

Cahiers
de **l'Admin**

Collection dirigée par
Nat Makarévitch

Debian

GNU/Linux

Squeeze

Raphaël Hertzog

Roland Mas

Contenu du DVD-rom
à télécharger



EYROLLES

Nous vous remercions pour l'achat de ce livre électronique.

La version papier de cet ouvrage étant accompagnée d'un support physique, nous vous proposons de télécharger les fichiers depuis le site ci-dessous, de manière à ce que vous puissiez pleinement profiter de votre achat.

Raphaël Hertzog, *Debian Squeeze* - ISBN : 978-2-212-13248-9

Le livre papier est accompagné d'un DVD contenant la distribution GNU/Linux Debian Squeeze 6.0.2.

Vous pouvez télécharger la dernière version de Debian Squeeze à l'adresse : <http://www.debian.org/releases/stable/>

Pour toute remarque ou suggestion, merci d'écrire à numerique@eyrolles.com

Cahiers
de l'Admin
Debian
Squeeze

Chez le même éditeur

Ouvrages sur Linux et autres systèmes et logiciels libres

Développement système sous Linux. *Ordonnancement multi-tâche, gestion mémoire, communications, programmation réseau.*

C. BLAESS. – G12881, 1004 pages.

Linux Administration. J.-F. BOUCHAUDY, G. GOUBET. – N°12882, 2011, 504 pages.

Linux embarqué. P. FICHEUX. – N°12452, 2010, 378 pages.

BSD, 2e édition (coll. *Cahiers de l'Admin*). E. DREYFUS. – N°11463, 2004, 300 pages.

Sécuriser un réseau Linux. B. BOUTHERIN, B. DELAUNAY. – N°11960, 3e édition, 2007, 250 pages.

Mémento UNIX/Linux. I. HURBAIN, avec la contribution d'E. DREYFUS. – N°11954, 2006, 14 pages.

Debian. *Administration et configuration avancées.* M. KRAFFT, adapté par R. HERTZOG et R. MAS. – N°11904, 2006, 674 pages.

Ouvrages sur la sécurité et l'administration

Nagios 3. *Pour la supervision et la métrologie Déploiement, configuration et optimisation.* J. GABÈS, N. MAKARÉVITCH. – N°12473, 2009, 482 pages.

Le livre de Packet Filter. PETER N. M. HANSTEEN. – N°12516, 2009, 190 pages.

Management de la sécurité de l'information. *Implémentation ISO 27001– Mise en place d'un SMSI et audit de certification*

A. FERNANDEZ-TORO. – N°12622, 2009, 284 pages.

Management de la continuité d'activité. E. BESLUAU. – N°12820, 2e édition 2010, 298 pages.

ITIL pour un service informatique optimal. C. DUMONT. – N°12102, 2e édition, 2007, 378 pages.

Sécurité informatique. L. BLOCH, C. WOLFHUGEL. – N°13233, 3e édition, 2011, 324 pages.

Tableaux de bord de la sécurité réseaux. C. LLORENS, L. LEVIER, D. VALOIS. – N°12821, 3e édition, 2010, 560 pages.

Admin'sys. *Gérer son temps...* T. LIMONCELLI, adapté par S. BLONDEEL. – N°11957, 2006, 274 pages.

SSL VPN. *Accès web et extranets sécurisés.* J. STEINBERG, T. SPEED, adapté par B. SONNTAG. – N°11933, 2006, 220 pages.

Sécurité PHP 5 et MySQL. D. SÉGUY, P. GAMACHE. – N°12554, 2e édition 2009, 268 pages.

Mémento VMware Server. *Virtualisation de serveurs.* F. MANZANO. N°12320, 2008, 14 pages.

Mémento Cisco. *IOS – Configuration générale.* R. BERGOIN, C. BOURG. N°12347, 2008, 14 pages.

Collection « Accès Libre »

Pour que l'informatique soit un outil, pas un ennemi !

Bien rédiger pour le Web.

I. CANIVET.

N°12883, 2e édition 2011, 552 pages.

Améliorer ses taux de conversion web. *Vers la performance des sites web au-delà du webmarketing.*

S. ROUKINE.

N°12858, 2e édition 2011, 260 pages.

Préserver son patrimoine numérique. *Classer, archiver et sauvegarder ses e-mails, photos et vidéo, contacts, documents administratifs - Guide à l'usage des particuliers et des entrepreneurs individuels.*

C. HUC.

N°12789, 2010, 344 pages.

WordPress 3 pour le blogueur efficace. *Installation, personnalisation, administration — Référencement et marketing — Monétisation — Usage nomade (iPhone/iPad, Android...).*

F.-X. et L. BOIS.

N°12829, 2010, 360 pages.

Concevoir et déployer ses sites web avec Drupal.

Y. BRAULT.

N°12780, 2010, 420 pages.

Se protéger sur Internet. *Conseils pour la vie en ligne.*

X. TANNIER.

N°12774, 2010, 232 pages.

Économie du logiciel libre.

F. ÉLIE.

N°12463, 2009, 195 pages.

Spip 2. *Premiers pas pour créer son site avec Spip 2.0.3.*

A.-L. QUATRAVAUX, D. QUATRAVAUX.

N°12502, 2009, 300 pages.

Réussir son site web avec XHTML et CSS.

M. NEBRA.

N°12489, 3e édition, 2010, 318 pages.

Réussir son site e-commerce avec osCommerce.

D. MERCER.

N°11932, 2007, 446 pages.

Open ERP. *Pour une gestion d'entreprise efficace et intégrée.*

F. PINCKAERS, G. GARDINER.

N°12261, 2008, 276 pages.

PGP/GPG. *Assurer la confidentialité de ses mails et fichiers.*

M. LUCAS, ad. par D. GARANCE, contrib. J.-M. THOMAS.

N°12001, 2006, 248 pages.

Monter son serveur de mails Postfix sous Linux.

M. BÄCK *et al.*, adapté par P. TONNERRE.

N°11931, 2006, 360 pages.

Ergonomie web.

A. BOUCHER.

N°12479, 2e édition 2009, 440 pages.

Joomla et VirtueMart. *Réussir sa boutique en ligne.*

V. ISAKSEN, avec la contribution de T. TARDIF.

N°12487, 2e édition 2009, 336 pages.

Raphaël Hertzog

Roland Mas

Cahiers
de
l'Admin

Debian
Squeeze

Collection dirigée par Nat Makarévitch

EYROLLES



ÉDITIONS EYROLLES
61, bd Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

Remerciements à Thierry Stempfél pour les illustrations.

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans autorisation de l'éditeur ou du Centre Français d'Exploitation du Droit de Copie, 20, rue des Grands-Augustins, 75006 Paris.

© Groupe Eyrolles, 2011, ISBN : 978-2-212-13248-9

Préface

Les professionnels adoptent de plus en plus souvent Debian GNU/Linux, dont le souci de créer une distribution riche, souple et requérant peu d'attention correspond bien à leurs attentes. Ils apprécient le soin apporté à la robustesse et la fiabilité, à l'automatisation des tâches subalternes ainsi qu'à la mise au point et au respect de spécifications garantes de la cohérence, donc de la pérennité des réalisations et des savoirs.

Dans le même temps, de grands acteurs de l'informatique perçoivent aujourd'hui l'intérêt stratégique d'une distribution Linux mûre et non gérée par une entité commerciale. Certains de leurs clients comprennent, dans le même registre, qu'une plate-forme logicielle ne dépendant pas d'accords tissés entre des fournisseurs réduit les contraintes pesant sur eux après l'achat.

De nombreux amateurs, enfin, découvrent Debian par les projets Knoppix et Ubuntu tandis que d'autres, souhaitant fuir l'empirisme, « ouvrent le capot ».

Debian, longtemps discrète, convainquit tout d'abord le passionné, souvent attiré par l'esprit qui l'anime. Il y trouve un projet aux objectifs clairs et aux réalisations transparentes, au sein duquel tous œuvrent afin de bien concevoir *avant* de construire — renonçant d'emblée aux échéances qui menacent la qualité de tant d'autres logiciels. Il y trouve un projet dirigé par ses acteurs. Il y adopte ou rejoint, en somme, un projet bénéficiant pleinement des avantages du logiciel libre... afin d'en produire lui-même.

Ce *Cabier de l'Admin* guidera et éclairera le lecteur afin de le rendre autonome. Seuls pouvaient le rédiger des auteurs maîtrisant les aspects techniques tout autant que les caractéristiques propres du projet Debian, et connaissant parfaitement les besoins des francophones, professionnels

aguerris comme amateurs éclairés. Raphaël Hertzog et Christophe Le Bars puis Roland Mas disposaient des qualités requises et surent créer puis mettre à jour cet ouvrage. Je les en remercie vivement et suis certain que sa lecture vous sera utile et agréable.

Nat Makarévitch

(empreinte PGP/GPG : 2010 4A02 9C0E 7D1F 5631 ADF0 453C 4549
0230 D602)

Table des matières

Avant-propos	XV	Intégration avec des machines Mac OS • 36
1. Le projet Debian	1	Intégration avec d'autres machines Linux/Unix • 36
Qu'est-ce que Debian ? • 2		Démarche de migration • 36
Un système d'exploitation multi-plates-formes • 2		Recenser et identifier les services • 37
La qualité des logiciels libres • 3		Réseau et processus • 37
Le cadre : une association • 4		Conserver la configuration • 38
Les textes fondateurs • 4		Prendre en main un serveur Debian existant • 39
L'engagement vis-à-vis des utilisateurs • 5		Installer Debian • 40
Les principes du logiciel libre selon Debian • 6		Installer et configurer les services sélectionnés • 41
Fonctionnement du projet Debian • 8		4. Installation
Les développeurs Debian • 8		Méthodes d'installation • 44
Le rôle actif des utilisateurs • 11		Installation depuis un CD-Rom/DVD-Rom • 44
Équipes et sous-projets • 13		Démarrage depuis une clé USB • 45
Sous-projets Debian existants • 14		Installation par boot réseau • 46
Équipes administratives • 15		Autres méthodes d'installation • 46
Équipes de développement, équipes transversales • 16		Étapes du programme d'installation • 47
Rôle d'une distribution • 18		Exécution du programme d'installation • 47
L'installateur : debian-installer • 18		Choix de la langue • 48
La bibliothèque de logiciels • 19		Choix du pays • 49
Cycle de vie d'une release • 19		Choix de la disposition du clavier • 49
Le statut Experimental • 19		Détection du matériel • 50
Le statut Unstable • 20		Chargement des composants • 50
La migration vers Testing • 21		Détection du matériel réseau • 50
La promotion de Testing en Stable • 22		Configuration du réseau • 51
2. Présentation de l'étude de cas	27	Configuration de l'horloge • 51
Des besoins informatiques en forte hausse • 28		Mot de passe administrateur • 51
Plan directeur • 28		Création du premier utilisateur • 52
Pourquoi une distribution GNU/Linux ? • 29		Détection des disques et autres périphériques • 52
Pourquoi la distribution Debian ? • 30		Démarrage de l'outil de partitionnement • 53
Distributions communautaires et commerciales • 31		Partitionnement assisté • 54
Pourquoi Debian Squeeze ? • 32		Partitionnement manuel • 55
3. Prise en compte de l'existant et migration	35	Emploi du RAID logiciel • 57
Coexistence en environnement hétérogène • 36		Emploi de LVM (Logical Volume Manager) • 57
Intégration avec des machines Windows • 36		Chiffrement de partitions • 58
		Installation du système de base Debian • 59
		Configuration de l'outil de gestion des paquets (apt) • 60

- Concours de popularité des paquets • 61
- Sélection des paquets à installer • 61
- Installation du chargeur d'amorçage GRUB • 62
- Terminer l'installation et redémarrer • 62
- Après le premier démarrage • 62**
 - Installation de logiciels supplémentaires • 63
 - Mise à jour du système • 64
- 5. Système de paquetage, outils et principes fondamentaux 67**
 - Structure d'un paquet binaire • 68**
 - Méta-informations d'un paquet • 70**
 - Description : fichier control • 70
 - Dépendances : champ Depends • 71
 - Conflits : champ Conflicts • 72
 - Incompatibilités : champ Breaks • 73
 - Éléments fournis : champ Provides • 73
 - La fourniture d'un « service » • 73
 - L'interchangeabilité avec un autre paquet • 74
 - Limitations actuelles • 74
 - Remplacements : champ Replaces • 75
 - Scripts de configuration • 75
 - Installation et mise à jour • 75
 - Suppression de paquet • 76
 - Sommes de contrôle, liste des fichiers de configuration • 77
 - Structure d'un paquet source • 78**
 - Format • 78
 - Utilité chez Debian • 80
 - Manipuler des paquets avec dpkg • 81**
 - Installation de paquets • 81
 - Suppression de paquet • 83
 - Autres fonctionnalités de dpkg • 84
 - Journal de dpkg • 87
 - Cohabitation avec d'autres systèmes de paquetages • 88**
- 6. Maintenance et mise à jour : les outils APT 91**
 - Renseigner le fichier sources.list • 92**
 - Les autres dépôts officiels disponibles • 94
 - Les mises à jour pour Stable • 94
 - Les rétroportages de backports.debian.org • 94
 - Le dépôt Experimental • 95
 - Ressources non officielles : apt-get.org et mentors.debian.net • 95
 - Commandes aptitude et apt-get • 96**
 - Initialisation • 97
 - Installation et suppression • 97
 - Mise à jour • 98
 - Options de configuration • 100
 - Gérer les priorités associées aux paquets • 100
 - Travailler avec plusieurs distributions • 103
 - Commande apt-cache • 104**
 - Frontaux : aptitude, synaptic • 105**
 - aptitude • 105
 - Gestion des recommandations, suggestions et tâches • 105
 - Suivi des paquets installés automatiquement • 107
 - Meilleurs algorithmes de résolution • 109
 - Synaptic • 109
 - Vérification d'authenticité des paquets • 110**
 - Mise à jour d'une distribution à la suivante • 112**
 - Démarche à suivre • 112
 - Gérer les problèmes consécutifs à une mise à jour • 113
 - Maintenir un système à jour • 114**
 - Mise à jour automatique • 116**
 - Configuration de dpkg • 117
 - Configuration d'APT • 117
 - Configuration de debconf • 117
 - Gestion des interactions en ligne de commande • 117
 - La combinaison miracle • 118
 - Recherche de paquets • 118**
- 7. Résolution de problèmes et sources d'information ... 123**
 - Les sources de documentation • 124**
 - Les pages de manuel • 124
 - Documentation au format info • 126
 - La documentation spécifique • 126
 - Les sites web • 127
 - Les tutoriels (HOWTO) • 128
 - Procédures types • 128**
 - Configuration d'un logiciel • 128
 - Surveiller l'activité des démons • 129
 - Demander de l'aide sur une liste de diffusion • 130
 - Signaler un bogue en cas de problème incompréhensible • 130
- 8. Configuration de base : réseau, comptes, impression... 133**
 - Francisation du système • 134**
 - Définir la langue par défaut • 134
 - Configurer le clavier • 135

- Migration vers UTF-8 • 136
- Configuration du réseau • 137**
 - Interface Ethernet • 138
 - Connexion PPP par modem téléphonique • 139
 - Connexion par modem ADSL • 139
 - Modem fonctionnant avec PPPOE • 140
 - Modem fonctionnant avec PPTP • 140
 - Modem fonctionnant avec DHCP • 140
 - Configuration réseau itinérante • 141
- Attribution et résolution des noms • 141**
 - Résolution de noms • 142
 - Configuration des serveur DNS • 142
 - Fichier /etc/hosts • 142
- Base de données des utilisateurs et des groupes • 143**
 - Liste des utilisateurs : /etc/passwd • 143
 - Le fichier des mots de passe chiffrés et cachés : /etc/shadow • 144
 - Modifier un compte ou mot de passe existant • 144
 - Bloquer un compte • 144
 - Liste des groupes : /etc/group • 145
- Création de comptes • 146**
- Environnement des interpréteurs de commandes • 147**
- Configuration de l'impression • 148**
- Configuration du chargeur d'amorçage • 148**
 - Identifier ses disques • 148
 - Configuration de LILO • 151
 - Configuration de GRUB 2 • 152
 - Configuration de GRUB Legacy • 153
 - Cas des Macintosh (PowerPC) : configuration de Yaboot • 153
- Autres configurations : synchronisation, logs, partages... • 154**
 - Fuseau horaire • 155
 - Synchronisation horaire • 156
 - Pour les stations de travail • 156
 - Pour les serveurs • 156
 - Rotation des fichiers de logs • 157
 - Partage des droits d'administration • 157
 - Liste des points de montage • 158
 - locate et updatedb • 160
- Compilation d'un noyau • 160**
 - Introduction et prérequis • 160
 - Récupérer les sources • 161
 - Configuration du noyau • 162
 - Compilation et génération du paquet • 163
 - Compilation de modules externes • 163
 - Emploi d'un patch sur le noyau • 165
- Installation d'un noyau • 166**
 - Caractéristiques d'un paquet Debian du noyau • 166
 - Installation avec dpkg • 167
- 9. Services Unix 169**
 - Démarrage du système • 170**
 - Connexion à distance • 174**
 - Connexion à distance : telnet • 174
 - Connexion à distance sécurisée : SSH • 175
 - Authentification par clé • 176
 - Utiliser des applications X11 à distance • 176
 - Créer des tunnels chiffrés avec le port forwarding • 177
 - Accéder à distance à des bureaux graphiques • 179
 - Gestion des droits • 180**
 - Interfaces d'administration • 182**
 - Administrer sur interface web : webmin • 182
 - Configuration des paquets : debconf • 183
 - Les événements système de syslog • 184**
 - Principe et fonctionnement • 184
 - Le fichier de configuration • 185
 - Syntaxe du sélecteur • 185
 - Syntaxe des actions • 186
 - Le super-serveur inetd • 186**
 - Planification de tâches : cron et atd • 187**
 - Format d'un fichier crontab • 188
 - Emploi de la commande at • 189
 - Planification asynchrone : anacron • 190**
 - Les quotas • 191**
 - Sauvegarde • 192**
 - Sauvegarde avec rsync • 193
 - Restauration des machines non sauvegardées • 195
 - Branchements « à chaud » : hotplug • 196**
 - Introduction • 196
 - La problématique du nommage • 196
 - Fonctionnement de udev • 196
 - Cas pratique • 199
 - Gestion de l'énergie • 200**
 - Gestion avancée de l'énergie : APM • 201
 - Économie d'énergie moderne : ACPI • 201
 - Cartes pour portables : PCMCIA • 201**

10. Infrastructure réseau	205	
Passerelle • 206		
Réseau privé virtuel • 208		
OpenVPN • 208		
Infrastructure de clés publiques easy-rsa • 208		
Configuration du serveur OpenVPN • 212		
Configuration du client OpenVPN • 213		
Réseau privé virtuel avec SSH • 214		
IPsec • 215		
PPTP • 215		
Configuration du client • 215		
Configuration du serveur • 216		
Qualité de service • 219		
Principe et fonctionnement • 219		
Configuration et mise en œuvre • 219		
Minimiser le temps de latence : wondershaper • 220		
Configuration standard • 220		
Routage dynamique • 221		
IPv6 • 221		
Serveur de noms (DNS) • 223		
Principe et fonctionnement • 223		
Configuration • 224		
DHCP • 226		
Présentation • 226		
Configuration • 226		
DHCP et DNS • 227		
Outils de diagnostic réseau • 228		
Diagnostic local : netstat • 228		
Diagnostic distant : nmap • 229		
Les sniffers : tcpdump et wireshark • 231		
11. Services réseau : Postfix, Apache, NFS, Samba, Squid, LDAP	235	
Serveur de messagerie électronique • 236		
Installation de Postfix • 236		
Configuration de domaines virtuels • 239		
Domaine virtuel d'alias • 240		
Domaine virtuel de boîtes aux lettres • 240		
Restrictions à la réception et à l'envoi • 241		
Restreindre l'accès en fonction de l'adresse IP • 241		
Vérifier la validité de la commande EHLO ou HELO • 243		
Accepter ou refuser en fonction de l'émetteur (annoncé) • 244		
Accepter ou refuser en fonction du destinataire • 244		
Restrictions associées à la commande DATA • 245		
		Application des restrictions • 245
		Filtrer en fonction du contenu du message • 246
		Mise en place du greylisting • 246
		Personnalisation des filtres en fonction du destinataire • 248
		Intégration d'un antivirus • 249
		SMTP authentifié • 250
		Serveur web (HTTP) • 252
		Installation d'Apache • 252
		Configuration d'hôtes virtuels • 253
		Directives courantes • 255
		Requérir une authentification • 256
		Restrictions d'accès • 256
		Analyseur de logs • 257
		Serveur de fichiers FTP • 259
		Serveur de fichiers NFS • 260
		Sécuriser NFS (au mieux) • 261
		Serveur NFS • 262
		Client NFS • 263
		Partage Windows avec Samba • 264
		Samba en serveur • 264
		Configuration avec debconf • 264
		Configuration manuelle • 265
		Modifications à smb.conf • 265
		Ajout des utilisateurs • 267
		Transformation en contrôleur de domaines • 267
		Samba en client • 268
		Le programme smbclient • 268
		Monter un partage Windows • 269
		Imprimer sur une imprimante partagée • 269
		Mandataire HTTP/FTP • 270
		Installation • 270
		Configuration d'un cache • 270
		Configuration d'un filtre • 271
		Annuaire LDAP • 271
		Installation • 272
		Remplissage de l'annuaire • 273
		Utiliser LDAP pour gérer les comptes • 274
		Configuration de NSS • 274
		Configuration de PAM • 276
		Sécuriser les échanges de données LDAP • 276
		Configuration côté serveur • 277
		Configuration côté client • 278

Introduction à SELinux • 371	
Les principes • 371	
La mise en route • 374	
La gestion d'un système SELinux • 374	
Gestion des modules SELinux • 375	
Gestion des identités • 375	
Gestion des contextes de fichiers, des ports et des booléens • 377	
L'adaptation des règles • 377	
Rédiger un fichier .fc • 378	
Rédiger un fichier .if • 378	
Rédiger un fichier .te • 380	
Compilation des fichiers • 382	
Autres considérations sur la sécurité • 382	
Risques inhérents aux applications web • 383	
Savoir à quoi s'attendre • 383	
Bien choisir les logiciels • 385	
Gérer une machine dans son ensemble • 385	
Les utilisateurs sont des acteurs • 386	
Sécurité physique • 386	
Responsabilité juridique • 387	
En cas de piratage • 387	
Détecer et constater le piratage • 388	
Mettre le serveur hors-ligne • 388	
Préserver tout ce qui peut constituer une preuve • 389	
Réinstaller • 389	
Analyser à froid • 390	
Reconstituer le scénario de l'attaque • 391	
15. Conception d'un paquet Debian 395	
Recompiler un paquet depuis ses sources • 396	
Récupérer les sources • 396	
Effectuer les modifications • 396	
Démarrer la recompilation • 398	
Construire son premier paquet • 399	
Méta-paquet ou faux paquet • 399	
Simple archive de fichiers • 400	
Créer une archive de paquets pour APT • 404	
Devenir mainteneur de paquet • 406	
Apprendre à faire des paquets • 406	
Les règles • 406	
Les procédures • 406	
Les outils • 407	
Le programme lintian • 407	
devscripts • 407	
debhelper et dh-make • 408	
dupload et dput • 408	
Processus d'acceptation • 408	
Prérequis • 408	
Inscription • 409	
Acceptation des principes • 409	
Vérification des compétences • 410	
Approbation finale • 411	
16. Conclusion : l'avenir de Debian 413	
Développements à venir • 414	
Avenir de Debian • 414	
Avenir de ce livre • 415	
A. Distributions dérivées 417	
Recensement et coopération • 418	
Ubuntu Linux • 418	
Knoppix • 419	
Linux Mint • 420	
SimplyMEPIS • 420	
Aptosid (anciennement Sidux) • 420	
Damn Small Linux • 421	
Et d'autres encore • 421	
B. Petit cours de rattrapage 423	
Interpréteur de commandes et commandes de base • 424	
Déplacement dans l'arborescence et gestion des fichiers • 424	
Consultation et modification des fichiers texte • 425	
Recherche de fichiers et dans les fichiers • 425	
Gestion des processus • 425	
Informations système : mémoire, espace disque, identité • 426	
Organisation de l'arborescence des fichiers • 427	
La racine • 427	
Le répertoire personnel de l'utilisateur • 427	
Fonctionnement d'un ordinateur : les différentes couches en jeu • 428	
Au plus bas niveau : le matériel • 428	
Le démarreur : le BIOS • 429	
Le noyau • 430	
L'espace utilisateur • 431	
Quelques fonctions remplies par le noyau • 431	
Pilotage du matériel • 431	
Systèmes de fichiers • 432	
Fonctions partagées • 434	

- Gestion des processus • 434
- Gestion des permissions • 435
- L'espace utilisateur • 435**
 - Processus • 435
 - Démons • 436
 - Communications entre processus • 436

- Bibliothèques • 438
- Glossaire 441**
- Index 463**

Avant-propos

Linux a le vent en poupe depuis quelques 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 elles ont chacune leurs particularités. Autant s'épargner de futurs efforts inutiles de migration !

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 ?

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, Mandriva. . .). Debian est pourtant loin d'être marginale : une étude allemande datant de début 2009 la plaçait en tête pour les serveurs (avec près de la moitié des sociétés interrogées disposant d'un serveur fonctionnant avec Debian), et seconde (après Ubuntu — une distribution dérivée de Debian !) pour les postes bureautiques.

► <http://www.heise.de/open/artikel/Eingesetzte-Produkte-224518.html>

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

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 incluant un noyau Linux, un programme d'installation, et surtout des applications et utilitaires transformant l'ordinateur en outil réellement exploitable.

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 Linux*, développée principalement par la société *Canonical Ltd*, mais aussi de *Mandriva Linux*, réalisée par la société française *Mandriva SA*, 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 l'Internet.

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. Mais depuis le formidable accueil que vous lui aviez accordé, de nombreux autres ouvrages sont apparus sur le sujet.

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 système (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 y 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 exaucés. 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

Comme tous les ouvrages de cette collection, ce livre s'articulera autour d'un cas d'étude concret qui servira à la fois de support et d'illustration pour tous les sujets traités.

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 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 y 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.

Cette cinquième édition est une mise à jour importante puisqu'elle couvre désormais la version 6.0 de Debian, qui porte le nom de code Squeeze. Parmi les changements, citons le support du noyau FreeBSD et des technologies associées (*jails*, *packet filter*...) au sein des nouvelles architectures *kfreebsd-i386* et *kfreebsd-amd64*, un noyau Linux 2.6.32 offrant les principales technologies de virtualisation (Xen/OpenVZ/LXC/KVM, voir page 303), et bien entendu des mises à jour de tous les logiciels fournis. De nombreuses améliorations concernent plus particulièrement les mainteneurs de paquets qui peuvent désormais mettre en œuvre un fichier `debian/rules` simplifié

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 *binnaire* 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.

(grâce à la commande `dh` de `debhelper`), et qui bénéficient d'un système de gestion de patch intégré à `dpkg-source` (en utilisant le format de paquet source nommé `3.0 (quilt)`).

Nous avons placé dans les marges 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 ;
- **CHARTRE 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...**

DVD-Rom d'accompagnement

Le DVD-Rom offert avec ce livre permet d'installer Debian GNU/Linux (pour architecture *i386* et *amd64*) en démarrant l'ordinateur sur le DVD-Rom. Ainsi, après avoir installé cette distribution, il sera directement possible de mettre en pratique les enseignements du livre. Le disque contient en effet la quasi-totalité des programmes étudiés.

Tous les détails sur le fonctionnement du programme d'installation sont donnés dans le chapitre 4.

Remerciements

De Raphaël Hertzog

En premier lieu, je tiens à remercier Nat Makarévitch, qui m'a proposé d'écrire ce livre et m'a accompagné tout au long de sa réalisation ; merci éga-

lement à toute l'équipe d'Eyrolles qui a contribué à ce livre et notamment à Muriel Shan Sei Fan, très patiente avec moi. Merci à Sébastien Blondeel, Florence Henry, Sébastien Mengin et Jean-Côme Charpentier pour leurs contributions.

Un merci particulier à Roland Mas, qui en plus de continuer son minutieux travail de relecture, m'a épaulé dès la troisième édition tant pour la mise à jour que pour la rédaction de chapitres.

Ce livre ne serait pas ce qu'il est sans les relecteurs qui m'ont fait part de leurs judicieuses remarques : Christophe Le Bars et Solveig en particulier. Merci aussi à Charles-André Habib.

Je remercie également Thierry Stempfél pour les belles illustrations introduisant chaque chapitre.

Merci enfin à Sophie d'avoir été si patiente avec moi et de m'avoir soutenu jusqu'au bout. Cette cinquième édition arrive quelques mois après la naissance de notre fils Baptiste et ce n'est pas toujours évident de concilier de longues heures de travail (que j'aimerais calmes et tranquilles) avec les demandes incessantes du petit.

De Roland Mas

Je commencerai bien entendu par remercier Raphaël Hertzog, qui m'a permis de participer, de plus en plus au fil des éditions, à la mise au point de cet ouvrage.

Un grand merci aussi à Muriel Shan Sei Fan et Nat Makarévitch, pour leur encadrement et leurs conseils éclairés, qui ont permis de garder le texte accessible à tout un chacun. Et à Sébastien Mengin et Jean-Côme Charpentier, dont la mise en forme complète d'un austère manuscrit est un plaisir à contempler.

Ayant moi-même relu les deux premières éditions, je ne peux que remercier chaleureusement Solveig, qui a assuré la relecture pour la troisième édition.

Enfin, d'innombrables remerciements à toutes les personnes qui m'ont encouragé au fil du temps — Xavier, Philippe, Laurent, les habitués du canal IRC #debian-devel-fr, et tant d'autres.

Site web et courriel des auteurs

Une section du site web de Raphaël est dédiée à ce livre, et hébergera tout ce qui peut le compléter utilement. On y trouvera par exemple une liste (cliquable) de toutes les URL citées, ou encore les éventuels errata découverts après impression. N'hésitez pas à la consulter et profitez-en pour nous faire part de vos remarques ou messages de soutien en nous écrivant à hertzog@debian.org (pour Raphaël) et lolando@debian.org (pour Roland).

► <http://raphaelhertzog.fr/livre/cahier-admin-debian/>

Chapitre 1



Le projet Debian

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

SOMMAIRE

- ▶ Qu'est-ce que Debian ?
- ▶ Les textes fondateurs
- ▶ Fonctionnement du projet Debian
- ▶ Rôle d'une distribution
- ▶ Cycle de vie d'une release

MOTS-CLÉS

- ▶ Objectif
- ▶ Moyens
- ▶ Fonctionnement
- ▶ Bénévole

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.

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 vivotement 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.

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).

OUTIL Créer un CD-Rom Debian

`debi an-cd` permet de créer des images ISO de CD-Rom d'installation prêts à l'emploi. Raphaël Hertzog est l'auteur de la dernière réécriture, mais la maintenance est essentiellement assurée par Steve McIntyre. Tout ce qui concerne ce logiciel se discute (en anglais) sur la liste de diffusion `debian-cd@lists.debian.org`.

Qu'est-ce que Debian ?

Debian est une distribution GNU/Linux et GNU/kFreeBSD. Nous reviendrons plus en détail sur ce qu'est une distribution en page 18, 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 ou FreeBSD, 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.

Un système d'exploitation multi-plates-formes

Debian, restée fidèle à ses principes initiaux, a connu un tel succès qu'elle atteint aujourd'hui une taille pharaonique. Les 11 architectures proposées couvrent 9 architectures matérielles et 2 noyaux (Linux et FreeBSD). Et avec plus de 14 500 paquets sources, les logiciels disponibles permettent de répondre à une grande partie des besoins que l'on peut avoir aussi bien en entreprise qu'à la maison.

Cet embonpoint devient parfois gênant : il est peu raisonnable de distribuer la cinquantaine de CD-Rom qu'occupe une version complète pour machines à processeurs Intel... C'est pourquoi on la considère de plus en plus comme une « méta-distribution », dont on extrait des distributions plus spécifiques et orientées vers un public particulier : Debian-Desktop pour un usage bureautique traditionnel, Debian-Edu pour un emploi éducatif et pédagogique en milieu scolaire, Debian-Med pour les applications médicales, Debian-Junior pour les jeunes enfants, etc. Une liste plus complète se trouve dans la section dédiée, page 14.

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 de Debian 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.

B.A.-BA À chaque ordinateur son architecture

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.

La plupart des logiciels disponibles pour Debian sont écrits avec des langages de programmation portables : le même code source est compilé sur les diverses architectures. En effet, un exécutable binaire, toujours compilé pour une architecture donnée, ne fonctionne généralement pas sur les autres.

Rappelons que chaque logiciel est créé en rédigeant un code source ; il s'agit d'un fichier textuel composé d'instructions provenant d'un langage de programmation. Avant de pouvoir utiliser le logiciel, il est nécessaire de compiler le code source, c'est-à-dire de le transformer en code binaire (une succession d'instructions machines exécutables par le processeur). Chaque langage de programmation dispose d'un compilateur pour effectuer cette opération (par exemple `gcc` pour le langage C).

La qualité des logiciels libres

Debian suit tous les principes du logiciel libre, et ses nouvelles versions ne sortent que lorsqu'elles sont prêtes. Aucun calendrier préétabli ne contraint les développeurs à bâcler pour respecter une échéance arbitraire. On reproche donc souvent à Debian ses délais de publication, mais cette prudence en garantit aussi la légendaire fiabilité : de longs mois de tests sont en effet nécessaires pour que la distribution complète reçoive le label « stable ».

Debian ne transige pas sur la qualité : tous les *bogues* critiques connus seront corrigés dans toute nouvelle version, même si cela doit parfois retarder la date de sortie initialement prévue.

Debian n'exclut aucune catégorie d'utilisateurs, aussi minoritaire soit-elle. Son programme d'installation est longtemps resté fruste, car c'était le seul capable de fonctionner sur toutes les architectures gérées par le noyau Linux. Il n'était pas envisageable de le remplacer par un programme plus convivial mais limité aux PC (architecture *i386*). Heureusement, depuis l'arrivée de `debian-installer`, cette époque est révolue.

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` sous la direction de Otavio Salvador et Joey Hess.

PERSPECTIVE Au-delà du logiciel

La première version du contrat social disait « Debian demeurera *un ensemble logiciel* totalement libre ». La disparition de ces trois mots (avec la ratification de la version 1.1 du contrat au mois d'avril 2004) traduit une volonté d'obtenir la liberté non seulement des logiciels mais aussi de la documentation et de tout ce que Debian souhaite fournir dans son système d'exploitation.

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 micro-codes (*firmwares*) dans les pilotes pose des problèmes : souvent non-libres, ils sont néanmoins nécessaires au bon fonctionnement du matériel correspondant.

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 un millier de *développeurs Debian* mais fédère un nombre bien plus important de contributeurs (traducteurs, rapporteurs de bogues, développeurs occasionnels...).

Pour mener à bien sa mission, Debian dispose d'une importante infrastructure, comportant de nombreux serveurs reliés à Internet, offerts par de nombreux mécènes.

COMMUNAUTÉ Derrière Debian, l'association SPI et des branches locales

Debian ne possède aucun serveur en son nom propre, puisque ce n'est qu'un projet au sein de l'association *Software in the Public Interest* (SPI), qui en gère les aspects matériels et financiers (dons, achat de matériel...). Bien qu'initialement créée sur mesure pour Debian, cette association coiffe maintenant d'autres projets du monde du logiciel libre, notamment la base de données PostgreSQL, Freedesktop.org (projet de standardisation de certaines briques des bureaux graphiques modernes tels que GNOME et KDE) ; la suite bureautique OpenOffice.org en a longtemps fait partie aussi.

► <http://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. Cela permet d'éviter de coûteux virements internationaux et correspond bien mieux à la nature décentralisée du projet. C'est dans cet esprit que l'association *Debian France* a été fondée au cours de l'été 2006. N'hésitez pas à en devenir membre pour soutenir le projet !

► <http://france.debian.net/>

Les textes fondateurs

Quelques années après son lancement, 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.

Le processus de développement est constamment débattu, mais ces textes fondateurs sont très consensuels, et n'évoluent que rarement. La constitution Debian offre toutefois des garanties supplémentaires : une majorité qualifiée de trois quarts est nécessaire pour approuver tout amendement.

L'engagement vis-à-vis des utilisateurs

On trouve aussi un « contrat social ». Quelle est la place d'un tel texte dans un projet ne visant qu'à concevoir un système d'exploitation ? C'est très simple, Debian œuvre pour ses utilisateurs et, par extension, pour la société. Ce contrat résume donc les engagements pris. Voyons ces points plus en détail :

1 Debian demeurera totalement libre.

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

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.

3 Nous ne dissimulerons pas les problèmes.

Debian n'est pas parfaite, et l'on y découvre tous les jours des problèmes à corriger. Tous ces bogues sont répertoriés et consultables librement, par exemple sur le Web.

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 accepte et comprend que ses utilisateurs souhaitent utiliser certains logiciels non libres. Elle s'engage donc à mettre à leur disposition une partie de son infrastructure, pour distribuer sous forme de paquets Debian les logiciels qui l'autorisent.

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).

Bien souvent, la ligne de démarcation n'est pas aussi nette. Le mainteneur Debian écrit parfois un correctif, qui profite à tous les utilisateurs du logiciel. De manière générale, Debian encourage l'implication des responsables de paquets dans le développement « amont » (ils deviennent alors contributeurs sans se cantonner au rôle de simples utilisateurs d'un logiciel).

COMMUNAUTÉ Responsable de paquet ou mainteneur ?

L'équipe chargée de l'adaptation de Debian en français (on parle de « localisation ») a retenu le terme de « responsable de paquet » pour désigner la personne chargée d'intégrer un paquet à Debian et de l'y faire évoluer. Le terme anglais correspondant est *maintainer* ; c'est pourquoi nous employons souvent le mot « mainteneur », plus concis et tout aussi explicite.

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.

La GNU GPL, utilisée et promue par la FSF (*Free Software Foundation*, ou fondation du logiciel libre), est la plus courante. Elle a pour particularité de s'appliquer à toute œuvre dérivée et redistribuée : un programme intégrant ou utilisant du code GPL ne peut être diffusé que selon ses termes. Elle interdit donc toute récupération dans une application propriétaire. Ceci pose également de gros problèmes pour le réemploi de code GPL dans des logiciels libres incompatibles avec cette licence. Ainsi, il est parfois impossible de lier une bibliothèque diffusée sous GPL à un programme placé sous une autre licence libre. En revanche, cette licence est très solide en droit américain : les juristes de la FSF ont participé à sa rédaction, et elle a souvent contraint des contrevenants à trouver un accord amiable avec la FSF sans aller jusqu'au procès.

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

La licence BSD est la moins restrictive : tout est permis, y compris l'intégration de code BSD modifié dans une application propriétaire. Microsoft ne s'en est d'ailleurs pas privé car la couche TCP/IP de Windows NT est fondée sur celle du noyau BSD.

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

Enfin, la licence artistique réalise un compromis entre les deux précédentes : l'intégration du code dans une application propriétaire est possible, mais toute modification doit être publiée.

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

Le texte complet (en anglais) de ces licences est disponible dans `/usr/share/common-licenses/` sur tout système Debian. Certaines de ces licences disposent de traductions en français, mais leur statut reste officieux, et leur valeur légale est encore en cours de discussion ; le texte de référence reste alors la version anglaise.

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 « VOCABULAIRE » page 93) 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.

Après une première tentative infructueuse en 2004, la suppression totale de la section *non-free* ne devrait plus revenir à l'ordre du jour avant plusieurs années, d'autant plus qu'elle contient de nombreuses documentations utiles qui y ont été déplacées parce qu'elles ne répondaient plus aux nouvelles exigences de la section *main*. C'est notamment le cas pour certaines documentations de logiciels issus du projet GNU (en particulier Emacs et Make).

Signalons que l'existence de *non-free* gêne considérablement la *Free Software Foundation*, car elle l'empêche de recommander officiellement Debian comme système d'exploitation.

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 ».

La licence publique générale de GNU (*GNU General Public License*), la licence BSD et la licence artistique sont des exemples de licences libres traditionnelles respectant les 9 points mentionnés dans ce texte. Vous en trouverez ci-dessous la traduction, telle que publiée sur le site web de Debian.

► http://www.debian.org/social_contract.fr.html#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, source ou binaire*).
- 5 **Aucune discrimination de personne ou de groupe** La licence ne doit discriminer aucune personne ou groupe de personnes.
- 6 **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.
- 7 **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.
- 8 **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.
- 9 **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 appel 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*.

La licence *copyleft* la plus célèbre est sans aucun doute la GNU GPL (elle a pour petites sœurs la GNU LGPL — GNU Lesser General Public License et la GNU FDL — GNU Free Documentation License). Malheureusement, les licences *copyleft* sont généralement incompatibles entre elles ! En conséquence, il est préférable de n'en utiliser qu'une seule.

COMMUNAUTÉ Bruce Perens, un leader chahuté

Bruce Perens, deuxième leader du projet Debian juste après Ian Murdock, fut très controversé pour ses méthodes dynamiques et assez dirigistes. Il n'en reste pas moins un contributeur important, à qui Debian doit notamment la rédaction des fameux « principes du logiciel libre selon Debian » (ou DFSG pour *Debian Free Software Guidelines*), idée originelle d'Ean Schuessler. Par la suite, Bruce en dérivera la célèbre « définition de l'Open Source » en y gommant toutes les références à Debian.

► <http://www.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.

Dernier point anecdotique, c'est à lui que l'on doit l'inspiration des « noms de code » des différentes versions de Debian (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, Testing — Wheezy, Unstable — Sid). Ils correspondent tous à des personnages de *Toy Story*. Ce film d'animation entièrement réalisé en images de synthèse fut produit par Pixar, employeur de Bruce à l'époque où il était leader Debian. Le nom « Sid » a un statut particulier puisqu'il restera éternellement associé à Unstable ; dans le film, il s'agit de l'enfant des voisins, incorrigible brise-tout — gare à vous donc si vous approchez Unstable de trop près ! Par ailleurs, Sid est l'acronyme de *Still In Development* (encore et toujours en cours de développement).

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.

Les développeurs Debian

Les développeurs Debian ont des responsabilités diverses : membres attirés du projet, ils infléchissent grandement les directions qu'il prend. Un développeur Debian est généralement responsable d'au moins un paquet, mais selon son temps disponible et ses envies il a le loisir de s'engager dans de nombreuses équipes, développant ainsi ses responsabilités.

La maintenance des paquets est une activité relativement codifiée, largement documentée voire réglementée. Il faut en effet y respecter toutes les normes édictées par la *charte Debian* (connue sous le nom de *Debian Policy*, en anglais). Fort heureusement, de nombreux outils facilitent le travail du mainteneur. Il peut ainsi se focaliser sur les particularités de son paquet et sur les tâches plus complexes, telles que la correction des bogues.

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

La charte, élément essentiel du projet Debian, énonce les normes assurant à la fois la qualité des paquets et la parfaite interopérabilité de l'ensemble. Grâce à elle, Debian reste cohérent malgré sa taille gigantesque. Cette charte n'est pas figée, mais évolue continuellement grâce aux propositions incessamment formulées sur la liste `debian-policy@lists.debian.org`. Les amendements emportant l'adhésion de tous sont acceptés et appliqués au texte par un petit groupe de mainteneurs sans tâche éditoriale (ils se contentent d'inclure les modifications décidées par les développeurs Debian membres de la liste mentionnée ci-dessus). On peut consulter les actuelles propositions d'amendements *via* le système de suivi de bogues : <http://bugs.debian.org/debian-policy>

OUTIL Base de données des développeurs

Debian dispose d'une base de données comprenant l'ensemble des développeurs enregistrés et les informations qui s'y rattachent (adresse, téléphone, coordonnées géographiques — latitude et longitude...). Certaines de ces informations (nom, prénom, pays, identifiant chez Debian, identifiant IRC, clé GnuPG...) sont publiques et disponibles sur le Web.

► <http://db.debian.org/>

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.

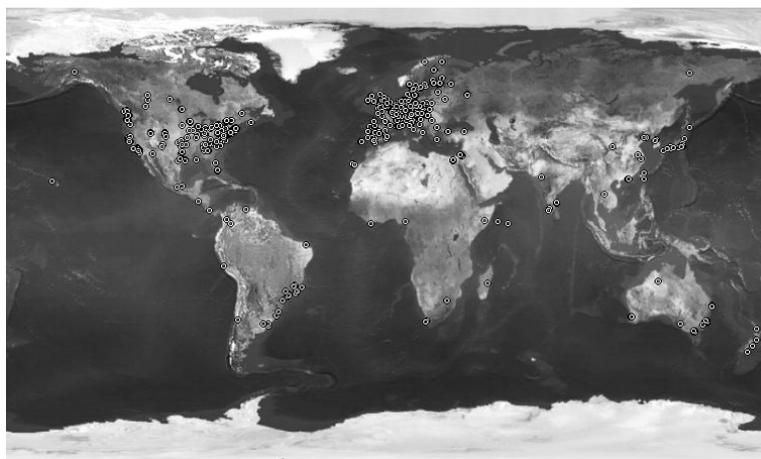


Figure 1.1 Répartition mondiale des développeurs Debian

B.A.-BA Maintenance d'un paquet, le travail du développeur

Maintenir un paquet suppose d'abord d'« emballer » 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 (ex : lors de l'emballage 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).

COMMUNAUTÉ Processus éditorial de la charte

Tout le monde peut proposer une modification de la charte Debian : il suffit de soumettre un rapport de bogue de « gravité » *wishlist* (souhait) sur le paquet *debian-policy*. Le processus qui débute alors est documenté dans `/usr/share/doc/debian-policy/Process.html` : s'il est reconnu que le problème soulevé doit être résolu par le biais d'une nouvelle règle dans la charte Debian, la discussion se poursuit sur `debian-policy@lists.debian.org` jusqu'à l'obtention d'un consensus et d'une proposition. Quelqu'un rédige alors la modification souhaitée et la soumet pour approbation (sous la forme d'un correctif à relire). Dès que 2 autres développeurs approuvent le fait que la formulation proposée reflète bien le consensus ayant émergé de la discussion précédente (en anglais, le verbe consacré est *to second*), la proposition peut être intégrée au document officiel par un des mainteneurs du paquet *debian-policy*. Si le processus échoue à l'une des étapes, les mainteneurs fermeront le bogue en classant la proposition comme rejetée.

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 raisonnables ont été épuisés de part et d'autre. Certains thèmes sont beaucoup plus sujets à polémique que d'autres (l'exemple type étant le choix d'un éditeur de texte, « préférez-vous vi ou emacs ? »). 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.

CHARTRE 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.

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. 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.

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.

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 no-

tamment en charge l'organisation des votes liés aux différentes élections et résolutions générales.

La procédure de « résolution générale » est entièrement détaillée dans la constitution, depuis la période de discussion préalable jusqu'à l'analyse des résultats des votes. Pour plus de détails, nous vous invitons à en consulter le texte intégral : <http://www.debian.org/devel/constitution.fr.html>

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 fonctionnelle et satisfaisante... celle à la réalisation de laquelle une personne compétente aura consacré une partie de son temps.

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. Cette méthode est envisageable car l'essentiel du travail de ces équipes est public, donc a fortiori accessible à tout développeur intéressé. C'est pourquoi Debian est souvent qualifiée de « méritocratie ».

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

Le rôle actif des utilisateurs

Est-il pertinent de citer les utilisateurs parmi les acteurs du fonctionnement de Debian ? Oui : ils y jouent un rôle crucial. Loin d'être « passifs », certains de nos utilisateurs se servent quotidiennement des versions de développement et nous envoient régulièrement des rapports de bogues signalant des problèmes. D'autres vont encore plus loin et formulent des améliorations (par l'intermédiaire d'un bogue de « gravité » *wishlist* — littéralement « liste de vœux »), voire soumettent directement des correctifs du code source (*patches*, voir encadré page 12).

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, l'actuel leader 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.

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 admettons que ce qui n'étaient 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.

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 utilise une échelle étendue de gravités permettant d'exprimer assez finement la gravité d'un bogue. Elle définit par ailleurs très précisément chacun de ces niveaux afin de faciliter le choix de l'un ou l'autre. <http://www.debian.org/Bugs/Developer.fr.html#severities>

► <http://www.debian.org/intl/french/index.fr.html>

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.

OUTIL Système de suivi de bogues

Le système de suivi de bogues *Debian Bug Tracking System* (*Debian BTS*) encadre le projet. Son interface web, partie émergée, permet de consulter tous les bogues répertoriés, et propose d'afficher une liste (triée) de bogues sélectionnés sur de nombreux critères : paquet concerné, gravité, statut, adresse du rapporteur, adresse du mainteneur concerné, étiquette ou *tag*, etc.). Il est encore possible de consulter l'historique complet et toutes les discussions se rapportant à chacun des bogues.

Sous la surface, Debian BTS 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 sera ainsi consigné dans l'historique du bogue 12345. Les personnes habilitées pourront « fermer » ce bogue en écrivant à 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 un rapport respectant un format précis, permettant d'identifier le paquet concerné. L'adresse control@bugs.debian.org propose enfin de manipuler toutes les « méta-informations » relatives à un bogue.

Debian BTS offre bien d'autres fonctionnalités (notamment les *tags*, ou étiquettes) — nous vous invitons à les découvrir en lisant sa documentation en ligne : <http://www.debian.org/Bugs/index.fr.html>

Par ailleurs, de nombreux utilisateurs satisfaits du service offert par Debian souhaitent à leur tour apporter une pierre à l'édifice. Pas toujours pourvus des compétences de programmation adéquates, ils choisissent parfois d'aider à la traduction de documents et aux relectures de celles-ci. Pour les francophones, tout ce travail est coordonné sur la liste debian-l10n-french@lists.debian.org.

B.A.-BA Patch, le moyen d'envoyer un correctif

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 remplaçant 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`.

OUTIL Signaler un bogue avec reportbug

L'outil `reportbug` facilite l'envoi d'un rapport de bogue sur un paquet Debian. Il peut vérifier au préalable que le bogue concerné n'a pas déjà été référencé, ce qui évite la création de doublons. Il rappelle la définition de la gravité pour qu'elle soit aussi juste que possible (le développeur pourra toujours affiner par la suite le jugement de l'utilisateur). Il permet d'écrire un rapport de bogue complet sans en connaître la syntaxe précise, en l'écrivant puis en proposant de le retoucher. Ce rapport sera ensuite transmis via un serveur de courrier électronique (local par défaut, mais `reportbug` peut aussi utiliser un serveur distant).

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.

Tous ces mécanismes sont accentués par le comportement des utilisateurs. Loin d'être isolés, ils forment une vraie communauté, au sein de laquelle de nombreux échanges ont lieu. Citons notamment l'activité impressionnante de la liste de discussion des utilisateurs francophones `debian-user-french@lists.debian.org` (le chapitre 7 vous révélera plus d'informations sur cette dernière).

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.

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

Cette méthode fonctionne plutôt bien puisqu'on retrouve des inconditionnels de Debian à tous les niveaux de la communauté du logiciel libre : dans les *install parties* (ateliers d'installation, encadrés par des habitués, pour les nouveaux venus à Linux) organisées par les groupes locaux d'utilisateurs de Linux, sur les stands associatifs des grands salons d'informatique traitant de Linux, etc.

Signalons que des volontaires réalisent affiches, tracts et autres supports utiles pour la promotion du projet, qu'ils mettent à disposition de tous et que Debian fournit librement sur son site web.

► <http://www.debian.org/events/material.fr.html>

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

PERSPECTIVE Debian en milieu scolaire

Debian-Edu est à l'origine un projet francophone réalisé par Stéphane Casset et Raphaël Hertzog au sein de la société Logidée, pour le compte d'un centre départemental de documentation pédagogique. Raphaël l'a ensuite intégré à Debian en tant que sous-projet. Faute de temps, il n'a plus progressé, comme c'est parfois le cas des logiciels libres dépourvus de contributeurs.

Parallèlement, une équipe de Norvégiens travaillait sur une distribution similaire, également basée sur `debian-installer`. Les progrès de SkoleLinux étant significatifs, Raphaël leur a proposé de s'intégrer à Debian et de reprendre le flambeau de Debian-Edu.

PERSPECTIVE Debian pour le multimédia

Agnula était un projet européen mené sous la direction d'une équipe italienne. Il consistait, pour sa partie « DeMuDi », à développer une version de Debian dédiée aux applications multimédias. Certains membres du projet, et notamment Marco Trevisani, ont voulu le pérenniser en l'intégrant dans Debian. Le sous-projet Debian-Multimedia était né.

► <http://wiki.debian.org/DebianMultimedia>
Le projet a toutefois bien du mal à se forger une identité et à décoller. Free Ekanayaka effectue le travail dans Debian mais propose le résultat sous forme d'une distribution dérivée : il s'agit de 64Studio. Cette distribution est affiliée à une entreprise qui propose de l'assistance technique.

► <http://www.64studio.com/>

de l'infrastructure, la gestion des tâches transversales à tous les paquets (assurance qualité, charte Debian, programme d'installation...), les dernières équipes s'articulant autour de sous-projets.

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 distribution dérivée de Debian la plus célèbre est sans conteste Ubuntu, mais il y en a un grand nombre, consultez l'annexe page A pour découvrir leurs particularités et leur positionnement vis-à-vis de Debian.

Sous-projets Debian existants

À chaque public sa 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...), cela suppose d'améliorer les paquets existants, de mettre en paquet les logiciels manquants, d'adapter l'installateur, de créer une documentation spécifique, etc.

Voici une petite sélection des sous-projets actuels :

- Debian-Junior de Ben Armstrong, vise à proposer aux enfants un système Debian facile et attrayant ;
- Debian-Edu, de Petter Reinholdtsen, se focalise sur la création d'une distribution spécialisée pour le monde éducatif ;
- Debian-Med d'Andreas Tille se consacre au milieu médical ;
- Debian-Multimedia des créateurs d'Agnula traite de création multimédia ;
- Debian-Desktop de Colin Walters s'intéresse à la bureautique ;
- Debian-Ham, créé par Bruce Perens, cible les radio-amateurs ;
- Debian-NP (comme *Non-Profit*) concerne les associations à but non lucratif ;
- Debian-Lex enfin, travaille pour le cadre des cabinets juridiques.

Gageons que cette liste s'étoffera avec le temps et une meilleure perception des avantages des sous-projets Debian. En s'appuyant pleinement sur l'infrastructure existante de Debian, ils peuvent en effet se concentrer sur un travail à réelle valeur ajoutée et n'ont pas à se soucier de « resynchroniser » avec Debian puisqu'ils évoluent dès le début au sein du projet.

Équipes administratives

La plupart des équipes administratives sont relativement fermées et ne recrutent que par cooptation. Le meilleur moyen d'y entrer est alors d'en aider intelligemment les membres actuels en montrant que l'on a compris leurs objectifs et leur mode de fonctionnement.

Les *ftpmasters* sont les responsables de l'archive de paquets Debian. Ils maintiennent le programme qui reçoit les paquets envoyés par les développeurs et les installe automatiquement, après quelques vérifications, sur le serveur de référence (`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`.

OUTIL FusionForge, le couteau suisse du développement collaboratif

FusionForge est un logiciel permettant de créer des sites similaires à `www.sourceforge.net`, `alioth.debian.org` ou encore `savannah.gnu.org`. Il s'agit d'héberger des projets et de leur proposer un ensemble de services facilitant le développement collaboratif. Chaque projet dispose alors d'un espace virtuel dédié, regroupant un site web, un système de suivi de bogues, un système de suivi de patches, un outil de sondages, un espace de dépôt de fichiers, des forums, des dépôts de suivi de sources, des listes de diffusion et divers services annexes.

`alioth.debian.org` est le serveur FusionForge de Debian, administré par Roland Mas, Tollef Fog Heen, Stephen Gran et Christian Bayle. Tout projet impliquant un ou plusieurs développeurs Debian peut y être hébergé.

► <http://alioth.debian.org/>

Très complexe de par l'étendue des services qu'il offre, FusionForge est désormais relativement facile à installer grâce au travail exceptionnel de Roland Mas et Christian Bayle sur le paquet Debian *fusionforge*.

L'équipe *debian-admin* (`debian-admin@lists.debian.org`), comme on peut s'y attendre, est responsable de l'administration système des nombreux serveurs exploités par le projet. Elle veille au fonctionnement optimal de l'ensemble des services de base (DNS, Web, courrier électronique, shell, etc.), installe les logiciels demandés par les développeurs Debian, et prend toutes les précautions en matière de sécurité.

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).

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.

CULTURE Le trafic sur les listes de diffusion : quelques chiffres

Les listes de diffusion sont sans doute le meilleur témoin de l'activité d'un projet, car elles gardent la trace de tout ce qui s'y passe. Quelques statistiques (datant de 2007) concernant nos listes de diffusion parleront d'elles-mêmes : Debian héberge plus de 180 listes totalisant 175 000 abonnements individuels. Les 45 000 messages écrits tous les mois provoquent chaque jour l'envoi de 1 million de courriers électroniques.

Chaque service spécifique dispose de sa propre équipe d'administration système, constituée généralement par les volontaires qui l'ont mise en place (et, souvent, programmé eux-mêmes les outils correspondants). C'est le cas du système de suivi de bogues (BTS), du système de suivi de paquets (*Package Tracking System* — PTS), d'alioth.debian.org (serveur FusionForge, voir encadré), des services disponibles sur qa.debian.org, lintian.debian.org, buildd.debian.org, cdimage.debian.org, etc.

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 en vrac 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.

► <http://packages.qa.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, traductions effectuées (pour les relire), etc.

Les utilisateurs avancés peuvent donc suivre tout cela de près, voire contribuer au projet après avoir bien compris son fonctionnement.

Une autre interface web fournit à chaque développeur un synoptique de l'état de tous les paquets Debian placés sous sa responsabilité.

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

Ces deux sites web constituent des outils pour *Debian QA (Quality Assurance)*, le groupe en charge de l'assurance qualité au sein de Debian.

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 Pa-*

ckage 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 programmes.

CULTURE CVS

CVS (*Concurrent Versions System*) 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, cvs 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. Ce système gère les modifications ligne par ligne en stockant des correctifs différentiels de type *diff* d'une « révision » (version) à l'autre.

CVS utilise une archive centrale (« dépôt » appelé *CVS repository* en anglais), stockant les fichiers et l'historique de leurs modifications (chaque révision est enregistrée sous la forme d'un fichier correctif de type *diff*, prévu pour être appliqué sur la version précédente). Chacun en extrait une version particulière (*working copy* ou « copie de travail ») pour travailler. L'outil permet notamment de consulter les modifications effectuées sur sa copie de travail (`cvs diff`), de les enregistrer dans l'archive centrale en créant une nouvelle entrée dans l'historique des versions (`cvs commit`), de mettre à jour sa copie de travail pour intégrer les modifications effectuées en parallèle par d'autres utilisateurs (`cvs update`), et d'enregistrer dans l'historique une configuration particulière afin de pouvoir facilement l'extraire plus tard (`cvs tag`).

Les experts de cvs sauront 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.

cvs souffre pourtant de quelques limitations. Incapable de gérer les liens symboliques, les changements de noms de fichiers ou de répertoires, la suppression de répertoires, etc. il a contribué à l'apparition de concurrents libres et plus modernes, corrigeant la plupart de ces défauts. Citons notamment *subversion* (svn), *git*, *bazaar* (bzd) et *mercurial* (hg).

- ▶ <http://subversion.tigris.org/>
- ▶ <http://git-scm.com/>
- ▶ <http://bazaar-vcs.org/>
- ▶ <http://mercurial.selenic.com/>

L'équipe la plus importante est probablement celle du programme d'installation de Debian, `debian-installer`, qui a accompli un travail titanique depuis sa conception en 2001. Il lui a fallu recourir à de nombreux contributeurs car il est difficile d'écrire un seul logiciel capable d'installer Debian sur une douzaine d'architectures différentes. Chacune a son propre mécanisme de démarrage et son propre chargeur d'amorçage (*bootloader*). Tout ce travail est coordonné sur la liste de diffusion `debian-boot@lists.debian.org`, sous la houlette de Otavio Salvador et Joey Hess.

-
- ▶ <http://www.debian.org/devel/debian-installer/>
 - ▶ http://kitenet.net/~joey/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` essaye 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*, coordonné par Cyril Brulebois).

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.

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 — ce qui explique l'intégration 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 à l'architecture i386) mais bien plus convivial aux yeux

NOTE DVD-Rom fourni avec le livre

Le DVD-Rom joint à ce livre utilise `debian-installer`. Il permet d'installer Debian 6.0 simplement en démarrant l'ordinateur depuis le DVD-Rom. Il contient en outre la plupart des logiciels étudiés dans ce livre.

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.

La bibliothèque de logiciels

Quantitativement, Debian est indiscutablement en tête avec plus de 14 500 paquets sources. Qualitativement, sa charte et la longue période de tests préalable à toute version stable justifient sa réputation de cohérence et de stabilité. Sur le plan de la disponibilité, on trouve tout en ligne sur de nombreux miroirs mis à jour toutes les 6 heures.

De nombreux commerçants vendent sur le Web des CD-Rom à bas prix (parfois à prix coûtant), dont chacun est libre de télécharger et graver les « images ». Seule ombre au tableau : la faible fréquence de sortie des versions stables (leur élaboration dépasse parfois deux ans), qui ralentit l'intégration de tout nouveau logiciel.

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 pour plus de détails sur le sujet).

Cycle de vie d'une release

Le projet dispose à tout instant de trois ou quatre versions différentes de chaque logiciel, nommées Experimental, Unstable, Testing, et Stable. 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.

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).

D'autre part, 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

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).

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*).

Le statut Unstable

Revenons au cas d'un paquet type. Le mainteneur crée un premier paquet, qu'il compile pour Unstable et place sur le serveur `ftp-master.debian.org`. Cette première manifestation implique inspection et validation par les *ftpmasters*. Le logiciel est alors disponible dans Unstable, distribution risquée mais choisie par des utilisateurs préférant le dernier cri à l'assurance de l'absence de bogues graves. Ceux-ci découvrent alors le programme et le testent.

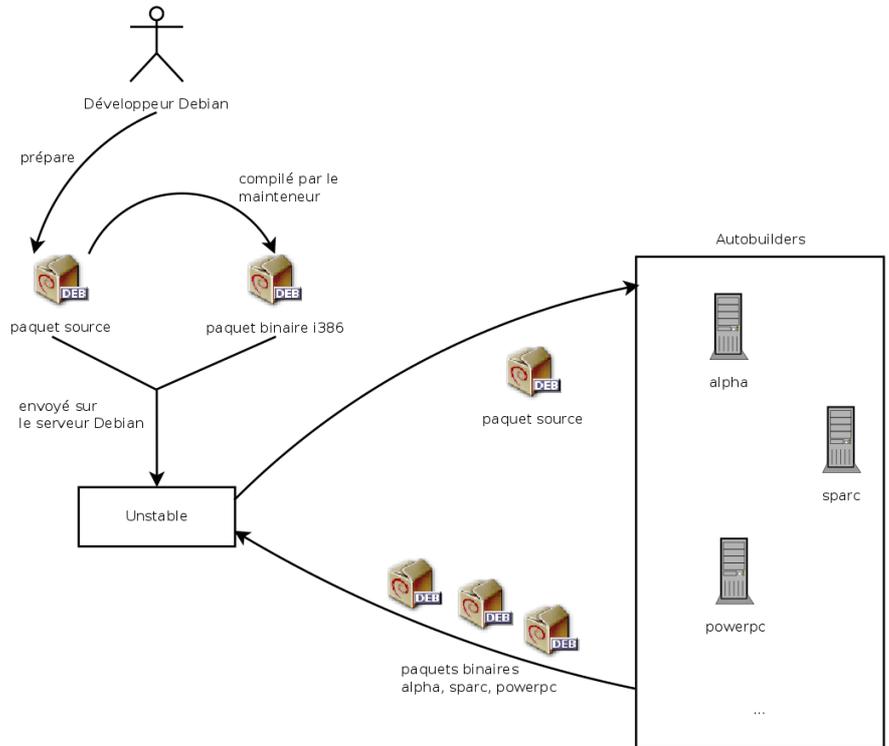


Figure 1.2 Compilation d'un paquet par les autobuilders

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.

Toute nouvelle mise en ligne est répercutée sur tous les miroirs Debian du monde dans les 6 heures. Les utilisateurs valident alors la correction et cherchent d'autres problèmes, consécutifs aux modifications. Plusieurs mises à jour peuvent ainsi s'enchaîner rapidement. Pendant ce temps, les robots *autobuilders* sont entrés en action. Le plus souvent, le mainteneur ne dispose que d'un PC traditionnel et aura compilé son paquet pour architecture i386 (ou amd64) ; les *autobuilders* ont donc pris le relais et compilé automatiquement des versions pour toutes les autres architectures. Certaines compilations pourront échouer ; le mainteneur recevra alors un rapport de bogue signalant le problème, à corriger dans les prochaines versions. Lorsque le bogue est découvert par un spécialiste de l'architecture concernée, il arrive que ce rapport soit accompagné d'un correctif prêt à l'emploi.

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 Villégiature minimale de 10 jours dans Unstable, ce qui laisse assez de temps pour trouver et signaler les problèmes graves.
- 3 Compilation réussie sur toutes les architectures officiellement prises en charge.
- 4 Dépendances pouvant toutes être satisfaites dans Testing, ou qui peuvent du moins y progresser de concert avec le paquet.

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.

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* — n'étant pas toujours satisfaisante).

Ainsi pour produire des binaires destinés à l'architecture *sparc*, le projet dispose de machines *sparc* (en l'occurrence de marque Sun). Le programme *buildd* y fonctionne en permanence afin de créer des paquets binaires pour *sparc* 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.

NOTE Limitations de Testing

Très intéressant dans son principe, Testing pose quelques problèmes pratiques : l'enchevêtrement des dépendances croisées entre paquets est tel que jamais un paquet ne peut y progresser tout seul. Les paquets dépendant tous les uns des autres, il est 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. D'autre part, le script identifiant les familles de paquets ainsi solidarisés peine beaucoup à les constituer (il s'agirait d'un problème NP-complet, auquel 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.

VOCABULAIRE Freeze : la dernière ligne droite

Pendant la période de *freeze* (gel), l'évolution du contenu de la distribution Testing est bloquée : plus aucune mise à jour automatique n'a lieu. Seul 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 corrigent des bogues importants.

COMMUNAUTÉ Le Release Manager

Release Manager (gestionnaire de version) 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 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.

La promotion de Testing en Stable

Supposons notre paquet désormais intégré à Testing. Tant qu'il est perfectible, son responsable doit persister à l'améliorer et recommencer le processus depuis Unstable (mais ces inclusions ultérieures dans Testing sont en général plus rapides : si elles n'ont pas trop évolué, toutes les dépendances sont déjà présentes). Quand il atteint la perfection, son mainteneur a fini son travail, et la prochaine étape est l'inclusion dans la distribution Stable, en réalité une simple copie de Testing à un moment choisi par le *Release Manager*. L'idéal est de prendre cette décision quand l'installateur est prêt et quand plus aucun programme de Testing n'a de bogue critique répertorié.

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.

Après la sortie de la nouvelle version stable, le *Stable Release Manager* en gère les évolutions ultérieures (appelées « révisions ». ex : 5.0.1, 5.0.2, 5.0.3 pour la version 5.0). Ces mises à jour intègrent systématiquement tous les correctifs de sécurité. On y trouve également les corrections les plus importantes (le mainteneur du paquet doit prouver la gravité du problème qu'il souhaite corriger pour faire intégrer sa mise à jour).

Fin du voyage : notre hypothétique paquet est désormais intégré à la distribution stable. Ce trajet, non dépourvu de difficultés, explique les délais importants séparant les versions stables de Debian. Il contribue surtout à sa réputation de qualité. De plus, la majorité des utilisateurs est satisfaite

par l'emploi de l'une des trois distributions disponibles en parallèle. Les administrateurs système, soucieux avant tout de la stabilité de leurs serveurs, se moquent de la dernière version de GNOME ; ils opteront pour Debian Stable et en seront satisfaits. Les utilisateurs finaux, plus intéressés par la dernière version de GNOME ou de KDE que par une stabilité irréprochable, trouveront en Debian Testing un bon compromis entre absence de problèmes graves et logiciels relativement à jour. Enfin, les développeurs et utilisateurs les plus expérimentés pourront ouvrir la voie en testant toutes les nouveautés de Debian Unstable dès leur sortie, au risque de subir les affres et bogues inhérents à toute nouvelle version de logiciel. À chaque public sa Debian !

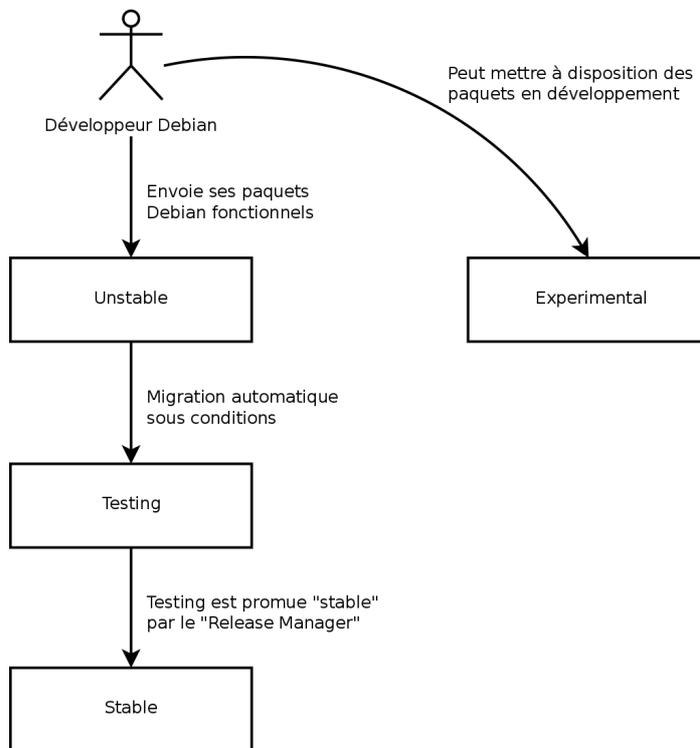


Figure 1.3 Parcours d'un paquet au sein des différentes versions de Debian

CULTURE GNOME et KDE, les bureaux graphiques

GNOME (« *GNU Network Object Model Environment* », ou environnement réseau de modèle objet de GNU) et KDE (« *K Desktop Environment* », ou environnement de bureau K) sont les deux « bureaux graphiques » les plus populaires dans le milieu du logiciel libre. On entend par là un ensemble de logiciels de bureautique permettant d'effectuer aisément les opérations les plus courantes au travers d'une interface graphique. Ils comportent notamment un gestionnaire de fichiers, une suite bureautique, un navigateur web, un logiciel de courrier électronique, des accessoires multimédias, etc. Leur différence la plus visible réside dans le choix de la bibliothèque graphique employée : GNOME a choisi GTK+ (logiciel libre sous licence LGPL) et KDE a opté pour Qt (de la société Trolltech qui la diffuse sous licence GPL).

- <http://www.gnome.org/>
- <http://www.kde.org/>

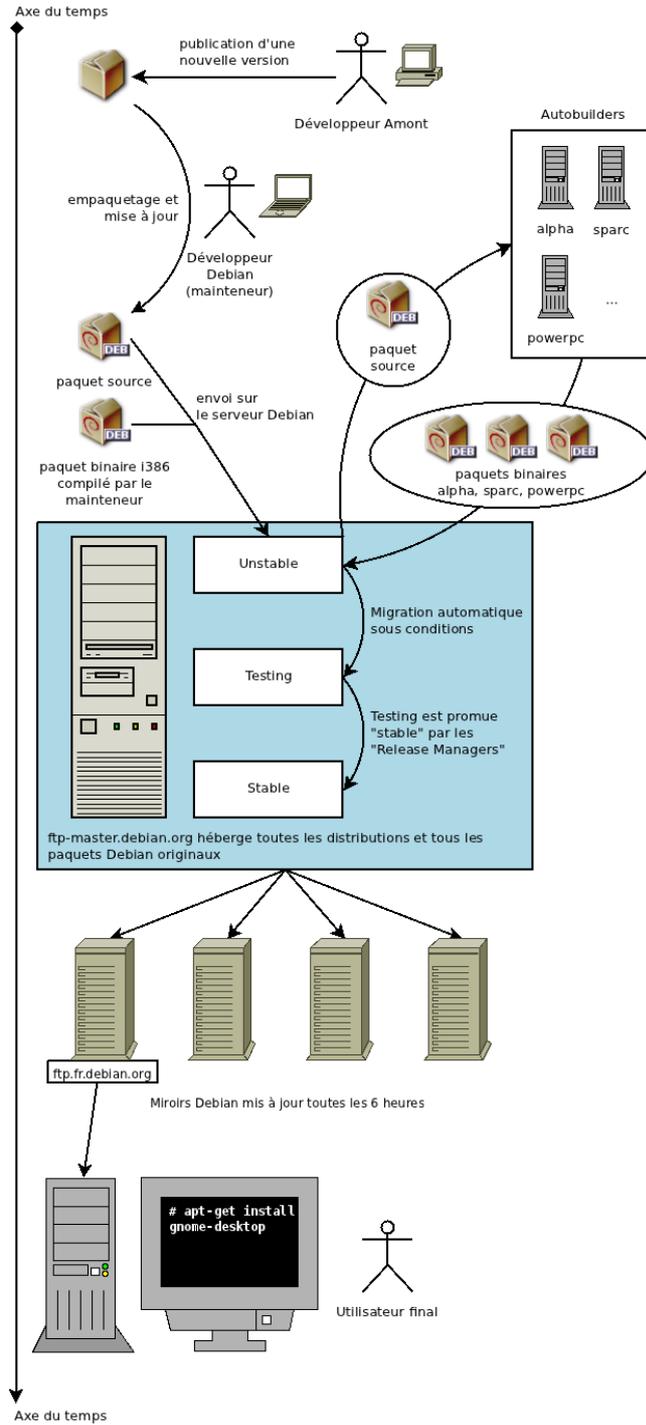
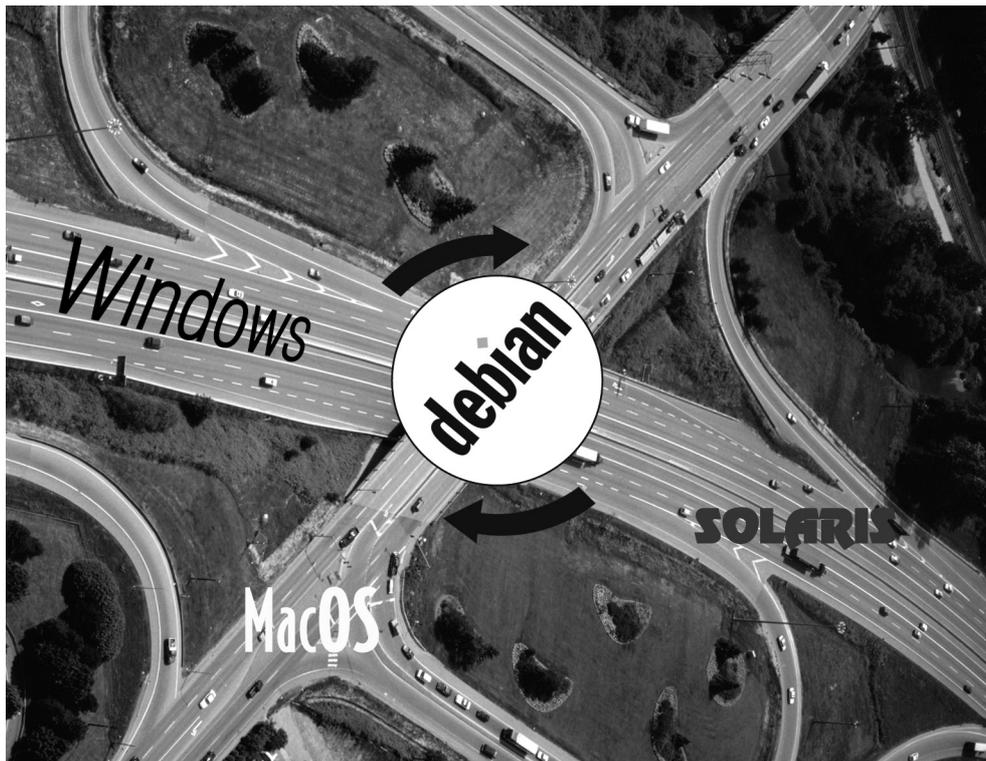


Figure 1.4 Parcours chronologique d'un paquet logiciel empaqueté par Debian



Chapitre **2**



Présentation de l'étude de cas

Vous êtes administrateur système d'une PME en pleine croissance. En collaboration avec votre direction, vous venez de redéfinir le plan directeur du système informatique pour l'année qui vient, et avez choisi de migrer progressivement vers Debian pour des raisons tant pratiques qu'économiques. Détaillons ce qui vous attend...

SOMMAIRE

- ▶ Des besoins informatiques en forte hausse
- ▶ Plan directeur
- ▶ Pourquoi une distribution GNU/Linux ?
- ▶ Pourquoi la distribution Debian ?
- ▶ Pourquoi Debian Squeeze ?

MOTS-CLÉS

- ▶ Falcot SA
- ▶ PME
- ▶ Forte croissance
- ▶ Plan directeur
- ▶ Migration
- ▶ Réduction des coûts