Guide de référence pour Debian

Osamu Aoki <debian@aokiconsulting.com>
Coordinateur de traduction en Français : Guillaume Erbs <gerbs@free.fr>
'Auteurs' page 163

CVS, Sun, 13 Oct 2002 22 :40 :13 -0600

Résumé

Ce Guide de référence pour Debian (http://qref.sourceforge.net/) se propose de donner une vue générale du système Debian comme **guide de l'utilisateur après installation**. Cela couvre beaucoup d'aspects de l'administration système à l'aide d'exemples de **commandes shell**. Des didacticiels sur les bases, des astuces, et d'autres informations sont fournies sur des sujets comme les concepts fondamentaux du système Debian, des astuces d'installation, la gestion des paquets Debian, le noyau Linux sous Debian, la configuration du système, la configuration d'une passerelle, les éditeurs de texte, CVS, la programmation, et GnuPG pour des **non-développeurs**.

Copyright

Copyright © 2001-2002 by Osamu Aoki <debian@aokiconsulting.com> Copyright (Chapter 2) © 1996-2001 by Software in the Public Interest

Ce document peut être utilisé selon les termes de la Licence Publique Générale de GNU version 2 ou suivante. (http://www.gnu.org/copyleft/gpl.html)

Il est permis de produire et distribuer des copies conformes de ce document à condition que la présente notice de copyright et la présente notice de permission soient préservées sur toutes les copies.

Il est permis de copier et distribuer des versions modifiées de ce document selon les conditions d'une copie conforme, à condition que le travail dérivé résultant soit entièrement distribué selon les termes d'une notice de permission identique à celle-ci.

Il est permis de copier et distribuer des traductions de ce document dans d'autres langues, selon les conditions pour versions modifiées ci-dessus, sauf que cette notice de permission peut être incluse sous forme d'une traduction approuvée par la Free Software Foundation à la place de l'Anglais original.

Table des matières

1 Préface **1**

1.1 Document officiel 1

1.2 Conventions du document 2

1.3 Configuration de base 2

1.4 Fondements de la distribution Debian 3

2 Notions fondamentales sur Debian **5**

2.1 Les archives Debian 5

2.1.1 Structure de répertoires 5

2.1.2 Distributions Debian 6

2.1.3 La distribution `stable` 6

2.1.4 La distribution `testing` 7

2.1.5 La distribution `unstable` 7

2.1.6 La distribution `frozen` 8

2.1.7 Les noms de code de la distribution Debian 8

2.1.8 Noms de code utilisés par le passé 8

2.1.9 Source d'inspiration pour les noms de code 9

2.1.10 Le répertoire `pool` 9

2.1.11 Notes historiques sur `sid` 10

2.1.12 Paquets téléchargés dans `incoming` 10

2.1.13 Récupérer un paquet ancien 10

2.1.14 Sections architectures 11
2.1.15 Le code source 11
2.2 Système de gestion des paquets Debian 12
2.2.1 Vue générale des paquets Debian 12
2.2.2 Format des paquets Debian 12
2.2.3 Conventions de nommage pour les fichiers de paquets Debian 13
2.2.4 Préservation de la configuration locale 13
2.2.5 Scripts de maintenance Debian 14
2.2.6 Priorité des paquets 14
2.2.7 Paquets virtuels 15
2.2.8 Dépendances des paquets 16
2.2.9 Signification de « pre-depends » 16
2.2.10 Etat d'un paquet 17
2.2.11 Garder des paquets lors d'une mise à jour 17
2.2.12 Paquets sources 18
2.2.13 Construire des paquets binaires à partir d'un paquet source 18
2.2.14 Créer de nouveaux paquets Debian 19
2.3 Mettre à jour un système Debian 19
2.3.1 Méthodes de mise à jour d'un système Debian 20
2.3.2 Vue générale des outils de gestion de paquets 20
2.3.3 `dpkg` 20
2.3.4 APT 20
2.3.5 `dselect` 21
2.3.6 Mise à jour d'un système en marche 21
2.3.7 Fichiers d'archive `.deb` téléchargés et sauvegardés 21
2.3.8 Garder une trace des mises à jour 22
2.4 Le processus de démarrage de Debian 22
2.4.1 Le programme `init` 22
2.4.2 Niveaux de fonctionnement 22

2.4.3 Personnaliser le processus de démarrage 23
2.5 Support de la diversité 24
2.6 Internationalisation 24
2.7 Debian et le noyau 24
2.7.1 Compiler un noyau avec des sources non Debian 25
2.7.2 Outils pour compiler un noyau personnalisé 25
2.7.3 Chargeurs de remplacement 26
2.7.4 Disquettes de démarrage personnalisées 26
2.7.5 Dispositions spéciales pour manipuler les modules 26
2.7.6 Désinstaller le paquet d'un vieux noyau 26

3 Installation du système Debian 29
3.1 Astuces générales sur l'installation du système Linux 29
3.1.1 Compatibilité matérielle 29
3.1.2 Déterminer le matériel et les puces du PC 30
3.1.3 Trouver le matériel du PC avec Debian 30
3.1.4 Trouver le matériel du PC avec d'autres systèmes d'exploitation (SE) 30
3.1.5 Le Mythe Lilo 31
3.1.6 Choix des disquettes de boot 31
3.1.7 Installation 31
3.1.8 Nom d'hôte et IP à utiliser pour le réseau local 32
3.1.9 Comptes utilisateurs 33
3.1.10 Création des systèmes de fichiers 33
3.1.11 Lignes directrices pour la mémoire DRAM 36
3.1.12 Espace de Swap 37
3.2 Configuration de Bash 37
3.3 Configuration de la souris 38
3.3.1 Souris PS/2 38
3.3.2 Souris USB 38

3.4 Configuration NFS 39
3.5 Configuration Samba 39
3.6 Configuration de l'imprimante 40
3.6.1 `lpr/lpd` 41
3.6.2 CUPS™ 41
3.7 Autres conseils de configuration de l'hôte 42
3.7.1 Installer quelques paquets supplémentaires après l'installation 42
3.7.2 Modules 42
3.7.3 Configuration de base de CD-RW 43
3.7.4 Grande capacité mémoire et arrêt automatique 43
3.7.5 Étranges problèmes d'accès à certains sites web 44
3.7.6 Configuration d'une connexion RTC avec PPP 44
3.7.7 Autre configuration à vérifier dans `/etc` 45

4 Didacticiels Debian 47
4.1 Sources d'information 47
4.2 La console Linux 47
4.2.1 Se connecter 47
4.2.2 Ajouter un compte utilisateur 48
4.2.3 Comment éteindre 48
4.2.4 Édition en ligne de commande 49
4.2.5 Commandes de base à retenir 49
4.2.6 Le système X Window 50
4.2.7 Commandes importantes au clavier 50
4.3 Midnight Commander (MC) 50
4.3.1 Installer MC 51
4.3.2 Démarrer MC 51
4.3.3 Gestionnaire de fichiers 51
4.3.4 Astuces en ligne de commande 52

4.3.5 Éditeur 52
4.3.6 Visionneur 53
4.3.7 Possibilités de démarrage automatique 53
4.3.8 Système de fichiers FTP virtuel 53
4.4 Étude approfondie 54

5 Mise à jour d'une distribution 55
5.1 Préparation de la transition (potato vers woody) 55
5.2 Mise à jour vers Woody 56
5.3 Configuration de Woody 57
5.4 Améliorer le fichier `sources.list` 57

6 Gestion des paquets Debian 59
6.1 Introduction 59
6.1.1 Outils principaux 59
6.1.2 Outils pratiques 60
6.2 Commandes de survie Debian 60
6.2.1 Installation de *tâches* avec `tasksel` 60
6.2.2 Système d'installation avec APT 60
6.2.3 Système de mise à jour avec APT 61
6.2.4 Rechercher les bogues de Debian et demander de l'aide 62
6.2.5 Résolution de problèmes de mise à jour avec APT 63
6.2.6 Sauvetage avec `dpkg` 63
6.2.7 Sauver un système après avoir effacé `/var` 64
6.2.8 Installer un paquet sur un système qui ne démarre plus 65
6.2.9 Que faire si `dpkg` est cassé 65
6.3 Commandes « nirvana » de Debian 65
6.3.1 Informations sur un fichier 66
6.3.2 Informations sur un paquet 66

6.3.3 Reconfigurer les paquets installés 67
6.3.4 Enlever et purger des paquets 67
6.3.5 Garder de vieux paquets 68
6.3.6 `dselect` – configuration générale 68
6.3.7 Supprimer les paquets du cache 69
6.3.8 Enregistrer/copier la configuration du système 69
6.3.9 Porter un paquet vers le système `stable` 69
6.3.10 Archive locale de paquets 71
6.3.11 Convertir ou installer un paquet non Debian 71
6.3.12 Vérifier les fichiers de paquets installés 71
6.4 Autres particularités de Debian 72
6.4.1 La commande `dpkg-divert` 72
6.4.2 Le paquet `equivs` 72
6.4.3 Commandes de rechange 72
6.4.4 `init` System-V et niveaux de fonctionnement 73
6.4.5 Services désactivés 74

7 Le noyau Linux et Debian 75
7.1 Recompilation du noyau 75
7.1.1 Méthode Debian standard 75
7.1.2 Méthode classique 76
7.2 Le noyau 2.4 modulaire 77
7.2.1 PCMCIA 77
7.2.2 SCSI 78
7.2.3 Fonctions réseau 78
7.2.4 Système de fichiers EXT3 (> 2.4.17) 79
7.2.5 Support Realtek RTL-8139 dans le noyau 2.4 80

8 Astuces Debian **81**

8.1 Démarrer le système 81

8.1.1 "J'ai oublié le mot de passe de root !" (1) 81

8.1.2 "J'ai oublié le mot de passe de root !" (2) 82

8.1.3 Je ne peux pas lancer le système 82

8.1.4 Autres astuces avec l'invite de démarrage. 83

8.2 Enregistrer les activités 84

8.2.1 Enregistrer les activités du shell 84

8.2.2 Enregistrer les activités sous X 84

8.2.3 Enregistrer les changements aux fichiers de configuration 84

8.3 Copier et archiver un sous-répertoire entier 85

8.3.1 Commandes de base pour copier un sous-répertoire entier 85

8.3.2 `cp` 85

8.3.3 `tar` 85

8.3.4 `pax` 86

8.3.5 `cpio` 86

8.3.6 `afio` 86

8.4 Récupération d'un système bloqué 87

8.4.1 Tuer un processus 87

8.4.2 ALT-SysRQ 87

8.5 Petites commandes utiles à se rappeler 87

8.5.1 Pager 87

8.5.2 Mémoire libre 88

8.5.3 Régler l'heure (BIOS) 88

8.5.4 Régler l'heure (NTP) 88

8.5.5 Comment désactiver l'écran de veille 89

8.5.6 Chercher dans la base de données administrative 89

8.5.7 Désactiver le son (beep) 90

8.5.8 Messages d'erreur sur l'écran de la console 90

8.5.9 Régler la console 90
8.5.10 Remettre la console dans un état sain 91
8.5.11 Convertir des fichiers texte DOS vers Unix 91
8.5.12 Modifier des fichiers grâce aux expressions rationnelles 91
8.5.13 Convertir un gros fichier en plusieurs petits 91
8.5.14 Bouts de scripts pour les tubes 92
8.5.15 Récupérer du texte ou une archive de liste de diffusion à partir d'une page Web 92
8.5.16 Imprimer joliement une page Web 92
8.5.17 Mesurer la durée d'une commande 93
8.5.18 Commande `nice` 93
8.5.19 Programmer des activités (`cron`, `at`) 93
8.5.20 Changement de console avec `screen` 94
8.5.21 Bases pour tester un réseau 96
8.5.22 Vider les courriers électroniques de la file locale 96
8.5.23 Supprimer les courriers électroniques gelés de la file locale 96
8.5.24 Supprimer le contenu d'un fichier 97
8.5.25 Fichiers fantômes 97
8.5.26 `chroot` 97
8.5.27 Monter une image de disque dur 98
8.5.28 Samba 98

9 Configurer un système Debian 99

9.1 Astuces sur l'initialisation du système 99
9.1.1 Personnaliser les scripts d'initialisation 99
9.1.2 Personnaliser le journal du système 99
9.1.3 Optimisation de l'accès au matériel 100
9.2 Contrôle d'accès 100
9.2.1 Contrôle d'accès avec PAM et login 100

9.2.2 Pourquoi GNU su ne supporte pas le groupe wheel . . . 101
9.2.3 Signification des groupes . . . 101
9.2.4 sudo – un environnement de travail plus sûr . . . 102
9.2.5 Contrôle d'accès aux démons . . . 103
9.2.6 Lightweight Directory Access Protocol . . . 103
9.3 Graveur de CD . . . 104
9.3.1 Introduction . . . 104
9.3.2 Première approche : modules + lilo . . . 104
9.3.3 Seconde approche : recompiler le noyau . . . 105
9.3.4 Étapes post-configuration . . . 105
9.3.5 Image de CD (bootable) . . . 106
9.3.6 Graver un CD . . . 106
9.3.7 Créer l'image d'un CD . . . 107
9.3.8 Images de CD Debian . . . 107
9.3.9 Système de sauvegarde sur CD-R . . . 108
9.3.10 Copier un CD audio sur un CD-R . . . 108
9.4 Le programme X . . . 108
9.4.1 Serveur X . . . 109
9.4.2 Client X . . . 110
9.4.3 Connexion TCP/IP à X . . . 111
9.4.4 Connexion X distante : xhost . . . 111
9.4.5 Connexion X distante : ssh . . . 111
9.4.6 xterm . . . 112
9.4.7 Devenir root sous X . . . 112
9.4.8 Polices TrueType dans X . . . 113
9.4.9 Navigateur Web (graphique) . . . 114
9.4.10 CJK et X . . . 114
9.5 SSH . . . 115
9.5.1 Bases . . . 115

9.5.2 Redirection de port – tunnel SMTP/POP3 117
9.5.3 Se connecter avec moins de mots de passe 117
9.5.4 Clients SSH étrangers 118
9.5.5 SSH agent 118
9.5.6 Problèmes 119
9.6 Logiciels de courrier électronique 119
9.6.1 Agent de transport de courrier électronique (Exim) 119
9.6.2 Tout récupérer pour les adresses e-mail non existantes (Exim) 119
9.6.3 Utilitaire de courrier électronique (fetchmail) 120
9.6.4 Utilitaire de courrier électronique (procmail) 120
9.6.5 Agent pour utilisateur de courrier électronique (Mutt) 121
9.7 Localisation et support des langues nationales 121
9.7.1 Support des langues 121
9.7.2 Locales 122
9.7.3 Activer le support des locales 123
9.7.4 Activer une locale particulière 123
9.7.5 Après `locale` 124

10 Construire une passerelle avec Debian **125**
10.1 Configuration réseau 125
10.1.1 Configuration de la passerelle 125
10.1.2 IP-masquerade 126
10.1.3 Points principaux de la configuration réseau pour une passerelle 127
10.2 Gérer plusieurs connexions à l'Internet 127

11 Editeurs **129**
11.1 Editeurs populaires 129
11.2 Editeurs de sauvetage 129
11.3 Emacs et Vim 130

11.3.1 Conseils pour Vim 130
11.3.2 Conseils pour Emacs 130
11.3.3 Démarrer l'éditeur 130
11.3.4 Résumé des commandes de l'éditeur (Emacs, Vim) 131
11.3.5 Configuration de Vim 134
11.3.6 Ctags 134
11.3.7 Convertir un écran avec syntaxe en surbrillance en un source HTML 134
11.3.8 Ecran scindé avec `vim` 134

12 CVS **137**

12.1 Installation d'un serveur CVS 137
12.2 Exemples de sessions CVS 138
12.2.1 CVS anonyme (téléchargement seulement) 138
12.2.2 Utilisation d'un serveur CVS local 138
12.2.3 Utilisation d'un pserver CVS distant 138
12.2.4 Utilisation d'un CVS distant avec `ssh` 138
12.2.5 Créer une nouvelle archive CVS 138
12.2.6 Travailler avec CVS 139
12.2.7 Exportation de fichiers à partir de CVS 140
12.2.8 Administrer CVS 140
12.3 Résoudre les problèmes de CVS 140
12.3.1 Permissions de fichiers dans le dépôt 140
12.3.2 Bit d'exécution 141
12.4 Commandes CVS 141

13 Programmation **143**

13.1 Où commencer 143
13.2 BASH 143
13.3 AWK 144

13.4 PERL 145
13.5 PYTHON 146
13.6 MAKE 147
13.7 C 148
13.7.1 Programme simple en C (gcc) 149
13.7.2 Déboguer (gdb) 150
13.7.3 Flex – un meilleur Lex 150
13.7.4 Bison – un meilleur Yacc 151
13.7.5 Autoconf – uninstall 151
13.8 SGML 151
13.9 Paquetage 152

14 GnuPG **153**
14.1 Installer GnuPG 153
14.2 Utiliser GnuPG 154
14.3 Gérer GnuPG 154
14.4 Utilisation avec Mutt 155

15 Support Debian **157**
15.1 Références 157
15.2 Trouver le sens d'un mot 159
15.3 Système de suivi des bogues Debian 159
15.4 Listes de diffusion 159
15.5 Internet Relay Chat 160
15.6 Moteurs de recherche 160
15.7 Sites internet 160

A Annexe **163**
A.1 Auteurs 163
A.2 Garanties 164

A.3 Retour 165
A.4 Format du document 165
A.5 Labyrinthe de Debian 165
A.6 Les citations Debian 166

Chapitre 1

Préface

Ce Guide de référence pour Debian (`http://qref.sourceforge.net/`) se propose de donner une vue générale du système Debian comme **guide de l'utilisateur après installation**. Le lecteur visé est quelqu'un qui veut lire des scripts shell. Je suppose que le lecteur a des bases sur les systèmes de type Unix pour lire ce document.

J'ai pris la décision de **ne pas** expliquer tout en détail si cela peut être trouvé dans une **page de manuel**, une **page info**, ou un **HOWTO**. Plutôt que de tout expliquer, j'essaie de donner des informations pratiques plus directement en fournissant des *séquences de commandes exactes* ou des *scripts d'exemple*. Beaucoup des informations données consistent en rappels ou en pointeurs vers les références officielles listées dans 'Références' page 157. Ceci partiellement parce que ce document a été démarré en tant que « **Référence Rapide** ».

Mon principe est de **le garder court et simple**.

Pour une aide en vue d'une maintenance urgente d'un système, rendez-vous à 'Commandes de survie Debian' page 60 immédiatement.

1.1 Document officiel

La dernière version du document officiel est dans l'archive Debian dans le paquet `debian-reference` et est aussi disponible à `http://www.debian.org/doc/manuals/debian-reference/`.

La dernière version de développement est à `http://qref.sourceforge.net/Debian/`. Le projet est hébergé à `http://qref.sourceforge.net/` où ce document est disponible au téléchargement dans les formats texte, HTML, PDF, SGML et PostScript.

1.2 Conventions du document

Le « Guide de référence pour Debian » procure de l'information par le biais de commandes simples en shell BASH. Voici les conventions utilisées :

```
# commande en compte «~root~»
$ commande en compte utilisateur
... description de l'action
```

Voir 'BASH' page 143 pour plus d'information sur Bash.

Références :
- une page de **manuel Unix** est donnée dans la forme `bash(1)`.
- une page **GNU TEXINFO** est donnée dans la forme `info libc`.
- un **livre** est donné dans la forme *Le langage C*.
- une **URL** est donnée dans la forme `http://www.debian.org/doc/manuals/debian-reference/`.
- un **fichier** est donné dans la forme `/usr/share/doc/Debian/reference/`.

Les abréviations suivantes sont utilisées :
- **LDP** : Linux Documentation Project (`http://www.tldp.org/`)
- **DDP** : Debian Documentation Project (`http://www.debian.org/doc/`)

Dans ce document, seules des URL sont fournies pour les documents du LDP, mais ils peuvent aussi être obtenus dans des paquets et installés dans `/usr/share/doc/HOWTO/`. Voir 'Références' page 157.

Des exemples de scripts sont fournis dans le répertoire des exemples (`examples/`); pour les fichiers cachés, le préfixe « . » est remplacé par un « _ ».

1.3 Configuration de base

Si le système est installé avec le strict minimum comme paquets, asssurez-vous d'exécuter les commandes suivantes pour installer quelques paquets essentiels et quelques documents importants :

```
# apt-get install info man-db boc-base dhelp apt apt-utils auto-apt \
                  dpkg less mc ssh nano-tiny elvis-tiny vim sash \
   kernel-package \
   manpages manpages-dev doc-debian doc-linux-text \
   debian-policy developers-reference maint-guide \
   apt-howto harden-doc install-doc \
   libpam-doc glibc-doc samba-doc exim-doc cvsbook \
   gnupg-doc
# apt-get install debian-reference # pour Sarge, faites cela aussi :)
```

1.4 Fondements de la distribution Debian

Debian se présente sous la forme simultanée de 3 éditions :

- stable : Convient pour un serveur de production. Ennuyeux pour une station de travail (ST). Voir 'La distribution `stable`' page 6.
- testing : Convient pour une ST. Voir 'La distribution `testing`' page 7.
- unstable : Ne pas charger cette édition aveuglément. Voir 'La distribution `unstable`' page 7.

Lisez au moins la liste de discussion centrale `debian-devel-announce@lists.debian.org` pour des mises à jour sur l'état de Debian. (NdT : Pour les francophones, vous trouverez de l'aide en vous abonnant à la liste de discussion francophone `debian-user-french-request@lists.debian.org`.

Au mois de mars 2002, ces trois versions correspondent à `Potato` (qualité de production), `Woody` (en beta-test, très stable maintenant), et `Sid` (en alpha-test). En août 2002, juste après la sortie de `Woody`, cela correspond à `Woody` (qualité de production), `Sarge` (en beta-test, elle sera assez difficile pendant quelque temps), et `Sid` (toujours en alpha-test). Lorsque les paquets dans `unstable` n'ont plus de bogues critiques (Release Critical, RC) répertoriés pendant environ une semaine de test, ils sont automatiquement inclus dans `testing`. Voir 'Les archives Debian' page 5.

En théorie, il y a deux choses que vous pouvez faire pour disposer des dernières versions des applications.

- 'Système d'installation avec APT' page 60 (principalement pour des stations de travail)
- 'Porter un paquet vers le système `stable`' page 69 (principalement pour des serveurs)

Après avoir expliqué quelques éléments fondamentaux de la distribution Debian dans 'Notions fondamentales sur Debian' page 5, je vais présenter quelques informations de base pour vous aider à vivre tranquillement avec les dernières applications, en tirant parti de la distribution `testing` et de la distribution `unstable` de Debian. Les impatients devront aller immédiatement à 'Commandes de survie Debian' page 60. Bonne et heureuse mise à jour !

Chapitre 2

Notions fondamentales sur Debian

Ce chapitre donne des notions fondamentales sur le système Debian pour des non-développeurs. Pour des informations officielles, voir :

- Charte Debian
- Manuel du paquetage Debian (Potato)
- Référence du développeur Debian
- Guide des nouveaux responsables Debian

listés dans 'Références' page 157.

Si l'on recherche des explications basées sur des solutions et sans les détails, se référer directement à 'Gestion des paquets Debian' page 59 ou aux chapitres appropriés.

Ce chapitre consiste en une réorganisation de documents pris dans la « FAQ Debian », afin qu'un administrateur système Debian puisse débuter.

2.1 Les archives Debian

2.1.1 Structure de répertoires

Les logiciels paquetés pour Debian sont disponibles dans un des nombreux arbres de répertoires sur chaque site mirroir Debian (`http://www.debian.org/misc/README.mirrors`) accessible par FTP ou HTTP.

Les répertoires suivants sont sur chaque mirroir Debian sous le répertoire `debian` :

`/dists/`: Ce répertoire contient les « distributions », et est utilisé pour accéder aux paquets actuellement disponibles dans les versions et pré-versions de Debian. Certains vieux paquets et fichiers `Packages.gz` sont toujours là.

/pool/ : Nouvelle place de tous les paquets des versions et pré-versions de Debian.

/tools/ : Utilitaires DOS pour créer des disquettes de démarrage, partitionner un disque dur, compresser/décompresser des fichiers, et démarrer Linux.

/doc/ : La documentation de base de Debian, telle que la FAQ, les instructions pour faire un rapport de bogues, etc.

/indices/ : Le fichier `Maintainers` et les fichiers `override`.

/project/ : Principalement des ressources pour les développeurs, comme :

project/experimental/ : Ce répertoire contient des paquets et des outils qui sont en développement, et sont encore en état de test alpha. Les utilisateurs ne devraient pas utiliser des paquets de ce répertoire parce qu'ils peuvent être dangereux même pour des utilisateurs expérimentés.

project/orphaned/ : Paquets qui ont été abandonnés par leur ancien responsable et ont été retirés de la distribution.

2.1.2 Distributions Debian

Normalement il y a trois distributions Debian dans le répertoire `dists`. Leurs noms sont la distribution « stable », la distribution « testing » et la distribution « unstable ». Quelquefois il y a aussi la distribution « frozen ». Chaque distribution est définie par un lien symbolique vers le répertoire réel, utilisant un nom de code et situé dans le répertoire `dists`.

2.1.3 La distribution `stable`

Les paquets de la distribution `stable`, Debian Woody (3.0r0), sont enregistrés dans le répertoire `stable` (lien symbolique vers `Woody`) :

- `stable/main/` : Ce répertoire contient les paquets constituant la version la plus récente du système Debian.
  Ces paquets sont aussi conformes aux Principes du logiciel libre selon Debian (http://www.debian.org/social_contract#guidelines) (aussi disponible dans le fichier `/usr/share/doc/debian/social-contract.txt` installé par le paquet `debian-doc`), et sont tous utilisables et redistribuables librement.
- `stable/non-free/` : Ce répertoire contient des paquets dont la distribution est restreinte et nécessite que les distributeurs prennent soigneusement en compte les exigences spécifiées par la licence.
  Par exemple, certains paquets ont une licence qui interdit la distribution commerciale. D'autres peuvent être redistribués mais sont en fait des partagiciels et non des logiciels libres. Les licences de chacun de ces paquets doivent être étudiées, et dans certains cas négociées, avant que les paquets soient inclus dans une redistribution (par exemple, sur un CD-ROM).

– `stable/contrib/` : Ce répertoire contient des paquets qui sont conformes aux principes du logiciel libre selon Debian et **distribuables librement**, mais dépendent d'un paquet qui **n'est pas** distribuable librement et n'est ainsi disponible que dans la section non-free.

En plus des emplacements ci-dessus, les paquets sont physiquement situés dans le répertoire `pool` ('Le répertoire `pool`' page 9).

L'état courant de la distribution `stable` est accessible sur la page web Les problèmes de 'stable' (`http://ftp-master.debian.org/testing/stable_probs.html`).

2.1.4 La distribution `testing`

Les paquets de la distribution `testing`, Debian Sarge, sont enregistrés dans le répertoire `testing` (lien symbolique vers `Sarge`) après avoir subi une certaine quantité de tests dans `unstable`. En plus de ces emplacements, les nouveaux paquets sont situés dans le répertoire `pool` ('Le répertoire `pool`' page 9). Les sous-répertoires `main`, `contrib` et `non-free` sont aussi présents dans `testing`, séparés par les mêmes critères que pour `stable`.

Les paquets doivent être synchronisés pour toutes les architectures où ils sont compilés et ne doivent pas avoir de dépendances qui les rendent ininstallables ; ils doivent aussi avoir moins de bogues critiques pour une sortie de version que ceux de `unstable`. De cette façon, on espère que `testing` est toujours prête à être candidate à une sortie. Plus de détails sur le méchanisme sont disponibles à `http://ftp-master.debian.org/testing/`.

L'état courant de la distribution `testing` est accessible sur les sites suivants (en Anglais) :
- update excuses (`http://ftp-master.debian.org/testing/update_excuses.html`)
- testing problems (`http://ftp-master.debian.org/testing/testing_probs.html`)
- release-critical bugs (`http://bugs.debian.org/release-critical/`)
- base system bugs (`http://base.debian.net/`)
- bugs in standard and task packages (`http://standard.debian.net/`)
- other bugs and bug-squashing party notes (`http://bugs.debian.net/`)

2.1.5 La distribution `unstable`

Les paquets de la distribution `unstable`, `sid`, sont enregistrés dans le répertoire `unstable` après avoir été téléchargés dans l'archive Debian et y restent jusqu'à ce qu'ils soient déplacés dans `testing` après quelque temps. Les nouveaux paquets sont situés dans le répertoire `pool` 'Le répertoire `pool`' page 9. Les sous-répertoires `main`, `contrib` et `non-free` sont aussi présents dans `unstable`, et ont les mêmes fonctions que dans `stable`.

La distribution `unstable` contient une image du système en développement le plus récent. Les utilisateurs sont encouragés à utiliser et tester ces paquets, mais sont prévenus de leur état. L'avantage à utiliser `unstable` est que vous êtes toujours à jour avec la dernière version du projet Debian—mais si ça casse, vous en découvrez les désavantages :-)

L'état courant de la distribution `unstable` est accessible à la page web : Problèmes de unstable (http://ftp-master.debian.org/testing/unstable_probs.html).

2.1.6 La distribution `frozen`

Lorsque la distribution `testing` est mûre, elle est gelée (NdT : *frozen* en Anglais), c'est-à-dire que l'on n'accepte plus de nouveau code, seulement des corrections de bogues, si nécessaire. De plus, un nouvel arbre testing est créé dans le répertoire `dists`, avec un nouveau nom de code. La distribution frozen subit quelques mois de test, avec par intermittence des mises à jour et des gelées complètes, ce qu'on appelle des 'cycles de test'. (Le récent processus de sortie de `woody` n'a pas vu la création d'un lien symbolique `frozen`, ainsi `frozen` n'était pas une distribution, juste une étape de développement de `testing`).

On garde une trace des bogues de la distribution frozen qui peuvent retarder la sortie d'un paquet ou qui peuvent retarder la sortie de la distribution complète. Lorsque le nombre de bogues descend en dessous des valeurs maximum acceptables, la distribution frozen devient stable, est sortie, et la distribution stable précédente devient obsolète (et est déplacée dans les archives).

2.1.7 Les noms de code de la distribution Debian

Les noms des répertoires physiques dans le répertoire `dists`, comme `Woody` et `Sarge`, sont juste des noms de code. Lorsqu'une distribution Debian est en développement, elle n'a pas de numéro de version mais un nom de code. Le but de ces noms de code est de faciliter le travail des mirroirs de la distribution Debian (si un répertoire réel comme `unstable` changeait soudainement son nom en `stable`, beaucoup de données seraient à télécharger de nouveau).

Actuellement, `stable` est un lien symbolique vers `Woody` et `testing` est un lien symbolique vers `Sarge`. Cela signifie que `Woody` est l'actuelle distribution stable et `Sarge` l'actuelle distribution testing.

`unstable` est un lien symbolique permanent vers `sid`, car `sid` est toujours la distribution unstable.

2.1.8 Noms de code utilisés par le passé

Les autres noms de code qui ont déjà été utilisés sont : `buzz` pour la version 1.1, `rex` pour la version 1.2, `bo` pour les versions 1.3.x, `hamm` pour la version 2.0, `slink` pour la version 2.1 et `potato` pour la version 2.2.

2.1.9 Source d'inspiration pour les noms de code

Jusqu'ici, les noms de code viennent des caractères du film *Toy Story* par Pixar.

- **buzz** (Buzz Lightyear) est le cosmonaute,
- **rex** est le tyranosaure,
- **bo** (Bo Peep) est la fille qui s'occupe du mouton,
- **hamm** est la tirelire en forme de cochon,
- **slink** (Slinky Dog) est le chien,
- **sarge** est un chef des Hommes de l'Armée de Plastique Vert
- **potato** est, bien sûr, Mr. Potato
- **woody** est le cowboy,
- **sid** est le garçon d'à côté qui détruit les jouets.

2.1.10 Le répertoire `pool`

Historiquement, les paquets étaient gardés dans le sous-répertoire `dists` correspondant à la distribution qui les contenait. Il apparut que cela posait certains problèmes, tels que la grande consommation de bande passante sur les mirroirs lorsque des changements majeurs étaient effectués.

Les paquets sont maintenant gardés dans un large 'bassin' (NdT : pool en Anglais), structuré selon le nom du paquet source. Pour rendre cela gérable, le bassin est subdivisé par section (`main`, `contrib` et `non-free`) et par la première lettre du nom du paquet source. Ces répertoires contiennent plusieurs fichiers : les paquets binaires pour chaque architecture, et les paquets source à partir desquels les paquets binaires ont été générés.

Vous pouvez trouver où se trouve chaque paquet en lançant une commande comme `apt-cache showsrc` *`mypackagename`* et en lisant la ligne 'Directory :'. Par exemple, les paquets `apache` sont dans `pool/main/a/apache/`. Il y a tellement de paquets `lib*` qu'ils sont traités différemment : par exemple, les paquets `libpaper` sont dans `pool/main/libp/libpaper/`.

Les répertoires `dists` sont toujours utilisés pour les fichiers d'index utilisés par des logiciels comme `apt`. De plus, les anciennes distributions n'ont pas été converties pour utiliser les bassins donc vous verrez des chemins contenant des distributions comme `Potato` ou `Woody` dans le champ d'en-tête « Filename ».

Normalement, vous n'avez pas à vous occuper de cela, puisque le nouvel `apt` et probalement l'ancien `dpkg-ftp` (voir 'Méthodes de mise à jour d'un système Debian' page 20) vont gérer cela de façon transparente. Si vous souhaitez plus d'information, consultez Debian Package Pools FAQ (`http://people.debian.org/~joeyh/poolfaq`) (en Anglais).

2.1.11 Notes historiques sur `sid`

Lorsque la `sid` d'aujourd'hui n'existait pas, l'organisation de l'archive Debian avait un défaut majeur : on supposait que lorsqu'une architecture était créée dans la distribution `unstable` courante, elle sortirait lorsque cette distribution deviendrait la nouvelle `stable`. Ce n'était pas le cas pour beaucoup d'architectures, ce qui entrainait que ces répertoires devaient être déplacés lors d'une sortie. Cela n'était pas pratique parce que cela consommerait beaucoup de bande passante.

Les administrateurs de l'archive contournèrent le problème pendant plusieurs années en plaçant les binaires des architectures non sorties dans un répertoire spécial nommé `Sid`. Lors de la sortie de ces architectures, un lien était créé entre la `stable` courante et `Sid`, et à partir de là elles étaient créées dans l'arbre `unstable` de façon normale. Cette disposition était quelque peu troublante pour les utilisateurs.

Avec l'arrivée des bassins de paquets (voir 'Le répertoire `pool`' page précédente) pendant le développement de la distribution `Woody`, les paquets binaires ont commencé à être stockés à un emplacement standard dans le bassin, quelle que soit la distribution, de façon à ce que sortir une distribution ne cause plus de grande consommation de bande passante sur les mirroirs (il y a, cependant, beaucoup de consommation de bande passante, mais graduellement, pendant le développement).

2.1.12 Paquets téléchargés dans `incoming`

Les paquets téléchargés sont d'abord placés dans `http://incoming.debian.org/` avant que l'on ne vérifie s'ils viennent bien d'un développeur Debian (et sont placés dans le sous-répertoire `DELAYED` dans le cas d'un téléchargement par un non responsable (Non-Maintainer Upload, NMU)). Une fois par jour, ils sont déplacés de `incoming` vers `unstable`.

En cas d'urgence, vous pouvez vouloir installer des paquets de `incoming` avant qu'ils n'atteignent `unstable`.

2.1.13 Récupérer un paquet ancien

Alors que les distributions Debian récentes sont gardées dans le répertoire `debian` de chaque mirroir Debian (`http://www.debian.org/misc/README.mirrors`), les archives des anciennes distribution comme `Slink` sont gardées sur `http://archive.debian.org/` ou dans le répertoire `debian-archive` de chaque mirroir Debian.

Les anciens paquets de `testing` et `unstable` sont situés à `http://snapshot.debian.net/`.

2.1.14 Sections architectures

Dans chacun des arbres de répertoires majeurs (`dists/stable/main`,`dists/stable/non-free`, `dists/unstable/main/`, etc.), les paquets binaires résident dans des sous-répertoires dont le nom indique l'architecture pour laquelle ils ont été compilés.

- `binary-all/`, pour les paquets indépendants de l'architecture Cela inclut, par exemple, des scripts Perl, ou de la documentation pure.
- `binary-`*`platform`*`/`, pour les paquets qui s'exécutent sur une plateforme particulière.

Veuillez noter que les paquets binaires pour `testing` et `unstable` ne résident plus dans ces répertoires, mais dans le répertoire de haut niveau `pool`. Les fichiers d'index (`Packages` et `Packages.gz`) ont été gardés, cependant, pour une compatibilité arrière.

Pour les architectures binaires supportées, consultez les Notes de version de chaque distribution. Elles sont disponibles sur les sites des notes de version pour stable (`http://www.debian.org/releases/stable/releasenotes`) et testing (`http://www.debian.org/releases/testing/releasenotes`).

2.1.15 Le code source

Le code source est inclut pour tout le système Debian. De plus, les termes de la licence de la plupart des logiciels du système *requièrent* que le code source soit distribué avec le programme, ou qu'une offre permettant d'obtenir le code source accompagne le programme.

Normalement, le code source est distribué dans les répertoires `source`, qui sont parallèles aux répertoires contenant les binaires spécifiques à une architecture, ou plus récemment dans le répertoire `pool` (voir 'Le répertoire `pool`' page 9). Pour récupérer le code source sans avoir à être familier avec la structure de l'archive Debian, essayez une commande comme `apt-get source` *`mypackagename`*.

Certains paquets, notamment `pine`, sont seulement disponibles sous forme de paquet source, à cause de limitations de leur licence. (Récemment, le paquet `pine-tracker` a été fourni pour faciliter l'installation de Pine.) Les procédures décrites dans 'Porter un paquet vers le système `stable`' page 69 et 'Paquetage' page 152 permettent de construire un paquet manuellement.

Le code source peut être ou ne pas être disponible pour les paquets dans les répertoires `contrib` et `non-free`, qui ne font pas formellement partie du système Debian.

2.2 Système de gestion des paquets Debian

2.2.1 Vue générale des paquets Debian

Les paquets contiennent généralement tous les fichiers nécessaires pour implémenter un ensemble de commandes ou caractéristiques. Il existe deux types de paquets Debian :

- Les **paquets binaires**, qui contiennent des exécutables, des fichiers de configuration, des pages de man/info, la licence, et d'autres documentations. Ces paquets sont distribués dans un format d'archive spécifique à Debian (voir 'Format des paquets Debian' de la présente page) ; on les reconnaît habituellement à leur extension `.deb`. Les paquets binaires peuvent être dépaquetés en utilisant l'utilitaire Debian `dpkg` ; les détails sont fournis dans sa page de manuel.
- Les **paquets sources**, qui consistent en un fichier `.dsc` décrivant le paquet source (y compris le nom des fichiers suivants), un fichier `.orig.tar.gz` qui contient le source original non-modifié compressé par tar et gzip, et habituellement un fichier `.diff.gz` qui contient les modifications du source original spécifiques à Debian. L'utilitaire `dpkg-source` empaquète et dépaquète les archives source Debian ; les détails sont fournis dans sa page de manuel.

L'installation de logiciels par le système de paquets utilise des « dépendances » qui sont soigneusement conçues par les responsables du paquet. Ces dépendances sont documentées dans le fichier `control` associé à chaque paquet. Par exemple, le paquet contenant le compilateur GNU C (`gcc`) « dépend » du paquet `binutils` qui inclut l'éditeur de liens et l'assembleur. Si un utilisateur tente d'installer `gcc` sans avoir d'abord installé `binutils`, le système de gestion de paquets (dpkg) renverra un message d'erreur disant qu'il a besoin de `binutils`, et cessera l'installation de `gcc`. (Cependant, un utilisateur insistant pourra passer outre ; voir `dpkg(8)`.) Pour plus de détails, voir 'Dépendances des paquets' page 16 ci-dessous.

Les outils de paquetage de Debian peuvent être utilisés pour :

- manipuler et gérer des paquets ou des parties de paquets,
- aider l'utilisateur à découper des paquets qui doivent être transmis à travers un média de taille limitée comme une disquette,
- aider les développeurs à construire des archives de paquets, et
- aider les utilisateurs à installer des paquets qui se trouvent sur un site d'archive Debian.

2.2.2 Format des paquets Debian

Un « paquet » Debian, ou un fichier d'archive Debian, contient les fichiers exécutables, les bibliothèques, et la documentation associés à un programme particulier ou un ensemble de programmes liés. Normalement, une archive Debian possède un nom de fichier se terminant par `.deb`.

Les données internes de ce format de paquets binaires Debian sont décrites dans la page de manuel `deb(5)`. Parce que ce format interne est sujet à des changements (entre les sorties majeures de Debian), utilisez toujours `dpkg-deb(8)` pour manipuler des fichiers `.deb`.

Au moins jusqu'à la distribution Woody, tous les fichiers d'archive Debian étaient manipulés par les commandes Unix standard `ar` et `tar`, même lorsque les commandes dpkg n'étaient pas disponibles.

2.2.3 Conventions de nommage pour les fichiers de paquets Debian

Les noms de fichiers des paquets Debian se conforment à la convention suivante :

foo_VersionNumber-DebianRevisionNumber.deb

où *foo* représente le nom du paquet. Pour vérification, on peut déterminer le nom du paquet associé à un fichier d'archive Debian particulier (fichier `.deb`) de l'une des façons suivantes :

- inspecter le fichier `Packages` dans le répertoire où il était stocké sur un site d'archive Debian. Ce fichier contient une description de chaque paquet ; le premier champ de chaque paragraphe est le nom de paquet formel.
- utiliser la commande `dpkg --info` *foo_VVV-RRR*`.deb` (où *VVV* et *RRR* sont les numéros de version et de révision du paquet en question, respectivement). Cela affiche, entre autres, le nom du paquet correspondant au fichier d'archive dépaqueté.

La composante `VVV` est le numéro de version spécifié par le développeur original. Il n'y a aucune norme spécifiant la numérotation des versions, donc elle peut avoir des formats aussi différents que « 19990513 » et « 1.3.8pre1 ».

La composante `RRR` est le numéro de révision Debian spécifié par le développeur Debian (ou un utilisateur s'il choisit de construire le paquet lui-même). Ce numéro correspond au niveau de révision du paquet Debian ; ainsi, un nouveau niveau de révision correspond habituellement à un changement dans le Makefile Debian (`debian/rules`), le fichier de contrôle Debian (`debian/control`), les scripts d'installation ou de suppression (`debian/p*`), ou les fichiers de configuration utilisés avec le paquet.

2.2.4 Préservation de la configuration locale

La préservation des fichiers configurables par l'utilisateur est activée par le mécanisme « conffiles » de Debian. Les fichiers de configuration de l'utilisateur (habituellement placés dans `/etc`) sont spécifiés dans le fichier `conffiles` du système de paquets Debian. Le système de gestion des paquets garantie que ces fichiers ne seront pas recouverts lors de la mise à jour d'un paquet.

Lorsqu'il est possible de configurer le système sans modifier les fichiers qui appartiennent aux différents paquets Debian, il est conseillé de ne pas les modifier même si ce sont des « conffiles ». Cela permet des opérations de mise à jour plus rapides et en douceur.

Pour déterminer exactement quels sont les fichiers préservés lors d'une mise à jour, lancez la commande :

```
dpkg --status package
```

et regardez la ligne « Conffiles : ».

Les détails du contenu d'un fichier Debian `conffiles` sont fournis dans la Charte Debian, section 11.7 (voir 'Références' page 157).

2.2.5 Scripts de maintenance Debian

Les scripts de maintenance Debian sont des scripts exécutables qui sont automatiquement exécutés avant ou après l'installation d'un paquet. Avec un fichier nommé `control`, tous ces fichiers font partie de la section « control » d'un fichier d'archive Debian.

Les fichiers individuels sont :

preinst Ce script est exécuté avant que son paquet soit dépaqueté de son archive Debian (`.deb`). Beaucoup de scripts « preinst » arrêtent les services fournis par les paquets mis à jour jusqu'à ce que leur installation ou mise à jour soit complète (après l'exécution avec succès du script « postinst »).

postinst Ce script complète la configuration requise par un paquet après son dépaquetage à partir de son archive Debian (`.deb`). Souvent, les scripts « postinst » demandent à l'utilisateur d'entrer des informations et/ou l'avertissent que s'il accepte les valeurs par défaut, il devrait se rappeler de revenir en arrière et reconfigurer le paquet lorsque la situation le requiert. Beaucoup de scripts « postinst » exécutent ensuite les commandes nécessaires au redémarrage d'un service une fois que le nouveau paquet a été installé ou mis à jour.

prerm Ce script arrête les daemons qui sont associés à un paquet. Il est exécuté avant la suppression de fichiers associés au paquet.

postrm Ce script modifie les liens ou les autres fichiers associés à un paquet, et/ou supprime les fichiers créés. (Voir aussi 'Paquets virtuels' page ci-contre.)

Actuellement, tous les fichiers de contrôle peuvent être trouvés dans le répertoire `/var/lib/dpkg/info`. Les fichiers associés au paquet `foo` commencent avec le nom « foo » et ont des extensions « preinst », « postinst », etc., tel qu'approprié. Le fichier `foo.list` dans ce répertoire liste tous les fichiers qui ont été installés avec le paquet `foo`. (Notez que l'emplacement de ces fichiers est interne à dpkg, et peut changer.)

2.2.6 Priorité des paquets

Chaque paquet Debian se voit assigner une **priorité** par les responsables de la distribution, comme aide au système de gestion des paquets. Les priorités sont :

- Les paquets **Required** (requis) sont nécessaires au bon fonctionnement du système.
  Ceci inclut tous les outils nécessaires pour réparer les défauts du système. Vous ne devez pas supprimer ces paquets, sinon le système peut devenir complètement planté et vous ne pourrez probablement plus utiliser dpkg pour remettre les choses en place. Un système avec seulement les paquets requis ne sera probablement pas utilisable, mais il sera suffisament fonctionnel pour que l'administrateur le démarre et installe plus de logiciels.
- Les paquets **Important** devraient se trouver sur n'importe quel système de type Unix.
  D'autres paquets sans lesquel le système ne fonctionnera pas bien ou ne sera pas utilisable se trouveront ici. Cela n'inclut *PAS* Emacs ou X11 ou TeX ou n'importe quelle autre grosse application. Ces paquets constituent seulement une infrastructure de base.
- Les paquets **Standard** sont standard sur n'importe quel système Linux, et comprennent un système en mode texte raisonnablement petit mais pas trop limité.
  C'est ce qui sera installé par défaut si les utilisateurs ne sélectionnent rien d'autre. Cela n'inclut pas beaucoup de grosses applications, mais cela inclut Emacs (qui est plus une partie d'infrastructure qu'une application) et un sous-ensemble raisonnable de TeX et LaTeX (si cela est possible sans X).
- Les paquets **Optional** (optionnel) incluent tous ceux que vous pourriez raisonnablement vouloir installer même s'ils ne vous sont pas familiers, et si vous n'avez pas de besoins spécifiques.
  Cela inclut X11, une distribution complète de TeX, et beaucoup d'applications.
- Les paquets **Extra** (en plus) sont des paquets qui soit entrent en conflit avec des paquets ayant une priorité plus haute, soit ne seront utiles que si vous les connaissez, soit ont besoin de prérequis spécifiques qui les rendent peu convenables pour « Optional ».

2.2.7 Paquets virtuels

Un paquet virtuel est un nom générique qui s'applique à n'importe quel paquet d'un groupe de paquets, qui tous fournissent une fonctionalité de base similaire. Par exemple, les logiciels `tin` et `trn` sont des lecteurs de groupes de discussion, et doivent donc satisfaire la dépendance d'un programme ayant besoin d'une tel lecteur sur le système pour fonctionner ou être utile. On dit qu'ils fournissent tous les deux le « paquet virtuel » appelé `news-reader`.

De façon similaire, `exim` et `sendmail` fournissent tous les deux la fonctionalité d'un agent de transport de courrier électronique. On dit donc qu'ils fournissent le paquet virtuel « mail transport agent ». Si l'un des deux est installé, un programme dépendant de l'installation d'un `mail-transport-agent` sera satisfait par la présence de ce paquet virtuel.

Debian fournit un mécanisme pour que, si plus d'un paquet qui fournit le même paquet virtuel est installé sur un système, l'administrateur puisse configurer l'un des deux comme paquet préféré. La commande utilisée est `update-alternatives`, et est décrite dans 'Commandes de rechange' page 72.

2.2.8 Dépendances des paquets

Le système de paquets Debian possède une série de « dépendances » de paquets qui sont conçues pour indiquer (avec un simple drapeau) le niveau auquel Programme A peut fonctionner indépendamment de la présence de Programme B sur le système :

- Paquet A **depends** (dépend) de Paquet B si B doit absolument être installé pour exécuter A. Dans certains cas, A dépend non seulement de B, mais d'une certaine version de B. Dans ce cas, la dépendance sur la version est habituellement une limite basse, dans le sens où A dépend de n'importe quelle version de B plus récente que la version spécifiée.
- Paquet A **recommends** (recommande) Paquet B si le responsable du paquet juge que la plupart des utilisateurs ne voudront pas de A sans avoir la fonctionnalité fournie par B.
- Paquet A **suggests** (suggère) Paquet B si B contient des fichiers qui sont liés à (et habituellement améliorent) la fonctionnalité de A.
- Paquet A **conflicts** (est en conflit) avec Paquet B lorsque A ne fonctionnera pas si B est installé sur le système. Souvent, les conflits sont dans des cas où A contient des fichiers qui fournissent une amélioration par rapport à ceux de B. « conflicts » est souvent associé avec « replaces ».
- Paquet A **replaces** (remplace) Paquet B lorsque les fichiers installés par B sont supprimés et (dans certains cas) recouverts par des fichiers de A.
- Paquet A **provides** (fournit) Paquet B lorsque tous les fichiers et fonctionnalités de B sont incorporés dans A. Ce mécanisme fournit un moyen aux utilisateurs ayant des limitations en espace disque de ne sélectionner que la partie de A dont ils ont réellement besoin.

Plus de détails sur l'utilisation de ces termes sont fournis dans le Manuel de Paquetage et dans la Charte Debian.

Notez que `dselect` permet un contrôle plus précis sur les paquets marqués **recommends** et **suggests** que `apt-get`, qui récupère simplement tous les paquets spécifiés par **depends** et laisse les paquets spécifiés par **recommends** et **suggests**. Les deux programmes utilisent APT comme dorsal dans leurs versions modernes.

2.2.9 Signification de « pre-depends »

« Pre-depends » est une dépendance spéciale. Dans le cas d'un paquet ordinaire, `dpkg` dépaquètera le fichier archive (càd. le fichier `.deb`) indépendamment de la présence ou non des fichiers dont il dépend sur le système. En simplifiant, dépaqueter signifie que `dpkg` extrait les fichiers de l'archive qui sont censés être installés sur votre système et les met à leur place. Si ces paquets **dépendent** de la présence d'autres paquets sur votre système, `dpkg` refusera de compléter l'installation (en exécutant son action « configure ») tant que les autres paquets ne seront pas installés.

Cependant, pour certains paquets, `dpkg` refusera même de les dépaqueter tant que certaines dépendances ne seront pas satisfaites. On dit que ces paquets « pré-dépendent » de la présence d'autres paquets. Le projet Debian fournissait ce mécanisme pour supporter la mise à jour sûre

des systèmes du format `a.out` au format ELF, pendant laquelle **l'ordre** dans lequel les paquets étaient dépaquetés était critique. Il y a d'autres situations de mise à jour pour lesquelles cette méthode est utile, par exemple pour les paquets avec la priorité « required » et leur dépendance à la libc.

Une fois de plus, de plus amples informations peuvent être trouvées dans le Manuel de Paquetage.

2.2.10 Etat d'un paquet

L'état d'un paquet peut être « unknown » (inconnu), « install » (installe), « remove » (supprime), « purge » (purge), ou « hold » (garde). Ces drapeaux « want » (volonté) indiquent ce que l'utilisateur souhaite faire avec un paquet (comme indiqué soit par les actions de l'utilisateur dans la section « Select » de `dselect`, soit par l'invocation directe de `dpkg`).

Leur signification est :
- unknown - (inconnu) l'utilisateur n'a jamais indiqué s'il souhaite le paquet.
- install - (installe) l'utilisateur veut que le paquet soit installé ou mis à jour.
- remove - (supprime) l'utilisateur veut que le paquet soit supprimé, mais ne veut pas supprimer les fichiers de configuration existants.
- purge - l'utilisateur veut que le paquet soit supprimé complètement, y compris ses fichiers de configuration.
- hold - (garde) l'utilisateur veut que le paquet ne soit pas traité, càd. qu'il veut garder la version actuelle dans l'état actuel.

2.2.11 Garder des paquets lors d'une mise à jour

Il y a deux mécanismes pour garder des paquets lors de la mise à jour, à l'aide de `dpkg`, ou, dans Woody, à l'aide d'APT.

Avec `dpkg`, exportez d'abord la liste des sélections de paquets :

```
dpkg --get-selections \* > selections.txt
```

Ensuite, éditez le fichier résultant *`selections.txt`*, en changeant la ligne contenant le paquet que vous souhaitez garder, e.g. `libc6`, de :

```
libc6                        install
```

en :

```
libc6                        hold
```

Sauvegardez le fichier, et rechargez-le dans la base de données de `dpkg` avec la commande :

```
dpkg --set-selections < selections.txt
```

Ou, si vous connaissez le nom du paquet à garder, exécutez simplement :

```
echo libc6 hold | dpkg --set-selections
```

Ce procédé garde les paquets pendant la procédure d'installation de chaque paquet.

Le même résultat peut être obtenu avec `dselect`. Entrez simplement dans l'écran [S]elect, trouvez le paquet que vous souhaitez garder en l'état et appuyez sur la touche '=' (ou 'H'). Les changements prendront effet immédiatement après que vous êtes sortis de l'écran [S]elect.

Le système APT dans la distribution Woody possède un nouveau mécanisme pour garder les paquets pendant la procédure de récupération des archives en utilisant `Pin-Priority`. Voir la page de manuel `apt_preferences(5)`, ainsi que http://www.debian.org/doc/manuals/apt-howto/ ou le paquet `apt-howto`.

2.2.12 Paquets sources

Les paquets sources sont distribués dans un répertoire appelé `source`, et vous pouvez soit les télécharger manuellement, soit utiliser

```
apt-get source foo
```

pour les récupérer (voir la page de manuel `apt-get(8)` pour configurer APT pour faire cela).

2.2.13 Construire des paquets binaires à partir d'un paquet source

Pour un paquet *foo*, vous aurez besoin de tous les fichiers *foo*`_*.dsc`, *foo*`_*.tar.gz` et *foo*`_*.`*diff*`.gz` pour compiler les sources (note : il n'y a pas de fichier `.diff.gz` pour les paquets Debian natifs).

Une fois que vous les avez, si vous avez le paquet `dpkg-dev` installé, la commande

```
$ dpkg-source -x foo_version-revision.dsc
```

va extraire le paquet dans un répertoire appelé *foo-version*.

Si vous souhaitez juste compiler le paquet, vous pouvez entrer dans le répertoire `foo-version` et lancer la commande

```
$ fakeroot debian/rules build
```

pour compiler le programme (vous aurez peut-être besoin d'installer le paquet `fakeroot` d'abord), puis

```
$ fakeroot debian/rules binary
```

en tant que root, pour construire le paquet, et enfin

```
$ su -c "dpkg -i ../foo_version-revision_arch.deb"
```

pour installer le paquet nouvellement construit. Voir 'Porter un paquet vers le système `stable`' page 69.

2.2.14 Créer de nouveaux paquets Debian

Pour une description plus détaillée de la création de nouveaux paquets, lisez le Guide des nouveaux responsables Debian, disponible dans le paquet `maint-guide`, ou à l'adresse http://www.debian.org/doc/manuals/maint-guide/.

2.3 Mettre à jour un système Debian

L'un des buts de Debian est de fournir un chemin de mise à jour consistant et un processus de mise à jour sûr, et nous faisons de notre mieux pour que la mise à jour lors de la sortie d'une nouvelle version depuis une précédente soit la plus douce possible. Les paquets vont alerter l'utilisateur lorsqu'il y a des avertissements importants pendant le processus de mise à jour, et vont souvent fournir une solution à un problème possible.

Vous devriez aussi lire les Notes de version, le document qui décrit les détails des mises à jour spécifiques, livré sur tous les CDs Debian, et disponible sur le WWW aux adresses http://www.debian.org/releases/stable/releasenotes et http://www.debian.org/releases/testing/releasenotes.

Un guide pratique pour les mises à jour est fourni dans la 'Gestion des paquets Debian' page 59. Cette section décrit les détails fondamentaux.

2.3.1 Méthodes de mise à jour d'un système Debian

On pourrait simplement exécuter une session FTP anonyme ou un appel à `wget` vers une archive Debian, parcourir les répertoires jusqu'à ce qu'on trouve le fichier désiré, le récupérer, et enfin l'installer en utilisant `dpkg`. (Notez que `dpkg` installera les fichiers de la mise à jour à leur place, même sur un système en marche.) Parfois, un paquet révisé aura besoin de l'installation d'une version révisée d'un autre paquet, auquel cas l'installation échouera si l'autre paquet n'est pas installé.

Beaucoup de gens trouvent cette approche trop gourmande en temps, car Debian évolue très rapidement — typiquement, une douzaine ou plus de nouveaux paquets sont téléchargés chaque semaine. Ce nombre est encore plus grand avant la sortie d'une version majeure. Pour gérer cette avalanche, beaucoup de gens préfèrent utiliser une méthode automatique. Plusieurs outils de gestion des paquets sont disponibles dans ce but.

2.3.2 Vue générale des outils de gestion de paquets

Le système de gestion de paquets Debian a deux objectifs : la manipulation des fichiers de paquets eux-mêmes et la récupération de fichiers de paquets depuis une archive Debian. `dpkg` réalise la première fonction, APT et `dselect` la seconde.

2.3.3 `dpkg`

C'est le programme principal pour manipuler les fichiers de paquets ; consultez `dpkg(8)` pour une description complète.

`dpkg` vient avec plusieurs programmes primitifs supplémentaires.

- dpkg-deb : Manipule les fichiers `.deb`. `dpkg-deb(1)`
- dpkg-ftp : Une ancienne commande de récupération de fichiers de paquets. `dpkg-ftp(1)`
- dpkg-mountable : Une ancienne commande de récupération de fichiers de paquets. `dpkg-mountable(1)`
- dpkg-split : Scinde un gros paquet en fichiers plus petits. `dpkg-split(1)`

`dpkg-ftp` et `dpkg-mountable` ont été rendus obsolètes par l'introduction du système APT.

2.3.4 APT

APT (*Advanced Packaging Tool*, outil avancé de paquetage) est une interface avancée pour le système de gestion des paquets Debian, qui consiste en plusieurs programmes dont les noms commencent par « apt- ». `apt-get`, `apt-cache` et `apt-cdrom` sont les outils en ligne de commande pour gérer les paquets. Ils fonctionnent aussi en tant que dorsal pour d'autres outils, comme `dselect` et `aptitude`.

Pour plus d'information, installez le paquet `apt` et lisez `apt-get(8)`,`apt-cache(8)`,`apt-cdrom(8)`, `apt.conf(5)`,`sources.list(5)`,`apt_preferences(5)` (woody), et `/usr/share/doc/apt/guide.html/index.html`.

Une autre source d'information est le APT HOWTO (http://www.debian.org/doc/manuals/apt-howto/). Il peut être installé par le paquet `apt-howto` à l'emplacement `/usr/share/doc/apt-howto/en/apt-howto-en.html/index.html`.

`apt-get upgrade` et `apt-get dist-upgrade` récupèrent seulement les paquets marqués « Depends : » et passe outre tous les paquets marqués « Recommends : » et « Suggests : ». Pour éviter cela, utilisez `dselect`.

2.3.5 `dselect`

Ce programme est une interface utilisateur avec un menu pour le système de gestion de paquets Debian. Il est particulièrement utile pour les premières installations et les grosses mises à jour.

Pour plus d'information, installez le paquet `install-doc` et lisez `/usr/share/doc/install-doc/dselect-beginner.en.html` ou Documentation dselect pour débutants (http://www.debian.org/releases/woody/i386/dselect-beginner).

2.3.6 Mise à jour d'un système en marche

Le noyau (système de fichier) des systèmes Debian supporte le recouvrement de fichiers même lorsqu'ils sont en utilisation.

Nous fournissons aussi un programme appelé `start-stop-daemon` qui est utilisé pour démarrer les daemon lors du démarrage du système ou pour les arrêter lorsque le niveau de fonctionnement du noyau est changé (par exemple de multi-utilisateur vers mono-utilisateur ou vers arrêt). Le même programme est utilisé par les scripts d'installation lorsqu'un nouveau paquet contenant un daemon est installé, pour arrêter les daemons en exécution, et les redémarrer lorsque cela est nécessaire.

Notez que le système Debian ne requiert pas l'utilisation du mode mono-utilisateur pour mettre à jour un système en marche.

2.3.7 Fichiers d'archive `.deb` téléchargés et sauvegardés

Si vous avez téléchargé manuellement des fichiers de paquets sur votre disque (ce qui n'est pas forcément nécessaire, voir ci-dessus pour la description de `dpkg-ftp` ou APT), vous pouvez supprimer les fichiers `.deb` de votre système lorsque les paquets ont été installés.

Si APT est utilisé, ces fichiers sont mis en cache dans le répertoire `/var/cache/apt/archives/`. Vous pouvez les effacer après l'installation (`apt-get clean`) ou les copier sur une autre machine dans le répertoire `/var/cache/apt/archives/` pour économiser du temps de téléchargement pendant les installations suivantes.

2.3.8 Garder une trace des mises à jour

`dpkg` garde un enregistrement des paquets qui ont été dépaquetés, configurés, supprimés, et/ou purgés, mais il ne garde pas (pour le moment) de journal de l'activité du terminal qui a eu lieu lorsqu'un paquet a été manipulé.

Le moyen le plus simple de contourner cela est de lancer vos sessions `dpkg`, `dselect`, `apt-get`, etc. avec le programme `script(1)`.

2.4 Le processus de démarrage de Debian

2.4.1 Le programme `init`

Comme tous les Unices, Debian démarre en exécutant le programme `init`. Le fichier de configuration de `init` (qui est `/etc/inittab`) spécifie que le premier script à exécuter doit être `/etc/init.d/rcS`. Ce script lance tous les scripts de `/etc/rcS.d/` en incluant le source ou en forkant un sous-processus, selon leur extension, pour exécuter des initialisations, comme la vérification et le montage des systèmes de fichiers, le chargement des modules, le démarrage des services réseau, le réglage de l'horloge, et l'exécution d'autres initialisations. Ensuite, pour compatibilité, il lance aussi les fichiers (sauf ceux ayant un « . » dans leur nom) de `/etc/rc.boot/`. Les scripts de ce dernier répertoire sont habituellement réservés à l'administrateur système, et leur utilisation dans des paquets est obsolète. Voir 'Astuces sur l'initialisation du système' page 99 pour plus d'information.

2.4.2 Niveaux de fonctionnement

Après le processus de démarrage, `init` exécute les scripts de démarrage situés dans le répertoire correspondant au niveau de fonctionnement par défaut (ce niveau de fonctionnement est donné par l'entrée `id` dans `/etc/inittab`). Comme la plupart des Unices compatibles System V, Linux a 7 niveaux de fonctionnement :

- 0 (arrête le système),
- 1 (mode mono-utilisateur),
- 2 à 5 (différents modes multi-utilisateur), et
- 6 (redémarre le système).

Les systèmes Debian sont livrés avec `id=2`, ce qui indique que le niveau de fonctionnement par défaut sera 2 lorsqu'on entrera dans l'état multi-utilisateur, et les scripts de `/etc/rc2.d/` seront exécutés.

En fait, les scripts des répertoires `/etc/rcN.d/` sont des liens symboliques vers les scripts de `/etc/init.d`. Cependant, les *noms* des fichiers dans chacun des répertoires `/etc/rcN.d/` sont sélectionnés pour indiquer la *façon* dont les scripts de `/etc/init.d/` seront exécutés. Spécifiquement, avant d'entrer dans un niveau de fonctionnement, tous les scripts commençant par 'K' sont lancés ; ils permettent d'arrêter des services. Ensuite, tous les scripts commençant par 'S' sont lancés ; ces scripts permettent de démarrer des services. Le nombre à deux chiffres suivant le 'K' ou le 'S' indique l'ordre dans lequel le script est lancé. Les scripts possédant les nombres les plus petits sont exécutés en premier.

Cette approche fonctionne parce que les scripts dans `/etc/init.d/` prennent tous un argument qui peut être "start", "stop", "reload", "restart" ou "force-reload" et exécuteront la tâche indiquée par cet argument. Ces scripts peuvent être utilisés même après que le système a été démarré, pour contrôler divers processus.

Par exemple, avec l'argument « reload », la commande

```
# /etc/init.d/sendmail reload
```

envoie au daemon sendmail un signal pour qu'il relise son fichier de configuration.

2.4.3 Personnaliser le processus de démarrage

Debian n'utilise pas de répertoire `rc.local` comme BSD pour personnaliser le processus de démarrage ; à la place, il fournit les mécanismes suivants.

Supposons qu'un système ait besoin d'exécuter le script `foo` au démarrage, ou à l'entrée dans un niveau de fonctionnement (System V) particulier. Alors, l'administrateur du système doit :

1. Placer le script `foo` dans le répertoire `/etc/init.d/`.
2. Exécuter la commande Debian `update-rc.d` avec les arguments appropriés, pour effectuer les liens entre les répertoires (spécifiés sur la ligne de commande) `rc ?.d` et `/etc/init.d /foo`. Ici, ? est un nombre entre 0 et 6 qui correspond à l'un des niveaux de fonctionnement System V.
3. Redémarrer le système.

La commande `update-rc.d` effectue les liens entre les fichiers des répertoires `rc ?.d` et le script de `/etc/init.d/`. Chaque lien commencera par un 'S' ou un 'K', suivi par un nombre, suivi par le nom du script. Les scripts commençant par 'S' dans `/etc/rcN.d/` sont exécutés lorsqu'on

entre dans le niveau de fonctionnement `N`. Les scripts commençant par 'K' sont exécutés lorsqu'on quitte le niveau de fonctionnement `N`.

On peut par exemple faire exécuter le script `foo` au démarrage, en le plaçant dans `/etc/init.d/` et en installant les liens par la commande `update-rc.d foo defaults 19`. L'argument `defaults` correspond aux niveaux de fonctionnement par défaut, qui sont 2 à 5. L'argument `19` assure que `foo` sera appelé avant les scripts contenant les nombres supérieurs à 20.

2.5 Support de la diversité

Debian offre plusieurs facilités pour exaucer les voeux des administrateurs du système sans casser ce dernier.

- `dpkg-divert`, voir 'La commande `dpkg-divert`' page 72.
- `equivs`, voir 'Le paquet `equivs`' page 72.
- `update-alternative`, voir 'Commandes de rechange' page 72.
- `make-kpkg` peut s'accomoder de beaucoup de chargeurs . Voir `make-kpkg(1)`.

Les fichiers situés sous `/usr/local/` appartiennent à l'administrateur du système et Debian n'y touchera pas. La plupart (ou tous) les fichiers sous `/etc` sont des `conffiles` (fichiers de configuration) et Debian n'écrira pas dessus lors d'une mise à jour sauf si l'administrateur le spécifie explicitement.

2.6 Internationalisation

Le système Debian est internationalisé et fournit le support pour l'affichage et l'entrée des caractères de beaucoup de langues, à la fois avec la console ou sous X. Beaucoup de documents, de pages de manuel, et de messages système ont été traduits dans un nombre toujours plus élevé de langues. Lors de l'installation, Debian demande à l'utilisateur de choisir une langue pour l'installation (et parfois une variante locale de cette langue).

Si votre système installé ne supporte pas toutes les possibilités de la langue dont vous avez besoin, si vous avez besoin de changer de langue ou d'installer un clavier différent pour supporter votre langue, voyez 'Localisation et support des langues nationales' page 121.

2.7 Debian et le noyau

Voir 'Le noyau Linux et Debian' page 75.

2.7.1 Compiler un noyau avec des sources non Debian

Il faut comprendre la politique Debian sur les en-têtes.

Les bibliothèques C de Debian sont compilées avec les en-têtes du **noyau stable** le plus récent.

Par exemple, la version Debian-1.2 utilisait la version 5.4.13 des en-têtes. Cette pratique contraste avec les paquets source du noyau Linux distribués dans toutes les archives FTP Linux, qui utilisent des versions encore plus récentes des en-têtes. Les en-têtes du noyau distribuées avec le source du noyau sont situées dans `/usr/include/linux/include/`.

Si vous avez besoin de compiler un programme avec des en-têtes du noyau plus récentes que celles fournies par `libc6-dev`, alors vous devez ajouter `-I/usr/src/linux/include/` à la ligne de commande lorsque vous compilez. Cela est arrivé, par exemple, avec l'empaquetage du daemon automounter (`amd`). Lorsque de nouveaux noyaux ont changé les commandes internes ayant trait à NFS, `amd` a dû en prendre connaissance. Cela a requis d'inclure les dernières en-têtes du noyau.

2.7.2 Outils pour compiler un noyau personnalisé

Les utilisateurs qui souhaitent (ou doivent) compiler un noyau personnalisé sont encouragés à télécharger le paquet `kernel-package`. Ce paquet contient le script pour construire le paquet du noyau, et fournit la possibilité de créer un paquet kernel-image Debian en exécutant la commande

```
# make-kpkg kernel_image
```

dans le répertoire le plus haut des sources du noyau. De l'aide est disponible en exécutant la commande

```
# make-kpkg --help
```

et dans la page de manuel `make-kpkg(8)` et 'Le noyau Linux et Debian' page 75.

Les utilisateurs doivent télécharger séparément le code source du dernier noyau (ou le noyau de leur choix) depuis leur archive FTP Linux favorite, à moins qu'un paquet kernel-source-*version* soit disponible (où *version* indique la version du noyau). Le script de démarrage `initrd` de Debian nécessite un patch spécial pour le noyau appelé `initrd`; voir `http://bugs.debian.org/149236`.

Des instructions détaillées pour utiliser le paquet `kernel-package` sont fournies dans le fichier `/usr/doc/kernel-package/README`.

2.7.3 Chargeurs de remplacement

Pour utiliser un autre chargeur, comme `grub` ou `loadlin`, copier l'image du noyau Linux compilée `bzimage` vers un autre emplacement (par exemple, vers `/boot/grub` ou une partition MS-DOS).

2.7.4 Disquettes de démarrage personnalisées

La création d'une disquette de démarrage personnalisée est aidée par le paquet Debian `boot-floppies`, qui se trouve normalement dans la section `admin` de l'archive FTP Debian. Les scripts shell de ce paquet produisent des disquettes de démarrage au format `syslinux`. Ce sont des disquettes formatées au format MS-DOS dont le « Master Boot Record » a été modifié pour qu'elles démarrent Linux directement (ou n'importe quel système d'exploitation défini dans la fichier `syslinux.cfg` sur la disquette). D'autres scripts dans ce paquet produisent des disquettes racines d'urgence et peuvent même reproduire les disquettes de base.

Vous trouverez plus d'information à ce propos dans le fichier `/usr/doc/boot-floppies/README` après avoir installé le paquet `boot-floppies`.

2.7.5 Dispositions spéciales pour manipuler les modules

Le paquet Debian `modconf` fournit un script shell (`/usr/sbin/modconf`) qui peut être utilisé pour personnaliser la configuration des modules. Ce script présente une interface à base de menus, demandant à l'utilisateur les pilotes de périphériques présents sous forme de modules chargeables qu'ils souhaite utiliser sur son système. Les réponses sont utilisées pour personnaliser le fichier de configuration `/etc/modules.conf` (qui liste les alias, et autres arguments qui doivent être utilisés par les différents modules) grâce aux fichiers `/etc/modutils/`, et `/etc/modules` (qui liste les modules qui doivent être chargés lors du démarrage).

Comme les (nouveaux) fichiers Configure.help qui sont maintenant disponibles pour supporter la compilation de noyaux personnalisés, le paquet `modconf` est livré avec une série de fichiers d'aide (dans `/usr/lib/modules_help/`) qui fournissent des informations détaillées sur les arguments possibles pour chacun des modules. Voir 'Le noyau 2.4 modulaire' page 77 pour des exemples.

2.7.6 Désinstaller le paquet d'un vieux noyau

Le script `kernel-image-NNN.prerm` vérifie que le noyau que vous exécutez actuellement n'est pas le noyau à désinstaller. Ainsi, vous pouvez supprimer de façon sûre les noyaux dont vous ne voulez plus avec cette commande :

```
dpkg --purge --force-remove-essential kernel-image-NNN
```

(Remplacez *NNN* par la version et la révision de votre noyau, bien sûr.)

Chapitre 3

Installation du système Debian

La documentation officielle pour installer Debian est située à http://www.debian.org/releases/stable/, et http://www.debian.org/releases/stable/installmanual.

Sa version de développement est située à http://www.debian.org/releases/testing/, et http://www.debian.org/releases/testing/installmanual (en développement, parfois elle n'existe pas).

Bien que « Guide de référence pour Debian » ait été commencé à l'époque de Potato, la majeure partie de son contenu a été mis à jour pour Debian Woody (3.0r0) et Debian Sarge.

3.1 Astuces générales sur l'installation du système Linux

En vue de minimiser les risques associés aux paquets de « testing » et « unstable », c'est une bonne habitude de configurer votre système Linux principal en dual boot avec un autre système Linux plus petit et stable.

3.1.1 Compatibilité matérielle

Linux est compatible avec la plupart des matériels PC. Linux peut être installé sur presque tout système. Pour moi il a été aussi facile à installer que Windows 95/98/Me. Il semble que la liste des compatibilités matérielles s'allonge encore.

Si vous possédez un PC portable, consultez : Linux on Laptops (http://www.linux-laptop.net/) pour identifier sa situation.

Ma recommandation pour le matériel type PC de bureau est « soyez juste conservateur » :

- SCSI plutôt qu'IDE pour le travail, disque dur IDE/ATAPI pour l'utilisation privée.

- CD-ROM (ou CD-RW) IDE/ATAPI.
- PCI plutôt qu'ISA, surtout pour la carte réseau.
- Utilisez une carte réseau de base. Des cartes Tulip sur bus PCI, NE2000 sur bus ISA conviennent bien.
- Évitez PCMCIA (portables) lors de la première tentative d'installation de Linux.
- Pas de clavier ou de souris USB, . . .à moins de vouloir faire face à un défi.

Pour une machine lente, extraire son disque dur et le connecter dans une autre machine plus rapide pour accélérer l'installation est une bonne idée.

3.1.2 Déterminer le matériel et les puces du PC

Durant l'installation, on sera interrogé sur le matériel ou les puces. Parfois, ces informations ne sont pas toujours faciles à trouver. Voici une méthode :

1. Ouvrez le PC et inspectez l'intérieur.
2. Notez les chiffres qui sont sur la grande puce de la carte graphique, de la carte réseau, sur la puce à côté des ports série et la puce à côté des ports IDE.
3. Notez les noms des cartes imprimés au dos des cartes PCI et ISA.

3.1.3 Trouver le matériel du PC avec Debian

Les commandes suivantes sur un système Linux devraient donner quelqu'idée sur le matériel présent et la configuration.

```
$ /usr/sbin/lspci -v |less
$ pager /proc/pci
$ pager /proc/interrupts
$ pager /proc/ioports
```

Ces commandes peuvent être lancées pendant le processus d'installation à partir de la console en appuyant sur ALT-F2.

3.1.4 Trouver le matériel du PC avec d'autres systèmes d'exploitation (SE)

D'autres sources d'information concernant le matériel peuvent être obtenues avec d'autres SE.

Installez une autre distribution Linux commerciale. La détection du matériel tend à être meilleure que celle de Debian, pour l'instant. Ceci peut changer avec la nouvelle Woody.

Installez Windows. La configuration matérielle peut être obtenue en faisant un clic droit sur « Poste de travail » pour aller sur Propriétés / Gestionnaire de périphériques. Enregistrez toutes

les ressources d'information comme IRQ, I/O port address, DMA. Quelques vieilles cartes ISA doivent être configurées sous DOS et utilisées en conséquence.

3.1.5 Le Mythe Lilo

Lilo est limité à 1024 cylindres. —FAUX !

Les versions récentes de `lilo` utilisées depuis Debian Potato supportent lba32. Si le BIOS de la carte mère est assez récent pour supporter lba32, `lilo` devrait être capable de charger au-delà de la vieille limite des 1024 cylindres.

Assurez-vous simplement d'ajouter la ligne « lba32 » vers le début de votre fichier `lilo.conf` si vous avez gardé un vieux `lilo.conf`.

3.1.6 Choix des disquettes de boot

Pour Potato, j'aimais bien les disquettes IDEPCI pour installer une machine de bureau. Pour Woody, j'aime bien les disquettes bf2.4. Elles utilisent une version de `boot-floppies` pour créer les disquettes de démarrage.

Si vous avez une carte réseau PCMCIA, vous devez utiliser les disquettes de démarrage standard (plus grand nombre de disquettes, mais tous les pilotes de périphériques sont disponibles) et configurer la carte réseau dans le dialogue PCMCIA ; n'essayez pas de la configurer dans le dialogue de configuration réseau standard.

Pour des systèmes spéciaux, vous pouvez avoir besoin de créer une disquette de secours personnalisée. Cela peut être fait en remplaçant l'image du noyau appelée « linux » sur la disquette de secours Debian par une autre image compressée du noyau compilée ailleurs pour la machine. Les détails sont documentés dans le fichier `readme.txt` de la disquette de secours. La disquette est formattée au format MSDOS, aussi vous pouvez utiliser n'importe quel système pour lire et éditer ce fichier. Ceci peut rendre la vie plus facile à ceux qui possèdent une carte réseau spéciale, etc.

Pour Sarge, le paquet `debian-installer` et/ou `pgi` est à utiliser pour créer les disquettes de démarrage.

3.1.7 Installation

Suivez les instructions officielles à `http://www.debian.org/releases/stable/installmanual` ou `http://www.debian.org/releases/testing/installmanual` (en développement, parfois elles n'existent pas).

Si vous installez un système avec les disquettes de démarrage de la distribution de testing, il vous faudra peut-être ouvrir une console en appuyant sur `ALT-F2` et remplacer manuellement les entrées `stable` par `testing` pour ajuster les sources APT.

J'ai l'habitude d'installer `lilo` à un endroit comme `/dev/hda3`, en installant `mbr` sur `/dev/hda`. Cela minimise les risques de recouvrir l'information de démarrage.

Voici ce que je choisis lors du processus d'installation.

- MD5 passwords « yes »
- shadow passwords « yes »
- Installation « advanced » (dselect **) et sélection de
  - exclure emacs (si sélectionné), nvi, tex, telnet, talk(d) ;
  - inclure mc, vim, nano-tiny ou elvis-tiny. Voir '`dselect` – configuration générale' page 68. Même si vous êtes un fan d'Emacs, laissez-le pour le moment et utilisez nano pendant l'installation. De même, évitez d'installer de gros paquets comme TeX (Potato le faisait) à ce point-là. Voir 'Editeurs de sauvetage' page 129 pour la raison de l'installation de nano-tiny ou elvis-tiny ici.
- Pour toutes les questions de configuration pendant l'installation de chaque paquet : « y » (remplace version courante).
- `exim` : sélectionner 2 pour envoyer le courrier électronique par le serveur SMTP du FAI.

Pour plus d'informations sur dselect, voir '`dselect` – configuration générale' page 68.

3.1.8 Nom d'hôte et IP à utiliser pour le réseau local

J'utilise un sous-réseau de classe C à la maison,

```
Internet
   |
   +--- FAI externe fournit le service POP (accédé par fetchmail)
   |
  le FAI fournit les services DHCP et relai SMTP
   |                    :
  Modem Cable        (Téléphone)
   |                    :
Port externe de la passerelle~: eth0 (IP donnée par le DHCP du FAI)
 utilise un vieux PC portable (IBM Thinkpad, 486 DX2 50 MHz, 20 MB RAM)
 tourne avec le noyau Linux 2.4 et le système de fichiers ext3
 tourne avec le paquet «~ipmasq~» (avec fortification, NAT et pare-feu)
 tourne avec le paquet «~dhcp-client~» configuré pour eth0 (remplace les paramèt
 tourne avec le paquet «~dhcp~» configuré pour eth1
 tourne avec «~exim~» comme hôte intelligent (mode 2)
 tourne avec «~fetchmail~» avec un long intervalle (fall back)
 tourne avec «~bind~» comme serveur de nom en cache pour l'Internet depuis le ré
 serveur de nom officiel pour le domaine du réseau local
 tourne avec «~ssh~» sur les ports 22 et 8080 (connexions depuis n'importe où)
```

```
 tourne avec «~squid~» comme serveur de cache pour l'archive Debian (pour APT)
Port interne de la passerelle~: eth1 (IP = 192.168.1.1, fixe)
                          |
         +--- LAN Switch (10 base T) ---+
         |                              |
Quelques clients avec IP fixe    Quelques clients DHCP
(IP = 192.168.1.2-127, fixe)     (IP = 192.168.1.128-200, dynamique)
```

Voir 'Construire une passerelle avec Debian' page 125 pour les détails de la configuration d'une passerelle pour le réseau local.

3.1.9 Comptes utilisateurs

Afin d'avoir une organisation cohérente entre toutes les machines, quelques premiers comptes sont toujours les mêmes sur mon système.

Je crée toujours un premier compte utilisateur avec un nom comme "admin" (uid=1000). Tous les messages de root sont redirigés dessus. Ce compte est membre du groupe `adm` (voir 'Pourquoi `GNU su` ne supporte pas le groupe `wheel`' page 101), à qui l'on peut donner beaucoup des privilèges de root au travers des programmes `su` en utilisant PAM ou `sudo`. Voir 'Ajouter un compte utilisateur' page 48 pour les détails.

3.1.10 Création des systèmes de fichiers

Partition du disque dur

Je préfère avoir différentes partitions pour différentes arborescences de répertoires, afin de limiter les dommages en cas de plantage du système. Par exemple :

```
/          == (/ + /boot + /bin + /sbin)
           == 50Mo+
/tmp       == 100Mo+
/var       == 100Mo+
/home      == 100Mo+
/usr       == 700Mo+ avec X
/usr/local == 100Mo
```

La taille du répertoire `/usr` dépend beaucoup des applications X-Window et de la documentation. `/usr` peut être de 300Mo si l'on n'utilise que le terminal, tandis que 2Go–3Go n'est pas inhabituel si l'on installe un certain nombre d'applications Gnome. Lorsque `/usr` devient trop gros, déplacer

`/usr/share/` vers une autre partition est le meilleur remède. Avec un noyau 2.4 récent, `/` peut avoir besoin de plus de 200Mo.

Par exemple, le statut actuel de la machine qui sert de passerelle Internet est le suivant (sortie de la commande `df -h`) :

```
Filesystem            Size  Used Avail Use% Mounted on
/dev/hda3             300M  106M  179M  38% /
/dev/hda7             100M   12M   82M  13% /home
/dev/hda8             596M   53M  513M  10% /var
/dev/hda6             100M  834k   94M   1% /var/lib/cvs
/dev/hda9             596M  222M  343M  40% /usr
/dev/hda10            596M  130M  436M  23% /var/cache/apt/archives
/dev/hda11            1.5G  204M  1.2G  14% /var/spool/squid
```

(La grande taille de `/var/spool/squid` est pour le proxy en vue du téléchargement des paquetages.)

Ci-dessous la sortie de `fdisk -l` pour donner une idée.

```
# fdisk -l /dev/hda # comment

/dev/hda1             1        41    309928+   6  FAT16 # DOS
/dev/hda2            42        84    325080   83  Linux # (pas utilisé)
/dev/hda3   *        85       126    317520   83  Linux # Principal
/dev/hda4           127       629   3802680    5  Extended
/dev/hda5           127       143    128488+  82  Linux swap
/dev/hda6           144       157    105808+  83  Linux
/dev/hda7           158       171    105808+  83  Linux
/dev/hda8           172       253    619888+  83  Linux
/dev/hda9           254       335    619888+  83  Linux
/dev/hda10          336       417    619888+  83  Linux
/dev/hda11          418       629   1602688+  83  Linux
```

Quelques partitions inutilisées existent. Elles sont réservées pour installer une seconde distribution Linux ou une réserve d'espace pour une arborescence de répertoires en expansion.

Monter les systèmes de fichiers

Montez les systèmes de fichiers ci-dessus proprement avec le fichier `fstab` suivant :

```
# /etc/fstab: information du fichier système statique.
#
# file system mount point type options        dump pass
/dev/hda3 / ext2 defaults,errors=remount-ro 0 1
/dev/hda5 none swap sw 0 0
proc /proc proc defaults 0 0
/dev/fd0 /floppy auto defaults,user,noauto 0 0
/dev/cdrom /cdrom iso9660 defaults,ro,user,noauto 0 0
#
# conserve les partitions séparées
/dev/hda7 /home ext2 rw 0 2
/dev/hda8 /var ext2 rw 0 2
/dev/hda6 /var/lib/cvs ext2 rw 0 2
/dev/hda9 /usr ext2 rw 0 2
/dev/hda10 /var/cache/apt/archives ext2 rw 0 2

# une très grande partition pour le cache proxy
/dev/hda11 /var/spool/squid ext2 rw 0 2

# backup bootable sous DOS
/dev/hda1 /mnt/dos vfat rw,noauto 0 0
# backup bootable sous système Linux (pas fait)
/dev/hda2 /mnt/linux ext2 rw,noauto 0 0
#
# montages nfs
mickey:/ /mnt/mickey nfs ro,noauto,intr 0 0
goofy:/ /mnt/goofy nfs ro,noauto,intr 0 0
# minnie:/ /mnt/minnie smbfs ro,soft,intr,credentials={filename} 0 2
```

Pour NFS, j'utilise `noauto,intr` avec l'option par défaut `hard`. De cette façon, il est possible d'arrêter un processus bloqué par une déconnexion en utilisant Control-C.

Pour une machine sous Windows connectée avec Samba (smbfs), `rw,auto,soft,intr` peut être une bonne idée. Voir 'Configuration Samba' page 39.

Pour un lecteur de disquettes, utiliser `noauto,rw,sync,user,exec` prévient les corruptions de données après une éjection accidentelle du disque sans le démonter, mais cela ralentit l'écriture.

Montage autofs

Points clés pour le montage automatique :

- Charger le module `vfat` pour autoriser `/etc/auto.misc` à contenir `fstype=auto` :

```
# modprobe vfat # avant d'essayer d'accéder à la disquette
 ... ou pour automatiser cela,
# cat >>/etc/modules
vfat
^D
 ... et redémarrer le système
```

– Editer le fichier `/etc/auto.misc` comme suit :

```
floppy -fstype=auto,sync,nodev,nosuid,gid=100,umask=000 :/dev/fd0
 ... où gid=100 est «~users~».
```

– Créer des liens dans `/home/`*`user`*, `cdrom` et `floppy`, qui pointent respectivement vers `/var/autofs/misc/cdrom` et `/var/autofs/misc/floppy`.
– Ajoutez *user* au groupe « users ».

Montage NFS

Le serveur Linux nfs externe (goofy) se trouve derrière un pare-feu (passerelle). J'ai une politique de sécurité très relâchée sur LAN puisque je suis le seul à l'utiliser. Pour obtenir un accès nfs, le côté du serveur nfs a besoin qu'on ajoute `/etc/exports` comme suit :

```
# /etc/exports: la liste des contrôle d'accès pour les systèmes de fichier
# qui peuvent être exportés vers les clients NFS.  Voir exports(5).
/       (rw,no_root_squash)
```

Ceci est utile pour activer le serveur nfs en plus de l'installation et de l'activation d'un client/serveur nfs.

Je crée généralement une seule partition de 2Go pour une installation expérimentale et/ou secondaire et paresseuse de Linux, pour plus de simplicité. Je partage optionnellement les partitions swap et `/tmp` pour ces installations. Le schéma de multi-partitionnement est trop complexe pour ces usages. Si on a besoin d'un système simple utilisé en console, 500Mo peuvent être largement suffisants.

3.1.11 Lignes directrices pour la mémoire DRAM

Ce qui suit sont des indications grossières pour la DRAM.

```
 4 Mo :  Minimum suffisant pour faire fonctionner le noyau Linux.
16 Mo :  Minimum pour un usage du système en mode console.
32 Mo :  Minimum pour un système X simple.
64 Mo :  Minimum pour un système X avec GNOME/KDE.
```

```
128 Mo :  Confortable pour le système X avec GNOME/KDE.
256+Mo :  Pourquoi pas si ça vous chante. La DRAM est bon marché.
```

L'option de boot `mem=4m` (ou lilo `append="mem=4m"`) montrera comment le système se comporterait en ayant 4Mo de mémoire installée. Un paramètre de démarrage pour `lilo` est requis pour un système ayant plus de 64Mo de mémoire avec un vieux BIOS.

3.1.12 Espace de Swap

J'utilise la ligne directrice suivante :

- Chaque partition de swap est < 128 Mo (avec un vieux noyau 2.0), < 2 Go (avec les noyaux récents)
- Total = soit (1 à 2 fois la taille de la RAM) soit (128 Mo à 2 Go)
- Installation sur des disques différents et montage avec les options `sw,pri=1` dans `/etc/fstab`. Cela permet de s'assurer que le noyau fait du RAID par bandes sur les partitions de swap et offre des performances maximum.
- Utilisation de la partie centrale du disque, si possible.

Même si vous n'en avez pas besoin, de l'espace de swap (128 Mo) est requis, sinon le système ralentit avant de planter avec un programme qui manque de mémoire.

3.2 Configuration de Bash

Je modifie les scripts de démarrage à mon goût sur tout le système :

```
/etc/bash.bashrc        Remplace avec un script personnalisé
/etc/profile            Garde copie de la distribution ( \w -> \W)
/etc/skel/.bashrc       Remplace avec une copie privée
/etc/skel/.profile      Remplace avec une copie privée
/etc/skel/.bash_profile Remplace avec une copie privée
~/.bashrc           Remplace avec une copie privée pour tous les comptes
~/.profile          Remplace avec une copie privée pour tous les comptes
~/.bash_profile     Remplace avec une copie privée pour tous les comptes
```

Voir détails dans mon exemple. (`examples/`) J'aime les systèmes transparents, j'ai donc paramétré umask à 002 ou 022.

`PATH` est paramétré par les fichiers de configuration suivants, dans cet ordre.

```
/etc/login.defs  - avant que le shell paramètre PATH
/etc/profile     (peut appeler /etc/bash.bashrc)
~/.bash_profile  (peut appeler ~/.bashrc)
```

3.3 Configuration de la souris

3.3.1 Souris PS/2

Dans le cas d'un connecteur de souris de type PS/2 sur une carte mère ATX, la succession de signaux sera :

```
mouse -> /dev/psaux -> gpm -> /dev/gpmdata = /dev/mouse -> X
```

Ceci permet au clavier ou à la souris d'être déconnecté et réinitialisé en redémarrant gpm après reconnexion. X restera activé !

Pour les souris Logitech 3 boutons PS/2, la configuration sera :

```
/etc/gpm.conf                  /etc/X11/X86Config ou X86Config4
=============================================================
device=/dev/psaux              Section "Pointer"
responsiveness=                  Protocol    "IntelliMouse"
repeat_type=ms3                  Device      "/dev/gpmdata"
type=ps2auto (Woody)
append=""
-------------------------------------------------------------
device=/dev/psaux              Section "Pointer"
responsiveness=                  Protocol    "IntelliMouse"
repeat_type=raw                  Device      "/dev/gpmdata"
type=ps2auto (Woody)
append=""
```

Si une souris normale à 2 boutons PS/2 est utilisée, paramétrez le protocole X à `Microsoft` et validez `Emulate3Buttons`. Pour les souris à roulette, vous pouvez configurer X avec le protocole réel, comme `IMPS/2`. Créez un lien symbolique `/dev/gpmdata` –> `/dev/mouse` pour réussir la configuration de certains utilitaires. Voir les détails dans mon script d'exemple. (`examples/`)

Pour certains portables Toshiba récents : activez `gpm` avant PCMCIA dans le script d'initialisation System-V. Ceci évite à `gpm` de planter. Étrange, mais vrai.

3.3.2 Souris USB

Assurez vous d'avoir :

- « Input Core Support » et « Input Core Support/Mouse Support » activés dans le noyau ou en modules.

– « Support for USB », « Preliminary USB device filesystem », « UHCI » ou « OHCI » et « USB HID Support » activés dans le noyau ou en modules.

Si vous n'utilisez pas devfs, créez une node de périphérique `/dev/input/mice` avec comme major 13 et minor 63, comme suit :

```
# cd /dev
# mkdir input
# mknod input/mice c 13 63
```

Pour les souris à 3 boutons Logitech, la configuration devrait être :

```
/etc/gpm.conf                  /etc/X11/X86Config or X86Config-4
=================================================================
device=/dev/input/mice         Section "InputDevice"
responsiveness=                  Option "Protocol" "ImPS/2"
repeat_type=ms3                  Option "Device" "/dev/input/mice"
type=ps2                         Option "ZAxisMapping" "4 5"
append=""                      EndSection
```

[Cette section sur les souris USB a été écrite par Jan Michael C Alonzo <`jmalonzo@softhome.net`>.]

Voir Linux USB Project (`http://www.linux-usb.org/`) pour plus d'information.

3.4 Configuration NFS

Configurer NFS dans `/etc/exports`.

```
# echo "/ *.domainname-for-lan-hosts(rw,no_root_squash,nohide)" \
        >> /etc/exports
```

Voir détails dans mon exemple. (`examples/`).

3.5 Configuration Samba

Références :

– `http://www.samba.org/`

– paquet `samba-doc`

Configurer Samba en mode « partage » est plus facile pour créer un partage du disque de type WfW. Mais il est mieux de le configurer en mode « utilisateur ».

Samba peut être configuré à l'aide de `debconf` ou `vi` :

```
# dpkg-reconfigure --priority=low samba # dans Woody
# vi /etc/samba/smb.conf
```

Voir détails dans mon script d'exemple. (`examples/`)

Ajouter un nouvel utilisateur au fichier smbpasswd peut être fait par `smbpasswd` :

```
$su -c "smbpasswd -a username"
```

Assurez-vous de l'utilisation des mots de passe cryptés pour une meilleure compatibilité.

Le rang de l'OS signifie ce qui suit et plus grand est le nombre, plus haute est la priorité du serveur.

```
0:      Samba avec des comportements relâchés
1:      Wfw 3.1, Win95, Win98, Win/me?
16:     Win NT WS 3.51
17:     Win NT WS 4.0
32:     Win NT SVR 3.51
33:     Win NT SVR 4.0
255:    Samba avec beaucoup de puissance
```

Assurez-vous que les utilisateurs sont membres du groupe possédant le répertoire qui est partagé et que le bit d'exécution du répertoire est positionné à accès.

3.6 Configuration de l'imprimante

La méthode traditionnelle est d'utiliser `lpr/lpd`. Il existe un nouveau système, CUPS™ (Common UNIX Printing System). PDQ est une autre approche. Voir le Linux Printing HOWTO (`http://www.tldp.org/HOWTO/Printing-HOWTO.html`) pour plus d'information.

3.6.1 lpr/lpd

Pour les *spoolers* comme `lpr/lpd` (paquets `lpr`, `lprng`, et `gnulpr`), configurez `/etc/printcap` comme suit s'ils sont connectés à une imprimante PostScript ou texte :

```
lp|alias:\
 :sd=/var/spool/lpd/lp:\
 :mx#0:\
 :sh:\
 :lp=/dev/lp0:
```

Signification des lignes ci-dessus :
- En-tête : *lp* — nom de la queue, *alias* = alias
- mx#0 — taille de fichier maximum illimitée
- sh — suppression de l'impression de la première page d'en-tête
- lp=/dev/lp0 — périphérique local, ou port@host pour une machine distante

C'est une bonne configuration si vous utilisez une imprimante PostScript. De plus, lors de l'impression depuis une machine Windows avec Samba, c'est une bonne configuration pour n'importe quelle imprimante supportée par Windows (communication bi-directionnelle non supportée). Il faut sélectionner l'imprimante correspondante sous Windows.

Si vous n'avez pas d'imprimante PostScript, il faut configurer un système de filtre avec `gs`. Il existe beaucoup d'outils d'auto-configuration pour configurer `/etc/printcap`. Voici quelques options :
- `gnulpr`, (`lpr-ppd`) et `printtool` — j'utilise celle-là.
- `lpr` et `apsfilter`
- `lpr` et `magicfilter`
- `lprng` et `lprngtool`
- `lprng` et `apsfilter`
- `lprng` et `magicfilter`

Pour exécuter des outils de configuration graphiques comme `printtool`, voir 'Devenir root sous X' page 112 pour obtenir les privilèges de root. Les queues d'imprimante créées avec `printtool` utilisent `gs` et agissent comme des imprimantes PostScript. Donc, lorsque vous y accédez, utilisez des pilotes PostScript. Du côté de Windows, « Apple LaserWriter » est le standard.

3.6.2 CUPS™

Installez le Système d'Impression Commun pour UNIX (Common UNIX Printing System, ou CUPS™) :

```
# apt-get install cupsys cupsomatic-ppd
# apt-get install cupsys-bsd cupsys-driver-gimpprint
```

Ensuite, configurez le système avec n'importe quel navigateur Web :

```
$ mybrowser http://localhost:631
```

3.7 Autres conseils de configuration de l'hôte

3.7.1 Installer quelques paquets supplémentaires après l'installation

Une fois rendu à ce point, vous avez un système Debian petit mais fonctionnel. Il est temps d'installer des paquets plus gros.

- Lancez `tasksel`. Voir 'Installation de *tâches* avec `tasksel`' page 60.
  Vous pouvez choisir ces options si vous en avez besoin :
  - Utilisateur — X window system
  - Développement — C et C++
  - Développement — Python
  - Développement — Tcl/Tk
  - Divers — environnement TeX/LaTeX
  - Pour les autres — je préfère utiliser `tasksel` comme guide en regardant les composants listés dans « Task Info » et en les installant manuellement avec `dselect`.
- Lancez `dselect`.

  Là, la première chose à faire est de sélectionner votre éditeur de texte favori, et tout programme dont vous avez besoin. Vous pouvez installer plusieurs variantes d'Emacs en même temps. Voir '`dselect` – configuration générale' page 68 et 'Editeurs populaires' page 129.

  De plus, vous pouvez remplacer des paquets par défaut avec d'autres ayant plus d'options.
  - lynx-ssh (à la place de lynx)
  - ...
- ...

J'édite habituellement `/etc/inittab` pour éteindre plus facilement ma machine.

```
...
# Que faire lorsque CTRL-ALT-DEL est pressé.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -h now
...
```

3.7.2 Modules

Les modules pour les pilotes de périphériques sont configurés lors de l'installation initiale. `modconf` permet de configurer les modules ensuite au travers d'une interface utilisant des menus. Ce pro-

gramme est utile lorsque des modules ont été oubliés lors de l'installation ou lorsqu'un nouveau noyau est installé.

Le nom des modules à précharger est listé dans `/etc/modules`. J'utilise `lsmod` et `depmod` pour les contrôler manuellement.

De plus, assurez-vous d' ajouter quelques lignes dans `/etc/modules` pour mettre en route ip-masquerading (ftp etc.) pour les noyaux 2.4. Voir 'Le noyau 2.4 modulaire' page 77, et notamment 'Fonctions réseau' page 78.

3.7.3 Configuration de base de CD-RW

Éditer les fichiers suivants :

```
/etc/lilo.conf  (ajouter append="hdc=ide-scsi",
                 lancer lilo pour activer)
/dev/cdrom      (lien relatif # cd /dev; ln -sf scd0 cdrom)
/etc/modules    (ajouter "ide-scsi" et "sg".  Si besoin est "sr"
                 après cela.)
```

Voir 'Graveur de CD' page 104 pour les détails.

3.7.4 Grande capacité mémoire et arrêt automatique

Éditer `/etc/lilo.conf` comme suit pour configurer les paramètres de démarrage pour une grande capacité mémoire (pour les noyaux 2.2) et l'arrêt automatique (pour apm) :

```
append="mem=128M apm=on apm=power-off"
```

Lancer `lilo` pour installer cette configuration. `apm=power-off` est requis pour un noyau multi-processeur (SMP). La même chose peut être faite directement à l'invite de démarrage. Voir 'Autres astuces avec l'invite de démarrage.' page 83.

Si apm est compilé comme module comme c'est le cas par défaut dans les noyaux 2.4 de Debian, lancez `# insmod apm power_off=1` après le boot ou configurez `/etc/modules` par :

```
# echo "apm power_off=1" >>/etc/modules
```

Autrement, compiler le support ACPI permet d'atteindre le même but avec les nouveaux noyaux et semble plus compatible avec SMP (requiert une carte mère récente). Le noyau 2.4 avec une carte mère récente devrait correctement détecter les grandes capacité mémoire.

```
CONFIG_PM=y
CONFIG_ACPI=y
...
CONFIG_ACPI_BUSMGR=m
CONFIG_ACPI_SYS=m
```

et ajouter les lignes suivantes dans `/etc/modules` selon cet ordre.

```
ospm_busmgr
ospm_system
```

ou recompiler le noyau avec toutes les options ci-dessus en cochant « y ». Dans tous les cas, avec ACPI, aucun des paramètres de l'invite de démarrage n'est requis.

3.7.5 Étranges problèmes d'accès à certains sites web

Les noyaux Linux récents activent ECN par défaut, ce qui peut causer des problèmes d'accès à certains sites web situés derrière de mauvais routeurs. Pour vérifier l'état d'ECN

```
# cat /proc/sys/net/ipv4/tcp_ecn
 ... ou
# sysctl net.ipv4.tcp_ecn
```

Pour le désactiver, utilisez

```
# echo "0" > /proc/sys/net/ipv4/tcp_ecn
 ... ou
# sysctl -w net.ipv4.tcp_ecn=0
```

Pour désactiver TCP ECN à chaque démarrage, éditer `/etc/sysctl.conf` et ajouter :

```
net.ipv4.tcp_ecn = 0
```

3.7.6 Configuration d'une connexion RTC avec PPP

Installer le paquet `pppconfig` pour configurer l'accès par PPP.

```
# apt-get install pppconfig
# pppconfig
 ... suivez les indications pour configurer PPP
# adduser user_name dip
 ... autorise user_name à appeler avec PPP
```

Un accès PPP peut être initié par l'utilisateur (*user_name*) :

```
$ pon ISP_name   # démarre l'accès PPP au FAI
 ... amusez-vous bien sur l'Internet
$ poff ISP_name # arrête l'accès PPP, ISP_name est
optionnel
```

Voir `/usr/share/doc/ppp/README.Debian.gz` pour plus de détails.

D'une autre façon, le paquet `wvdial` peut être utilisé pour configurer l'accès PPP.

3.7.7 Autre configuration à vérifier dans `/etc`

Vous pouvez avoir envie d'ajouter un fichier `/etc/cron.deny`, qui manque dans l'installation standard de Debian (vous pouvez copier `/etc/at.deny`).

Chapitre 4

Didacticiels Debian

Cette section propose une orientation basique dans le monde Linux pour les débutants. Si vous utilisez Linux depuis un certain temps déjà, utilisez-la comme un inventaire.

4.1 Sources d'information

Jetez un oeil au Debian Documentation Project (DDP) (`http://www.debian.org/doc/ddp`) qui fait référence pour Debian. La plupart de ces documents sont généralement installés dans `/usr/share/doc/`. Regardez aussi dans `/usr/share/doc-base/`, qui offre des liens vers des documents sur votre système. Ajoutez `export CDPATH=. :/usr/share/doc :/usr/src/local` à votre `~/.bash_profile` pour accéder facilement aux répertoires de documentation.

Le Linux Documentation Project (LDP) (`http://www.linuxdoc.org/`) est une référence sur Linux en général. En principe le contenu du LDP est installé dans `/usr/share/doc/HOWTO/`.

La navigation a travers les documents locaux ou sur des sites ftp distants se fait en utilisant la touche `F9` dans Midnight Commander (voir 'Midnight Commander (MC)' page 50).

4.2 La console Linux

4.2.1 Se connecter

Dans un système Linux ordinaire, il y a 6 pseudo-terminals indépendants. Le passage de l'un a l'autre se fait avec la combinaison de touches `Left-Alt` et `F1` à `F6` simultanément. Chaque pseudo-terminal permet de se connecter de manière indépendante à vos différents comptes. L'environnement multi-utilisateurs est une des grandes fonctionnalités d'Unix à laquelle on devient vite accro.

Sous Unix, une bonne habitude à prendre est de se connecter en tant qu'utilisateur ordinaire pour une utilisation normale. Je dois admettre que je continue a utiliser le compte du super utilisateur (root) plus qu'il ne se doit juste à cause de son aisance et de ma négligence.

Désormais j'utilise un compte ordinaire avec les commandes `sudo`, `super` ou `su -c` pour acquérir des accès root limités.

4.2.2 Ajouter un compte utilisateur

Après l'installation du système, j'ajoute généralement un compte utilisateur ordinaire. Si le nom d'utilisateur est "pingouin",

```
# adduser pingouin
```

le créera.

J'utilise la commande `vigr` pour éditer `/etc/group` comme cela :

```
src:x:40:admin, debian, ...
staff:x:50:admin
...
```

J'utilise le groupe `staff` pour les utilisateurs qui effectuent des tâches administratives et ont le privilège exclusif de `su` (voir 'Pourquoi `GNU su` ne supporte pas le groupe `wheel`' page 101) et le groupe `src` pour CVS (voir 'CVS' page 137).

Dans le système installé par défaut, le groupe `staff` possède `/home`, donc ses membres peuvent maintenir les comptes utilisateurs. Le groupe `src` possède `/usr/src`, utilisé pour compiler un noyau, etc.

Utilisez `adduser`, `addgroup`, `vipw`, `vipw -s`, `vigr`, et `vigr -s` pour configurer proprement les utilisateurs et les groupes.

4.2.3 Comment éteindre

Comme tout autre système d'exploitation moderne où les fichiers sont mis en cache en mémoire, Linux a besoin d'être éteint proprement avant que le courant ne soit coupé sans danger. Voici la commande en mode multi-utilisateur :

```
# shutdown -h now
```

Voici la commande en mode mono-utilisateur (single-user mode) :

```
# poweroff -i -f
```

Attendez jusqu'à ce que le système vous affiche "System halted" (NdT : système arrêté), puis coupez le courant. Si apm est activé dans le BIOS et sous Linux, le système s'éteindra puis coupera l'alimentation par lui même. Voir 'Grande capacité mémoire et arrêt automatique' page 43 pour plus de détails.

4.2.4 Édition en ligne de commande

L'interpréteur de commande (shell), `bash`, est doté d'une possibilité d'édition de commandes. Utilisez simplement la flèche vers le haut pour entrer dans l'historique, ensuite utilisez les flèches comme vous l'entendriez. Autres combinaisons de touches importantes à se rappeler :

```
Ctrl-C~: Interrompre un programme
Ctrl-D~: Interrompre une entrée
Ctrl-S~: Interrompre la sortie à l'écran
Ctrl-Q~: Réactiver la sortie à l'écran
Ctrl-Alt-Del~: Redémarrer/arrêter le système (voir /etc/inittab)
Click-gauche-et-déplace-souris~: Sélectionner et copier dans le presse-papier (g
Ctrl-click-souris~: Coller le contenu du presse-papier au niveau du curseur (gpm
```

Sur une console Linux normale, seules les touches `Ctrl` gauche et `Alt` fonctionnent comme on le pense.

4.2.5 Commandes de base à retenir

Ce qui suit sont les commandes de base Unix :

```
ls, ls -al, ls -d, pwd, cd, cd ~user, cd -,
cat /etc/passwd, less, bg, fg, kill, killall,
uname -a, type commandname, sync, netstat,
ping, traceroute, top, vi, ps aux, tar, zcat,
grep, ifconfig, ...
```

Cherchez leur signification en les entrant à une invite de commande ou en tapant `man` ou `info` suivi de leur nom (NdT : si votre système est localisé correctement, vous obtiendrez de l'aide en français). Beaucoup de commandes Linux afficheront une aide brève si vous les invoquez de l'une des manières suivantes :

```
$ nomdelacommande --help
$ nomdelacommande -h
```

`whatis` *`nomdelacommande`* donne un résumé d'une ligne de n'importe quelle commande ayant une entrée dans les pages de manuel.

4.2.6 Le système X Window

Pour démarrer le système X Window depuis la console :

```
# exec startx
```

Un clic droit sur la fenêtre principale (root window) affichera un menu de sélections.

4.2.7 Commandes importantes au clavier

Quelques combinaisons importantes pour la console Linux à retenir :

```
Alt-F1 à F6~:          Passer à d'autres pseudo-terminaux
Ctrl-Alt-F1 à F6~:     Passer à d'autres pseudo-terminaux
                       (depuis un X-window, DOSEMU, etc.)
Alt-F7~:               Retourner à X-window
Ctrl-Alt-moins~:       Changer la résolution de l'écran dans X-window
Ctrl-Alt-plus~:        Changer la résolution dans l'autre sens
                       dans X-window
Ctrl-Alt-Backspace~:   Arrêter X-window
Alt-X, Alt-C, Alt-V~: Combinaisons de touches usuelles Windows/Mac pour
        Couper, Copier, Coller  avec la touche Ctrl- qui est remplacée
        par ces Alt- dans certain programmes comme Netscape Composer.
```

4.3 Midnight Commander (MC)

Midnight Commander (MC) est le "couteau suisse" GNU pour la console Linux et autres environnements de terminaux.

4.3.1 Installer MC

```
# apt-get install mc
```

Ensuite, ajoutez la fonction suivante à votre `~/.bashrc` (ou dans `/etc/bash.bashrc`, appelé depuis le `.bashrc`).

```
mc ()
{
  mkdir -p ~/.mc/tmp 2> /dev/null
  chmod 700 ~/.mc/tmp
  MC=~/.mc/tmp/mc-$$
  /usr/bin/mc -P "$@" > "$MC"
  cd "$(cat $MC)"
  rm -f "$MC"
  unset MC;
}
```

Ceci autorise MC à changer de répertoire de travail avant de sortir.

Si vous utilisez un terminal, comme `kon` et `Kterm` pour le japonais, qui utilise certains caractères graphiques, ajouter `-a` à la ligne de commande de mc peut aider à prévenir quelques problèmes.

4.3.2 Démarrer MC

```
$ mc
```

Toutes les opérations sur fichiers peuvent être effectuées depuis les menus de MC, moyennant un effort minimal de la part de l'utilisateur.

4.3.3 Gestionnaire de fichiers

Par défaut on a deux panneaux contenant la liste des fichiers de répertoires. Un autre mode utile est de configurer la fenêtre de droite sur "information" pour voir l'information sur le type de privilèges, etc. Ci-dessous, quelques raccourcis clavier essentiels. Avec le démon `gpm`, on peut aussi utiliser une souris. (Vérifiez d'avoir appuyé sur la touche Majuscule pour obtenir le comportement normal concernant le "couper/coller" dans MC.)

- `F1` : Menu d'aide
- `F3` : Afficheur de fichier interne
- `F4` : Éditeur interne

- `F9` : Active le menu déroulant
- `F10` : Quitte Midnight Commander
- `Tab` : Passe d'une fenêtre à l'autre
- `Insert` : Marque le fichier pour des opérations multiples telles que copier
- `Del` : Efface le fichier (Soyez prudent — réglez MC pour qu'il soit en mode de sauvegarde.)
- Touches du curseur : Auto-explicatif

4.3.4 Astuces en ligne de commande

- Toute commande `cd` changera le répertoire affiché à l'écran.
- `Contrôle-Entrée` ou `Alt-Entrée` copiera un nom de fichier sur la ligne de commande. Utilisez ceci avec les commandes `cp` ou `mv` associées avec l'édition en ligne de commande.
- `Alt-Tab` affichera les différentes applications à ouvrir (dans un terminal) ou ouvertes, si la commande est utilisée hors d'un terminal (NdT : et avec un clavier configuré en français).
- On peut spécifier le répertoire de départ pour les deux fenêtres comme arguments à MC ; par exemple, `mc /etc /root`.
- `Esc` + *numéro de touche* == `Fn` (càd., `Esc` + `1` = `F1`, etc. ; `Esc` + `0` = `F10`)
- `Esc` key == `Alt` key (= `Meta`, `M-`) ; càd., tapez `Esc` + `c` pour `Alt-c`

4.3.5 Éditeur

L'éditeur interne possède une manière intéressante de procéder à un copier-coller. Presser la touche `F3` marque le début de la sélection, une seconde fois `F3` marque la fin de la sélection et met celle-ci en surbrillance. Ensuite vous pouvez bouger votre curseur. Si vous appuyez sur `F6`, la zone sélectionnée sera déplacée jusqu'à l'endroit où se trouve le curseur. Si vous pressez sur `F5`, la zone sélectionnée sera copiée et insérée à l'endroit où se trouve le curseur. `F2` sauvera le fichier. `F10` vous fera quitter l'éditeur. La plupart des raccourcis clavier fonctionnent à l'intuition.

Cet éditeur peut être démarré directement avec un fichier chargé :

```
$ mc -e fichier_à_éditer
$ mcedit fichier_à_éditer
```

Il ne s'agit pas d'un éditeur multi-fenêtres mais on peut faire usage de plusieurs consoles Linux pour atteindre le même effet. Pour copier d'une fenêtre sur l'autre, utilisez les touches `Alt-Fn` pour basculer d'une console virtuelle à l'autre et utilisez "File->Insert file" ou "File->Copy to file" pour déplacer une portion d'un fichier dans un autre fichier.

Cet éditeur interne peut être remplacé par n'importe quel autre éditeur externe de votre choix.

De plus, de nombreux programmes utilisent les variables d'environnement `EDITOR` ou `VISUAL` pour décider quel éditeur utiliser. Si vous n'êtes pas à l'aise avec `vim`, configurez ces variables sur `mcedit` en ajoutant ces lignes dans le fichier `~/.bashrc` :

```
...
export EDITOR=mcedit
export VISUAL=mcedit
...
```

Je recommande vraiment de positionner ces variables à `vim`, si c'est possible. Utiliser fréquemment les commandes de vi(m) est une bonne chose à faire, puisqu'elles sont toujours présentes dans le monde Linux/Unix.

4.3.6 Visionneur

MC possède un visionneur intelligent. C'est un très bon outil pour chercher des mots dans des documents. Je l'utilise toujours pour lire les fichiers qui sont dans le répertoire `/usr/share/doc`. C'est la manière la plus rapide de surfer dans la masse d'informations sous Linux. Ce visionneur peut être chargé directement ainsi :

```
$ mc -v fichier _à_voir
```

(Notez que certains paquets violent la Charte Debian et continuent à stocker leurs documents dans `/usr/doc`.)

4.3.7 Possibilités de démarrage automatique

Faites `Entrée` sur un fichier, et le programme approprié vous montrera le contenu du fichier. Il s'agit là d'une possibilité très utile de MC.

```
exécutable~:            Exécute la commande
fichiers man, html~:    Envoie le contenu à un visionneur
fichiers tar, gz, rpm~: Affiche le contenu comme sous-répertoire
```

Afin de permettre à ces possibilités de visualisation de fonctionner, les fichiers visionnables ne doivent pas être exécutables. Changez leur statut en utilisant la commande `chmod` ou via le menu de MC.

4.3.8 Système de fichiers FTP virtuel

MC peut être utilisé pour accéder à des fichiers sur l'Internet via FTP. Accédez au menu avec la touche `F9`, puis tapez `p` pour activer le système de fichiers FTP. Entrez une URL sous la forme `username :passwd@hostname.domainname`, ce qui permettra d'être dans un répertoire distant comme s'il était local.

4.4 Étude approfondie

Il existe beaucoup de bonnes références pour débutants sur UNIX ailleurs que sur ce site. Les livres de l'éditeur O'Reilly sont très souvent de bons guides dans n'importe quel domaine de l'informatique. Le document Tips-HOWTO (`http://www.tldp.org/HOWTO/Tips-HOWTO.html`) du LDP est une autre source d'information à consulter. Voir 'Support Debian' page 157 pour plus de ressources.

Chapitre 5

Mise à jour d'une distribution

Les notes de version officielles pour mettre à jour sont situées à http://www.debian.org/releases/stable/releasenotes et http://www.debian.org/releases/testing/releasenotes (en développement).

5.1 Préparation de la transition (potato vers woody)

Une mise à jour via le réseau vers "testing" peut être effectuée comme suit (lancer le script suivant go-woody (examples/) pour le faire en une seule commande) :

```
# cd /etc/apt
# cp -f sources.list sources.old
# :>sources.list
# cd /
# apt-setup noprobe
 ... sélectionner http ou ftp
# cd /etc/apt
# grep -e "^deb " sources.list             >sources.deb
# grep -e "^deb-" sources.list             >sources.src
# sed -e "s/^d/#d/" /usr/share/doc/apt/examples/sources.list >sources.list
# sed -e "s/stable/testing/"  sources.deb >>sources.list
# apt-get update
# apt-get install apt apt-utils
# cat >preferences <<EOF
> Package: *
> Pin: release a=testing
> Pin-Priority: 700
```

```
>
> Package: *
> Pin: release a=unstable
> Pin-Priority: 70
>
> EOF
# sed -e "s/stable/unstable/" sources.deb >>sources.list
# sed -e "s/stable/unstable/" sources.src | \
> sed -e "s/^deb-/#deb-/"                 >>sources.list
```

Conseils pour `/etc/apt/preferences` (voir `apt_preferences(5)`):

```
passer à la version stable : changer la Pin-Priority de testing sur  80
passer à la version testing : garde en l'état (installe les paquets
  instables avec apt-get en ajoutant "/unstable" après le nom du paquet)
passer à testing(unstable) : changer la Pin-Priority de unstable sur 600
passer à unstable(testing) : changer la Pin-Priority de unstable sur 800
```

Un conseil concernant le choix de Pin-Priority : aller de haut en bas de la table précédente, commencer en haut immédiatement après une version de distribution et descendre dans la table au fur et à mesure que le temps approche pour le gel de la version suivante de la distribution.

Faites en sorte de configurer `apt` pour l'usage d'un proxy, si nécessaire, en réglant la variable d'environnement `http_proxy` ou en réglant la valeur de http dans `/etc/apt/apt.conf`.

La procédure décrite dans cette section ne fait que mettre à jour `apt` et un ensemble minimum de paquets afin d'éviter des problèmes de dépendance.

Comme cette méthode de mise à jour utilise `apt-get`, la gestion des paquets *recommends* et *suggests* est limitée. Lisez 'Dépendances des paquets' page 16 et utilisez `dselect` à la place pour avoir un contrôle plus fin.

5.2 Mise à jour vers Woody

Après la préparation ci-dessus :

```
# apt-get update        # commande essentielle AVANT de mettre à jour
 ... pour mettre à jour l'ensemble du système avec des "suggestions"
# apt-get -u dist-upgrade
 ... pour mettre à jour et rester avec les réglages de dselect actuels
    (new, better)
# apt-get -u dselect-upgrade # utiliser le résultat de configuration de
                             # dselect
```

5.3 Configuration de Woody

(Ceci peut ne pas être à utiliser du fait de l'évolution constante de Woody.)

Pour un système Woody fraîchement installé, éditer `/etc/apt/sources.list`,`/etc/apt/apt.conf`, et `/etc/apt/preferences` pour réaliser la même structure que celle décrite dans les sections ci-dessus.

5.4 Améliorer le fichier `sources.list`

Améliorer le fichier `sources.list` en testant chaque site sur son temps de latence et sa bande passante.

```
# apt-get install apt-spy
# cd /etc/apt ; mv sources.list sources.list.org
# apt-spy -d testing -l sources.apt
```

`netselect-apt` est similaire à `apt-spy`. Il crée un fichier `sources.list` plus complet, mais utilise une méthode moins bonne pour choisir le meilleur mirroir (comparaison des temps de ping). `apt-setup` est la méthode manuelle pour sélectionner les mirroirs de `sources.list`, mais c'est toujours le meilleur moyen de choisir des mirroirs tant que `apt-spy` n'a pas été amélioré.

Chapitre 6

Gestion des paquets Debian

Assurez-vous de configurer un proxy HTTP local en utilisant `squid` pour les paquets téléchargés par APT. Cela permet d'améliorer considérablement les performances de la mise à jour réseau, spécialement avec plusieurs machines Debian sur le LAN. Ce document traite des systèmes Woody mais la plupart des informations s'appliquent aussi à un système Potato (à part `apt_preferences(5)` et ce qui a rapport à `/etc/preferences)`).

6.1 Introduction

Si lire toute la documentation pour les développeurs est trop pour vous, lisez d'abord ce chapitre et commencez à goûter à la puissance de Debian avec `testing/unstable` :-)

6.1.1 Outils principaux

```
dselect   -- outil de gestion des paquets à l'aide de menus (haut niveau)
dpkg      -- installation de paquets (focalisé sur les fichiers de
                 paquets)
apt-get   -- installation de paquets (focalisé sur les archives de
                 paquets, interface en ligne de commande pour APT)
tasksel   -- installation de tâches (ensemble de paquets)
aptitude  -- installation de paquets (paquets et tâches, interface
                 ncurses pour APT)
deity     -- autre interface ncurses pour APT
synaptic, gsynaptic -- autres interfaces graphiques pour APT
```

Ces outils ne sont pas de même niveau. `dselect` tourne au-dessus d'APT (en ligne de commande : `apt-get`) et `dpkg`.

APT utilise `/var/lib/apt/lists/*` pour suivre l'état des paquets tandis que `dpkg` utilise `/var/lib/dpkg/status`. Si vous avez installé des paquets directement en utilisant `apt-get` ou un programme similaire tel que `aptitude`, assurez-vous de mettre à jour le fichier `/var/lib/dpkg/status` par le menu `[U]pdate` de `dselect` ou avec la commande shell `dselect update` avant d'exécuter `dselect select`, `tasksel` ou `dpkg -l`.

Quant aux dépendances de paquets, `apt-get` récupère automatiquement les paquets marqués **depends** mais ne s'occupe pas des paquets marqués **recommends** et **suggests**, alors que `dselect` offre un contrôle avancé sur le choix de ces paquets, et récupère les paquets marqués **depends** et **recommends** par défaut. Voir 'Dépendances des paquets' page 16.

6.1.2 Outils pratiques

```
apt-cache         - recherche un paquet dans le cache local
dpkg-reconfigure  - reconfigure un paquet déjà installé
dpkg-source       - gère les paquets sources
dpkg-buildpackage - automatise la création d'un paquet
...
```

6.2 Commandes de survie Debian

Avec ces connaissances, on peut vivre une vie d'**éternelle** « mise à jour » :-)

Référez-vous aussi à 'Installation du système Debian' page 29, 'Mise à jour d'une distribution' page 55 et 'Editeurs de sauvetage' page 129.

6.2.1 Installation de *tâches* avec `tasksel`

`tasksel` est l'**Installateur de Tâches Debian**, qui est offert comme option `simple` pendant l'installation du système.

Si on souhaite installer une fonction courante qui requiert plusieurs paquets, c'est le meilleur moyen. Assurez-vous d'exécuter les commandes dans l'ordre suivant :

```
# dselect update
# tasksel
```

6.2.2 Système d'installation avec APT

Vous pouvez installer des paquets provenant de différentes archives en utilisant une version récente de `apt-get` (>woody). Ceci permet, par exemple, une mise à jour de certains paquets vers

`unstable` et une remise à un niveau inférieur de certains paquets vers `stable` tout en fonctionnant avec `testing`.

Pour une mise à jour de certains paquets, ajoutez les sources pour `unstable` (`testing` si vous utilisez `stable`) à votre fichier `/etc/apt/sources.list` et éditez `/etc/apt/preferences` comme suit :

```
Package: *
Pin: release a=unstable
Pin-Priority: 50
```

(remplacez `unstable` par `testing` si vous utilisez `stable`).

Maintenant, vous pouvez exécuter `apt-get install package/unstable` et installer un paquet de `unstable`, avec toutes ses dépendances. Mais un `apt-get upgrade` ou `apt-get install package` normal n'installe pas un paquet de `unstable` (ou `testing`).

```
# apt-cache policy libc6 libc6-dev locales # vérifie l'état des paquets
# apt-get install libc6=2.2.4-1 libc6-dev=2.2.4-1 locales=2.2.4-1
# apt-get install libc6/unstable libc6-dev/unstable locales/unstable
# apt-get install -t unstable libc6 libc6-dev locales
# apt-get -u install interesting-new-package remove-package-
# apt-get remove useless-old-package
# apt-get remove --purge really-useless-old-package
```

Pour faire une mise à niveau inférieur vers `stable`, éditez `/etc/apt/preferences` comme suit :

```
Package: *
Pin: release a=stable
Pin-Priority: 1001
```

et lancez `apt-get upgrade`, ce qui force la mise à niveau inférieur grâce à Pin-priority > 1000.

6.2.3 Système de mise à jour avec APT

Système de mise à jour avec APT :

```
# apt-get update
... puis lancez l'une des commandes suivantes~:
```

```
# apt-get -u upgrade          # récupère tous les paquets
                                «~depends~»
# apt-get -u dist-upgrade     # récupère tous les paquets
                                «~depends~» et résoud les
 dépendances
# apt-get -u dselect-upgrade  # suit les sélections de dselect
```

Les commandes suivantes positionnent l'option `-u` en tant qu'action par défaut :

```
$ cat >> /etc/apt/apt.conf
// Always show packages to be upgraded (-u).
APT::Get::Show-Upgraded "true";
^D
```

Utilisez l'option `-s` pour simuler la mise à jour sans l'effectuer.

`dselect` offre une interface à base de menus au-dessus d'APT. `deity` et `aptitude` offriront des solutions de rechange pour `dselect`.

6.2.4 Rechercher les bogues de Debian et demander de l'aide

Si vous avez des problèmes avec un paquet, consultez ces sites en premier avant de demander de l'aide ou de remplir un rapport de bogue. (`lynx`, `links` et `w3m` fonctionnent bien) :

```
$ lynx http://bugs.debian.org/
$ lynx http://bugs.debian.org/package-name   # si vous connaissez le nom du paqu
$ lynx http://bugs.debian.org/bugnumber    # si vous connaissez le numéro du bog
```

Utilisez le moteur de recherche Google (www.google.com) en incluant "site :debian.org" comme mot-clé.

En cas de doute, lisez la documentation. Positionnez `CDPATH` comme suit :

```
export CDPATH=.:/usr/local:/usr/share/doc
```

et tapez

```
$ cd <packagename>
$ mc
```

Plus de ressources sont listées dans 'Support Debian' page 157.

6.2.5 Résolution de problèmes de mise à jour avec APT

Des problèmes de dépendances peuvent arriver lors de la mise à jour vers unstable/testing. La plupart du temps, c'est parce qu'un paquet qui doit être mis à jour possède une dépendance qui n'est pas trouvée. Ces problèmes sont résolus en utilisant

```
# apt-get dist-upgrade
```

Si cela ne marche pas, répétez les opérations suivantes jusqu'à ce que le problème se résolve tout seul :

```
# apt-get upgrade -f          # continue «~upgrade~» malgré les erreurs
... ou
# apt-get dist-upgrade -f     # continue «~dist-upgrade~» malgré les erreurs
```

De mauvais scripts de mise à jour peuvent causer des problèmes récurrents. Il est recommandé de résoudre ce genre de situations en consultant les scripts `/var/lib/dpkg/info/packagename.{post-,pre-}` du paquet en question et de lancer :

```
# dpkg --configure -a    # configure les paquets partiellement installés
```

Si un script se plaint d'un fichier de configuration manquant, regardez dans `/etc` pour le fichier de configuration correspondant. Si un fichier de configuration avec une extension `.new` (ou quelquechose similaire) existe, renommez-le (`mv`) en un fichier sans suffixe.

Des problèmes de dépendances peuvent survenir en installant des paquets de unstable/testing. Il existe des moyens de passer outre les dépendances.

```
# apt-get install -f package # passe outre les dépendances non résolues
```

Une autre méthode pour résoudre ces problèmes est d'utiliser le paquet `equivs`. Voir 'Le paquet `equivs`' page 72. Voir `/usr/share/doc/equivs/README.Debian` et 'Le paquet `equivs`' page 72.

6.2.6 Sauvetage avec `dpkg`

La récupération d'un `dselect` (APT) qui ne fonctionne plus en utilisant `dpkg` sans APT est possible sur un système complètement cassé :

```
# cd /var/cache/apt/archives
# dpkg -i libc6* libdb2* perl*
# dpkg -i apt* dpkg* debconf*
# dpkg -i *  # jusqu'à ce qu'il n'y ait plus d'erreurs
```

Si un paquet manque, récupérez-le avec :

```
# mc            # utilisez «~FTP link~» vers un serveur FTP Debian
```

Les paquets sur le serveur HTTP/FTP peuvent ne pas être localisés sous le répertoire habituel `/dist` mais sous le nouveau répertoire `/pool`. (Voir 'Le répertoire `pool`' page 9.)

Installez ensuite avec :

```
# dpkg -i /var/cache/apt/archives/packagefile.deb
```

Pour une dépendance non résolue, résolvez-la ou utilisez :

```
# dpkg --ignore-depends=package1,... -i packagefile.deb
# dpkg --force-depends -i packagefile.deb
# dpkg --force-depends --purge package
# dpkg --force-confmiss -i packagefile.deb # Installe fichier de configuration n
```

6.2.7 Sauver un système après avoir effacé `/var`

Si tout ce qui est sous `/var` est effacé, vous pouvez retrouver le contrôle du système si vous avez des sauvegardes de `/var/lib/dpkg/status` en exécutant :

```
# cd /
# install -d /var/cache/apt/archives
# install -d /var/cache/apt/archives/partial
# install -d /var/lib/dpkg/
# cp status-old /var/lib/dpkg/status
# apt-cache gencaches
```

Cherchez le vieux fichier `/var/lib/dpkg/status` dans `/var/lib/dpkg/status-old` ou `/var/backups/dpkg.status.*`.

Garder `/var/backups/` dans une partition séparée peut être une bonne idée puisque ce répertoire contient beaucoup de données système importantes.

6.2.8 Installer un paquet sur un système qui ne démarre plus

Démarrez Linux en utilisant une disquette/CD de sauvetage Debian ou une autre partition sur un système Linux multi-boot. Montez le système qui ne démarre plus sur `/target` et utilisez le mode d'installation chroot de `dpkg`.

```
# dpkg --root /target -i packagefile.deb
```

Configurez ensuite et résolvez les problèmes.

Au fait, si c'est seulement un `lilo` cassé qui empêche le démarrage, vous pouvez démarrer avec le disque de sauvetage standard Debian. A l'invite de démarrage, en considérant que la partition racine de Linux est sur `/dev/hda12` et que vous souhaitez le niveau de fonctionnement 3, entrez :

```
boot: rescue root=/dev/hda12 3
```

Vous avez maintenant démarré un système presque complètement fonctionnel avec le noyau de la disquette. (Il peut y avoir de petits problèmes dus au manque d'options du noyau ou de modules.)

6.2.9 Que faire si `dpkg` est cassé

Cela pose de sérieux problèmes pour l'installation des fichiers `.deb`. Les commandes suivantes vous guident pour récupérer cette situation. (Sur la première ligne, vous pouvez remplacer « links » par votre navigateur favori.)

```
$ links http://http.us.debian.org/debian/pool/main/d/dpkg/
  ... téléchargez le bon dpkg_version_arch.deb
$ ar x dpkg_version_arch.deb
$ su
password: *****
# mv data.tar.gz /data.tar.gz
# cd /
# tar xzfv data.tar.gz
```

Pour une machine `i386`, `http ://packages.debian.org/dpkg` peut aussi être utilisé comme URL.

6.3 Commandes « nirvana » de Debian

A la **lumière** de ces commandes, on sera sauvé de l'éternelle lutte karmique contre l'enfer de la mise à jour afin d'atteindre le **nirvana** Debian. :-)

6.3.1 Informations sur un fichier

De l'information sur un fichier peut être trouvée avec :

```
$ dpkg {-S|--search} pattern # cherche «~pattern~» dans les paquets installés
$ zgrep -e pattern /local/copy/of/debian/woody/Contents-i386.gz
             # cherche «~pattern~» dans les fichiers de l'archive Debian
```

Sinon, utilisez les commandes spécialisées pour les paquets :

```
# apt-get install dlocate
                 # crée un conflit avec slocate (version sécurisée de locate)
$ dlocate filename         # solution de rechange rapide pour
                                        dpkg -L et dpkg -S
...
# apt-get install auto-apt # outil d'installation de paquets à la demande
# auto-apt update          # crée le fichier db pour auto-apt
$ auto-apt search pattern
 # recherche exhaustive dans l'archive de paquets, installés ou
 # non
```

6.3.2 Informations sur un paquet

Cherche et affiche les informations sur un paquet. Assurez-vous qu'apt pointe vers la bonne archive en éditant `/etc/apt/sources.list`. Si vous voulez voir les différences entre un paquet de testing/unstable et un paquet installé, utilisez `apt-cache policy`—plutôt sympa.

```
# apt-get   check         # met à jour le cache et vérifie les dépendances
$ apt-cache search  texte  # cherche un paquet à partir de "texte"
$ apt-cache policy  paquet # information sur la priorité d'un paquet
$ apt-cache show -a paquet # affiche la description d'un paquet
                                        dans toutes les distributions
$ apt-cache showpkg paquet # informations de debogage sur un paquet
# dpkg  --audit|-C         # cherche les paquets partiellement installés
$ dpkg {-s|--status} paquet ... # état et description d'un paquet installé
$ dpkg -l paquet ...       # état du paquet installé (1 ligne)
$ dpkg -L paquet ...       liste les noms de fichiers installés par le paquet
```

`apt-cache showsrc` n'est pas documenté à la sortie de Woody, mais fonctionne :)

Vous pouvez aussi trouver des informations sur les paquets (j'utilise `mc` pour naviguer) dans :

```
/var/lib/apt/lists/*
/var/lib/dpkg/{available|status}
```

6.3.3 Reconfigurer les paquets installés

Utilisez les commandes suivantes pour reconfigurer un paquet installé.

```
# dpkg-reconfigure --priority=medium package [...]
# dpkg-reconfigure --all   # reconfigure tous les paquets
# dpkg-reconfigure locales # génère de nouvelles locales
# dpkg-reconfigure --p=low xserver-xfree86 # reconfigure le serveur X
```

Faites cela pour `debconf` si vous avez besoin de changer le mode de dialogue de `debconf` de façon permanente.

Certains logiciels sont livrés avec des scripts de configuration.

```
apt-setup     - crée /etc/sources.list
install-mbr   - installe un gestionnaire de Master Boot Record
tzconfig      - configure le fuseau horaire local
gpmconfig     - configure le gestionnaire de souris gpm
sambaconfig   - configure samba dans potato (woody utilise debconf)
eximconfig    - configure Exim (MTA)
texconfig     - configure teTeX
apacheconfig  - configure Apache (httpd)
cvsconfig     - configure CVS
sndconfig     - configure le système sonore
...
update-alternatives - configure la commande par défaut~; par exemple, vim pour v
update-rc.d         - gestion des scripts de démarrage System-V
update-menus        - système de menus Debian
...
```

6.3.4 Enlever et purger des paquets

Enlever un paquet en gardant la configuration :

```
# apt-get remove package ...
# dpkg  --remove package ...
```

Enlever un paquet et sa configuration :

```
# apt-get remove --purge package ...
# dpkg    --purge        package ...
```

6.3.5 Garder de vieux paquets

Par exemple, pour garder `libc6` et `libc6-dev` lors de l'utilisation de `dselect` et `apt-get -u upgrade` *`paquet`*, on peut utiliser les commandes suivantes :

```
# echo -e "libc6 hold\nlibc6-dev hold" | dpkg --set-selections
```

`apt-get -u upgrade` *`paquet`* ne sera pas gêné par cela. Pour garder un paquet en forçant une mise à niveau inférieur automatique avec `apt-get -u upgrade` *`paquet`* ou `apt-get -u dist-upgrade`, ajoutez ce qui suit dans `/etc/apt/preferences` :

```
Package: libc6
Pin: release a=stable
Pin-Priority: 2000
```

Ici, l'entrée « `Package :` » ne peut pas utiliser d'entrées comme « `libc6*` ». Si vous voulez garder tous les paquets binaires liés au paquet source `glibc` synchronisés, il faut les lister explicitement.

La commande suivante liste les paquets à garder :

```
dpkg --get-selections "*"|grep -e "hold$"
```

6.3.6 `dselect` – configuration générale

Ajoutez une ligne « expert » dans `/etc/dpkg/dselect.cfg` pour réduire la quantité d'informations affichée.

Quand `dselect` est lancé, il sélectionne automatiquement tous les paquets « Required », « Important » et « Standard ». Sur le système Potato, certains programmes, comme teTeX et Emacs, appartenaient à cette list et il fallait mieux ne pas les sélectionner lors de l'installation initiale en les désélectionnant manuellement (_). Dans Woody, ils ont été déplacés dans la catégorie « Optional ».

`dselect` a une interface utilisateur déroutante. 4 commandes importantes (en majuscules !) :

```
Touche     Action
Q           Quitter. Confirme la sélection courante et quitte
            (passe outre les dépendances).
R           Rétablir. Je ne voulais pas faire ça.
D           (Damn it!) Je me fiche de ce que dselect pense. Fais comme ça~!
U           Positionne tout dans l'état sUggéré.
```

Avec `D` et `Q`, vous pouvez sélectionner des paquets en conflit à vos risques et périls. Faites attention en utilisant ces commandes. Pour des machines plus lentes, lancez `dselect` sur une machine rapide pour trouver un paquet et utilisez `apt-get install` pour l'installer. `apt-get dselect-upgrade` utilise au mieux la sélection de `dselect`.

6.3.7 Supprimer les paquets du cache

Les installations de paquets avec APT laissent des fichiers de paquets dans le cache `/var/cache/apt/archives`; ce cache doit être nettoyé.

```
# apt-get autoclean # efface les fichiers de paquets inutiles
# apt-get clean     # efface tous les fichiers de paquets du cache
```

6.3.8 Enregistrer/copier la configuration du système

Pour faire une copie locale de l'état de la sélection des paquets :

```
$ dpkg --get-selections "*" >myselections   # ou utiliser \*
```

`"*"` inclut aussi les noms des paquets à purger dans *`myselections`*.

Vous pouvez transférer ce fichier vers un autre ordinateur, et l'installer avec :

```
# apt-get update
# dpkg --set-selections <myselections
# apt-get -u dselect-upgrade
```

6.3.9 Porter un paquet vers le système `stable`

Pour des mises à jour partielles du système stable, recréer un paquet pour l'environnement utilisé à partir des sources est recommandé. Cela évite de nombreuses mises à jour de paquets à cause des dépendances. D'abord, ajouter l'entrée suivante dans `/etc/apt/sources.list` :

```
deb-src http://http.us.debian.org/debian testing \
 main contrib non-free
deb-src http://non-us.debian.org/debian-non-US testing/non-US \
 main contrib non-free
deb-src http://http.us.debian.org/debian unstable \
 main contrib non-free
deb-src http://non-us.debian.org/debian-non-US unstable/non-US \
 main contrib non-free
```

Chaque entrée de `deb-src` est ici scindée en 2 lignes à cause de contraintes d'impression, mais les entrées réelles dans `sources.list` doivent tenir en une seule ligne.

Ensuite, récupérez les sources et construisez un paquet :

```
$ apt-get source package/unstable
$ dpkg-source -x package.dsc
$ cd package-version
  ... cherche parmi les paquets requis (Build-depends dans le fichier .dsc) et
      les installe. Il faut le paquet «~fakeroot~» aussi.

$ dpkg-buildpackage -rfakeroot

  ...ou (si pas de signature)
$ dpkg-buildpackage -rfakeroot -us -uc # utilisez «~debsign~» ensuite si nécessa

  ...ou (si pas de signature)
$ fakeroot ./debian/rules binary
$ fakeroot ./debian/rules clean
$ cd ..
$ fakeroot dpkg-source -b package-version
  ...ensuite pour installer
$ su -c "dpkg -i packagefile.deb"
```

Normalement, il n'y a besoin d'installer que peu de paquets avec un suffixe « -dev » pour satisfaire les dépendances. `debsign` est dans le paquet `devscripts`. `auto-apt` peut aider à résoudre les dépendances. L'utilisation de `fakeroot` évite une utilisation non nécessaire du compte root.

Dans Woody, ces problèmes de dépendances peuvent être simplifiés. Par exemple, pour compiler le paquet source de `pine` :

```
# apt-get build-dep pine
# apt-get source -b pine
```

6.3.10 Archive locale de paquets

Afin de créer une archive locale de paquets qui soit compatible avec les systèmes APT et dselect, un fichier `Packages` doit être créé. Installer le paquet `dpkg-dev` et :

```
# cd /usr/local
# install -d pool # les paquets physiques sont placés ici
# install -d dists/unstable/main/binary-i386
# ls -1 pool | sed 's/_.*$/ extra BOGUS/' | uniq > override
# editor override # adjust BOGUS
# dpkg-scanpackages pool override /usr/local/ \
   > dists/unstable/main/binary-i386/Packages
# cat > dists/unstable/main/Release << EOF
Archive: unstable
Version: 3.0
Component: main
Origin: Local
Architecture: i386
EOF
# echo "deb file:/usr/local unstable main" \
   >> /etc/apt/sources.list
```

6.3.11 Convertir ou installer un paquet non Debian

`alien` permet de convertir des paquets binaires fournis aux formats `rpm` de Redhat, `slp` de Stampede, `tgz` de Slackware, et `pkg` de Solaris en paquet `deb` de Debian. Si vous voulez utiliser un paquet d'une autre distribution Linux, vous pouvez utiliser `alien` pour le convertir vers votre format de paquet préféré et l'installer. `alien` supporte aussi les paquets LSB.

6.3.12 Vérifier les fichiers de paquets installés

`debsums` permet la vérification des fichiers de paquets installés grâce à des sommes de contrôle MD5. Certains paquets n'ont pas de somme de contrôle MD5. Une solution temporaire pour les administrateurs système :

```
# cat >>/etc/apt/apt.conf.d/90debsums
DPkg::Post-Install-Pkgs {"xargs /usr/bin/debsums -sg";};
^D
```

par Joerg Wendland <joergland@debian.org> (non testé).

6.4 Autres particularités de Debian

6.4.1 La commande **dpkg-divert**

dpkg-divert force dpkg à ne pas installer un fichier à sa place par défaut, mais à une place **détournée**. Les **détournements** peuvent être utilisés dans les scripts de paquets Debian pour déplacer un fichier qui crée un conflit. Les administrateurs système peuvent utiliser ceci pour recouvrir les fichiers de configuration d'un paquet, ou lorsque des fichiers (non marqués **conffiles**) ont besoin d'être préservés par dpkg, lors de l'installation d'une version plus récente d'un paquet qui contient ces fichiers (voir 'Préservation de la configuration locale' page 13).

```
# dpkg-divert [--add]  filename # ajoute un détournement
# dpkg-divert --remove filename # supprime un détournement
```

Ne pas utiliser dpkg-divert si cela n'est pas absolument nécessaire.

6.4.2 Le paquet **equivs**

Si vous compilez un programme depuis les sources, il est mieux d'en faire un paquet Debian (*.deb). equivs est le dernier recourt.

```
Package: equivs
Priority: extra
Section: admin
Description: Passer outre les dépendances de paquets.
 C'est un paquet qui peut être utilisé pour créer des paquets Debian qui ne
 contiennent que des informations sur les dépendances.
```

6.4.3 Commandes de rechange

Pour utiliser vim à la place de vi, utilisez update-alternatives :

```
# update-alternatives --display vi
...
# update-alternatives --config vi
  Selection    Command
-----------------------------------------------
      1        /usr/bin/elvis-tiny
      2        /usr/bin/vim
```

```
*+    3        /usr/bin/nvi

Enter to keep the default[*], or type selection number: 2
```

Les commandes de rechange du système sont maintenues dans `/etc/alternatives` en tant que liens symboliques.

Pour avoir votre gestionnaire de fenêtres favori, utilisez `x-window-manager` à la place.

`/bin/sh` est un lien direct de `/bin/bash` ou `/bin/ash`. Il est plus sûr d'utiliser `/bin/bash` pour garder la compatibilité avec de vieux scripts bash mais `/bin/ash` permet d'être plus discipliné et d'être compatible POSIX. La mise à jour vers un noyau Linux 2.4 a tendance à positionner ce lien vers `/bin/ash`.

6.4.4 `init` System-V et niveaux de fonctionnement

Le niveau de fonctionnement par défaut peut être configuré dans `/etc/inittab`.

Contrairement à d'autres distributions, Debian laisse la gestion des niveaux de fonctionnement complètement à l'administrateur. La gestion de `init` de type System-V est sensée être effectuée par les scripts `update-rc.d`.

Démarrer `/etc/init.d/`*`name`* dans les niveaux 1, 2, 3 et le stopper dans les niveaux 4, 5 avec une priorité de 20 (normal) peut se faire avec :

```
# update-rc.d name start 20 1 2 3 . stop 20 4 5 .
```

Enlever le lien symbolique alors que le script dans `init.d` existe toujours peut se faire avec :

```
# update-rc.d -f name remove
```

Pour éditer le niveau de fonctionnement, je triche. J'édite manuellement en utilisant la commande `mv` à l'invite shell de `mc` et je copie les liens avec `Alt-Enter`. Par exemple :

```
# mv S99xdm K99xdm # désactive xdm (X display manager)
```

Je désactive même un daemon en insérant `exit 0` au début du script `init.d`. Après tout, ce sont des fichiers de configuration.

6.4.5 Services désactivés

La distribution Debian prend très au sérieux la sécurité du système et s'attend à ce que l'administrateur soit compétent. Ainsi, la facilité d'utilisation est quelquefois mise en second plan et certains services fournis par des daemons sont livrés avec le niveau de sécurité le plus élevé, avec le moins de services (voire aucun) disponibles dans leur état d'installation par défaut.

Exécutez `ps aux` ou consultez le contenu de `/etc/init.d/*` et `/etc/inetd.conf`, si vous avez des doutes (à propos d'Exim, DHCP, ...). De plus, consultez `/etc/hosts.deny` de la même façon qu'à la section 'Contrôle d'accès avec PAM et login' page 100. La commande `pidof` est aussi utile (voir `pidof(8)`).

X11 ne permet pas les connexions TCP/IP (distantes) par défaut dans les versions récentes de Debian. Voir 'Connexion TCP/IP à X' page 111. La redirection X est aussi désactivée dans SSH. Voir 'Connexion X distante : `ssh`' page 111.

Chapitre 7

Le noyau Linux et Debian

Debian a une manière de recompiler le noyau Linux et les modules associés qui lui est propre. Voir aussi 'Debian et le noyau' page 24.

7.1 Recompilation du noyau

L'utilisation de `gcc`, `binutils` et `modutils` de Debian `unstable` peut aider lors de la compilation du dernier noyau Linux.

7.1.1 Méthode Debian standard

Utilisez le nouveau paquet `kernel-package` de Woody. De plus, soyez attentifs aux rapports de bogues relatifs à `gcc`, `binutils` et `modutils`.

Sous Debian, compiler un noyau personnalisé à partir des sources nécessite de prendre certaines précautions. Utilisez la nouvelle option `--append_to_version` avec `make-kpkg` pour compiler plusieurs images du noyau.

```
# apt-get install debhelper modutils kernel-package libncurses5-dev
# apt-get install kernel-source-2.4.18   # utilisez la dernière version
# vi /etc/kernel-pkg.conf     # entrez votre nom et adresse électronique
$ cd /usr/src                 # dossier où va se faire la compilation
$ tar --bzip2 -xvf kernel-source-2.4.18.tar.bz2
$ cd kernel-source-2.4.18     # si c'est votre source du noyau
$ rm -rf */pcmcia
        # [FACULTATIF] si on veut utiliser des modules de pcmcia-cs
```

```
$ cp /boot/config-2.4.18-386 .config
        # récupérez la configuration actuelle comme defaut
$ make menuconfig             # personnalisez votre noyau
$ make-kpkg clean             # obligatoire  (voir : man make-kpkg)
$ fakeroot make-kpkg --append_to_version -486 --initrd \
        --revision=rev.01 kernel_image \
 modules_image # modules_image pour pcmcia-cs* etc.
$ cd ..
# dpkg -i kernel-image*.deb pcmcia-cs*.deb # installation
```

En réalité, `make-kpkg kernel_image` lance `make oldconfig` et `make dep`. N'utilisez pas `--initrd` si initrd n'est pas utilisé. Le script de démarrage `initrd` actuel de Debian a besoin de la rustine pour le noyau `cramfs`, si vous utilisez les sources du noyau d'une archive non Debian voir http://bugs.debian.org/149236.

Vous pouvez éviter de faire `rm -fr */pcmcia` en sélectionnant « General setup —> » dans « PCMCIA/CardBus support —> » dans `make menuconfig` et en mettant la configuration à « < > PCMCIA/CardBus support » (càd décocher la case).

Sur une machine SMP, configurez CONCURRENCY_LEVEL selon `kernel-pkg.conf(5)`.

7.1.2 Méthode classique

Récupérez les sources officielles depuis :

- linux : http://www.kernel.org/
- pcmcia-cs : http://pcmcia-cs.sourceforge.net/

ou utilisez les sources équivalentes dans Debian et faites ce qui suit :

```
# cd /usr/src
# tar xfvz linux-whatever.tar.gz
# rm -rf linux
# ln -s linux-whatever linux
# tar xfvz pcmcia-cs-whatever.tar.gz
# ln -s pcmcia-cs-whatever pcmcia
# cd linux
# rm -rf */pcmcia
        # [FACULTATIF] si on veut utiliser les modules de pcmcia-cs
# make menuconfig
... configurez ...
# make dep
# make bzImage
```

```
... editions des fichiers de configuration pour lilo / grub ...
... déplacez /usr/src/linux/arch/i386/boot/bzImage vers boot ...
... /sbin/lilo ou ce qui se fait pour grub
# make modules; make modules_install
# cd ../pcmcia
# make config
# make all
# make install
... ajoutez les noms des modules dont vous avez besoin dans /etc/modules
# shutdown -r now
... redémarrez avec le nouveau noyau ...
```

7.2 Le noyau 2.4 modulaire

Les nouveaux noyaux 2.4 Debian fournis dans kernel-image-2.4.*NN* sont très modulaires. Vous devez vous assurer que les modules effectuant les fonctions du noyau que vous recherchez sont là.

7.2.1 PCMCIA

`/etc/modules` doit contenir les lignes suivantes pour que PCMCIA fonctionne :

```
# driver ISA PnP
isa-pnp
# driver PCMCIA bas niveau
# yenta_socket # je n'en ai apparemment pas besoin
```

Le reste est pris en charge par des scripts PCMCIA (du paquet `pcmcia-cs`), `depmod` et `kmod`. J'ai besoin de `isa-pnp`, sûrement parce que mon portable possède une vieille interface ISA-PCMCIA. Des portables récents avec une interface Cardbus/PCMCIA peuvent ne pas en avoir besoin.

Miquel van Smoorenburg `<miquels@cistron.nl>` nous dit :

« J'ai tout simplement supprimé tout ce qui a rapport à pcmcia sur mon portable au travail, y compris cardmgr, et j'ai installé un noyau 2.4 avec le support cardbus, et le nouveau paquet `hotplug` de Woody.

Si vous avez une carte 32 bits, vous n'avez pas besoin du paquet pcmcia ; le noyau 2.4 comprend cardservices. Et le pilote tulip standard devrait marcher avec votre carte dlink.

—Mike »

7.2.2 SCSI

[NON TESTE] `/etc/modules` doit contenir les lignes suivantes pour que SCSI fonctionne :

```
# core SCSI
scsi_mod
# driver générique SCSI
sg
# disque SCSI
sd_mod
# Tous les autres modules nécessaires pour le matériel
...
```

`depmod` peut peut-être se charger lui-même de certains de ces modules.

7.2.3 Fonctions réseau

`/etc/modules` doit contenir les lignes suivantes pour obtenir des fonctionnalités réseau supplémentaires :

```
# net/ipv-4
ip_gre
ipip

# net/ipv-4/netfilter
# iptable (dans l'ordre)
ip_tables
ip_conntrack
ip_conntrack_ftp
iptable_nat
iptable_filter
iptable_mangle
#
ip_nat_ftp
ip_queue
#
ipt_LOG
ipt_MARK
ipt_MASQUERADE
ipt_MIRROR
```

```
ipt_REDIRECT
ipt_REJECT
ipt_TCPMSS
ipt_TOS
ipt_limit
ipt_mac
ipt_mark
ipt_multiport
ipt_owner
ipt_state
ipt_tcpmss
ipt_tos
ipt_unclean
#
#ipchains
#ipfwadm
```

Ces lignes ne sont pas optimisées. `depmod` pourrait se charger de certains des modules.

7.2.4 Système de fichiers EXT3 (> 2.4.17)

Les étapes suivantes sont nécessaires pour activer le système de fichiers journalisé Ext3, en utilisant une image du noyau Debian précompilée (> 2.4.17) :

```
# cd /etc; mv fstab fstab.old
# sed 's/ext2/ext3,ext2/g' <fstab.old >fstab
# vi /etc/fstab
... mettre le type du système de fichier racine à "auto" au lieu de "ext3,ext2"
# cd /etc/mkinitrd
# echo jbd >>modules
# echo ext3 >>modules
# echo ext2 >>modules
# cd /
# apt-get update; apt-get install kernel-image-2.4.17-686-smp
... installe le noyau le plus récent et configure le démarrage du système
    (lilo est exécuté)
# tune2fs -j -i 0 /dev/hda1
# tune2fs -j -i 0 /dev/hda2
... Pour tous les systèmes de fichier EXT2 convertis en EXT3
# shutdown -r now
```

La journalisation EXT3 du système de fichier est maintenant activée. Une entrée « type » `ext3,ext2` dans `fstab` assure un passage sûr à EXT2 si le noyau ne supporte pas EXT3 pour les partitions autres que la partition racine.

Si un noyau 2.4 est déjà installé et que vous ne souhaitez pas le réinstaller, faites la même chose jusqu'aux commandes `apt-get` et :

```
# mkinitrd -o /boot/initrd.img-2.4.17-686-smp
# /lib/modules/2.4.17-686-smp
# lilo
# tune2fs -j -i 0 /dev/hda1
# tune2fs -j -i 0 /dev/hda2
... pour tous les FS EXT2 convertis en EXT3
# shutdown -r now
```

La journalisation EXT3 du système de fichier est maintenant activée.

Si `/etc/mkinitrd/modules` n'a pas été configuré quand `mkinitrd` a été exécuté et que vous souhaitez ajouter des modules au démarrage du système :

```
... à l'invite de initrd pour accéder au shell (5 sec.), appuyer sur RETURN
# insmod jbd
# insmod ext3 # modprobe ext3 prend peut-être tout en charge
# insmod ext2
# ^D
... continue le démarrage
```

Sur l'écran de démarrage (`dmesg`), « cramfs : wrong magic » apparaît mais cette erreur est connue pour être sans conséquence. Cela sera résolu dans une prochaine version (2002/8). Voir http://bugs.debian.org/135537 et le EXT3 File System mini-HOWTO (http://www.symonds.net/~rajesh/howto/ext3/index.html) ou `/usr/share/doc/HOWTO/en-txt/mini/extra/ext3-mini-HOWTO.gz` pour plus d'information.

Sur certains systèmes, le noyau se bloque lorsque EXT3 est activé mais je n'ai eu aucun problème (avec 2.4.17).

7.2.5 Support Realtek RTL-8139 dans le noyau 2.4

Pour une raison inconnue, le module RTL-8139 ne s'appelle plus rtl8139, mais 8139too. Éditez simplement votre fichier /etc/modules pour changer l'entrée correspondante lorsque vous mettez à jour un noyau 2.2 en 2.4.

Chapitre 8

Astuces Debian

8.1 Démarrer le système

Consultez le BootPrompt-HOWTO (http://www.tldp.org/HOWTO/BootPrompt-HOWTO.html) du LDP pour des informations détaillées sur l'invite de démarrage.

8.1.1 "J'ai oublié le mot de passe de root !" (1)

On peux démarrer dans le compte administrateur (root) sans connaître le mot de passe de root si on a un accès au clavier de la console. (Cela suppose qu'il n'y a pas de mot de passe requis pour le BIOS ni pour le chargeur de démarrage comme `lilo` qui empêche de lancer le système).

Cette procédure ne nécessite ni disquette de démarrage externe ni changement dans les réglages du BIOS. Ici, "Linux" est le nom du noyau Linux par défaut qui lance le système Debian.

A l'écran d'invite de `lilo`, dès que `boot :` apparaît (il faut appuyer sur la touche MAJ sur certains systèmes), entrez :

```
boot: Linux init=/bin/sh
```

Le système lance alors le noyau et exécute `/bin/sh` au lieu du processus `init` standard. Vous avez maintenant les droits et le shell de l'administrateur root. Puisque `/` est monté en lecture seule et que beaucoup de partitions ne sont pas encore montées, vous devez faire ce qui suit pour obtenir un système à peu près fonctionnel.

```
init-2.03# mount -n -o remount,rw /
init-2.03# mount -avt nonfs,noproc,nosmbfs
```

```
init-2.03# cd /etc
init-2.03# vi passwd
init-2.03# vi shadow
```

(Si la seconde colonne de `/etc/passwd` est « x » pour tous les utilisateurs, votre système utilise les "shadow passwords" (NdT : mots de passes cachés), et il faut éditer `/etc/shadow`.) Une entrée vide dans la seconde colonne de l'un de ces fichiers désactive le mot de passe. Maintenant le système peut être lancé sans le mot de passe de root. Debian (au moins après Potato) nécessite d'entrer un mot de passe si le système est lancé au niveau d'exécution (runlevel) 1, ce que d'anciennes distributions ne faisait pas.

C'est une bonne idée d'avoir un éditeur minimum dans `/bin` au cas où `/usr` ne serait pas accessible (voir 'Editeurs de sauvetage' page 129).

Vous pouvez aussi installer le paquet `sash`. Si le système ne démarre plus, lancez :

```
boot: Linux init=/bin/sash
```

`sash` sert de substitution intéractive à `sh` même lorsque `/bin/sh` est inutilisable. Il est lié statiquement, et inclut beaucoup d'utilitaires standard (tapez « help » à l'invite pour une liste).

8.1.2 "J'ai oublié le mot de passe de root !" (2)

Démarrez depuis n'importe quel ensemble de disquettes de boot/root. Si *`/dev/hda3`* est la partition racine originale, la suite vous laissera éditer le fichier de mots de passe aussi facilement que précédemment.

```
# mkdir fixit
# mount /dev/hda3 fixit
# cd fixit/etc
# vi shadow
# vi passwd
```

L'avantage de cette méthode sur la précédente est qu'on n'est pas obligé de connaître le mot de passe de `lilo`, mais on doit pouvoir régler le BIOS pour que le système se lance depuis la disquette ou le CD.

8.1.3 Je ne peux pas lancer le système

Pas de problème, personne ne prend la peine de faire une disquette de démarrage lors de l'installation. Si `lilo` est cassé, récupérez la disquette de démarrage de l'installation de Debian (rescue).

Démarrez depuis cette disquette. A l'invite de démarrage, en supposant que la partition racine de Linux est */dev/hda12* et que vous voulez le niveau d'exécution (runlevel) 3, entrez :

```
boot: rescue root=/dev/hda12 3
```

Vous avez maintenant lancé un système presque totalement fonctionnel depuis le noyau présent sur la disquette. Il peut y avoir quelques problèmes et messages d'erreur mineurs à cause d'un manque dans les options du noyau ou d'un module absent.

Si vous avez besoin d'une disquette de démarrage personnalisée, lisez `readme.txt` sur la disquette rescue.

8.1.4 Autres astuces avec l'invite de démarrage.

Le système peut être lancé à un niveau d'exécution particulier et avec des options de configurations en utilisant l'invite de démarrage de `lilo`. Les détails sont données dans le BootPrompt-HOWTO (http://www.tldp.org/HOWTO/BootPrompt-HOWTO.html) (LDP).

Si vous voulez lancer le système au niveau d'exécution 4, entrez la ligne suivante à l'invite de `lilo`.

```
boot: Linux 4
```

Si vous voulez lancer le système en mode de fonctionnement mono-utilisateur normal, et que vous connaissez le mot de passe de root, un des exemples suivants à l'invite de `lilo` fonctionnera.

```
boot: Linux S
boot: Linux 1
boot: Linux -s
```

Si vous voulez lancer le système en utilisant moins de mémoire vive que le système n'en a en réalité (par exemple 48Mo sur un ordinateur qui a 64Mo), utilisez la commande suivante à l'invite de démarrage de `lilo` :

```
boot: Linux mem=48M
```

Assurez-vous de ne pas spécifier plus de mémoire que le système n'en a en réalité, sinon le noyau va planter. Si vous avez plus de 64Mo de mémoire vive, 128Mo par exemple, à moins de spécifier `mem=128M` à l'invite de `lilo` ou d'inclure une ligne "append" similaire dans `/etc/lilo.conf`, les vieux noyaux et/ou les cates mères avec un vieux BIOS n'utiliseront pas la mémoire au-delà des 64Mo.

8.2 Enregistrer les activités

8.2.1 Enregistrer les activités du shell

L'administration système est une série d'activités beaucoup plus élaborées dans un environnement Unix que dans un environnement de PC ordinaire. Soyez sûr de connaître les méthodes de configuration de base si vous avez besoin de réparer les problèmes d'un système. Les interfaces graphiques des outils de configuration peuvent avoir l'air agréable et commode, mais sont parfois limitées surtout dans de telles situations.

Dans ce contexte, enregistrer les activités du shell est une bonne habitude, surtout si l'on est root.

Sous Emacs : utilisez `M-x shell` pour démarrer l'enregistrement dans le buffer, et utilisez `C-x C-w` pour écrire le contenu du buffer dans un fichier.

Sous le shell : utilisez la commande `script`.

```
$ script
Script started, file is typescript
 .... faites ce que vous voulez ...
 Control-D
$ col -bx <typescript >savefile
$ vi savefile
```

La méthode suivante peut être utilisée à la place de `script`.

```
$ bash -i 2>&1 | tee typescript
```

8.2.2 Enregistrer les activités sous X

Si vous avez besoin d'enregistrer l'image d'une application sous X, y compris une fenêtre de xterm, utilisez `gimp`. Il peut capturer une fenêtre isolée ou l'écran entier. Des alternatives existent comme `xwd` (paquet `xbase-clients`), `import` (`imagemagick`), ou `scrot` (`scrot`).

8.2.3 Enregistrer les changements aux fichiers de configuration

Le paquet `Changetrack` enregistrera les changements aux fichiers de configuration dans des archives RCS. Voir `changetrack(1)`.

```
# apt-get install changetrack
# vi changetrack.conf
```

8.3 Copier et archiver un sous-répertoire entier

8.3.1 Commandes de base pour copier un sous-répertoire entier

En cas de réarrangement de la structure de fichiers, déplacez le contenu des dossiers y compris les liens par les commandes :

```
Méthode classique~:
# cp -a /source/directory /dest/directory # version GNU de cp
# (cd /source/directory && tar cf - . ) | \
        (cd /dest/directory && tar xvfp - )
S'il y a des liens durs, la méthode suivante est nécessaire~:
# cd /path/to/old/directory
# find . -depth -print0 | afio -p -xv -0a /mount/point/of/new/directory
Si le système de fichier est distant~:
# (cd /source/directory && tar cf - . ) | \
        ssh user@host.dom (cd /dest/directory && tar xvfp - )
S'il n'y a pas de fichiers liés~:
# scp -pr user1@host1.dom:/source/directory \
          user2@host2.dom:/dest/directory
```

Ici, `scp` <==> `rcp` et `ssh` <==> `rsh`.

La méthode de copie d'un sous-répertoire entier est basée sur l'information fournie par Manoj Srivastava <srivasta@debian.org> sur la liste de diffusion debian-user@lists.debian.org.

8.3.2 `cp`

Traditionnellement, `cp` n'était pas réellement utilisable puisqu'il ne déréférence pas les liens symboliques, ni ne préserve les liens durs. Une autre chose à considérer était les fichiers à faible densité (avec des trous).

La version GNU de `cp` a dépassé ces limitations ; cependant, sur un système non GNU, `cp` peut toujours avoir ces problèmes. De plus, on ne peut pas générer de petites archives portables avec `cp`.

```
% cp -a . newdir
```

8.3.3 `tar`

`tar` a éliminé certains des problèmes que `cp` avait avec les liens symboliques. Cependant, bien que `cpio` gère les fichiers spéciaux, le `tar` traditionnel ne les gère pas.

`tar` gère de multiples liens durs en plaçant une copie du lien sur la bande, mais le nom de cette copie est le *seul* que vous pouvez utiliser pour récupérer le fichier ; `cpio` met une copie pour chaque lien, et vous pouvez le récupérer en utilisant n'importe lequel des noms.

La commande `tar` a changé l'option pour les fichiers `.bz2` entre Potato et Woody, donc utilisez `--bzip2` dans des scripts au lieu de `-I` (Potato) ou `-j` (Woody).

8.3.4 pax

C'est le nouvel utilitaire d'archivage portable, compatible POSIX (IEEE Std 1003.2-1992, pages 380-388 (section 4.48) et pages 936-940 (section E.4.48)), "qui chante et danse". `pax` lit, écrit, et liste les fichiers d'une archive, et copie des hiérarchies de répertoires. `pax` travaille indépendament du format de l'archive et supporte un grand nombre de formats d'archives.

Les implémentations de `pax` sont récentes et encore chaudes.

```
# apt-get install pax
$ pax -rw -p e . newdir
 ou
$ find . -depth  | pax -rw -p e  newdir
```

8.3.5 cpio

`cpio` stocke ou extrait les fichiers dans ou à partir d'une archive `cpio` ou `tar`. L'archive peut être un autre fichier sur le disque, une bande magnétique ou un tube.

```
$ find . -depth -print0 | cpio --null --sparse -pvd new-dir
```

8.3.6 afio

`afio` permet de mieux gérer les archives au format `cpio`. Il est généralement plus rapide que `cpio`, fournit plus d'options pour les bandes magnétiques, gère mieux les erreurs de données en entrée et supporte des archives multi-volume en utilisation intéractive. `afio` peut créer des archives compressées qui sont plus sûres que les archives compressées de `tar` ou `cpio`. `afio` est utilisé au mieux en tant que machine à archiver dans un script de sauvegarde.

```
$ find . -depth -print0 | afio -px -0a new-dir
```

Toutes mes sauvegardes sur bande sont faites avec `afio`.

8.4 Récupération d'un système bloqué

8.4.1 Tuer un processus

Exécutez `top` pour voir quel processus agit bizarrement. Appuyez sur « P » pour trier par temps CPU utilisé, « M » pour trier par mémoire utilisée, et « k » pour tuer un processus.

Utilisez `kill` pour tuer (ou envoyer un signal à) un processus à partir de son numéro de processus, ou `killall` pour faire la même chose à partir du nom de la commande du processus. Signaux fréquemment utilisés :

```
 1: HUP,  redémarre un daemon (NdT : serveur)
15: TERM, termine un process normalement
 9: KILL, tue un process
```

8.4.2 ALT-SysRQ

L'option de compilation du noyau "Magic SysRq key" donne une assurance contre un mauvais fonctionnement du système. Appuyer sur `ALT-SysRQ` sur un i386, suivi par une des touches `r 0 k e i s u b`, agit magiquement.

Un'R'aw récupère le clavier en cas de plantage de logiciels comme X. Mettre le niveau de log de la console à '0' réduit le nombre de messages d'erreur. sa'K' (touche d'attention système, NdT : system attention key en Anglais) tue tous les processus de la console vituelle courante. t'E'rminate tue tous les processus du terminal courant excepté init. k'I'll tue tous les processus excepté init.

'S'ync, 'U'mount, et re'B'oot permettent de se sortir de très mauvaises situations.

Les noyaux de l'installation par défaut de Debian ne sont pas compilés avec cette option au moment où ce document est écrit. Recompilez le noyau pour activer cette fonction. Des informations détaillées se trouvent dans `/usr/share/doc/kernel-doc-version/Documentation/sysrq.txt.gz` ou `/usr/src/`*`kernel-version`*`/Documentation/sysrq.txt.gz`.

8.5 Petites commandes utiles à se rappeler

8.5.1 Pager

`less` est le *pager* (visualisateur de contenu des fichiers). Appuyez sur 'h' pour de l'aide. Il peut faire beaucoup plus que `more`. `less` peut être amélioré en exécutant `eval $(lesspipe)` ou `eval $(lessfile)` dans les scripts de démarrage du shell. Plus d'information dans `/usr/share`

/doc/lessf/LESSOPEN. L'option -R permet l'affichage de caractères bruts et active les séquences d'échappement couleur ANSI. Voir less(1).

w3m peut être une bonne solution de rechange pour les systèmes de code (EUC).

8.5.2 Mémoire libre

free et top donnent de bonnes informations sur les ressources mémoire. Ne vous inquiétez pas de la taille de "used" sur la ligne "Mem :", lisez plutôt la ligne en-dessous (38792 dans l'exemple ci-dessous).

```
$ free -k # for 256MB machine
             total       used       free     shared    buffers cached
Mem:        257136     230456      26680      45736     116136 75528
-/+ buffers/cache:      38792     218344
Swap:       264996          0     264996
```

La quantité exacte de mémoire physique peut être confirmée par grep '^Memory' /var/log/dmesg, qui dans notre cas d'exemple donne « Memory : 256984k/262144k available (1652k kernel code, 412k reserved, 2944k data, 152k init) ».

```
Total         = 262144k = 256M (1k=1024, 1M=1024k)
Free to dmesg = 256984k = Total - kernel - reserved - data - init
Free to shell = 257136k = Total - kernel - reserved - data
```

Environ 5Mo ne sont pas utilisables par le système parce que le noyau les utilise.

8.5.3 Régler l'heure (BIOS)

```
# date MMJJhhmmAAAA
# hwclock --utc
# hwclock --systohc
# hwclock --show
```

Cela va régler l'heure du système et du matériel à JJ/MM hh :mm, AAAA. L'heure est affichée selon le fuseau horaire local, mais le matériel utilise UTC.

8.5.4 Régler l'heure (NTP)

Référence : Managing Accurate Date and Time HOWTO (http://www.tldp.org/HOWTO/TimePrecision-H index.html).

Régler l'heure avec une connexion Internet permanente

Régler l'horloge système automatiquement via un serveur distant :

```
# ntpdate server
```

Il est bon d'avoir cette commande dans `/etc/cron.daily` si votre système possède une connexion permanente à Internet.

Régler l'heure avec une connexion Internet intermitente

Utilisez le paquet `chrony`.

8.5.5 Comment désactiver l'écran de veille

En mode console :

```
# setterm -powersave off
```

Démarrez la console kon2 (kanji) avec :

```
# kon -SaveTime 0
```

Sous X :

```
# xset s off
 ou
# xset -dpms
 ou
# xscreensaver-command -prefs
```

Lisez les pages de manuel correspondantes.

8.5.6 Chercher dans la base de données administrative

La Glibc offre `getent(1)` pour chercher dans les entrées des bases de données administratives, càd., passwd, group, hosts, services, protocols, ou networks.

```
getent database [key ...]
```

8.5.7 Désactiver le son (beep)

On peut toujours débrancher le haut-parleur PC ;-) Pour le shell :

```
echo "set bell-style none">> ~/.inputrc
```

8.5.8 Messages d'erreur sur l'écran de la console

Afin de ne pas avoir de messages d'erreur sur l'écran, le premier endroit à regarder est `/etc/init.d/klogd`. Réglez `KLOGD="-c 3"` dans ce script et lancez `/etc/init.d/klogd restart`. Une autre méthode est de lancer `dmesg -n3`.

Voici la signification des niveaux d'erreur :
- 0 : KERN_EMERG, system is unusable
- 1 : KERN_ALERT, action must be taken immediately
- 2 : KERN_CRIT, critical conditions
- 3 : KERN_ERR, error conditions
- 4 : KERN_WARNING, warning conditions
- 5 : KERN_NOTICE, normal but significant condition
- 6 : KERN_INFO, informational
- 7 : KERN_DEBUG, debug-level messages

Si un message d'erreur particuler vous embête, pensez à faire une rustine facile pour le noyau comme `shutup-abit-bp6` (disponible dans le sous-répertoire des exemples (`examples/`)).

Un autre endroit à regarder peut être le fichier `/etc/syslog.conf`; vérifiez si les messages d'erreurs ne sont pas envoyés vers une console.

8.5.9 Régler la console

Les consoles sur les systèmes Unix sont habituellement accédées par des fonctions de la bibliothèque (n)curses. Cela donne à l'utilisateur une méthode indépendante du terminal pour mettre à jour les caractères sur l'écran avec une optimisation raisonnable. Voir `ncurses(3X)` et `terminfo(5)`.

Sur un système Debian, il existe beaucoup d'entrées prédéfinies :

```
$ toe | less                    # toutes les entrées
$ toe /etc/terminfo/ | less     # entrées reconfigurables
```

Exportez votre sélection dans une variable d'environnement `TERM`.

Si l'entrée terminfo pour xterm ne marche pas avec un xterm non Debian, changez le type de terminal de « xterm » à l'une des versions limitée en options comme « xterm-r6 » lorsque vous

vous connectez à un système Debian à distance. Voir `/usr/share/doc/libncurses5/FAQ` pour plus d'information. « dumb » est le plus petit dénominateur commun pour terminfo.

8.5.10 Remettre la console dans un état sain

Lorsque l'écran devient illisible après `$ cat` *`fichierbinaire`* (il se peut que vous ne voyiez pas la commande en tapant) :

```
$ reset
```

8.5.11 Convertir des fichiers texte DOS vers Unix

Pour convertir un fichier texte DOS (fin de ligne `^M^J`) en fichier texte Unix (`^J`) :

```
# apt-get install sysutils
$ dos2unix dosfile
```

8.5.12 Modifier des fichiers grâce aux expressions rationnelles

Pour remplacer toutes les occurences de *FROM_REGEX* par *TO_REGEX* dans tous les fichiers *FILES* . . . :

```
# perl -i -p -e 's/FROM_REGEX/TO_REGEX/g;' FILES ...
```

`-i` est pour « édition sur place », `-p` est pour « boucle implicite sur *FILES* . . . ». Si la substitution est complexe, vous pouvez faciliter le retour sur des erreurs en utilisant le paramètre `-i.bak` au lieu de `-i` ; cela permet de garder les fichiers originaux en leur ajoutant le suffixe `.bak`.

8.5.13 Convertir un gros fichier en plusieurs petits

```
$ split -b 650m file    # sépare file en morceaux de 650Mo
$ cat x* >largefile     # assemble les fichiers en 1 gros fichier
```

8.5.14 Bouts de scripts pour les tubes

Les scripts suivants font des choses sympa avec les tubes.

```
xargs -n 1 command
        # exécute command avec tous les arguments de stdin
xargs -n 1 echo|
        # place chaque argument séparé par des espaces sur une ligne
grep -e pattern|     # extrait les lignes contenant pattern
cut -d: -f3 -|       # extrait le 3e champ séparé par :
                     # (fichier passwd, etc.)
col -bx |            # supprime backspace et transforme les tabulations en espac
expand -|            # transforme les tabulations
sort -u|             # trie et supprime les doublons
tr '\n' ' '|         # concatène les lignes en une seule
tr '\r' ''|          # supprime CR
tr 'A-Z' 'a-z'|      # convertit les majuscules en minuscules
sed 's/^/# /'|       # change chaque ligne en commentaire
sed 's/\.ext//g'|    # supprime .ext
sed  -n -e 2p|       # affiche la 2e ligne
head -n 2 -|         # affiche les 2 premières lignes
tail -n 2 -|         # affiche les 2 dernières lignes
```

8.5.15 Récupérer du texte ou une archive de liste de diffusion à partir d'une page Web

Les commandes suivantes récupèrent une page web dans un fichier texte. Très utile pour copier des configurations depuis le Web.

```
$ lynx -dump http://www.remote-site.com/help-info.html >textf
ile
```

`links` et `w3m` peuvent aussi être utilisés, avec de légères différences dans le rendu.

S'il s'agit d'une archive de liste de diffusion, utilisez `munpack` pour obtenir les contenus MIME à partir du texte.

8.5.16 Imprimer joliement une page Web

Les commandes suivantes vont lire une page Web sur une imprimante/un fichier PostScript.

```
$ apt-get install html2ps
$ html2ps URL | lpr
```

Voir '`lpr/lpd`' page 41. Regardez aussi `a2ps` et `mpage` pour créer des fichiers PostScript.

8.5.17 Mesurer la durée d'une commande

Affiche le temps pris par un processus.

```
# time commande >/dev/null
real    0m0.035s       # durée sur l'horloge (temps réel écoulé)
user    0m0.000s       # durée en mode utilisateur
sys     0m0.020s       # durée en mode noyau
```

8.5.18 Commande `nice`

Utilisez `nice` (du paquet GNU `shellutils`) pour régler la « gentillesse » (valeur *nice*) d'une commande lors de son exécution. `renice` (paquet `bsdutils`) ou `top` peuvent changer la valeur *nice* d'un processus. Le processus le plus lent (priorité la plus basse) aura une valeur de 19 ; les valeurs négatives sont *not-nice* (NdT : litt. pas gentilles), une valeur de -20 étant donnée au processus le plus rapide (priorité la plus haute). Seul le superutilisateur peut donner des valeurs *nice* négatives.

```
# nice  -19 top                                          # très gentil
# nice --20 cdrecord -v -eject speed=2 dev=0,0 disk.img # très rapide
```

Parfois, une valeur extrême de *nice* fait plus de mal que de bien au système. Utilisez cette commande en faisant attention.

8.5.19 Programmer des activités (`cron`, `at`)

Utilisez `cron` et `at` pour programmer des tâches sous Linux. Consultez `at(1)`, `crontab(5)`, `crontab(8)`.

Exécutez la commande `crontab -e` pour créer ou éditer un fichier crontab pour programmer des événements régulièrement. Exemple de fichier crontab :

```
# utilise /bin/sh pour exécuter les commandes, quoi que dise /etc/passwd
SHELL=/bin/sh
# envoie un courrier électronique à Paul contenant tous les résultats, quelle
# que soit la personne a qui appartienne la crontab
MAILTO=paul
# Minute Heure JourDuMois Mois JourDeLaSemaine commande
# est exécuté à 00:05 chaque jour
5  0  *  * *   $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# est exécuté à 14:15 le 1e de chaque mois -- le résultat est envoyé à Paul
# par courrier électronique
15 14 1  * *   $HOME/bin/monthly
# est exécuté à 22:00 chaque jour de la semaine(1-5), ennuie Joe. % pour une
# nouvelle ligne, dernier % pour cc.
0 22 *   * 1-5 mail -s "It's 10pm" joe%Joe,%%Where are your kids?%.%%
23 */2 1 2 *   echo "run 23 minutes after 0am, 2am, 4am ..., on Feb 1"
5  4 *   * sun echo "run at 04:05 every sunday"
# est exécuté à 03:40 le premier lundi de chaque mois
40 3 1-7 * *   [ "$(date +%a)" == "Mon" ] && command -args
```

Exécutez la commande `at` pour programmer une tâche unique :

```
$ echo 'command -args'| at 3:40 monday
```

8.5.20 Changement de console avec `screen`

Le programme `screen` permet d'exécuter plusieurs terminaux virtuels avec un shell intéractif sur chacun, sur un **unique** terminal physique ou émulateur de terminal. Même si vous utilisez les consoles virtuelles de Linux ou plusieurs fenêtres xterm, explorer `screen` vaut le coup, de par son nombre de **possibilités**, qui incluent

- un historique,
- le copier/coller,
- l'enregistrement de l'affichage,
- une entrée de digraphe, et
- la possibilité de **détacher** une session `screen` du terminal pour la réattacher plus tard.

Scénario d'accès distant

Si vous vous connectez fréquemment sur une machine Linux depuis un terminal distant ou en utilisant un terminal VT100, `screen` vous facilitera la vie avec la possibilité de **détacher** une session.

1. Supposez que vous êtes connectés via une connexion téléphonique, et que vous exécutez une session `screen` complexe avec des éditeurs et des programmes ouverts dans plusieurs fenêtres.
2. Vous avez alors besoin de quitter votre terminal, mais vous ne voulez pas perdre votre travail en raccrochant.
3. Tapez simplement `^A d` pour détacher la session, puis déconnectez-vous. (Ou encore plus rapide, tapez `^A DD` pour que `screen` détache la session et déconnecte lui-même.)
4. Quand vous vous connectez de nouveau, entrez la commande `screen -R`, et `screen` va automatiquement réattacher toutes les fenêtres que vous aviez ouvertes.

Commandes typiques de `screen`

Une fois que `screen` est démarré, toutes les entrées au clavier sont envoyées à la fenêtre excepté la combinaison de touches de commande, par défaut `^A`. Toutes les commandes `screen` sont entrées en tapant `^A` et une touche unique [ainsi que des paramètres]. Commandes utiles :

```
^A ?     affiche un écran d'aide (affiche les combinaisons de touches)
^A c     crée une nouvelle fenêtre et l'affiche
^A n     va à la fenêtre suivante
^A p     va à la fenêtre précédente
^A 0     va à la fenêtre numéro 0
^A w     affiche la liste des fenêtres
^A a     envoie Ctrl-A à la fenêtre courante
^A h     enregistre une copie de la fenêtre courante dans un fichier
^A H     commence/termine l'enregistrement de la fenêtre courante dans
         un fichier
^A ^X    verrouille le terminal (protection par mot de passe)
^A d     détache la session screen du terminal
^A DD    détache la session screen du terminal et déconnecte
```

Ceci est seulement un petit aperçu des possibilités de `screen`. S'il y a quelquechose que vous souhaiteriez que `screen` fasse pour vous, il y a des chances qu'il puisse ! Consultez `screen(1)` pour plus de détails.

Backspace et/ou Ctrl-H dans une session `screen`

Si la touche backspace et/ou Ctrl-H ne marchent plus correctement lorsque vous exécutez `screen`, éditez `/etc/screenrc`, trouvez la ligne :

```
bindkey -k kb stuff "\177"
```

et commentez-la (càd, ajoutez « # » devant).

Programme équivalent de `screen` pour X

Consultez `xmove`. Voir `xmove(1)`.

8.5.21 Bases pour tester un réseau

Installez les paquets `netkit-ping`, `traceroute`, `dnsutils`, `ipchains` (noyau 2.2) `iptables` (noyau 2.4), et `net-tools`, puis :

```
$ ping yahoo.com            # teste la connexion Internet
$ traceroute yahoo.com      # trace les paquets IP
$ ifconfig                  # affiche la configuration de la machine
$ route -n                  # affiche la configuration de routage
$ dig [@dns-server.com] host.dom [{a|mx|any}] |less
      # affiche les enregistrements DNS de host.dom
      # sur dns-server.com pour un enregistrement {mx|any}
$ ichains  -L -n |less      # affiche le filtre de paquets (noyau 2.2)
$ iptables -L -n |less      # affiche le filtre de paquets (noyau 2.4)
$ netstat -a                # recherche tous les ports ouverts
$ netstat -l --inet         # recherche tous les ports en écoute
$ netstat -ln --tcp         # recherche tous les ports tcp en écoute (affichage
```

8.5.22 Vider les courriers électroniques de la file locale

Pour vider la file locale des courriers électroniques présents :

```
# exim -q    # vide les courriers électroniques en attente
# exim -qf   # vide tous les courriers électroniques
# exim -qff  # vide même les courriers électroniques gelés
```

`-qff` est peut-être une meilleure option pour le script `/etc/ppp/ip-up.d/exim`.

8.5.23 Supprimer les courriers électroniques gelés de la file locale

Pour supprimer les courriers électroniques gelés de la file locale en renvoyant un message d'erreur :

```
# exim -Mg `mailq | grep frozen | awk '{ print $3 }'`
```

8.5.24 Supprimer le contenu d'un fichier

Pour supprimer le contenu d'un fichier, par exemple un fichier de journal, n'utilisez pas `rm` pour effacer le fichier et créer ensuite un fichier vide, parce que le fichier peut toujours être accédé entre les commandes. Ce qui suit est une façon sûre de supprimer le contenu d'un fichier.

```
$ :>file-to-be-cleared
```

8.5.25 Fichiers fantômes

Les commandes suivantes créent des fichiers fantômes ou vides.

```
$ dd if=/dev/zero    of=filename bs=1k count=5 # 5Ko sans contenu
$ dd if=/dev/urandom of=filename bs=1m count=7 # 7Mo de contenu aléatoire
$ touch filename # crée un fichier de taille 0 (si le fichier existe, met à jour
```

8.5.26 `chroot`

Supposez que vous avez installé une distribution Linux complète (cela peut être une autre version de Debian ou même une distribution Red Hat) dans `/dev/hda1` et que vous exécutez un autre système Linux installé sur `/dev/hda2`. Vous pouvez exécuter le système de `/dev/hda1` sans redémarrer le système de `/dev/hda2` en partageant le même noyau.

```
# mount /dev/hda1 /mnt/target
 ... en supposant que /dev/hda1 contient un système
# chroot /mnt/target
 ... Maintenant le contenu de /dev/hda1 est vu comme répertoire racine /
# mount proc /proc              # au cas où
 ... exécutez les commandes depuis /dev/hda1
```

Cela permet d'installer stable/testing/unstable sur une machine. De plus, on peut exécuter un programme nécessitant beaucoup de mémoire comme `dselect` en le lançant sur une machine pendant que l'on monte par NFS une machine auxiliaire en r/w et que l'on pointe `chroot` vers la machine auxiliaire.

Il est facile de créer un système `chroot` avec la commande `debootstrap` de Woody.

```
# mkdir potatochroot
# debootstrap potato potatochroot
# chroot potatochroot
# apt-setup # configure /etc/apt/sources.list
```

Il existe un paquet chroot plus spécialisé, `pbuilder`, qui construit un système chroot et compile un paquet dans le chroot. C'est un système idéal pour vérifier que les dépendances de compilation d'un paquet sont correctes, et pour être sûr que des dépendances non nécessaires ou fausses n'existent dans le paquet résultant.

8.5.27 Monter une image de disque dur

Si *`file.img`* contient l'image d'un disque dur et que le disque dur original avait une configuration *xxxx* = (octets/secteur) * (secteurs/cylindres), les commandes suivantes vont le monter sur `/mnt` :

```
# mount -o loop,offset=xxxx file.img /mnt
```

Notez que la plupart des disques durs ont 512 octets/secteur.

8.5.28 Samba

Bases pour récupérer des fichiers depuis Windows :

```
# mount -t smbfs -o username=myname,uid=my_uid,gid=my_gid \
        //server/share /mnt/smb  # monte un répertoire Windows sous Linux
# smbmount //server/share /mnt/smb \
        -o "username=myname,uid=my_uid,gid=my_gid"
# smbclient -L 192.168.1.2 # list the shares on a computer
```

Les voisins de Samba peuvent être trouvés depuis Linux :

```
# smbclient -N -L ip_address_of_your_PC | less
# nmblookup -T "*"
```

Chapitre 9

Configurer un système Debian

Ce chapitre décrit seulement les bases de la configuration du système. Lire 'Installation du système Debian' page 29 est pré-requis pour ce chapitre.

Pour ceux qui sont intéressés par la sécurité, il est hautement recommandé de lire Securing Debian Manual (`http://www.debian.org/doc/manuals/securing-debian-howto/`) qui se trouve aussi dans le paquet `harden-doc`.

9.1 Astuces sur l'initialisation du système

Voir 'Le programme `init`' page 22 pour les bases sur les scripts d'initialisation de Debian.

9.1.1 Personnaliser les scripts d'initialisation

Debian utilise le système de scripts d'initialisation sys-V. Bien que ces scripts d'initialisation dans `/etc/init.d/*` soient marqués comme des fichiers de configuration et que les administrateurs système soient libres de les modifier, la personnalisation de ces scripts d'initialisation grâce aux fichiers de `/etc/defaults/*` est l'approche préférable.

Par exemple `/etc/init.d/rcS` peut être utilisé pour personnaliser les défauts au démarrage pour `motd`, `sulogin`, etc.

9.1.2 Personnaliser le journal du système

Le journal du système peut être configuré par le fichier `/etc/syslog.conf`. Utilisez le paquet `colorize` pour colorier les fichiers du journal. Voir `syslogd(8)` et `syslog.conf(5)`.

9.1.3 Optimisation de l'accès au matériel

Quelques outils pour l'optimisation du matériel sont donnés à l'administrateur système avec Debian.

- `hdparm`
  - Optimisation du disque dur. Très efficace.
  - Dangereux. Il faut lire `hdparm(8)` d'abord.
  - `hdparm -tT` *`/dev/hda`* pour tester la vitesse du disque.
  - `hdparm -c1 -d1 -u1 -m16 -A` *`/dev/hda`* pour accélérer un système IDE moderne. (Cela peut être dangereux.)
- `setserial`
  - Collection d'outils pour gérer les ports série.
- `scsitools`
  - Collection d'outils pour gérer le matériel SCSI.
- `memtest86`
  - Collection d'outils pour gérer la mémoire.
- `hwtools`
  - Collection d'outils pour la gestion bas niveau du matériel.
    - irqtune : change la priorité d'IRQ des périphériques pour autoriser ceux qui ont besoin d'une priorité haute et d'un service rapide (par exemple ports série, modems) à l'avoir.
    - scanport : scanne l'espace I/O de 0x100 à 0x3ff pour trouver les périphériques ISA installés.
    - inb : un petit programme rapide qui lit un port I/O et retourne la valeur en hexadécimal et en binaire.
- `schedutils`
  - Utilitaires pour l'ordonnanceur Linux.
  - `taskset`, `irqset`, `lsrt` et `rt` sont inclus.
  - Avec `nice` et `renice` (non inclus), ils permettent un contrôle complet des paramètres de l'ordonnancement des processus.

9.2 Contrôle d'accès

9.2.1 Contrôle d'accès avec PAM et login

PAM (*Pluggable Authentication Modules*, Modules d'Authentification Greffables) fournit le contrôle de la connexion.

```
/etc/pam.d/*            # fichiers de contrôle de PAM
```

```
/etc/pam.d/login          # fichier de contrôle de PAM pour la connexion
/etc/security/*           # paramètres des modules de PAM
/etc/securetty            # contrôle les connexions de root sur la console
/etc/login.defs           # contrôle le comportement de login
```

Changez le contenu du fichier `/etc/pam.d/login` comme suit, si vous voulez des terminaux non sécurisés mais ne demandant pas de mot de passe. Faites le à vos propres risques.

```
#auth       required   pam_unix.so nullok
auth       required   pam_permit.so
```

Des astuces similaires peuvent être appliquées pour `xdm`, `gdm`, ..., pour avoir une console X sans mot de passe.

Le nombre maximum de processus peut être réglé avec `ulimit -u 1000` dans un shell Bash ou avec la configuration de `/etc/security/limits.conf`. D'autres paramètres comme `core` peuvent être réglés d'une façon similaire. La valeur initiale de `PATH` peut être réglée dans le fichier `/etc/login.defs` avant le script de démarrage du shell.

La documentation de PAM est dans le paquet `libpam-doc`. Le Guide de l'Administrateur d'un Système Linux-PAM couvre la configuration de PAM, les modules disponibles, etc. La documentation inclut aussi le Guide du Développeur d'Applications Linux-PAM et le Guide du Développeurs de Modules Linux-PAM.

9.2.2 Pourquoi `GNU su` ne supporte pas le groupe `wheel`

C'est la célèbre phrase de Richard M. Stallman à la fin de l'ancienne page `info su`. Ne pas s'inquiéter : le `su` actuel de Debian utilise PAM, donc on peut restreindre l'accès de `su` à n'importe quel groupe en utilisant `pam_wheel.so` dans `/etc/pam.d/su`. Ce qui suit réglera le groupe `adm` du système Debian comme équivalent du groupe BSD `wheel` et autorisera `su` sans mot de passe pour les membres du groupe.

```
# anti-RMS configuration in /etc/pam.d/su
auth       required   pam_wheel.so group=adm

# Wheel members to be able to su without a password
auth       sufficient pam_wheel.so trust group=adm
```

9.2.3 Signification des groupes

Quelques groupes intéressants :

- Le groupe `root` est le groupe `wheel` par défaut pour `su` si `pam_wheel.so` est utilisé sans l'argument `group=`.
- Le groupe `adm` peut lire les fichiers du journal.
- Le groupe `cdrom` peut être utilisé localement pour donner accès au lecteur de CDROM à un ensemble d'utilisateurs.
- Le groupe `floppy` peut être utilisé localement pour donner accès au lecteur de disquettes à un ensemble d'utilisateurs.
- Le groupe `audio` peut être utilisé localement pour donner accès à un périphérique audio à un ensemble d'utilisateurs.
- Le groupe `src` possède le code source, y compris les fichiers de `/usr/src`. Il peut être utilisé localement pour donner la possibilité de gérer le code source du système à un utilisateur.
- Le groupe `staff` est utile pour donner la possibilité de faire des choses dans `/usr/local` et créer des répertoires dans `/home` à plus d'administrateurs système (support/junior).

Pour une liste complète, voir la section « FAQ » dans Securing Debian Manual (`http://www.debian.org/doc/manuals/securing-debian-howto/`), qui se trouve aussi dans le paquet `harden-doc`.

9.2.4 sudo – un environnement de travail plus sûr

J'utilise `sudo` principalement comme protection contre ma stupidité. Utiliser `sudo` est toujours mieux que d'utiliser le système avec le compte root.

Installez `sudo` et activez-le en configurant /etc/sudoers (`examples/`). Réglez aussi les possibilités de `sudo` pour les groupes dans `/usr/share/doc/sudo/OPTIONS`.

Cette configuration fournit aux membres du groupe « staff » un accès à toutes les commandes lancées en tant que root avec `sudo` et donne aux membres du groupe « src » un accès à certaines commandes lancées en tant que root avec `sudo`.

L'avantage de `sudo` est qu'il requiert seulement les mots de passe utilisateurs pour se connecter et que l'activité est surveillée. C'est un moyen simple de donner des pouvoirs à un administrateur débutant. Par exemple :

```
$ sudo chown -R myself:mygrp .
```

Bien sûr, si vous connaissez le mot de passe de root (comme beaucoup d'utilisateurs individuels), n'importe quelle commande peut être lancée en tant que root depuis un compte utilisateur :

```
$ su -c "shutdown -h now"
Password:
```

(Je sais que je devrais limiter les privilèges du compte administrateur avec `sudo`. Mais comme c'est mon serveur personnel, je ne me suis pas encore embêté avec ça.)

Un autre programme permet à des utilisateurs ordinaires de lancer des commandes avec les privilèges de root, voir le paquet `super`.

9.2.5 Contrôle d'accès aux démons

Le *super-serveur* internet, `inetd`, est exécuté lors du démarrage par `/etc/rc2.d/S20inetd` (pour le niveau d'exécution 2) qui est un lien vers `/etc/init.d/inetd`. `inetd` permet de n'exécuter qu'un démon qui en invoque plusieurs autres, réduisant ainsi la charge du système.

Lorsqu'une requête pour un service arrive, le protocole et le service sont identifiés en regardant dans la base de données stockée dans `/etc/protocols` et `/etc/services`. Pour un service internet normal, `inetd` utilise `/etc/inetd.conf`. Pour un service basé sur Sun-RPC, `inetd` utilise `/etc/rpc.conf`.

Pour la sécurité du système, assurez-vous de désactiver les services non utilisés dans `/etc/inetd.conf`. Les services Sun-RPC doivent être actifs pour NFS et d'autres programmes basés sur RPC.

Parfois, `inetd` n'exécute pas le serveur demandé directement, mais exécute le programme d'enveloppe de démon TCP/IP `tcpd` avec le serveur demandé comme argument dans `/etc/inetd.conf`. Dans ce cas, `tcpd` lance le serveur approprié après avoir enregistré dans le journal la requête et avoir fait quelques autres vérifications en utilisant les fichiers `/etc/hosts.deny` et `/etc/hosts.allow`.

Si vous avez un problème avec l'accès à distance sur un système Debian récent, commentez la ligne "ALL : PARANOID" dans `/etc/hosts.deny` si elle existe.

Pour plus de détails, voir `inetd(8)`, `inetd.conf(5)`, `protocols(5)`, `services(5)`, `tcpd(8)`, `hosts_access(5)`, et `hosts_options(5)`.

Pour plus d'information sur Sun-RPC, voir `rpcinfo(8)`, `portmap(8)`, et `/usr/share/doc/portmap/portmapper.txt.gz`.

9.2.6 Lightweight Directory Access Protocol

Références :
- OpenLDAP (`http://www.openldap.org/`)
- OpenLDAP Admin Guide dans le paquet `openldap-guide`
- LDP : LDAP Linux HOWTO (`http://www.tldp.org/HOWTO/LDAP-HOWTO/index.html`)
- LDP : LDAP Implementation HOWTO (`http://www.tldp.org/HOWTO/LDAP-Implementation-HOW index.html`)

- OpenLDAP, extensive use reports (http://portal.aphroland.org/~aphro/ldap-docs/ldap.html)
- Open LDAP with Courier IMAP and Postfix (http://annapolislinux.org/docs/plc/postfix-courier-howto.txt)

9.3 Graveur de CD

Les graveurs de CD sur interface ATAPI/IDE sont devenus des périphériques courants. Le CD un bon média pour sauvegarder et archiver un système pour un utilisateur individuel ayant besoin d'une capacité < 640 Mo. Pour des informations plus officielles, consultez le CD-Writing-HOWTO (http://www.tldp.org/HOWTO/CD-Writing-HOWTO.html) du LDP.

9.3.1 Introduction

D'abord, sachez qu'une interruption de l'envoi des données vers le graveur entraînera des erreurs irrécupérables sur le CD. Prenez donc un graveur avec un cache aussi gros que possible. Si le prix n'a pas d'importance, optez pour une version SCSI plutôt qu'ATAPI/IDE. Si vous avez le choix de l'interface IDE sur laquelle connecter le graveur, préférez celle sur bus PCI (une de celles de la carte mère) plutôt qu'une sur bus ISA (carte SB16, etc.).

Un graveur connecté sur interface IDE doit être piloté par le pilote IDE-SCSI et non par le pilote CD IDE habituel. Le pilote SCSI générique doit aussi être activé. Pour cela, deux approches existent pour les noyaux distribués avec des distributions modernes (à la date de mars 2001).

9.3.2 Première approche : modules + `lilo`

Ajoutez la ligne suivante à `/etc/lilo.conf` si vous utilisez le noyau distribué avec Debian. Si vous utilisez plusieurs options, listez-les en les séparant par des espaces.

```
append="hdx=ide-scsi ignore=hdx"
```

Ici, le graveur, accédé par le pilote ide-scsi, est identifié par `hdx`, où *x* représente :

```
hdb      esclave sur le premier port IDE
hdc      maître sur le second port IDE
hdd      esclave sur le second port IDE
hde ... hdh  sur un port IDE externe
```

Lancez les commandes suivantes, en tant que root, pour activer les options après avoir terminé la configuration :

```
# lilo
# shutdown -h now
```

9.3.3 Seconde approche : recompiler le noyau

Debian utilise `make-kpkg` pour créer le noyau. Utilisez la nouvelle option `--append_to_version` avec make-kpkg pour créer plusieurs images du noyau. Utilisez la configuration suivante avec `make menuconfig`:

- bzImage
- Excluez le pilote CD IDE (non obligatoire mais c'est plus simple comme ça)
- Ajoutez ide-scsi et sg, ou compilez-les en tant que modules

9.3.4 Étapes post-configuration

Le support pour le graveur peut être activé dans le noyau lors du démarrage avec les commandes suivantes :

```
# echo ide-scsi >>/etc/modules
# echo sg       >>/etc/modules
# cd /dev; ln -sf scd0 cdrom
```

Une activation manuelle peut aussi être effectuée avec :

```
# modprobe ide-scsi
# modprobe sg
```

Après un redémarrage, vous pouvez vérifier l'installation avec :

```
$ dmseg|less
# apt-get install cdrecord
# cdrecord -scanbus
```

[Par Warren Dodge] Quelquefois, il peut y avoir des conflits entre `ide-scsi` et `ide-cd` s'il y a à la fois un lecteur de CDROM et un graveur CD-R/RW sur le système. Essayez d'ajouter ce qui suit dans votre fichier `/etc/modutils/aliases`, exécutez `update-modules` et redémarrez.

```
pre-install      ide-scsi      modprobe ide-cd
```

Cela force le pilote IDE à être chargé avant `ide-scsi`. Le pilote `ide-cd` prend le contrôle du lecteur de CDROM ATAPI — tout ce qu'on ne lui a pas dit d'**ignorer**. Cela laisse simplement les périphériques ignorés au contrôle de `ide-scsi`.

9.3.5 Image de CD (bootable)

Pour créer un CDROM contenant des fichiers contenus dans le répertoire `target-directory/` en tant qu'image nommée `cd-image.raw` (CD bootable, format Joliet TRANS.TBL. Pour un CD non bootable, enlevez les options `-b` et `-c`), insérez une disquette de démarrage dans le premier lecteur de disquette et faites :

```
# dd if=/dev/fd0 target-directory/boot.img
# mkisofs -r -V volume_id -b boot.img -c bootcatalog -J -T \
        -o cd-image.raw target_directory/
```

Une utilisation amusante est de créer un CDROM DOS bootable. Si *`boot.img`* contient l'image d'une disquette de démarrage DOS, le CDROM démarrera comme si une disquette DOS était dans le premier lecteur de disquette (A :). Faire cela avec freeDOS est encore plus intéressant.

L'image créée peut être vérifiée en la montant en utilisant le périphérique de boucle (loop device).

```
# mount -t iso9660 -o ro,loop cd-image.raw /cdrom
# cd /cdrom
# mc
# umount /cdrom
```

9.3.6 Graver un CD

Premier test avec les commandes suivantes (cas d'un graveur double vitesse)

```
# nice --10 cdrecord -dummy speed=2 dev=0,0 disk.img
```

Si le test est réussi, gravez un CD-R avec

```
# nice --10 cdrecord -v -eject speed=2 dev=0,0 disk.img
```

Ou gravez un CD-RW avec

```
# nice --10 cdrecord -v -eject blank=fast speed=2 dev=0,0 disk.img
```

Certains graveurs de CD marchent mieux avec

```
# nice --20 cdrecord -v blank=all speed=2 dev=0,0
```

suivi de

```
# nice --10 cdrecord -v -eject speed=2 dev=0,0 disk.img
```

Deux étapes sont nécessaires pour empêcher que des fins d'attente SCSI lors du formatage n'interfèrent avec l'étape de gravure. Les arguments passés à `nice` peuvent avoir besoin d'un ajustement.

9.3.7 Créer l'image d'un CD

Certains CD-R et CD commerciaux ont des secteurs parasites à la fin qui les rendent incopiables par `dd` (le CD de Windows 98 est l'un d'eux). Le paquet `cdrecord` comprend la commande `readcd`. Utilisez-la pour copier le contenu de n'importe quel CD dans un fichier image. S'il s'agit d'un CD de données, montez-le et exécutez `mount` pour connaître sa taille. Divisez le nombre affiché (en blocs de 1024 octets) par 2 pour connaître le nombre de secteurs du CD (2048 octets). Lancez `readcd` avec les bonnes options et utilisez l'image du CD pour graver un CD-R/RW.

```
# readcd target lun scsibusno # choisissez la
fonction 11
```

Ici, positionnez tous les paramètres de ligne de commande à 0 dans la plupart des cas. Parfois, le nombre de secteurs donné par `readcd` est trop important ! Utilisez le nombre donné par mount pour de meilleurs résultats.

```
Mon CD-R         = +2 secteurs
CD de MS Windows = +1 secteur, càd +2048 octets
```

9.3.8 Images de CD Debian

Pour obtenir les dernières informations, consultez le site Debian CD. (http://cvs.debian.org/debian-cd/)

Si vous avez une connexion Internet rapide, installez par le réseau en utilisant :

- quelques images de disquettes (http://www.debian.org/distrib/floppyinst).
- une image de CD minimale (http://www.debian.org/CD/netinst/).

Si vous n'avez pas de connexion Internet rapide, achetez des CD depuis un vendeur de CD (http://www.debian.org/CD/vendors/).

Merci de ne pas gâcher la bande passante en téléchargeant des images de CD standard à moins que vous ne soyez un testeur d'images de CD (même avec la nouvelle méthode jigdo).

Une image de CD à noter est celle de KNOPPIX - Système de Fichier Linux Fonctionnel sur CD (http://www.knopper.net/knoppix/index-en.html). Ce CD démarre avec un système Debian fonctionnel sans s'installer sur le disque dur.

9.3.9 Système de sauvegarde sur CD-R

Permet de copier les fichiers de configuration et de données importants sur CD-R. Utilisez le script suivant : backup (examples/)

9.3.10 Copier un CD audio sur un CD-R

Je n'ai pas testé cela personnellement :

```
# apt-get get install cdrecord cdparanoia
# cdparanoia -s -B
# cdrecord dev=0,0,0 speed=2 -v -dao -eject defpregap=1 -audio *.wav
```

ou,

```
# apt-get install cdrdao #disk at once
# cdrdao read-cd --device /dev/cdrom --paranoia-mode 3 my_cd # lit le CD
# cdrdao write --device /dev/cdrom --speed 8 my_cd    # grave un nouveau CD
```

`cdrdao` fait une vraie copie (pas de trous, etc.).

9.4 Le programme X

'Serveur X' page ci-contre un programme sur la machine locale qui affiche des fenêtres X et/ou un bureau sur le moniteur (CRT, LCD) d'un utilisateur et accepte des entrées au clavier et à la souris.

'Client X' page suivante un programme sur une machine (locale ou distante) qui exécute un logiciel d'application compatible avec X-Window.

Cela inverse l'utilisation habituelle de « serveur » et « client » dans d'autres contextes. Pour plus de détails, référez-vous au `X(7)`, au XWindow-User-HOWTO (`http://www.tldp.org/HOWTO/XWindow-User-HOWTO.html`), et au Remote X Apps mini-HOWTO (`http://www.tldp.org/HOWTO/mini/Remote-X-Apps.html`) du LDP.

Il y a plusieurs façons de faire accepter les connexions distantes d'un client X (côté application) à un serveur X (côté affichage) :

- `xhost`
  - le mécanisme de liste d'hôtes (non sûr).
  - protocole non chiffré (sujet à des attaques d'écoute de ligne).
  - à ne pas utiliser, si possible.
  - voir 'Connexion X distante : `xhost`' page 111 et `xhost(1x)`.
- `xauth`
  - le mécanisme des cookies magiques du MIT (non sûr mais mieux que `xhost`).
  - protocole non chiffré (sujet à des attaques d'écoute de ligne).
  - à utiliser seulement pour des connexions locales.
  - voir 'Devenir root sous X' page 112 et `xauth(1x)`.
- `xdm`, `wdm`, `gdm`, `kdm`,...
  - voir `xdm(1x)` et `Xsecurity(7)` pour les bases du contrôle d"accès à un affichage X.
  - voir `wdm(1x)`, `gdm(8)`, et `kdm.options(5)` pour plus d'information, si ces programmes sont installés.
  - voir '`init` System-V et niveaux de fonctionnement' page 73 pour savoir comment désactiver `xdm` pour avoir accès à la console Linux après le démarrage sans purger le paquet `xdm`.
- `ssh -X`
  - mécanisme de redirection de port à travers SSH (sûr).
  - protocole chiffré (gâchis de ressources en utilisation locale).
  - à utiliser pour des connexions distantes.
  - voir 'Connexion X distante : `ssh`' page 111.

Toutes les méthodes de connexion distante, sauf `ssh`, ont besoin de l'activation des connexions TCP/IP au serveur X. Voir 'Connexion TCP/IP à X' page 111.

9.4.1 Serveur X

Consultez `XFree86(1)` pour des informations sur le serveur X.

Pour (re)configurer X4 sous Woody, exécutez :

```
# dpkg-reconfigure --p=low xserver-xfree86
```

Exécution du serveur X depuis une console locale :

```
$ startx -- :<display> vtXX
par exemple~:
$ startx -- :1 vt8
... exécution sur un terminal vt8 connecté à localhost:1
```

9.4.2 Client X

La plupart des clients X peuvent être démarrés comme ceci :

```
client $ xterm -geometry 80x24+30+200 -fn 6x10 -display hostname:0 &
```

Ici, les arguments de ligne de commande optionnels signifient :

- `-geometry` *`WIDTH`*`x`*`HEIGHT`*`+`*`XOFF`*`+`*`YOFF`* : taille initiale et emplacement de la fenêtre.
- `-fn` *`FONTNAME`* : police utilisée pour l'affichage du texte. *`FONTNAME`* peut être :
  - a14 : Taille normale
  - a24 : Grande taille
  - ...(consultez les polices disponibles avec `xlsfont`.)
- `-display` *`displayname`* : le nom du serveur X à utiliser. *`displayname`* peut être :
  - *`hostname`* `:`*`D.S`* signifie écran *S* sur affichage *D* de l'hôte `hostname` ; le serveur X de cet affichage écoute le port TCP 6000+D.
  - *`host`*`/unix :`*`D.S`* signifie écran *S* sur affichage *D* de l'hôte `host` ; le serveur X de cet affichage écoute la socket UNIX `/tmp/.X11-unix/XD` (et n'est donc utilisable depuis *`host`*).
  - `:`*`D.S`* est équivalent à *`host`*`/unix :`*`D.S`*, où `host` est le nom de l'hôte local

Le *`displayname`* (nom d'affichage) par défaut pour un programme client X (côté application) peut être défini par la variable d'environnement DISPLAY. Par exemple, avant de lancer un programme client X, on peut exécuter l'une des commandes suivantes pour cela :

```
$ export DISPLAY=:0
        # Valeur par défaut, machine locale avec premier écran X
$ export DISPLAY=hostname.fulldomain.name:0.2
$ export DISPLAY=localhost:0
```

9.4.3 Connexion TCP/IP à X

Comme les connexions TCP/IP distantes sans chiffrage peuvent être sujettes à des attaques d'écoute de ligne, la configuration par défaut de X de Debian désactive les sockets TCP/IP. Pensez à utiliser `ssh` pour des connexions X distantes (see 'Connexion X distante : `ssh`' de la présente page).

La méthode décrite ici n'est pas encouragée à moins que l'on soit dans un environnement très sûr derrière un bon pare-feu et en présence d'utilisateurs de confiance uniquement. La configuration suivante sur le serveur X restaure les connexions TCP/IP :

```
# find /etc/X11 -type f -print0 | xargs -0 grep nolisten
/etc/X11/xinit/xserverrc:exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

Supprimez `-nolisten` pour permettre les connexions TCP/IP à X.

9.4.4 Connexion X distante : `xhost`

`xhost` autorise l'accès en se basant sur les noms d'hôtes, ce qui n'est pas sûr du tout. Les commandes suivantes permettent de désactiver la vérification de l'hôte et autorisent des connexions de n'importe où si les connexions TCP/IP sont autorisées ((see 'Connexion TCP/IP à X' de la présente page) :

```
$ xhost +
```

Vous pouvez réactiver la vérification de l'hôte avec :

```
$ xhost -
```

`xhost` ne fait pas la différence entre les utilisateurs sur l'hôte distant. De plus, les noms d'hôtes (les adresses en fait) peuvent être falsifiés.

Cette méthode doit être évitée même avec des critères sur les hôtes plus restrictifs si vous êtes sur un réseau dans lequel vous ne pouvez avoir confiance (par exemple avec une connexion PPP sur Internet). Voir `xhost(1x)`.

9.4.5 Connexion X distante : `ssh`

L'utilisation de `ssh` active une connexion sûre d'un serveur X local vers un serveur d'applications distant.

- Réglez `X11Forwarding` et `AllowTcpForwarding` à `yes` dans `/etc/ssh/sshd_config` sur l'hôte distant.
- Démarrez le serveur X sur la station locale.
- Démarrez un `xterm` sur l'hôte local.
- Lancez `ssh` pour établir une connexion vers le site distant.

```
localname @ localhost $ ssh -q -X -l loginname remotehost.domain
Password:
.....
```

- Lancez des applications X sur le site distant.

```
loginname @ remotehost $ gimp &
```

Cette méthode permet l'affichage du client X distant comme s'il était connecté par une socket UNIX locale.

9.4.6 `xterm`

Apprenez tout sur `xterm` à l'adresse http://dickey.his.com/xterm/xterm.faq.html.

9.4.7 Devenir root sous X

Si une application graphique a besoin d'être lancée avec les privilèges de root, utilisez les procédures suivantes pour le faire à partir du serveur X d'un utilisateur. N'essayez jamais de démarrer un serveur X depuis le compte root, pour vous garder d'éventuels risques de sécurité.

Démarrez le serveur X avec un utilisateur normal et ouvrez un `xterm`. Ensuite :

```
$ XAUTHORITY=$HOME/.Xauthority
$ export XAUTHORITY
$ su root
Password:*****
# printtool &
```

Lorsque vous utilisez cette astuce pour effectuer un `su` vers un utilisateur non root, faites attention à ce que le fichier `$HOME/.Xauthority` soit lisible par le groupe de cet utilisateur.

Cette séquence de commandes peut être automatisée en ajoutant quelques fichiers. Depuis le compte root, créez le fichier `/etc/X11/Xsession.d/00xfree86-common_environment` avec les lignes suivantes :

```
if [ -f "$HOME/.xenvironment" ]; then
    . $HOME/.xenvironment
fi
```

Depuis le compte utilisateur, créez le fichier `$HOME/.xenvironment`, avec les lignes suivantes :

```
# Cela permet que X marche quand je fais un su pour devenir root
if [ -z "$XAUTHORITY" ]; then
    XAUTHORITY=$HOME/.Xauthority
    export XAUTHORITY
fi
```

Ensuite, exécutez `su` (pas `su -`) dans une fenêtre `xterm` de l'utilisateur. Maintenant vous pouvez exécuter des applications graphiques avec les privilèges de root sur l'écran X d'un utilisateur normal. Cette astuce fonctionne tant que le Xsession par défaut est utilisé. Si un utilisateur personnalise `$HOME/.xinit` ou `$HOME/.xsession`, la variable d'environnement `XAUTHORITY` mentionnée ci-dessus doit être positionnée de la même façon dans ces scripts.

Une autre méthode est d'utiliser le paquet `sudo` :

```
$ sudo xterm
... ou
$ sudo -H -s
```

Ici le fichier `/root/.bashrc` doit contenir :

```
if [ $SUDO_USER ]; then
    sudo -H -u $SUDO_USER xauth extract - $DISPLAY | xauth merge -
fi
```

Cela marche bien même si le répertoire home de l'utilisateur est sur une partition NFS, puisque root ne lit pas le fichier `.Xauthority`.

Il existe aussi des paquets spécialisés dans ce but : `kdesu`, `gksu`, `gksudo`, `gnome-sudo`, et `xsu`. Enfin, trois autres méthodes peuvent être utilisées pour atteindre des résultats similaires : créer un lien symbolique depuis `/root/.Xauthority` vers celui de l'utilisateur ; utiliser le script sux (http://fgouget.free.fr/sux/sux-readme.shtml) ; ou mettre `xauth merge ~USER_RUNNING_X/.X` dans le script d'initialisation de root.

Plus d'informations sur la liste de diffusion debian-devel (http://lists.debian.org/debian-devel/2002/debian-devel-200207/msg00259.html).

9.4.8 Polices TrueType dans X

Le `xfs` standard de XFree86-4 fonctionne bien avec les polices TrueType. Il faut installer un autre serveur de polices comme `xfs-xtt` si vous utilisez XFree86-3.

Vous avez simplement besoin de vous assurer que les applications que vous voulez utiliser avec les polices TrueType sont liées avec les bibliothèques libXft ou libfreetype (vous n'avez probablement pas à vous en soucier si vous utilisez des .deb précompilés).

Souvenez-vous d'installer les fichiers de polices et de générer les fichiers `fonts.{scale,dir}` afin que les fontes puissent être indexées et utilisées.

Puisque les polices **libres** sont parfois limitées, l'installation ou le partage de certaines polices TrueType commerciales est une option pour les utilisateurs Debian. Afin de rendre ce processus facile pour l'utilisateur, il existe certains paquets :
- `ttf-commercial`
- `msttcorefonts` (Plus utile depuis 08/2002 à cause d'un changement de politique chez MS)

Vous aurez ainsi une très bonne sélection de polices TT au prix de la contamination de votre système **libre** par des polices non-libres.

9.4.9 Navigateur Web (graphique)

Il existe quelques paquets de navigateurs web disponibles avec la sortie de Woody :
- `mozilla`Le navigateur Mozilla (nouveau)
- `galeon`Navigateur basé sur Mozilla avec une interface Gnome (nouveau)
- `konqueror`Navigateur KDE
- `amaya-gtk`Navigateur de référence du W3C
- `amaya-lesstif`Navigateur de référence du W3C
- `netscape-...` (plusieurs, vieux)
- `communicator-...` (plusieurs, vieux)
- ...

La version de `mozilla` doit correspondre avec la version que `galeon` requiert. A part l'interface graphique, ces deux logiciels partagent le moteur de rendu HTML Gecko.

Des ajouts pour les navigateurs comme `mozilla` et `galeon` peuvent être activés en installant des fichiers `*.so` directement dans le répertoire `plugin` et en relançant les navigateurs.

Ressources :
- Java : installer le binaire "J2SE" depuis `http://java.sun.com`.
- Flash : installer le binaire "Macromedia Flash Player 5" depuis `http://www.macromedia.com/software/flashplayer/`.
- `freewrl` : navigateur VRML et plugin Netscape
- ...

9.4.10 CJK et X

Références :
- 'Localisation et support des langues nationales' page 121

  - Pages Suse pour CJK (http://www.suse.de/~mfabian/suse-cjk/suse-cjk.html)

Ici, essayons pour le Japonais :

- installer les paquets pour le Japonais :
  - kinput2-canna-wnn — un serveur d'entrée pour les applications X11.
  - kterm, mlterm, et jfbterm : terminaux compatibles avec le Japonais.
  - egg — Input Method Architecture for Emacsen
  - canna — un système d'entrée japonais (serveur et dictionnaire).
  - freewnn-jserver — système de conversion Kana vers Kanji.
  - ...et tous les paquets de polices japonaises.

  En réalité, utilisez tasksel ou aptitude pour sélectionner « Japanese Environment » et éviter d'installer des logiciels qui entrent en conflit avec le système normal.

- ajouter une locale qui supporte les caractères japonais (par exemple ja_JP.UTF-8, voir 'Localisation et support des langues nationales' page 121).

- ajouter les valeurs d'environnement suivantes dans ~/.xenvironment en utilisant la même astuce que dans 'Devenir root sous X' page 112.

```
XMODIFIERS="@im=kinput2"
LC_CTYPE=ja_JP.UTF-8 # locale japonaise
```

  (Ou faites le manuellement dans un xterm avant de lancer une application.)

- activer XIM kinput2 en ajoutant *inputMethod : kinput2 dans votre fichier de ressources X (on dirait que Debian le fait).

- Quelques applications (comme mlterm) permettent aussi de configurer *inputMethod : dynamiquement au lancement (appuyez sur Ctrl-BoutonSouris-3 dans mlterm).

Une fois l'application démarrée, vous devez appuyer sur **Shift+Espace** et une fenêtre devrait s'ouvrir indiquant que vous pouvez entrer des caractères japonais.

9.5 SSH

SSH (Secure SHell) est le moyen sécurisé de se connecter à d'autres machines au travers d'Internet. Une version libre de SSH appelée OpenSSH est disponible dans le paquet ssh de Debian.

9.5.1 Bases

Installez d'abord le serveur et le client OpenSSH.

```
# apt-get update && apt-get install ssh
```

L'entrée non-US doit être présente dans le fichier `/etc/apt/source.list` Le fichier `/etc/ssh/sshd_not_to_be_run` ne doit pas être présent si l'on veut exécuter le serveur OpenSSH.

SSh possède 2 protocoles d'identification :

- Protocole SSH version 1 :
  - la version de Potato supporte seulement ce protocole.
  - méthodes d'identification disponibles :
    - RSAAuthentication : identification basée sur une clé RSA
    - RhostsAuthentication : identification basée sur .rhosts (non sûr, désactivé)
    - RhostsRSAAuthentication : identification .rhosts combinée avec clé RSA (désactivé)
    - ChallengeResponseAuthentication : identification par challenge-response RSA
    - PasswordAuthentication : identification basée sur un mot de passe
- Protocole SSH version 2 :
  - les versions d'après Woody utilisent ce protocole par défaut.
  - méthodes d'identification disponibles :
    - PubkeyAuthentication : identification de l'utilisateur basée sur une clé publique
    - HostbasedAuthentication : identification par `.rhosts` ou `/etc/hosts.equiv` combinée avec identification de l'hôte par clé publique (désactivé)
    - ChallengeResponseAuthentication : identification par challenge-response RSA
    - PasswordAuthentication : identification basée sur un mot de passe

Faites attention avec ces différences si vous migrez vers Woody ou si vous utilisez un système non Debian.

Voir `/usr/share/doc/ssh/README.Debian.gz`,`ssh(1)`,`sshd(8)`,`ssh-agent(1)`,et `ssh-keygen(1)` pour plus de détails.

Les fichiers suivants sont les fichiers de configuration importants.

- `/etc/ssh/ssh_config` : valeurs par défaut pour le client SSH. Voir `ssh(1)`. Entrées à noter :
  - `Host` : Restreint les entrées suivantes (jusqu'au prochain Host) aux hôtes qui correspondent au motif donné après le mot clé.
  - `Protocol` : Spécifie la version du protocole SSH. Par défaut, « 2,1 ».
  - `PreferredAuthentications` : Spécifie la méthode d'identification pour le client SSH2. Par défaut, « hostbased,publickey,keyboard-interactive,password ».
  - `ForwardX11` : Désactivé par défaut. Peut être outrepassé par l'option `-X` en ligne de commande.
- `/etc/ssh/sshd_config` : valeurs par défaut pour le serveur SSH. Voir `sshd(8)`. Entrées à noter :
  - `ListenAddress` : Spécifie les adresses locales que `sshd` doit écouter. Plusieurs options sont permises.
  - `AllowTcpForwarding` : Désactivé par défaut.
  - `X11Forwarding` : Désactivé par défaut.

– `$HOME/.ssh/authorized_keys` : liste de clés publiques par défaut utilisées pour se connecter à ce compte sur cette machine. Voir `ssh-keygen(1)`.
– `$HOME/.ssh/identity` : Voir `ssh-add(1)` et `ssh-agent(1)`.

Les commandes suivantes permettent de démarrer une connexion `ssh`.

```
$ ssh username@hostname.domain.ext
$ ssh -1 username@hostname.domain.ext # Force SSH version 1
```

Pour un utilisateur, `ssh` est plus intelligent que `telnet` (il ne crashe pas avec ^]).

9.5.2 Redirection de port – tunnel SMTP/POP3

Pour établir un tunnel pour se connecter au port 25 de *remote-server* depuis le port 4025 de la machine locale, et au port 110 de *remote-server* depuis le port 4110 de la machine locale par `ssh`, exécutez les commandes suivantes sur la machine locale :

```
# ssh -q -L 4025:remote-server:25 4110:remote-server:110 \
          username@remote-server
```

C'est un moyen sécurisé d'effectuer une connexion vers un serveur SMTP/POP3 au travers d'Internet. Positionnez l'entrée `AllowTcpForwarding` à `yes` dans `/etc/ssh/sshd_config` sur la machine distante.

9.5.3 Se connecter avec moins de mots de passe

On peut éviter de se rappeler le mot de passe pour chaque système distant en utilisant RSAAuthentication (protocole SSH1) ou PubkeyAuthentication (protocole SSH2).

Sur le système distant, le contenu de `/etc/ssh/sshd_config` doit contenir « RSAAuthentication yes » ou « PubkeyAuthentication yes ».

Générez ensuite les clés d'identification localement et installez la clé publique sur le système distant :

```
$ ssh-keygen          # RSAAuthentication~: clé RSA1 pour SSH1
$ cat .ssh/id_rsa.pub | ssh user1@remote \
        "cat - >>.ssh/authorized_keys"
...
$ ssh-keygen -t rsa   # PubkeyAuthentication~: clé RSA pour SSH2
```

```
$ cat .ssh/id_rsa.pub | ssh user1@remote \
        "cat - >>.ssh/authorized_keys"
...
$ ssh-keygen -t dsa   # PubkeyAuthentication~: clé DSA pour SSH2
$ cat .ssh/id_dsa.pub | ssh user1@remote \
        "cat - >>.ssh/authorized_keys"
```

On peut changer la phrase de passe par la suite avec `ssh-keygen -p`. Assurez-vous de vérifier la configuration en testant la connexion. En cas de problème, utilisez `ssh -v`.

Vous pouvez ajouter des options dans les entrées de `authorized_keys` pour limiter les hôtes et exécuter des commandes spécifiques. Consultez `sshd(8)` pour plus de détails.

Notez que SSH2 fait une identification de type `HostbasedAuthentication`. Pour que cela fonctionne, il faut régler la configuration de `HostbasedAuthentication` à `yes` dans `/etc/ssh/sshd_config` sur le serveur et `/etc/ssh/ssh_config` ou `$HOME/.ssh/config` sur le client.

9.5.4 Clients SSH étrangers

Il existe quelques clients SSH gratuits disponibles pour des systèmes non Unix.

Windows puTTY (http://www.chiak.greenend.org.uk/~sgtatham/putty/) (GPL)

Windows (cygwin) SSH in cygwin (http://www.cygwin.com/) (GPL)

Macintosh Classic macSSH (http://www.macssh.com/) (GPL) [Notez que Mac OS X inclut OpenSSH ; utilisez `ssh` depuis l'application Terminal]

Voir SourceForge.net, documentation du site (http://www.sourceforge.net/docman/?group_id=1), « 6. CVS Instructions ».

9.5.5 SSH agent

Mettez simplement votre clé publique dans `~/.ssh/authorized_keys`, et c'est bon :

```
$ ssh-agent
$ # copiez la sortie dans votre shell
$ ssh-add .ssh/identity
$ # ou ssh-add .ssh/id_dsa selon le nom de votre clé privée
$ scp remote.host.with.public.key
```

Pour plus d'information, consultez `ssh-agent(1)` et `ssh-add(1)`.

9.5.6 Problèmes

Si vous rencontrez des problèmes, vérifiez les permission du fichier de configuration et lancez `ssh` avec l'option `-v`.

Utilisez `-P` si vous êtes root et avez des problèmes avec un pare-feu ; cela évite d'utiliser des ports serveur 1–1023.

Si les connexions `ssh` vers un site distant arrêtent subitement de fonctionner, cela peut être suite à des bidouilles de l'administrateur, sûrement un changement de `host_key` pendant une maintenance du système. Après s'être assuré que c'est bien le cas et que personne n'essaie de se faire passer pour la machine distante par une bidouille, on peut se reconnecter en enlevant l'entrée `host_key` de `~/.ssh/known_hosts` sur la machine locale.

9.6 Logiciels de courrier électronique

La configuration du courrier électronique se divise en trois parties :

- MTA : `exim`
- MUA : `mutt`
- Utilitaires : `procmail`, `fetchmail`, `mail`,...

9.6.1 Agent de transport de courrier électronique (Exim)

Références :
  - paquets `exim-doc` et `exim-doc-html`
  - http://www.exim.org/

Utilisez `exim` comme agent de transport de courrier électronique (Mail Transport Agent, MTA). Configurez-le :

```
/etc/exim.conf          "eximconfig" pour créer et éditer
/etc/inetd.conf         mettez smtp en commentaire pour exécuter exim en tant
que daemon
/etc/email-addresses    ajoutez les listes d'adresses sources spoofées
Vérifiez le filtre avec exim -brw, -bf, -bF, -bV, ... etc.
```

9.6.2 Tout récupérer pour les adresses e-mail non existantes (Exim)

Dans `/etc/exim/exim.conf` (Woody ou plus), dans la partie DIRECTORS, à la fin (après localuser : director) ajoutez une directive "catch-all" qui correspond à toutes les adresses que la directive précédente ne pouvait résoudre (par Miquel van Smoorenburg) :

```
catchall:
  driver = smartuser
    new_address = webmaster@mydomain.com
```

Si vous souhaitez une recette plus détaillée pour chaque domaine virtuel, ajoutez ce qui suit à la fin de `exim.conf` (par moi, pas bien testé) :

```
*@yourdomain.com ${lookup{$1}lsearch*{/etc/email-addresses} \
        {$value}fail} T
```

Et ajoutez une entrée « * » dans `/etc/email-addresses`.

9.6.3 Utilitaire de courrier électronique (fetchmail)

`fetchmail` est exécuté en mode démon et récupère le courrier électronique du compte POP3 du FAI sur le système de mail local. Configurez :

```
/etc/init.d/fetchmail   voir ci-dessous pour le script
/etc/rc?.d/???fetchmail exécutez update-rc.d fetchmail defaults 30
/etc/fetchmailrc        fichier de configuration (chown 600)
/etc/init.d/fetchmail
```

Les informations pour démarrer fetchmail en mode démon depuis un script `init.d` dans Potato ne sont pas claires (Woody corrige cela). Voir /etc/init.d/fetchmail et /etc/fetchmailrc (`examples/`).

Si vos en-têtes de courriers électroniques sont contaminées par des ^M dûs au logiciel de courrier électronique de votre FAI, ajoutez « stripcr » aux options dans `$HOME/.fetchmailrc`:

```
options fetchall no keep stripcr
```

9.6.4 Utilitaire de courrier électronique (procmail)

`procmail` est un filtre pour le courrier électronique délivré. Il faut créer un `$HOME/.procmailrc` pour chaque compte qui l'utilise. Exemple : _procmailrc (`examples/`).

9.6.5 Agent pour utilisateur de courrier électronique (Mutt)

Utilisez `mutt` comme agent pour utilisateur de courrier électronique (Mail User Agent, MUA) associé à `vim`. Personnalisez-le dans `~/.muttrc` :

```
# utiliser le mode visuel et "gq" pour reformater les citations
set editor="vim -c 'set tw=72 et ft=mail'"
#
# sélection des en-têtes prise dans le manuel ("Sven's Draconian header
# weeding")
#
ignore *
unignore from: date subject to cc
unignore user-agent x-mailer
hdr_order from subject to cc date user-agent x-mailer
auto_view application/msword
....
```

Ajoutez ce qui suit dans `/etc/mailcap` ou `$HOME/.mailcap` pour afficher les courriers électroniques en HTML et les attachements MS Word dans le message :

```
text/html; lynx -force_html %s; needsterminal;
application/msword; /usr/bin/antiword '%s'; copiousoutput;
description="Microsoft Word Text"; nametemplate=%s.doc
```

9.7 Localisation et support des langues nationales

Debian est internationalisé et supporte un nombre toujours plus élevé de langues et de conventions locales. La sous-section suivante liste quelques formes de la diversité que Debian supporte actuellement, et la sous-section suivante parle de la **localisation**, processus de personnalisation de votre environnment de travail pour permettre l'entrée et l'affichage de votre langue et conventions pour les dates, les formats numériques et monétaires, et d'autres aspects d'un système qui diffèrent selon votre région.

9.7.1 Support des langues

Clavier Debian est distribuée avec le support pour près de deux douzaines de claviers, et avec des utilitaires (dans le paquet `kbd`) pour installer, visualiser et modifier les tables de caractères. Si vous avez besoin d'installer un clavier différent, exécutez simplement

```
#dpkg-reconfigure console-date
```

et debconf vous demandera la carte de clavier à utiliser.

Données La grande majorité des paquets de logiciels Debian supporte l'entrée de caractères non US-ASCII grâce à la technologie des `locales` offerte par la glibc (voir 'Locales' de la présente page.

- Propres en 8 bits : presque tous les programmes
- Autres langues latines (par exemple ISO-8859-1 ou ISO-8859-2) : majorité des programmes
- langues multi-octets comme le Japonais ou le Coréen : beaucoup de nouvelles applications

Affichage X supporte toutes les polices. La liste inclut non seulement les polices 8 bits mais aussi des polices 16 bits comme pour le Chinois, le Japonais et le Coréen. Voir 'CJK et X' page 114.

Traduction Des traductions existent pour beaucoup de messages texte et documents qui sont affichés par le système Debian, comme les messages d'erreur, l'affichage standard des programmes, les menus et les pages de manuel. Actuellement, le support pour les langues Allemand, Espagnol, Finlandais, Français, Hongrois, Italien, Japonais, Koréen et Polonais dans les pages de manuel est fourni par les paquets `manpages-`*`LANG`* (où *LANG* est le code ISO à deux lettres du pays).

Pour accéder à une page de manuel NLS, l'utilisateur doit configurer la variable du shell LC_MESSAGES de la façon appropriée. Par exemple, dans le cas des pages de manuel en Italien, LC_MESSAGES doit être positionnée à `it`. Le programme `man` cherchera alors les pages de manuel en Italien sous `/usr/share/man/it/`. Voir 'Locales' de la présente page pour plus d'information sur la configuration des variables de locale.

9.7.2 Locales

Debian supporte la technologie **locale**. C'est un mécanisme qui permet aux programmes de fournir un affichage et des fonctionnalités adaptés aux conventions locales comme le jeu de caractères, le format de la date et de l'heure, le symbole de la monnaie, et ainsi de suite. Il utilise des variables d'environnement pour déterminer le comportement approprié. Par exemple, en supposant que vous avez les locales Anglais américain et Français installées sur votre système, les messages d'erreur de beaucoup de programmes peuvent être bilingues :

```
$ LANG="en_US" cat foo
cat: foo: No such file or directory
$ LANG="fr_FR" cat foo
cat: foo: Aucun fichier ou répertoire de ce type
```

La glibc offre le support pour cette fonctionnalité dans la bibliothèque. Voir `locale(7)`.

9.7.3 Activer le support des locales

Debian n'est **pas** livrée avec toutes les locales disponibles précompilées. Consultez le fichier `/usr/lib/locale` pour voir quelles locales (à part le défaut, « C ») sont compilées pour votre système. Si celle dont vous avez besoin n'est pas présente, vous avez deux possibilités :

- Editer `/etc/locale.gen` pour ajouter la locale souhaitée, et lancer `locale-gen` en tant que root pour la compiler. Voir `locale-gen(8)` et les pages de manuel listées dans la section « Voir aussi ».
- Lancer `dpkg-reconfigure locales` pour reconfigurer le paquet `locales`. Ou s'il n'est pas déjà installé, l'installation de `locales` invoquera l'interface debconf pour choisir les locales désirée et les compiler.

9.7.4 Activer une locale particulière

Les variables d'environnement suivantes sont évaluées dans cet ordre pour fournir les valeurs de locale aux programmes :

1. LANGUAGE : cette variable d'environnement consiste en une liste de locales séparées par deux points verticaux, dans l'ordre de priorité. Utilisé seulement si la locale POSIX est positionnée à une valeur autre que "C" [dans Woody ; la version de Potato a toujours priorité sur la locale POSIX]. (GNU extension)
2. LC_ALL : Si cette variable est non nulle, sa valeur est utilisée pour toutes les catégories de locales. (POSIX.1) Habituellement "" (null).
3. LC_* : Si cette variable est non nulle, sa valeur est utilisée pour la catégorie correspondante. (POSIX.1) Habituellement "C".

   Les variables LC_* sont :
   - LC_CTYPE : Classification des caractères et conversion de casse.
   - LC_COLLATE : Ordre de collation.
   - LC_NUMERIC : Formats des nombres non monétaires.
   - LC_MONETARY : Formats monétaires.
   - LC_MESSAGES : Formats des messages d'information et de diagnostic et des réponses intéractives.
   - LC_PAPER : Taille du papier.
   - LC_NAME : Formats des noms.
   - LC_ADDRESS : Formats des adresses et information sur les lieux.
   - LC_TELEPHONE : Formats des numéros de téléphone.
   - LC_MEASUREMENT : Unités de mesure (Metric ou Other).
   - LC_IDENTIFICATION : Donnée générale sur l'information de la locale.
4. LANG : Si cette variable est non nulle, et que LC_ALL n'est pas définie, sa valeur est utilisée pour toutes les catégories de locales LC_* sans valeur définie. (POSIX.1) Habituellement, "C".

Notez que certaines applications (par exemple Netscape 4) ignorent les valeurs LC_*.

Le programme `locale` peut afficher la locale active et les locales disponibles ; voir `locale(1)`. (NOTE : `locale -a` liste toutes les locales du système ; cela ne signifie pas qu'elles sont toutes compilées ! Voir 'Activer le support des locales' page précédente.)

NdT : pour configurer Debian pour le Français avec le support de l'Euro, voir le document Utiliser et configurer Debian pour le Français (`http://www.debian.org/doc/manuals/fr/debian-fr-howto/`).

9.7.5 Après `locale`

Certains programmes peuvent nécessiter une configuration en plus de la `locale` pour obtenir un environnement de travail confortable.

Le paquet `language-env` avec sa commande `set-language-env` est un script qui facilite la configuration de l'environnement de langue nationale sur un système Debian.

De plus, des entrées spécifiques à chaque langue dans le système de « tâches » que l'on accède avec `tasksel` ou `aptitude` sont une autre ressource utile.

Pour un example avec X, voir 'CJK et X' page 114.

Pour plus d'information, voir Pages Suse pour CJK (`http://www.suse.de/~mfabian/suse-cjk/suse-cjk.html`). Voir aussi le document sur l'internationalisation, Introduction to i18n (`http://www.debian.org/doc/manuals/intro-i18n/`). Il est destiné aux développeurs mais est aussi utile aux administrateurs système.

Chapitre 10

Construire une passerelle avec Debian

Linux permet d'avoir une machine passerelle tout usage, avec NAT, mail, DHCP, cache DNS, cache proxy HTTP, CVS, NFS, et services Samba pour un réseau local personnel.

10.1 Configuration réseau

10.1.1 Configuration de la passerelle

Le réseau local utilise des adresses IP des plages ci-dessous pour éviter les collisions d'adresses avec l'Internet.

```
Class A: 10.0.0.0                    avec masque 255.0.0.0
Class B: 172.16.0.0 - 172.31.0.0     avec masque 255.255.0.0
Class C: 192.168.0.0 - 192.168.255.0 avec masque 255.255.255.0
```

Debian utilise `/etc/network/interfaces` pour la configuration IP.

Par exemple, si `eth0` est connectée à l'Internet avec une adresse IP fournie par DHCP et `eth1` est connectée au réseau local, `/etc/network/interfaces` est configuré comme suit (Woody ou version plus récente) :

```
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
```

```
address 192.168.1.1
network 192.168.1.0
netmask 255.255.255.0
broadcast 192.168.1.255
```

Exécutez les commandes suivantes pour mettre à jour la configuration réseau avec le nouveau `/etc/network/interfaces` :

```
# /etc/init.d/networking restart
```

Rappel : le fichier `interfaces` de Woody est incompatible avec celui de Potato.

Si le système utilise une carte réseau PCMCIA, il faut configurer le réseau dans `/etc/pcmcia/network.opts` à la place.

En cas de doute, consultez la sortie des commandes suivantes :

```
# ifconfig
# cat /proc/pci
# cat /proc/interrupts
# dmesg|more
```

Parfois, les connexions DSL (PPPoE ?) ont des problèmes de MTU. Consultez le DSL-HOWTO (http://www.linuxdoc.org/HOWTO/DSL-HOWTO/index.html) du LDP.

10.1.2 IP-masquerade

Les machines du LAN peuvent accéder à l'Internet à travers une passerelle qui utilise IP-masquerade (NAT).

```
# apt-get install ipmasq
```

Appliquez les règles de l'exemple pour fortifier la protection `ipmasq`. Consultez `/usr/share/doc/ipmasq/examples/stronger/README`. Pour l'image du noyau Debian 2.4, assurez-vous de charger les modules appropriés. Consultez 'Fonctions réseau' page 78.

Pour l'image du noyau Debian 2.2, éditez `Z92timeouts.rul` dans `/etc/masq/rules` comme suit :

```
# tcp, tcp-fin, udp
# 2hr, 10 sec, 160 sec - default
# 1 day, 10 min, 10 min - longer example
$IPCHAINS -M -S 86400 600 600
```

De plus, si le réseau est accédé par une carte PCMCIA, `ipmasq` a besoin d'être démarré depuis `/etc/pcmcia/network.opts`. Consultez `/usr/share/doc/ipmasq/ipmasq.txt.gz`.

10.1.3 Points principaux de la configuration réseau pour une passerelle

Ensemble de programmes typique :

```
# apt-get install nfs samba dhcpd dhcp-client bind squid procmail fetchmail
# apt-get install ssh cvs
```

Ensuite, consultez les fichiers suivants :

```
/etc/init.d/dhcpd       (éditer pour avoir LAN = eth1)
/etc/host.allow         (ALL: 192.168.0.0/16 127.0.0.0/8) pour NFS
/etc/exports            (besoin pour NFS)
/etc/bind/db.192.168.1  (à ajouter)
/etc/bind/db.lan        (à ajouter)
/etc/bind/named.conf    (à éditer)
/etc/resolve.conf       (à éditer)
/etc/hosts
/etc/dhcpd.conf         (éditer pour avoir LAN = eth1)
/etc/dhclient.conf      (éditer pour forcer le DNS local)
/etc/samba/smb.conf
/etc/exim/exim.conf
/etc/mailname
/etc/aliases
/etc/squid.conf         (ajouter toutes les adresses IP des machines du LAN
comme machines autorisées)
```

`bind` crée un cache DNS local et transforme l'hôte local en serveur DNS. Consultez le fichier `/etc/resolve.conf` :

```
nameserver 127.0.0.1
search lan.aokiconsulting.com
```

10.2 Gérer plusieurs connexions à l'Internet

[FIXME] Politique de routage : (par Phil Brutsche pbrutsch@tux.creighton.edu) Voir le Manuel d'iproute (http://ds9a.nl/2.4Routing/) pour les détails. Le contrôle de traffic peut aussi être intéressant.

Environnement :

```
eth0: 192.168.1.2/24; gateway 192.168.1.1
eth1: 10.0.0.2/24; gateway 10.0.0.1
Pas de masquerading sur cette machine.
```

Un peu de magie :

```
# ip rule add from 192.168.1.2 lookup 1
# ip rule add from 10.0.0.2 lookup 2
# ip route add to default via 10.0.0.1 metric 0
# ip route add to default via 192.168.1.1 metric 1
# ip route add table 1 to 192.168.1.0/24 via eth0
# ip route add table 1 to 10.0.0.2/24 via eth1
# ip route add table 1 to default via 192.168.1.1
# ip route add table 2 to 192.168.1.0/24 via eth0
# ip route add table 2 to 10.0.0.2/24 via eth1
# ip route add table 2 to default via 10.0.0.2
```

[FIXME] Je n'ai jamais fait cela. Comment configurer une connexion RTC en cas de défaillance d'une connexion rapide ?

Chapitre 11

Editeurs

11.1 Editeurs populaires

Linux offre plusieurs solutions comme éditeurs en mode texte. Parmi eux :
- vim : Éditeur puissant et léger hérité de BSD. VI iMproved (VI aMélioré).
- emacs : L'éditeur ultime et lourd hérité de GNU. L'original de RMS (Richard M. Stallman).
- xemacs : Emacs : The Next Generation (la nouvelle génération), à l'origine par Lucid.
- mcedit : Éditeur GNU pour débutant. Identique à l'éditeur interne de mc.
- ae : Petit éditeur par défaut (Potato). À éviter.
- nano : Petit éditeur GNU par défaut (Woody). Emule pico.
- joe : Pour les vieux habitués de WordStar ou de TurboPascal.
- jed : Éditeur rapide et complet avec menus et raccourcis emacs.
- jove : Très petit éditeur avec raccourcis emacs.
- nvi : Nouveau vi. Compatible bogue pour bogue avec le vi original.

Utilisez `update-alternatives --config editor` pour changer d'éditeur par défaut.

Il existe aussi quelques éditeurs de texte pour X :
- `gvim` : Vim avec une interface graphique (paquet `vim-gtk`)
- `emacs` : Le vrai Emacs (détecte X automatiquement)
- `xemacs` : L'Emacs nouvelle génération (détecte X automatiquement)

Ces commandes xclient prennent des options standard comme `-fn` *`a24`* qui rendent les choses faciles pour les plus vieux comme moi :-) Voir 'Client X' page 110.

11.2 Editeurs de sauvetage

Il y a peu d'éditeurs dans `/bin`. L'un d'eux doit être installé pour faciliter l'édition de fichiers quand `/usr` n'est pas accessible.

- `elvis-tiny` : Editeur minimum type vi (`vi` pour le démarrer)
- `nano-tiny` : Editeur minimum non vi (`nano-tiny` pour le démarrer)
- `ed` : Editeur minimum (toujours là mais difficile à utiliser)

11.3 Emacs et Vim

11.3.1 Conseils pour Vim

Lisez la page "VIM - main help file" en tapant <F1>.

```
<F1>     Aide
<esc>    Retour au mode normal
V              Mode visuel
i              Mode insertion
:              Commandes en ligne
:set tw=72     Paramètre la largeur du texte à 72
<F11>    Mode Insertion (collage)
:r! date -R    Insertion de la date selon RFC-822
qa  Enregistre la frappe dans le registre a
@a  Exécute la frappe enregistrée dans le registre a
:edit foo.txt Edite un autre fichier en chargeant
foo.txt
:wnext         Enregistre le fichier courant et édite le fichier suivant
```

`q` et `@` peuvent être utilisés pour enregistrer et exécuter de simples macros. Par exemple, pour créer une macro qui insère des balises HTML pour l'italique autour du mot sur lequel se trouve le curseur, vous pourriez taper `qii<i>^[ea</i>^[q` (où `^[` est la touche ESC). Taper `@i` au début d'un mot ajoute les balises et .

11.3.2 Conseils pour Emacs

```
<F1>          Aide
<F10>         Menu
C-u M-! date -R     Insertion de la date selon RFC-822
```

11.3.3 Démarrer l'éditeur

```
démarrer l'éditeur : emacs nom_de_fichier vim nom_de_fichier
```

```
démarrage en compatibilité vi : vim -C
démarrage en non-compatible vi : vim -N
démarrage en mode par défaut : emacs -q vim -N -u NONE
```

11.3.4 Résumé des commandes de l'éditeur (Emacs, Vim)

```
Sortie~:                    C-x C-c         :qa /:wq /:xa /:q!
Retour/mode commande~:    C-g             <esc>
Retour(gauche)~:          C-b             h
Après(droite)~:           C-f             l
Suivant(en bas)~:         C-n             j
Précédent(en haut)~:      C-p             k
Début de ligne(^)~:       C-a             0
Fin de ligne($)~:         C-e             $
Commandes mUltiples~:     C-u nnn cmd     :count cmd
Commandes Multiples~:     M-digitkey cmd
Sauvegarder le fichier~:  C-x C-f         :w file
Début du tampon~:         M-<          1G
Fin du tampon~:           M->           G
Écran suivant~:           C-v             ^F
1/2 écran suivant~:                       ^D
Ligne suivante~:                          ^E
Revenir d'1 écran~:       M-v             ^B
Revenir d'1/2 écran~:                     ^U
Ligne précédente~:                        ^Y
Avance l'autre fenêtre~:  M-C-v
Effacer sous le curseur~: C-d             x
Effacer à partir du curseur
jusqu'à la fin de ligne~: C-k             D
iRecherche en avant~:     C-s
iRecherche en arrière~:   C-r
Chercher en avant~:       C-s enter       /
Chercher en arrière~:     C-r enter       ?
iRecherche avec
expression rationnelle~:  M-C-s
iRecherche en arrière avec
expression rationnelle~:  M-x isearch-backward-regexp
Chercher avec
expression rationnelle~:  M-C-s enter     /
Chercher en arrière avec
expression rationnelle~:  M-x isearch-backward-regexp enter
```

```
                                                  ?
Aide~:                    C-h C-h         :help
Aide Apropos~:            C-h a
Aide sur les liens~:      C-h b           :help key
Aide Info~:               C-h i
Help mode Majeur~:        C-h m
Aide tutorial~:           C-h t           :help howto
Annulation~:              C-_             u
Refaire~:                 C-f             ^R
Marquer la position
du curseur~:              C-@             m{a-zA-Z}
Échange Marque
et position~:             C-x C-x
Aller à la marque dans
le fichier actuel~:                       '{a-z}
Aller à la marque dans
n'importe quel fichier~:                  '{A-Z}
Copier région~:           M-w             {visual}y
Tuer région~:             C-w             {visual}d
Copier et garder tampon~: C-y
Copier depuis le tampon~: M-y             p
Conversion d'une zone
en majuscules~:           C-x C-u
Conversion d'une zone
en minuscules~:           C-x C-l
Insère un caractère
spécial~:                 C-q octalnum/keystroke  ^V decimal/keystroke
Remplace~:                M-x replace-string      :%s/aaa/bbb/g
Remplace regexp~:         M-x replace-regexp      :%s/aaa/bbb/g
Recherche et remplace~:   M-%                     :%s/aaa/bbb/gc
Recherche et remplace~:   M-x query-replace
Recherche et remplace avec
expression rationnelle~:  M-x query-replace-regexp
Ouvre un fichier~:        C-x C-f         :r file
Sauvegarde le fichier~:   C-x C-s         :w
Sauve tous les tampons~:  C-x s           :wa
Sauvegarde sous~:         C-x C-w file    :w file
Demande un tampon~:       C-x b
Liste le tampon~:         C-x C-b         :buffers
Commute en lecture seule~:C-x C-q         :set ro
Demande et tue le tampon~:C-x k
Scinde verticalement~:    C-x 2           :split
```

```
Scinde horizontalement~:  C-x 3           :vsplit (ver. 6)
Va à une autre fenêtre~:  C-x o           ^Wp
Efface cette fenêtre~:    C-x 0           :q
Efface autre fenêtre~:    C-x 1
Lance un shell
en arrière-plan~:         M-x compile
Tue un shell lancé
en arrière-plan~:         M-x kill-compilation
Lance make~:                              :make Makefile
Inspecte les messages d'erreur~: C-x`
Lance un shell et enregistre~:   M-x shell       :!script -a tmp
...clean BS, ...                                :!col -b <tmp >record
sauve/rappelle le shell/enregistre~: C-x C-w record  :r record
Lance shell~:                    M-! sh          :sh
Lance une commande~:             M-! cmd         :!cmd
Lance une commande et insère~:   C-u M-! cmd     :r!cmd
Lance un filtre~:                M-| file        {visual}:w file
Lance un filtre et insère~:      C-u M-| filter  {visual}:!filter
Montre une option                                :se[t] {option}?
relance l'option                                 :se[t] {option}&
relance une option booléenne                     :se[t] no{option}
commute une option booléenne                     :se[t] inv{option}
met le texte sur 72 colonnes                     :se tw=72
pas d'organisation du texte                      :se tw=0
autoindentation                                  :se ai
étend la tabulation                              :se et
spécifie un commentaire (courrier)               :se comments=n:>,n:\|

Lance GDB                       M-x gdb
décrit le mode GDB              C-h m
saute une ligne                 M-s
va à la ligne suivante          M-n
saute une instruction (stepi)   M-i
Finit le cadre de la pile courante      C-c C-f
continue                        M-c
up arg frames                   M-u
down arg frames                 M-d
copie le numéro à partir du point, insère à la fin
                                C-x &
positionne un point d'arrêt C-x SPC
```

11.3.5 Configuration de Vim

Afin de mettre en surbrillance les spécificités et la syntaxe avec Vim, ajoutez les lignes suivantes dans `~/.vimrc` ou `/etc/vimrc` :

```
set nocompatible
set nopaste
set pastetoggle=<f11>
syn on
```

Le mode paste permet d'éviter que l'autoindentation n'interfère avec le couper/coller en mode console/terminal. C'est mieux que faire un simple " :set noai".

11.3.6 Ctags

`apt-get install exuberant-ctags` et lancez ctags dans les fichiers source. Taper `:tag` *`function_name`* dans vim pour aller à la ligne commence *function_name*. Les balises marchent pour C, C++, Java, Python, beaucoup d'autres langages.

Emacs a les mêmes possibilités avec les ctags.

11.3.7 Convertir un écran avec syntaxe en surbrillance en un source HTML

`so \$VIMRUNTIME/syntax/2html.vim` avec Vim en mode commande convertira le texte en texte html. Sauvegarder avec `:w file.html` et `:q`. Très utile pour du code C, etc.

11.3.8 Ecran scindé avec `vim`

`vim` peut éditer plusieurs fichiers dans un environnement multi-fenêtré en scindant l'écran. Tapez `:help usr_08.txt` pour plus de détails.

Pour scinder l'écran pour afficher différents fichiers, tapez à l'invite de commande de vi :

```
:split another-file
:vsplit another-file
```

Ou à l'invite du shell :

```
$ vi -o file1.txt file2.txt   # Séparation horizontale
$ vi -O file1.txt file2.txt   # Séparation verticale
```

lancera vi en mode multi-fenêtré.

```
$ vimdiff file.txt~ file.txt         # donne les changements récents de
file.txt
$ vimdiff file.en.sgml file.fr.sgml  # donne les changements dans la
traduction
$ gvimdiff file.txt~ file.txt        # sous X
```

fournit une vue sympa des différences entre un original et un fichier de sauvegarde. En SGML, cela fait la correspondance des balises, donc comparer les traductions dans ce mode marche très bien.

Mouvements spéciaux du curseur avec CTRL-W :

```
CTRL-W +      augmente la taille d'une fenêtre
CTRL-W -      réduit la taille d'une fenêtre
CTRL-W h      se déplace vers la fenêtre de gauche
CTRL-W j      se déplace vers la fenêtre en dessous
CTRL-W k      se déplace vers la fenêtre au dessus
CTRL-W l      se déplace vers la fenêtre de droite
...
```

Les commandes suivantes permettent de contrôler le défilement de l'écran :

```
:set scrollbind
:set noscrollbind
```

Chapitre 12

CVS

Consultez `/usr/share/doc/cvs/html-cvsclient`,`/usr/share/doc/cvs/html-info`,`/usr/share/doc/cvsbook` avec lynx ou lancez `info cvs` et `man cvs` pour de l'information détaillée.

12.1 Installation d'un serveur CVS

La configuration suivante autorise un "commit" sur le dépôt cvs seulement par un membre du groupe "src", et l'administration de cvs seulement par un membre du groupe "staff", ceci afin de réduire les risques de conflits.

```
# cd /var/lib; umask 002 ; sudo mkdir cvs # [WOODY] FSH
# apt-get install cvs cvs-doc cvsbook
# export CVSROOT=/var/lib/cvs
# cd $CVSROOT
# chown root:src .
        # "staff" => restriction importante pour un projet débutant.
# chmod 3775 .
        # Si la ligne précédente utilise "staff", alors régler sur 2775
# cvs -d=/var/lib/cvs init
        # il est plus prudent de spécifier explicitement -d ici !
# cd CVSROOT
# chown -R root:staff .
# chmod 2775 .
# touch val-tags
# chmod 664 history val-tags
# chown root:src history val-tags
```

12.2 Exemples de sessions CVS

12.2.1 CVS anonyme (téléchargement seulement)

```
$ export CVSROOT=:pserver:anonymous@cvs.qref.sf.net:/cvsroot/qref
$ cvs login
$ cvs -z3 co qref
```

12.2.2 Utilisation d'un serveur CVS local

```
$ export CVSROOT=/var/lib/cvs
```

12.2.3 Utilisation d'un pserver CVS distant

Ceci n'est pas très sûr, mais convenable pour du CVS anonyme :

```
$ export CVSROOT=:pserver:account@cvs.foobar.com:/var/lib/cvs
$ cvs login
```

12.2.4 Utilisation d'un CVS distant avec `ssh`

Vous pouvez aussi utiliser l'authentification RSA ('Se connecter avec moins de mots de passe' page 117) qui élimine la demande du mot de passe.

```
$ export CVSROOT=:ext:account@cvs.foobar.com:/var/lib/cvs
```

ou pour SourceForge :

```
$ export CVSROOT=:ext:account@cvs.qref.sf.net:/cvsroot/qref
```

12.2.5 Créer une nouvelle archive CVS

Pour,

```
OBJET                VALEUR           SIGNIFICATION
Arbre source :       ~/projet-x       Tous les codes sources
Nom du Projet :      projet-x         Nom pour ce projet
Balise vendeur :     Main-branch      Balise pour la branche entière
Balise de version :  Release-original Balise pour une version spécifique
```

Ensuite,

```
$ cd ~/projet-x                 # va dans le répertoire du source
 ... créer un arbre source ...
$ cvs import -m "Start projet-x" projet-x Main-branch Release-initial
$ cd ..; rm -R ~/projet-x
```

12.2.6 Travailler avec CVS

Pour rappeler et travailler sur les sources locales de projet-x avec une archive CVS :

```
$ cd                            # va à la zone de travail.
$ cvs co project-x              # importe les sources du CVS en local
$ cd project-x
 ... effectuez des changements au contenu ...
$ cvs diff -u                   # similaire à diff -u repository/ local/
$ cvs ci -m "Description"       # sauvegarde les sources locales sur CVS
$ vi newfile_added
$ cvs add newfile_added
$ cvs ci -m "Ajout de newfile_added"
$ cvs up                        # récupère la dernière version du CVS
$ cvs tag Release-1             # ajoute une balise de version
... modifiez un peu plus ...
$ cvs tag -d Release-1          # enlève une balise de version
$ cvs ci -m "commentaires supplémentaires"
$ cvs tag Release-1             # rajoute une balise de version
$ cd                            # retourne à la zone de travail.
$ cvs co -r Release-initial -d old project-x
... récupère la version originale dans le répertoire old
$ cd old
$ cvs tag -b Release-initial-bugfixes # balise création d'une branche (-b)
 ... Maintenant vous pouvez travailler sur l'ancienne version
$ cvs update -r Release-initial-bugfixes
 ... L'arbre des sources a maintenant la balise permanente "Release-initial-bugf
 ... Travaillez sur cette branche
$ cvs up # synchronise avec les autres développeurs de la branche
$ cvs ci -m "ajout à la branche"
$ cvs update -r HEAD # change la branche en tronc principal HEAD
$ cvs update -kk -j Release-initial-bugfixes
 ... Ajoute la branche au tronc principal sans le remplacement des mots clé
 ... fixez les conflits avec un éditeur
```

```
$ cvs ci -m "merge Release-initial-bugfixes into the main trunk"
$ cd
$ tar -cvzf old-project-x.tar.gz old     # créez une archive, -j pour bz2
$ cvs release -d old               # supprimez les sources locales (optionnel)
```

Options utiles à se rappeler (utiliser juste après `cvs`) :

```
-n      lancement sec, sans effet
-t      message des étapes de l'activité cvs
```

12.2.7 Exportation de fichiers à partir de CVS

Pour avoir la dernière version par CVS, utiliser "tomorrow" :

```
$ cvs ex -D tomorrow module_name
```

12.2.8 Administrer CVS

Ajouter un alias à un projet (serveur local) :

```
$ su - admin           # un membre de l'équipe
Password:
$ export CVSROOT=/var/lib/cvs ; cvs co CVSROOT/modules
$ cd CVSROOT
$ echo "px -a project-x" >>modules
$ cvs ci -m "Now px is an alias for project-x"
$ cvs release -d .
$ exit                 # control-D pour quitter su
$ cvs co -d project px
 ... projet-x (alias:px) de CVS au répertoire du projet
$ cd project
 ... faites les changements pour les contenus
```

12.3 Résoudre les problèmes de CVS

12.3.1 Permissions de fichiers dans le dépôt

CVS ne va pas écraser le fichier du dépôt courant mais va le remplacer par un autre. Ainsi, *la permission en écriture sur le répertoire de dépôt* est critique. Pour chaque nouvelle création d'un dépôt, lancez ce qui suit pour vous assurer que cette condition et remplie.

```
# cd /var/lib/cvs
# chown -R root:src repository
# chmod -R ug+rwX   repository
# chmod    2775     repository # si besoin est, ça et les sous-répertoires
```

12.3.2 Bit d'exécution

Le bit d'exécution de fichier est conservé lorsqu'on quitte la session. Si on observe n'importe quel problème de permission d'exécution dans les fichiers de contrôle, on peut changer la permission sur ce fichier dans le dépôt CVS avec la commande suivante.

```
# chmod ugo-x nom_du_fichier
```

12.4 Commandes CVS

```
{add|ad|new} [-k kflag] [-m 'message'] files...
{admin|adm|rcs} [rcs-options] files...
{annotate|ann} [options] [files...]
{checkout|co|get} [options] modules...
{commit|ci|com}   [-lnR]  [-m  'log_message'  |  -f  file] \
        [-r revision] [files...]
{diff|di|dif} [-kl] [rcsdiff_options] [[-r rev1 | -D date1] \
        [-r rev2 |  -D date2]] [files...]
{export|ex|exp} [-flNn] -r rev|-D date [-d dir] [-k kflag] module...
{history|hi|his} [-report] [-flags] [-options args] [files...]
{import|im|imp} [-options] repository vendortag releasetag...
{login|logon|lgn}
{log|lo|rlog} [-l] rlog-options [files...]
{rdiff|patch|pa} [-flags] [-V vn] [-r t|-D d [-r t2|-D d2]] modules...
{release|re|rel} [-d] directories...
{remove|rm|delete} [-lR] [files...]
{rtag|rt|rfreeze} [-falnR]  [-b]  [-d]  [-r  tag  |  -D  date] \
         sym_bolic_tag modules...
{status|st|stat} [-lR] [-v] [files...]
{tag|ta|freeze} [-lR] [-F] [-b] [-d] [-r tag | -D date]  [-f] \
         sym_bolic_tag [files...]
{update|up|upd} [-AdflPpR] [-d] [-r tag|-D date] files...
```

Chapitre 13

Programmation

N'utilisez pas "test" comme nom d'exécutable pour un fichier de test. `test` est une commande du shell.

13.1 Où commencer

Linux Programming Bible (John Goerzen/IDG books)

Documents et exemples sous `/usr/share/doc/<packages>`

Beaucoup de longs documents informatifs peuvent être obtenus sur papier par GNU (http://www.gnu.org/).

Les quatres sections suivantes contiennent quelques scripts d'exemple dans différents langages pour créer un fichier texte d'information sur les comptes utilisateur qui peut être ajouté à `/etc/passwd`, avec par exemple le programme `newusers`. Chaque script prend en entrée un fichier contenant des lignes de la forme `prénom nom mot_de_passe`. (Les répertoires "home" des utilisateurs ne seront pas créés avec ces scripts.)

13.2 BASH

Learning the bash Shell, 2nd edition (O'Reilly)

```
$ info bash
$ mc   /usr/share/doc/bash/examples/ /usr/share/doc/bash/
```

(Il faut installer le paquet `bash-doc` pour avoir les fichiers d'exemple.)

Court exemple de programme (crée un nouvel utilisateur) :

```
#!/bin/bash
# (C) Osmu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
pid=1000;
while read n1 n2 n3 ; do
if [ ${n1:0:1} != "#" ]; then
let pid=$pid+1
echo ${n1}_${n2}:password:${pid}:${pid}:,,,/home/${n1}_${n2}:/bin/bash
fi
done
```

13.3 AWK

Effective awk Programming, 3rd edition, et *sed & awk*, 2nd edition (O'Reilly)

```
$ man awk
```

Court exemple de programme (crée un nouvel utilisateur) :

```
#!/usr/bin/awk -f
# Script pour créer un fichier utilisable avec la commande 'newusers' à partir
# d'un fichier contenant des IDs utilisateur et des mots de passe sous la
# forme :
# Prénom Nom Mot_de_passe
# Copyright (c) KMSelf Sat Aug 25 20:47:38 PDT 2001
# Distributed under GNU GPL v 2, or at your option, any later version.
# This program is distributed WITHOUT ANY WARRANTY.

BEGIN {
    # Assign starting UID, GID
    if ( ARGC > 2 ) {
 startuid = ARGV[1]
 delete ARGV[1]
    }
    else {
        printf( "Usage:  newusers startUID file\n" \
   "...where startUID is the starting userid " \
```

```
    "to add, and file is \n" \
    "an input file in form firstname last name password\n" \
  )
  exit
     }

     infile = ARGV[1]
     printf( "Starting UID: %s\n\n", startuid )
}

/^#/ { next }

{
     ++record
     first = $1
     last = $2
     passwd = $3
     user= substr( tolower( first ), 1, 1 ) tolower( last )
     uid = startuid + record - 1
     gid = uid
     printf( "%s:%s:%d:%d:%s %s,,/home/%s:/bin/bash\n",  \
  user, passwd, uid, gid, first, last, user \
  )
}
```

13.4 PERL

Programming Perl, 3rd edition (O'Reilly)

```
$ man perl
```

Court exemple de programme (crée un nouvel utilisateur) :

```
#!/usr/bin/perl
# (C) Osmu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
$pid=1000;
while (<STDIN>) {
 if (/^#/) { next;}
        chop;
        $pid++;
```

```
        ($n1, $n2, $n3) = split / /;
        print $n1,"_",$n2,":", $n3, ":",$pid,
                  ":",$pid,",,,/home/",$n1,"_",$n2,":/bin/bash\n"
}
```

Installer le module Perl *module name* :

```
# perl -MCPAN -e 'install module name'
```

13.5 PYTHON

Learning Python (O'Reilly). C'est un interpréteur sympa.

```
$ man python
```

Court exemple de programme (crée un nouvel utilisateur) :

```
#! /usr/bin/env python
import sys, string

# (C) Osmu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
# Ported from awk script by KMSelf Sat Aug 25 20:47:38 PDT 2001
# This program is distributed WITHOUT ANY WARRANTY.

def usages():
    print \
"Usage:  ", sys.argv[0], " start_UID [filename]\n" \
"\tstartUID is the starting userid to add.\n" \
"\tfilename is input file name. If not specified, standard input.\n\n" \
"Input file format:\n"\
"\tfirstname lastname password\n"
 return 1

def parsefile(startuid):
    #
    # main filtering
    #
    uid = startuid
    while 1:
```

```
        line = infile.readline()
        if not line:
            break
        if line[0] == '#':
            continue
        (first, last, passwd) = string.split(string.lower(line))
        # above crash with wrong # of parameters :-)
        user = first[0] + last
        gid = uid
        lineout = "%s:%s:%d:%d:%s %s,,/home/%s:/bin/bash\n" %  \
            (user, passwd, uid, gid, first, last, user)
        sys.stdout.write(lineout)
        +uid

if __name__ == '__main__':
    if len(sys.argv) == 1:
        usages()
    else:
        uid = int(sys.argv[1])
        #print "# UID start from: %d\n" % uid
        if len(sys.argv) > 1:
            infilename   = string.join(sys.argv[2:])
            infile = open(infilename, 'r')
            #print "# Read file from: %s\n\n" % infilename
        else:
            infile = sys.stdin
        parsefile(uid)
```

13.6 MAKE

Managing Projects with make, 2nd edition (O'Reilly)

```
$ info make
```

Variables automatiques simples :

Syntaxe des règles :

```
Cible: [ Prérequis ... ]
< TAB > command1
```

```
< TAB > -command2 # peut générer une erreur
< TAB > @command3 # pas d'echo
```

Chaque ligne est interprétée par le shell après la substitution des variables par make. Utiliser \ à la fin d'une ligne pour continuer le script. Utiliser `$$` pour entrer `$` pour les variables d'environnement du script shell.

Equivalents des règles implicites :

```
.c:   header.h == %  : %.c header.h
.o.c: header.h == %.c: %.o header.h
```

Variables automatiques pour les règles ci-dessus :

```
foo.o: new1.c new2.c.c old1.c new3.c
$@ == foo.o                        (cible)
$< == new1.c                       (premier)
$? == new1.c new2.c new3.c         (plus récent)
$^ == new1.c new2.c.c old1.c new3.c (tout)
$* == '%' motif correspondant dans le motif cible.
```

Référence des variables :

```
foo1 := bar    # expansion unique
foo2  = bar    # expansion récursive
foo3 += bar    # ajoute
SRCS := $(wildcard *.c)
OBJS := $(foo:c=o)
OBJS := $(foo:%.c=%.o)
OBJS := $(patsubst %.c,%.o,$(foo))
DIRS  = $(dir directory/filename.ext) # Extrait "directory"
$(notdir NAMES...), $(basename NAMES...), $(suffix NAMES...) ...
```

Exécuter `make -p -f/dev/null` pour voir les règles automatiques internes.

13.7 C

Kernighan & Ritchie, *The C Programming Language*, 2nd edition (Prentice Hall). Pour obtenir la référence de la bibliothèque C GNU :

```
# apt-get install glibc6-doc manpages-dev
```

Pour lire la référence de la bibliothèque C GNU, utilisez "`info libc`". Pour chaque fonction, "printf" par exemple, utilisez "`man 3 printf`".

13.7.1 Programme simple en C (gcc)

Un exemple simple pour compiler `example.c` avec la bibliothèque `libm` dans l'exécutable `run_example` :

```
$ cat exmple.c
#include <stdio.h>
#include <math.h>
#include <string.h>

int main(int argc, char **argv, char **envp){
 double x;
 char y[11];
 x=sqrt(argc+7.5);
 strncpy(y, argv[0], 10); /* prevent buffer overflow */
 y[10] = '\0'; /* fill to make sure end with '\0' */
 printf("%5i, %5.3f, %10s, %10s\n", argc, x, y, argv[1]);
 return 0;
}

$ gcc -Wall -g -o run_example exmple.c -lm
$ ./run_example
    1, 2.915, ./run_exam,     (null)
$ ./run_example 1234567890qwerty
    2, 3.082, ./run_exam, 1234567890qwerty
```

Ici, `-lm` est nécessaire pour lier la bibliothèque `libm`, pour utiliser sqrt(). La bibliothèque est en fait dans `/lib` sous le nom `libm.so.6` qui est un lien symbolique vers `libm-2.1.3.so`.

Regardez le dernier paramètre dans le texte imprimé. Il y a plus de 10 caractères imprimés alors que `%10s` est spécifié.

L'utilisation de fonctions effectuant des opérations sur des pointeurs sans vérifier la limite, comme `sprintf` et `strcpy`, est dépréciée pour prévenir les exploits de dépassement de buffer, annulant les effets ci-dessus. Utilisez `snprintf` et `strncpy` à la place.

13.7.2 Déboguer (gdb)

Didacticiel GDB (http://www.dirac.org/linux/gdb/) (NdT : en Anglais)

Utilisez `gdb` pour déboguer des programmes compilés avec l'option `-g`. Beaucoup de commandes peuvent être abbréviées. L'expansion par tab marche comme avec le shell.

```
$ info gdb
...
$ gdb program
(gdb) b 1                # positionne un point d'arrêt à la ligne 1
(gdb) run arg1 arg2 arg3 # exécute le programme
(gdb) next               # ligne suivante
...
(gdb) step               # pas en avant
...
(gdb) p parm             # affiche parm
...
(gdb) p parm=12          # met la valeur de parm à 12
```

Les commandes suivantes sont aussi utiles :

- `ldd` : affiche les dépendances des bibliothèques partagées
- `strace` : trace les appels et signaux système
- `ltrace` : trace les appels aux bibliothèques

Pour déboguer depuis emacs, référez-vous à 'Résumé des commandes de l'éditeur (Emacs, Vim)' page 131.

13.7.3 Flex – un meilleur Lex

Vous devez fournir vos propres `main()` et `yywrap()`, ou votre `program.l`. Il ne devrait pas y avoir besoin de bibliothèque pour compiler. (`yywrap` est une macro ; `%option main` active `%option noyywrap` implicitement) :

```
%option main
%%
.|\n ECHO ;
%%
```

Autrement, vous pouvez compiler avec l'option du linker -`lfl` à la fin de la ligne de commande de `cc`, comme ATT-Lex avec -`ll` (pas de %option nécessaire).

13.7.4 Bison – un meilleur Yacc

Vous devez fournir vos propres `main()` et `yyerror()`.`main()` appelle `yyparse()` qui appelle `yylex()`, habituellement créé avec Flex.

```
%%

%%
```

13.7.5 Autoconf – uninstall

SI vous avez toujours les sources et SI elles utilisent autoconf/automake et SI vous vous souvenez comment vous avez configuré la compilation :

```
$ ./configure les-options-de-configuration
# make uninstall
```

13.8 SGML

Un moyen de gérer plusieurs formats pour un document est SGML. Un SGML plus facile est offert avec debiandoc, qui est utilisé ici. Cela requiert des conversions mineurs dans le fichier texte original, pour les caractères suivants :

```
<   &lt;
>   &gt;
&   &amp;
©   &copy;
-   &ndash;
--   &mdash;
```

Pour marquer une section en tant que commentaire non imprimable, faites :

```
<!-- State issue here ... -->
```

Pour marquer une section en tant que commentaire ou pas à l'impression, faites :

```
<![ %FIXME [ State issue here ... ]]>
```

En SGML, la *première définition* d'une entité l'emporte. Par exemple :

```
<!entity % qref "INCLUDE">
<![ %qref [ <!entity param "Data 1"> ]]>
<!entity param "Data 2">
&param;
```

Cela termine en tant que "Data 1". Si la première ligne inclut "IGNORE" au lieu de "INCLUDE", cela termine en tant que "Data 2" (la deuxième ligne est conditionnelle).

Pour plus de détails, faites `apt-get install debiandoc-sgml-doc`. Lisez aussi *DocBook : The Definitive Guide*, by Walsh and Muellner (O'Reilly).

13.9 Paquetage

Lisez la documentation fournie dans le paquet `packaging-manual` (Potato) ou `debian-policy` (Woody).

Utilisez `dh_make` du paquet `dh-help` pour créer l'architecture du paquet. Ensuite, suivez les instructions de `dh-make(1)`, qui utilise `debhelper` dans `debian/rules`.

Une autre approche est d'utiliser `deb-make` du paquet `debmake`. Cela n'utilise aucun script `debhelper` et dépend seulement du shell.

Pour plusieurs paquets source, voir "mc" (`dpkg-source -x mc_4.5.54.dsc`) qui utilise "sys-build.mk" par Adam Heath (`<doogie@debian.org>`) et "glibc" (`dpkg-source -x glibc_2.2.4-1.dsc`) qui utilise un autre système par Joel Klecker (`<espy@debian.org>`).

Chapitre 14

GnuPG

Consultez `/usr/share/doc/gnupg/README.gz` pour des informations détaillées ou lisez `man gpg`.

14.1 Installer GnuPG

Lisez le manuel de GNU privacy (dans woody, `gnupg-doc`).

```
# gpg --gen-key                    # génère une nouvelle clé
# gpg --gen-revoke my_user_ID      # génère une clé de révocation pour
                                   # my_user_ID
# host -l pgp.net | grep www|less # cherche les serveurs de clés pgp
```

Aujourd'hui, les bons serveurs de clés sont :

```
keyserver wwwkeys.eu.pgp.net
keyserver wwwkeys.pgp.net
```

Il faut faire attention de ne pas créer plus de deux sous-clés. Si vous le faites, les serveurs de clés de pgp.net vont corrompre votre clé. De plus, un seul serveur de clés peut être spécifié dans votre `$HOME/.gnupg/options`

Malheureusement, les serveurs suivants ne fonctionnent plus :

```
keyserver search.keyserver.net
keyserver pgp.ai.mit.edu
```

14.2 Utiliser GnuPG

Gestion de fichiers :

```
$ gpg  [options]  command  [args]
$ gpg {--armor|-a} {--sign|-s} file # signe file dans le fichier file.asc
$ gpg --clearsign file              # signe file sans le chiffrer
$ gpg --clearsign --not-dash-escaped patchfile  # signe sans le chiffrer patchfi
$ gpg --verify file         # vérifie un fichier signé sans chiffrement
$ gpg -o file.sig {-b|--detach-sign} file # crée une signature détachée
$ gpg --verify file.sig file        # vérifie file à l'aide de file.sig
$ gpg -o crypt_file {--recipient|-r} name {--encrypt|-e} file
       # chiffrement à clé publique pour le destinataire name
$ gpg -o crypt_file {--symmetric|-c} file # chiffrement symétrique
$ gpg -o file --decrypt crypt_file        # déchiffrement
```

14.3 Gérer GnuPG

Gestion des clés :

```
$ gpg --edit-key user_ID               # "help" pour l'aide, intéractive
$ gpg -o file --exports                # exporte toutes les clés vers le fichier
$ gpg --imports file                   # importe toutes les clés depuis un fichi
$ gpg --send-keys user_ID              # envoie la clé de user_ID vers un serveu
$ gpg --recv-keys user_ID              # reçoit la clé de user_ID depuis le serv
$ gpg --list-keys user_ID              # liste les clés de user_ID
$ gpg --list-sigs user_ID              # liste les  signatures de user_ID
$ gpg --check-sigs user_ID             # vérifie la signature de user_ID
$ gpg --fingerprint user_ID            # vérifie l'empreinte de user_ID
$ gpg --list-sigs | grep '^sig' | grep '[User id not found]' \
  | awk '{print $2}' | sort -u | xargs gpg --recv-keys # get unknownkeys
  # met à jour les clés pour toutes les signatures inconnues
```

Code de confiance :

```
-         Pas de confiance assignée au possesseur / pas encore calculée
e         Calcul de la confiance échoué
q         Pas assez d'informations pour le calcul
n         Ne jamais faire confiance à cette clé
```

```
m         Confiance marginale
f         Confiance complète
u         Confiance ultime
```

14.4 Utilisation avec Mutt

Ajoutez les instructions suivantes à `~/.muttrc` pour éviter que GnuPG, qui est lent, s'exécute automatiquement mais soit quand même utile.

```
macro index S ":toggle pgp_verify_sig\n"
set pgp_verify_sig=no
```

Chapitre 15

Support Debian

Les références suivantes fournissent de l'aide et des conseils pour Debian. Utilisez au mieux ces documents pour vous aider vous-même avant de venir crier sur les listes de diffusion :)

Notez que vous avez accès à beaucoup de documentation sur votre système en utilisant un navigateur web, en utilisant les commandes `dwww` ou `dhelp`, que vous trouverez dans les paquets du même nom.

15.1 Références

Les références suivantes sont disponibles pour Debian et Linux en général. Si leur contenu est en conflit avec l'un l'autre, comptez toujours davantage sur les sources primaires d'informations que sur les sources secondaires telles que ce document.

- Manuel d'installation (primaire)
  - A lire avant d'installer et de mettre à jour.
  - Web : http://www.debian.org/releases/stable/installmanual
  - Web : http://www.debian.org/releases/testing/installmanual (en cours d'écriture)
  - Paquet : `install-doc`
  - Fichier : `/usr/share/doc/install-doc/index.html`
- Les notes de version (primaire)
  - A lire absolument avant l'installation et la mise à jour même si avez de bonnes connaissances.
  - Web : http://www.debian.org/releases/stable/releasenotes
  - Web : http://www.debian.org/releases/testing/releasenotes (work in progress)
  - Paquet : `install-doc`

- FAQ (secondaire)
  - Foire aux questions (un peu ancien)
  - Web : http://www.debian.org/doc/manuals/debian-faq/
  - Paquet : doc-debian
  - Fichier : /usr/share/doc/debian/FAQ/index.html
- APT HOWTO (secondaire)
  - Guide détaillé pour l'utilisateur de la gestion de paquets Debian (woody)
  - Web : http://www.debian.org/doc/manuals/apt-howto/
  - Paquet : apt-howto,
  - Fichier : /usr/share/doc/apt-howto
- Documentation dselect pour débutants (secondaire)
  - Tutoriel pour dselect
  - Web : http://www.debian.org/releases/woody/i386/dselect-beginner
  - Paquet : install-doc,
- La charte Debian (primaire)
  - Charte Technique de Debian.
  - Web : http://www.debian.org/doc/debian-policy/
- Référence du développeur Debian (primaire)
  - Savoir de base pour les développeurs.
  - Nous devrions parcourir ça au moins une fois.
  - Web : http://www.debian.org/doc/manuals/developers-reference/
- Guide des nouveaux responsables Debian (primaire)
  - Guide pratique pour les développeurs.
  - Tutoriel d'empaquetage.
  - http://www.debian.org/doc/manuals/maint-guide/
- Manuel d'empaquetage (potato)
  - packaging-manual dans potato.
- Pages de manuel Unix (primaire)
  - man *package-name*
- Pages info GNU (primaire)
  - info *package-name*
- Documents spécifiques à chaque paquet(primaire)
  - Retrouvez-les sous /usr/share/doc/*package-name*
- LDP : Linux Documentation Project (secondaire)
  - HOWTOs et mini-HOWTOS généraux pour Linux
  - Web : http://www.tldp.org/
  - Paquet : doc-linux
  - Fichier : /usr/share/doc/HOWTO/
- Projet de Documentation Debian (secondaire)
  - Manuels spécifiques pour Debian
  - Web : http://www.debian.org/doc/
- Le coin du développeur (secondaire)

    - Informations clef pour les responsables de paquets Debian
    - permet à l'utilisateur final de mieux comprendre la Debian
    - Web : http://www.debian.org/devel/
- Code source (absolument primaire)
    - Personne ne peut donner d'arguments contre ça :-)
    - Téléchargez le code source en suivant les instructions de 'Le code source' page 11

15.2 Trouver le sens d'un mot

Beaucoup de mots utilisés dans Debian sont des mots issus du jargons techniques ou des acronymes. L'instruction suivante répondra à la plupart des questions de vocabulaire :

```
$ dict <mettez un mot bizarre ici>
```

15.3 Système de suivi des bogues Debian

La distribution Debian a un système de suivi de bogues ou bug tracking system (BTS) (http://bugs.debian.org/) dans lequel nous enregistrons les détails des bogues rapportés par les utilisateurs et les développeurs. Chaque bogue reçoit un numéro, et est répertorié dans un fichier jusqu'à ce qu'il soit marqué comme ayant été traité.

Vous devriez vérifier que votre bogue n'a pas déjà été signalé par quelqu'un d'autre avant de le soumettre. Les listes des bogues non encore résolus sont disponibles sur la Toile (http://bugs.debian.org/) et ailleurs (http://www.debian.org/Bugs/Access). Voyez aussi 'Rechercher les bogues de Debian et demander de l'aide' page 62.

La méthode pour rapporter un bogue est décrite à l'adresse http://www.debian.org/Bugs/Reporting

15.4 Listes de diffusion

Lisez au moins debian-devel-announce (Anglais, lecture seule & trafic bas) pour être au courant des nouvelles sur Debian.

Les listes de diffusions les plus utiles aux utilisateurs de Debian sont debian-user (Anglais, ouverte & trafic important) et autres debian-user-*langue* (pour les autres langues).

Pour plus d'informations sur ces listes et la manière de s'y inscrire voyez http://lists.debian.org/. Veuillez rechercher dans les archives si vos questions n'ont pas déja trouvé de réponses avant de poster, et respecter le code de conduite des listes.

15.5 Internet Relay Chat

Debian a un canal IRC dédié au support et à l'assistance des utilisateurs Debian qui fait partie du réseau IRC Open Projects, un réseau consacré à la mise à disposition de ressources pour les échanges d'information dans la communauté du Logiciel Libre. Pour accéder au canal avec votre client IRC favori, vous devez lui indiquer de se connecter à irc.openproject.net et rejoindre le canal `#debian`.

Veuillez suivre les instructions du canal et respecter les autres utilisateurs. Pour plus d'informations sur Open Project visitez le site internet (`http://www.openprojects.net`).

15.6 Moteurs de recherche

De nombreux moteurs de recherche offrent des documentations liées à Debian :

- page de recherche du site internet de Debian (`http://search.debian.org/`).
- Google (`http://www.google.com/`) : inclure "site :debian.org" comme terme de recherche.
- Google Groups (`http://groups.google.com/`) : un moteur de recherche pour groupes de nouvelles. Inclure "linux.debian.user" comme terme de recherche.
- AltaVista (`http://www.altavista.com/`)

Par exemple, chercher le terme "cgi-perl" donne une description de ce paquet plus détaillée que la texte bref dans le champs description dans son fichier de contrôle. Voir 'Rechercher les bogues de Debian et demander de l'aide' page 62 pour de détails à ce sujet.

15.7 Sites internet

Il y a quelques sites internet populaires qui contiennent des informations générales de support.

- Debian planet (`http://www.debianplanet.org/`)
- Linux.com (`http://linux.com/`)
- Les pages de Matt Chapman (Unix Guide) (`http://www.belgarath.demon.co.uk/`)

Les sites suivants sont des sites que j'ai référencé pour des sujets spécifiques.

- Paquets d'Adrian Bunk pour faire tourner un noyau 2.4.x sur potato (`http://www.fs.tum.de/~bunk/kernel-24.html`)
- Linux sur portables (`http://www.linux-laptop.net/`)
- Xterm FAQ (`http://dickey.his.com/xterm/xterm.faq.html`)
- EXT3 File System mini-HOWTO (`http://www.symonds.net/~rajesh/howto/ext3/index.html`)
- Support de Grands Fichiers par Linux (`http://www.suse.de/~aj/linux_lfs.html`)

- LNX-BBC (Projet de boot CD au format Business-card) (http://www.lnx-bbc.org/)
- Linux info par Karsten Self (Partitioning, backup, browsers...) (http://kmself.home.netcom.com/Linux/)
- Backup info HOWTO par Alvin Oga (http://www.Linux-Backup.net/)
- Security info HOWTO par Alvin Oga (http://www.Linux-Sec.net/)
- Various UNOFFICIAL sources for APT (http://www.internatif.org/bortzmeyer/debian/apt-sources/)
- Configuration Ethernet d'un portable (http://www.orthogony.com/gjw/lap/lap-ether-intro.html)

Annexe A

Annexe

A.1 Auteurs

Guide de référence pour Debian a été démarré par Osamu Aoki <debian@aokiconsulting.com> en tant que mémo d'installation personnel et finalement appelé « Quick Reference . . . ». Une grande partie du contenu venait des archives de la liste de diffusion « debian-user ». « Debian – Manuel d'installation » et « Debian – Notes de version » ont aussi été utilisés.

Suite à une suggestion de Josip Rodin, qui est très actif dans le Projet de Documentation Debian (http://www.debian.org/doc/ddp) (DDP) et est le responsable actuel de la « FAQ Debian », ce document a été renommé en « Guide de référence pour Debian » et fusionné avec des chapitres de la « FAQ Debian » ayant un contenu de référence.

Ce document a été modifié, traduit et amélioré par les membres de l'équipe QREF suivants :

- Relecture et réécriture en Anglais du « Quick Reference. . . » original
  - Osamu Aoki <debian@aokiconsulting.com\textgreater{}> (leader : tout le contenu)
- Relecture et réécriture en Anglais
  - David Sewell <dsewell@virginia.edu> (leader : style en)
  - Brian Nelson <nelson@bignachos.com>
  - Daniel Webb <webb@robust.colorado.edu>
- Traduction en Français
  - Guillaume Erbs <gerbs@free.fr> (leader : fr)
  - Rénald Casagraude <rcasagraude@interfaces.fr>
  - Jean-Pierre Delange <delange@imaginet.fr>
  - Daniel Desages <daniel@desages.com>
- Traduction en Italien
  - Davide Di Lazzaro <mc0315@mclink.it> (leader : it)
- Traduction en Espagnol

  - Walter Echarri <wecharri@infovia.com.ar>
  - José Carreiro <ffx@urbanet.ch>

QREF est un raccourci pour le document original « Quick Reference... » et aussi le nom du projet sur qref.sourceforge.net.

La plus grande partie du contenu de 'Notions fondamentales sur Debian' page 5 vient de la « FAQ Debian » (mars 2002)

- 5. Les archives FTP Debian `ftparchives.sgml` (chapitre entier)
- 6. Bases du système de gestion des paquets Debian `pkg_basics.sgml` (chapitre entier)
- 7. Outils de gestion des paquets Debian `pkgtools.sgml` (chapitre entier)
- 8. Maintenir votre système à jour `uptodate.sgml` (chapitre entier)
- 9. Debian et le noyau `kernel.sgml` (chapitre entier)
- 10. Personnaliser votre installation de Debian GNU/Linux `customizing.sgml` (part of chapter)

Ces sections de la « FAQ Debian » ont été incluses dans ce document après des réorganisations majeures reflétant des changements récents dans le système Debian. Le contenu de ce document est plus récent.

La « FAQ Debian » originale a été écrite et maintenue par J.H.M. Dassen (Ray) et Chuck Stickelman. Les auteurs de la « FAQ Debian » réécrite sont Susan G. Kleinmann et Sven Rudolph. Après eux, la « FAQ Debian » a été maintenue par Santiago Vila. Le responsable actuel est Josip Rodin.

Une partie de l'information utilisée pour la « FAQ Debian » venait de

- L'annonce de sortie de Debian-1.1, par Bruce Perens (http://www.perens.com/).
- La FAQ Linux, par Ian Jackson (http://www.chiark.greenend.org.uk/~ijackson/).
- Archives de la liste de diffusion Debian (http://lists.debian.org/),
- Le manuel des développeurs dpkg et la Charte Debian (voir 'Références' page 157)
- beaucoup de développeurs, volontaires, beta testeurs, et
- la mémoire voilée de ses auteurs. :-)

Les auteurs souhaitent remercier tous ceux qui ont aidé à la création de ce document.

A.2 Garanties

Comme je ne suis pas un expert, je ne prétends pas tout connaître sur Debian ou Linux en général. Les considérations de sécurité que j'utilise peuvent être utilisées seulement dans le cas d'une utilisation personnelle.

Ce document ne remplace aucun guide officiel.

Toute garantie est désavouée. Toutes les marques de commerce sont la propriété de leurs propriétaires respectifs.

A.3 Retour

Les commentaires et aditions à ce document sont toujours les bienvenus. Merci d'envoyer un courriel à Osamu Aoki (http://www.aokiconsulting.com/) <debian@aokiconsulting.com> en Anglais ou à chacun des traducteurs dans la langue respective.

Bien que je vive aux Etats-Unis, l'Anglais n'est pas ma langue maternelle. Toute correction grammaticale est la bienvenue.

Le meilleur retour est un diff du SGML, mais un diff de la version texte est bienvenu. Voir 'Document officiel' page 1 pour le site du document officiel.

Les fichiers SGML originaux utilisés pour créer ce document sont aussi disponibles par CVS à :pserver :anonymous@cvs.qref.sf.net/cvsroot/qref ou http://qref.sourceforge.net/Debian/qref.tar.gz.

A.4 Format du document

Ce document a été écrit en utilisant la DTD SGML DebianDoc (réécrite à partir de LinuxDoc SGML). Le système SGML DebianDoc permet d'obtenir des fichiers dans de nombreux formats à partir d'une seule source, par exemple ce document peut être visualisé en HTML, texte, TeX DVI, PostScript, PDF, ou GNU info.

Les utilitaires de conversion pour DebianDoc SGML sont disponibles dans le paquet Debian debiandoc-sgml.

A.5 Labyrinthe de Debian

Le système Linux est une plateforme informatique très performante lorsqu'elle est utilisée en réseau. Cependant, connaître ses possibilités et l'utiliser n'est pas si facile. Configurer une imprimante en est un bon exemple.

Il existe une carte très détaillée appelée « CODE SOURCE ». Elle est très précise mais très dure à comprendre. Il existe aussi des références appelées HOWTO et mini-HOWTO. Elles sont plus faciles à comprendre mais ont tendance à trop se concentrer sur des détails et perdent de vue les aspects généraux. J'ai de temps en temps des problèmes à trouver la bonne section dans un long HOWTO quand j'ai besoin de quelques commandes à exécuter.

Afin de m'y retrouver dans ce labyrinthe d'information sur la configuration d'un système Linux, je m'étais fait des mémos dans un fichier texte pour une référence rapide. Ces mémos ont grossi et dans le même temps j'avais appris debiandoc. Le résultat est ce *Guide de référence pour Debian*.

A.6 Les citations Debian

Voici quelques citations intéressantes venant de la liste de diffusion Debian.

- "This is Unix. It gives you enough rope to hang yourself." — Miquel van Smoorenburg <miquels@cistron.nl> (C'est Unix. Il vous donne assez de corde pour vous pendre vous-même.)
- "Unix **IS** user friendly. . .It's just selective about who its friends are." — Tollef Fog Heen <tollef@add.no> (Unix **EST** l'ami de l'utilisateur. . .Il choisit juste qui sont ses amis.)