moment grâce au programme `modconf`.

5.10.1 Configuration du réseau

Vous aurez à configurer le réseau même si vous n'en avez pas. Dans ce cas, vous n'aurez qu'à répondre aux deux premières questions – *Choose the hostname* (Choisir le nom de domaine) et *Is your systeme connected to a network ?* (Est–ce que votre système est connecté à un réseau ?).

Si vous êtes connecté à un réseau, les informations de la partie 4.2.1 vous seront nécessaires. Si votre connexion principale au réseau utilise `ppp`, il ne vous faut PAS configurer le réseau ici.

`dbootstrap` va vous poser un certain nombre de questions au sujet de votre réseau. Vous les obtiendrez à partir du paragraphe 4.2.1. Le système va au final faire un résumé des informations réseau et vous demander une confirmation. Ensuite, vous devrez spécifier le périphérique réseau que votre connexion réseau principale utilise. Généralement, il s'agit de `eth0` (le premier périphérique Ethernet). Sur un ordinateur portable, on rencontre plus souvent comme périphérique réseau principal une interface `pcmcia`.

Voici quelques détails techniques pratiques : le programme s'assure que l'adresse réseau IP est bien la résultante d'un `AND` de votre adresse IP et de celle de votre masque réseau. Il s'assure aussi que votre adresse de réseau de diffusion est bien la résultante d'un `OR` de votre adresse IP avec l'inverse des bits de l'adresse du masque réseau. Il s'assure aussi que votre passerelle de routage est identique à votre serveur DNS. Vous pouvez les changer une fois le système installé, si nécessaire, en éditant le fichier */etc/init.d/network* (sur un système Debian, les *daemons* sont lancés par scripts dans le répertoire */etc/init.d*).

5.11 Installation du système de base.

Lors la phase *Install the Base System* (Installation du système de base), on vous proposera une liste de périphériques à partir desquels installer votre système de base. Vous devrez sélectionner ici le périphérique

cédérom.

Il apparaîtra alors un champ pour spécifier le chemin vers le fichier *base2_1.tgz*. Si vous avez une version officielle du support Debian, la valeur par défaut devrait être correcte. Sinon, entrez le chemin où le système de base peut être trouvé, relativement au point de montage du support. Comme lors de l'étape *Install Operating System Kernel and Modules*, vous pouvez soit laisser `dbootstrap` trouver seul le fichier ou taper le chemin dans le champ.

5.11.1 Configuration du système de base

À ce moment, vous avez chargé tous les fichiers qui font un système Debian minimal mais vous devez faire quelques ajustements de configuration avant que le système ne se lance.

On vous demandera de sélectionner la zone horaire (*time zone*). Il y a de nombreuses façons de spécifier votre zone horaire ; nous vous suggérons d'aller dans le tableau *Directories* et de sélectionner votre pays (ou continent). Cela change les zones horaires disponibles donc allez–y et sélectionnez votre localité (c'est–à–dire, pays, province, état ou ville) dans le tableau *Timezones*.

Ensuite, on vous demandera si votre horloge système est réglée sur l'heure GMT ou sur une heure locale. Sélectionnez GMT (c'est–à–dire, répondez *Yes*) si vous ne désirez faire tourner que Linux sur votre ordinateur ; sélectionnez l'heure local (*local time*) (c'est–à–dire, *No*) si vous utilisez un système autre que Debian. Les systèmes Unix en général (et Linux en particulier) utilisent l'heure GMT comme horloge système en interne et la convertissent en heure locale pour les utilisateurs. Cela permet de garder des traces des sauvegardes journalières au fil des ans et autorise aussi un utilisateur à se loguer à partir d'une autre zone horaire pour utiliser individuellement son heure locale sur son terminal.

5.11.2 Faire que Linux soit bootable directement à partir du disque dur

Si vous choisissez de booter directement sur Linux à partir du disque dur, on vous demandera d'installer un secteur d'enregistrement principal (*master boot record* ou encore *MBR*). Si vous utilisez un gestionnaire d'amorce (*boot manager*) (et c'est probablement le cas si vous ne savez pas ce que c'est) et que vous n'avez pas d'autres systèmes d'exploitation sur votre machine, répondez *Yes* à cette question[5]. Remarquez bien que dans ce cas, il ne vous sera plus possible de booter le DOS normalement sur votre machine par exemple. Faîtes donc bien attention. Si vous répondez *Yes*, la question suivante concernera l'endroit chercher Linux pour booter au démarrage. Cela doit se régler dans la *partition bootable* – celle qui doit être chargée à partir du disque dur.

Remarquez bien que booter différents systèmes d'exploitation sur une seule machine relève un peu de la cuisine. Ce livre ne documente pas l'ensemble des nombreux gestionnaires d'amorçage, qui varient beaucoup selon l'architecture et même selon les sous architectures. Vous devrez lire la documentation de votre gestionnaire d'amorçage pour avoir plus d'informations. Dans tous les cas, gardez bien à l'esprit qu'en travaillant sur les gestionnaires d'amorçage, il ne faut jamais être trop prudent...

Le gestionnaire d'amorçage standard pour les architectures i386 est appelé `LILO`. C'est un programme complexe qui offre plein de fonctionnalités, avec entre autre, un support du lancement de DOS, NT et

[5] Make Linux Bootable Directly from the Hard Disk

OS/2. Pour obtenir plus d'informations à ce sujet, vous pouvez lire la documentation dans */usr/doc/lilo* après que votre système ait été configuré.

5.12 Fabriquer une disquette de boot

Vous devrez faire une disquette de boot (*Boot floppy*) même si vous avez prévu de booter directement à partir du disque. La raison est qu'il est toujours possible d'avoir un défaut d'installation à partir de `dbootstrap` mais jamais à partir d'une disquette. Sélectionnez *Make a Boot Floppy* (Fabriquer une disquette de boot) à partir du menu principal et insérer une disquette vierge directement. Assurez–vous que la disquette n'est pas protégée en écriture car le logiciel va la formater et écrire dessus. Écrivez dessus *Boot personnalisé* et protégez la en écriture une fois qu'elle a été fabriquée.

5.13 Le moment de vérité

Le premier boot de votre système de son propre chef est ce que les ingénieurs en électronique appellent le *test de la fumée*. Si vous avez une disquette dans son lecteur, retirez–la. Puis sélectionnez le menu *Reboot the system* (Rebooter le système).

Si vous bootez directement sur Debian et que le système ne se lance pas, utilisez soit votre support original d'installation (par exemple, les disquettes de secours), soit votre disquette de boot personnalisé si vous en avez créé une, et relancez votre système. Si vous n'utilisez pas la disquette de boot personnalisé, il vous sera probablement nécessaire d'ajouter des arguments lors du boot. Si vous bootez via la disquette de secours ou tout autre support similaire, il vous sera nécessaire de spécifier `rescue root=rootfs` où `rootfs` est votre partition racine, comme */dev/hda1*.

Debian devrait alors booter, et vous devriez voir les mêmes messages que lorsque vous avez booté la première fois le système, avec ensuite quelques nouveaux messages.

5.14 Ajouter un mot de passe root

Le compte root est aussi appelé le compte *super–utilisateur* ; c'est un utilisateur qui permet d'outrepasser tous les systèmes de protection sur votre système. Le compte root ne devrait être utilisé que pour des tâches d'administration et pour une durée aussi courte que possible.

Chaque mot de passe créé devrait contenir six à huit caractères, mélangeant minuscules et majuscules ainsi que des caractères de ponctuation. Faîtes très attention en choisissant le mot de passe de root car ce compte est vraiment très sensible. Évitez tous mots contenus dans un dictionnaire ou bien tout renseignement personnel qui peut être deviné.

Si quelqu'un vous demande un jour le mot de passe de root, soyez très prudent. Vous ne devriez jamais avoir à le donner, à moins de ne pas être le seul à administrer la machine.

5.15 Créer un compte utilisateur ordinaire

Le système va alors vous demander de créer un compte utilisateur ordinaire (*Create an Ordinary User*). Ce compte pourrait être votre compte principal personnel. Vous ne *devez pas utiliser* le compte root pour une utilisation ordinaire quotidienne ou comme compte personnel.

Pourquoi donc ? Il est beaucoup plus difficile d'endommager le système en étant un simple utilisateur qu'en root ; le système de fichiers est bien protégé. Une autre raison est que vous pouvez à votre insu faire tourner un *troyen*[6] – c'est–à–dire un programme qui pourra utiliser les avantages du super–utilisateur pour compromettre la sécurité du système dans votre dos...Tout bon livre d'administration Unix traite ce sujet en détail. Il est nécessaire d'en lire un si ce sujet est nouveau pour vous.

Nommez le compte utilisateur à votre convenance. Si votre nom est John Smith, vous pouvez utiliser « smith », « john », « jsmith » ou « js ».

5.16 Support des mots de passe cachés.

Ensuite, on vous demandera si vous voulez activer les mots de passe cachés (*shadow Password*[7]). C'est un système d'authentification qui sécurise un peu plus votre système Linux. Nous vous recommandons ainsi de l'activer. La reconfiguration du *shadow password* peut être faite ultérieurement avec le programme `shadowconfig`.

5.17 Retirer le support PCMCIA

Si vous n'utilisez pas d'interface PCMCIA, vous pouvez choisir de la retirer maintenant. Cela permettra de faire un démarrage plus propre ; cela permet de remplacer votre noyau plus facilement (l'interface PCMCIA requiert un grand nombre de dépendances entre les pilotes PCMCIA, les modules du noyau et le noyau proprement dit). De façon général, vous n'avez pas besoin du support PCMCIA si vous n'utilisez pas de portable...

5.18 Sélection et installation de *profil*

Le système va maintenant vous demander si vous désirez utiliser les configurations prédéterminées offertes par Debian. Il est toujours possible d'installer paquet par paquet ce que l'on veut installer sur une nouvelle machine. C'est la tâche de `dselect` dont la description se trouve plus loin. Mais cela peut être une longue tâche avec les milliers de paquets disponibles dans Debian !

Ainsi, vous avez la possibilité de choisir à la place des *tâches* ou des *profils*. Une *tâche* est la finalité du travail que vous allez effectuer avec la machine, comme la programmation en PERL, le traitement HTML ou bien la mise en page de document en chinois. Vous pouvez choisir de nombreuses *tâches*. Un

[6] programme planqué sous un autre pour ouvrir des brèches de sécurité dans votre système

[7] terme tellement usité qu'il vaut mieux le connaître...

profil est une catégorie de machines, comme serveur de réseau ou station de travail personnelle. À la différence des tâches, vous ne pouvez choisir qu'un profil à la fois.

Pour résumé, si vous êtes pressé, choisissez un profil. Si vous avez plus de temps, choisissez le profil personnalisé (*Custom Profile*) et sélectionnez un ensemble de tâches. Si vous avez beaucoup de temps et que vous désirez être minutieux avec le contrôle de ce qui sera ou non installé, sortez de cette étape et lancez–vous dans la toute puissance de `dselect`.

Ensuite, vous allez entrer dans `dselect`. Si vous avez sélectionné des tâches ou des profils, n'oubliez pas de sauter l'étape *Select* (Sélectionner) de `dselect` car les sélections ont déjà été effectuées.

Un petit avertissement en ce qui concerne la taille des tâches telle qu'elles sont présentées : la taille présentée pour chaque tâche est la somme des tailles de ses paquets. Si vous choisissez deux tâches qui se partagent certains paquets, le véritable espace disque requis sera inférieur à la somme des tailles de deux tâches.

Une fois que vous avez ajouté tous les *logins* (root et personnel(s)), vous vous retrouvez dans le programme `dselect`. `dselect` vous permet de sélectionner les paquets à installer sur votre système. Si vous avez un cédérom ou un disque dur contenant les paquets additionnels de Debian que vous désirez installer sur votre système, ou bien si vous êtes connecté à Internet, cela vous facilitera la vie. Sinon, vous pouvez quitter `dselect` et le relancer plus tard après avoir ramener les paquets Debian sur votre système. Vous devez être super–utilisateur (root) quand vous voulez vous servir de `dselect`. Pour en savoir plus `dselect`, voir section suivante.

5.19 Installation de paquets avec `dselect`

Il est maintenant temps d'installer les paquets de logiciels de votre choix sur votre système Debian. Ceci est effectué via l'outil de gestion des paquets de Debian, `dselect`.

5.19.1 Introduction

Cette partie documente `dselect` pour les utilisateurs débutants. Elle n'a pas pour but de tout expliquer alors n'hésitez pas à vous servir des écrans d'aide pour en savoir plus.

`dselect` est utilisé pour sélectionner quel paquet vous désirez installer (il y a à l'heure actuelle près de 2250 paquets dans la Debian 2.1). Cela serait fait pour vous durant l'installation. C'est un outil très puissant et quelque fois un peu complexe. Ainsi, posséder quelques connaissances sur lui avant de l'utiliser est fortement recommandé. Une utilisation incongrue de `dselect` peut gravement endommager votre système.

`dselect` va vous accompagner durant le processus d'installation décrit ci–après :

– Choix de la méthode d'accès à utiliser
– Mise à jour de la liste des paquets disponibles, si possible
– Sélection des paquets que vous voulez sur votre système
– Installation et mis à jour des paquets désirés
– Configuration des paquets non configurés

– Enlèvement des paquets non désirés

À chaque fois qu'une étape est achevée avec succès, `dselect` vous emmène automatiquement à la suivante. Effectuez chaque étape dans l'ordre sans en omettre une.

Nous parlons ici et là dans ce document de lancer un autre *shell*. Linux a six sessions de contrôle ou shell disponibles en même temps. Vous pouvez circuler entre eux en pressant la combinaison *Alt–F1* jusqu'à *Alt–F6*. Il vous suffit alors de vous loguer dans un nouveau shell. La console utilisée lors de l'installation est la première, c'est–à–dire `tty1` alors presser *Alt–F1* si vous désirez y retourner.

5.19.2 Une fois que `dselect` est lancé

Une fois dans `dselect`, vous devriez avoir l'écran suivant :

```
Debian Linux 'dselect' package handling frontend.
0. [A]ccess Choose the access method to use.
1. [U]pdate Update list of available packages, if possible.
2. [S]elect Request which packages you want on your system.
3. [I]nstall Install and upgrade wanted packages.
4. [C]onfig Configure any packages that are unconfigured
5. [R]emove Remove unwanted software
6. [Q]uit Quit dselect
```

Détaillons un peu ces six étapes :

Access (Accès)

```
COPIE D'ÉCRAN DE dselect
```

L'écran d'accès est visible sur la figure 5.2.

C'est le lieu où l'on dit à `dselect` où sont les paquets. Ignorez l'ordre d'apparition à l'intérieur du menu. Il est très important de sélectionner la bonne méthode pour l'installation. Il se peut que vous les ayez dans un ordre différent, ou bien que vous en ayez plus ou moins ; ne vous inquiétez pas. Dans la suite, on décrit les différentes méthodes.

– multi_cd
  Grande et puissante, cette méthode complexe est celle recommandée pour installer une version récente de Debian à partir d'un jeu de cédéroms de binaires. Chacun de ces cédéroms devraient contenir des informations sur ses paquets ainsi que sur les paquets des cédéroms précédents (dans le fichier *Package.cd*). Lorsque vous utilisez cette méthode pour la première fois, assurez–vous que le cédérom n'est pas monté. Placez ensuite le *dernier cédérom de binaires* du jeu (les sources ne sont pas nécessaires) dans le lecteur et répondez aux questions qui vous sont posés :
  ```
  CD-ROM drive location[8]
  Confirmation that your are using a multi-cd set[9]
  ```

[8] emplacement du lecteur de CD-ROM

[9] Confirmation de l'utilisation de l'option multi-cd

```
The location of the Debian distribution on the disk(s)[10]
[ possibly ] the location(s) of the Packages file(s)[11].
```

Une fois que vous avez mis à jour la liste des paquets disponibles et que vous avez sélectionné les paquets à installer, la méthode *multi_cd* diverge d'une procédure normale. Il vous sera demandé de lancer une installation pour chaque cédérom que vous possédez, tour à tour. Malheureusement, en raison d'une limitation de `dselect`, il n'est pas possible de changer de cédérom à chaque étape ; la façon dont se déroule le travail pour chaque cédérom est le suivant :

- Insérez le cédérom dans votre lecteur.
- Sélectionnez *Install* à partir du menu principal de `dselect`.
- Attendez jusqu'à ce que `dpkg` en ait fini avec l'installation à partir du cédérom. (Cela peut indiquer une installation effectuée avec succès, ou de possibles erreurs. Ne vous en occupez pas pour le moment).
- Appuyez sur *Entrée* pour retourner à l'écran principal du menu `dselect`.
- Répétez le tout avec le cédérom suivant.

Il se peut qu'il soit nécessaire d'effectuer l'étape d'installation plus d'une fois pour respecter l'ordre d'installation des paquets. Certains paquets installés auparavant peuvent demander l'installation d'autres paquets pour se configurer proprement.

Lancer une étape de *Configure* est recommandée pour aider à débloquer les paquets qui le seraient à cette étape.

– multi_nfs, multi_mount
Ceci est proche de la méthode de *multi_cd* : elles apportent quelques raffinements supplémentaires sur le thème de la copie en changeant de support – par exemple, installation à partir de multi_cd exporté via NFS sur le lecteur de cédérom d'une autre machine.

– apt
Une des meilleures options pour installer à partir d'un miroir local des archives Debian ou à partir du réseau. Cette méthode utilise le système `apt` pour effectuer l'analyse complète des dépendances et de l'ordre. Il est donc recommandé d'installer les paquets dans le meilleur des ordres possibles.

La configuration de cette méthode est traitée plus loin. Vous pouvez choisir autant de sources différentes d'emplacements de paquets, en les mélangeant avec des fichiers : *URLs* (pour les disques locaux ou les disques montés par NFS) et *http :* ou *ftp* pour les accès sur Internet. Attention, les options *http* et *ftp* ne supportent pas l'authentification locale via un proxy.

Si vous avez un serveur de proxy soit pour le http soit pour le ftp (ou pour les deux), assurez–vous que les variables d'environnement `http_proxy` et `ftp_proxy` soient bien positionnés. Positionnez–les à partir du shell avant de lancer `dselect` en utilisant les commandes suivantes :

```
# export http_proxy=http://gateway:3128/
# dselect
```

Update (mise à jour)

`dselect` va lire les fichiers *Packages* ou *Packages.gz* à partir du miroir et créer une base de données sur votre système de tous les fichiers disponibles. Cela peut prendre un peu de temps, tant pour le téléchargement que pour l'exécution du traitement.

Select (Sélection)

[10] Emplacement de la distribution Debian sur le(s) disque(s)

[11] [peut-être] l'(es) emplacement(s) des paquets

Asseyez–vous sur votre chapeau : c'est ici que tout se passe. Le but de l'exercice est de sélectionner seulement le paquet que l'on désire installer.

Appuyez sur *Entrée*. Si vous avez une machine lente, faîtes très attention car l'écran va s'effacer et peut rester vierge pendant environ 15 secondes. Alors, n'essayez pas de taper sur des touches pendant ce temps–là.

La première chose qui apparaît à l'écran est la première page de l'aide. Vous pouvez obtenir l'aide à tout moment dans l'écrans *Select* en tapant sur « ? ». Vous pouvez dérouler l'aide par l'intermédiaire du «.».

Avant de plonger plus en avant, notons bien ceci :

– Pour sortir de l'écran de sélection (*Select*) après avoir tout sélectionné, il suffit de presser sur *Entrée*. Cela retourne à l'écran principal s'il n'y a pas de problème de sélection. Sinon, on vous demandera de régler le problème. Lorsque vous êtes content avec tous les écrans proposés, appuyez sur *Entrée* pour sortir.
– Les problèmes sont quelque chose de normal et sont même attendus. Si vous sélectionnez un paquet A qui demande un paquet B pour fonctionner, `dselect` vous avertira du problème et tentera de vous proposer la meilleure des solutions. Si le paquet A est en conflit avec le paquet B (c'est–à–dire qu'ils sont mutuellement exclusifs), on vous demandera de trancher et d'en choisir un.

Regardez les deux premières lignes de l'écran de `dselect`. Cette en–tête est un rappel de certains raccourcis claviers dressés dans le tableau suivant :

Touches	Description
+	Sélectionne le paquet à installer
=	Laisse le paquet en suspend
-	Retire le paquet
_	Retire le paquet, ainsi que ses fichiers de configuration
i, I	Affichage cyclique des informations
o, O	Affichage cyclique des options de tri
v, V	Passage en mode verbeux/sobre

Le tableau suivant affiche les états que `dselect` utilise pour marquer l'état de chaque paquet et son comportement.

Drapeau	État	Valeurs possibles
E	Erreur	Espace, R, I
I	État d'installation	Espace, *,-,U,C,I
O	Marqueur désuet	*, -, =, _, n
M	Marqueur	*, -, =, _, n

Plutôt que de tout décrire ici, je vous renvoie aux écrans d'aide qui sont bien expliqués. Un exemple quand même :

Vous entrez dans `dselect` et trouver une ligne du genre :

```
EIOM  Pri  Section   Package    Description
**    Opt  misc      loadlin    a loader (running under DOS) for LINUX
```

Ceci dit que `loadlin` a été sélectionné lorsque vous avez lancé `dselect` la dernière fois, et qu'il est toujours sélectionné mais non installé. Pourquoi donc ? Sûrement parce que le paquet `loadlin` n'est pas disponible physiquement. Il doit être absent de votre miroir.

L'information que `dselect` utilise pour obtenir tous les bons paquets installés se cache à l'intérieur des paquets eux–mêmes. Rien en ce bas monde n'est parfait, et il peut arriver que les dépendances d'un paquet soient erronées, ce qui veut simplement dire que `dselect` ne sera pas en mesure de résoudre la situation. Une solution est possible via l'utilisateur pour redonner le contrôle : cela prend la forme des commandes Q et X qui sont disponibles dans l'écran de sélection.

– Q
  surcharge. Force `dselect` à ignorer les dépendances et à faire ce que vous lui imposez. Les résultats, bien sûr, sont sous votre entière responsabilité.
– X
  Utilisez X si vous êtes complètement perdu. Cela remet les choses telles qu'elles étaient auparavant et sort.

Les raccourcis de l'écran de sélection de `dselect` qui vous aide à ne pas être perdu sont R, U et D.

– R
  Détruit toutes les sélections faîtes durant ce niveau. N'affecte pas les sélections faîtes aux niveaux précédents.
– U
  Si `dselect` a proposé des changements, et que vous les avez modifiés, U revient aux sélections de `dselect`.
– R
  Enlève les sélections de `dselect` en ne laissant que les vôtres.

Un exemple suit. Le paquet *boot–floppies* (pas un exemple de débutant, je sais, mais il a été choisi car il possède de nombreuses dépendances) dépend des paquets suivants :

- libc6-pi
- slang1-picn
- sysutils
- makedev
- newt0.25
- newt0.25-dev
- popt
- zlibig
- zlibig-dev
- recode

Le développeur maintenant le paquet *boot–floppies* pense que les paquets suivants devraient aussi être installés. Ils ne sont pas, en tout cas, indispensables :

```
- lynx
- debiandoc-sgml
- unzip
```

Lorsque vous sélectionnez *boot–floppies*, `dselect` vous envoie dans l'écran de résolution des conflits. Vous remarquerez que les paquets requis ont été sélectionnés.

Pressez alors sur la touche R pour remettre les choses à leur point de départ.

```
EIOM Pri  Section   Package      Description
  __ Opt  admin     boot-floppie Scripts to create the Debian
  __ Opt  devel     newt0.25-dev Developer's toolkit for newt
  __ Opt  devel     slang1-dev   The S-Lang programming library
  __ Opt  devel     slang1-pic   The S-Lang programming library
```

Si vous décidez maintenant que vous ne voulez plus du paquet *boot–floppies*, appuyez simplement sur *Entrée*.

Pressez alors sur la touche D remet les choses à la même place que lorsque vous les aviez sélectionnés la première fois :

```
EIOM Pri  Section   Package      Description
  _* Opt  admin     boot-floppie Scripts to create the Debian
  __ Opt  devel     newt0.25-dev Developer's toolkit for newt
  __ Opt  devel     slang1-dev   The S-Lang programming library
  __ Opt  devel     slang1-pic   The S-Lang programming library
```

Pressez alors sur la touche U revient les sélections initiales de `dselect` :

```
EIOM Pri  Section   Package      Description
  _* Opt  admin     boot-floppie Scripts to create the Debian
  _* Opt  devel     newt0.25-dev Developer's toolkit for newt
  _* Opt  devel     slang1-dev   The S-Lang programming library
  _* Opt  devel     slang1-pic   The S-Lang programming library
```

Je vous suggère de laisser les paramètres par défaut pour le moment : vous aurez amplement l'occasion de les modifier plus tard.

Quoique vous décidiez, appuyez sur la touche *Entrée* pour valider et retourner à l'écran principal. Si l'action résultante conduit toujours à des problèmes à résoudre, vous serez à nouveau renvoyé à un nouvel écran de résolution de conflit.

Les touches R, U et D sont très utiles dans les situations « qu'est–ce qui se passe si ? ». Vous pouvez expérimenter à volonté et ensuite revenir à la situation initiale. *Ne les regardez pas* comme une vitrine où il y aurait marqué « à ne briser qu'en cas de danger ».

Après avoir fait vos sélections dans l'écran de sélection, appuyez sur I pour obtenir une grande fenêtre, appuyez sur t pour aller au début et utilisez les touches *Page Up* et *Page Down*[12] pour vous déplacer rapidement parmi les sélections. Vous pouvez ainsi vérifier le résultat de votre travail et détecter d'éventuelles erreurs. Certaines personnes ont déselectionné des groupes entiers de paquets par erreur et ne s'en sont rendu compte que trop tard. `dselect` est un outil très puissant : ne le sous–utilisez pas...

Vous devriez avoir la situation suivante :

Package category	Status
Recquired	all selected
Important	all selected
Standard	mostly selected
Optional	mostly deselected
Extra	mostly deselected

Heureux ? Appuyez sur la touche *Entrée* pour sortir du processus de sélection. Vous pouvez y revenir et à nouveau sélectionnez si vous le désirez.

Install (Installation)

`dselect` travaille sur la totalité des paquets et installe ceux sélectionnés. Il vous pose des questions pour appliquer vos décisions. Il est souvent pratique de passer sur un autre shell pour comparer une ancienne configuration avec une nouvelle. Si l'ancien fichier est *conf.modules*, le nouveau sera *conf.modules.dpkg–dist*.

Le défilement à l'écran se passe parfois très rapidement sur les machines puissantes. Vous pouvez toujours le stopper et le redémarrer à l'aide des combinaisons respectives de touches *Ctrl–s* et *Ctrl–q*. À la fin de la tâche, vous aurez une liste des paquets non installés.

Il peut arriver qu'un paquet ne soit pas installé car il dépend d'un autre qui est décrit comme à installer mais qui ne l'est pas encore. La solution consiste à relancer *Install* une nouvelle fois. Il a été reporté des cas où il a été nécessaire de le relancer quatre fois avant que tout ne se mette en place correctement. Cela varie beaucoup suivant votre méthode d'acquisition.

Configure (Configuration)

La plupart des paquets sont configurés à l'étape précédente mais toute chose restée en suspens peut être achevée ici.

Remove (Suppression)

Suppression des paquets installés que l'on désire retirer.

Quit (Quitter)

Je suggère de lancer `/etc/cron.daily/find` à ce moment car vous avez de nombreux nouveaux fichiers sur votre système. Ensuite, vous pourrez utiliser `locate` pour obtenir l'emplacement de n'importe quel fichier.

[12] Écran haut et bas sur les claviers français : ce sont les touches de droite du pavé situé au–dessus des flèches de déplacement.

5.19.3 Quelques remarques pour conclure

Lorsque le processus d'installation lance `dselect` pour vous, vous devez être impatient de lancer Debian aussitôt que possible. Hé bien, préparez–vous à prendre une heure – ou plus – pour apprendre à vous en servir et à l'utiliser judicieusement. Lorsque vous entrez pour la première fois dans l'écran de sélection, n'en faîtes *aucune* – appuyez simplement sur la touche *Entrée* – et observez les problèmes de dépendances. Essayez de les résoudre. Si vous vous retrouvez une nouvelle fois dans l'écran principal, lancer *Select* une nouvelle fois.

Vous pouvez vous faire une idée de la taille d'un paquet en appuyant deux fois sur i et en regardant le champ *Size* (Taille). Ceci est la taille du paquet avec compression donc la décompression du paquet risque d'être nettement plus importante (voir le champ *Installed–Size* (Taille après installation) qui est en kilo–octets, pour la déterminer).

Installer un nouveau système Debian est une chose complexe mais `dselect` peut vous aidez à rendre la chose la plus aisée possible. Prenez donc bien le temps d'apprendre à vous en servir. Lisez les écrans d'aide et expérimentez les touches i, I, o et O. Utilisez la touche R. Ce sera tout pour ici mais cela devrait être suffisant pour vous en servir avec efficacité.

5.20 Glossaire

Les termes suivants sont largement utilisés dans ce livre et de façon générale lorsque l'on parle de Debian.

Paquet

C'est un fichier contenant tout le nécessaire à l'installation, à la désinstallation et à l'utilisation d'un programme particulier. Le logiciel qui gère les paquets est `dpkg`. `dselect` est une interface à `dpkg`. Les utilisateurs expérimentés utilisent souvent `dpkg` pour installer ou retirer un paquet.

Nom de paquet

Tous les noms de paquets sont de la forme *xxxxxxxxxx.deb*. Voici un exemple de noms de paquets simples :

- `efax_08a-1.deb`
- `lrzsz_0.12b-1.deb`
- `mgetty_0.99.2-6.deb`
- `minicom_1.75-1.deb`
- `term_2.3.5-5.deb`
- `uucp_1.06.1-2.deb`
- `uutraf_1.1-1.deb`
- `xringd_1.10-2.deb`
- `xtel_3.1-2.deb`

Chapitre 6

Se loguer

Votre système est maintenant installé ! Vous pouvez vous congratuler du bon travail effectué ! Il est temps maintenant de lancer le système pour l'utiliser. Dans ce chapitre, nous allons vous faire découvrir les lignes de commande Debian, quelques principes de sécurité et comment sortir du système. Dans les chapitres suivants, nous détaillerons plus ces sujets et nous vous ferons découvrir l'interface graphique de Debian, X11.

6.1 Les premiers pas

Après avoir quitter `dselect`, le *login* suivi du prompteur se présentent à vous. Vous pouvez maintenant vous loguer[1] en utilisant le *login* et le mot de passe que vous avez choisi. Votre système est maintenant prêt à être utilisé. Examinons ce que veut dire se loguer et comment cette procédure fonctionne.

Pour utiliser Debian, vous devez vous identifier personnellement au système. Cela lui permettra de savoir qui vous êtes, quels sont vos droits et quelles sont vos préférences.

Pour en arriver là, vous avez un *nom d'utilisateur* ou un *login*[2]. Si vous avez installé Debian vous–même, il vous a été demandé de fournir un tel nom durant l'installation. Si vous vous connectez à un système administré par quelqu'un d'autre, vous aurez à lui demander de vous fournir un compte sur le système ainsi que son nom.

Vous avez aussi un mot de passe et ainsi personne ne peut prétendre s'identifier à votre place. Si vous n'avez pas de mot de passe, tout le monde peut se connecter à votre place sur votre ordinateur depuis Internet et faire des tas de mauvaises choses. Si vous vous préoccupez un tant soit peu de sécurité, vous devez avoir un mot de passe.

De nombreuses personnes préfèrent croire que les autres ne feront rien de méchant sur leur compte. Il est raisonnable de croire que votre environnement de travail n'encourage pas la paranoïa. C'est une attitude parfaitement raisonnable. Cela dépend de vos priorités personnelles et de votre environnement. Il est certain qu'un système familial n'a pas les mêmes besoins en sécurité qu'une installation militaire. Debian vous permet aussi bien d'être sécurisé qu'insécurisé. Cela reste à votre convenance.

[1] J'ai délibérément francisé le terme anglais *to log in* tellement ce terme est devenu un standard en français

[2] Encore une fois, le terme anglais est tellement usité qu'il vaut mieux l'apprendre et l'utiliser

Lorsque vous démarrez Debian, vous apercevez un prompteur (*prompt*) : c'est l'ordinateur qui se met en attente d'informations. Dans ce cas, le prompteur est `login :`.

Vous devez taper votre nom d'utilisateur et, lorsque l'on vous le demande, votre mot de passe. Le mot de passe n'apparaît pas à l'écran lors de sa frappe. Appuyez sur la touche *Entrée* après avoir entré le nom d'utilisateur et après le mot de passe. Si vous trompez lors de la frappe de votre nom d'utilisateur ou lors de celle du mot de passe, vous aurez à recommencer.

Si vous effectuez l'opération correctement, vous verrez un message rapide et ensuite un $ suivi d'un prompteur. Le $ est écrit par un programme spécial appelé le *shell*[3]. Il est appelé le *prompteur du shell*. C'est l'endroit où vous envoyez les commandes au système.

Essayez d'entrer la commande `whoami` maintenant. Il y a un *curseur* à la droite du prompteur du shell. Votre curseur est un petit tiret ou rectangle qui indique où l'on va taper. Il doit se déplacer au cours de la frappe. Finissez toujours une commande shell par la touche *Entrée*.

`whoami`[4] vous renvoie votre nom d'utilisateur. Vous obtenez ensuite un nouveau prompteur de shell.

Dans le reste du livre, lorsque nous parlons d'entrer une commande, vous aurez à la taper après le prompteur du shell puis la valider avec la touche *Entrée*.

Lorsque vous avez fini de travailler, il se peut que vous désiriez quitter votre système. Pour sortir du shell, entrez la commande `exit`. Gardez bien à l'esprit que si vous restez logué, quelqu'un peut venir et utiliser votre compte. Par chance, vous pouvez faire confiance en vos collègues de travail ou aux personnes de chez vous ; mais si vous ne pouvez accorder votre confiance dans votre entourage, vous devrez être certain de vous être délogué avant votre départ.

6.2 Historique des commandes et édition de la ligne de commande

Tout ce que vous tapez après le prompteur du shell et avant de valider par *Entrée* est appelé une *ligne de commande*. C'est une ligne de texte qui demande à l'ordinateur de faire quelque chose. Le shell par défaut de Debian offre de nombreuses fonctionnalités pour rendre plus aisée l'entrée de la ligne de commande.

Vous pouvez revenir aux commandes précédentes pour les lancer à nouveau ou vous pouvez les modifier avec facilité et ensuite les relancer. Essayez ceci : entrez n'importe quelle commande, comme `whoami`. Ensuite, appuyez sur la flèche de direction verticale vers le haut. La commande `whoami` va réapparaître après le prompteur. Vous pouvez alors appuyer sur *Entrée* pour lancer une seconde fois la commande `whoami`.

Si vous avez entré de nombreuses commandes, vous pouvez appuyer sur la flèche plusieurs fois pour naviguer au travers des différentes commandes. Cette fonctionnalité est très pratique si vous répétez la même chose plusieurs fois de suite, ou si vous tapez mal une commande et que vous la rappeliez pour la corriger. Vous pouvez aussi utiliser la flèche verticale vers le bas pour circuler dans l'autre sens, vers les commandes les plus récentes. Lorsqu'il n'y a plus de commande à afficher, l'ordinateur envoie alors un signal sonore.

[3] ou *interpréteur de commandes* en français mais le terme *shell* est aussi utilisé que *log* ou *login*

[4] littéralement *Qui suis–je*

Vous pouvez aussi vous déplacer sur la ligne de commande pour effectuer des transformations. La façon la plus simple est d'utiliser les flèches de déplacements horizontaux gauche et droite. Essayez de taper `whoami` à la place de `whoami`. Utilisez alors la flèche de déplacement horizontal vers la gauche pour revenir en arrière vers le s. Vous pouvez effacer le s à l'aide de la touche *backspace*[5] ou *Suppr*.

Il y a bien d'autres fonctionnalités avancées mais il n'est pas nécessaire de les mémoriser maintenant. Essayez *Ctrl–a* pour vous retrouver directement au début de la ligne de commande. *Ctrl–k* (le k est pour *kill*, tuer en anglais) détruira tous les caractères situés à droite jusqu'à la fin de la ligne ; essayez–la à partir du milieu d'une ligne de commande. *Ctrl–y* renvoie la dernière chose que vous avez détruite en l'insérant à la position courante du curseur (y est pour *yank* en anglais). *Ctrl–e* déplace le curseur jusqu'à la fin de la ligne de commande.

Allez de l'avant et jouez avec l'édition de la ligne de commandes pour commencer à la maîtriser. Expérimentez.

6.3 Travailler en tant que Root

Puisque Debian est un système multi–utilisateurs, il est nécessaire d'en désigner un de particulier ou un programme pour être capable de tout faire sur le système. Le noyau n'autorise pas l'utilisateur normal à changer des fichiers importants du système. Cela signifie que les choses restent dans l'état où elles sont, à l'abris des accidents, des virus et même des tours de magie. À la différence de certains autres systèmes d'exploitation, Debian est à l'abris de ce genre de tracas. Il n'est pas nécessaire d'utiliser un programme d'anti–virus.

Parfois, il est quand même nécessaire de changer des fichiers importants. Par exemple, vous pouvez avoir envie d'installer un nouveau logiciel ou bien de configurer votre connexion réseau. Pour le faire, il vous faut des pouvoirs plus étendus que ceux d'un utilisateur normal. Vous devez devenir l'utilisateur *root* (parfois appelé aussi le *super–utilisateur*).

Pour devenir root, il vous suffit de vous loguer avec le nom d'utilisateur *root* et le mot de passe de root. Ceci a été fait durant l'installation : voir la section 5.15 pour plus de détails.

Dans de nombreux sites, seul l'administrateur système possède le mot de passe de root et seul l'administrateur système peut faire les tâches dévolues à root. Si vous utilisez votre propre ordinateur personnel, *vous* êtes l'administrateur système, bien–sûr. Si vous n'avez pas les privilèges du super–utilisateur, il vous sera nécessaire de passer la main à votre administrateur système pour les tâches qui requièrent les privilèges de root.

Parfois, vous aurez accès au mot de passe de root, souvent dans des organisations ou sur des serveurs d'écoles, car l'administrateur système vous fait confiance dans sa juste utilisation. Dans ce cas, il vous sera possible d'aider l'administrateur et de personnaliser le système pour vos besoins. Mais vous devrez toujours agir de façon responsable, en respectant tout le temps les autres utilisateurs.

Si vous possédez le mot de passe de root, essayez de vous loguer en tant que root maintenant. Entrer la commande `whoami` pour vérifier votre identité. Et ensuite, *déloguez–vous immédiatement*. Lorsque vous êtes root, le noyau ne vous protège pas de vous–même car root a les permissions de tout faire

[5] touche située au–dessus de la touche *Entrée* et comportant une flèche horizontal en direction arrière du défilement normal de l'écran, d'où son nom (*back* arrière et *space* espace)

sur tout le système. N'expérimentez pas en tant que root. En fait, ne faîtes rien sous root qui ne soit absolument nécessaire. Ce n'est pas tant un problème de sécurité qu'un problème de stabilité. Votre système s'en portera d'autant mieux qu'il est protégé de vos erreurs.

Il se peut que vous trouviez la commande `su` plus pratique pour se loguer en tant que root. `su` vous permet de prendre l'identité d'un autre utilisateur, généralement root à moins qu'elle ne soit suivie du nom de quelqu'un d'autre (Vous pouvez essayer de vous en souvenir en disant que `su` signifie *Super User* bien que certains disent que c'est pour *Set UserID* (positionne l'ID de l'utilisateur)).

Voici quelque chose à essayer. Loguez–vous en tant que simple utilisateur, c'est–à–dire, en n'étant pas root. Après, votre session devrait ressembler à quelque chose comme cela :

```
$ whoami                    Vérifie votre nom d'utilisateur
username                    Renvoie votre nom d'utilisateur
$ su                        Demande au système les droits super-utilisateurs
Password:                   Taper votre mot de passe root ici
machine:~# whoami
root                        Vous êtes maintenant root
machine:~# exit             Sort de votre shell root
$ exit                      Sort de votre shell "normal"
```

Lorsque vous faîtes des tâches d'administration système, vous devriez les faire autant que possible en tant que simple utilisateur. Ensuite, utilisez `su` pour la partie qui requiert les privilèges de root. Pour le faire, taper `su user` où `user` est l'utilisateur que vous voulez devenir. Vous devrez connaître le mot de passe de l'utilisateur en question, bien–sûr, à moins que vous ne soyez root à ce moment là ou bien que l'utilisateur n'ait pas de mot de passe.

6.4 Consoles virtuelles

Le noyau Linux supporte les consoles virtuelles. Ceci permet de faire de votre simple écran et clavier un système de type terminaux multiples connectés au même système. Heureusement, l'utilisation des consoles virtuelles est triviale sous Debian : il y a des touches dédiées pour aller d'une console à l'autre rapidement. Pour l'essayer, loguez dans votre système puis pressez sur *Alt–F2* (appuyez simultanément sur la touche *Alt* et sur *F2* qui est la touche de fonction numéro deux).

Vous devriez vous retrouver avec une demande de login. Ne paniquez pas. Vous êtes maintenant sur la console virtuelle (*VC* pour *virtual console* en anglais) numéro deux ! Loguez–vous et faîtes quelque chose – quelques commandes `whoami` ou n'importe quoi d'autre – pour vous assurez qu'il s'agit bien d'un nouveau shell. Maintenant vous pouvez retourner à la console numéro un en pressant *Alt–F1*. Ou bien vous pouvez aussi aller sur la console numéro trois avec *Alt–F3*.

Debian est fourni avec six consoles actives par défaut, qui sont accessibles avec la combinaison de la touche *Alt* et des touches de fonctions *F1* jusqu'à *F6*. (Techniquement, il y a plus de six consoles actives mais seules six sont accessibles pour vous loguer. Les autres sont réservées pour le système X Window ou d'autres tâches.)

Si vous utilisez le système X Window, il se lance généralement sur la première console virtuelle non utilisable – probablement la sept. Dès–lors, pour passer de la console virtuelle sous X à une des six

autres consoles, vous devez ajouter la touche *Ctrl* à la séquence de touches. Ainsi, il faut faire *Ctrl–Alt–F1* pour aller à la console virtuelle numéro un. Mais vous pouvez aller d'une console texte vers une console X en n'utilisant que la touche *Alt*. Si vous ne quitter jamais X, vous n'avez pas à vous inquiéter à ce sujet, X passera automatiquement sur sa console virtuelle au démarrage.

Une fois que vous les aurez maîtrisées, les consoles virtuelles deviendront certainement un outil indispensable pour faire plusieurs choses en même temps (Le système X Window offre la même fonctionnalité, au détail près qu'il s'agit de fenêtres multiples plutôt que de consoles). Vous pouvez lancer différents programmes dans chaque console virtuelle ou vous loguer en root dans une et en vous–même dans une autre. Ou bien chacun dans la famille n'utilise que sa console virtuelle. Ceci est particulièrement pratique si vous utilisez X car dans ce cas, vous pouvez lancer plusieurs sessions X en même temps dans différentes consoles.

6.5 Arrêter le système

N'éteignez pas simplement l'ordinateur ! Vous risquez de perdre des données de valeurs !

Si vous êtes le seul utilisateur de la machine, vous pouvez avoir envie de l'éteindre une fois votre travail terminé.

Pour éviter de fatiguer certains composants, il est recommandé de n'éteindre sa machine qu'une fois par jour, lorsque tout le travail est terminé. L'allumage et l'arrêt des ordinateurs sont les deux causes principales d'usure et de fatigue des composants internes de l'ordinateur. Allumer et fermer votre ordinateur une fois par jour est certainement le meilleur compromis entre votre facture d'électricité et la durée de vie de votre machine.

C'est une mauvaise chose que d'éteindre l'ordinateur en appuyant simplement sur le bouton d'alimentation du courant lorsque vous avez fini de travailler. C'est aussi une mauvaise idée que de rebooter la machine (avec le bouton *Reset*) sans avoir pris auparavant quelques précautions. Le noyau Linux, pour augmenter ses performances, utilise un *cache disque*. Cela signifie qu'il garde temporairement en mémoire certaines informations stockées sur le disque. Puisque la mémoire est des milliers de fois plus rapide qu'un disque dur, certaines opérations sur les fichiers sont ainsi beaucoup plus rapides. Périodiquement, ces informations en mémoire sont retranscrites sur le disque. Cette opération est appelée *syncing*. En vue d'éteindre ou de rebooter votre machine, il est nécessaire de lui dire de tout nettoyer en mémoire et de le placer sur l'unité de stockage permanent.

Pour rebooter, tapez simplement `reboot` ou appuyez sur *Alt–Ctrl–Suppr*.

Pour éteindre l'ordinateur il vous faut vous loguer en tant que root. Vous n'avez alors qu'à taper la commande `shutdown -h now`. Le système va alors dérouler la procédure d'extinction, y compris la commande `sync` qui nettoie le cache disque comme décrit précédemment. Lorsque vous apercevez *System halted*, il est possible d'éteindre sans danger l'ordinateur. Si vous avez un BIOS qui supporte l'APM (*Advanced Power Management*) et un noyau aussi, l'ordinateur peut se couper tout seul et ainsi vous épargner des problèmes. APM est très commun pour les ordinateurs portables et on le trouve aussi parfois dans certaines cartes mères d'ordinateurs de bureau.

Chapitre 7

Les bases

Il est maintenant temps d'explorer le système plus en détail. Vous avez vu comment se loguer et éteindre le système. Dans ce chapitre, nous allons explorer la ligne de commandes sous Linux, comment Linux gère fichiers et répertoires et quelques règles de base pour vous identifier des autres.

7.1 La ligne de commandes et les pages de `man`

Nous avons déjà discuté de la ligne de commandes – qui sont, rappelons–le, les commandes que vous tapez après le prompteur du shell. Cette partie décrit la structure de lignes de commandes beaucoup plus compliquées.

Une ligne de commandes minimale contient juste le nom d'une commande, comme `whoami`. Mais bien d'autres choses sont possibles. Par exemple, vous pouvez taper `man whoami`. Cette commande appelle l'aide en ligne de la commande `whoami` (Vous devrez vous déplacer à l'aide de la touche *espace* page par page et taper sur *q* pour sortir). Une commande plus compliquée est `man -k PostScript`. Cette ligne de commandes se compose de trois parties. Elle commence par le nom de commande `man`. Ensuite, on trouve une *option* ou *indicateur*, `-k`, suivi d'un *argument*, `PostScript`. Certains disent que tout sauf le nom de la commande est *paramètre* de la commande. Ainsi, les options et les arguments sont tous deux des paramètres.

Les options changent le comportement d'une commande, en activant une fonctionnalité particulière. Généralement, on trouve un « – » les précédant. Les utilitaires GNU ont aussi une « forme longue » pour les options ; la forme longue de `-k` est `--apropos`. Vous pouvez entrer `man -h` ou `man --help` pour obtenir la liste complète des options de la commande `man`. Chaque commande a sa propre liste d'options, bien qu'elle partage généralement les options `--help` et `--version`. Certaines commandes, comme `tar`, ne demande pas de « – » devant leurs options pour des raisons historiques.

Tout ce qui n'est pas une option et qui n'est pas le nom d'une commande est un *argument* (et dans le cas présent, `PostScript`). Les arguments ont des tas de finalités ; dans la plupart des cas, ce sont les noms de fichiers dont la commande doit s'occuper. Dans ce cas, `PostScript` est le mot que l'on fait chercher à la commande `man`. Dans le cas de `man whoami`, l'argument est le nom de la commande dont vous recherchez des informations.

Voici un résumé de la ligne de commande `man -k PostScript`

- `man`
  Le nom de la commande demande à l'ordinateur d'aller chercher les pages du manuel. Ceci est une documentation des commandes. Par exemple, `man whoami` ira vous chercher la documentation de la commande `whoami`.
- `-k`
  L'option, change le comportement de la commande `man`. Normalement, `man` s'attend à un nom de commande, tel `whoami`, comme argument et va chercher la documentation de cette commande. Mais avec l'option `-k` ou `--apropos`, il s'attend à ce que l'option soit un mot–clé. Il sort alors la liste de toutes les pages de manuel le comportant.
- `PostScript`
  est l'argument ; en raison de la présence de l'option `-k`, c'est le mot–clé à rechercher.

Allez–y et tapez la commande `man -k PostScript` et vous verrez apparaître une liste de pages de manuel de votre système qui ont un rapport avec le postscript. Si vous n'avez pas installé de nombreux logiciels, il se peut que vous voyiez apparaître à la place : `PostScript : nothing appropriate` (`PostScript : rien de trouvé`).

7.2 Description de la ligne de commande

Remarque : vous pouvez sauter cette partie si vous voulez avancer rapidement.

Il existe une façon concise et traditionnelle de décrire la *syntaxe* d'une commande. La *syntaxe* veut dire la façon correcte de combiner les options variées ainsi que les arguments. Par exemple, si vous tapez `man man` pour obtenir la page de manuel de `man`, vous apercevrez un certain nombre de descriptions de syntaxe débutant par le nom de commande `man`. Une d'entre elle devrait ressembler à ceci :

```
man -k [-M path] keyword ...
```

Tout ce qui se trouve entre crochet ([ ]) est optionnel. Dans ce cas, il n'est pas nécessaire d'utiliser l'option `-M` mais si vous le faîtes, il est alors nécessaire d'utiliser un argument de type `path`[1]. Vous devez utiliser l'option `-k` et l'argument `keyword` (mot–clé). Les `...` signifient qu'il peut y avoir encore beaucoup de choses à la suite, notamment plusieurs mots clés.

Regardons d'un peu plus près les descriptions complexes de la page de manuel de `man` :

```
man [-c|-w|-tZT device] [-adhu7V]
[-m system[,...]] [-L locale] [-p string]
[-M path] [=P pager] [-r prompt] [-S list]
[-e extension] [[section] page ...] ...
```

Il n'est pas nécessaire d'aller plus en avant (et ne vous inquiétez pas quant à sa signification), mais attirez votre attention sur l'organisation de la description.

Premièrement, ce rassemblement en grappe d'options veut généralement dire que vous pouvez en utiliser une ou plusieurs dans différentes combinaisons. Ainsi, `-adhu7V` veut tout simplement dire que

[1] C'est–à–dire le chemin d'accès en bon français.

vous pouvez aussi n'utiliser que `-h`. Bien sûr, vous pouvez utiliser toutes les combinaisons. Cette description ne rend pas forcément les choses très claires. Par exemple, `-h` est incompatible avec les autres options mais vous pouvez faire `man -du`. Malheureusement, cette description de format ne rend pas cette approche très claire.

Ensuite, le symbole | signifie « ou ». Ainsi, vous pouvez utiliser l'option `-c`, l'option `-w` *ou* l'option `-tZT`, suivi d'un argument `device`.

Enfin, remarquez bien que vous devez respecter les crochets car ils indiquent les unités optionnelles. Ainsi, si vous avez une `section`, vous devez aussi avoir une `page` parce que `page` n'est pas optionnelle à l'intérieur du groupe `[[section] page]`.

Il n'est pas nécessaire de mémoriser tout cela. Il suffit juste de revenir à cette partie pour lire la documentation.

7.3 Fichiers et répertoires

Les *fichiers* sont une facilité pour stocker et organiser l'information, analogue aux documents papier. Ils sont organisés en *répertoires* qui sont appelés *dossiers* sur certains autres systèmes. Regardons d'un peu plus près l'organisation des fichiers sur un système Debian :

– /
  Un simple / représente la répertoire racine. Tous les autres fichiers et répertoires sont contenus dans le répertoire racine. Si vous venez du monde DOS ou Windows, / est similaire à ce que le C : signifie pour le DOS qui est la racine du système de fichiers. Une différence importante entre DOS et Linux est que DOS garde toujours C : (pour le premier disque dur), A : (premier lecteur de disquettes) et D : (soit le lecteur de cédérom soit le second disque dur) alors que Linux organise tous ses fichiers autour de la même racine /.
– `/home/janeq`
  C'est le répertoire personnel de l'utilisateur *janeq*. En lisant de gauche à droite, pour aller dans ce répertoire, on commence dans le répertoire racine, puis dans le répertoire `home` et enfin dans celui de `janeq`.
– `/etc/X11/XF86Config`
  C'est le fichier de configuration du système X Window. Il réside dans le sous–répertoire `X11` du répertoire `/etc`. `/etc` est lui–même un sous–répertoire du répertoire racine /.

Choses importantes à retenir

– Les noms de fichiers sont sensibles à la casse utilisée. Cela veut dire que MONFICHIER et MonFichier sont différents.
– Le répertoire racine (*root* en anglais) se réfère uniquement à /. Ne confondez pas avec le root de l'utilisateur root, l'utilisateur du système avec les « supers pouvoirs ».
– Chaque répertoire a un nom, qui peut contenir n'importe quels lettres ou symboles *à l'exception de* /. Le répertoire racine est une exception ; son nom est / (prononcé « slash » ou « répertoire racine » ou « répertoire root ») et il ne peut être renommé.
– Bien que vous puissiez utiliser n'importe quels lettres ou symboles pour un nom de fichier, en pratique c'est une assez mauvaise idée. Il est souhaîtable d'éviter les caractères qui ont souvent une significa-

tion particulière dans les lignes de commandes comme `{ } ( ) [ ] ' ` " \ / > < | ; ! # & ^ * %`
- Évitez d'ajouter des espaces dans les noms de fichiers. Si vous désirez séparer les mots dans un nom, les meilleurs choix sont la période, l'hyphénation ou le tiret souligné « _ ». Vous pouvez aussi mettre en majuscule le début de chaque mot comme dans `CommeCeci`.
- Chaque fichier et répertoire sont désignés par un *nom de fichier pleinement qualifié*, un *nom de fichier absolu* ou un *chemin* qui donne la séquence des répertoires que l'on doit traverser pour atteindre le fichier. Les trois termes sont synonymes. Tous les noms de fichiers absolus commencent par le répertoire racine / et il y a un / devant chaque répertoire ou fichier du nom de fichier. Le premier / est le nom du répertoire mais les autres sont de simples séparateurs pour distinguer les parties du nom de fichier.
- Les noms utilisés portent parfois à confusion. Prenons l'exemple suivant :

  `/usr/share/keytables/us.map.gz`

  C'est un nom de fichier pleinement qualifié. Pourtant, certaines personnes le désignent comme un *chemin*. D'autres personnes s'en servent pour désigner le fichier `us.map.gz` comme un seul nom de fichier.
- Il y a aussi une autre utilisation du mot *chemin*[2]. Mais le contexte est toujours suffisamment clair pour faire la distinction.
- Les répertoires sont rangés sous forme d'arbre. Tous les noms de fichiers absolus commencent avec le répertoire racine. Le répertoire racine possède un certain nombre de branches, comme `/etc` et `/usr`. Ces sous–répertoires se subdivisent eux–mêmes en branches en de nombreux sous–répertoires, comme `/etc/init.d` et `/usr/local`. L'ensemble est appelé *l'arbre des répertoires*.
- Vous pouvez faire l'analogie avec un nom de fichier absolu et la route partante de la base de l'arbre (/) à la fin d'une branche (le fichier). Vous entendrez aussi parler certaines personnes de l'arbre des répertoires comme d'un *arbre familial*. Certains répertoires ont des « parents » et un chemin montre l'arbre généalogique complet.
- Il existe aussi des chemins relatifs qui commencent ailleurs que dans le répertoire racine. On y reviendra plus tard.
- Aucun répertoire ne correspond à un périphérique physique comme votre disque dur. Ceci diffère profondément du DOS et de Windows pour qui tout chemin commence par un nom de périphérique comme C :\. L'arbre des répertoires est une abstraction du périphérique matériel et vous pouvez donc l'utiliser sans savoir ce qu'il y a dessous. Tous vos fichiers peuvent être contenus dans un disque – ou vous pouvez très bien avoir vingt disques, certains connectés sur des ordinateurs différents répartis n'importe où sur le réseau. Vous ne pouvez deviner en jetant juste un coup d'oeil sur l'arbre des répertoires sur quel(s) périphérique(s) matériel(s) se trouvent vos fichiers.

Ne vous inquiétez si tout ceci n'est pas complètement clair pour vous : il y a de nombreux exemple à venir.

7.3.1 Utilisation des fichiers : un tutoriel

Pour utiliser votre système, vous devrez connaître comment créer, déplacer, renommer et détruire fichiers et répertoires. Cette partie a pour but de montrer comment le faire avec les commandes standard de Debian.

[2] *path* en anglais. À connaître aussi absolument.

La meilleure méthode d'apprentissage est encore d'essayer. Tant que vous n'êtes pas root (et vous n'avez pas maintenant à créer d'importants fichiers personnels), vous ne pouvez faire de catastrophe irrémédiable. Lancez–vous – tapez chacune de ces commandes après le prompteur et appuyer ensuite sur la touche *entrée* :

```
pwd
```

Le répertoire en cours est toujours considéré comme le *répertoire courant de travail* pour le shell que vous utilisez. Vous pouvez toujours le visualiser à l'aide de la commande `pwd` qui signifie *Print Working Directory* (affiche le répertoire de travail). `pwd` affiche donc le nom du répertoire dans lequel vous travaillez – et probablement `/home/votrenom`.

```
ls
```

`ls` signifie *list* (affiche) comme pour *list files* (affiche les fichiers). Lorsque vous tapez la commande `ls`, le système renvoie une liste de tous les fichiers de votre répertoire courant. Si vous venez d'installer Debian, votre répertoire de travail est certainement vide et `ls` ne renvoie rien en sortie car il n'y a pas de fichier à afficher.

```
cd /
```

`cd` signifie *change directory* (change de répertoire). Dans ce cas, on lui a demandé de changer pour le répertoire racine.

```
pwd
```

Ceci pour vérifier que vous êtes bien dans le répertoire racine.

```
ls
```

Pour jeter un coup d'oeil au contenu de /.

```
cd
```

Taper `cd` sans argument sélectionne votre répertoire personnel – `/home/votrenom` – comme répertoire courant. Essayez `pwd` pour vérifier.

Avant de continuer, il vous faut savoir qu'il y a vraiment deux sortes de fichiers. Certains d'entre eux commencent par /, le répertoire racine, comme `/etc/profile`. Ils sont appelés *noms de fichiers absolus* car ils se réfèrent au même fichier quelque soit le répertoire courant. L'autre sorte de fichiers sont les *fichiers relatifs*.

Seuls deux répertoires ne sont utilisés qu'avec des noms de fichiers relatifs : « . » et « .. ». Le répertoire . se réfère au répertoire courant et .. au répertoire parent. Ce sont des raccourcis. Ils existent dans *tous* les répertoires. Même le répertoire racine à un répertoire parent – c'est son propre parent !

Ainsi, tous les fichiers comprenant . ou .. sont *relatifs* car leur expression dépend du répertoire courant. Si je suis dans `/usr/bin` et que je tape `../etc`, je me réfère à `/usr/etc`. Si je suis dans `/var` et que je tape `../etc`, je me réfère à `/etc`. Remarquez bien que le nom de fichier sans le répertoire racine

possède implicitement à son commencement ./ . Ainsi, vous pouvez taper indifféremment `/local/bin` ou bien `./local/bin`, ce qui signifie exactement la même chose.

Un dernier petit truc : le *tilda* ~ est équivalent à votre répertoire personnel. Ainsi, taper `cd ~` revient au même que taper `cd` sans argument. Vous pouvez donc taper `cd ~/divers/monsousrepertoire` à la place de `cd /home/votrenom/divers/monsousrepertoire`. De façon analogue, `~myuser` est l'équivalent du répertoire personnel de l'utilisateur *myuser* qui est probablement quelque chose comme `/home/myuser`; ainsi, `~myuser/docs/debian.ps` est équivalent à `/home/myuser/doc/debian.ps`.

Il y a encore quelques commandes à essayer maintenant que vous maîtrisez les chemins relatifs. Tapez `cd` pour revenir dans votre répertoire personnel.

```
mkdir divers
```

Cela fabrique dans votre répertoire personnel un répertoire nommé `divers`. Vous allez utiliser ce répertoire pour essayer quelques autres commandes. Vous pouvez essayer `ls` pour vérifier que votre nouveau répertoire existe bien.

```
cd divers
```

Ceci change le répertoire vers `divers`.

```
mkdir monsousrepertoire
```

Ceci crée un sous–répertoire de `divers`.

```
cp /etc/profile .
```

`cp` est un raccourci pour *copy* (copier). `/etc/profile` est juste un fichier quelconque de votre système. Ne vous souciez pas de ce qu'il fait pour le moment. Nous l'avons recopié dans . (rappelez–vous que . signifie simplement « le répertoire dans lequel je me trouve maintenant » c'est–à–dire le répertoire courant de travail). Donc, ceci crée une copie de `/etc/profile` et la place dans votre répertoire `divers`. Essayez de taper `ls` pour vérifier qu'il y a bien un fichier appelé `profile` dans votre répertoire courant de travail au même niveau que le répertoire `monsousrepertoire`.

```
more profile
```

Ceci vous permet de visualiser le contenu du fichier `profile`. `more` est utilisé pour visualiser le contenu des fichiers textes. Il est appelé `more` car il ne montre qu'une page d'écran du fichier à la fois et il faut appuyer sur la barre d'espace pour visualiser la suite. `more` sortira de lui–même lorsqu'il aura atteint la fin du fichier ou lorsque vous aurez taper sur q (*quit* pour quitter).

```
more /etc/profile
```

va s'assurer que la copie est bien conforme à l'original.

```
mv profile monsousrepertoire
```

`mv` signifie *move* (déplacer). Vous avez déplacé le fichier `profile` du répertoire courant dans le sous–répertoire `monsousrepertoire` que vous avez créé précédemment.

```
ls
```

vérifie qu'il n'existe plus de fichier `profile` dans le répertoire courant.

```
ls monsousrepertoire
```

s'assure que `profile` a bien été déplacé dans le répertoire `monsousrepertoire`.

```
cd monsousrepertoire
```

Ceci change de répertoire vers le répertoire `monsousrepertoire`.

```
mv profile myprofile
```

Remarquez bien que contrairement à de nombreux autres systèmes, il n'y a pas de différence entre renommer un fichier et le déplacer. Ainsi, on ne trouve pas de commande `rename` (renomme). Signalons aussi que le second argument peut aussi bien être un répertoire pour déplacer le fichier dedans qu'un nom de fichier pour le renommage. `cp` fonctionne de la même façon.

Comme d'habitude, vous pouvez taper `ls` pour regarder le résultat.

```
mv myprofile ..
```

De la même façon que . signifie « le répertoire où je suis maintenant », .. signifie « le parent du répertoire courant ». Et dans le cas présent, le répertoire `divers` créé précédemment. Utilisez `ls` pour vérifier où se trouve `myprofile` maintenant.

```
cd ..
```

Change de répertoire vers le répertoire parent, c'est–à–dire ici `divers` où vous venez de déplacer `myprofile`.

```
rm myprofile
```

`rm` signifie *remove* (effacer) donc cela efface le fichier `myprofile`. Faîtes très attention ! Détruire un fichier sur un système GNU–Linux est *définitif* car il n'y a pas de fonction inverse. Si vous effacez un fichier, c'est fini *à jamais*...Prenez bien vos précautions.... On répète : détruire un fichier sur un système GNU–Linux est *définitif* car il n'y a pas de fonction inverse. Si vous effacez un fichier, c'est fini *à jamais*...Prenez bien vos précautions[3]....

```
rmdir monsousrepertoire
```

[3] Relisez–le encore une fois : détruire un fichier sur un système GNU–Linux est *définitif* car il n'y a pas de fonction inverse. Si vous effacez un fichier, c'est fini *à jamais*...Prenez bien vos précautions....

`rmdir` est la même chose que `rm` mais pour les répertoires. Remarquez que `rmdir` ne fonctionne qu'avec des répertoires vides. Si le répertoire contient des fichiers, vous devez les détruire en premier ou bien utiliser `rm -r` à la place de `rmdir`.

```
cd ..
```

Ceci vous permet de sortir du répertoire courant et de vous déplacer dans le répertoire parent. Maintenant vous pouvez taper la séquence suivante :

```
rmdir divers
```

Ceci détruira les derniers restes de notre essai.

Nous savons maintenant comment créer, copier, déplacer, renommer et détruire fichiers et répertoires. Nous avons aussi appris quelques raccourcis comme taper simplement `cd` pour revenir dans son répertoire personnel et comment . et .. désignent respectivement le répertoire courant et le répertoire parent. Vous devez aussi retenir les concepts de répertoire racine, ou /, et de l'alias ~ pour votre répertoire personnel.

7.3.2 Fichiers cachés et `ls -a`

Lorsque vous tapez `ls` les fichiers commençant par un . ne sont pas affichés. Traditionnellement, ces fichiers contiennent des informations de configurations, de préférences et toutes sortes de choses du même acabit. Ils vous sont cachés dans votre travail quotidien. Des exemples simples de fichiers cachés sont `~/.emacs`, `~/.newsrc`, `~/.bashrc`, `~/.xsession` et `~/.fvwmrc`. Ils sont respectivement utilisés par *Emacs*, le lecteur de niouses, le shell *Bash*, le système *X–Window* et le gestionnaire de fenêtres *fvwm*. Il est conventionnel d'ajouter un `rc` final aux fichiers cachés mais certains programmes ne le font pas. Il y a aussi des répertoires cachés, comme `~/.gimp` ou `~/.netscape` qui rassemblent les préférences pour *The Gimp* et *Netscape*.

Parfois, un fichier créera automatiquement un fichier caché ; par exemple, *Netscape* vous permets d'éditer dans une fenêtre graphique vos préférences et de sauvegarder vos choix. D'autres fois, vous devrez les créer vous–même à l'aide d'un éditeur de texte. C'est la façon traditionnelle de faire mais vous avez à apprendre le format particulier de chaque fichier, ce qui est un inconvénient au début mais vous donne par la suite beaucoup de puissance.

7.4 Processus

Nous avons déjà souligné le fait que GNU–Linux est un système multi–tâches. Il peut traiter plusieurs choses simultanément. Chacune de ces tâches est appelé *processus* (*process* en anglais). La meilleure façon de comprendre ce qui précède est de taper `top` après le prompteur du shell. Vous obtiendrez une liste des processus triés suivant le temps que votre ordinateur met à les traiter. Leur ordre change continuellement devant vos yeux. En haut de l'écran, vous devez apercevoir quelques informations sur le système : combien il y a d'utilisateurs connectés, combien il y a de processus, combien de mémoire vous avez et combien vous en utilisez réellement.

Dans la colonne à l'extrême gauche, vous apercevrez les utilisateurs propriétaires de chaque processus. À l'extrême droite, vous trouverez la liste des commandes évoquées. Vous avez certainement remarqué que `top` lui–même, lancé par vous, est presqu'en haut de la liste (car à chaque fois que `top` vérifie l'usage du microprocesseur (CPU), il l'active lui–même et utilise le CPU pour faire la vérification).

Remarquez que toutes les commandes se terminant pas « d » – comme `kflushd` ou `inetd` – sont des *daemons*.

Les *daemons*[4] signifient à l'origine *Disk And Extensions MOnitor* (surveillance des disques et des extensions). Un daemon est un processus non interactif ce qui veut dire qu'il est lancé par le système et que l'utilisateur n'a pas à s'en occuper. Les daemons servent à fournir des services comme la connexion Internet, l'impression ou le courrier électronique.

Maintenant, appuyez sur `u` et entrez votre nom d'utilisateur. La commande `u` sert à afficher les seules processus appartenant à un utilisateur ; cela permet d'ignorer tous les daemons et tous les processus appartenant à d'autres utilisateurs. Vous devez voir `bash`, le nom de votre shell. Vous aurez toujours le nom de votre shell.

La colonne numéro deux vous renvoie le PID pour *Process IDentication* (identificateur de processus). On assigne à chaque signal un unique PID. Vous pouvez utiliser les PID pour contrôler individuellement les processus (on approfondira le sujet ultérieurement). Un truc utile est la commande `?` qui vous renvoie une liste des commandes disponibles sous `top`.

Il se peut que vous vous posiez la question de la différence entre un processus et un programme. En pratique, on utilise indifféremment l'un pour l'autre. Techniquement parlant, un *programme* est le jeu d'instruction écrit par un programmeur et gardé sur le disque. Le *processus* est l'instanciation en mémoire de ce programme sous Linux. Mais il n'est pas primordial de faire une réelle différence.

Beaucoup de vos interventions avec un ordinateur vont consister à contrôler les processus. Vous allez devoir les faire partir, les arrêter et les observer. Votre outil privilégié pour cela est le shell.

7.5 Le shell

Le *shell*, ou *interpréteur de commandes*, vous permet d'interagir avec votre ordinateur. Il est appelé *shell* (coquillage) car il vous fournit un environnement de travail, comme une sorte de petite maison électronique pour vos travaux informatiques (pensez à l'abri de l'ermite).

La fonction la plus simple du shell est de lancer les autres programmes. Vous tapez le nom du programme que vous voulez lancer, avec à la suite les arguments que vous voulez et le shell demandera au système de lancer le programme pour vous.

Bien sûr, les systèmes graphiques avec fenêtres fournissent le même type de services. Techniquement, Windows 95 fournit un shell graphique et le système X Window fournit un autre type de shell graphique. Mais le terme *shell* est communément employé en sous–entendant *shell pour lignes de commandes*.

Cela va sans le dire mais les hackers travaillant sur un shell ne se contentent pas de lancer simplement des commandes. Votre shell regorge d'astuces cachées et de fonctionnalités puissantes si vous désirez vous en servir.

[4] *Daemons* en anglais se traduit par démons mais j'ai préféré garder le terme anglo–saxon car ce n'est pas un nom mais un acronyme.

Il y a de très nombreux shells différents disponibles. La plupart d'entre eux sont basés soit sur le *Bourne Shell* soit sur le *C shell*, deux des plus vieux shells. Le nom originel du Bourne Shell est `sh` et celui du C shell, `csh`. Les différentes variantes du Bourne Shell sont le *Bourne Again Shell* du projet GNU (`bash`, le shell par défaut de Debian), le *Korn Shell* (`ksh`) et le *Z shell* (`zsh`). Il y a aussi `ash`, une implémentation traditionnelle du Bourne shell. La plus connue des variantes du C shell est `tcsh` (le `t` est un tribu payé aux systèmes d'exploitation *TENEX and TOPS–20* qui ont inspiré quelques parties de `tcsh` en sus de `csh`).

`bash` est certainement le meilleur choix pour un nouvel utilisateur. C'est le shell par défaut et il comprend des tas de fonctionnalités que vous allez apprécier. Mais chaque shell possède ses avantages. Si vous voulez expérimenter, installez les différents paquets de shell et changez votre shell avec la commande `chsh`. Tapez simplement `chsh`, fournissez votre mot de passe et lorsque demandé, choisissez votre nouveau shell. Lorsque vous vous reloguerez de nouveau, vous utiliserez le nouveau shell.

7.6 Gérer les processus avec `bash`

Debian est un système multi–tâches donc vous avez besoin d'un moyen de faire plus d'une chose à la fois. Les environnements graphiques, comme X, fournissent un élément naturel pour le faire : ils autorisent plusieurs fenêtres simultanément. Naturellement, `bash`, et tous les autres shells, fournissent des facilités similaires.

Récemment, vous avez utilisé `top` pour jeter un coup d'oeil sur tous les processus de votre système. Votre shell vous propose des façons aisées de suivre les commandes que vous avez lancées à partir de ce shell. Chaque ligne de commande lance un *job* (aussi appelé un *groupe de processus*) qui sera suivi par le shell. Un job peut être simplement un processus ou bien un ensemble de processus tubés (plus sur les tubes un peu plus tard).

Entrer une ligne de commande lance un job. Essayez de taper `man cp` et la page de manuel de `cp` va apparaître à l'écran. Le shell va tourner en arrière–plan et revenir en avant–plan une fois la lecture de la page terminée (ou vous pouvez taper sur q pour éviter d'avoir à dérouler toute la page pour sortir).

Mais admettons que vous désiriez faire quelque chose d'autre pendant la lecture du manuel. Pas de problème. Appuyez sur *Ctrl–z* pour suspendre la lecture du manuel (mis en arrière–plan du job courant) et ainsi remettre en avant–plan le shell. Lorsque vous suspendez un job, `bash` vous donne immédiatement quelques informations à son sujet et ensuite il renvoie un prompteur. Vous devriez apercevoir quelque chose comme cela à l'écran :

```
NAME  cp  - copy files SYNOPSIS cp [options] source
--More--
[1]+ Stopped man cp
$
```

Observez bien les deux dernières lignes. Les deux dernières comportent l'information sur le job et vous avez le prompteur du shell.

`bash` assigne à chaque ligne de commande un *numéro de job* donné par le shell. Cela vous permet de vous référer facilement à un processus. Dans ce cas, `man cp` porte le numéro 1, indiqué par `[1]`. Le

signe + veut dire que c'est le dernier job à avoir été mis en arrière–plan. `bash` vous renvoie aussi l'état courant du job – `Stopped` – ainsi que la ligne de commande.

Il y a de nombreuses choses à faire avec les jobs. Avec `man cp` toujours suspendu, essayez la commande suivante :

```
man ls
```

Cela crée un nouveau job.

```
Ctrl-z
```

Cela le suspend ; vous devriez voir les informations du job.

```
man mv
```

Lance un nouveau job.

```
Ctrl-z
```

Le suspend aussi.

```
jobs
```

Demande au `bash` d'afficher l'état de tous les jobs en cours. Le résultats devraient ressembler à :

```
{$} jobs
[1] Stopped man cp
[2]- Stopped man ls
[3]+ Stopped man mv
{$}
```

Remarquez bien que les – et + qualifient respectivement l'avant–dernier et le dernier job en arrière–plan.

```
fg
```

Cela place le dernier job en arrière–plan (`man mv`, celui avec le +) de nouveau en avant–plan. Si vous pressez la barre d'espace, la page de manuel va se dérouler.

```
Ctrl-z
```

Suspend à nouveau `man mv`.

```
fg %1
```

Vous pouvez vous référer à n'importe quel job en utilisant % suivi de son numéro. Si vous utilisez `fg` sans lui spécifier de job, c'est le dernier actif qui sera réveillé.

```
Ctrl-z
```

Suspend à nouveau `man cp`.

```
kill %1
```

Détruit le job numéro 1. `bash` fait un rapport d'information du job qui devrait ressembler à :

```
$ kill %1
[1] - Terminated man cp
$
```

`bash` se contente de demander au job de se terminer et parfois ce dernier ne veut pas. Dans ce cas, vous devez ajouter l'option `-KILL`[5] pour arrêter la demande de destruction et commencer la destruction. Par exemple :

```
$ kill -KILL %1
[1]- Killed man mv
$
```

L'option `-KILL` force une destruction inconditionnelle du job.

En des termes plus techniques, `kill` se contente d'envoyer un signal. Par défaut, il envoie un signal de terminaison (`TERM` ou `signal 15`) mais vous pouvez très bien spécifier un signal et l'option `-KILL` (`signal 9`) est le signal qui force la terminaison. La commande `kill` n'est pas forcément appropriée pour l'envoi de signaux ; par exemple, envoyer le signal `TSTP` (`terminal stop`) suspend le processus et permet de le récupérer plus tard.

```
top
```

Cela nous renvoie à l'écran la sortie de `top`. Envoyez la commande `u` dans `top` pour ne visualiser que vos processus. Regardez dans la colonne de droite les commandes `man ls` et `man mv`. `man cp` n'est plus là puisque vous l'avez tuée. `top` vous montre les processus vous appartenant ; remarquez bien que le PID à gauche de l'écran ne correspond pas au numéro de job.

Il se peut que vous ne voyez pas vos processus car trop bas pour apparaître à l'écran. Si vous utilisez X (voir chapitre 11), vous pouvez redimensionner votre `xterm` pour résoudre le problème.

Même ces simples jobs sont en réalité des processus multiples, incluant le processus `man` et le visualiseur `more` qui vous permet de faire défiler les pages une par une. Vous devez apercevoir les processus `more` dans `top`.

Vous vous demandez peut–être comment nettoyer les deux jobs restant. Vous pouvez soit les tuer (avec la commande `kill`) ou bien les remettre en avant–plan (avec `fg`) puis les quitter. Rappelez–vous que la commande `jobs` vous renvoie la liste des jobs en cours et leur état.

Une touche finale : la documentation de `bash` est plutôt bonne mais il vaut mieux aller la chercher dans le système d'aide `info` que dans les pages de manuel. Pour la lire, tapez `info bash`. Voir la partie

[5]. De nombreuses personnes utilisent le signal –9 plutôt que le nom du signal. Dans tous les cas, c'est techniquement moins portable que d'utiliser le nom du signal

16.2 pour les instructions pour se servir de la commande `info`. `bash` contient aussi une très bonne documentation en ligne accessible via la commande `help`. `help` renvoie une liste de sujets disponibles. On peut accéder directement à l'information en tapant `help sujet en question`. Essayez `help cd` par exemple. Cela vous fournira des détails sur les arguments `-L` et `-P` reconnus par `cd`.

7.7 Quelques fonctionnalités de `bash`

Cette partie mentionne rapidement quelques fonctionnalités couramment utilisées sous `bash`. Pour une discussion plus complète, reportez–vous au chapitre 8.

7.7.1 Le complètement par tabulation

Le shell `bash` est capable de deviner quel fichier ou commande vous êtes en train de taper et les compléter automatiquement pour vous. Il suffit simplement de taper le début de la commande ou du fichier et d'appuyer sur la touche de tabulation. Si `bash` trouve un complètement unique, il finira le mot et ajoutera un espace à la fin. S'il trouve de multiples complètements[6], il propose l'ensemble de ses choix possibles en émettant un signal sonore. Vous pouvez alors entrer le nombre de lettres suffisantes pour le rendre unique et appuyez sur la touche de tabulation à nouveau. S'il n'existe pas de complètement, `bash` émet simplement un signal sonore.

7.8 Protéger votre identité

Les systèmes de type Unix sont multi–utilisateurs et vous avez donc votre propre identité comme utilisateur sur le système. Tapez `finger votrenomdutilisateur` pour voir les informations qui sont publiquement disponibles. Pour changer le nom et le shell affichés ici, vous pouvez utiliser la commande `chfn` et `chsh`. Seul root peut vous changer votre login et votre répertoire d'attache. Vous devez avoir vu *No plan*. Un *plan* est une simple information que vous voulez rendre publique. Pour créer un plan, vous placer toute l'information que vous voulez montrer aux autres dans un fichier appelé `.plan`. Pour le faire, vous devrez vous servir d'un éditeur de texte (voir chapitre 10.2). Ensuite, utilisez à nouveau `finger` pour observer le résultat. Les autres peuvent utiliser finger sur vous pour voir votre *plan* ou vérifier si vous avez reçu ou lu votre courrier électronique.

Remarquez bien que l'information de `finger` est disponible à travers tout l'Internet par défaut. Si vous ne le voulez pas, lisez la documentation de `inetd` et le fichier `/etc/services`. Éventuellement, la page de manuel peut décrire la mise en oeuvre mais il n'est pas évident que cela puisse réellement vous aider.

[6] On trouve parfois *complétion* qui n'est qu'un barbarisme engendré à partir du terme anglais *completion*.

Chapitre 8

Utiliser le shell

En lisant les chapitres précédents, vous avez déjà interagit avec le shell. Le shell est le programme qui reçoit vos commandes et qui les exécute. Dans ce chapitre, nous allons explorer le shell plus en avant, avec une mention spéciale pour le personnaliser afin de lui faire faire le travail que vous voulez qu'il fasse.

8.1 Les variables d'environnement

Chaque processus a un *environnement* associé. Un environnement est ensembles de *variables d'environnement*. Une variable est une valeur modifiable avec un nom déterminé. Par exemple, le nom *EMAIL* peut se référer à la valeur `joe@nowwhere.com`. Sa valeur peut changer ; *EMAIL* peut aussi se référer à `jane@somewhere.com`.

Puisque le shell n'est qu'un processus parmi d'autres, il possède aussi son environnement. Vous pouvez visualiser votre environnement en entrant la commande `printenv`. La figure suivante donne un exemple de résultat de sortie de cette commande. Sur votre système, la sortie sera différente bien que similaire.

```
PAGER=less
HOSTNAME=icon
MAILCHECK=60
PS1=$
USER=username
MACHTYPE=i486-pc-linux-gnu
EDITOR=emacs
DISPLAY=:0.0
LOGNAME=username
SHELL=/bin/bash
OSTYPE=linux-gnu
HISTSIZE=150
TERM=xterm-debian
```

```
TEXEDIT=jed
PATH=/usr/sbin:/usr/sbin/:/usr/local/bin/:
/usr/bin/:/bin/:/usr/bin/X11/:/usr/games
_=/usr/bin/printenv
```

Les variables d'environnement sont une façon de configurer le système. Par exemple, la variable *EDITOR* vous permet de sélectionner votre éditeur préféré pour poster des *news*, écrire du courrier électronique et ainsi de suite.

Régler ses variables d'environnement est quelque chose de triviale. En pratique, commençons par personnaliser votre prompteur de shell ainsi que votre visualiseur de texte avec les variables d'environnement. Tout d'abord, commençons par un peu d'informations basiques.

```
man less
```

Cette commande vous permet d'accéder au manuel en ligne de la commande `less`. Pour éviter de dérouler tout le texte d'un seul coup à l'écran, `man` invoque un *pageur* qui vous déroule le texte page par page, à chaque fois que vous appuyez sur la touche *espace*. Par défaut, le pageur utilisé est `more`.

Continuons et regardons un peu la page de manuel de `less`, qui est un pageur très puissant. Affichez la page suivante en appuyant sur espace et q pour quitter. `more` quitte aussi automatiquement la page en arrivant à la fin.

```
export PAGER=less
```

Après avoir lu les avantages de `less`, il se peut que vous vouliez l'utiliser pour lire les pages de manuel. Pour ce faire, vous devez positionner la variable d'environnement *PAGER*.

La commande pour ajuster une variable d'environnement dans `bash` a toujours le format suivant :

```
export NOM=valeur
```

`export` signifie déplacer la variable du shell dans votre environnement. Cela veut dire que les programmes autres que le shell (par exemple, un visualiseur de textes) seront capables d'y accéder.

```
echo $PAGER
```

C'est la façon la plus aisée d'accéder au contenu de la variable. *$PAGER* demande en fait au shell d'insérer la valeur de *PAGER avant* d'invoquer la commande. `echo` renvoie simplement l'argument : dans ce cas, cela renvoie simplement la valeur du contenu de *PAGER*, c'est–à–dire `less`.

```
man more
```

renvoie maintenant la page de manuel de `more`. Mais maintenant, `man` utilise `less` comme pageur.

`less` a beaucoup plus de fonctionnalités que `more`. Par exemple, vous pouvez paginer en arrière à l'aide de la touche `b`. Vous pouvez aussi vous déplacer en avant et en arrière à l'aide des flèches de direction. `less` ne sort pas lorsque vous avez atteint le bas du fichier de la page de manuel : il attend que vous appuyez sur la touche `q`.

Vous pouvez essayer quelques spécificités de la commande `less`, comme `b`, pour vérifier qu'elles ne fonctionnent pas avec `more`.

```
unset PAGER
```

Si vous ne désirez pas spécifier un pageur, vous pouvez appliquer la commande `unset` à la variable. `man` utilisera alors `more` par défaut, comme il le faisait avant que vous ne changiez la variable.

```
echo $PAGER
```

Comme la variable *PAGER* a été détruite, `echo` ne renverra rien.

```
PS1=bonjour:
```

Pour s'amuser, tentons de modifier le prompteur. $ devrait maintenant être modifié. Voici les étapes en détails :

```
$ echo $PS1
$
$PS1=bonjour:
bonjour:echo Mon prompteur est $PS1
Mon prompteur est bonjour:
bonjour:
```

`export` n'est pas nécessaire car vous ne modifier que le comportement de votre propre shell. Il n'y a pas de raison d'exporter dans l'environnement de tous les programmes pour faire le test. Techniquement parlant, *PS1* est plutôt une *variable de shell* qu'une variable d'environnement.

Si vous le voulez, vous pouvez exporter la variable de shell, la transformant ainsi en variable d'environnement. Si vous agissez de la sorte, les programmes lancés à partir du shell pourront la voir.

8.2 Où les commandes résident : la variable *PATH*

Lorsque vous tapez une commande dans votre shell, ce dernier doit trouver le programme correspondant sur votre disque avant de l'exécuter. Si le shell devait examiner l'ensemble du disque dur, cela serait extrêmement lent. À la place, il se contente d'aller chercher dans un ensemble de répertoires contenu dans la variable d'environnement *PATH*. Cette liste de répertoires est le *chemin d'accès* du shell. Lorsque vous entrez une commande, le shell va chercher dans chacun des répertoires de ce chemin pour trouver la commande que vous lui avez demandé de lancer.

Il se peut que vous ayez à changer la variable *PATH* si vous installez vous–même des programmes à des endroits non conventionnels. La valeur du *PATH* est une liste de répertoires séparés par des « : ».

```
/usr/local/bin:/usr/bin/:/bin:/usr/bin/X11:/usr/games
```

Cette valeur est définie dans le fichier `/etc/profile` et applicable à tous les utilisateurs. Il est très facile de modifier cette valeur, comme d'ailleurs l'ensemble des variables d'environnement. Si vous tapez la commande `ls`, le shell va d'abord la chercher dans `/usr/local/bin`; `ls` ne s'y trouve pas donc il va continuer et la chercher dans `/usr/bin/`; comme cela échoue encore, il vérifie `/bin`.

Il la trouve enfin et stoppe sa recherche pour exécuter le programme `/bin/ls`. Si la commande `/usr/bin/X11/ls` existe (ce n'est pas le cas mais c'est pour l'exemple), elle serait tout simplement ignorée.

Vous pouvez voir quel `ls` le shell va utiliser grâce à la commande `type`. `type ls` va vous renvoyer `/bin/ls`. Essayez–la vous–même.

```
$ type type
type is a shell builtin[1]
```

`type` n'est pas vraiment un programme : c'est une fonctionnalité du shell mais vous pouvez l'utiliser comme un programme extérieur.

Il y a de nombreuses commandes de la même sorte. Tapez `man builtins` pour visualiser la page les décrivant. De façon générale, il n'est pas nécessaire de savoir si une commande est une fonctionnalité interne ou bien un programme extérieur. Mais il faut garder à l'esprit qu'une fonctionnalité interne ne sera pas visible via un `ps` ou un `top` car il ne s'agit pas d'un processus séparé : c'est une partie intégrante du shell.

8.3 Fichiers de configuration

De nombreuses applications sous Linux vous permettent de modifier leur comportement à certains moment en transformant les fichiers contenant leurs informations de configuration. Ces fichiers de configuration peuvent contenir des informations de lancement, des réglages de configuration ou bien des ajustements pour la sortie. En général, un nom de fichier de configuration est basé sur le nom de l'application qu'il est sensé modifier. Cette convention de nom permet de déterminer rapidement à quelle application appartient quel fichier de configuration pour les réglages.

8.3.1 Configuration système contre configuration personnelle

Il est important de se rappeler qu'il y a deux façons différentes de configurer sur un système Debian. La *configuration système* affecte tous les utilisateurs. Ces configurations se trouve dans le répertoire `/etc` et donc il est généralement nécessaire d'être root pour modifier ces fichiers. Vous pouvez avoir besoin de configurer la façon dont se connecte le système à Internet par exemple, ou bien configurer les butineurs HTML pour leur faire afficher la page de la société en se lançant. Si vous désirez effectuer le changement pour tous les utilisateurs à la fois, vous devez faire les transformations dans `/etc`. Des fichiers de configuration simples de `/etc` sont `/etc/X11/XF86Config`, `/etc/lynx.cfg` et `/etc/ppp/options`. En fait, presque tous les fichiers de `/etc` sont des fichiers de configuration.

La *configuration personnelle* n'affecte qu'un simple utilisateur. Les fichiers cachés sont utilisés à cette fin. Par exemple, le fichier `~/.newrc` contient une liste d'articles sur USENET (groupes de discussions) que vous avez lus et à quels groupes vous avez souscrit. Cela permet à des lecteurs de news comme `trn` ou `GNUS` de n'afficher que les articles non lus dans les groupes qui vous intéressent. Cette information

[1] type est une fonction interne du shell

est bien–sûr propre à chaque utilisateur du système et ainsi chacun possède son propre `.newrc` dans son répertoire de travail personnel.

8.4 Aliases

Si vous utilisez la même commande de nombreuses fois, il se peut que vous vous fatiguiez à la taper. `bash` vous permet d'écrire un raccourci – on dit *alias* – de votre commande.

Admettons que vous désiriez toujours utiliser les options `--almost-all` et `--color=auto` de `ls`. Faîtes un alias :

```
alias myls='ls --almost-all --color=auto'
```

Maintenant, il vous suffit de taper `myls` à la place de la commande complète. Pour voir ce qu'est réellement `myls`, il suffit de taper `type myls`. Pour voir la liste complète des *aliases* que vous avez défini, taper simplement `alias` tout seul comme commande.

8.5 Contrôler les entrées et les sorties

Au fur et à mesure de vos expériences avec Linux, vous allez comprendre combien manipuler les entrées et sorties des applications est un concept très puissant. Cette partie décrit quelques points de ce que peut vous apportez la redirection des entrées–sorties.

8.5.1 `stdin`, `stdout`, tubes et redirection

Chaque processus a au moins trois connexions avec le monde extérieur. L'*entrée standard* est une des sources des données du processus. La *sortie standard* est un des endroits où le processus envoie ses données et l'*erreur standard* est l'endroit où le processus envoie ses messages d'erreurs (on les abrège généralement par `stdin`, `stdout` et `stderr`).

Les mots « source » et « endroit » sont intentionnellement vagues. Ces lieux d'entrées–sorties standards peuvent changer avec les utilisateurs. Cela peut être l'écran, le clavier, un fichier et même une connexion réseau. Vous pouvez spécifier les endroits à utiliser.

Lorsque vous lancez un programme à partir du shell, l'entrée standard est généralement le clavier et la sortie standard, ainsi que les erreurs, s'affichent sur votre écran. Vous pouvez bien sûr demander à votre shell de modifier ce comportement.

Par exemple, la commande `echo` envoie son entrée standard sur sa sortie standard, normalement à l'écran. Mais vous pouvez décider de l'envoyer plutôt dans un fichier à l'aide de *l'opérateur de redirection de sortie*, $>$. Par exemple, pour écrire le mot « Bonjour » dans le fichier `monfichier`, utilisez la commande suivante :

```
echo Bonjour > monfichier
```

Utilisez ensuite `cat` ou bien votre pageur (`more` ou `less`) pour visualiser le contenu du fichier `monfichier`.

```
$ echo Bonjour > monfichier
$ cat monfichier
Bonjour
$
```

Vous pouvez changer l'entrée standard d'une commande avec *l'opérateur de redirection d'entrée*, `<` . Par exemple, `cat < monfichier`. En pratique, ce n'est pas commode. Pour simplifier, `cat` accepte un fichier pour argument. Ainsi, vous pouvez vous contenter de taper `cat monfichier` et l'effet sera identique.

Si l'on approfondit un peu, la commande `cat <monfichier` signifie que le shell ouvre le fichier `monfichier` et envoie son contenu dans l'entrée standard de `cat`. `cat monfichier`, sans l'opérateur de redirection, signifie que la commande `cat` reçoit un argument unique (`monfichier`), l'ouvre d'elle–même et ensuite l'affiche à l'écran.

Il y a bien–sûr une raison pour cette double fonctionnalité. Par exemple, il est possible de connecter la sortie standard d'une commande sur l'entrée standard d'une autre commande. Cela s'appelle un *tube* (*pipeline* en anglais). Cela utilise l'*opérateur tube*[2], `|` .

Peut–être désirez–vous voir la *GNU General Public License* à l'envers ? Pour le faire, utilisez la commande `tac` (c'est un `cat` mais inversé). Essayez :

```
tac /usr/doc/copyright/GPL
```

Malheureusement, cela défile beaucoup trop rapidement pour être lisible. Donc vous ne voulez avoir que quelques paragraphes : la solution est un tube :

```
tac /usr/doc/copyright/GPL | less
```

Cela prend donc la sortie standard de `tac`, qui est la GPL lue à l'envers, et l'envoie dans l'entrée standard de `less`.

Vous pouvez enchaîner ainsi autant de commandes que vous le désirez. S'il vous prend une envie inexpliquable de transformer tous les G en Q, il vous suffit d'utiliser la commande `tr G Q` comme ceci :

```
tac /usr/doc/copyright/GPL | tr G Q | less
```

Vous pouvez obtenir le même résultat en utilisant des fichiers temporaires et des redirections, par exemple :

```
tac /usr/doc/copyright/GPL > tmpfile
tr G Q < tmpfile > tmpfile2
less < tmpfile2
rm tmpfile tmpfile2
```

Il est clair qu'un tube est bien plus pratique.

[2]Cela dépend de votre clavier : cela peut apparaître comme une barre verticale ou une barre verticale en deux parties mais cela doit toujours se trouver au–dessus du contre *slash* (NDT : pour les claviers *qwerty* uniquement...tapez `Alt Gr 6` pour les *azerty)*

8.6 Expansion de nom de fichier

Vous désirez souvent utiliser une commande sur tout un groupe de fichiers. Des caractères spéciaux (*wildcards*) sont utilisés pour créer des *termes génériques d'expansion de nom de fichiers*. Une série de lettres et des caractères spéciaux permettent ainsi de lister des fichiers. Par exemple, le motif `/etc/*` remplace la liste de tous[3] les fichiers de `/etc`.

* est un motif qui remplace n'importe quelle série de caractères et ainsi l'expression `/etc/*` fournira la liste de tous les fichiers commençant par `/etc/`.

Cette liste de fichiers est très utile comme un jeu d'arguments pour une commande. Par exemple, le répertoire `/etc` contient une série de sous–répertoires appelés `rc0.d, rc1.d`, etc...Normalement, pour visualiser le contenu de ces répertoires, vous devriez taper les commandes suivantes :

```
ls /etc/rc0.d /etc/rc1.d /etc/rc2.d /etc/rc3.d
ls /etc/rc4.d /etc/rc5.d /etc/rc6.d /ect/rcS.d
```

C'est particulièrement lourd et pénible. À la place, il suffit d'utiliser le caractère spécial « ? » comme suit :

```
ls /etc/rc?.d
```

`/etc/rc?.d` renvoie alors une liste de nom de fichiers, commençant par `rc`, suivant d'un caractère et se terminant pas `.d`.

Les caractères spéciaux disponibles sont les suivants :

– *
  Correspond à n'importe quel groupe de 0 ou plus de caractères.
– ?
  Correspond à un caractère quelconque mais unique.
– `[...]`
  Si vous entourez des caractères de crochets, le résultat est une correspondance avec ces caractères. Par exemple, `[abc]` correspond soit à un `a`, soit à un `b` ou soit à un `c`. Si vous ajouter un ^ après le premier crochet, on retourne le sens. Ainsi, `[^abc]` correspond à tous les caractères qui ne sont pas un `a`, un `b` ou un `c`. Vous pouvez aussi inclure un intervalle comme `[a-j]` qui correspond à tout entre `a` et `j`. La correspondance est sensible à la casse des caractères donc si vous voulez inclure toutes les lettres, vous devz utilisez `[a-zA-Z]`.

Les motifs d'expansion sont très simples, à partir du moment où vous les voyez dans des exemples concrets :

– `*.txt`
  Cela vous fournit la liste de tous les fichiers finissant pas `.txt` puisque * correspond à tout.
– `*.[hc]`
  Cela vous fournit une liste de fichier se terminant soit par `.h` soit par `.c`.
– `a??`
  Cela vous fournit la liste de tous les fichiers de trois lettres commençant par un `a`.

[3] Actuellement, les fichiers commençant par un point ne sont pas inclus dans l'expansion de *

– `[^a]??`
  Cela vous fournit la liste de tous les fichiers de trois lettres ne commençant pas par la lettre `a`.
– `a*`
  Cela vous fournit l'ensemble des fichiers commençant par `a` sans tenir compte de leur nombre de lettres.

Chapitre 9

Un peu plus sur les fichiers

Dans la partie 7.2, nous avons montré comment déplacer et renommer des fichiers avec `mv`, comment les copier avec `cp`, comment les détruire avec `rm`, comment détruire des répertoires avec `rmdir` et comment en créer avec `mkdir`. Ce chapitre va couvrir des aspects un peu plus avancés du traitement de fichiers.

9.1 Permissions

Les systèmes Unix et GNU sont paramétrables pour pouvoir autoriser de nombreuses personnes à utiliser la même machine tout en conservant secrets certains fichiers ou en empêchant certaines personnes de modifier certains fichiers. Vous pouvez le vérifier par vous–même. Loguez–vous en tant que vous–même, c'est–à–dire *PAS en root*.

```
whoami
```

Ceci pour vérifier que vous n'êtes pas root. Ensuite, entrer la commande suivante :

```
rm /etc/resolv.conf
```

L'ordinateur devrait vous répondre `Permission denied`[1]. `/etc/resol.conf` est un des fichiers essentiels de configuration du système. Vous n'êtes pas autorisé à le modifier ou l'effacer si vous n'êtes pas root. Cela préserve le système d'une maladresse accidentelle. Si l'ordinateur est dans un organisme public, comme dans un bureau ou une école, cela le préserve des utilisateurs travaillant dessus.

Maintenant, tapez `ls -l /etc/resolv.conf`.

Cela devrait vous renvoyer quelque chose comme cela :

```
-rw-r--r-- 1 root root 119 Feb 23 1997 /etc/resolv.conf
```

[1] `Permission refusée`

L'option `-l` de `ls` lui demande d'ajouter des informations additionnelles. Les informations sur la partie droite sont faciles à interpréter. La taille du fichier fait 119 octets. La date de dernière modification est le 23 février 1997 et le nom du fichier est `/etc/resolv.conf`. Sur la partie gauche de l'écran, les choses sont un tout petit peu plus compliquées.

Tout d'abord, une brève explication technique : `-rw-r--r--` est le *mode* du fichier, le `1` correspond au nombre de liens physiques vers ce fichier (ou le nombre de fichiers dans le répertoire) et les deux `root` sont les propriétaires respectifs du fichier et du groupe auquel appartient le fichier.

Tout ceci est du charabia. Étudions les choses plus calmement.

9.1.1 Les droits de propriété d'un fichier

Tous les fichiers ont deux *propriétaires* : un utilisateur et un groupe. Le cas précédent peut porter à confusion car il y a un groupe nommé *root* et en même temps un utilisateur nommé *root*. Les groupes sont juste un rassemblement d'utilisateurs à qui on donne collectivement le droit d'accéder à une partie du système. Un bon exemple est le groupe *games*. Vous pouvez créer un groupe *games* sur votre ordinateur et le paramétrer de façon à ce que seuls les gens appartenant à ce groupe aient l'autorisation de jouer sur la machine.

Il y a des exemples plus pratiques : considérons le cas de la configuration d'une machine pour une école. Il y a certains fichiers dont vous voulez restreindre l'accès aux professeurs seulement, et pas aux élèves. Vous créez donc un groupe dans lequel vous rassemblez tous les professeurs. Ensuite, vous pouvez indiquer au système que certains fichiers appartiennent au groupe *professeurs* et que personne d'autre n'y a accès.

Explorons les groupes du système. En premier lieu, vous pouvez utiliser la commande `groups` du shell. Cela vous fournit une liste de groupes auxquels vous appartenez :

```
$ groups
system-wide configuration!permissions!file
ownershipusername dialout: cdrom floppy audio
```

Vous semblez donc n'appartenir qu'à un groupe, qui est identique à votre nom d'utilisateur. Dans tous les cas, root peut vous ajouter dans d'autres groupes. L'exemple suivant montre une personne membre de cinq groupes :

```
less /etc/group
```

Ce fichier contient la liste des groupes disponibles sur votre système. Remarquez bien le groupe root (le seul membre de ce groupe est l'utilisateur root) et le groupe qui correspond à votre nom d'utilisateur. Il y a aussi des groupes comme *dialout* (les utilisateurs autorisés à lancer le modem) et *floppy* (les utilisateurs qui ont accès au lecteur de disquette). Il est probable que votre système ne soit pas configuré pour avoir l'usage de ces groupes. Il est préférable que seul root ait accès au lecteur de disquettes ou au modem pour le moment. Pour les détails en ce qui concernent ce fichier, reportez–vous au `man group`.

```
ls -l /home
```

Cette commande vous montre que tous les répertoires utilisateurs sont la propriété de leur utilisateur et de leur groupe personnel.

Si vous venez d'installer Debian, vous êtes peut–être le seul utilisateur. Vous pouvez ajouter d'autres utilisateur à l'aide de la commande `adduser`.

9.1.2 Mode

En plus d'être la propriété d'un seul utilisateur et d'un seul groupe, chaque fichier et répertoire possède aussi un mode qui détermine qui a le droit de lire, d'écrire et d'exécuter un fichier (et de le lancer s'il s'agit d'un programme). Il y a d'autre petites fonctionnalités apportées par le mode mais ce sont des réglages avancées et nous les laisserons de côté pour le moment.

Un mode ressemble à cette sortie de `ls` : `-rw-r--r--`. À partir de maintenant, nous allons découvrir neuf ensembles : celui qui contrôle les permissions de *lecture*, d'*écriture* et d'*exécution* pour le propriétaire du fichier, pour le groupe auquel se rattache le fichier et pour les autres personnes (*others*) (tous ceux du système, quelque fois appelé le *monde*).

Dans la ligne mode, le premier « élément » donne le type de fichier. Le – signifie que c'est un fichier normal (*regular file*). Si nous avions eu `d`, cela aurait été un répertoire[2]. Il y a aussi quelques autres possibilités mais qui sont trop complexes pour le cadre de cette partie. Pour plus détails, allez voir la partie 15.2.2.

Les neuf éléments restant sont utilisés pour définir le mode du fichier. Les neuf bits basics (*read, write* et *execute* pour l'utilisateur, le groupe et les autres) sont définis en trois blocs de `rxw`.

Si toutes les permissions sont activées et qu'il s'agit d'un fichier normal, le mode va ressembler à ceci :

```
-rxwrxwrxw
```

Si c'était un répertoire avec toutes les permissions désactivées pour le monde et activées pour tous les utilisateurs et le groupe, cela ressemblerait à :

```
drxwrxw---
```

Le tableau suivant indique la signification des permission de lecture, écriture et exécution pour les fichiers et les répertoires :

Code	Nom	Ce que cela permet aux fichiers	Ce que cela permet aux répertoires
r	lecture	Examen du contenu du fichier	Liste le contenu du répertoire
w	écriture	Modification du fichier	Ajout ou destruction de fichiers dans le

[2] `d` pour *directory* en anglais

			répertoire
x	exécution	lancer en tant que commande	Accéder aux fichiers du répertoire

Les modes du répertoire peuvent porter un peu à confusion donc voici une série d'exemples des effets de combinaisons diverses.

```
r--
```

L'utilisateur, le groupe et les autres avec ces permissions peuvent afficher le contenu du répertoire mais ne peuvent rien faire d'autre. Les fichiers contenus dans ce répertoire ne peuvent être lus, modifiés ou manipulés de quelle que façon que ce soit. La seule action autorisée consiste à lire le répertoire en lui–même ce qui revient à afficher ce qu'il contient et donc la liste des fichiers.

```
rw-
```

La permission d'écriture n'a pas d'effet en l'absence de celle d'exécution donc ce mode se comporte exactement comme le mode précédent.

```
r-x
```

Ce mode permet aux fichiers présents dans le répertoire d'être visibles et permet aussi d'accéder à ces fichiers. Dans tous les cas, les fichiers ne peuvent être créés ou détruits. *Accéder* signifie que vous pouvez voir, changer et exécuter les fichiers suivant leurs permissions respectives.

```
--x
```

Les fichiers de ce répertoire sont accessibles mais le contenu du répertoire ne l'est pas. Il vous faut donc connaître à l'avance le nom du fichier que vous recherchez (à moins d'avoir des talents de devin). Les fichiers ne peuvent être ni créés ni détruits.

```
rwx
```

Vous pouvez faire tout ce que vous voulez avec les fichiers de ce répertoire tant que vous respectez leurs permissions respectives.

Les permissions d'écriture d'un répertoire détermine si vous pouvez détruire des fichiers dans ce répertoire. Un fichier en lecture seule pourra être effacé si vous avez les permissions d'écriture dans le répertoire le contenant. Vous ne pouvez détruire un fichier d'un répertoire en lecture seule même si vous avez les droits d'écriture sur le fichier.

Cela signifie que si vous êtes le propriétaire d'un répertoire, vous pouvez toujours détruire un de ses fichiers, y compris si ce fichier appartient à root.

La permission d'exécution d'un répertoire détermine la possibilité d'avoir accès aux fichiers – si bien sûr leurs permissions respectives l'autorisent. Si vous avez le droit d'exécution d'un répertoire, les droits d'exécution de ses fichiers devraient suivre. Sinon, les permissions des fichiers ne servent à rien : vous y avez accès dans tous les cas.

9.1.3 Les permissions en pratique

Cette partie contient une série d'exemples pour montrer comment utiliser les permissions. Pour changer les permissions, on utilise la commande `chmod`.

```
cd ; touch monfichier
```

Il y a une série de nouveaux trucs ici. En premier, vous pouvez utiliser `;` pour mettre deux commandes sur la même ligne. Vous pouvez entrer la commande précédente comme suit :

```
$ cd
$ touch monfichier
```

ce qui revient au même que

```
$ cd; touch monfichier
```

Les deux méthodes conduisent au même résultat.

Rappelez–vous que `cd` employé seul vous renvoie dans votre répertoire personnel. `touch` est normalement utilisé pour modifier la date de création d'un fichier à la date courante. Mais cela a une autre fonctionnalité intéressante. Si le fichier n'existe pas, `touch` le crée. Ainsi, vous êtes en train de créer un fichier pour pouvoir faire des expérimentations dessus. Utilisez `ls -l` pour observer que le fichier a bien été créé et remarquez son mode de permission :

```
$ ls -l
-rw-r--r-- 1 user user 0 Nov 18 22:04 monfichier
```

Bien–sûr, la date et le nom du groupe et de l'utilisateur seront différents chez vous. La taille du fichier est `0` car `touch` a créé un fichier vide. `-rw-r--r--` est le mode de permission par défaut sous Debian.

```
chmod u+x monfichier
```

Cette commande *ajoute* (`+`) des permissions d'*exécution* (`x`) pour l'*utilisateur* (`u`) à qui appartient le fichier. Utilisez `ls -l` pour en voir les effets.

```
chmod go-r monfichier
```

Maintenant, vous *ôtez* (`-`) les permissions de *lecture* (`r`) au *groupe* (`g`) à qui appartient le fichier et à tous les *autres* (`o`). À nouveau, utilisez `ls -l` pour voir les modifications.

```
chmod ugo=rx monfichier
```

Ici, vous positionnez (`=`) les permissions de lecture et d'exécution pour l'utilisateur, le groupe et les autres. Cela positionne les permissions exactement comme vous l'avez spécifié et enlève les autres permissions. Ainsi, seules les permissions `rx` seront activées, et toutes les permissions `w` seront retirées. À partir de maintenant, personne ne peut modifier le fichier.

```
chmod a-x monfichier
```

est un raccourci[3] pour `ugo`. Ainsi, seules les permissions d'exécution vont être retirées.

```
rm monfichier
```

Avec cette commande, nous allons tenter de détruire le fichier mais sans les droits d'écriture. `rm` va alors s'assurer de votre certitude en affichant le message suivant[4] :

```
rm: remove 'monfichier', overriding mode 0444 ?
```

Vous devrez répondre en tapant `y` et ensuite en appuyant sur `Entrée`. Ceci est une fonctionnalité de `rm` et non une conséquence des permissions. L'autorisation de détruire un fichier vient des droits sur le répertoire et vous avez les droits sur ce répertoire. Dans ce cas, `rm` essaie d'être le plus convivial possible en se figurant que si vous ne vouliez pas changer les permissions (et *a fortiori* enlever les droits d'écriture), vous ne vouliez pas plus le détruire et ainsi il vous demande de faire le choix.

Qu'est–ce donc cette affaire `0444` dans la question de `rm` ? Les modes de permission sont en fait un nombre de 12 bits comme celui–ci : `000100100100`. `0444` est représentation octal (i–e en base 8) qui est la base conventionnelle pour écrire un mode. Ainsi, vous pouvez écrire `chmod 444 monfichier` à la place de `chmod ugo=r monfichier`.

9.2 Présentation des fichiers et de leurs emplacements

Maintenant que vous êtes capable de naviguer au gré des répertoires, allons faire un tour auprès des fichiers et des répertoires que vous avez créés en installant Debian. Si vous êtes curieux, faîtes un `cd` dans chacun d'entre eux et tapez `ls` pour voir leur contenu. Si le résultat ne tient pas dans l'écran, essayez `ls | less` où | est le caractère *tube* que l'on trouve généralement au même emplacement que la touche *retour arrière*[5].

– `/`
  Comme on l'a déjà mentionné, il s'agit du répertoire racine qui contient tous les autres répertoires.
– `/root`
  Attention, ne confondez pas `/` avec `/root` ! `/root` est le répertoire personnel de l'utilisateur root (super–utilisateur). C'est un répertoire appelé `/root` mais ce n'est pas le répertoire racine (NDT : *root directory* en anglais d'où la confusion chez nos voisins anglophones).
– `/home`
  C'est l'endroit où l'on place le répertoire personnel de tous les utilisateurs normaux – c'est–à–dire tous sauf le root. Chaque répertoire personnel est appelé du nom de l'utilisateur ; par exemple `/home/jane`. Si vous utilisez un système de taille conséquente, comme pour une école ou une entreprise, votre administrateur système a peut–être créé des répertoires supplémentaires pour contenir les répertoires personnels : `/home1` et `/home2` par exemple. Dans d'autres systèmes, on peut apercevoir un niveau supplémentaire de sous–répertoire :

[3] `a` est pour *all* (tous) en anglais.

[4] `rm : détruire le fichier protégé en écriture 'monfichier' ?`

[5] NDT : du moins chez nos voisins anglo–saxons. Les francophones, avec des claviers *azerty*, le trouverons à la touche 6 en faisant *Alt Gr – 6*.

```
/home/eleves/nomdutilisateur,
/home/equipe/nomdutilisateur, etc...
```

Votre répertoire personnel est l'endroit où vous placez vos travaux personnels, votre courrier électronique et tous les autres documents, ainsi que vos fichiers de configurations personnelles. C'est votre « chez vous » dans le système...

- `/bin`
  Ce répertoire contient les « binaires » et les fichiers exécutables indispensables au fonctionnement du système. Des exemples sont le shell (`bash`) et les commandes relatives aux fichiers comme `cp`.
- `/sbin`
  Ce répertoire contient les « binaires systèmes ». Ce sont les utilitaires que l'utilisateur root ou l'administrateur système auront à utiliser. Vous n'aurez probablement jamais à y avoir accès dans votre travail quotidien.
- `/usr`
  Ce répertoire contient beaucoup de fichiers susceptibles de vous intéresser. Il a de nombreux sous–répertoires. `/usr/bin` et `/usr/sbin` sont très proches de `/bin` et `/sbin` à l'exception près que les fichiers y figurant ne sont pas considérés comme « vitaux pour le système ».
  Bien que non essentiel au bon fonctionnement de l'ordinateur, `/usr` contient bien des applications très utiles pour travailler. Ainsi, dans `/usr`, vous trouverez les répertoires `/usr/man`, `/usr/doc` et `/usr/info`. Ils contiennent respectivement les pages de manuel, la documentation et les pages info. Et n'oubliez pas non plus `/usr/games` !
- `/usr/local`
  Le système Debian n'installe rien dans ce répertoire. Vous devrez l'utiliser pour installer des logiciels que vous compilerez vous même ou tout logiciel non contenu dans un paquet Debian. Vous pouvez aussi l'installer dans votre répertoire personnel si vous êtes le seul à l'utiliser.
- `/etc`
  Ce répertoire contient l'ensemble des fichiers de configuration du système. Lorsque vous désirez changer quelque chose au système qui impacte tous les utilisateurs de votre ordinateur – comme votre connexion Internet ou bien un changement de carte vidéo – vous aurez probablement à vous loguer en root et à modifier un fichier dans `/etc`.
- `/tmp`
  Vous trouverez ici les fichiers temporaires de votre système, la plupart d'entre eux créés par le système. Ce répertoire est généralement nettoyé régulièrement ou à chaque *boot* du système. Vous pouvez créer des fichiers ici mais gardez à l'esprit qu'ils peuvent être détruits automatiquement.
- `/var`
  Ce répertoire contient les fichiers « variables » que le système change automatiquement. Par exemple, les courriels entrants sont stockés ici. Le système garde une trace de ses actions ici. Il y a un grand nombre d'autres fichiers engendrés automatiquement ici. Vous pouvez prêter attention au contenu du répertoire de `/var/log` où vous pourrez trouver des messages d'erreurs qui peuvent vous aider à comprendre ce qui ne va pas dans votre système en cas de défaillance.

Il y a évidemment bien d'autres répertoires dans votre système, bien trop d'ailleurs pour que l'on puisse tous les décrire ici.

Pour changer quelque chose, vous aurez certainement à vous contenter de votre répertoire personnel et de `/etc`. Sur un système Debian, il y a rarement une occasion de changer quelque chose d'autre car tout est automatiquement installé pour vous.

`/etc` est utilisé pour configurer le système comme un tout. Vous n'utiliserez votre propre répertoire personnel que pour configurer votre propre environnement et stocker vos propres données. L'idée de base est de vous confiner dans `/home/votrenom` pour le travail quotidien et ainsi, vous ne pourrez rien casser ailleurs. À l'occasion, vous vous loguerez en tant que root pour paramétrer quelque chose mais seulement si c'est absolument nécessaire. Bien–sûr, si vous utilisez Debian à l'école ou bien en entreprise et que quelqu'un d'autre est l'administrateur système, vous n'aurez pas accès au compte root et vous ne pourrez changer des choses que dans votre compte personnel (ou dans les comptes où vous avez accès). Ceci limite ce que vous pouvez faire avec le système.

9.3 La compression de fichier avec `gzip`

Il est souvent bien pratique de réduire la taille d'un fichier – disons, pour le télécharger plus rapidement, ou bien pour gagner de la place sur votre disque dur. Le programme affecté à cette tâche est `gzip` (GNU zip). Voici comment il fonctionne :

```
$ cd; cp /etc/profile ./monfichier
```

Cela vous renvoie dans votre répertoire personnel et copie un fichier arbitrairement choisi (`/etc/profile`) dans votre répertoire courant en le renommant `monfichier`.

```
$ ls -l
```

Affiche le contenu du répertoire courant. Remarquez bien la taille de `monfichier`.

```
$ gzip monfichier
```

Ceci comprime `monfichier`.

```
$ ls -l
```

Observez le résultat de cette commande : `monfichier` s'appelle maintenant `monfichier.gz`. Il est aussi beaucoup plus petit.

```
$ gunzip monfichier.gz; ls -l
```

Ceci décomprime le fichier. Remarquez bien que `monfichier` est retourné dans son état d'origine. Et remarquez aussi que pour décomprimer, il faut utiliser `gunzip` et non `gzip`.

```
$ rm monfichier
```

Utilisez cette commande pour effacer le fichier puisqu'on ne l'avait créé juste que pour expérimenter.

9.4 Rechercher des fichiers

Il y a deux utilitaires pour rechercher un fichier : `find` et `locate`. `find` recherche vraiment les fichiers dans leur état présent. `locate` recherche dans un index engendré automatiquement par le système chaque matin à 6h42 (c'est un `cron` dont on trouvera une explication un peu plus loin dans le livre). `locate` ne sera jamais capable de trouver un fichier créé après la génération de l'index. Bien–sûr, puisque `locate` recherche dans un index, il est beaucoup plus rapide – un peu comme rechercher dans l'index d'un livre plutôt qu'effeuiller chaque page.

Pour comparer les deux façons de faire une recherche, on va supposer que l'on ne se rappelle plus où est le fichier de configuration `XF86Config` du serveur X.

```
$ locate XF86Config
```

Cela devrait être très rapide. Vous allez obtenir une liste de noms de fichier *contenant* `XF86Config`, quelque chose comme :

```
/etc/X11/XF86Config
/usr/X11R6/lib/X11/XF86Config
/usr/X11R6/lib/X11/XF86Config.eg
/usr/X11R6/man/man5/XF86Config.5x.gz
```

Maintenant, essayez la commande `find` :

```
$ find / -name XF86Config
```

Vous allez entendre une activité intense du disque dur, et cela va prendre un temps assez long. Le résultat devrait ressembler à :

```
/ect/X11/XF86Config
/usr/X11R6:lib/X11/XF86Config
find: /var/spool/cron/atjobs: Permission denied
find: /var/spool/cron/atspool: Permission denied
find: /var/lib/xdm/authdir: Permission denied
```

Remarquez que `find` n'a trouvé que les fichiers appelés *exactement* `XF86Config` et non pas ceux contenant cette chaîne de caractères. `find` a aussi essayé de rechercher dans tous les répertoires du système – y compris ceux dont vous n'avez pas les droits d'accès. C'est la raison pour laquelle vous avez les messages `Permission denied` (`Permission non accordée`).

Leur syntaxe aussi est différente. Avec `find`, vous devez choisir à partir de quel répertoire effectuer la recherche alors que `locate` choisit automatiquement le répertoire racine. Et vous devez spécifiez votre recherche en utilisant l'option `-name`. Vous pouvez aussi rechercher des fichiers avec des tas d'autres critères, comme les dates de modification ou les propriétaires. Pour utiliser `find` dans une recherche de fichiers dont le nom contient `XF86Config`, il vous faut utiliser un motif de recherche :

```
$ find / -name '*XF86Config*'
```

Comme la plupart des outils en lignes de commandes, `find` accepte des motifs comme arguments.

De façon générale, `find` est un outil beaucoup plus puissant et `locate` est un outil très rapide pour des recherches journalières. L'intégralité des recherches possible prendrait un temps fou à expliquer. Pour plus de détails, tapez `info find` qui vous conduira au travers des méandres des subtilités de `find` et `locate`.

9.5 Déterminer le contenu d'un fichier

Debian vous offre un utilitaire capable de deviner le contenu d'un fichier à votre place. Bien que cela ne soit pas à 100% sûr, vous pouvez utiliser cette commande pour explorer votre système.

```
$ file /bin/cp
```

Vous devriez avoir quelque chose comme ceci :

```
/bin/cp: ELF 32-bit LSB executable, Intel 80386, version 1, dyna-
mically linked (uses shared libs), stripped
```

Si l'on retire les parties techniques, on peut dire que c'est un fichier exécutable pour machine Intel.

```
$ file /etc/init.d/boot
```

La commande précédente vous donne la réponse suivante :

```
/etc/init.d/boot: Bourne shell script text
```

ce qui veut dire que c'est un fichier texte contenant un script écrit en Bourne shell.

9.6 Utiliser un gestionnaire de fichier

Plutôt que manipuler les fichiers à la main, vous pouvez utilisez un *gestionnaire de fichiers*. Si vous manipulez de nombreux fichiers à la fois, un gestionnaire de fichiers peut rendre votre travail plus efficace. Il y a des gestionnaires de fichiers orienté texte, comme `GNU Midnight Commander (mc)` et un grand nombre de gestionnaires de fichiers pour X Window (par exemple `gmc` qui est la version X Window de `GNU Midnigt Commander`).

Décrire chacun d'entre est hors du propos de ce livre mais vous pouvez les essayer si la ligne de commande ne vous sied pas.

Chapitre 10

Travailler avec des fichiers textes

Les fichiers textes sont prédominants sur un système GNU–Linux. Ils se trouvent absolument partout, des fichiers de documentations aux fichiers de configuration. Par chance, il est très facile de les manipuler.

10.1 Lire des fichiers textes

Un fichier texte est simplement un fichier qui ne contient que des caractères interprêtables par un être humain. Il n'y a rien à dire de spécial de plus à son sujet. L'autre type de fichier est le fichier binaire qui est destiné à être interprêté par l'ordinateur.

Vous pouvez visualiser n'importe quel type de fichiers avec le pagineur `less` si vous l'avez installé (installez–le si vous ne l'avez pas, c'est très simple). Tapez `less /etc/profile` pour lire un exemple de fichier texte. Remarquez bien que vous pouvez lire des caractères mêmes si leur signification reste obscure. Tapez `less /bin/ls` pour visualiser un fichier binaire. Comme vous pouvez vous en apercevoir, le programme `ls` n'est pas lisible par un être humain.

Parfois, vous trouverez des fichiers finissant par `.gz` . Ces fichiers sont lisibles avec `zless`. Vous pouvez l'utiliser comme ceci :

```
zless /usr/doc/ae/changelog.Debian.gz
```

`zless` est idéal pour lire la documentation qui est la plupart du temps comprimée au format `.gz` .

La différence entre les deux sortes de fichiers est purement en rapport avec ce qu'ils contiennent à la différence de certains autres systèmes (comme le DOS ou MacOS) qui les traitent de façon différente.

Les fichiers textes peuvent contenir des scripts shell, de la documentation, des notices de *copyrights* ou tout autre texte lisible par un être humain.

Incidemment, ceci illustre la différence entre le *code source* et les *binaires exécutables*. `/bin/ls` est un fichier binaire exécutable que vous pouvez télécharger sur votre Debian mais vous pouvez aussi télécharger un fichier texte qui explique à l'ordinateur comment créer `/bin/ls`. Le fichier texte est alors le code source. Comparer la nature de `/bin/ls` et `/etc/profile` illustre combien le code source est important pour quelqu'un qui veut comprendre ou modifier une partie d'un logiciel. Les logiciels libres vous fournissent, à vous ou bien à vos consultants, tout le code source de chaque application.

10.2 Les éditeurs de textes

Un *éditeur de textes* est le programme utilisé pour créer ou changer le contenu d'un fichier texte. La plupart des systèmes d'exploitation possède leur éditeur de textes : DOS a `edit`, Windows a `Notepad` et MacOS a `SimplyText`.

Debian fournit une grande quantité d'éditeurs de textes. `vi` et `Emacs` en sont les deux grands classiques et ce sont probablement les plus puissants et les plus utilisés. `vi` et `Emacs` sont tous les deux complexes et requièrent un certain apprentissage, mais ils peuvent éditer du texte avec une efficacité extraordinaire. `Emacs` tourne à la fois en mode console et sous X Window ; `vi` tourne normalement seulement en mode console mais la variante `vim` possède une option `-g` qui lui permet de tourner sous X.

Il existe des éditeurs de textes plus simples comme `nedit`, `ae`, `jed` et `xcoral`. `nedit` et `xcoral` fournissent une interface basée sur le système X Window très conviviale et facile à utiliser. Il existe aussi de multiples variantes de `vi`. Vous pouvez aussi trouver une variante de `Emacs` appelée `XEmacs`.

Ce livre n'aborde pas l'utilisation d'un éditeur en détail, bien que nous allons rapidement faire une introduction à `ae` qui est petit, rapide et peut être trouvé sur les disquettes de secours de Debian. Il est donc recommandé de savoir s'en servir au cas où...Lorsque vous désirerez faire de l'édition de texte de façon plus sérieuse, voyez du côté de `vim` ou de `GNU Emacs`. `Emacs` fournit un excellente présentation interactif[1] de ses capacités ; pour le lire, lancer `Emacs` avec la commande `emacs` et taper `F1 t`. `Emacs` est un très bon choix pour les nouveaux utilisateurs intéressés par un éditeur tout usage ou pour la programmation.

10.3 Utiliser `ae`

Vous pouvez lancer `ae` en lui donnant le nom d'un fichier à éditer, comme cela :

```
$ ae fichier.txt
```

Cela va vous faire entrer dans l'écran d'édition. La partie supérieure vous propose une aide sommaire ; la partie inférieure montre le nom du fichier que vous éditez. Se déplacer dans cet éditeur est simple : il suffit d'utiliser les flèche de déplacement. Vous pouvez sauver le fichier en tapant `C-x C-s` et ensuite sortir en tapant `C-x C-c`. Lorsque vous serez à l'aise, vous pourrez supprimer l'aide en tapant `C-x C-h`. C'est tout ! Gardez à l'esprit que cet éditeur ne vous permet que de l'édition très basique. Pour programmer, ou bien pour des travaux d'édition plus avancés, vous chercherez un autre éditeur de textes comme il a été dit précédemment.

[1] mais en anglais...

Chapitre 11

Le système X Window

Le but de ce chapitre est de décrire l'interface graphique utilisateur du système X Window. Cela suppose que vous avez réussi à configurer X comme décrit dans le manuel d'installation (même si le manuel d'installation n'est pas encore écrit ; vous aurez besoin d'utiliser le XFree86 HOWTO, le contenu de `/usr/doc/X11` et ce chapitre). Une fois X installé, vous pouvez entrer dans l'environnement X soit en tapant `startx` soit via `xdm` : cela dépend de vos choix pendant la configuration.

11.1 Introduction à X

Une `GUI` (*Graphical User Interface*, une interface graphique utilisateur) fait partie des systèmes Windows et de MacOS. Il est pratiquement impossible d'écrire une application pour ces systèmes sans utiliser ces GUI et les systèmes ne peuvent être utilisés à l'aide de lignes de commandes. GNU–Linux est plus modulaire car il est composé de petits composants indépendants que l'on peut ou non utiliser suivant ses besoins ou ses préférences. Un des ces composants est le système X Window, que l'on appelle simplement X.

Ce composant est aussi appelé parfois X11. Remarquez bien que l'appellation « X Windows » est incorrecte.

X en lui–même est une manière pour les programmes de discuter avec la souris et la carte vidéo sans savoir quel type de matériel il y a au bout. En fait, c'est une *abstraction* du matériel graphique. Les applications des utilisateurs parlent au serveur X dans le langage du serveur X, le serveur X le traduisant alors dans le langage de votre matériel. Cela signifie que votre programme n'a qu'à être écrit une fois et qu'ensuite il tournera sur tous les ordinateurs.

Dans le jargon de X, le programme qui parle au matériel est connu sous le vocable *serveur X*. Les applications qui demandent au serveur X d'afficher une fenêtre sur l'écran sont appelés *clients X*. Le serveur X inclut un pilote pour la carte vidéo et donc vous devez avoir un serveur X qui reconnaît votre carte vidéo.

Le serveur X ne fournit aucune fonctionnalité que l'on peut attendre d'une *GUI*, comme redimensionner une fenêtre ou bien les réarranger. Un client X spécial, appelé le *gestionnaire de fenêtres*, dessine les bords des fenêtres et les barres de titre, redimensionne et réarrange les fenêtres et fournit des facilités

pour lancer d'autres clients X via un menu. Des gestionnaires de fenêtres spécifiques peuvent avoir d'autres fonctionnalités.

Les gestionnaires de fenêtres présents dans un système Debian sont `fvwm`, `fvwm2`, `icewm`, `afterstep`, `olvwm`, `wmaker`, `twm` et `enlightenment`. Vous devrez probablement tous les essayer avant de ne garder que votre favori.

Ni le serveur X, ni le gestionnaire de fenêtres ne fournissent de *gestionnaire de fichiers*, c'est–à–dire qu'il n'existe pas de fenêtres contenant les icônes de vos fichiers et répertoires. Vous pouvez lancer un gestionnaire de fichiers en tant qu'application séparée, et il y en a de nombreux de disponibles. Le projet de bureau GNOME se développe autour d'un gestionnaire de fichiers basé sur des icônes et sur d'autres fonctionnalités basées sur des *GUI*. Allez voir la page GNOME[1] pour obtenir les dernières nouvelles.

La dernière fonctionnalité de X est sa *transparence au réseau* : cela signifie que les clients X ne s'occupent pas de savoir s'ils discutent avec un serveur X tournant sur la même machine qu'eux ou sur un serveur X fonctionnant quelque part sur le réseau. De façon pratique, cela veut dire que vous pouvez faire tourner à distance votre programme sur une autre machine très puissante et sortir l'affichage sur votre machine.

11.2 Lancer l'environnement X

Il y a deux façons de lancer X. La première est de le lancer manuellement aussitôt que vous en avez besoin. Pour ce faire, Loguez–vous dans une console et tapez `startx`. Cela lance X et vous met automatiquement sur sa console virtuelle.

La seconde façon (et recommandée) de lancer X est d'utiliser `xdm` (*X display manager*, le gestionnaire d'affichage graphique). Basiquement, `xdm` vous propose un joli prompteur graphique de login sur la console virtuelle de X (probablement la 7) et vous pouvez vous loguer à partir de là.

Par défaut, n'importe quelle méthode lancera aussi une `xterm` qui est une petite fenêtre contenant un prompteur de shell. À partir de ce prompteur, vous pouvez taper n'importe quelles commandes comme vous le faisiez sur une console texte. Vous pouvez ainsi refaire tous les exemples du livre en utilisant l'`xterm`. La seule différence entre une `xterm` et la console texte est qu'il n'est pas nécessaire de se loguer dans l'`xterm` puisque vous l'êtes déjà sous X.

Il y a d'autre part beaucoup de choses que vous ne pouvez faire que sous X et dont on va parler dans ce chapitre.

Attention : la fenêtre par défaut de l'`xterm` a une toute petite fonte. Si vous avez un tout petit moniteur ou une définition élevée ou bien encore de mauvais yeux, il se peut que vous vouliez changer cet état de fait. Pour ce faire, effectuez les étapes suivantes :

– Déplacez le curseur de la souris au centre de la fenêtre `xterm`.
– Appuyez sur la touche *Control* et le bouton droit de la souris simultanément. Vous devriez voir apparaître un menu de fontes.
– Pointez sur la fonte que vous voulez et relâchez le bouton de la souris.

[1] http ://www.gnome.org/

11.3 Les opérations élémentaires sous X

Il y a certaines opérations très communes sous X avec lesquelles il va falloir vous familiariser. Cette partie décrit quelques opérations élémentaires que vous pouvez trouver très pratiques.

11.3.1 La souris

La souris sous X fonctionne presque de la même façon que sous les autres systèmes d'exploitation à l'exception du fait qu'elle possède trois boutons. Si votre souris n'en possède que deux, vous pouvez simuler le troisième bouton du milieu en appuyant sur les deux boutons simultanément. Ce n'est pas très pratique ; donc investir dans une souris trois boutons bon marché n'est pas une mauvaise idée. Il y en a de disponible chez tous les détaillants informatiques.

Les boutons sont numérotés de gauche à droite, en supposant que vous placiez votre souris dans votre main droite. Le premier bouton est alors sur la gauche, le deuxième au milieu et le troisième sur la droite. Vous pouvez rencontrer soit la numérotation soit les positions dans la documentation.

X possède une fonctionnalité interne simple de copier–coller. Pour sélectionner le texte à copier, vous devez à cliquer puis à déplacer la souris avec le bouton gauche. Cela devrait sélectionner le texte à copier, en supposant bien sûr que l'application en question supporte le copier–coller. Pour restituer le texte, vous n'avez qu'à cliquer avec le bouton du milieu dans une autre application X. Par exemple, vous recevez un courriel contenant une URL. Vous pouvez la sélectionner avec le bouton gauche et ensuite cliquer dans le champ *ad hoc* de votre butineur avec le bouton du milieu pour la recopier.

11.3.2 Les clients X

Les programmes communiquant avec le serveur X sont appelés des *clients X*. La plupart de ces programmes lui demandent d'afficher une fenêtre à l'écran.

Vous pouvez lancer un client X de la même façon que vous lancez un autre programme Debian. Tapez simplement le nom du programme en ligne de commandes. Essayez de taper `xterm` dans la fenêtre `xterm` déjà existante et vous aurez une nouvelle fenêtre `xterm` qui apparaîtra à l'écran.

Vous devez vous apercevoir que la première `xterm` est bloquée car le shell attend que la seconde `xterm` se finisse. Pour éviter ce problème, vous devez lancer le client X en arrière–plan, en utilisant le & après le nom de la commande, comme ceci : `xterm &`. Si vous oubliez, vous pouvez lancer un processus déjà existant en arrière–plan : commencez par le suspendre avec *Ctrl–z* et ensuite placez–le en arrière–plan avec la commande `bg`.

Si vous utilisez régulièrement un programme, votre gestionnaire de fenêtres doit vous fournir un moyen plus pratique de le lancer via un menu graphique.

11.3.3 Problèmes

Parfois, lorsque vous lancez un client X à partir d'un menu graphique, vous n'arrivez pas à voir les messages d'erreurs en cas d'échec. Vous pouvez les trouver dans le fichier `~/.xsession-errors`.

11.3.4 Quitter l'environnement X

Pour sortir de X, il est nécessaire d'utiliser un menu. Malheureusement pour les débutants, cela change avec chaque gestionnaire de fenêtres et pour la plupart d'entre eux, cela peut être fait de nombreuses façons différentes. S'il existe un menu évident, rechercher une entrée du style *Exit* (quitter) ou bien *Close the Window Manager* (Fermer le gestionnaire de fenêtres). Si vous n'apercevez aucun menu, essayez de cliquer avec chacun des boutons sur l'arrière–plan de l'écran. Si tout cela a échoué, vous pouvez toujours tuer le serveur X en pressant sur la combinaison *Ctrl–Alt–Backspace*. Bien entendu, tuer le serveur vous fait perdre toutes les données non sauvegardées des applications encore ouvertes.

11.3.5 Personnaliser le lancement de X

Lorsque vous lancez X, Debian exécute quelques scripts shell qui lancent votre gestionnaire de fenêtres et d'autres clients X. Par défaut, il y a un gestionnaire de fenêtres, un `xconsole` et une `xterm`.

Pour personnaliser votre lancement de X, le fichier `/etc/X11/config` doit contenir la ligne `allow-user-xsession`. Si ce n'est pas le cas, loguez–vous en root et ajoutez la maintenant. Ensuite reloguez en tant que vous–même et continuez cette présentation.

Vous pouvez voir ce que fait Debian au lancement de X dans le fichier `/etc/X11/Xsession`. Remarquez bien que vous pouvez changer le comportement de `/ect/X11/Xsession` en modifiant le fichier `/ect/X11/config` qui spécifie les caractéristiques du système par défaut.

Pour lancer les clients de votre choix au démarrage de X, vous devez créer un fichier script exécutable appelé `.xsession` dans votre répertoire personnel.

```
$ touch ~/.xession
```

pour le créer et

```
$ chmod u+x ~/.xsession
```

pour le rendre exécutable.

Une fois que `.xession` est créé, il est nécessaire de le modifier pour pouvoir faire quelque chose d'utile avec votre éditeur de textes favori. Vous pouvez faire ce que vous voulez dans ce script. Dans tous les cas, lorsque les processus du script s'arrêteront, la session X aussi s'arrêtera.

De façon pratique, cela signifie que vous devez toujours terminer vos scripts avec un appel à `exec`. Quelque soit le programme que vous appelez avec `exec`, il remplacera le processus en cours par ce programme et ainsi les commandes suivantes après la ligne `exec` seront ignorées. Le programme que vous appellerez via `exec` deviendra le nouveau propriétaire du processus de script, ce qui signifie que X s'arrêtera lorsque le nouveau programme s'arrêtera.

Mettez par exemple à la fin de votre `.xsession` une ligne avec `exec fvwm`. Cela signifie que le gestionnaire de fenêtres `fvwm` sera lancé au démarrage de X. Lorsque vous quitterez votre gestionnaire de fenêtres `fvwm`, vous quitterez aussi la session X et tous les autres clients seront fermés. Il n'est pas absolument nécessaire de lancer un gestionnaire de fenêtres ici ; vous pouvez y écrire `exec xterm` et dans ce cas, tapez `exit` dans cette `xterm` particulière fermera toute la session X.

Si vous désirez lancer d'autres clients avant votre `exec`, il sera nécessaire de les lancer en arrière–plan. Sinon, `.xsession` s'arrêtera jusqu'à ce que chaque client existant disparaisse, et ainsi de suite, ligne après ligne. Voyez la partie précédente pour envoyer un processus en arrière–plan (basiquement, vous devez ajouter un & à la fin de la ligne, comme pour `xterm&`).

Vous pouvez tirer parti de ceci. Si vous désirez lancer des commandes à la fin de votre session X, vous pouvez lancer votre gestionnaire de fenêtres, ou quoi que ce soit d'autre équivalent, dans votre `.xsession` et attendre qu'il finisse. Pour ce faire, n'utilisez pas `exec` ni les &. Entrez simplement `fvwm`. Ensuite, placez les commandes de votre choix après `fvwm`.

Quelques exemples de fichiers `.xession` devraient vous aider. Dans les exemples, remplacez `fvwm` par le gestionnaire de fenêtres de votre choix.

Le plus simple des `.xsession` est celui qui ne fait que lancer un gestionnaire de fenêtres :

```
exec fvwm
```

Cela lance `fvwm` et la session X s'achève lorsque l'on quitte `fvwm`. Si vous faîtes la même chose mais sans le `exec`, rien ne semblera différent à première vue. Mais en arrière–plan, `.xsession` sera en attente des réactions de `fvwm` et `.xsession` ne s'arrêtera qu'à la fin de `fvwm`. Utilisez `exec` est plus léger car `fvwm` remplace alors `.xsession` plutôt que d'attendre qu'il s'arrête. Vous pouvez le vérifier à l'aide des commandes `ps` ou `top`.

Un fichier `.xsession` plus complet lance quelques clients X avant de lancer le gestionnaire de fenêtres. Par exemple, vous pouvez avoir envie d'avoir des `xterm` et un `xclock` après le démarrage de X. Pas de problème : entrez simplement `xterm& xterm& xclock& exec fvwm`. Deux `xterms` se lanceront au démarrage en arrière–plan, ainsi qu'un `xclock` et ensuite le gestionnaire de fenêtres sera lancé. Lorsque vous quittez le gestionnaire de fenêtres, vous quitterez aussi X.

Vous pouvez essayer sans la mise en arrière–plan pour voir ce qui se passe. Entrez la commande : `xterm xclock exec fvwm`. `xterm` se lance, et attend que vous en sortiez. Ensuite, `xclock` se lance. Vous devez quitter `xclock` pour lancer `fvwm`. Les commandes sont lancées séquentiellement car le script attend que chacune d'entre elles se finisse avant passer à la suivante.

Vous pouvez utiliser le caractère séquentiel de cette exécution à votre avantage. Peut–être avez–vous envie de garder chaque jour une trace de l'heure de votre arrêt de travail :

```
xterm&
xclock&
fvwm
date >> ~/heures-de-sortie
```

Cela va mettre en arrière–plan `xterm` et `xclock` et lancer `fvwm` et attendre qu'il finisse. À la sortie de `fvwm`, cela exécutera la dernière ligne, qui extrait la date courante et l'envoie dans le fichier `~/heures-de-sortie`.

Pour finir, vous pouvez avoir un programme autre qu'un gestionnaire de fenêtres pour déterminer la sortie du serveur X :

```
xclock&
```

```
fvwm&
exec xterm
```

Ce script va envoyer `xclock`et `fvwm` en arrière–plan et se remplacer par `xterm`. Lorsque vous quittez cette `xterm`, votre session X s'arrête.

La meilleure façon d'apprendre comment fonctionne `.xession` est encore de l'essayer. N'oubliez pas de le rendre rendre exécutable : l'oublier est une erreur très classique.

Chapitre 12

Les systèmes de fichiers

Un système Debian utilise un système de fichiers pour stocker et gérer vos données. Ce chapitre est une introduction au système de fichiers, décrivant comment ajouter et retirer des systèmes de fichiers et comment sauvegarder votre système.

12.1 Concepts

C'est probablement une bonne idée de faire un peu de théorie avant de discuter de l'utilisation des disques. En particulier, vous devez comprendre le concept de *système de fichiers*. Cela peut être un peu déroutant à cause des significations diverses d'un système de fichiers.

Le système de fichiers concerne l'arborescence entière des répertoires qui débute par le répertoire racine « / » comme décrit dans les chapitres précédents.

Un système de fichiers signifie généralement n'importe quelle organisation de fichiers et répertoires sur un périphérique physique particulier. « Organisation » signifie la structure hiérarchique des répertoires et toute les informations sur les fichiers que l'on veut garder : leur taille, leurs permissions, etc. . .Ainsi, vous pourriez avoir un système de fichiers sur votre disque dur et un autre sur chacune de vos disquettes.

On utilise aussi le terme « système de fichiers » pour désigner le *type particulier* de système de fichiers. Par exemple, MS–DOS et Windows 3.1 organisent les fichiers d'une certaine façon : les noms de fichiers ne peuvent dépasser huit caractères par exemple et il n'existe aucune information de permission sur les fichiers. Linux appelle ça le système de fichier `msdos`. Linux a aussi son propre système de fichiers appelé `ext2` (version 2 du système de fichiers `ext`). Vous utiliserez la plupart du temps le système de fichiers `ext2` à moins d'accéder à des fichiers issus d'autres systèmes d'exploitation, ou bien d'avoir des besoins particuliers.

Tout périphérique physique que vous désirez utiliser pour garder des fichiers doit avoir au moins un système de fichiers. Cela signifie un système de fichiers au second sens du terme, c'est–à–dire une hiérarchie de fichiers et de répertoires avec l'information qui va avec. On parle là du troisième sens. Si vous avez plus d'un système de fichiers sur un périphérique unique, chaque système de fichiers peut avoir un type différent. Par exemple, vous pouvez avoir en même temps une partition DOS et une partition Linux sur votre disque dur.

12.2 `mount` et `/etc/fstab`

Cette partie décrit comment monter une disquette ou un lecteur ZIP, parle du répertoire `/dev` et discute comment l'arbre des répertoires peut se distribuer sur de nombreux périphériques physiques ou des partitions.

12.2.1 Monter un système de fichiers

Dans un système GNU–Linux, il n'y a pas forcément de correspondance entre les répertoires et les périphériques physiques comme cela existe sous Windows pour lequel chaque périphérique a son propre arbre de répertoires commençant par une lettre (comme C :\).

Au lieu de ça, chaque périphérique physique, un disque dur ou bien une disquette, peut contenir un ou plusieurs systèmes de fichiers. Pour rendre un système de fichiers accessible, on lui donne un répertoire particulier dans l'autre système de fichiers. Pour éviter des problèmes de circularité le système de fichiers racine (qui contient le répertoire racine « / ») n'est contenu dans aucun autre système de fichiers. Vous y avez accès automatiquement après avoir lancé Debian.

Un répertoire dans un système de fichiers qui contient un autre système de fichiers est appelé *point de montage*. Un point de montage est un répertoire dans un premier système de fichier (sur un périphérique, comme votre disque dur par exemple), qui « contient » un second système de fichiers, lequel peut être placé ailleurs, sur un lecteur de disquette par exemple. Pour accéder à un système de fichiers, vous devez le monter sur un point de montage.

Ainsi, par exemple, vous pouvez monter un cédérom au point de montage `/cdrom`. Cela signifie que si vous allez jeter un coup d'oeil dans le répertoire `cdrom`, vous y verrez le contenu du cédérom. Le répertoire `/cdrom` lui–même est en fait sur votre disque dur. Pour des tas de raisons pratiques, le contenu du cédérom devient une partie du système de fichiers racine et lorsque vous tapez des commandes ou bien que vous utilisez des programmes, l'emplacement réel des fichiers n'importe pas. Vous auriez pu avoir créé un répertoire appelé `/cdrom` et mis quelques fichiers à l'intérieur et tout ce serait passé de la même façon. Lorsque vous avez monté un système de fichiers, il n'est plus nécessaire de prêter la moindre attention à son emplacement physique.

Toutefois, avant de monter un système de fichiers ou d'en créer un nouveau sur un disque qui n'en contient pas, il est nécessaire de se référer aux périphériques eux–mêmes. Chaque périphérique a son nom et ces noms se trouvent dans le répertoire `/dev`. Si vous tapez `ls /dev` maintenant, vous apercevrez une longue liste de l'ensemble des périphériques possibles (dont la liste est fournie chapitre 4). Une liste plus détaillée peut être trouvée sur votre système dans le fichier `/usr/src/linux/Documentation/devices.t`

Vouloir monter un système de fichiers, c'est dire à Linux d'associer un système de fichiers sur tel périphérique avec tel point de montage particulier. Dans la procédure, il faudra aussi indiquer quel type de système de fichiers on veut.

12.2.2 Exemple : monter un cédérom

Comme exemple élémentaire, nous allons monter un cédérom comme celui que vous avez utilisé pour installer Debian. Vous avez besoin d'être root pour faire cela donc faîtes attention : chaque fois que vous

êtes root, vous avez le pouvoir de manipuler tout le système et pas simplement vos seuls fichiers. Ces commandes supposent d'autre part que vous avez un cédérom dans le lecteur : vous devez donc le mettre maintenant. Ensuite, commencez avec la commande suivante :

```
su
```

Si vous ne l'avez pas déjà fait, vous devez soit vous loguer en tant que root soit obtenir les privilèges du super–utilisateur via la commande `su`. Si vous utilisez `su`, ajoutez le mot de passe de root quand on vous le demande.

```
ls /cdrom
```

Utilisez cette commande pour voir ce que contient le répertoire `/cdrom` avant de commencer. Si vous n'avez pas de répertoire `/cdrom`, il vous faudra le créer en utilisant `mkdir /cdrom`.

```
mount
```

Taper simplement `mount` sans argument donne la liste des systèmes de fichiers actuellement montés.

```
mount -t iso9660 CD-device /cdrom
```

Avec cette commande, vous devez mettre le nom de votre cédérom à la place de `CD-device`. Si vous n'êtes pas sûr, mettez `/dev/cdrom` car la procédure d'installation devrait avoir créé ce lien symbolique sur le système. Si cela échoue, essayez les différents périphériques IDE : `/dev/hdc`, etc. . .Vous devez voir un message du type[1] :

```
mount: block device /dev/hdc is write protected, mounting read-only
```

L'option `-t` spécifie le type de système de fichiers, dans ce cas `iso9660`. L'argument suivant est le nom du périphérique à monter et le dernier argument est le point de montage. Il y a de nombreux autres arguments possibles pour `mount` ; allez voir la page de manuel pour plus de détails.

Une fois le cédérom monté, vous remarquerez que le plateau du cédérom ne s'ouvre pas : : vous devez le démonter avant de pouvoir le démonter.

```
ls /cdrom
```

confirme maintenant que `/cdrom` contient bien ce qu'il y a dans votre cédérom.

```
mount
```

Affiche à nouveau la liste des systèmes de fichiers ; remarquez que votre lecteur est maintenant monté.

```
umount /cdrom
```

Ceci démonte le cédérom. Il est maintenant à nouveau possible de sortir le cédérom de son lecteur. Attention, la commande est `umount` sans `n` même s'il s'agit de démonter[2] le système de fichiers.

```
exit
```

N'oubliez pas de vous déloguer du compte root. Déloguez–vous immédiatement, juste pour être sûr.

[1] mount : le périphérique bloc /dev/hdc est protégé en écriture, montage en lecture seule

[2] *unmount* en anglais, d'où la remarque

12.2.3 `/etc/fstab` : automatiser la procédure de montage.

Le fichier `/etc/fstab` (qui veut dire *file system table*, table des systèmes de fichiers) contient les descriptions des fichiers que vous montez souvent. Ces systèmes de fichiers peuvent être montés avec des commandes réduites, comme `mount /cdrom`. Vous pouvez aussi faire qu'ils soient montés automatiquement au démarrage.

Jetons un coup d'oeil à ce fichier en tapant `more /etc/fstab`. Il devrait y avoir deux ou plusieurs entrées configurées automatiquement lors de la procédure d'installation. Cela devrait probablement ressembler à quelque chose comme cela :

```
# /etc/fstab: static file system information
#
# < file system > < mount point > < type > < options >
# < dump >  < pass >/dev/hda1 / ext2 defaults 0 1
/dev/hda3 none swap sw 0 0
proc /proc proc defaults 0 0
/dev/hda5 /tmp ext2 defaults 0 2
/dev/hda6 /home ext2 defaults 0 2
/dev/hda7 /usr ext2 defaults 0 2
/dev/hdc /cdrom iso9660 ro, noauto 0 0
/dev/fd0 /floppy auto noauto,sync 0 0
```

La première colonne affiche les périphériques présents sur le système. La seconde colonne est le point de montage, la colonne suivante étant le type de système de fichiers. Remarquez que la partition de *swap* (`/dev/hda3` dans l'exemple) n'a pas de point de montage et donc il y a un `none` dans la colonne correspondante.

Les trois dernières colonnes méritent une explication plus approfondie.

La cinquième colonne est utilisée par l'utilitaire `dump` pour décider quand sauvegarder le système de fichiers. Dans la plupart des cas, vous pouvez mettre 0 ici.

La sixième colonne est utilisée par `fsck` pour décider dans quel ordre vérifier les systèmes de fichiers au démarrage. Le système de fichiers racine ou principal doit avoir le `1` dans son champ, les systèmes de fichiers n'ayant besoin d'aucune vérification (comme la partition de *swap*) doivent avoir un `0` et tous les autres doivent avoir un `2`. Il est vrai que la partition de *swap* ne possède pas vraiment un système de fichiers, au sens où elle ne contient pas des fichiers et des répertoires mais elle est utilisée par le noyau Linux comme mémoire secondaire. Quoiqu'il en soit, pour des raisons historiques, la partition de *swap* est toujours contenu dans le fichier incluant les systèmes de fichiers.

La quatrième colonne contient une ou plusieurs options à utiliser lors du montage du système de fichiers. Vous pouvez vérifier la page de manuel de `mount` pour un résumé. Allez voir aussi le chapitre 7.1.

12.2.4 Supports extractibles (disquettes, lecteur Zip, etc.)

Ajoutez les lignes suivantes dans votre fichier `/etc/fstab` :

```
/dev/sda1 /mnt/zip ext2 noauto,user 0 0
/dev/sda4 /mnt/dos msdos no,user 0 0
```

À partir de maintenant, vous êtes capable de monter une cartouche Zip formatée en DOS avec la commande `mount /mnt/dos` et vous pouvez monter une cartouche Zip formatée pour Linux avec la commande `mount /mnt/zip`.

Si vous avez des disques durs SCSI sur votre système, vous devez changer `sda` en `sdb` ou `sdc` dans l'exemple ci–dessus.

12.3 Les outils de sauvegarde

Les sauvegardes sont quelque chose de fondamental dans tous les systèmes d'exploitation. Debian GNU–Linux fournit différents utilitaires que vous pourrez utiliser à votre convenance. De plus, bien que la plupart de ces utilitaires aient été créés originellement pour des sauvegardes sur bandes, vous pourrez les utiliser pour plein d'autres choses. Par exemple, `tar` est utilisé pour distribuer des programmes sur Internet. Voici une liste des utilitaires que vous pourrez trouver sur le système :

– `Taper` est un programme basé sur des menus, très convivial et qui peut sauvegarder sur de nombreux supports. Sa limitation est qu'il n'est pas capable de faire des sauvegardes supérieures à 4 Go.
– `dump` a été créé spécialement pour les bandes : sa force principale réside dans son interface pour la restauration de fichiers, les sauvegardes de systèmes de fichiers de bas niveau et la sauvegarde incrémentale. Sa limitation réside dans le fait qu'il n'est pas capable de sauvegarder via NFS ou d'autres systèmes de fichiers différents de `ext2` et quelques petits défauts de conception.
– `GNU tar` (raccourci de *Tape ARchiver*, archiveur d'enregistrements) est une implémentation de ce qui est probablement la façon la plus utilisée de sauvegarder quelque chose sous Linux aujourd'hui. Cela fournit un bon outil multi–usage et peut s'accomoder de l'ensemble des supports utilisés. De plus, de nombreux systèmes sont capables de lire un fichier `tar` ce qui le rend très portable. Les défauts de `tar` sont qu'il contient un pauvre outil de sauvegarde incrémentale en comparaison de `dump` et aucun écran de sélection de restauration automatique.

12.3.1 `tar`

Parce que `tar` est vraiment très utilisé, et pas seulement pour des tâches de sauvegardes, nous allons en parler un peu ici. Pour plus de détails, lisez la page de manuel de `tar`, les instructions pour lire une page de manuel sont dans la partie 7.1.

`tar` est un archiveur. Cela veut dire que `tar` est capable de rassembler plusieurs fichiers et les combiner en un seul grand fichier pour les écrire sur un périphérique de sauvegarde comme une bande. Une fois que vous avez ce grand fichier, vous voulez souvent le comprimer : l'option `-z` est alors très bien. Ainsi, `tar` est une façon pratique de distribuer des programmes et des données sur Internet et vous vous rendrez compte qu'il est beaucoup utilisé à cette fin.

Voici une ligne de commande `tar` simple :

```
tar -zcvf monfichier.tar.gz /usr/local/bin
```

Regardons un peu comment se décompose la commande `tar` :

- `tar`
  Nom de la commande
- `-`
  Options de la commande `tar`
  - `z`
    demande à `tar` d'utiliser `gzip` pour comprimer automatiquement . Si vous utilisez cette option, il est bon d'ajouter l'extension `.gz` à votre fichier `tar`.
  - `c`
    demande à `tar` de créer une nouvelle archive
  - `v`
    active le mode verbeux : `tar` vous explique ce qu'il fait pendant qu'il crée l'archive.
  - `f`
    ceci indique que la chaîne suivante de la ligne de commande est le nom du fichier à créer ou le périphérique à utiliser. Si j'utilisais `/dev/st0` ici, par exemple, cela enverrait l'écriture de la sauvegarde sur le lecteur de bandes.
- `monfichier.tar.gz`
  C'est le nom du fichier à créer.
- `/usr/local/bin`
  C'est le nom du fichier ou du répertoire à placer dans l'archive. Il est aussi possible de spécifier différents emplacements ici.

Vous trouverez de nombreux fichiers `tar.gz` (ou plus simplement `.tgz`) sur Internet. Vous pouvez les désarchiver avec la commande suivante :

```
tar -zxvf nomdufichier.tar.gz
```

Chapitre 13

Réseau

Un des très grands avantages de GNU–Linux sur les autres systèmes réside dans son support réseau. Peu de systèmes peuvent rivaliser avec les fonctionnalités présentes au sein d'un système GNU–Linux. Dans ce chapitre, nous allons vous expliquer comment configurer vos périphériques réseau.

13.1 PPP

Cette partie est une aide pour une installation rapide de PPP sur Debian. Si vous désirez aller plus loin, tournez–vous vers l'excellent PPP–HOWTO du Projet de Documentation Linux. Si vous êtes curieux ou si vous avez des besoins particuliers,le HOWTO examine ces questions en détail.

13.1.1 Introduction

Si vous vous connectez à Internet via une ligne de téléphone, vous voudrez certainement utiliser PPP (*Point–to–Point Protocol*, protocole de liaison point–à–point). C'est la méthode standard de connexion auprès des FAI (Fournisseurs d'Accès à Internet). En plus d'utiliser PPP pour vous connecter chez votre FAI, vous pourrez autoriser les connexions de l'extérieur – ce qui vous permettra de vous connecter sur une machine à distance.

13.1.2 Préparation

Configurer une liaison PPP sur un système Debian GNU–Linux est élémentaire une fois que vous possédez toutes les informations dont vous allez avoir besoin. Debian rend les choses très faciles avec ses outils simples de configuration.

Avant de vous lancer, assurez–vous de bien avoir toutes les informations de votre FAI. Cela peut inclure :

– Nom d'utilisateur ou login
– Mot de passe
– Votre adresse IP (*Internet Protocol*) fixe si nécessaire (cela doit ressembler à quelque chose comme `209.81.8.242`). Cette information n'est pas nécessaire pour la plupart des FAI.

– Le masque réseau (cela doit ressembler à quelque chose comme `255.255.255.248`). Cette information n'est pas nécessaire pour la plupart des FAI.
– Les adresses IP des serveurs DNS de votre FAI
– Toute procédure de login particulière requis chez votre FAI

Ensuite, il va falloir examiner votre configuration matérielle : la façon dont votre modem fonctionner avec GNU–Linux et sur quel port série il est connecté.

Voici une règle simple pour déterminer si votre modem va fonctionner : si c'est un « winmodem » ou bien un modem intégré, cela ne fonctionnera pas. Ces modems sont bon marché car ils ont très peu de fonctionnalités et ils demandent à l'ordinateur de compenser leur incapacité. Malheureusement, cela signifie qu'ils sont complexes à programmer et les fabricants ne donnent généralement pas leurs spécifications aux développeurs.

Si vous avez un modem avec son propre circuit embarqué ou bien un modem extérieur, vous ne devriez pas avoir de problème.

Sur les systèmes GNU–Linux, les ports séries sont référencés comme `/dev/ttyS0`, `/dev/ttys1` et ainsi de suite. Votre modem est certainement connecté soit au port 0 soit au port 1, l'équivalent du COM1 et COM2 sous Windows. Si vous ne savez pas où est connecté votre modem, lancez le programme `wvdialconf` pour essayer de le détecter (voir plus bas). Sinon, vous pouvez toujours essayer les deux et regardez lequel fonctionne...

Si vous désirez discuter avec votre modem ou vous connecter à votre FAI sans utiliser `PPP`, vous pouvez utiliser le programme `minicom`. Il se peut que vous ayez besoin d'installer le paquet `minicom` pour disposer du programme.

13.1.3 La voie facile : `wvdial`

La façon la plus facile d'obtenir une liaison `PPP` fonctionnelle est d'utiliser le programme `wvdial`. Il fait des propositions raisonnables et tente de paramétrer les choses pour vous. Si cela fonctionne, vous êtes verni. Sinon, vous aurez à faire les choses à la main.

Assurez–vous d'avoir bien installé les paquets suivants :

– `ppp`
– `ppp-pam`
– `wvdial`

Lorsque vous installez `wvdial`, on vous donne l'occasion de le configurer. Sinon, pour configurer `wvdial`, procédez comme suit en vous loguant en root à l'aide de la commande `su` :

```
touch /etc/wvdial.conf
```

`touch` va créer le fichier suivant s'il n'existe pas : le programme de configuration requiert un fichier déjà existant.

```
wvdialconf /etc/wvdial.conf
```

Cela signifie que vous êtes en train de créer le fichier de configuration `/etc/wvdial.conf`.

Répondez aux questions qui apparaissent à l'écran. `wvdialconf` va aussi rechercher votre modem et vous indiquer sur quel port série il se trouve. Inscrivez le résultat quelque part pour vous en souvenir un peu plus tard.

13.2 Ethernet

Une autre façon très populaire de se connecter à Internet est de passer via un LAN (*Local Area Network*, réseau local) qui utilise Ethernet.Cela vous donne un réseau local à haut débit en plus de la connexion à Internet. Heureusement, vous avez déjà dû configurer le réseau pendant l'installation et dès lors, il n'y rien d'autre à faire. Si jamais vous désirez modifier votre configuration, voici les fichiers intéressants :

– `/etc/init.d/network`
  Pour votre adresse IP, le masque réseau et la route par défaut.
– `/etc/hostname`
  Pour enregistrer votre nom d'hôte.
– `/etc/hosts`
  Pour enregistrer aussi votre nom d'hôte et votre adresse IP.

Chapitre 14

Installer et désinstaller des logiciels

Ce chapitre décrit les façons d'installer et de désinstaller des paquets de logiciels. Il y a plusieurs façons de le faire. Ici, nous n'allons parler de l'installation et de la désinstallation de logiciels pré–empaquetés comme les paquets Debian. L'installation à partir de sources restera à votre charge.

14.1 À quoi sert un utilitaire de maintenance de paquet

Une application ou un programme utilitaire rassemblent généralement plusieurs fichiers. Cela peut inclure des bibliothèques, des fichiers de données comme des scénarios de jeux ou des icônes, des fichiers de configurations, des pages de manuels et la documentation. Lorsque vous installez un programme, vous voulez être sûr d'avoir tous les fichiers dont vous avez besoin à la bonne place.

Vous devez aussi être capable de désinstaller un programme. Lorsque vous désinstallez, vous devez être sûr que tous les fichiers associés sont détruits. Et de plus, si les fichiers sont partagés par d'autres programmes dans le système, vous voulez être sûr de les conserver.

Enfin, vous devez être capable de faire la mise à jour d'un programme. Lorsque vous effectuez une mise à jour, vous désirez détruire les fichiers obsolètes et ajouter les nouveaux, sans ne rien casser au système existant.

Le système de gestion des paquets de Debian offre une solution à ces problèmes. Il vous autorise à installer, désinstaller et mettre à jours les paquets de logiciels, qui ne sont qu'un petit ensemble contenant les fichiers du programme et les informations qui aident l'ordinateur à gérer le paquet correctement. Les paquets Debian ont un nom se terminant par `.deb` et ils sont disponibles sur le site `FTP` ou bien sur votre cédérom officiel de Debian.

14.2 `dpkg`

La manière la plus simple d'installer un simple paquet que vous venez de télécharger est d'utiliser la commande `dpkg -i` (qui est un raccourci pour `dpkg --install`). Admettons que vous ayez téléchargé le paquet `icewm_0.8.12-1.deb` et que vous désirez l'installer. Loguez–vous d'abord

en tant que root et tapez ensuite `dpkg -i icewm_0.8.12-1.deb` et `icewm`, version 0.8.12, sera installé. Si vous avez déjà une ancienne version, `dpkg` se chargera de la mise à jour plutôt que d'installer deux versions en même temps.

Si vous désirez désinstaller un paquet, vous avez deux options. La première est la plus intuitive : `dpkg -r icewm`. Cela retirera le paquet `icewm` (`-r` est un raccourci pour `--remove` (désinstaller)). Remarquez bien qu'il est juste nécessaire de donner le nom `icewm` pour la désinstallation tandis que l'installation requiert l'extension `.deb`.

`--remove` va laisser les fichiers de configuration du paquet dans votre système. Un fichier de configuration est défini comme tout fichier que vous avez pu éditer pour personnaliser le programme du système ou vos préférences. Ainsi, si vous décidez de réinstaller un peu plus tard le programme, vous n'aurez pas à le re–personnaliser.

Mais si vous désirez réellement nettoyer les fichiers de configuration, `dpkg` propose une option `--purge`. `dpkg --purge icewm` nettoiera tous les derniers fichiers associés au paquet `icewm`.

14.3 `dselect`

`dselect` est une très bonne interface à `dpkg`. `dselect` offre une interface à base de menus à `dpkg` et est capable de sélectionner automatiquement tous les fichiers appropriés à partir d'un cédérom ou bien d'un site FTP. Pour plus de détails sur `dselect`, reportez–vous à la partie 4.20.

14.4 Compiler un logiciel

De nombreux programmes sont fournis sous forme de sources, la plupart du temps au format `tar.gz`. Premièrement, vous devez détarer le fichier `tar.gz` ; pour plus de détails sur comment faire cette opération, reportez–vous à la partie 12.3.1. Avant de compiler un paquet, vous aurez besoin d'installer `gcc`, `libc6-dev` et autres assimilés « `-dev` ». Vous les trouverez pour la plupart d'entre eux dans la section `devel` de `dselect`.

Lorsque vous aurez installé les paquets appropriés, allez via `cd` dans le répertoire que `tar` a créé pour vous. À partir de ce moment–là, vous aurez à lire les instructions d'installation. La plupart des programmes fournissent un fichier INSTALL ou README qui vous indique comment procéder.

Chapitre 15

Pour en savoir plus...

À présent, vous possédez de bonnes bases pour construire votre système GNU–Linux. Dans ce chapitre, nous allons aborder quelques notions très utiles des fonctionnalités avancées de GNU–Linux.

15.1 Les expressions rationnelles

Une *expression rationnelle* est une description d'un ensemble de caractères. On utilise cette description pour éplucher un fichier à la recherche d'un texte correspondant à cette expression rationnelle. Les expressions rationnelles sont analogues aux motifs de recherche du shell (voir partie 8.6) mais elles sont plus compliquées et beaucoup plus puissantes. elles sont à la fois plus compliquées et plus puissantes.

Une expression rationnelle est formée de texte et de *méta–caractères*. Un méta–caractère est simplement un caractère avec une signification particulière. Les méta–caractères comprennent les signes suivants :

```
. * [] - \ ^ $
```

Si une expression rationnelle ne comprend que du texte (pas de méta–caractère), cela correspond exactement au texte. Par exemple, l'expression rationnelle `mon expression rationnelle` correspond exactement au texte `mon expression rationnelle` et à rien d'autre. Les expressions rationnelles prennent habituellement en compte la casse.

Vous pouvez utiliser la commande `egrep` pour extraire les lignes d'un fichier qui contiennent une expression rationnelle. Sa syntaxe est la suivante :

```
egrep 'regexp' nomdefichier1 ...
```

Les guillemets simples ne sont pas obligatoires mais cela ne mange pas de pain de les ajouter.

Par exemple, pour trouver toutes les lignes contenant le mot GNU dans la GPL, vous tapez :

```
egrep 'GNU' /usr/doc/copyright/GPL
```

`egrep` va vous égrener les lignes correspondantes sur la sortie standard. Si vous voulez toutes les lignes contenant `freedom` (liberté) suivi d'un ensemble quelconque de caractères et suivi du mot GNU, vous pouvez taper :

```
egrep 'freedom.*GNU' /usr/doc/copyright/GPL
```

Le `.` signifie « n'importe quel caractère » et le `*` signifie « zéro ou plus de la chose précédente » soit dans ce cas « zéro ou plus de n'importe quel caractère ». Ainsi, `.*` correspond bien à n'importe quel texte. `egrep` effectue un traitement ligne par ligne donc `freedom`et `GNU` doivent se trouver sur la même ligne.

Voici un récapitulatif des méta–caractères dans expressions rationnelles :

– `.`
  Correspond à un seul caractère, à l'exception du caractère *newline* (retour chariot)
– `*`
  Correspond à zéro ou plus d'occurences de ce qui précède. Ainsi, l'expression `a*` correspond à zéro ou plus de `a` minuscule et `.*` correspond à zéro ou plus de caractères quelconques.
– `[caractères]`
  Les crochets doivent contenir un ou plusieurs caractères. L'expression complète correspond exactement à l'ensemble des caractères. Ainsi, `[abc]` correspond à un `a`, un `b` et un `c`. Cela ne correspond pas à une chaîne vide et cela ne correspond pas non plus à un caractère autre que ces trois–là.
– `^`
  Impose la correspondance en début de ligne. L'expression `^The` ne correspond à `The` que si elle apparaît en début de ligne : il ne peut y avoir d'espace, ou de texte, avant le `The`. Si vous désirez autoriser des espaces, vous pouvez autoriser 0 ou plus d'espaces avec `^ *The`.
– `$`
  Impose la correspondance en fin de ligne. `fin$` requiert que le texte `fin` soit placé en fin de ligne sans espace ni texte à la fin.
– `[^caractères]`
  C'est la négation de la liste des caractères entre crochets. `[^abc]` correspond à tout caractère simple, à l'exception de a, b ou c.
– `[caractères-caractères]`
  Vous pouvez inclure un intervalle dans une liste de caractères entre crochets. Pour faire la correspondance avec toutes les lettres minuscules, utilisez `[a-z]`. Vous pouvez multiplier les intervalles : ainsi, pour avoir les trois premières lettres de l'alphabet et les trois dernières, essayez `[a-cx-z]`. Pour avoir toutes les lettres dans toutes les casses, essayez `[a-zA-Z]`. Vous pouvez mélanger les intervalles avec des caractères simples ou bien le méta–caractère `^` : `[^a-zBZ]` signifie « tout sauf une lettre minuscule, un B ou ou un Z ».
– `()`
  Vous pouvez utiliser les parenthèses pour regrouper les expressions rationnelles, comme dans des expressions mathématiques.
– `||`
  signifie « ou ». Vous pouvez vous en servir pour fournir une série d'expressions alternatives. On met généralement ces expressions entre parenthèses comme ceci : `c(ad|ab|at)` correspond à `cad`, `cab` ou `cat`. Sans les parenthèses, cela aurait correspondu à `cad` ou bien `ab` ou bien `at`.
– `\`

Protège tout caractère spécial. Si vous désirez trouver un `*`, vous devez taper `\*`. Le `\` signifie d'ignorer le sens usuel de `*`.

Voici une série d'exemples concrets pour vous aider.

- `c.pe` correspond à `cope, cape, caper`
- `c\.pe` correspond à `c.pe, c.per`
- `sto*p` correspond à `stp, stop, stoop`
- `car.*n` correspond à `carton, cartoon, carmen`
- `xyz.*` correspond à `xyz` et tout ce qui s'ensuit. Certains outils, comme `egrep`, ne font la correspondance que jusqu'à la fin de la ligne.
- `^The` correspond à `The` en début de ligne
- `atime$` correspond à `atime` en fin de ligne
- `^Only` correspond à `Only` en début de ligne – pas d'espace, de caractère, rien. Seul `Only` est accepté.
- `b[aou]rn` correspond à `barn, born, burn`
- `ver[D-F]` correspond à `verD, verE, verF`
- `ver[^0-9]` correspond à `Ver` suivi de tout sauf d'un chiffre.
- `the[ir][re]` correspond à `thier, therr, there, theie`
- `[A-Za-z][A-Za-z]*` correspond à tous les mots composés seulement de lettres et d'au moins une lettre. Cela ne peut pas contenir des chiffres ou des espaces.

15.2 Notions avancées sur les fichiers

Maintenant que vous avez compris les bases sur les fichiers, il est temps d'apprendre plus de choses à leur sujet.

15.2.1 La véritable nature des fichiers : les liens physiques et les i–noeuds

Chaque fichier de votre système est représenté par un *i–noeud* (en anglais, *inode* pour *Information Node*, prononcez « aïe–node », noeud d'information). un i–noeud contient toutes les informations du fichier. Cependant, l'i–noeud n'est pas directement visible. De plus, chaque i–noeud est lié au système de fichiers par un ou plusieurs liens physiques. Les liens physiques contiennent le nom du fichier et le numéro d'i–noeud. L'i–noeud contient le fichier en lui–même, c'est–à–dire, l'emplacement où l'information est stockée sur le disque, les permissions d'accès, le type de fichier et ainsi de suite. Le système peut trouver n'importe quelle i–noeud s'il a son numéro.

Un simple fichier peut avoir plus d'un lien physique. Cela signifie que tous ces fichiers multiples correspondent au même fichier (c'est–à–dire qu'ils sont associés au même numéro d'i–noeud). Cependant, vous ne pouvez faire de lien physique entre des systèmes de fichiers : tous les liens physiques à un fichier particulier (i–noeud) doivent être dans le même système de fichiers. C'est parce que chaque système de fichiers possède sa propre table d'i–noeuds et que l'on ne peut dupliquer des numéros d'i–noeud sur différents systèmes de fichiers.

Parce que tous les liens physiques à une certaine i–noeud se réfèrent au *même fichier*, vous pouvez modifier le fichier en l'appelant par un nom et voir ensuite les modifications en l'appelant par un des autres noms. Essayez ceci :

```
cd; echo "Bonjour" > premierlien
```

Vous allez dans votre répertoire personnel et vous créez un premier fichier appelé `premierlien` contenant le mot « Bonjour ». Ce que vous venez de faire est juste de rediriger la sortie de `echo` (`echo` renvoie juste ce que vous lui donnez) en la plaçant dans `premierlien`. Allez voir le chapitre sur le shell pour de plus amples explications.

```
cat premierlien
```

Pour vérifier le contenu de `premierlien`.

```
ln premierlien secondlien
```

Crée un lien physique : `secondlien` pointe maintenant sur le même i–noeud que `premierlien`.

```
cat secondlien
```

Pour confirmer que `secondlien` est bien identique à `premierlien`.

```
ls -l
```

Remarquez bien que le nombre de liens physiques affiché pour `premierlien` et `secondlien` est 2.

```
echo "change" >> secondlien
```

C'est une nouvelle redirection du shell (ne vous souciez pas des détails). Vous avez ajouté le mot « change » à `secondlien`. Confirmez–le avec `cat secondlien`.

```
cat premierlien
```

`premierlien` a aussi le mot « change » ! C'est parce que `premierlien` et `secondlien` se réfèrent au *même fichier*. Peu importe comment vous l'appelez lorsque vous le modifiez.

```
chmod a+rwx premierlien
```

Cela modifie les permissions de `premierlien`. Entrez la commande `ls -l` pour vérifier que les permissions de `secondlien` ont aussi été modifiées. Cela signifie que les informations de permission résident dans l'i–noeud, et non dans le lien.

```
rm premierlien
```

Cela détruit le lien. C'est une astuce de `rm`. Cela détruit en réalité le lien et non le fichier. Maintenant, tapez `ls -l` et remarquez que `secondlien` est toujours là. Notez bien aussi que le nombre de liens est passé de 2 à 1.

```
rm secondlien
```

Cela détruit le second lien. Quand il ne reste plus de lien rattaché à un fichier, Linux détruit le fichier lui–même c'est–à–dire son i–noeud.

Tous les fichiers fonctionnent ainsi, et même les fichiers spéciaux comme les périphériques (comme `/dev/hda`).

Un répertoire n'est qu'une liste de nom de fichiers et de numéros d'i–noeuds, c'est–à–dire une liste de liens physiques. Lorsque vous créez un lien physique, vous ajouter seulement une association numéro–nom à un répertoire. Lorsque vous détruisez un fichier, vous détruisez juste un lien physique d'un répertoire.

15.2.2 Types de fichiers

Nous avons dissimulé jusqu'à présent le fait que le noyau Linux considère presque tout comme un fichier. Cela inclut les répertoires et les périphériques. Ce sont juste des genres de fichiers un peu spéciaux.

Comme vous devez vous le rappelez, le premier caractère dans `ls -l` renvoie le type du fichier. Pour un fichier ordinaire, cela ne renvoie que − tout simplement. Les autres possibilités sont les suivantes :

– `d` : répertoire
– `l` : lien symbolique
– `b` : périphérique bloc
– `c` : périphérique caractère
– `p` : tube nommé
– `s` : socket

Liens symboliques

Les liens symboliques (aussi appelés *symlink* ou liens logiciels) sont l'autre type de lien avec les liens physiques. Un lien symbolique est un fichier spécial qui « pointe vers » un lien physique de n'importe quel système de fichiers monté. Lorsque vous tentez de lire le contenu d'un lien symbolique, cela vous renvoie le contenu du fichier pointé plutôt que le contenu du lien en lui–même. Puisque les répertoires, les périphériques et autres liens symboliques sont des fichiers, vous pouvez pointer un lien symbolique sur chacun de ces objets.

Pour résumé, un lien physique est un nom de fichier et son numéro d'i–noeud. Un fichier est réellement un i–noeud : un emplacement sur un disque, le type du fichier, les modes de permission, etc...Un lien symbolique est un i–noeud qui contient le nom du lien physique. Un lien symbolique appareille deux noms de fichiers ensemble tandis qu'un lien physique appareille un nom de fichier avec un numéro d'i–noeud.

Tous les liens physiques d'un même fichier ont le même état. Cela veut dire qu'il n'y en a pas un de privilégié. Si vous effectuez une opération sur l'un, c'est comme la faire sur n'importe lequel d'entre eux. C'est parce que le lien physique réfère au même i–noeud. Les opérations sur les liens symboliques, pour leurs parts, affectent parfois le propre i–noeud du lien (celui contenant le nom du lien physique) et parfois affectent le lien physique sur lequel ils pointent.

Il y a un certain nombre de différences importantes entre les liens symboliques et les liens physiques.

Les liens symboliques peuvent traverser des systèmes de fichiers. C'est parce qu'ils contiennent les noms complets des fichier, en partant d'un répertoire racine et que les noms complets de fichiers sont

uniques. Puisque les liens physiques pointent vers un numéro d'i–noeud et que les numéros d'i–noeuds sont uniques qu'à l'intérieur d'un système de fichiers, ils deviendraient impossible de les qualifier avec certitude si l'on ne connaissait pas le système de fichiers.

Vous pouvez faire des liens symboliques vers des répertoires mais cela n'est pas possible pour les liens physiques. Chaque répertoire a des liens physiques – son référencement dans son répertoire parent, son entrée `.` et l'entrée `..` dans chacun de ses sous–répertoires – mais pour pouvoir ordonner le système de fichiers, aucun autre lien physique n'est autorisé. En conséquence de quoi, le nombre de fichiers d'un répertoire est égal au nombre de liens physiques du répertoire moins deux (vous retirez le nom du répertoire et le lien `.`).

Pour comparer un lien physique et un lien symbolique : vous pouvez uniquement faire un lien physique sur un fichier existant car il doit exister un numéro d'i–noeud sur lequel se référer. Par contre, vous pouvez faire un lien symbolique sur n'importe quel fichier, qu'il existe +réellement ou pas.

Détruire un lien symbolique détruit seulement le lien. Cela n'a pas d'impact sur le fichier anciennement pointé. Détruire le seul lien physique d'un fichier conduit à détruire le fichier.

Essayez ceci :

```
cd; ln -s /tmp/moi MonTmp
```

Vous vous positionnez dans votre répertoire personnel. `ls` avec l'option `-s` fabrique un lien symbolique et dans ce cas, ce lien s'appelle `MonTmp` et pointe sur le fichier `/tmp/moi`.

```
ls -l MonTmp
```

Vous devriez observer une sortie du genre :

```
lrxwrxwrxw 1 havoc havoc 7 Dec 6 12:50 MyTmp -> /tmp/me
```

La date ainsi que l'utilisateur et le groupe devrait être différents pour vous. Notez le type du fichier : `l`. Cela indique un lien symbolique. Regardez aussi les permissions. Les liens symboliques ont toujours ces permissions. Si vous tentez de changer les permissions d'un fichier symbolique via `chmod`, vous allez réellement changer les permissions du fichier sur lequel il pointe.

```
chmod 700 MonTmp
```

Vous devriez obtenir un `No such file or directory` (pas trouvé de fichier ou de répertoire portant ce nom) car le fichier `/tmp/moi` n'existe pas. Remarquez que cela n'a pas empêché la création du lien symbolique.

```
mkdir /tmp/moi
```

Cela crée le répertoire `/tmp/moi`.

```
chmod 700 MonTmp
```

devrait maintenant fonctionner sans problème.

```
touch MonTmp/monfichier
```

Pour créer un fichier dans `MonTmp`.

```
ls /tmp/moi
```

Le fichier a vraiment été créé dans `/tmp/moi`.

```
rm MonTmp
```

Détruit le lien symbolique. Remarquez que cela ne retire que le lien et non ce vers quoi il pointe. Ensuite, vous devez utiliser `rm` et non pas `rmdir` :

```
rm /tmp/moi/monfichier; rmdir /tmp/moi
```

Cela nettoie tout ce qu'il reste.

Fichiers périphériques

Les fichiers périphériques se réfèrent à un périphérique physique ou virtuel de votre système, comme votre disque dur, votre carte vidéo, votre écran et votre clavier. Un exemple de périphérique virtuel est la console, représentée par `/dev/console`.

Il y a deux sortes de périphériques : bloc et caractère. On peut avoir accès aux *périphériques caractères* caractère par caractère. Rappelez–vous que la plus petite unité de données qui peut être écrite ou bien être lue à partir du périphérique est un caractère (octet).

On peut accéder aux *périphériques bloc* avec des unités plus grandes appelées blocs, qui contiennent un certain nombre de caractères. Votre disque dur est un périphérique bloc.

Vous pouvez lire ou écrire dans des fichiers périphériques comme vous le feriez avec n'importe quelle autre sorte de fichiers, quoique le fichier puisse contenir alors un étrange baragouin incompréhensible à nous autres humains. Écrire n'importe quoi dans ces fichiers n'est pas une bonne idée. Parfois si. Par exemple, vous pouvez envoyer un fichier PostScript dans le périphérique de l'imprimante `/dev/lp0` ou bien envoyer des commandes de modem au fichier périphérique idoine à partir du bon port série.

Il existe un fichier particulier appelé `/dev/null`. Il détruit tout ce qu'il reçoit. Si vous ne voulez pas quelque chose, envoyez–le dans `/dev/null`. C'est essentiellement un puits sans fond. Si vous tentez de lire `/dev/null/`, vous obtiendrez un *end–of–file* (EOF, caractère de fin de fichier) immédiatement. `/dev/zero` est similaire, à l'exception du fait que sa lecture vous renverra le caractère `\0` (pas le même que le chiffre 0).

Tubes nommés (FIFO)

Un tube nommé est un fichier qui agit comme un tube. Vous entrez quelque chose dans le fichier et cela ressort. Ceci est appelé une FIFO (*First–In–First–Out*, premier entré, premier sorti) car la première chose que vous entrez est aussi la première chose que l'on sort.

Si vous écrivez dans un tube nommé, le processus d'écriture ne s'arrête pas tant qu'il y a quelque chose à lire dans le tube. Si vous lisez dans un tube nommé, le processus de lecture attend qu'il y ait quelque chose à lire avant de s'arrêter. La taille du tube est toujours zéro. Il ne stocke pas de donnée : il lie

simplement deux processus à travers le signe « | » du shell. Dans tous les cas, puisque le tube a un nom, les deux processus n'ont pas à appartenir à la même ligne de commandes ou même à être lancé par le même utilisateur.

Vous pouvez essayer ceci :

```
cd; mkfifo montube
```

Activez alors le tube :

```
echo "bonjour" > montube &
```

Cela place le processus qui essaie d'écrire « bonjour » dans le tube en arrière–plan. Remarquez bien que le processus ne rend pas la main : il attend que quelqu'un vienne lire dans le tube :

```
cat montube
```

À ce moment, le processus `echo` s'achève car `cat` a lu le tube et le processus `cat` a affiché `bonjour`.

```
rm montube
```

Vous pouvez détruire les tubes comme n'importe quel autre fichier.

Les sockets

Les sockets sont similaires aux tubes mais elles ne fonctionnent qu'à travers un réseau. C'est de cette façon que votre ordinateur travaille en réseau. Vous avez peut–être entendu parler de « WinSock », qui sont les sockets de Windows.

Nous n'irons pas plus loin sur ce sujet car vous n'aurez pas l'occasion de vous en servir, à moins de programmer. En tout cas, si vous apercevez un fichier marqué avec un type `s` sur votre ordinateur, vous saurez de quoi il s'agit.

15.2.3 Le système de fichiers `proc`

Le noyau Linux fournit un système de fichiers particulier qui est monté sous `/proc` dans les systèmes Debian. C'est ce que l'on appelle un « pseudo–système de fichiers » parce qu'il n'existe pas réellement sur vos périphériques physiques.

Le système de fichiers `proc` contient des informations sur le système et sur les processus en cours. Certains des « fichiers » de `proc` sont raisonnablement compréhensibles par un être humain (essayez `cat /proc/meminfo` et `cat /proc/cpuinfo`). Les autres fichiers sont un amas incompréhensible de chiffres. Les utilitaires systèmes traduisent les informations contenues dans ces chiffres et vous les présentent de façon plus compréhensible.

Les gens paniquent souvent lorsqu'ils remarquent qu'un fichier en particulier – `/proc/kcore` – est souvent énorme, C'est, à un cheval près, une copie du contenu de la mémoire de votre ordinateur. Il est utilisé pour déboguer le noyau. Il n'existe nulle part physiquement donc ne vous préoccupez pas de sa taille.

Si vous désirez en savoir plus sur `/proc`, tapez `man 5 proc`.

15.2.4 Copie à grande échelle

Il est parfois nécessaire de recopier un répertoire dans un autre emplacement. Peut–être avez–vous ajouté un nouveau disque dur et vous désirez faire une copie de `/usr/local` dessus. Il y a plusieurs façons de procéder :

La première est d'utiliser la commande `cp`. La commande `cp -a` demande à `cp` de préserver le maximum d'informations. Vous pourriez ainsi utiliser la commande suivante :

```
cp -a /usr/local /destination
```

Cependant, il existe quelques cas qui ne fonctionnent pas[1]. Dès–lors, la meilleure façon de procéder lors d'une grande copie est d'enchaîner deux commandes `tar` consécutives, comme ceci :

```
tar -cSpf - /usr/local | tar -xvSpf -C /destination
```

La première commande `tar` archive le répertoire et l'entube dans le second. La seconde commande déploie l'archive dans l'emplacement spécifié avec l'option `-C`.

15.3 Sécurité

Dans la partie 9.1, nous avons parlé des permissions sur les fichiers dans Linux. C'est une façon fondamentale de sécuriser votre système. Si vous faîtes tourner un système multi–utilisateurs ou un serveur, il est primordial d'établir des bonnes permissions. Une bonne règle de base consiste à configurer les fichiers avec les permissions les plus basses autorisant leur bon usage.

Si vous faîtes tourner un serveur en réseau, il y a quelques petites bricoles auxquelles il faut faire attention. Premièrement, vous devez désinstaller ou arrêter les services que vous n'utilisez pas. Un bon point de départ est le fichier `/ect/inetd.conf`. Vous pouvez probablement désactiver quelques services. Pour la plupart des services réseau, vous pouvez contrôler qui y a accès. Les fichiers `/etc/host.allow` et `/etc/hosts.deny` (documentés dans `man 5 hosts_access`) peuvent contrôler qui a accès à quel service. Vous devez aussi être à jour des rustines (*patches*) ou des mises–à–jour proposées par Debian. Elles sont disponibles dans votre miroir FTP de Debian.

Les règles suivantes de bon sens doivent aussi s'appliquer :

- Ne révélez à personne votre mot de passe.
- N'envoyez jamais votre mot de passe en clair à travers Internet en utilisant `telnet` ou `ftp`. Utilisez des protocoles de cryptage ou évitez de vous loguer à distance.
- Évitez d'utiliser le compte root le plus possible
- N'installez pas de logiciels non sûrs et ne les installez par en tant que root
- Évitez de laisser des accès libre en écriture dans la mesure du possible. `/tmp` est une exception à cette règle.

Bien que cela soit de peu d'utilité pour quelqu'un qui ne fait pas tourner un serveur, cela vaut la peine de se renseigner sur les questions de sécurité. Le mécanisme de sécurité de Debian vous protège de la plupart des virus.

[1] comme les fichiers morcelés et les liens physiques par exemple

15.4 Le développement de logiciels avec Debian

Debian est une plateforme privilégiée pour le développement de logiciels et la programmation. Voici une liste des langages et pseudo–langages supportés : C, C++, Objective–C, Perl, Python, m4, Ada, Pascal, Java, awk, Tcl/Tk, SQL, assembleur, Bourne shell, csh et bien d'autres encore. L'écriture de programmes est hors du propos de ce livre mais voici les programmes de développement parmi les plus populaires que l'on trouve sur Debian :

- `gcc`
  Le compilateur C de GNU : un compilateur moderne et optimisé de C.
- `g++`
  Le compilateur C++ extrait de `gcc`
- `cpp`
  Le pré–processeur de C de `gcc`
- `perl`
  L'interpréteur Perl. Perl est un super langage multi–usages.
- `gdb`
  Le débogueur GNU. Il est utilisé pour déboguer de nombreux langages.
- `gprof`
  Extraction de profile. Ce programme vous aide à optimiser votre programme pour augmenter ses performances.
- `emacs`
  GNU Emacs est un éditeur et un environnement intégré de travail.
- `as`
  C'est l'assembleur de GNU.

Chapitre 16

Lire la documentation et obtenir de l'aide

16.1 Les sources de documentation

Sur un système Debian, vous pouvez trouver la documentation aux emplacements suivants :

- Pages de manuel, que l'on peut lire avec la commande `man`
- Pages info, que l'on peut lire avec la commande `info`
- Les répertoires de `/usr/doc/paquets` où *paquet* est le nom du paquet Debian[1]
- Le répertoire `/usr/doc/HOWTO` contient les HOWTOs du Projet Linux de Documentation, si vous avez installé les paquets les contenant.
- La plupart des commandes ont une option `-h` ou `--help`. Tapez le nom de la commande suivi d'une de ces options pour essayer.
- Le Projet de Documentation de Debian (*http ://www.debian.org/~elphick/ddp/*) fournit aussi quelques manuels.
- La page d'assistance de Debian (*http ://www.debian.org/support/*) possède sa FAQ et quelques autres documents. Vous pouvez aussi essayer le site oueb de Linux (*http ://www/linux.org*).

Cette déroutante profusion de sources de documentations existe pour des tas de raisons. Par exemple, `info` est supposé remplacer le `man` mais `man` n'a toujours pas disparu à ce jour. Mais dans tous les cas, il est rassurant de savoir que tant de documentations existent !

Ainsi, où chercher pour obtenir de l'aide ? Voici quelques suggestions :

- Utilisez la page info ou l'option `--help` pour avoir un résumé rapide de la syntaxe et des options de la commande. Utilisez aussi la page de manuel si le programme n'a pas de page info.
- Utilisez la page info si le programme possède une documentation à ce format.
- Si cela ne donne rien, allez voir dans `/usr/doc/nomdepaquet`
- Utilisez les HOWTOs pour régler une chose particulière ou chercher une information sur un matériel particulier. Par exemple, l'`Eternet-HOWTO` regorge d'informations sur les cartes Ethernet et le `PPP-HOWTO` explique en détail comment configurer une liaison `ppp`.
- Utilisez les manuels du Projet de Documentation Debian pour les concepts inhérents et spécifiques à Debian.

[1] `zless` est très pratique pour lire les fichiers de `/usr/doc`. Voir chapitre 10.1 pour plus de détails.

– Si tout cela ne donne rien, demandez de l'aide à quelqu'un. Voir la partie 16.4 pour cela.

Comment utiliser les pages de manuel est expliqué à la partie 7.1. C'est très simple : appuyez sur la touche espace pour aller à la page suivante et appuyez sur `q` pour quitter le manuel. Utiliser `info`, visualiser les fichiers de `/usr/doc` et demander de l'aide à quelqu'un font l'objet des parties suivantes du chapitre.

16.2 Utiliser `info`

`info` est le visualiseur de documentation de GNU. Certains programmes fournissent leur documentation au format `info` et vous pouvez utiliser `info` pour la lire. Vous pouvez lancer le visualiseur en tapant simplement `info` ou en lui passant un argument comme ceci :

```
info emacs
```

Vous pouvez obtenir aussi des informations sur `info` lui–même en appelant le tutoriel interne :

```
info info
```

Maintenant, vous pouvez vous déplacer à l'aide des touches suivantes :

– `flèches`
  Pour déplacer le curseur au sein du document.
– `m RET`
  Sélectionne le choix d'un menu sous le curseur.
– `u`
  Pour « remonter » dans le document.
– `n`
  Pour aller à la page suivante.
– `p`
  Pour aller à la page précédente.
– `s`
  Pour rechercher quelque chose.
– `g`
  Pour aller à une page spécifique.
– `q`
  Quitter `info`

Vous remarquerez que la ligne supérieure de l'écran indique les pages *next* (suivantes), *previous* (précédentes) et *up* (supérieures) qui correspondent aux actions des touches `n`, `p` et `u`.

16.3 HOWTO

En plus de ses livres, le Projet de Documentation Linux a diffusé une série de petits documents décrivant comment configurer des aspects particuliers de GNU–Linux. Par exemple, le SCSI–HOWTO décrit les

complications de l'utilisation du SCSI – une façon standard de communiquer avec des périphériques – sous GNU–Linux. De façon générale, les HOWTOs ont plus de détails techniques sur des configurations matérielles et sont souvent plus à jour que ce manuel.

Il y a des paquets Debian pour les HOWTOs. `doc-linux-text` contient l'ensemble des HOWTOs au format texte et `doc-linux-html` les contient (surprise !) au format html pour naviguer. Remarquez bien que Debian fournit des traductions de ces HOWTOs si l'anglais n'est pas votre langue maternelle. Debian a des paquets en allemand, français, espagnol, italien, japonais, coréen, polonais, suédois et chinois des HOWTOs. On les trouve généralement dans les paquets `doc-linux-`*`languagecode`* où *languagecode* est `fr` pour le français, `es` pour l'espagnol, etc...Si vous les avez installés, vous les trouverez dans `/usr/doc/HOWTO`. Quoi qu'il en soit, vous trouverez toujours les toutes dernières versions sur le réseau sur la page du LDP :

```
http://metalab.unc.edu/LDP/
```

16.4 Aide personnalisée

L'emplacement correct[2] pour demander de l'aide sur Debian est la liste de distribution des utilisateurs de Debian à `debian-user@list.debian.org` (envoyer un courriel à `debian-user@list.debian.org`). Si vous savez vous servir d'IRC, vous pouvez utiliser le canal `#debian` sur `irc.debian.org`. Vous pouvez trouver de l'aide sur GNU–Linux en général dans la hiérarchie USENET `comp.os.linux.*`. Il est aussi possible d'embaucher des consultants pour fournir un service d'assistance garantie. Le site oueb de Debian (*http ://www.debian.org*) fournit d'autres informations sur beaucoup de ces ressources.

Une nouvelle fois, ne contactez pas les auteurs (et le traducteur !) de ce livre pour obtenir de l'aide. Nous ne serons de toute façon certainement pas capables de répondre à votre problème spécifique. Si vous vous adressez à `debian-user`, vous recevrez quantité de messages de qualité, et qui plus est rapidement.

Restez toujours courtois et faîtes l'effort de lire la documentation. Rappelez–vous bien que Debian résulte d'un effort collectif de volontaires et que les personnes vous font une faveur en vous donnant un peu de leur temps pour vous aider. Beaucoup d'entre eux font payer ce même service plusieurs centaines de dollars pendant la journée.

Ce qu'il faut faire pour poser des questions

- Lire tout d'abord la documentation de base en premier. Les trucs comme les options des commandes et ce que fait une commande en fait partie. Cela inclut aussi les pages de manuel et la documentation au format info.
- Vérifier les documents HOWTOs si votre question concerne un réglage comme une connexion `ppp` ou `Ethernet`.
- Essayez d'être sûr que la réponse ne se trouve pas dans ce livre.
- N'ayez pas peur de demander, après avoir fait l'effort d'essayer de comprendre.
- N'ayez pas peur de demander des explications conceptuelles, des conseils sur tous les aspects que l'on ne trouve pas dans les documentations.

[2] Les francophones ont leur propre liste avec l'avantage du support de leur langue maternelle : debian–french@lists.debian.org

– Joignez toute information intéressante. Vous devez toujours mentionner la version de Debian que vous utilisez. Vous pouvez aussi mentionner le numéro de version de tout paquet pertinent. La commande `dpkg -l nomdupaquet` vous le fournit. Vous pouvez aussi signaler ce que vous avez tenté et ce qu'il est advenu. N'oubliez d'inclure les messages d'erreur complets et exacts, s'il y en a.
– Ne vous excusez pas d'être un nouveau sur Linux. Il n'y a pas de raison d'être un expert pour pouvoir utiliser GNU–Linux, pas plus qu'il est nécessaire d'être une mécanique pour conduire une automobile.
– Ne postez pas ou n'envoyez pas de courriel en HTML. Certaines versions de Netscape ou d'Internet Explorer postent par défaut en HTML plutôt qu'en texte. Beaucoup de gens vont carrément alors ignorer ces courriels car de nombreux lecteurs de courriels ont des difficultés à lire ce format. Il y a une préférence quelque part pour désactiver le support HTML.
– Soyez courtois. Rappelez–vous que Debian est un effort collectif de bénévoles et quiconque vous aide vous donne une partie de son temps libre.
– Renvoyez votre question à la liste si vous n'avez pas eu de réponse après quelques jours. Il y avait peut–être de nombreux messages et il est passé inaperçu. Ou peut–être que personne ne connaissait la réponse et si personne n'est capable de répondre la fois suivante, il s'agit d'une sacrée colle. Vous pourriez peut–être fournir plus d'informations la fois suivante.

16.5 Extraire des informations du système

Lorsque vous diagnostiquez un problème ou lorsque vous demandez de l'aide, vous aurez besoin d'extraire des informations de votre système. Voici quelques façons d'y parvenir :

– Examinez les fichiers placés dans `/var/log`
– Examinez la sortie du message de la commande `dmesg`
– Lancez la commande `uname -a`

Chapitre 17

Problèmes

Dans Debian, comme partout ailleurs, les choses ne fonctionnent pas toujours comme vous vous y attendez ou comme vous voulez qu'elles le fassent. Bien que Debian ait une solide réputation d'être robuste et stable, ses réactions à vos commandes peuvent parfois être inattendues. Ici, nous allons tenter de lever le voile sur des problèmes courants très fréquemment rencontrés.

17.1 Difficultés classiques

Cette partie a pour but de fournir quelques solutions aux difficultés les plus fréquemment rencontrées à l'usage par les utilisateurs.

17.1.1 Travailler avec des noms de fichiers étranges

Vous pourrez parfois accidentellement créer un fichier dont le nom comporte un caractère normalement absent d'un nom de fichier. Les exemples sont l'inclusion d'espace, un trait d'union placé au début du nom ou tout simplement des guillemets. Il se peut que l'accès, la destruction ou le renommage de ces fichiers vous posent des difficultés.

Voici quelques trucs pour vous aider :

– Essayez d'entourer le nom de fichier d'apostrophes simples, comme ceci :
  `less 'Fichier avec espaces.txt'`
– Insérez un `./` avant le nom de fichier :
  `less './-un étrange fichier.txt'`
– Utilisez des motifs :
  `less Fichier?avec?espaces.txt`
– Utilisez un contre–slash devant chaque caractère inhabituel
  `less Fichier\ avec\ espaces.txt`

17.1.2 Imprimer

Une source d'ennui commune est le système d'impression sous Debian. Traditionnellement, l'impression a toujours été un aspect puissant mais complexe d'Unix. Cependant, Debian la rend plus simple. La façon la plus simple est d'utiliser le paquet `magicfilter`. `magicfilter` va vous poser une série de questions au sujet de votre imprimante et la configurer pour vous. Si vous avez des soucis d'impression, essayez `magicfilter`.

17.1.3 Les problème sous X

De nombreuses questions tournent autour de X. Voici quelques trucs si vous avez quelques difficultés à paramétrer le système X Window.

- Pour les problèmes de souris, lancez `XF86Setup` et essayez les options PS/2, Microsoft, MouseSystems et Logitech. La plupart des souris devraient correspondre à l'une d'entre elles. Le périphérique pour les souris est `/dev/psaux` pour celle PS/2 et un port série comme `/dev/ttyS0` pour les souris série.
- Si vous ne connaissez pas votre chipset vidéo, lancez `SuperProbe` : cela permet souvent de le déterminer pour vous.
- Si votre écran ne possède pas beaucoup de couleurs, essayez de sélectionner une autre carte ou indiquez à X quelle quantité de mémoire vidéo vous possédez.
- Si votre écran reste blanc, ou bien si vous avez un texte illisible en lançant X, vous avez probablement sélectionné un taux de rafraîchissement incorrect. Retournez dans `XF86Setup` ou `xf86config` et revérifiez ces réglages.
- `xvidtune` peut vous aider à régler l'image si elle est trop à droite ou trop à gauche ou bien si elle est trop haute ou trop basse.
- `xdpyinfo` peut vous donner des informations sur une session X en cours.
- `XF86Setup` peut vous régler votre profondeur de couleurs par défaut.
- Vous pouvez sélectionner votre gestionnaire de fenêtres par défaut en modifiant `/etc/X11/window-managers`.
- `/var/log/xdm-errors` peut contenir des informations utiles si vous avez des problèmes en lançant `xdm`.

17.2 Problème lors de la procédure de boot

Si vous avez des problèmes durant la phase de boot, comme un blocage du noyau pendant la procédure d'amorçage, ou que le noyau ne reconnaisse pas les périphériques que vous avez ou encore que les lecteurs ne soient reconnus proprement, la première chose à faire est de vérifier les paramètres de boot. On les obtient en pressant sur la touche `F1` lors du boot sur la disquette de secours.

La plupart du temps, les problèmes se résolvent en supprimant quelques périphériques et en bootant à nouveau. Les modems internes, les cartes son et les périphériques *Plug–n–Play* sont souvent problématiques.

Tecras, et quelques autres portables, et certains non portables n'arrivent pas à vider le cache lorsqu'ils passent en A20. Ceci est provoqué par les noyaux bzImage et non par les noyaux zImages. Si votre or-

dinateur souffre de ce problème, vous verrez un message lors du boot disant `saying A200 gating failed`. Dans ce cas, il est nécessaire d'utiliser les images de boot « tecra ».

Si vous avez toujours des problèmes, envoyez un rapport de bogue. Envoyez un courriel à

```
submit@bugs.debian.org
```

Vous DEVEZ inclure les lignes suivantes en en–tête de votre courriel :

```
Package: boot-floppies
Version: version
```

Assurez–vous que la *version* que vous fournissez est celle que vous avez utilisé avec les disquettes de boot. Si vous ne connaissez pas la version exacte, fournissez la date à laquelle vous avez téléchargé les disquettes, et ajoutez à partir de quelle distribution vous les avez obtenues : « stable » ou « frozen ».

Vous devez aussi inclure les informations suivantes dans votre rapport de bogues :

– `architecture` : i386
– `model` : votre matériel principal et le modèle
– `memory` : la quantité de RAM
– `SCSI` : les adapteurs, s'il y en a.
– `cd-rom` : type de cédérom et son interface (ATAPI)
– `network card` : votre carte réseau, si vous en avez une.
– `pcmcia` : le détail de vos périphériques PCMCIA

Cela dépend de la nature du bogue mais il peut être pertinent de donner le modèle du disque dur, la capacité du disque et le modèle de carte vidéo.

Dans le rapport de bogue, décrivez le problème et joignez les derniers messages visibles du noyau dans le cas d'un blocage du noyau. Décrivez les étapes qui vous ont amenées dans cet état.

Chapitre 18

Booter le système

Cette partie montre ce qui se passe lors du boot de GNU–Linux.

La façon dont vous amorcez votre système dépend de la façon dont vous l'avez configuré à l'installation. Dans le meilleur des cas, vous n'avez qu'à allumer votre ordinateur. Mais il se peut que vous deviez insérer une disquette.

Linux est appelé par un programme appelé LILO (LInux LOader, littéralement « chargeur de linux »). LILO peut aussi charger un autre système d'exploitation et vous demander sur quel système vous désirez booter.

La première chose qui se passe lorsque vous allumez un PC de type Intel est l'exécution du BIOS. BIOS signifie *Basic Input Output System* (système d'entrée–sortie simple). C'est un programme résidant perpétuellement sur l'ordinateur et stocké en lecture seule sur un composant. Il effectue quelques tests basiques et ensuite recherche une disquette dans le premier lecteur de disquettes. S'il la trouve, il recherche un secteur d'amorce (*boot sector*) et lance le code se trouvant dessus s'il existe. S'il y a une disquette sans secteur d'amorce, le BIOS va vous afficher un message du type : `Non-system disk` ou `disk error`[1]. Retirez alors la disquette et appuyez sur une touche pour relancer la procédure de boot.

S'il n'y a pas de disquette dans le lecteur, le BIOS recherche un MBR (*Master Boot Record*) sur le disque dur. Il lance le code exécutable trouvé dessus qui charge le système d'exploitation. Sur des systèmes GNU–Linux, LILO peut occuper le MBR et charger GNU–Linux.

Ainsi, si vous avez décidé d'installer LILO sur votre disque dur, vous devriez apercevoir LILO au démarrage de votre ordinateur. À partir de ce moment–là, vous pouvez appuyer sur la touche *shift* pour sélectionner quel système d'exploitation charger et appuyez sur *Tab* pour afficher la liste des options. Tapez sur une de ces options et appuyez sur la touche *Entrée*. LILO va charger le système demandé.

Si vous n'appuyez pas sur la touche *shift*, LILO chargera automatiquement le système par défaut au bout de cinq secondes. Si vous le désirez, vous pouvez changer le système par défaut, le nombre de systèmes et le temps d'attente avant le chargement.

Si vous n'avez pas installé LILO sur votre disque dur, vous avez probablement créé une *disquette de boot*. La disquette de boot va avoir LILO dessus. Tout ce que vous avez à faire est d'installer la disquette

[1] *Disque non–système* et *mauvaise disquette*

dans son lecteur et rallumer l'ordinateur ; le BIOS va la trouver avant de vérifier le MBR du disque. Pour retourner à un système non–linux, sortez la disquette et redémarrez l'ordinateur. Assurez–vous bien de suivre la procédure (voir 6.6) de redémarrage correctement.

LILO charge le noyau linux à partir du disque et ensuite le laisse agir. (le noyau est le programme central du système d'exploitation qui contrôle tous les autres programmes). Le noyau est totalement indépendant du BIOS ou de LILO.

Sur les plateformes non Intel, les choses se passent un peu différemment. Mais après avoir booté, les choses sont identiques – ou quasiment.

Linux examine le matériel sur lequel il va tourner. Il veut savoir quel type de disque dur vous avez, si vous avez ou non un bus souris, un réseau et quelques autres choses du même acabit. Linux ne peut se rappeler ces choses entre chaque boot donc il doit se poser ces questions à chaque démarrage. Heureusement, il ne *vous* pose pas les questions mais le fait directement au matériel. Lors du boot, le noyau Linux affiche des messages pour indiquer ce qu'il fait.

Le processus de recherche peut poser des problèmes mais s'il devait le faire, vous auriez déjà dû rencontrer ces problèmes lors de la première installation de GNU–Linux. Si vous avez des problèmes, reportez–vous aux instructions d'installation ou bien posez des questions sur les listes de distribution.

Le noyau permet simplement l'existence d'autres programmes donc une fois que tout est ok pour lui, il doit lancer un autre programme pour rendre les choses plus fonctionnelles. Le programme que le noyau lance est `init`. Après que le noyau ait lancé `init`, il ne lance jamais plus d'autres programmes. Le noyau devient alors un gestionnaire et un fournisseur de services.

Lorsque `init` se lance, il lance un certain nombre de scripts (des fichiers contenant des commandes) qui préparent le système à être utilisé. Il y a des routines de maintenance et de nombreux programmes qui font des choses comme fournir un prompteur de login, écouter les connexions réseau et garder une trace des activités de l'ordinateur.

Chapitre 19

La Licence Publique Générale de GNU

This is an unofficial translation of the GNU General Public License into french. It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU GPL—only the original English text of the GNU GPL does that. However, we hope that this translation will help french speakers understand the GNU GPL better.

Ceci est une traduction non officielle de la *GNU General Public License* en français. Elle n'a pas été publiée par la *Free Software Foundation*, et ne détermine pas les termes de distribution pour les logiciels qui utilisent la GNU GPL—seul le texte anglais original de la GNU GPL en a le droit. Cependant, nous espérons que cette traduction aidera les francophones à mieux comprendre la GPL.

Nous autorisons la FSF à apporter toute modification qu'elle jugera nécessaire pour rendre la traduction plus claire.

19.1 GNU GENERAL PUBLIC LICENSE

Version 2, juin 1991

Copyright © 1989, 1991, Free Software Foundation Inc. 675 Mass Ave, Cambridge, MA02139, États–Unis.

Il est permis à tout le monde de reproduire et distribuer des copies conformes de ce document de licence, mais aucune modification ne doit y être apportée.

Préambule

Les licences relatives à la plupart des logiciels sont destinées à supprimer votre liberté de les partager et de les modifier. Par contraste, la licence publique générale *GNU General Public License* veut garantir votre liberté de partager et de modifier les logiciels libres, pour qu'ils soient vraiment libres pour tous leurs utilisateurs. La présente licence publique générale s'applique à la plupart des logiciels de la *Free Software Foundation*, ainsi qu'à tout autre programme dont les auteurs s'engagent à l'utiliser. (Certains autres logiciels sont couverts par la Licence Publique Générale pour Bibliothèques GNU à la place). Vous pouvez aussi l'appliquer à vos programmes.

Quand nous parlons de logiciels libres, nous parlons de liberté, non de gratuité. Nos licences publiques générales veulent vous garantir :

– que vous avez toute liberté de distribuer des copies des logiciels libres (et de facturer ce service, si vous le souhaitez) ;
– que vous recevez les codes sources ou pouvez les obtenir si vous le souhaitez ;
– que vous pouvez modifier les logiciels ou en utiliser des éléments dans de nouveaux programmes libres ;
– et que vous savez que vous pouvez le faire.

Pour protéger vos droits, nous devons apporter des restrictions, qui vont interdire à quiconque de vous dénier ces droits, ou de vous demander de vous en désister. Ces restrictions se traduisent par certaines responsabilités pour ce qui vous concerne, si vous distribuez des copies de logiciels, ou si vous les modifiez.

Par exemple, si vous distribuez des copies d'un tel programme, gratuitement ou contre une rémunération, vous devez transférer aux destinataires tous les droits dont vous disposez. Vous devez vous garantir qu'eux–mêmes, par ailleurs, reçoivent ou peuvent recevoir le code source. Et vous devez leur montrer les présentes dispositions, de façon qu'ils connaissent leurs droits.

Nous protégeons vos droits en deux étapes :

1. Nous assurons le droit d'auteur (*copyright*) du logiciel, et
2. Nous vous proposons cette licence, qui vous donne l'autorisation légale de dupliquer, distribuer et/ou modifier le logiciel.

De même, pour la protection de chacun des auteurs, et pour notre propre protection, nous souhaitons nous assurer que tout le monde comprenne qu'il n'y a aucune garantie portant sur ce logiciel libre. Si le logiciel est modifié par quelqu'un d'autre puis transmis à des tiers, nous souhaitons que les destinataires sachent que ce qu'ils possèdent n'est pas l'original, de façon que tous problèmes introduits par d'autres ne se traduisent pas par une répercussion négative sur la réputation de l'auteur original.

Enfin, tout programme libre est en permanence menacé par des brevets de logiciels. Nous souhaitons éviter le danger que des sous–distributeurs d'un programme libre obtiennent à titre individuel des licences de brevets, avec comme conséquence qu'ils ont un droit de propriété sur le programme. Pour éviter cette situation, nous avons fait tout ce qui est nécessaire pour que tous brevets doivent faire l'objet d'une concession de licence qui en permette l'utilisation libre par quiconque, ou bien qu'il ne soit pas concédé du tout.

Nous présentons ci–dessous les closes et dispositions concernant la duplication, la distribution et la modification.

Conditions d'exploitation portant sur la duplication, la distribution et la modification

1. Le présent contrat de licence s'applique à tout programme ou autre ouvrage contenant un avis, apposé par le détenteur du droit de propriété, disant qu'il peut être distribué au titre des dispositions de la présente Licence Publique Générale. Ci–après, le « Programme » désigne l'un quelconque de ces programmes ou ouvrages, et un « ouvrage fondé sur le programme » désigne soit le programme, soit un ouvrage qui en dérive au titre de la loi sur le droit d'auteur ; plus précisément, il

s'agira d'un ouvrage contenant le programme ou une version de ce dernier, soit mot à mot, soit avec des modifications et/ou traduit en une autre langue (ci–après, le terme « modification » englobe, sans aucune limitation, les traductions qui en sont faites). Chaque titulaire de licence sera appelé « concessionnaire ».

Les activités autres que la duplication, la distribution et la modification ne sont pas couvertes par la présente licence ; elles n'entrent pas dans le cadre de cette dernière. L'exécution du programme n'est soumise à aucune restriction, et les résultats du programme ne sont couverts que si son contenu constitue un ouvrage fondé sur le programme (indépendamment du fait qu'il a été réalisé par exécution du programme). La véracité de ce qui précède dépend de ce que fait le programme.

2. Le concessionnaire peut dupliquer et distribuer des copies mot à mot du code source du programme tel qu'il les reçoit, et ce sur un support quelconque, du moment qu'il appose, d'une manière parfaitement visible et appropriée, sur chaque exemplaire, un avis approprié de droits d'auteur (*Copyright*) et de renonciation à garantie ; qu'il maintient intacts tous les avis qui se rapportent à la présente licence et à l'absence de toute garantie ; et qu'il transmet à tout destinataire du programme un exemplaire de la présente licence en même temps que le programme.

   Le concessionnaire peut facturer l'acte physique de transfert d'un exemplaire, et il peut, à sa discrétion, proposer en échange d'une rémunération une protection en garantie.

3. Le concessionnaire peut modifier son ou ses exemplaires du programme ou de toute portion de ce dernier, en formant ainsi un ouvrage fondé sur le programme, et dupliquer et distribuer ces modifications ou cet ouvrage selon les dispositions de la section 1 ci–dessus, du moment que le concessionnaire satisfait aussi à toutes ces conditions :
   - Le concessionnaire doit faire en sorte que les fichiers modifiés portent un avis, parfaitement visible, disant que le concessionnaire a modifié les fichiers, avec la date de tout changement.
   - Le concessionnaire doit faire en sorte que tout ouvrage qu'il distribue ou publie, et qui, en totalité ou en partie, contient le programme ou une partie quelconque de ce dernier ou en dérive, soit concédé en bloc, à titre gracieux, à tous tiers au titre des dispositions de la présente licence.
   - Si le programme modifié lit normalement des instructions interactives lors de son exécution, le concessionnaire doit, quand il commence l'exécution du programme pour une telle utilisation interactive de la manière la plus usuelle, faire en sorte que ce programme imprime ou affiche une annonce, comprenant un avis approprié de droits d'auteur, et un avis selon lequel il n'y a aucune garantie (ou autrement, que le concessionnaire fournit une garantie), et que les utilisateurs peuvent redistribuer le programme au titre de ces dispositions, et disant à l'utilisateur comment visualiser une copie de cette licence (exception : si le programme par lui–même est interactif mais n'imprime normalement pas une telle annonce, l'ouvrage du concessionnaire se fondant sur le programme n'a pas besoin d'imprimer une annonce).

4. Les exigences ci–dessus s'appliquent à l'ouvrage modifié pris en bloc. Si des sections identifiables de cet ouvrage ne dérivent pas du programme et peuvent être considérées raisonnablement comme représentant des ouvrages indépendants et distincts par eux–mêmes, alors la présente licence, et ses dispositions, ne s'appliquent pas à ces sections quand le concessionnaire les distribue sous forme d'ouvrages distincts. Mais quand le concessionnaire distribue ces mêmes sections en tant qu'élément d'un tout qui représente un ouvrage se fondant sur le programme, la distribution de ce tout doit se faire conformément aux dispositions de la présente licence, dont les autorisations, portant sur d'autres concessionnaires, s'étendent à la totalité dont il est question, et ainsi à chacune de ces parties, indépendamment de celui qu'il a écrite.

Ainsi, cette section n'a pas pour but de revendiquer des droits ou de contester vos droits sur un ouvrage entièrement écrit par le concessionnaire ; bien plus, l'intention est d'exercer le droit de surveiller la distribution d'ouvrages dérivée ou collective se fondant sur le programme.

De plus, un simple assemblage d'un autre ouvrage ne se fondant pas sur le programme, avec le programme (ou avec un ouvrage se fondant sur le programme) sur un volume d'un support de stockage ou distribution, ne fait pas entrer l'autre ouvrage dans le cadre de la présente licence.

5. Le concessionnaire peut dupliquer et distribuer le programme (ou un ouvrage se fondant sur ce dernier, au titre de la Section 2), en code objet ou sous une forme exécutable, au titre des dispositions des Sections 1 et 2 ci–dessus, du moment que le concessionnaire effectue aussi l'une des opérations suivantes :
   – Lui joindre le code source complet correspondant, exploitable par une machine, code qui doit être distribué au titre des Sections 1 et 2 ci–dessus sur un support couramment utilisé pour l'échange de logiciels ; ou bien
   – Lui joindre une offre écrite, dont la validité se prolonge pendant au moins 3 ans, de transmettre à un tiers quelconque, pour un montant non supérieur au coût pour le concessionnaire, de réalisation physique de la distribution de la source, un exemplaire complet, exploitable par une machine, du code source correspondant, qui devra être distribué au titre des dispositions des Sections 1 et 2 ci–dessus sur un support couramment utilisé pour l'échange des logiciels ; ou bien
   – Lui joindre les informations que le concessionnaire a reçues, pour proposer une distribution du code source correspondant (cette variante n'est autorisée que pour la distribution non commerciale, et seulement si le concessionnaire a reçu le programme sous forme exécutable ou sous forme d'un code objet, avec une telle offre, conformément à l'alinéa b) ci–dessus).

   Le code source d'un ouvrage représente la forme préférée de l'ouvrage pour y effectuer des modifications. Pour un ouvrage exécutable, le code source complet représente la totalité du code source pour tous les modules qu'il contient, plus tous fichiers de définitions d'interface associés, plus les informations en code machine pour commander la compilation et l'installation du programme exécutable. Cependant, à titre d'exceptions spéciales, le code source distribué n'a pas besoin de comprendre quoi que ce soit qui est normalement distribué (sous forme source ou sous forme binaire) avec les composants principaux (compilateur, noyau de système d'exploitation, etc.) du système d'exploitation sur lequel est exécuté le programme exécutable, à moins que le composant, par lui–même, soit joint au programme exécutable.

   Si la distribution de l ?exécutable ou du code objet est réalisée de telle sorte qu'elle offre d'accéder à une copie à partir d'un lieu désigné, alors le fait d'offrir un accès équivalent à la duplication du code source à partir de ce même lieu s'entend comme distribution du code source, même si des tiers ne sont pas contraints de dupliquer la source en même temps que le code objet.

6. Le concessionnaire ne peut dupliquer, modifier, concéder en sous–licence ou distribuer le programme, sauf si cela est expressément prévu par les dispositions de la présente licence. Toute tentative pour autrement dupliquer, modifier, concéder en sous–licence ou distribuer le programme est répétée nulle, et met automatiquement fin aux droits du concessionnaire au titre de la présente licence. Cependant, les parties qui ont reçu des copies, ou des droits, de la part du concessionnaire au titre de la présente licence, ne verront pas expirer leur contrat de licence, tant que ces parties agissent d'une manière parfaitement conforme.

7. Il n'est pas exigé du concessionnaire qu'il accepte la présente licence, car il ne l'a pas signée. Cependant, rien d'autre n'octroie au concessionnaire l'autorisation de modifier ou de distribuer

le programme ou ses ouvrages dérivés. Ces actions sont interdites par la loi si le concessionnaire n'accepte pas la présente licence. En conséquence, par le fait de modifier ou de distribuer le programme (ou un ouvrage quelconque se fondant sur le programme), le concessionnaire indique qu'il accepte la présente licence, et qu'il a la volonté de se conformer à toutes les closes et dispositions concernant la duplication, la distribution ou la modification du programme ou d'ouvrages se fondant sur ce dernier.

8. Chaque fois que le concessionnaire redistribue le programme (ou tout ouvrage se fondant sur le programme), le destinataire reçoit automatiquement une licence de l'émetteur initial de la licence, pour dupliquer, distribuer ou modifier le programme, sous réserve des présentes closes et dispositions. Le concessionnaire ne peut imposer aucune restriction plus poussée sur l'exercice, par le destinataire, des droits octroyés au titre des présentes. Le concessionnaire n'a pas pour responsabilité d'exiger que des tiers se conforment à la présente licence.
9. Si, en conséquence une décision de justice ou une allégation d'infraction au droit des brevets, ou pour toute autre raison (qui n'est pas limitée à des problèmes de propriétés industrielles), des conditions sont imposées au concessionnaire (par autorité de justice, par convention ou autrement), qui entrent en contradiction avec les dispositions de la présente licence, elles n'exemptent pas le concessionnaire de respecter les dispositions de la présente licence. Si le concessionnaire ne peut procéder à la distribution de façon à satisfaire simultanément à ces obligations au titre de la présente licence et à toutes autres obligations pertinentes, alors, en conséquence de ce qui précède, le concessionnaire peut ne pas procéder du tout à la distribution du programme. Par exemple, si une licence de brevet ne permettait pas une redistribution du programme, sans redevances, par tous ceux qui reçoivent des copies directement ou indirectement par l'intermédiaire du concessionnaire, alors le seul moyen par lequel le concessionnaire pourrait satisfaire tant à cette licence de brevet qu'à la présente licence, consisterait à s'abstenir complètement de distribuer le programme.

   Si une partie quelconque de cette section est considérée comme nulle ou non exécutoire dans certaines circonstances particulières, le reste de cette section est réputé s'appliquer, et la section dans son ensemble est considérée comme s'appliquant dans les autres circonstances.

   La présente section n'a pas pour objet de pousser le concessionnaire à enfreindre tous brevets ou autres revendications à droit de propriété, ou encore à contester la validité de une ou plusieurs quelconques de ces revendications ; la présente section a pour objet unique de protéger l'intégrité du système de distribution des logiciels libres, système qui est mis en oeuvre par les pratiques liées aux licences publiques. De nombreuses personnes ont apporté une forte contribution à la gamme étendue des logiciels distribués par ce système, en comptant sur l'application systématique de ce système ; c'est à l'auteur/donateur de décider s'il a la volonté de distribuer le logiciel par un quelconque autre système, et un concessionnaire ne peut imposer ce choix.

   La présente section veut rendre parfaitement claire ce que l'on pense être une conséquence du reste de la présente licence.
10. Si la distribution et/ou l'utilisation du Programme est restreinte dans certains pays, sous l'effet de brevets ou d'interfaces présentant un droit d'auteur, le détenteur du droit d'auteur original, qui soumet le Programme aux dispositions de la présente licence, pourra ajouter une limitation expresse de distribution géographique excluant ces pays, de façon que la distribution ne soit autorisée que dans les pays ou parmi les pays qui ne sont pas ainsi exclus. Dans ce cas, la limitation fait partie intégrante de la présente licence, comme si elle était écrite dans le corps de la présente licence.

    La *Free Software Foundation* peut, de temps à autre, publier des versions révisées et/ou nouvelles du *General Public License*. Ces nouvelles versions seront analogues, du point de vue de leur

esprit, à la présente version, mais pourront en différer dans le détail, pour résoudre de nouveaux problèmes ou de nouvelles situations.

Chaque version reçoit un numéro de version qui lui est propre. Si le programme spécifie un numéro de version de la présente licence, qui s'applique à cette dernier et "à toute autre version ultérieure", le concessionnaire a le choix de respecter les closes et dispositions de cette version, ou une quelconque version ultérieure publiée par la *Free Software Foundation*. Si le programme ne spécifie pas de numéro de version de la présente licence, le concessionnaire pourra choisir une version quelconque publiée à tout moment par la *Free Software Foundation*.

11. Si le concessionnaire souhaite incorporer des parties du programme dans d'autres programmes libres dont les conditions de distribution sont différentes, il devrait écrire à l'auteur pour demander son autorisation. Pour un logiciel soumis à droit d'auteur par la *Free Software Foundation*, il devra écrire à la *Free Software Foundation* ; nous faisons quelquefois des exceptions à cette règle. Notre décision va être guidée par le double objectif de protéger le statut libre de tous les dérivés de nos logiciels libres, et de favoriser le partage et la réutilisation des logiciels en général.

    ABSENCE DE GARANTIE

12. COMME LA LICENCE DU PROGRAMME EST CONCÉDÉE À TITRE GRATUIT, IL N'Y AUCUNE GARANTIE S'APPLIQUANT AU PROGRAMME, DANS LA MESURE AUTORISÉE PAR LA LOI EN VIGUEUR. SAUF MENTION CONTRAIRE ÉCRITE, LES DÉTENTEURS DU DROIT D'AUTEUR ET/OU LES AUTRES PARTIES METTENT LE PROGRAMME À DISPOSITION "EN L'ÉTAT", SANS AUCUNE GARANTIE DE QUELQUE NATURE QUE CE SOIT, EXPRESSE OU IMPLICITE, Y COMPRIS, MAIS SANS LIMITATION, LES GARANTIES IMPLICITES DE COMMERCIALISATION ET DE L'APTITUDE À UN OBJET PARTICULIER. C'EST LE CONCESSIONNAIRE QUI PREND LA TOTALITÉ DU RISQUE QUANT A LA QUALITÉ ET AUX PERFORMANCES DU PROGRAMME. SI LE PROGRAMME SE RÉVÉLAIT DÉFECTUEUX, C'EST LE CONCESSIONNAIRE QUI PRENDRAIT À SA CHARGE LE COÛT DE L'ENSEMBLE DES OPÉRATIONS NÉCESSAIRES D'ENTRETIEN, RÉPARATION OU CORRECTION.

13. EN AUCUN CAS, SAUF SI LA LOI EN VIGUEUR L'EXIGE OU SI UNE CONVENTION ÉCRITE EXISTE À CE SUJET, AUCUN DÉTENTEUR DE DROITS D'AUTEUR, OU AUCUNE PARTIE AYANT LE POUVOIR DE MODIFIER ET/OU DE REDISTRIBUER LE PROGRAMME CONFORMÉMENT AUX AUTORISATIONS CI–DESSUS, N'EST RESPONSABLE VIS–À–VIS DU CONCESSIONNAIRE POUR CE QUI EST DES DOMMAGES, Y COMPRIS TOUS DOMMAGES GÉNÉRAUX, SPÉCIAUX, ACCIDENTELS OU INDIRECTS, RÉSULTANT DE L'UTILISATION OU DU PROGRAMME OU DE L'IMPOSSIBILITÉ D'UTILISER LE PROGRAMME (Y COMPRIS, MAIS SANS LIMITATION, LA PERTE DE DONNÉES, OU LE FAIT QUE DES DONNÉES SONT RENDUES IMPRÉCISES, OU ENCORE LES PERTES ÉPROUVÉES PAR LE CONCESSIONNAIRE OU PAR DES TIERS, OU ENCORE UN MANQUEMENT DU PROGRAMME À FONCTIONNER AVEC TOUS AUTRES PROGRAMMES), MÊME SI CE DÉTENTEUR OU CETTE AUTRE PARTIE A ÉTÉ AVISÉ DE LA POSSIBILITÉ DE TELS DOMMAGES.

    FIN DES CONDITIONS D'EXPLOITATION

19.2 Comment appliquer ces dispositions à vos nouveaux programmes ?

Si le concessionnaire développe un nouveau programme, et s'il en souhaite l'utilisation la plus large possible dans le public, le meilleur moyen d'y arriver est d'en faire un logiciel libre, que tout le monde pourra redistribuer et modifier au titre des présentes dispositions.

Dans ce but, il convient de rattacher au programme les avis suivants. Le moyen le plus sûr consiste à les rattacher au début de chaque fichier source, pour avertir le plus efficacement possible de l'exclusion de garantie ; et chaque fichier doit comporter au moins la ligne *copyright*, et un pointeur indiquant où est localisée la totalité de l'avis.

Une ligne pour donner le nom du programme et une idée de ce qu'il fait.

```
Copyright © 19yy nom de l'auteur
```

```
Ce programme est un logiciel libre; vous pouvez le redistribuer et/ou
le modifier conformément aux dispositions de la Licence Publique Géné-
rale GNU, telle que publiée par la Free Software Foundation; version 2
de la licence, ou encore (à votre choix) toute version ultérieure.
```

```
Ce programme est distribué dans l'espoir qu'il sera utile, mais SANS AU-
CUNE GARANTIE; sans même la garantie implicite de COMMERCIALISATION ou
D'ADAPTATION A UN OBJET PARTICULIER. Pour plus de détail, voir la Licence
Publique Générale GNU .
```

```
Vous devez avoir reçu un exemplaire de la Licence Publique Générale GNU
en même temps que ce programme; si ce n'est pas le cas, écrivez à la Free
Software Foundation Inc., 675 Mass Ave, Cambridge, MA 02139, États-Unis.
```

Ajoutez aussi des informations sur le moyen permettant d'entrer en contact avec vous par courrier électronique (e–mail) et courrier normal.

Si le programme est interactif, prévoyez en sortie un court avis, tel que celui qui est présenté ci–dessous, lors du démarrage en mode interactif.

```
Gnomovision version 69, Copyright © 19 yy nom de l'auteur
```

```
Gnomovision est livré absolument SANS AUCUNE GARANTIE; pour plus de dé-
tail, tapez « show w ». Il s'agit d'un logiciel libre, et vous avez le
droit de le redistribuer dans certaines conditions; pour plus de détail,
tapez « show c ».
```

Les instructions hypothétiques « show w » et « show c » doivent présenter les parties appropriées de la Licence Publique Générale. Bien évidemment, les instructions que vous utilisez peuvent porter d'autres noms que « show w » et « show c » ; elles peuvent même correspondre à des clics de souris ou à des éléments d'un menu, selon ce qui convient à votre programme.

Si nécessaire, vous devrez aussi demander à votre employeur (si vous travaillez en tant que programmeur) ou à votre éventuelle école ou université, de signer une "renonciation à droit d'auteur" concernant le programme. En voici un échantillon (il suffit de modifier les noms) :

```
Yoyodyne, Inc., par la présente, renonce à tout intérêt de droits d'au-
teur dans le programme « Gnomovision » (qui fait des passages au niveau
```

```
des compilateurs) écrit par James Hacker.

Signature de Ty Coon, 1er avril 1989

Ty Coon, President of Vice
```

La présente Licence Publique Générale n'autorise pas le concessionnaire à incorporer son programme dans des programmes propriétaires. Si votre programme est une bibliothèque de sous–programmes, vous pouvez considérer comme plus intéressant d'autoriser une édition de liens des applications propriétaires avec la bibliothèque. Si c'est ce que vous souhaitez, vous devrez utiliser non pas la présente licence, mais la Licence Publique Générale pour Bibliothèques GNU.