

debian

Référence Debian

Osamu Aoki

Copyright © 2013-2026 Osamu Aoki

Ce guide de référence Debian (version 2.153) (2026-11-29 05:38:00 UTC) est destiné à procurer un large aperçu du système Debian en tant que guide de l’utilisateur d’un système installé. Il couvre de nombreux aspects de l’administration du système à l’aide d’exemples de commandes de l’interpréteur pour les non-développeurs.

Table des matières

1 Didacticiels GNU/Linux **1**

1.1 Bases pour la console . . . 1

1.1.1 L'invite de l'interpréteur de commandes . . . 1

1.1.2 Invite de l'interpréteur de commandes avec une interface graphique . . . 2

1.1.3 Compte de l'administrateur (root) . . . 2

1.1.4 Invite de l'interpréteur de commandes pour l'administrateur . . . 3

1.1.5 Outils graphiques d'administration du système . . . 3

1.1.6 Consoles virtuelles . . . 3

1.1.7 Comment quitter l'invite de l'interpréteur de commandes . . . 4

1.1.8 Comment arrêter le système . . . 4

1.1.9 Récupérer une console propre . . . 4

1.1.10 Suggestions de paquets supplémentaires pour le débutant . . . 4

1.1.11 Compte pour un utilisateur supplémentaire . . . 5

1.1.12 Configuration de sudo . . . 5

1.1.13 À vous de jouer . . . 6

1.2 Système de fichiers de type UNIX . . . 6

1.2.1 Bases concernant les fichiers UNIX . . . 7

1.2.2 Fonctionnement interne du système de fichiers . . . 8

1.2.3 Permissions du système de fichiers . . . 8

1.2.4 Contrôle des permissions pour les fichiers nouvellement créés : umask . . . 10

1.2.5 Permissions pour les groupes d'utilisateurs (group) . . . 11

1.2.6 Horodatage . . . 12

1.2.7 Liens . . . 13

1.2.8 Tubes nommés (FIFO) . . . 14

1.2.9 Sockets . . . 15

1.2.10 Fichiers de périphériques . . . 15

1.2.11 Fichiers spéciaux de périphériques . . . 16

1.2.12 procfs et sysfs . . . 17

1.2.13 tmpfs . . . 17

1.3 Midnight Commander (MC) . . . 17

1.3.1 Personnalisation de MC . . . 18
1.3.2 Démarrer MC . . . 18
1.3.3 Gestionnaire de fichiers de MC . . . 18
1.3.4 Astuces de la ligne de commandes dans MC . . . 19
1.3.5 Éditeur interne de MC . . . 19
1.3.6 Visualisateur interne de MC . . . 19
1.3.7 Possibilités de démarrage automatique de MC . . . 20
1.3.8 Système de fichiers virtuel de MC . . . 20
1.4 L'environnement élémentaire de travail de type UNIX . . . 20
1.4.1 L'interpréteur de commandes de connexion . . . 20
1.4.2 Personnaliser bash . . . 20
1.4.3 Combinaisons particulières de touches . . . 22
1.4.4 Opérations de souris . . . 22
1.4.5 Le visualisateur de fichiers . . . 23
1.4.6 L'éditeur de texte . . . 23
1.4.7 Définir un éditeur de texte par défaut . . . 24
1.4.8 Utilisation de vim . . . 24
1.4.9 Enregistrer les actions de l'interpréteur de commandes . . . 26
1.4.10 Commandes UNIX de base . . . 26
1.5 La commande simple de l'interpréteur de commandes . . . 28
1.5.1 Exécution d'une commande et variables d'environnement . . . 28
1.5.2 La variable « $LANG » . . . 28
1.5.3 La variable « $PATH » . . . 30
1.5.4 La variable « $HOME » . . . 30
1.5.5 Options de la ligne de commandes . . . 30
1.5.6 Motifs génériques (« glob ») de l'interpréteur de commandes . . . 31
1.5.7 Valeur de retour d'une commande . . . 32
1.5.8 Séquences de commandes typiques et redirection de l'interpréteur de commandes . . . 32
1.5.9 Alias de commande . . . 34
1.6 Traitement des données textuelles à la UNIX . . . 34
1.6.1 Outils de traitement de texte d'UNIX . . . 34
1.6.2 Expressions rationnelles . . . 35
1.6.3 Expressions de remplacement . . . 37
1.6.4 Substitution globale avec des expressions rationnelles . . . 37
1.6.5 Extraire des données d'un tableau contenu dans un fichier texte . . . 38
1.6.6 Bouts de scripts pour les tubes . . . 40

2 Gestion des paquets Debian 41

2.1 Prérequis pour la gestion des paquets Debian . . . 41
2.1.1 Système de gestion des paquets Debian . . . 41
2.1.2 Configuration de paquets . . . 41
2.1.3 Précautions de base . . . 42
2.1.4 La vie avec d'éternelles mises à jour . . . 43
2.1.5 Bases concernant l'archive Debian . . . 44
2.1.6 Debian est totalement libre . . . 48
2.1.7 Dépendances des paquets . . . 49
2.1.8 Flux des événements dans la gestion d'un paquet . . . 50
2.1.9 Première réponse aux problèmes de gestion de paquets . . . 50
2.1.10 Comment obtenir des paquets Debian . . . 51
2.1.11 Comment faire face à des exigences conflictuelles . . . 51
2.2 Opérations de base de la gestion des paquets . . . 52
2.2.1 `apt` vs. `apt-get` / `apt-cache` vs. `aptitude` . . . 52
2.2.2 Opérations de base de gestion des paquets en ligne de commandes . . . 53
2.2.3 Utilisation interactive d'aptitude . . . 53
2.2.4 Raccourcis clavier d'aptitude . . . 55
2.2.5 Vues des paquets sous aptitude . . . 56
2.2.6 Options de la méthode de recherche avec aptitude . . . 57
2.2.7 Les formules d'expressions rationnelles d'aptitude . . . 57
2.2.8 Résolution des dépendances par aptitude . . . 59
2.2.9 Journaux d'activité des paquets . . . 59
2.3 Exemples d'opérations avec aptitude . . . 59
2.3.1 Rechercher des paquets intéressants . . . 59
2.3.2 Afficher les paquets dont les noms correspondent à une expression rationnelle . . . 60
2.3.3 Parcours en correspondance avec une expression rationnelle . . . 60
2.3.4 Purger pour de bon les paquets supprimés . . . 60
2.3.5 Toilettage de l'état d'installation automatique/manuel . . . 60
2.3.6 Mise à jour pour l'ensemble du système . . . 61
2.4 Opérations avancées de gestion des paquets . . . 62
2.4.1 Opérations avancées de gestion des paquets en ligne de commandes . . . 62
2.4.2 Vérifier les fichiers de paquets installés . . . 64
2.4.3 Protection contre les problèmes de paquets . . . 64
2.4.4 Rechercher dans les métadonnées du paquet . . . 64
2.5 Fonctionnement interne de la gestion des paquets Debian . . . 65
2.5.1 Métadonnées de l'archive . . . 65
2.5.2 Fichier « Release » de plus haut niveau et authenticité . . . 65
2.5.3 Fichiers « Release » de niveau de l'archive . . . 66

2.5.4 Récupérer les métadonnées d'un paquet . . . 67
2.5.5 État des paquets pour APT . . . 67
2.5.6 État des paquets pour aptitude . . . 67
2.5.7 Copies locales des paquets téléchargés . . . 68
2.5.8 Nom de fichier d'un paquet Debian . . . 68
2.5.9 La commande dpkg . . . 69
2.5.10 La commande update-alternatives . . . 69
2.5.11 Commande dpkg-statoverride . . . 70
2.5.12 Commande dpkg-divert . . . 70
2.6 Récupérer un système cassé . . . 70
2.6.1 Incompatibilité avec une ancienne configuration de l'utilisateur . . . 71
2.6.2 Erreurs de mise en cache des données du paquet . . . 71
2.6.3 Récupération avec la commande dpkg . . . 71
2.6.4 Échec d'installation à cause de dépendances manquantes . . . 71
2.6.5 Différents paquets ayant des fichiers communs . . . 72
2.6.6 Corriger les scripts cassés des paquets . . . 72
2.6.7 Récupérer les données de sélection des paquets . . . 73
2.7 Astuces pour la gestion des paquets . . . 73
2.7.1 Qui a envoyé le paquet ? . . . 73
2.7.2 Diminuer la bande passante utilisée par APT . . . 73
2.7.3 Chargement et mise à niveau automatique de paquets . . . 74
2.7.4 Mises à jour et rétroportages . . . 74
2.7.5 Archives de paquets externes . . . 75
2.7.6 Paquets de sources mixtes d'archives sans **apt-pinning** . . . 75
2.7.7 Ajustement de la version candidate avec **apt-pinning** . . . 76
2.7.8 Blocage des paquets installés par « Recommends » . . . 77
2.7.9 Suivre `testing` avec quelques paquets d'`unstable` . . . 78
2.7.10 Suivre `unstable` avec quelques paquets d'`experimental` . . . 79
2.7.11 Retour d'urgence à une version précédente (downgrade) . . . 80
2.7.12 Paquet equivs . . . 80
2.7.13 Porter un paquet vers le système stable . . . 81
2.7.14 Serveur mandataire (proxy) pour APT . . . 82
2.7.15 Autres lectures concernant la gestion des paquets . . . 82

3 Initialisation du système 83
3.1 Aperçu du processus d'amorçage du système . . . 83
3.1.1 Étage 1 : UEFI . . . 83
3.1.2 Étage 2 : le chargeur initial . . . 84
3.1.3 Étage 3 : le système mini-Debian . . . 85
3.1.4 Étage 4 : le système Debian normal . . . 86
3.2 Rescue system . . . 86
3.2.1 GRUB UEFI rescue system on USB . . . 87
3.2.2 Linux live rescue system on USB . . . 88
3.2.3 Linux live rescue system from GRUB . . . 89
3.3 Systemd . . . 89
3.3.1 Initialisation avec Systemd . . . 89
3.3.2 Connexion avec Systemd . . . 90
3.4 Messages du noyau . . . 90
3.5 Messages du système . . . 90
3.6 Gestion du système . . . 91
3.7 Autres moniteurs du système . . . 93
3.8 Configuration du système . . . 93
3.8.1 Nom de machine (« hostname ») . . . 93
3.8.2 Le système de fichiers . . . 93
3.8.3 Initialisation de l'interface réseau . . . 94
3.8.4 Initialisation du système d'infonuagique . . . 94
3.8.5 Exemple de personnalisation pour ajuster le service sshd . . . 94
3.9 Le système udev . . . 95
3.10 Initialisation des modules du noyau . . . 95

4 Contrôle d'authentification et d'accès 97
4.1 Authentification normale d'UNIX . . . 97
4.2 Gestion des informations des comptes et des mots de passes . . . 99
4.3 Mot de passe de qualité . . . 99
4.4 Créer un mot de passe chiffré . . . 99
4.5 PAM et NSS . . . 100
4.5.1 Fichiers de configuration auxquels accèdent PAM et NSS . . . 101
4.5.2 Le système de gestion centralisée moderne . . . 101
4.5.3 « Pourquoi la commande su de GNU ne gère-t-elle pas le groupe wheel » . . . 102
4.5.4 Règle de mots de passe plus stricte . . . 102
4.6 Sécurité de l'authentification . . . 103
4.6.1 Mot de passe sûr avec Internet . . . 103
4.6.2 Le shell sûr (Secure Shell) . . . 103

4.6.3 Mesures de sécurité supplémentaires pour Internet . . . 104
4.6.4 sécuriser le mot de passe de l'administrateur . . . 104
4.7 Autres contrôles d'accès . . . 105
4.7.1 Listes de contrôle d'accès (ACL) . . . 105
4.7.2 sudo . . . 105
4.7.3 PolicyKit . . . 106
4.7.4 Restreindre l'accès à certains services du serveur . . . 106
4.7.5 Caractéristiques de sécurité de Linux . . . 107

5 Configuration du réseau 108
5.1 L'infrastructure de base du réseau . . . 108
5.1.1 Résolution du nom d'hôte . . . 108
5.1.2 Nom de l'interface réseau . . . 110
5.1.3 Plage d'adresses réseau du réseau local (« LAN ») . . . 111
5.1.4 La gestion du périphérique réseau . . . 111
5.2 Configuration moderne de réseau pour ordinateur de bureau . . . 111
5.2.1 Outils graphiques de configuration du réseau . . . 112
5.3 Configuration moderne de réseau sans interface graphique . . . 112
5.4 Configuration moderne de réseau pour l'infonuagique . . . 113
5.4.1 Configuration moderne de réseau pour l'infonuagique avec DHCP . . . 113
5.4.2 Configuration moderne de réseau pour l'infonuagique avec IP statique . . . 113
5.4.3 Configuration moderne de réseau pour l'infonuagique avec Network Manager . . . 113
5.5 Configuration réseau de bas niveau . . . 114
5.5.1 Commandes Iproute2 . . . 114
5.5.2 Opérations sûres de bas niveau sur le réseau . . . 114
5.6 Optimisation du réseau . . . 114
5.6.1 Rechercher le MTU optimal . . . 114
5.6.2 Optimisation de TCP sur le réseau Internet . . . 116
5.7 Infrastructure de netfilter . . . 117

6 Applications réseau 118
6.1 Navigateurs Web . . . 118
6.1.1 Usurpation de la chaîne User-Agent . . . 119
6.1.2 Extension de navigateur . . . 119
6.2 Le système de courrier électronique . . . 119
6.2.1 Bases du courrier électronique . . . 119
6.2.2 Limite du service de courriels moderne . . . 120
6.2.3 Attente du service de courriels historique . . . 120
6.2.4 Agent de transport de courrier électronique (« MTA ») . . . 121

6.2.4.1 Configuration d'exim4 . . . 121
6.2.4.2 Configuration de postfix avec SASL . . . 123
6.2.4.3 Configuration de l'adresse de courriel . . . 124
6.2.4.4 Opération de base du MTA . . . 125
6.3 Le serveur et les utilitaires d'accès à distance (SSH) . . . 125
6.3.1 Bases de SSH . . . 126
6.3.2 Nom d'utilisateur sur l'hôte distant . . . 127
6.3.3 Se connecter sans mot de passe distant . . . 127
6.3.4 Clients SSH exotiques . . . 127
6.3.5 Configurer ssh-agent . . . 128
6.3.6 Envoi d'un courriel à partir d'un hôte distant . . . 128
6.3.7 Redirection de port pour un tunnel SMTP/POP3 . . . 128
6.3.8 Comment arrêter le système distant par SSH . . . 128
6.3.9 Résoudre les problèmes avec SSH . . . 129
6.4 Le serveur et les utilitaires d'impression . . . 129
6.5 Autres serveurs d'applications réseau . . . 129
6.6 Autres clients d'applications réseau . . . 130
6.7 Le diagnostic des démons du système . . . 130

7 Système d'interface graphique 133
7.1 Environnement de bureau avec interface graphique . . . 133
7.2 Protocole de communication graphique . . . 134
7.3 Infrastructure d'interface graphique . . . 135
7.4 Applications graphiques . . . 135
7.5 Répertoires de l'utilisateur . . . 137
7.6 Fontes de caractères . . . 137
7.6.1 Fontes de base . . . 137
7.6.2 Matricialisation des fontes . . . 139
7.7 Bac à sable . . . 139
7.8 Bureau à distance . . . 140
7.9 Connexion au serveur X . . . 140
7.9.1 Connexion locale au serveur X . . . 140
7.9.2 Connexion à distance au serveur X . . . 142
7.9.3 Connexion avec chroot au serveur X . . . 142
7.10 Presse-papier . . . 142

8 I18N et L10N 144

8.1 Les paramètres linguistiques (« locale ») . . . 144
8.1.1 Justification de l'utilisation d'UTF-8 dans les paramètres linguistiques . . . 144
8.1.2 Reconfiguration des paramètres linguistiques . . . 145
8.1.3 Coder les noms de fichiers . . . 146
8.1.4 Messages et documentation traduits . . . 146
8.1.5 Effet des paramètres linguistiques . . . 147
8.2 L'entrée clavier . . . 147
8.2.1 La saisie avec le clavier pour la console Linux et X Window . . . 148
8.2.2 La saisie avec le clavier pour Wayland . . . 148
8.2.3 Prise en charge de la méthode d'entrée avec iBus . . . 148
8.2.4 The input method support with Fcitx . . . 148
8.2.5 Un exemple pour le japonais . . . 151
8.3 L'affichage de sortie . . . 151
8.3.1 Configuration du terminal . . . 151
8.3.2 Largeur des caractères ambigus d'Asie orientale . . . 152

9 Astuces du système 153

9.1 Conseils pour la console . . . 153
9.1.1 Enregistrer proprement l'activité de la console . . . 153
9.1.2 Le programme screen . . . 154
9.1.3 Navigation dans les répertoires . . . 155
9.1.4 Enveloppe pour Readline . . . 155
9.1.5 Analyse de l'arborescence du code source . . . 156
9.2 Personnaliser vim . . . 156
9.2.1 Personnalisation de vim avec des fonctionnalités internes . . . 156
9.2.2 Personnalisation de vim avec des paquets externes . . . 158
9.3 Enregistrer et présenter des données . . . 159
9.3.1 Le démon de journal . . . 159
9.3.2 Analyseur de journaux . . . 160
9.3.3 Affichage personnalisé des données de texte . . . 160
9.3.4 Affichage personnalisé de la date et de l'heure . . . 160
9.3.5 Écho colorisé de l'interpréteur de commandes . . . 161
9.3.6 Commandes colorisées . . . 161
9.3.7 Enregistrer l'activité de l'éditeur pour des répétitions complexes . . . 162
9.3.8 Enregistrer l'image graphique d'une application X . . . 162
9.3.9 Enregistrer les modifications dans des fichiers de configuration . . . 163
9.4 Surveiller, contrôler et démarrer l'activité des programmes . . . 163
9.4.1 Temps d'un processus . . . 164

9.4.2 La priorité d'ordonnancement . . . 164
9.4.3 La commande ps . . . 164
9.4.4 La commande top . . . 164
9.4.5 Afficher les fichier ouverts par un processus . . . 165
9.4.6 Tracer l'activité d'un programme . . . 165
9.4.7 Identification des processus qui utilisent des fichiers ou des sockets . . . 165
9.4.8 Répéter une commande avec un intervalle constant . . . 165
9.4.9 Répéter une commande en bouclant entre des fichiers . . . 166
9.4.10 Lancer un programme depuis l'interface graphique . . . 166
9.4.11 Personnaliser le programme à lancer . . . 167
9.4.12 Tuer un processus . . . 168
9.4.13 Planifier des tâches qui s'exécutent une fois . . . 168
9.4.14 Planifier des tâches qui s'exécutent régulièrement . . . 168
9.4.15 Planifier des tâches lors d'évènements . . . 170
9.4.16 touche Alt-SysRq . . . 170
9.5 Astuces de maintenance du système . . . 171
9.5.1 Qui se trouve sur le système ? . . . 171
9.5.2 Prévenir tout le monde . . . 171
9.5.3 Identification du matériel . . . 171
9.5.4 Configuration matérielle . . . 173
9.5.5 Heure système et matérielle . . . 173
9.5.6 L'infrastructure de gestion du son . . . 173
9.5.7 Désactiver l'économiseur d'écran . . . 174
9.5.8 Désactiver les bips . . . 174
9.5.9 Utilisation de la mémoire . . . 175
9.5.10 Vérification de la sécurité et de l'intégrité du système . . . 175
9.6 Astuces relatives au stockage des données . . . 176
9.6.1 Utilisation de l'espace disque . . . 176
9.6.2 Configuration de la partition du disque . . . 177
9.6.3 Accès à une partition en utilisant l'UUID . . . 177
9.6.4 LVM2 . . . 178
9.6.5 Configuration de systèmes de fichiers . . . 178
9.6.6 Création et vérification de l'intégrité d'un système de fichiers . . . 179
9.6.7 Optimisation du système de fichiers à l'aide des options de montage . . . 179
9.6.8 Optimisation du système de fichiers à l'aide du superbloc . . . 180
9.6.9 Optimisation du disque dur . . . 180
9.6.10 Optimisation du SSD . . . 181
9.6.11 Utiliser SMART pour prédire les défaillances des disques durs . . . 181
9.6.12 Indication du répertoire de stockage temporaire à l'aide de $TMPDIR . . . 181

9.6.13 Étendre l'espace de stockage utile à l'aide de LVM . . . 181
9.6.14 Extension de l'espace de stockage en montant une autre partition . . . 181
9.6.15 Extension de l'espace de stockage en remontant un autre répertoire . . . 182
9.6.16 Extension de l'espace de stockage utilisable en montant en superposition (overlay) un autre répertoire . . . 182
9.6.17 Extension de l'espace utilisable à l'aide de liens symboliques . . . 182
9.7 Le fichier image du disque . . . 183
9.7.1 Créer le fichier image du disque . . . 183
9.7.2 Écrire directement sur le disque . . . 183
9.7.3 Monter le fichier image du disque . . . 184
9.7.4 Nettoyage d'un fichier image du disque . . . 185
9.7.5 Réaliser le fichier image d'un disque vide . . . 186
9.7.6 Créer un fichier image ISO9660 . . . 186
9.7.7 Écriture directe sur CD/DVD-R/RW . . . 187
9.7.8 Monter le fichier image ISO9660 . . . 187
9.8 Les données binaires . . . 188
9.8.1 Afficher et éditer des données binaires . . . 188
9.8.2 Manipulating files without mounting the filesystem . . . 188
9.8.3 Redondance des données . . . 188
9.8.4 Récupération de fichiers de données et analyse par autopsie . . . 190
9.8.5 Éclater un gros fichier en petits fichiers . . . 190
9.8.6 Effacer le contenu d'un fichier . . . 190
9.8.7 Fichiers fictifs . . . 190
9.8.8 Effacer l'ensemble du disque dur . . . 191
9.8.9 Effacer l'ensemble du disque dur . . . 191
9.8.10 Récupérer des fichiers supprimés mais encore ouverts . . . 191
9.8.11 Rechercher tous les liens physiques . . . 192
9.8.12 Consommation d'espace disque invisible . . . 192
9.9 Astuces de chiffrement des données . . . 193
9.9.1 Chiffrement des disques amovibles à l'aide de dm-crypt/LUKS . . . 193
9.9.2 Monter des disques amovibles chiffrés à l'aide de dm-crypt/LUKS . . . 194
9.10 Le noyau . . . 194
9.10.1 Paramètres du noyau . . . 194
9.10.2 En-têtes du noyau . . . 195
9.10.3 Compiler le noyau et les modules associés . . . 195
9.10.4 Compiler les sources du noyau : recommandations de l'équipe en charge du noyau Debian . . . 196
9.10.5 Pilotes de matériel et microprogramme . . . 196
9.11 Système virtualisé . . . 197
9.11.1 Outils de virtualisation et d'émulation . . . 197
9.11.2 Étapes de la virtualisation . . . 198
9.11.3 Monter le fichier image du disque virtuel . . . 198
9.11.4 Système protégé (chroot) . . . 200
9.11.5 Systèmes de bureaux multiples . . . 201

10 Gestion des données 202

10.1 Partager, copier et archiver . . . 202
10.1.1 Outils d'archivage et de compression . . . 202
10.1.2 Outils de copie et de synchronisation . . . 204
10.1.3 Idiomes pour les archives . . . 205
10.1.4 Idiomes pour la copie . . . 205
10.1.5 Idiomes pour la sélection de fichiers . . . 206
10.1.6 Support d'archive . . . 207
10.1.7 Périphériques d'enregistrement amovibles . . . 208
10.1.8 Choix de système de fichiers pour les données partagées . . . 208
10.1.9 Partage de données au travers du réseau . . . 210
10.2 Sauvegarde et restauration . . . 211
10.2.1 Politique de sauvegarde et de restauration . . . 211
10.2.2 Suites d'utilitaires de sauvegarde . . . 212
10.2.3 Astuces de sauvegarde . . . 212
10.2.3.1 Sauvegarde depuis une interface graphique . . . 214
10.2.3.2 Sauvegarde déclenchée par des évènements de montage . . . 214
10.2.3.3 Sauvegarde déclenchée par des évènements d'horloge . . . 215
10.3 Infrastructure de sécurité des données . . . 216
10.3.1 Gestion de clés pour GnuPG . . . 216
10.3.2 Utilisation de GnuPG sur des fichiers . . . 218
10.3.3 Utiliser GnuPG avec Mutt . . . 218
10.3.4 Utiliser GnuPG avec Vim . . . 220
10.3.5 La somme de contrôle MD5 . . . 220
10.3.6 Trousseau de mots de passe . . . 220
10.4 Outils pour fusionner le code source . . . 220
10.4.1 Extraire des différences pour des fichiers sources . . . 222
10.4.2 Fusionner les mises à jour des fichiers source . . . 222
10.4.3 Fusion interactive . . . 222
10.5 Git . . . 222
10.5.1 Configuration du client Git . . . 222
10.5.2 Commandes de base pour Git . . . 223
10.5.3 Conseils pour Git . . . 224
10.5.4 Références de Git . . . 226
10.5.5 Autres systèmes de gestion de versions . . . 226

11 Conversion de données **227**
11.1 Outils de conversion de données textuelles . . . 227
11.1.1 Convertir un fichier texte avec iconv . . . 227
11.1.2 Vérifier que les fichiers sont codés en UTF-8 avec iconv . . . 229
11.1.3 Convertir les noms de fichiers avec iconv . . . 229
11.1.4 Convertir les fins de ligne (EOL) . . . 229
11.1.5 Convertir les tabulations (TAB) . . . 230
11.1.6 Éditeurs avec conversion automatique . . . 230
11.1.7 Extraire du texte brut . . . 231
11.1.8 Mettre en évidence et formater des données en texte brut . . . 232
11.2 Données XML . . . 232
11.2.1 Conseils de base pour XML . . . 232
11.2.2 Traitement XML . . . 233
11.2.3 Extraire des données XML . . . 234
11.2.4 Analyse statique (lint) de données XML . . . 234
11.3 Composition . . . 236
11.3.1 Composition roff . . . 236
11.3.2 TeX/LaTeX . . . 236
11.3.3 Imprimer convenablement une page de manuel . . . 237
11.3.4 Créer une page de manuel . . . 237
11.4 Données imprimables . . . 237
11.4.1 Ghostscript . . . 238
11.4.2 Fusionner deux fichiers PS ou PDF . . . 238
11.4.3 Utilitaires pour les données imprimables . . . 238
11.4.4 Imprimer avec CUPS . . . 240
11.5 La conversion de données de courrier électronique . . . 240
11.5.1 Bases concernant les données de courrier électronique . . . 240
11.6 Outils de données graphiques . . . 241
11.6.1 Outils de données graphiques (métapaquet) . . . 241
11.6.2 Outils de données graphiques (interface graphique) . . . 242
11.6.3 Outils de données graphiques (ligne de commande) . . . 242
11.7 Diverses conversions de données . . . 242

12 Programmation **245**
12.1 Les scripts de l'interpréteur de commande . . . 245
12.1.1 Compatibilité de l'interpréteur de commandes avec POSIX . . . 246
12.1.2 Paramètres de l'interpréteur de commandes . . . 246
12.1.3 Opérateurs conditionnels de l'interpréteur . . . 247
12.1.4 Boucles de l'interpréteur de commandes . . . 248

12.1.5 Variables d'environnement de l'interpréteur de commandes . . . 248
12.1.6 Séquence de traitement de la ligne de commandes de l'interpréteur . . . 249
12.1.7 Programmes utilitaires pour les scripts de l'interpréteur de commandes . . . 249
12.2 Scriptage avec des langages interprétés . . . 250
12.2.1 Débogage du code d'un langage interprété . . . 251
12.2.2 Programmes graphiques avec des scripts de d'interpréteur de commandes . . . 251
12.2.3 Actions personnalisées pour le gestionnaire de fichiers graphique . . . 252
12.2.4 Extravagances des scripts courts en Perl . . . 252
12.3 Codage dans les langages compilés . . . 252
12.3.1 C . . . 253
12.3.2 Programme simple en C (gcc) . . . 253
12.3.3 Flex -- un meilleur Lex . . . 254
12.3.4 Bison -- un meilleur Yacc . . . 254
12.4 Outils d'analyse du code statique . . . 256
12.5 Déboguer . . . 256
12.5.1 Exécution de base de gdb . . . 256
12.5.2 Déboguer un paquet Debian . . . 258
12.5.3 Obtenir une trace . . . 259
12.5.4 Commandes avancées de gdb . . . 259
12.5.5 Vérifier les dépendances avec les bibliothèques . . . 260
12.5.6 Outils de traçage dynamique des appels . . . 260
12.5.7 Déboguer les erreurs de X . . . 260
12.5.8 Outils de détection des fuites de mémoire . . . 260
12.5.9 Désassembler un binaire . . . 260
12.6 Outils de construction . . . 261
12.6.1 Make . . . 261
12.6.2 Autotools . . . 262
12.6.2.1 Compiler et installer un programme . . . 262
12.6.2.2 Désinstaller un programme . . . 262
12.6.3 Meson . . . 263
12.7 Web . . . 263
12.8 La conversion du code source . . . 263
12.9 Créer un paquet Debian . . . 263

A Annexe **265**

A.1 Le labyrinthe de Debian . . . 265
A.2 Historique du Copyright . . . 265
A.3 Format du document . . . 266

Liste des tableaux

1.1 Liste de paquets de programmes intéressants en mode texte . . . 5
1.2 Liste de paquets de documentation . . . 5
1.3 Utilisation des répertoires-clés . . . 8
1.4 Liste des premiers caractères de la sortie de « `ls -l` » : . . . 9
1.5 Mode numérique des permissions de fichiers dans les commandes chmod(1) . . . 10
1.6 Exemples de valeurs de **umask** . . . 11
1.7 Liste des groupes importants fournis par le système pour l'accès aux fichiers . . . 12
1.8 Liste des groupes importants fournis par le système pour l'exécution de commandes particulières . . . 12
1.9 Liste des types d'horodatage . . . 13
1.10 Liste des fichiers spéciaux de périphériques . . . 16
1.11 Touches de raccourcis de MC . . . 18
1.12 Réaction à la touche Entrée dans MC . . . 20
1.13 Liste d'interpréteurs de commandes (« shells ») . . . 21
1.14 Liste des raccourcis clavier de bash . . . 22
1.15 Liste des opérations de la souris et des actions associées sur les touches pour Debian . . . 23
1.16 Liste des saisies de clavier basiques pour vim . . . 25
1.17 Liste des commandes UNIX de base . . . 27
1.18 Les trois parties des paramètres linguistiques . . . 29
1.19 Liste des recommandations de paramètres linguistiques . . . 29
1.20 Afficher les valeurs de la variable « `$HOME` » . . . 30
1.21 Motifs génériques d'expansion du nom de fichier de l'interpréteur de commandes . . . 31
1.22 Codes de retour de la commande . . . 32
1.23 Idiomes des commandes de l'interpréteur . . . 33
1.24 Descripteurs de fichier prédéfinis . . . 33
1.25 Métacaractères pour BRE et ERE . . . 36
1.26 Expressions de remplacement . . . 37
1.27 Liste de parties de scripts pour enchaîner (piping) les commandes . . . 40

2.1 Liste des outils de gestion des paquets de Debian . . . 42
2.2 Liste des sites d'archive de Debian . . . 45

2.3 Liste des sections de l'archive de Debian . . . 46
2.4 Relation entre version et nom de code . . . 46
2.5 Liste de sites web clés pour résoudre les problèmes avec un paquet particulier . . . 51
2.6 Opérations de base de gestion des paquets avec la ligne de commandes en utilisant apt(8), aptitude(8), apt-get(8) et apt-cache(8) . . . 54
2.7 Options importantes de la commande aptitude(8) . . . 54
2.8 Liste des raccourcis clavier d'aptitude . . . 55
2.9 Liste des vues d'aptitude . . . 56
2.10 Classement par catégories des vues de paquets standard . . . 57
2.11 Liste des formules d'expressions rationnelles d'aptitude . . . 58
2.12 Fichiers journaux de l'activité des paquets . . . 59
2.13 Liste des opérations avancées de gestion des paquets . . . 63
2.14 Contenu des métadonnées de l'archive Debian . . . 65
2.15 Structure du nom des paquets Debian : . . . 68
2.16 Caractères utilisables pour chacune des composantes des noms de paquets Debian . . . 68
2.17 Fichiers particuliers créés par `dpkg` . . . 69
2.18 Liste de valeurs remarquables des priorités d'épinglage pour la technique de**apt-pinning** . . . 77
2.19 Liste des outils de proxy spécifiques à l'archive Debian . . . 82

3.1 Liste des chargeurs initiaux . . . 84
3.2 La signification de l'entrée de menu de la partie ci-dessus de `/boot/grub/grub.cfg` . . . 85
3.3 Liste d'utilitaires d'amorçage initial pour le système Debian : . . . 87
3.4 Liste des niveaux d'erreur du noyau . . . 91
3.5 Liste de bribes de commande utilisant `journalctl` . . . 91
3.6 Liste de bribes de commande typiques utilisant `journalctl` . . . 92
3.7 Liste d'autres bribes de commandes de surveillance sous `systemd` . . . 93

4.1 3 fichiers de configuration importants pour pam_unix(8) . . . 97
4.2 Contenu de la seconde entrée de « `/etc/passwd` » . . . 98
4.3 Liste des commandes servant à gérer les informations des comptes . . . 99
4.4 Liste d'outils permettant de générer des mots de passe . . . 100
4.5 Liste des paquets importants des systèmes PAM et NNS . . . 100
4.6 Liste des fichiers de configuration auxquels PAM et NSS accèdent . . . 101
4.7 Liste des services et ports sûrs et non sûrs . . . 103
4.8 Liste des outils fournissant des mesures de sécurité supplémentaires . . . 104

5.1 Liste des outils de configuration du réseau . . . 109
5.2 Liste des plages d'adresses de réseau . . . 111
5.3 Table de conversion depuis les commandes obsolètes `net-tools` vers les nouvelles commandes `iproute2` . . . 114
5.4 Liste des commandes de réseau de bas niveau . . . 115

5.5 Liste des outils d'optimisation du réseau . . . 115
5.6 Lignes directrices pour une valeur optimale de MTU . . . 116
5.7 Liste d'outils de pare-feu . . . 117

6.1 Liste de navigateurs web . . . 118
6.2 Liste d'agents de courrier électronique de l'utilisateur (MUA) . . . 120
6.3 Liste de paquets basiques concernant des agents de transport du courriel . . . 121
6.4 Liste des pages de manuel importantes de postfix . . . 123
6.5 Liste des fichiers de configuration liés aux adresses de courriel . . . 124
6.6 Liste des opérations de base du MTA . . . 125
6.7 Liste des serveurs et des utilitaires d'accès à distance . . . 126
6.8 Liste des fichiers de configuration de SSH . . . 126
6.9 Liste d'exemples de démarrage du client SSH . . . 127
6.10 Liste des clients SSH libres pour d'autres plateformes . . . 127
6.11 Liste des serveurs et utilitaires d'impression . . . 129
6.12 Liste d'autres serveurs d'applications réseau . . . 130
6.13 Liste de clients d'applications réseau . . . 131
6.14 Liste des RFC courantes . . . 132

7.1 Liste des environnements de bureau . . . 133
7.2 Liste de paquets d'infrastructure graphique notables . . . 135
7.3 Liste d'applications graphiques notables . . . 136
7.4 Liste de fontes notables TrueType et OpenType . . . 138
7.5 Liste d'environnements de fontes notables et de paquets connexes . . . 139
7.6 Liste des environnements bac à sable notables et des packages connexes . . . 140
7.7 Liste des serveurs et des utilitaires notables d'accès à distance . . . 141
7.8 Liste des méthodes de connexion au serveur X . . . 141
7.9 Liste de programmes en rapport avec la manipulation du presse-papiers « caractères » . . . 143

8.1 List of IBus and its plugin packages . . . 149
8.2 Liste des paquets Fcitx5 et de ses moteurs . . . 150

9.1 Liste des programmes de prise en charge d'activités avec une console . . . 153
9.2 Liste des raccourcis clavier de screen . . . 155
9.3 Informations sur l'initialisation de `vim` . . . 159
9.4 Liste des analyseurs de journaux système . . . 160
9.5 Exemples d'affichage pour la commande « `ls -l` » avec la **valeur de style pour l'heure** . . . 161
9.6 Liste des outils de manipulation d'images . . . 162
9.7 Liste des paquets pouvant enregistrer l'historique de configuration . . . 163
9.8 Liste des outils de surveillance et de contrôle de l'activité des programmes . . . 163

9.9 Liste des valeurs de politesse pour la priorité d'ordonnancement . . . 164
9.10 Liste des styles de la commande ps . . . 164
9.11 Liste des signaux couramment utilisés avec la commande kill . . . 169
9.12 Listes des touches notables de commande SAK (« Secure attention keys ») . . . 170
9.13 Listes des outils d'identification du matériel . . . 172
9.14 Liste des outils de configuration du matériel . . . 172
9.15 Liste des paquets son . . . 174
9.16 Liste des commandes pour désactiver l'économiseur d'écran . . . 175
9.17 Taille mémoire affichée . . . 175
9.18 Liste d'outils pour la vérification de la sécurité et de l'intégrité du système . . . 176
9.19 Listes de paquets de gestion de la partition du disque . . . 177
9.20 Liste des paquets de gestion des systèmes de fichiers . . . 179
9.21 Liste des paquets permettant de visualiser et d'éditer des données binaires . . . 188
9.22 List of packages to manipulate files without mounting the filesystems . . . 188
9.23 Liste d'outils pour ajouter des données de redondance aux fichiers . . . 189
9.24 Liste de paquets pour la récupération de données et l'analyse par autopsie . . . 189
9.25 Liste d'utilitaires de chiffrement des données . . . 193
9.26 Liste des paquets-clés à installer pour la compilation du noyau sur un système Debian . . . 195
9.27 Liste des outils de virtualisation . . . 199

10.1 Liste des outils d'archivage et de compression . . . 203
10.2 Liste des outils de copie et de synchronisation . . . 204
10.3 Liste de choix de systèmes de fichiers pour des périphériques amovibles avec des scénarios typiques d'utilisation . . . 209
10.4 Liste des services réseau à choisir avec le scénario typique d'utilisation . . . 210
10.5 Liste de suites d'utilitaires de sauvegarde . . . 213
10.6 Liste des outils d'une infrastructure de sécurité des données . . . 216
10.7 Liste des commandes de GNU Privacy Guard pour la gestion des clés . . . 217
10.8 Liste de la signification des codes de confiance . . . 217
10.9 Liste des commandes de GNU Privacy Guard sur des fichiers . . . 219
10.10 Liste d'outils destinés à fusionner du code source . . . 221
10.11 Liste des paquets et des commandes relatifs à git . . . 223
10.12 Principales commandes de Git . . . 224
10.13 Conseils pour Git . . . 225
10.14 Liste des autres outils de système de gestion de versions . . . 226

11.1 Liste des outils de conversion de texte . . . 227
11.2 Liste de valeurs de codage et leur utilisation . . . 228
11.3 Liste des styles d'EOL pour différentes plateformes . . . 230
11.4 Liste des commande de conversion de TAB des paquets `bsdmainutils` et `coreutils` . . . 230

11.5 Liste d'outils pour extraite des données en texte brut . . . 231
11.6 Liste des outils pour mettre en évidence des données de texte brut . . . 232
11.7 Liste des entités XML prédéfinies . . . 233
11.8 Liste d'outils XML . . . 234
11.9 Liste des outils DSSSL . . . 234
11.10 Liste d'outils d'extraction de données XML . . . 235
11.11 Liste d'outils d'impression élégante du XML . . . 235
11.12 Liste des outils de typographie . . . 236
11.13 Liste de paquets facilitant la création de pages de manuel . . . 237
11.14 Liste des interpréteurs Ghostscript PostScript . . . 238
11.15 Liste des utilitaires pour les données imprimables . . . 239
11.16 Liste de paquets facilitant la conversion de données de courrier électronique . . . 240
11.17 Liste d'outils pour les données graphiques (métapaquet) . . . 241
11.18 Liste d'outils pour les données graphiques (interfaces graphiques) . . . 242
11.19 Liste d'outils pour les données graphiques (ligne de commande) . . . 243
11.20 Liste d'outils divers de conversion de données . . . 244

12.1 Liste de bashismes typiques . . . 246
12.2 Liste des paramètres de l'interpréteur de commandes . . . 246
12.3 Liste des expansions de paramètre de l'interpréteur . . . 247
12.4 Liste des substitutions-clés de paramètres de l'interpréteur . . . 247
12.5 Liste des opérateurs de comparaison dans les expressions conditionnelles . . . 248
12.6 Liste des opérateurs de comparaison de chaîne de caractères dans les expressions conditionnelles . . . 248
12.7 Lites des paquets comportant des petits programmes utilitaires pour les scripts de l'interpréteur de commandes . . . 250
12.8 Liste des paquets relatifs aux interpréteurs de commandes . . . 250
12.9 Liste des programmes de dialogue . . . 251
12.10 Liste des paquets relatifs à un compilateur . . . 253
12.11 Liste de générateurs d'analyseur LALR compatible avec Yacc . . . 254
12.12 Liste des outils d'analyse du code statique : . . . 257
12.13 Liste des paquets de débogage . . . 257
12.14 Liste des commandes avancées de gdb . . . 259
12.15 Liste des outils de détection des fuites de mémoire . . . 260
12.16 Liste des paquets d'outil de construction . . . 261
12.17 Liste des variables automatiques de make . . . 261
12.18 Liste de l'expansion des variables de make . . . 262
12.19 Liste des outils de conversion de code source . . . 264

Résumé

Ce livre est libre ; vous pouvez le redistribuer et le modifier selon les termes de la Licence Publique Générale GNU (« GNU GPL ») avec n'importe quelle version compatible avec les Règles des Logiciels Libres selon Debian (DFSG).

Préface

Cette Debian Reference (version 2.153) (2026-11-29 05:38:00 UTC) est destinée à fournir un large aperçu de l'administration d'un système Debian, sous la forme d'un guide utilisateur de post-installation.

Le lecteur cible est quelqu'un qui désire apprendre les scripts de l'interpréteur de commandes mais qui ne souhaite pas lire tous les sources en C pour comprendre le fonctionnement du système GNU /Linux.

Pour le guide d'installation, voir :

— Debian GNU/Linux : guide d'installation pour le système stable
— Debian GNU/Linux : guide d'installation pour la version en cours de test

Clause de non responsabilité

Toute garantie est rejetée. Toutes les marques déposées sont la propriété de leurs détenteurs respectifs.

Le système Debian lui-même est une cible mouvante. Cela rend difficile le maintien à jour et l'exactitude de sa documentation. Bien que la version instable `testing` du système Debian ait été utilisée pour écrire ce document, certaines parties peuvent être dépassées au moment où vous lisez cela.

Veuillez prendre ce document comme une référence secondaire. Ce document ne remplace aucun des guides autorisés. L'auteur et les contributeurs ne pourront être tenus pour responsables des conséquences des erreurs, omissions ou ambiguïtés que comporte ce document.

Ce qu'est Debian

Le Projet Debian est une association de personnes qui ont fait cause commune afin de créer un système d'exploitation libre. Sa distribution est caractérisée par :

— un engagement dans la liberté du logiciel : Le contrat social Debian et les Lignes directrices du logiciel libre selon Debian (DFSG) ;
— travaux partagés et non rémunérés par des volontaires, basés sur Internet :https://www.debian.org ;
— grand nombre de logiciels pré-compilés de haute qualité ;
— l'accent mis sur la stabilité et la sécurité avec un accès facile aux mises à jour de sécurité ;
— l'accent mis sur une mise à niveau en douceur vers les dernières versions des logiciels existants dans les archives de `testing` ;
— la prise en charge d'un grand nombre d'architectures matérielles.

Les éléments des logiciels libres de Debian proviennent de GNU, Linux, BSD, X, ISC, Apache, Ghostscript, Common Unix Printing System , Samba, GNOME, KDE, Mozilla, LibreOffice, Vim, TeX, LaTeX, DocBook, Perl, Python, Tcl, Java, Ruby, PHP, Berkeley DB, MariaDB, PostgreSQL, SQLite, Exim, Postfix, Mutt, FreeBSD, OpenBSD, Plan 9 et de nombreux autres projets de logiciels libres indépendants. Debian intègre cette diversité de logiciels libres dans un seul système.

À propos de ce document

Règles

Les règles suivantes ont été suivies lors de la compilation de ce document.

— fournir un aperçu et passer les cas marginaux (**vue d'ensemble**) ;
— le garder court et simple (**KISS**) ;
— ne pas réinventer la roue (utiliser des liens pointant vers **les références existantes**) ;
— mettre l'accent sur les outils n'ayant pas d'interface graphique ou en mode console (utiliser **des exemples en ligne de commande**) ;
— Soyez objectif (Utilisez popcon, etc.)

ASTUCE
J'ai essayé d'éclaircir les aspects hiérarchiques et les niveaux les plus bas du système.

Exigences de départ



AVERTISSEMENT
On attend de vous que vous fassiez des efforts pour rechercher des réponses par vous-même au-delà de cette documentation. Ce document ne donne que des points de départs efficaces.

Vous devez chercher vous-même une solution dans les sources primaires.

— Le site de Debian https://www.debian.org pour des informations générales
— La documentation dans le répertoire « `/usr/share/doc/`*`nom_paquet`* »
— Les **pages de manuel (manpage)** de style UNIX : « `dpkg -L` *`nom_paquet`* `|grep '/man/man.*/'` »
— Les **pages info** de style GNU : « `dpkg -L` *`nom_paquet`* `|grep '/info/'` »
— Les signalements de bogues : https://bugs.debian.org/*nom_paquet*
— Le Wiki Debian en https://wiki.debian.org/ pour les sujets spécifiques ou changeants
— « Single UNIX Specification » depuis la page d'accueil du « Système UNIX » de l'Open Group
— L'encyclopédie libre de Wikipedia à http://www.wikipedia.org/
— Le cahier de l'administrateur Debian
— Les HOWTO du projet de documentation Linux (TLDP)

Note
Pour accéder à une documentation détaillée, vous devrez installer les paquets de documentation qui correspondent au nom du paquet avec le suffixe « `-doc` ».

Conventions

Ce document fournit des informations en utilisant le style de présentation simplifié suivant, avec des exemples de commandes de l'interpréteur `bash`(1).

```
# command-in-root-account
$ command-in-user-account
```

Ces invites de l'interpréteur de commandes permettent de distinguer le compte utilisé et correspondent à la définition des variables d'environnement « `PS1='\$'` » et « `PS2=' '` ». Ces valeurs ont été choisies pour ce document dans un but de lisibilité, elles ne sont pas représentatives d'un système réel.

Tous les exemples de commande sont exécutées avec la locale anglaise "`LANG=en_US.UTF8`". N'espérez pas que les chaines de substitution telles que *`commande-avec-compte-superutilisateur`* et *`commande-avec-compte-utilis`* soient traduites dans les exemples de commandes. Cela est un choix voulu pour conserver tous les exemples traduits à jour.

Note
Consultez la signification des variables d'environnement « `$PS1` » et « `$PS2` » dans bash(1).

L'**action** demandée à l'administrateur du système est écrite sous forme d'une phrase impérative, par exemple « Pressez la touche Entrée après la saisie de chaque chaîne de commande dans l'interpréteur de commandes. »

La colonne de **description** ou similaire dans le tableau peut contenir une **locution nominale** selon la convention de description courte du paquet qui supprime les articles se trouvant en tête tels que « un » et « le » (« a », « the »). Elle peut contenir une phrase à l'infinitif comme **locution nominale** sans le « to » de tête (NdT : en français, une phrase impérative commençant par un verbe à l'infinitif), suivie de la description courte de la commande selon la convention des pages de manuel. Cela peut sembler bizarre à certaines personnes mais ce sont les choix voulus par l'auteur afin de garder cette documentation la plus simple possible. Ces **locutions nominales**, selon cette convention de description courte, n'ont pas de majuscule à la première lettre et ne se terminent pas par un point.

Note
Les noms propres, y compris les noms de commandes, gardent leur casse indépendamment de l'endroit où ils se trouvent.

Un **morceau de commande** cité dans le paragraphe d'un texte sera signalé par une fonte « typewriter » (machine à écrire) entre guillemets, comme par exemple « `aptitude safe-upgrade` ».

Les **données textuelles** d'un fichier de configuration citées dans un paragraphe seront signalées par une fonte de type machine à écrire entre guillemets, comme par exemple « `deb-src` ».

Une **commande** sera indiquée par son nom dans une fonte machine à écrire suivi, de manière facultative, par le numéro de section de la page de manuel entre parenthèses, comme par exemple bash(1). Vous êtes encouragé à rechercher des informations complémentaires en entrant :

```
$ man 1 bash
```

Une **page de manuel** est indiquée par son nom dans une fonte machine à écrire suivi, entre parenthèses, du numéro de la section de la page de manuel, comme par exemple, `sources.list` (5). Vous êtes encouragé à rechercher des informations complémentaires en entrant :

```
$ man 5 sources.list
```

Une **page info** est indiquée par un fragment entre guillemets de la commande correspondante dans une fonte machine à écrire, comme par exemple, « `info make` ». Vous êtes encouragé à rechercher des informations complémentaires en entrant :

```
$ info make
```

Un **nom de fichier** est indiqué par une fonte machine à écrire entre guillemets, comme par exemple, « `/etc/passwd` ». En ce qui concerne les fichiers de configuration, vous êtes encouragé à rechercher des informations complémentaires en entrant :

```
$ sensible-pager "/etc/passwd"
```

Un **nom de répertoire** est indiqué par une fonte machine à écrire entre guillemets, comme par exemple, « `/etc/apt/` ». Vous êtes encouragé à explorer son contenu en tapant ce qui suit :

```
$ mc "/etc/apt/"
```

Un **nom de paquet** est indiqué par son nom dans une fonte machine à écrire, comme par exemple `vim`. Vous êtes encouragé à rechercher des informations complémentaires en entrant :

```
$ dpkg -L vim
$ apt-cache show vim
$ aptitude show vim
```

On peut indiquer l'emplacement d'une **documentation** par son nom de fichier dans une fonte machine à écrire entre guillemets, comme par exemple « `/usr/share/doc/sysv-rc/README.runlevels.gz` » et « `/usr/share/doc/base` » ou par son URL, comme par exemple https://www.debian.org. Vous êtes encouragé à lire la documentation en entrant :

```
$ zcat "/usr/share/doc/base-passwd/users-and-groups.txt.gz" | sensible-pager
$ sensible-browser "/usr/share/doc/base-passwd/users-and-groups.html"
$ sensible-browser "https://www.debian.org"
```

Une **variable d'environnement** est indiquée par son nom précédé d'un « $ » dans une fonte machine à écrire entre guillemets, comme par exemple « `$TERM` ». Vous êtes encouragé à obtenir sa valeur actuelle en entrant :

```
$ echo "$TERM"
```

Le concours de popularité (« popcon »)

Les données du popcon sont présentées comme une manière objective de mesurer la popularité de chaque paquet. Elles sont téléchargées depuis 2026-11-28 09:48:51 UTC et contiennent un total de 299780 soumissions de rapports pour 229395 paquets binaires et 26 architectures.

Note
Vous remarquerez que l'archive `amd64` `unstable` ne contient actuellement que 78709 paquets. Les données de popularité contiennent des rapports venant de nombreuses installations anciennes.

Le numéro de « popcon », précédé par un « V: » pour « votes », est calculé par « 1000 * (soumissions popcon pour le paquet exécuté récemment sur le PC)/(nombre total des soumissions popcon) ».

Le numéro de « popcon », précédé par un « I: » pour « installations », est calculé par « 1000 * (soumissions popcon pour le paquet installé sur le PC)/(nombre total des soumissions popcon) ».

Note
Les données numériques de popcon ne doivent pas être considérées comme des mesures absolues de l'importance des paquets. Il y a de nombreux facteurs qui peuvent fausser les statistiques. Par exemple, certains systèmes participant au popcon ont monté des répertoires tels que « `/usr/bin` » avec l'option « `noatime` » afin d'améliorer les performances du système et ont, de ce fait, désactivé le « vote » de tels systèmes.

Taille du paquet

Les données correspondant à la taille du paquet sont aussi présentées comme une mesure objective de chacun des paquets. Elles sont basées sur « `Installed-Size:` » (« taille installée ») indiquée par la commande « `apt-cache show` » ou « `aptitude show` » (actuellement sur l'architecture `amd64` et pour la version `unstable`). La taille est indiquée en Kio (kibioctet = unité pour 1024 octets).

Note
Si un paquet a une taille de faible valeur numérique, cela peut vouloir dire que le paquet de la version `unstable` est un paquet « dummy » qui permet l'installation par dépendances d'autres paquets ayant un contenu significatif. Un paquet dummy permet une transition en douceur ou un éclatement du paquet.

Note
Une taille de paquet suivie de « (*) » indique que la version `unstable` du paquet est absente et que la taille du paquet venant de la version `experimental` a été utilisée en remplacement.

Signalements de bogues concernant ce document

Si vous découvrez des problèmes dans ce document, veuillez signaler les bogues du paquet `debian-reference` en utilisant reportbug(1). Veuillez inclure des suggestions de correction en lançant « `diff -u` » sur la version en texte brut ou sur le source.

Rappels pour les nouveaux utilisateurs

Voici quelques rappels pour les nouveaux utilisateurs :

- Sauvegardez vos données
  - Consultez Section 10.2.
- Sécurisez votre mot de passe et vos clés de sécurité
- KISS (keep it simple stupid)
  - Ne modifiez pas trop votre système
- Lire vos fichiers journaux
  - La **PREMIÈRE** erreur est celle qui compte
- LLSM (Lisez Le Super Manuel)
- Faîtes une recherche sur Internet avant de poser des questions
- Ne soyez pas administrateur (root) quand vous n'avez pas à l'être.
- Ne modifiez pas le système de gestion des paquets
- Ne tapez rien que vous ne compreniez pas.
- Ne modifiez pas les permissions des fichiers (avant un examen complet de la sécurité)
- Ne pas quitter l'interpréteur root avant de **TESTER** les modifications.
- Always have an alternative boot media (USB flash drive, CD-ROM, …)

Quelques citations pour les nouveaux utilisateurs

Voici quelques citations intéressantes provenant de la liste de diffusion Debian qui pourraient aider les nouveaux utilisateurs à y voir plus clair.

- « This is UNIX. It gives you enough rope to hang yourself. » --- Miquel van Smoorenburg <`miquels@cistron.nl`> (C'est UNIX. Il vous donne assez de corde pour vous pendre vous-même).
- « UNIX IS user friendly... It's just selective about who its friends are. » --- Tollef Fog Heen <`tollef@add.no`> (UNIX **est** l'ami de l'utilisateur… Il choisit juste qui sont ses amis).

L'article Wikipédia « Philosophie d'Unix » contient une liste de citations intéressantes.

Chapitre 1

Didacticiels GNU/Linux

Je pense qu'apprendre un système d'exploitation est comme apprendre une nouvelle langue étrangère. Bien que les livres de didacticiels et de documentation soient utiles, vous devrez pratiquer vous-même. Pour vous aider à vous lancer en douceur, je vais développer quelques points fondamentaux.

La puissance de la conception de Debian GNU/Linux vient du système d'exploitation UNIX, c'est-à-dire un système d'exploitation multi-utilisateurs, multi-tâches. Vous devrez apprendre à tirer parti de la puissance de ces fonctionnalités et des similitudes entre UNIX et GNU/Linux.

N'écartez pas des textes orientés UNIX en ne vous reposant que sur les textes GNU/Linux, cela vous priverait de beaucoup d'informations utiles.

Note
Si vous avez utilisé pendant un certain temps des systèmes ressemblant à UNIX avec des outils en ligne de commande, vous connaissez sans doute tout ce que j'explique ici. Vous pourrez utiliser ce document pour actualiser vos connaissances.

1.1 Bases pour la console

1.1.1 L'invite de l'interpréteur de commandes

Au démarrage du système, si vous n'avez pas installé d'environnement graphique (GUI) tel que avec un gestionnaire d'affichage tel que les systèmes de bureau GNOME ou KDE, un écran d'identification en mode caractères vous est présenté. Supposons que le nom d'hôte de votre machine soit `toto`, l'invite d'identification ressemblera alors à ce qui suit.

Si vous avez installé un environnement graphique (GUI), vous pouvez toujours obtenir une telle invite d'identification avec Ctrl-Alt-F3 et vous pourrez ensuite revenir à l'environnement graphique avec Ctrl-Alt-F2 (consultez Section 1.1.6 ci-dessous pour davantage d'informations).

```
foo login:
```

À l'invite d'identification, entrez votre nom d'utilisateur, par exemple `pingouin`, et pressez la touche Entrée, entrez ensuite votre mot de passe et pressez de nouveau la touche Entrée.

Note
Conformément à la tradition UNIX, l'identifiant de l'utilisateur et le mot de passe sur un système Debian sont sensibles à la casse. L'identifiant de l'utilisateur est habituellement choisi uniquement en minuscules. Le premier compte d'utilisateur est normalement créé lors de l'installation. Des comptes d'utilisateurs supplémentaires peuvent être créés avec la commande adduser(8) par l'administrateur (root).

Le système démarre avec le message de bienvenue qui se trouve dans « `/etc/motd` » (Message du jour : « Message Of The Day ») et présente une invite de commande comme :

```
Debian GNU/Linux 12 foo tty3

foo login: penguin
Password:

Linux foo 6.5.0-0.deb12.4-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.5.10-1~bpo12+1 (2023-11-23) ↩
     x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Last login: Wed Dec 20 09:39:00 JST 2023 on tty3
foo:~$
```

Vous êtes maintenant sous l'interpréteur de commandes (« shell »). Le shell interprète vos commandes.

1.1.2 Invite de l'interpréteur de commandes avec une interface graphique

Si, lors de l'installation, vous avez installé avec un environnement graphique (GUI), un écran d'identification graphique vous sera présenté au démarrage de votre système. Entrez votre nom d'utilisateur et votre mot de passe pour vous connecter à un compte d'utilisateur sans privilège. Utilisez la touche de tabulation pour passer du champ d'entrée de l'utilisateur à celui du mot de passe ou utilisez la souris et un clic-gauche.

Sous environnement graphique, vous pouvez obtenir une invite de l'interpréteur de commandes en lançant un programme d'émulation de terminal-x comme gnome-terminal(1), konsole(1) ou xterm(1). Sous l'environnement de bureau GNOME, presser la touche SUPER (touche Windows) et saisir « terminal » pour l'invite de recherche fait l'affaire.

Sous d'autres environnements de bureau (comme `fluxbox`), il peut ne pas y avoir de point d'accès évident au menu. Si cela se produit, essayez simplement de faire un clic-droit sur le fond d'écran de l'environnement de bureau en espérant voir apparaître un menu.

1.1.3 Compte de l'administrateur (root)

Le compte de l'administrateur (root) est encore appelé superutilisateur ou utilisateur privilégié. Depuis ce compte, vous pouvez effectuer les opérations d'administration du système suivantes :

— lire, écrire et effacer n'importe quel fichier du système quelles que soient ses permissions ;
— Défini l'appartenance et les permissions de n'importe quel fichier du système
— définir le mot de passe de n'importe quel utilisateur non-privilégié du système ;
— vous connecter à n'importe quel compte sans mot de passe.

La puissance illimitée du compte de l'administrateur fait que vous devez être attentif et responsable lorsque vous l'utilisez.



AVERTISSEMENT
Ne donnez jamais le mot de passe de l'administrateur à d'autres personnes.

Note
Les permissions d'un fichier (y compris les fichiers de périphériques tels que les CD-ROM ou autres, qui ne sont que des fichiers parmi d'autres pour le système Debian) peuvent le rendre inutilisable ou inaccessible à des utilisateurs autres que l'administrateur. Bien que l'utilisation du compte de l'administrateur soit un moyen rapide de test dans une telle situation, sa résolution devra être effectuée en définissant correctement les droits de fichiers et les membres des groupes d'utilisateurs (consultez Section 1.2.3).

1.1.4 Invite de l'interpréteur de commandes pour l'administrateur

Voici quelques méthodes de base pour obtenir l'invite de l'interpréteur de commande de l'administrateur en utilisant le mot de passe de « root » .

- entrez `root` à l'invite de connexion en mode caractère ;
- entrez « `su -l` » depuis une invite quelconque de l'interpréteur de commandes ;
  - cela ne préserve pas l'environnement de l'utilisateur actuel ;
- entrez « `su` » depuis une invite quelconque de l'interpréteur de commandes ;
  - cela préserve une partie de l'environnement de l'utilisateur actuel.

1.1.5 Outils graphiques d'administration du système

Lorsque le menu de votre environnement de bureau ne démarre pas automatiquement, avec les droits appropriés, les outils graphiques d'administration du système, vous pouvez les démarrer depuis l'invite de l'interpréteur du compte administrateur dans un émulateur de terminal tel que gnome-terminal(1), konsole(1) ou xterm(1). Consultez Section 1.1.4 et Section 7.9.



AVERTISSEMENT
Ne jamais lancer un affichage graphique ou un gestionnaire de session graphique pour le compte administrateur en entrant `root` à l'invite d'un gestionnaire d'écran tel que gdm3(1).
Ne jamais faire tourner de programme distant non sûrs avec une interface graphique sous X Window lorsque des informations critiques sont affichées parce que votre écran X peut être espionné.

1.1.6 Consoles virtuelles

Il y a, dans le système Debian par défaut, six consoles commutables en mode caractères de type VT100 disponibles pour lancer directement l'interpréteur de commandes sur la machine Linux. À moins que vous ne disposiez d'un environnement graphique, vous pouvez basculer entre les consoles virtuelles en pressant la `touche Alt de gauche` et, simultanément, l'une des touches `F1` à `F6`. Chaque console en mode caractères permet de se connecter de manière indépendante à un compte et offre un environnement multi-utilisateurs. Cet environnement multi-utilisateurs est une fonctionnalité intéressante d'UNIX, très vite, on ne peut plus s'en passer.

Si vous vous trouvez dans l'environnement graphique, vous accédez à la console caractères 3 en appuyant sur la touche `Ctrl-Alt-F3`, c'est-à-dire en appuyant simultanément sur la touche `Ctrl gauche`, sur la touche `Alt gauche` et sur la touche `F3`. Vous pouvez revenir à l'environnement graphique qui fonctionne normalement sur la console virtuelle 2, en appuyant sur `Alt-F2`.

Vous pouvez aussi changer de console virtuelle, par exemple pour la console 3, à partir de la ligne de commande.

```
# chvt 3
```

1.1.7 Comment quitter l'invite de l'interpréteur de commandes

Pour arrêter l'activité de l'interpréteur de commandes, entrez `Ctrl-D`, c'est-à-dire la `touche Ctrl de gauche` et la `touche d` pressées simultanément, à l'invite de l'interpréteur. Si vous êtes sur une console en mode caractères, cela vous fera retourner alors à l'invite d'identification. Même si on se réfère à ces caractères de commande avec « control D » en majuscule, il n'est pas nécessaire de presser la touche majuscule. Le raccourci `^D`, est aussi utilisé pour `Ctrl-D`. Vous pouvez aussi entrer « exit ».

Si vous êtes sur un `émulateur de x-terminal`(1), vous pouvez fermer la fenêtre de `l'émulateur de terminal-x` de la même manière.

1.1.8 Comment arrêter le système

Comme tout autre système d'exploitation moderne où les opérations sur les fichiers mettent en œuvre un cache de données en mémoire afin d'améliorer les performances, le système Debian demande que l'on exécute une procédure d'arrêt adaptée avant que le système ne puisse être mis hors tension. Cela permet de maintenir l'intégrité des fichiers, en forçant l'écriture sur disque de toutes les informations conservées en mémoire. Si un logiciel de contrôle de l'alimentation est disponible, la procédure d'arrêt coupera automatiquement l'alimentation du système. (Sinon, vous devrez presser le bouton d'alimentation pendant quelques secondes une fois la procédure d'arrêt terminée).

Dans le mode normal multi-utilisateurs, vous pouvez arrêter le système depuis la ligne de commandes :

```
# shutdown -h now
```

Dans le mode normal mono-utilisateur, vous pouvez arrêter le système depuis la ligne de commandes :

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

Consultez Section 6.3.8.

1.1.9 Récupérer une console propre

Lorsque l'écran est corrompu après que vous ayez fait des choses amusantes comme « `cat` *`un-fichier-binaire`* » entrez « `reset` » à l'invite de commande. Il est possible que vous ne voyiez pas ce que vous entrez. Vous pouvez aussi entrer « `clear` » pour nettoyer l'écran.

1.1.10 Suggestions de paquets supplémentaires pour le débutant

Bien que même une installation minimale du système Debian sans aucune tâche d'environnement de bureau fournisse les fonctionnalités UNIX de base, c'est une bonne idée pour les débutants d'installer pour commencer quelques paquets de ligne de commandes ou de terminal en mode caractères basé sur « curses » comme `mc` et `vim` avec apt-get(8) en utilisant les commandes suivantes :

```
# apt-get update
 ...
# apt-get install mc vim sudo aptitude
 ...
```

Si vous avez déjà installé ces paquets, aucun nouveau paquet ne sera installé.

Ce peut être une bonne idée de lire quelques documentations.

Vous pouvez installer quelques-uns de ces paquets en passant les commandes suivantes :

```
# apt-get install package_name
```

paquet	popularité	taille	description
mc	V:42, I:172	1590	gestionnaire de fichiers plein écran en mode texte
sudo	V:739, I:870	6817	programme donnant aux utilisateurs des privilèges d'administration limités
vim	V:73, I:330	4415	éditeur de texte UNIX Vi amélioré (Vi IMproved), éditeur de texte pour programmeurs (version standard)
vim-tiny	V:55, I:979	1931	éditeur de texte UNIX Vi amélioré (Vi IMproved), éditeur de texte pour programmeurs (version compacte)
emacs-nox	V:3, I:12	46564	GNU Emacs, éditeur de texte extensible basé sur Lisp
w3m	V:11, I:125	2853	navigateurs WWW en mode texte
gpm	V:8.0, I:8.7	524	couper-coller à la mode UNIX sur une console texte (démon)

Table 1.1 – Liste de paquets de programmes intéressants en mode texte

paquet	popularité	taille	description
doc-debian	I:887	185	projet de documentation Debian, (FAQ Debian) et autres documents
debian-policy	I:6.4	5147	Charte Debian et documents associés
developers-reference	V:0.2, I:2.0	2656	Guides et informations pour les responsables Debian
debmake-doc	I:0.32	11807	Guide des nouveaux responsables Debian
debian-history	I:0.49	6513	Histoire du projet Debian
debian-faq	I:886	791	FAQ Debian

Table 1.2 – Liste de paquets de documentation

1.1.11 Compte pour un utilisateur supplémentaire

Si vous ne souhaitez pas utiliser votre compte d'utilisateur principal pour les activités de formation qui suivent, vous pouvez créer un compte de formation, par exemple `fish` en passant la commande qui suit :

```
# adduser fish
```

Répondez à toutes les questions.

Cela va créer un nouveau compte appelé `fish`. Après vos exercices, vous pourrez supprimer ce compte d'utilisateur et son répertoire personnel par :

```
# deluser --remove-home fish
```

Sur les systèmes non-Debian et les systèmes Debian spécialisés, les activités ci-dessus ont besoin à la place d'utiliser les utilitaires useradd(8) et userdel(8) de bas niveau.

1.1.12 Configuration de sudo

Pour une station de travail typique avec un seul utilisateur tel que le système de bureau Debian sur un PC de bureau, il est habituel de mettre en œuvre une configuration simple de sudo(8) comme suit afin que l'utilisateur non privilégié, par exemple `pingouin`, puisse obtenir les privilèges d'administration avec simplement son mot de passe personnel et non avec le mot de passe de l'administrateur :

```
# echo "penguin  ALL=(ALL) ALL" >> /etc/sudoers
```

Il est aussi habituel, en remplacement, de faire comme suit afin que l'utilisateur non privilégié, par exemple `pingouin`, puisse obtenir les privilèges d'administration sans aucun mot de passe personnel.

```
# echo "penguin  ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers
```

Cette astuce ne doit être utilisée qu'avec une station de travail mono-utilisateur que vous administrez et dont vous êtes le seul utilisateur.



AVERTISSEMENT
Ne configurez pas de cette manière les utilisateurs normaux d'une station de travail multi-utilisateurs parce que cela serait très grave pour la sécurité du système.



Attention
Le mot de passe et le compte du `pingouin` de l'exemple précédent doivent bénéficier de la même protection que le mot de passe et le compte de l'administrateur.
Le privilège d'administration, dans ce contexte, appartient à quelqu'un autorisé à effectuer les tâches d'administration du système sur la station de travail. Ne jamais donner un tel privilège à un responsable du département administratif de votre entreprise ni à votre patron, à moins qu'ils n'y soient autorisés et en soient capables.

Note
Pour donner un accès privilégié à certains périphériques et certains fichiers, vous devriez envisager l'utilisation d'un **groupe** donnant un accès limité plutôt que d'utiliser le privilège de `root` par l'intermédiaire de sudo(8).
Avec une configuration plus approfondie et prudente, sudo(8) peut permettre à d'autres utilisateurs d'obtenir des privilèges limités d'administration sur un système partagé sans partager le mot de passe de l'administrateur. Cela peut améliorer la confiance sur les machines ayant plusieurs administrateurs de manière à ce que vous puissiez dire qui a fait quoi. D'un autre côté, vous ne devriez pas confier de tels privilèges à quelqu'un d'autre.

1.1.13 À vous de jouer

Vous êtres maintenant prêt à jouer avec le système Debian, sans risque aussi longtemps que vous utiliserez un compte d'utilisateur sans privilège.

Nous allons apprendre le système Debian en tant que système semblable à UNIX (« UNIX-like ») avec :
— Section 1.2 (concept de base) ;
— Section 1.3 (méthode de survie) ;
— Section 1.4 (méthode de base) ;
— Section 1.5 (mécanisme de l'interpréteur de commandes) ;
— Section 1.6 (méthode de traitement des données textuelles).

1.2 Système de fichiers de type UNIX

Avec GNU/Linux et d'autres systèmes d'exploitation semblables à UNIX, les fichiers sont organisés en répertoires. Tous les fichiers et les répertoires sont disposés sous forme d'une grosse arborescence ancrée sur « `/` ». On l'appelle un arbre parce que si vous dessinez le système de fichiers, il ressemble à un arbre qui se trouverait disposé la tête en bas.

Ces fichiers et répertoires peuvent être répartis sur plusieurs périphériques. La commande mount(8) sert à attacher les systèmes de fichiers se trouvant sur certains périphériques à la grosse arborescence des fichiers. À l'opposé, la commande umount(8) les détachera de nouveau. Avec les noyaux Linux récents, mount(8) avec certaines options peut lier une partie d'une arborescence de fichiers à un autre emplacement ou peut monter un système de fichiers de manière partagée, privée, esclave ou « non-liable ». Vous trouverez les options de montage prises en compte par chaque système de fichiers dans « `/usr/share/doc/linux-doc-*/Documentation/filesystems/` ».

Les **répertoires** d'un système UNIX sont appelés **dossiers** sur d'autres systèmes. Vous remarquez aussi qu'il n'y a, sur aucun système UNIX, de concept de **lecteur** tel que « `A:` ». Il y a un système de fichiers qui comprend tout. C'est un gros avantage comparé à Windows.

1.2.1 Bases concernant les fichiers UNIX

Voici les bases des fichiers UNIX :

— les noms de fichiers sont **sensibles à la casse**. Ce qui veut dire que « `MONFICHIER` » et « `MonFichier` » sont des fichiers différents ;

— on se réfère au **répertoire racine** (« root directory »), qui est la racine du système de fichiers, simplement par « `/` ». Ne pas le confondre avec le répertoire personnel de l'utilisateur root : « `/root` » ;

— un nom de répertoire peut être constitué de n'importe quelle lettre ou symbole **sauf « / »**. Le répertoire racine est une exception, son nom est « `/` » (prononcé « slash » ou « le répertoire racine »), il ne peut pas être renommé ;

— chaque fichier ou répertoire est désigné par un **nom de fichier entièrement qualifié**, **nom de fichier absolu** ou **chemin**, indiquant la séquence de répertoires que l'on doit traverser pour l'atteindre. Les trois expressions sont synonymes ;

— tous les **noms de fichiers entièrement qualifiés** commencent par le répertoire « `/` » et il y a un `/` » entre chaque répertoire ou fichier dans le nom du fichier. Le premier « `/` » est le répertoire de plus haut niveau, et les autres « `/` » séparent les sous-répertoires successifs jusqu'à ce que l'on atteigne la dernière entrée qui est le nom fichier proprement dit. Les mots utilisés ici peuvent être source de confusion. Prenez comme exemple le **nom pleinement qualifié** suivant : « `/usr/share/keytables/us.map.gz` ». Cependant, les gens utiliseront souvent son nom de base « `us.map.gz` » seul comme nom de fichier ;

— le répertoire racine comporte de nombreuses branches, telles que « `/etc/` » et « `/usr/` ». Ces sous-répertoires se décomposent eux-mêmes en d'autres sous-répertoires comme « `/etc/systemd/` » et « `/usr/local/` ». L'ensemble de la chose, vu globalement, s'appelle l'**arborescence des répertoires**. Vous pouvez imaginer un nom de fichier absolu comme une route partant de la base de l'arbre (« `/` ») jusqu'à l'extrémité de certaines branches (le fichier). Vous entendrez aussi certains parler de l'arborescence des répertoires comme d'un arbre **généalogique** englobant tous les descendants directs d'un seul personnage appelé le répertoire racine (« `/` ») : les sous-répertoires ont alors des **parents** et un chemin montre l'ascendance complète d'un fichier. Il y a aussi des chemins relatifs qui commencent quelque part ailleurs qu'au niveau du répertoire racine. Il faut vous souvenir que le répertoire « `../` » indique le répertoire parent. Cette terminologie s'applique de la même manière aux autres structures ressemblant aux répertoires comme les structures de données hiérarchiques ;

— il n'y a pas de chemin de répertoire spécial correspondant à un périphérique physique tel que votre disque dur. C'est différent de RT-11, CP/M, OpenVMS, MS-DOS, AmigaOS et Microsoft Windows, où le chemin comporte le nom du périphérique comme « `C:\` ». (Il existe cependant des entrées de répertoire qui font référence aux périphériques physiques en tant qu'élément du système de fichiers normal. Consulter la Section 1.2.2).

Note
Bien que vous **puissiez** utiliser la plupart des lettres ou symboles dans un nom de fichier, c'est en pratique un mauvaise idée de le faire. Il est préférable d'éviter tous les caractères qui ont une signification particulière sur la ligne de commandes, comme les espaces, tabulations, sauts de ligne, et autres caractères spéciaux : `{ } ( ) [ ] ' ` " \ / > < | ; ! # & ^ * % @ $`. Si vous voulez séparer des mots dans un nom, de bons choix sont le point, le tiret et le tiret souligné. Vous pouvez aussi mettre une majuscule en tête de chaque mot « `CommeCeci` ». Les utilisateurs Linux expérimentés on tendance à aussi à éviter les espaces dans les noms de fichiers.

Note
Le mot « root » signifie soit « utilisateur root », soit « répertoire root » Le contexte de son utilisation devrait permettre de les distinguer.

Note
Le mot **chemin** (« path ») n'est pas utilisé que pour les **noms de fichiers entièrement qualifiés** comme ci-dessus mais aussi pour le **chemin de recherche des commandes**. La signification voulue est habituellement claire selon le contexte.

Les meilleures façons de faire en ce qui concerne la hiérarchie des fichiers sont détaillées dans la « norme de hiérarchie du système de fichiers » (« Filesystem Hierarchy Standard ») (« `/usr/share/doc/debian-policy/fhs/fhs-2.3` et hier(7)). Vous devriez, pour commencer, mémoriser les éléments suivants :

répertoire	utilisation du répertoire
`/`	répertoire racine
`/etc/`	fichiers de configuration valables pour l'ensemble du système
`/var/log/`	fichiers journaux du système
`/home/`	tous les répertoires personnels des utilisateurs non privilégiés

Table 1.3 – Utilisation des répertoires-clés

1.2.2 Fonctionnement interne du système de fichiers

En suivant la **tradition UNIX**, le système Debian GNU/Linux fournit un système de fichiers où les données physiques se trouvent sur des disques durs et d'autres périphériques de stockage et où les interaction avec les périphériques physiques tels que les écrans de console et les consoles distantes connectées en séries sont représentées de manière unifiée dans « `/dev/` ».

Chaque fichier, répertoire, tube nommé (une manière pour les programmes de partager des données) ou périphérique physique sur un système Debian GNU/Linux possède une structure de données appelée inœud (« inode ») qui décrit les attributs qui lui sont associés comme l'utilisateur qui le possède (propriétaire), le groupe auquel il appartient, l'heure de dernier accès, etc. L'idée de pouvoir presque tout représenter dans le système de fichiers était une innovation d'UNIX, et les noyaux modernes de Linux ont encore développé plus loin cette idée. À l'heure actuelle, même les informations concernant les processus qui tournent sur le système se trouvent sur le système de fichiers.

Cette représentation abstraite et unifiée des entités physiques et des processus internes est très puissante puisque cela nous permet d'utiliser la même commande pour le même type d'opération sur des périphériques complètement différents. Il est même possible de changer la manière dont fonctionne le noyau en écrivant des données dans des fichiers spéciaux liés aux processus en cours d'exécution.

ASTUCE
Si vous avez besoin de connaître la correspondance entre une arborescence de fichiers et un périphérique physique, lancez la commande mount(8) sans paramètre.

1.2.3 Permissions du système de fichiers

Les permissions du système de fichiers d'un système basé sur UNIX sont définies pour trois catégories d'utilisateurs :

— l'**utilisateur** qui possède le fichier (**u**) ;
— les autres utilisateurs du **groupe** à qui appartient le fichier (**g**) ;
— tous les **autres** utilisateurs (**o**) dont on parle aussi en tant que « monde entier » ou « tout le monde ».

Pour les fichiers, chaque permission correspondante permet les actions suivantes :

— la permission en **lecture** (**r**) permet à son propriétaire de voir le contenu du fichier ;
— la permission en **écriture** (**w**) permet à son propriétaire de modifier le fichier ;
— la permission d'**exécution** (**x**) permet à son propriétaire de lancer le fichier comme une commande.

Pour les répertoires, chaque permission correspondante permet les actions suivantes :

— la permission en **lecture** (**r**) permet à son propriétaire d'afficher le contenu du répertoire ;
— la permission en **écriture** (**w**) permet à son propriétaire d'ajouter ou supprimer des fichiers de ce répertoires ;
— la permission d'**exécution** (**x**) permet à son propriétaire d'accéder aux fichiers du répertoire.

Ici, la permission en **exécution** sur un répertoire ne signifie pas uniquement l'autorisation de lire des fichiers dans ce répertoire mais aussi l'autorisation de voir leurs attributs, tels que leur taille et l'heure de modification.

ls(1) est utilisé pour afficher les informations de permissions (et davantage) des fichiers et répertoires. Lorsque cette commande est passée avec l'option « `-l` », elle affiche les informations suivantes dans l'ordre donné :

— **type de fichier** (premier caractère) ;

— **autorisation** d'accès au fichier (neuf caractères, constitués de trois caractères pour l'utilisateur, le groupe et « les autres », dans cet ordre) ;
— **nombre de liens physiques** vers le fichier ;
— nom de l'**utilisateur** propriétaire du fichier ;
— nom du **groupe** à qui appartient le fichier ;
— **taille** du fichier en caractères (octets) ;
— **date et heure** du fichier (mtime) ;
— **nom** du fichier.

caractère	signification
-	fichier normal
d	répertoire
l	lien symbolique
c	nœud de périphérique en mode caractère
b	nœud de périphérique en mode bloc
p	tube nommé
s	socket

Table 1.4 – Liste des premiers caractères de la sortie de « `ls -l` » :

chown(1) est utilisé depuis le compte de l'administrateur pour modifier le propriétaire d'un fichier. chgrp(1) est utilisé depuis le compte du propriétaire du fichier ou de l'administrateur pour changer le groupe du fichier. chmod(1) est utilisé depuis le compte du propriétaire du fichier ou de l'administrateur pour changer les droits d'accès à un fichier ou un répertoire. La syntaxe de base pour manipuler le fichier `toto` est la suivante :

```
# chown newowner foo
# chgrp newgroup foo
# chmod  [ugoa][+-=][rwxXst][,...] foo
```

Vous pouvez, par exemple, faire qu'une arborescence de répertoires soit la propriété de l'utilisateur `toto` et partagée par le groupe `titi` en faisant ce qui suit :

```
# cd /some/location/
# chown -R foo:bar .
# chmod -R ug+rwX,o=rX .
```

Il existe trois bits qui donnent des permissions particulières :
— le bit **set user ID** (**s** ou **S** situé à la place du **x**) de l'utilisateur ;
— le bit **set group ID** (**s** ou **S** situé à la place du **x**) du groupe ;
— le **bit collant** « sticky bit » (**t** ou **T** situé à la place du **x**) des « autres ».

Ici la sortie de « `ls -l` » avec ces bits est **en majuscules** si les bits d'exécution cachés par ces sorties ne sont **pas positionnés**.

Définir **set user ID** sur un fichier exécutable permet à un utilisateur d'exécuter les fichiers avec l'identifiant du propriétaire du fichier (par exemple **root**). De la même manière, définir **set group ID** sur un fichier exécutable permet d'exécuter le fichier avec l'identifiant de groupe du fichier (par exemple **root**). Parce que ces positionnements de bits peuvent créer des risques de sécurité, il ne faut les activer qu'avec des précautions extrêmes.

Définir **set group ID** sur un répertoire permet la création de fichiers à la BSD où tous les fichiers créés dans un répertoire appartiennent au **groupe** du répertoire.

Positionner le **sticky bit** d'un répertoire empêche un fichier de ce répertoire d'être supprimé par un utilisateur qui n'est pas le propriétaire du fichier. Pour sécuriser le contenu d'un fichier dans des répertoires pouvant être écrits par tout le monde tels que « `/tmp` » ou dans des répertoires pouvant être écrits par le groupe, il ne faut pas uniquement supprimer la permission du fichier en **écriture** mais aussi positionner le **sticky bit** sur le répertoire. Sinon, le fichier pourra être supprimé et un nouveau fichier créé avec le même nom par un utilisateur quelconque ayant accès en écriture au répertoire.

Voici quelques exemples intéressants de permissions de fichiers.

```
$ ls -l /etc/passwd /etc/shadow /dev/ppp /usr/sbin/exim4
crw------T 1 root root   108, 0 Oct 16 20:57 /dev/ppp
-rw-r--r-- 1 root root     2761 Aug 30 10:38 /etc/passwd
-rw-r----- 1 root shadow   1695 Aug 30 10:38 /etc/shadow
-rwsr-xr-x 1 root root   973824 Sep 23 20:04 /usr/sbin/exim4
$ ls -ld /tmp /var/tmp /usr/local /var/mail /usr/src
drwxrwxrwt 14 root root  20480 Oct 16 21:25 /tmp
drwxrwsr-x 10 root staff  4096 Sep 29 22:50 /usr/local
drwxr-xr-x 10 root root   4096 Oct 11 00:28 /usr/src
drwxrwsr-x  2 root mail   4096 Oct 15 21:40 /var/mail
drwxrwxrwt  3 root root   4096 Oct 16 21:20 /var/tmp
```

Il existe aussi un mode numérique pour décrire les permissions des fichiers avec chmod(1). Ce mode numérique utilise des nombres en base 8 (radix=8) codés sur 3 ou 4 chiffres.

chiffre	signification
1er chiffre optionnel	somme de **set user ID** (=4), **set group ID** (=2), et **sticky bit** (=1)
2ème chiffre	somme des permissions de **lecture** (=4), **écriture** (=2), et **exécution** (=1) pour l'**utilisateur**
3ème chiffre	identique pour **groupe**
4ème chiffre	identique pour **autres**

Table 1.5 – Mode numérique des permissions de fichiers dans les commandes chmod(1)

Cela peut sembler compliqué mais c'est en fait assez simple. Si vous regardez les quelques premières colonnes (2-10) de la sortie de la commande « `ls -l` » et que vous lisez en représentation binaire (base 2) les permissions des fichiers (le « - » représentant « 0 » et « rwx » représentant « 1 »), les trois derniers chiffres de la valeur numérique du mode devraient vous donner la représentation des permissions du fichier en octal (base 8).

Essayez, par exemple, ce qui suit :

```
$ touch foo bar
$ chmod u=rw,go=r foo
$ chmod 644 bar
$ ls -l foo bar
-rw-r--r-- 1 penguin penguin 0 Oct 16 21:39 bar
-rw-r--r-- 1 penguin penguin 0 Oct 16 21:35 foo
```

ASTUCE
Si vous avez besoin d'accéder aux informations affichées par « `ls -l` » depuis un script de l'interpréteur de commandes, vous devrez utiliser des commandes pertinentes telles que test(1), stat(1) et readlink(1). Vous pouvez aussi utiliser les commandes internes du shell, telles que « `[` » ou « `test` ».

1.2.4 Contrôle des permissions pour les fichiers nouvellement créés : umask

Les permissions appliquées à un fichier ou à un répertoire venant d'être créé sont restreintes par la commande interne du shell `umask`. Consultez dash(1), bash(1) et bash-builtins(7).

```
(file permissions) = (requested file permissions) & ~(umask value)
```

Le système Debian utilise par défaut un schéma de groupe privé par utilisateur (UPG). Un UPG est créé chaque fois qu'un utilisateur est ajouté au système. Un UPG a le même nom que l'utilisateur pour lequel il a été créé et cet utilisateur est le seul membre de l'UPG. Un principe d'UPG rend sûr le positionnement de umask à `0002` car chaque utilisateur a son propre groupe privé. (Sur certaines variantes d'UNIX, il est assez courant de faire appartenir tous les utilisateurs au même groupe **`users`** et, dans ce cas, c'est une bonne idée de définir umask à `0022` pour des raisons de sécurité.)

umask	permissions des fichiers créés	permissions des répertoires créés	utilisation
0022	-rw-r--r--	-rwxr-xr-x	ne peut être écrit que par l'utilisateur
0002	-rw-rw-r--	-rwxrwxr-x	peut être écrit par le groupe

Table 1.6 – Exemples de valeurs de **umask**

ASTUCE
Activez UPG en plaçant « `umask 002` » dans le fichier `~/.bashrc`.

1.2.5 Permissions pour les groupes d'utilisateurs (group)



AVERTISSEMENT
Assurez-vous d'enregistrer les modifications non enregistrées avant de redémarrer ou d'effectuer des actions similaires.

Vous pouvez ajouter un utilisateur `penguin` à un groupe `bird` en deux étapes :

— modifier la configuration de groupe en utilisant une des choses suivantes :
  — exécuter « `sudo usermod -aG bird penguin` »,
  — exécuter « `sudo adduser penguin bird` » (seulement sur les systèmes classiques),
  — exécuter « `sudo vigr` » pour `/etc/group` et "`sudo vigr -s`" pour `/etc/gshadow` pour ajouter `penguin` dans la ligne pour `bird` ;
— appliquer la configuration en utilisant une des choses suivantes :
  — redémarrage complet et connexion (meilleure option),
  — Logout via GUI menu and login. (This may not work under the modern Desktop environment.)

Vous pouvez retirer un utilisateur `penguin` d'un groupe `bird` en deux étapes :

— modifier la configuration de groupe en utilisant une des choses suivantes :
  — exécuter « `sudo usermod -rG bird penguin` »,
  — exécuter « `sudo deluser penguin bird` » (seulement sur les systèmes Debian classiques),
  — exécuter « `sudo vigr` » pour `/etc/group` et « `sudo vigr -s` » pour `/etc/gshadow` pour retirer `penguin` dans la ligne pour `bird` ;
— appliquer la configuration en utilisant une des choses suivantes :
  — redémarrage complet et connexion (meilleure option),
  — exécuter « `kill -TERM -1` » et faire quelques actions de réglage telles que « `systemctl restart NetworkMar`
  — la déconnexion à l'aide du menu graphique n'est pas une option pour le bureau Gnome.

Tout essai de redémarrage à chaud est un remplacement fragile d'un redémarrage complet dans les systèmes de bureau moderne.

Note
Vous pouvez aussi ajouter dynamiquement des utilisateurs aux groupes durant le processus d'authentification en ajoutant la ligne « `auth optional pam_group.so` » au fichier « `/etc/pam.d/common-auth` » et en définissant « `/etc/security/group.conf` » (consultez Chapitre 4).

groupe	description des fichiers et périphériques accessibles
`dialout`	accès complet et direct aux ports série (« `/dev/ttyS[0-3]` »)
`dip`	accès limité aux ports série pour une connexion « **Dialup IP** (réseau commuté) vers des pairs de confiance
`cdrom`	lecteurs et graveurs de CD-ROM, DVD+/-RW
`audio`	périphérique audio
`video`	périphérique vidéo
`scanner`	dispositifs de numérisation (scanners)
`adm`	journaux de surveillance du système
`staff`	quelques répertoires où effectuer du travail d'administration de début : « `/usr/local` », « `/home` »

Table 1.7 – Liste des groupes importants fournis par le système pour l'accès aux fichiers

The hardware devices are just another kind of file on the Debian system. If you have problems accessing devices such as USB flash drive and CD-ROM from a user account, you should make that user a member of the relevant group.

Certains groupes importants fournis par le système permettent à leurs membres l'accès à des fichiers et des périphériques particuliers sans avoir les privilèges de l'`administrateur`.

ASTUCE
Vous devez être membre du groupe `dialout` pour pouvoir reconfigurer un modem, numéroter vers n'importe où, etc. Mais si l'`administrateur` crée dans « `/etc/ppp/peers/` » des fichiers de configuration pour des pairs de confiance, vous ne devrez appartenir qu'au groupe `dip` pour créer une connexion **commutée** (« Dialup IP ») vers ces pairs de confiance avec commandes pppd(8), pon(1) et poff(1).

Certains groupes faisant partie du système, permettent à leurs membres d'exécuter des commandes particulières sans les privilèges de l'administrateur (`root`).

groupe	commandes accessibles
`sudo`	execute any command with superuser privileges
`lpadmin`	exécuter des commandes pour ajouter, modifier et supprimer des imprimantes de la base de données des imprimantes

Table 1.8 – Liste des groupes importants fournis par le système pour l'exécution de commandes particulières

Pour une liste complète des groupes et des utilisateurs fournis par le système, veuillez consulter une version récente du document « Utilisateurs et groupes » (« Users and Groups » se trouvant dans `/usr/share/doc/base-passwd/users` qui est fourni par le paquet `base-passwd`.

Consultez passwd(5), group(5), shadow(5), newgrp(1), vipw(8), vigr(8) et pam_group(8) pour les commande de gestion des utilisateurs et des groupes du système.

1.2.6 Horodatage

Il existe trois types d'horodatage pour un fichier GNU/Linux.

Note
ctime n'est pas la date de création du fichier.

type	signification (définition historique d'Unix)
mtime	date de modification du fichier (`ls -l`)
ctime	date de changement d'état du fichier (`ls -lc`)
atime	date de dernier accès au fichier (`ls -lu`)

Table 1.9 – Liste des types d'horodatage

Note
La valeur d'**atime** réelle sur GNU/Linux peut être en fait différente de celle donnée par la définition Unix historique.

— Écraser un fichier va modifier tous les attributs **mtime**, **ctime** et **atime** du fichier.
— Modifier le propriétaire ou les droits d'un fichier va changer les attributs **ctime** et **atime** du fichier.
— La lecture d'un fichier va modifier l'attribut **atime** du fichier sur le système Unix d'historique.
— La lecture d'un fichier va modifier l'attribut **atime** du fichier que un système GNU/Linux si son système de fichier est monté avec "`strictatime`".
— Lire un fichier pour la première fois ou après un jour modifie son attribut **atime** sur un système GNU/Linux avec un système de fichiers monté en « `relatime` » (par défaut depuis Linux 2.6.30).
— Lire un fichier ne modifie pas son attribut **atime** sous GNU/Linux si le système de fichiers est monté avec « `noatime` ».

Note
Les options de montage « `noatime` » et « `relatime` » sont introduites pour améliorer les performances de lecture du système de fichiers en utilisation normale. La lecture de fichiers simples avec l'option « `strictatime` » accompagne l'opération chronophage d'écriture d'une mise à jour de l'attribut **atime**. Mais cet attribut **atime** est rarement utilisé sauf pour le fichier mbox(5). Voir mount(8).

Utilisez la commande touch(1) pour modifier l'horodatage des fichiers existants.

En ce qui concerne l'horodatage, la commande `ls` affiche des chaînes avec les paramètres linguistiques non anglais (« `fr_FR.UTF-8` »).

```
$ LANG=C  ls -l foo
-rw-rw-r-- 1 penguin penguin 0 Oct 16 21:35 foo
$ LANG=en_US.UTF-8  ls -l foo
-rw-rw-r-- 1 penguin penguin 0 Oct 16 21:35 foo
$ LANG=fr_FR.UTF-8  ls -l foo
-rw-rw-r-- 1 penguin penguin 0 oct. 16 21:35 foo
```

ASTUCE
Consultez Section 9.3.4 pour personnaliser la sortie de « `ls -l` ».

1.2.7 Liens

Il existe deux méthodes pour associer le fichier « `toto` » avec un nom de fichier différent « `titi` » :

— Hard link
  — Nom dupliqué d'un fichier existant
  — "`ln foo bar`"
— Lien symbolique ou <symlink>

— Fichier spécial pointant vers un autre fichier par son nom
— "`ln -s foo bar`"

Consultez l'exemple suivant pour des modifications du nombre de liens et les subtiles différences dans le résultat de la commande `rm`.

```
$ umask 002
$ echo "Original Content" > foo
$ ls -li foo
1449840 -rw-rw-r-- 1 penguin penguin 17 Oct 16 21:42 foo
$ ln foo bar     # hard link
$ ln -s foo baz  # symlink
$ ls -li foo bar baz
1449840 -rw-rw-r-- 2 penguin penguin 17 Oct 16 21:42 bar
1450180 lrwxrwxrwx 1 penguin penguin  3 Oct 16 21:47 baz -> foo
1449840 -rw-rw-r-- 2 penguin penguin 17 Oct 16 21:42 foo
$ rm foo
$ echo "New Content" > foo
$ ls -li foo bar baz
1449840 -rw-rw-r-- 1 penguin penguin 17 Oct 16 21:42 bar
1450180 lrwxrwxrwx 1 penguin penguin  3 Oct 16 21:47 baz -> foo
1450183 -rw-rw-r-- 1 penguin penguin 12 Oct 16 21:48 foo
$ cat bar
Original Content
$ cat baz
New Content
```

Le lien physique peut être mis en place à l'intérieur du même système de fichiers et partage le même numéro d'inœud, ce que montre l'option « `-i` » de ls(1).

Le lien symbolique a les permissions d'accès nominales au fichier de « `rwxrwxrwx` » comme il apparaît dans l'exemple ci-dessus, alors que les permissions d'accès effectives sont celles du fichier vers lequel il pointe.



Attention
En règle générale — à moins d'avoir une très bonne raison pour cela — il faudrait s'abstenir de créer des liens physiques ou des liens symboliques compliqués. Cela peut provoquer des cauchemars lorsque la combinaison logique des liens symboliques crée une boucle dans le système de fichiers.

Note
Il est généralement préférable d'utiliser des liens symboliques plutôt que des liens physiques à moins que vous n'ayez une bonne raison d'utiliser un lien physique.

Le répertoire « `.` » est lié au répertoire dans lequel il apparaît, le nombre de liens de n'importe quel nouveau répertoire commence donc à 2. Le répertoire « `..` » est lié au répertoire parent, le nombre de liens du répertoire augmente donc lors de l'ajout de nouveaux sous-répertoires.

Si vous venez de passer à Linux depuis Windows, la bonne conception d'un système de fichiers UNIX comparé à l'équivalent le plus proche que sont les « raccourcis Windows » deviendra vite claire. Parce qu'il est implémenté dans le système de fichiers, les applications ne voient pas de différence entre un fichier lié et son original. Dans le cas de liens physiques, il n'y a vraiment aucune différence.

1.2.8 Tubes nommés (FIFO)

Un tube nommé est un fichier qui se comporte comme un tuyau. Vous mettez quelque chose dans le tuyau et il ressort à l'autre bout. C'est donc appelé une FIFO, ou « premier entré-premier sorti » (First-In-First-Out) : la première chose que vous mettez dans le tuyau est la première chose qui ressortira à l'autre bout.

Si vous écrivez vers un tube nommé, le processus qui écrit dans le tube ne se termine pas avant que l'information ne soit lue depuis le tube. Si vous effectuez une lecture depuis un tube nommé, le processus de lecture attendra jusqu'à ce qu'il n'y ait plus rien à lire avant de se terminer. La taille d'un tube est toujours nulle — il ne stocke pas de données, ce n'est qu'un lien entre deux processus, comparable à la fonction fournie par l'opérateur « | » dans l'interpréteur de commandes. Cependant, comme ce tube a un nom, il n'est pas nécessaire que les deux processus se trouvent sur la même ligne de commandes ni même qu'ils soient lancés par le même utilisateur. Les tubes sont une innovation déterminante d'UNIX.

Essayez, par exemple, ce qui suit :

```
$ cd; mkfifo mypipe
$ echo "hello" >mypipe & # put into background
[1] 8022
$ ls -l mypipe
prw-rw-r-- 1 penguin penguin 0 Oct 16 21:49 mypipe
$ cat mypipe
hello
[1]+  Done                    echo "hello" >mypipe
$ ls mypipe
mypipe
$ rm mypipe
```

1.2.9 Sockets

Les sockets sont utilisées de manière intensives dans les communications par Internet, les bases de données et le système d'exploitation lui-même. Elles sont semblables aux tubes nommés (FIFO) et permettent aux processus d'échanger des informations même s'ils tournent sur des ordinateurs différents. Pour la socket, ces processus n'ont pas besoin de tourner en même temps ni de tourner en tant que fils du même processus père. C'est le point d'aboutissement du processus d'intercommunication (IPC). (« interprocess communication »). L'échange d'informations peut avoir lieu entre deux machines au travers du réseau. Les deux plus courantes sont la socket Internet (« Internet socket ») et la socket du domaine UNIX (« UNIX domain socket »).

ASTUCE
"`ss -t -u -a`" (from the `iproute2` package) provides a very useful overview of sockets that are open on a given system.

1.2.10 Fichiers de périphériques

Fichiers de périphériques fait référence aux périphériques virtuels ou physiques de votre système, tels que vos disques durs, carte vidéo, écran ou clavier. Un exemple de périphérique virtuel est la console, représentée par « `/dev/console` ».

Il y a 2 types de fichiers de périphériques :

- **périphérique en mode caractère** (« character device ») :
  - permettent l'accès à un caractère à la fois
  - 1 caractère = 1 octet
  - par exemple, les claviers, les ports série, …
- **Périphériques en mode bloc** (« block device ») :
  - accèdent aux données par éléments plus importants appelés blocs
  - 1 bloc > 1 octet
  - par exemple, les disques durs, …

Vous pouvez lire et écrire sur les fichiers de périphériques, cependant, le fichier peut fort bien contenir des données binaires qui peuvent être du charabia incompréhensible pour les êtres humains. Écrire des données directement dans ces fichiers est parfois utile pour déboguer des connexions matérielles. Vous pouvez, par exemple, vider un fichier texte vers le périphérique d'impression « `/dev/lp0` » ou envoyer des commandes de modem vers le port série approprié « `/dev/ttyS0` ». Mais, à moins que ce ne soit fait avec précautions, cela peut provoquer un désastre majeur. Soyez donc prudent.

Note
Pour l'accès normal à une imprimante, utilisez lp(1).

On affiche le numéro de nœud du périphérique en utilisant la commande ls(1) comme suit :

```
$ ls -l /dev/sda /dev/sr0 /dev/ttyS0 /dev/zero
brw-rw---T  1 root disk     8,  0 Oct 16 20:57 /dev/sda
brw-rw---T+ 1 root cdrom   11,  0 Oct 16 21:53 /dev/sr0
crw-rw---T  1 root dialout  4, 64 Oct 16 20:57 /dev/ttyS0
crw-rw-rw-  1 root root     1,  5 Oct 16 20:57 /dev/zero
```

— « `/dev/sda` » a le numéro majeur de périphérique 8 et le numéro mineur de périphérique 0. Il est accessible en lecture et écriture aux utilisateurs qui appartiennent au groupe `disk`.
— « `/dev/sr0` » a le numéro majeur de périphérique 11 et le numéro mineur de périphérique 0. Il est accessible en lecture et écriture aux utilisateurs qui appartiennent au groupe `cdrom`.
— « `/dev/ttyS0` » a le numéro majeur de périphérique 4 et le numéro mineur de périphérique 64. Il est accessible en lecture et écriture aux utilisateurs qui appartiennent au groupe `dialout`.
— « `/dev/zero` » a le numéro majeur de périphérique 1 et le numéro mineur de périphérique 5. Il est accessible en lecture et écriture à tout le monde.

Sur les systèmes Linux modernes, le système de fichiers sous « `/dev/` » est automatiquement rempli par udev(7).

1.2.11 Fichiers spéciaux de périphériques

Il y a aussi certains fichiers spéciaux de périphériques.

fichier de périphérique	action	description de la réponse
`/dev/null`	lire	retourne le « caractère fin de fichier (EOF) »
`/dev/null`	écrire	ne retourne rien (un puits de données sans fond)
`/dev/zero`	lire	retourne le « caractère `\0` (NULL) » (qui n'est pas identique au chiffre zéro ASCII)
`/dev/random`	lire	retourne des caractères aléatoires depuis un générateur de nombres aléatoires réel, en donnant une vraie entropie (lent)
`/dev/urandom`	lire	retourne des caractères aléatoires depuis un générateur de nombres pseudo-aléatoires, sécurisé par chiffrement
`/dev/full`	écrire	retourne une erreur disque plein (ENOSPC)

Table 1.10 – Liste des fichiers spéciaux de périphériques

Ils sont fréquemment utilisés en conjonction avec une redirection du shell (consultez Section 1.5.8).

1.2.12 procfs et sysfs

procfs et sysfs montés sur « /proc » et « /sys sont des pseudo-systèmes de fichiers, ils présentent dans l'espace utilisateur des structures de données internes du noyau. En d'autres termes, ces entrées sont virtuelles, ce qui signifie qu'elles présentent une fenêtre sur le fonctionnement du système d'exploitation.

Le répertoire « /proc » contient (entre autres choses), pour chacun des processus tournant sur le système, un sous-répertoire dont le nom est l'identifiant du processus (PID). Les utilitaires du système qui accèdent aux information des processus, tels que ps(1), obtiennent leurs informations depuis cette structure de répertoires.

Les répertoires qui se trouvent sous « /proc/sys/ » contiennent des interfaces permettant de modifier certains paramètres du système alors qu'il est en fonctionnement. (Vous pouvez faire la même chose au travers de la commande spécialisée sysctl(8) ou de son fichier de configuration « /etc/sysctl.d/*.conf .)

Certaines personnes paniquent lorsqu'elles remarquent un fichier particulier - « /proc/kcore » — qui est particulièrement énorme. C'est (plus ou moins) le contenu de la mémoire de votre ordinateur. Il est utilisé pour déboguer le noyau. C'est un fichier virtuel qui pointe vers la mémoire de l'ordinateur, ne vous inquiétez donc pas de sa taille.

Les répertoires sous « /sys » contiennent des structures de données exportées depuis le noyau, leurs attributs et les liens entre elles. Ils contiennent aussi des interfaces pour modifier certains paramètres du noyau pendant son fonctionnement.

Consultez « proc.txt(.gz) », « sysfs.txt(.gz) » et d'autres documents en rapport dans la documentation du noyau de Linux (« /usr/share/doc/linux-doc-*/Documentation/filesystems/* ») fournie par le paquet linux-doc-*.

1.2.13 tmpfs

Le tmpfs est un système de fichiers temporaire qui garde tous les fichiers en mémoire virtuelle. Les données du tmpfs dans le page cache en mémoire peuvent être déplacées dans l'espace d'échange sur disque en cas de besoin.

Le répertoire « /run » est monté en tmpfs au tout début du processus de démarrage. Cela permet d'écrire dessus même quand le répertoire racine « / » est monté en lecture seule. C'est le nouvel emplacement pour le stockage de fichiers en état transitoire qui remplace plusieurs emplacements décrits dans la norme de hiérarchie du système de fichiers (« Filesystem Hierarchy Standard ») version 2.3 :

— « /var/run » → « /run »
— « /var/lock » → « /run/lock »
— "/dev/shm" → "/run/shm"

Consultez « tmpfs.txt(.gz) » dans la documentation du noyau de Linux (« /usr/share/doc/linux-doc-*/Docume
fournie par le paquet linux-doc-*.

1.3 Midnight Commander (MC)

Midnight Commander (MC) est un « couteau Suisse » GNU pour la console Linux et d'autres environnements de terminaux. Il permet au débutant d'acquérir une expérience de la console pilotée par des menus, ce qui est bien plus facile à apprendre que les commandes UNIX standard.

Il vous faudra peut-être installer le paquet Midnight Commander dont le nom est « mc » en effectuant ce qui suit :

```
$ sudo apt-get install mc
```

Utilisez la commande mc(1) pour parcourir le système Debian. C'est la meilleure manière d'apprendre. Vous pouvez explorer certains emplacements intéressants en utilisant simplement les touches de curseur et la touche Entrée :

— « /etc » et ses sous-répertoires
— « /var/log » et ses sous-répertoires
— « /usr/share/doc » et ses sous-répertoires
— « /usr/sbin » et « /usr/bin »

1.3.1 Personnalisation de MC

Pour que MC modifie le répertoire de travail en quittant, et cd vers le répertoire, je vous suggère de modifier « ~/.bashrc » afin d'inclure un script fourni par le paquet mc :

```
. /usr/lib/mc/mc.sh
```

Vous trouverez une explication dans mc(1) (option « -P »). (Si vous ne comprenez pas exactement ce dont je parle ici, vous pourrez le faire plus tard.)

1.3.2 Démarrer MC

MC peut être lancé par :

```
$ mc
```

MC prend en charge toutes les opérations sur les fichiers par l'intermédiaire de son menu, ce qui ne demande que peu d'effort de la part de l'utilisateur. Pressez simplement F1 pour obtenir l'écran d'aide. Vous pouvez jouer avec MC simplement en pressant les touches de curseur et les touches de fonctions.

Note
Sur certaines consoles telles que gnome-terminal(1), les actions sur les touches de fonction peuvent être récupérées par le programme de console. Vous pouvez désactiver cette fonctionnalité dans le menu « Préférences » → « Général » et « Raccourcis » pour le `terminal gnome`.

Si vous rencontrez un problème de codage de caractères qui entraîne une corruption de l'affichage, ajouter « -a » à la ligne de commandes de MC peut aider à éviter les problèmes.

Si cela ne résout pas vos problèmes d'affichage avec MC, consultez Section 8.3.1.

1.3.3 Gestionnaire de fichiers de MC

Il y a par défaut, deux panneaux de répertoires affichant les listes de fichiers. Un autre mode utile est de définir la fenêtre de droite à « information » afin de voir les informations de privilèges d'accès aux répertoires. Vous trouverez ci-après quelques raccourcis clavier essentiels. Si le démon gpm(8) tourne, une souris est utilisable avec les consoles Linux en mode caractères. (Assurez-vous de presser la touche majuscules pour obtenir le comportement normal de couper-coller avec MC.)

touche	affectation
F1	menu d'aide
F3	visualisateur interne de fichiers
F4	éditeur interne
F9	activer le menu déroulant
F10	quitter Midnight Commander
Tab	passer d'une fenêtre à l'autre
Insert or Ctrl-T	marquer le fichier pour des opérations sur plusieurs fichiers telles que copier
Del	effacer le fichier (attention, configurez MC dans le mode d'effacement sécurisé)
Touches de curseur	autoexplicatif

Table 1.11 – Touches de raccourcis de MC

1.3.4 Astuces de la ligne de commandes dans MC

— la commande `cd` changera le répertoire affiché sur l'écran sélectionné ;
— `Ctrl-Entrée` ou `Alt-Entrée` copiera un nom de fichier sur la ligne de commandes. Utilisez cela avec les commandes cp(1) et mv(1) en association avec l'édition de la ligne de commandes ;
— `Alt-Tab` affichera les choix de l'interpréteur de commandes pour l'expansion du nom de fichier ;
— on peut indiquer le répertoire de départ pour les deux fenêtres en paramètre de MC. Par exemple « `mc /etc /root` » ;
— `Échap` + `touche n` → `Fn` (par exemple `Échap` + `1` → `F1`, etc. ; `Échap` + `0` → `F10`) ;
— Presser la touche `Échap` avant une touche a le même effet que presser simultanément `Alt` et la touche. Par exemple, entrez `Échap` + `c` pour `Alt-C`. `Échap` est appelée métatouche et parfois notée « `M-` ».

1.3.5 Éditeur interne de MC

L'éditeur interne possède une manière intéressante d'effectuer un copier-coller. Presser `F3` marque le début de la sélection, `F3` pressé une seconde fois marque la fin de la sélection et la met en surbrillance. Vous pouvez ensuite déplacer votre curseur. Si vous appuyez sur F6, la zone sélectionnée sera déplacée jusqu'à l'emplacement du curseur. Si vous pressez sur F5, la zone sélectionnée sera copiée et insérée à l'emplacement du curseur. `F2` enregistrera le fichier. `F10` vous permettra de quitter l'éditeur. La plupart des touches de déplacement du curseur fonctionnent de manière intuitive.

Cet éditeur peut être directement lancé avec un fichier en utilisant l'une des commandes suivantes :

```
$ mc -e filename_to_edit
```

```
$ mcedit filename_to_edit
```

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

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

De nombreux programmes utilisent aussi les variables d'environnement « `$EDITOR` » ou « `$VISUAL` » afin de décider quel éditeur utiliser. Si, au départ, vous n'êtes à l'aise ni avec vim(1) ni avec nano(1), vous pouvez définir ces variable à « `mcedit` » en ajoutant les lignes suivantes au fichier « `~/.bashrc` » :

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

Je recommande de les définir à « `vim` » si possible.

Si vous n'êtes pas à l'aise avec vim(1), vous pouvez continuer à utiliser mcedit(1) pour la plupart des tâches de maintenance du système.

1.3.6 Visualisateur interne de MC

MC possède un visualisateur intelligent. C'est un très bon outil pour rechercher des mots dans des documents. Je l'utilise toujours pour lire les fichiers qui sont dans le répertoire « `/usr/share/doc` ». C'est la manière la plus rapide de naviguer dans les masses d'informations sur Linux. Ce visualisateur peut être chargé directement utilisant l'une des commandes suivantes :

```
$ mc -v path/to/filename_to_view
```

```
$ mcview path/to/filename_to_view
```

1.3.7 Possibilités de démarrage automatique de MC

Pressez Entrée sur un fichier, et le programme approprié prendra en charge le contenu du fichier (consultez Section 9.4.11). Il s'agit là d'une fonctionnalité très pratique de MC.

type de fichier	réaction à la touche Entrée
fichier exécutable	exécuter la commande
fichier de page de manuel	envoyer (« pipe ») le contenu au logiciel de visualisation
fichier html	envoyer (« pipe ») le contenu au navigateur web
fichiers « `*.tar.gz` » et « `*.deb` »	parcourir le contenu comme si c'était un sous-répertoire

Table 1.12 – Réaction à la touche Entrée dans MC

Afin de permettre le fonctionnement de ces visualisateurs et de ces fonctionnalités de fichiers virtuels, les fichiers pouvant être visualisés ne doivent pas être définis comme étant exécutables. Modifiez leur état avec la commande chmod(1) ou par l'intermédiaire du menu fichiers de MC.

1.3.8 Système de fichiers virtuel de MC

MC peut être utilisé pour accéder à des fichiers au travers d'Internet. Allez au menu en pressant « `F9` », « `Entrée` » et « h » pour activer le système de fichiers Shell. Entrez un URL sous la forme « `sh://[utilisateur@]machine[:options` qui récupère un répertoire distant qui apparait alors comme local en utilisant `ssh`.

1.4 L'environnement élémentaire de travail de type UNIX

Bien que MC vous permette de faire à peu près n'importe quoi, il est très important que vous appreniez à utiliser les outils en ligne de commande appelés depuis l'invite de l'interpréteur de commandes, et que vous vous familiarisiez avec un environnement de travail de type UNIX.

1.4.1 L'interpréteur de commandes de connexion

Étant donné que l'interpréteur de connexion peut être utilisé par certains programmes d'initialisation du système, il est prudent de le conserver en tant que bash(1) et d'éviter de basculer l'interpréteur de connexion en tant que chsh(1).

Si vous souhaitez utiliser une autre invite interactive d'interpréteur, définissez-la à partir de la configuration de l'émulateur de terminal graphique ou démarrez-la à partir de `~/.bashrc`, par exemple, en y plaçant « `exec /usr/bin/zsh -i -l` » ou « `exec /usr/bin/fish -i -l` ».

ASTUCE
Les interpréteurs de commandes POSIX partagent une syntaxe commune, mais leur comportement peut diverger, même pour des choses aussi élémentaires que les variables de l'interpréteur ou les expansions de motifs. Veuillez consulter leur documentation pour une description détaillée.

Dans ce chapitre du didacticiel, l'interpréteur interactif sera toujours `bash`.

1.4.2 Personnaliser bash

Vous pouvez personnaliser le comportement de bash(1) à l'aide de « `~/.bashrc` ».

Essayez, par exemple, ce qui suit :

paquet	popularité	taille	Interpréteur POSIX	description
bash	V:900, I:1000	7309	Oui	Bash : Shell GNU Bourne Again. (standard de fait)
bash-completion	V:41, I:964	1952	N/A	autocomplétion programmable pour l'interpréteur bash
dash	V:933, I:999	211	Oui	Le Shell Almquist de Debian. Bon pour les scripts en shell
zsh	V:38, I:67	2571	Oui	Z shell : interpréteur standard avec de nombreuses améliorations
tcsh	V:4, I:14	1366	Non	Shell TENEX C : version améliorée de csh de Berkeley
mksh	V:5.0, I:7.3	7713	Oui	Une version de Korn shell
csh	V:1.0, I:5.0	348	Non	C Shell OpenBSD, une version de csh de Berkeley
sash	V:0.3, I:4.6	1249	Oui	Interpréteur de commandes autonome avec des commandes intégrées. (Ne convient pas en tant que « `/usr/bin/sh` » standard)
ksh	V:0.2, I:7.5	65	Oui	la vraie version de AT&T du shell Korn
rc	V:0.08, I:0.69	182	Non	implémentation du shell rc Plan 9 de AT&T
posh	V:0.01, I:0.21	195	Oui	Policy-compliant Ordinary SHell (dérivé de `pdksh`)

Table 1.13 – Liste d'interpréteurs de commandes (« shells »)

```
# enable bash-completion
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

# CD upon exiting MC
. /usr/lib/mc/mc.sh

# set CDPATH to a good one
CDPATH=.:/usr/share/doc:~:~/Desktop:~
export CDPATH

PATH="${PATH+$PATH:}/usr/sbin:/sbin"
# set PATH so it includes user's private bin if it exists
if [ -d ~/bin ] ; then
  PATH="~/bin${PATH+:$PATH}"
fi
export PATH

EDITOR=vim
export EDITOR
```

ASTUCE
Vous pourrez trouver davantage d'informations concernant les astuces de personnalisation de `bash`, comme Section 9.3.6, dans Chapitre 9.

ASTUCE
Le paquet `bash-completion` permet la complétions programmable pour `bash`.

1.4.3 Combinaisons particulières de touches

Dans un environnement « de type UNIX », certaines séquences de touches ont une signification particulière. Vous remarquerez que sur une console Linux normale en mode caractères, seules les touches `Ctrl` et `Alt` situées à gauche fonctionnent de la manière voulue. Voici quelques séquences de touches dont il est intéressant de se souvenir :

touche	description des raccourcis clavier
`Ctrl-U`	effacer la ligne avant le curseur
`Ctrl-H`	effacer le caractère précédant le curseur
`Ctrl-D`	terminer l'entrée (quitter l'interpréteur si vous en utilisez un)
`Ctrl-C`	terminer un programme en cours d'exécution
`Ctrl-Z`	arrêter temporairement un programme en le mettant en tâche de fond
`Ctrl-S`	arrêter le défilement de l'affichage à l'écran
`Ctrl-Q`	reprendre le défilement de l'affichage
`Ctrl-Alt-Del`	redémarrer ou arrêter le système, consultez inittab(5)
`Touche Alt de gauche` (optionnellement, `touche Windows`)	touche « meta » pour les interfaces utilisateurs Emacs et similaires
`Up-arrow`	lancer la recherche dans l'historique des commandes sous `bash`
`Ctrl-R`	lancer la recherche incrémentale dans l'historique des commandes sous `bash`
`Tab`	complèter l'entrée du nom de fichier de la ligne de commandes sous `bash`
`Ctrl-V Tab`	entrer une `Tabulation` sans expansion de la ligne de commande sous `bash`

Table 1.14 – Liste des raccourcis clavier de bash

ASTUCE
La fonctionnalité `Ctrl-S` du terminal peut être désactivée en utilisant stty(1).

1.4.4 Opérations de souris

Les opérations de souris pour le texte sur le système Debian mélangent 2 styles avec quelques acrobaties :

- Opérations de souris de style Unix traditionnel :
  - utilisation de 3 boutons (clic) ;
  - utilisation de PRIMAIRE ;
  - utilisation par des applications X telles que `xterm` et des applications textuelles dans la console Linux.
- Opérations de souris pour le style graphique moderne :
  - utilisation de 2 boutons (glisser + cliquer) ;
  - utilisation de PRIMAIRE et PRESSE-PAPIER ;
  - utilisation dans les applications graphiques modernes telles que `gnome-terminal`.

Ici, la sélection PRIMAIRE est la plage de texte surlignée. Dans un programme de terminal, `Shift-Ctrl-C` est utilisé pour éviter de mettre fin à un programme en cours.

action	réponse
Clic-gauche et glisser de la souris	sélectionner la plage comme sélection PRIMAIRE
Clic-gauche	sélectionner le début de plage pour la sélection PRIMAIRE
clic droit (traditionnel)	sélectionner la fin de plage pour la sélection PRIMAIRE
clic droit (moderne)	action dépendant du contexte (couper/copier/coller)
clic central ou Maj-`Inser`	insérer la sélection PRIMAIRE à la position du curseur
`Ctrl-X`	couper la sélection PRIMAIRE vers le PRESSE-PAPIER
`Ctrl-C` (`Shift-Ctrl-C` dans un terminal)	copier la sélection PRIMAIRE dans le PRESSE-PAPIER
`Ctrl-V`	coller le contenu de PRESSEPAPIERS à l'emplacement du curseur

Table 1.15 – Liste des opérations de la souris et des actions associées sur les touches pour Debian

La molette centrale des souris à molette modernes est considérée comme le bouton du milieu et peut être utilisée pour les clics-milieu. Cliquer simultanément le bouton de gauche et le bouton de droite sert à émuler le bouton du milieu sur les systèmes ayant une souris à deux boutons.

Pour pouvoir utiliser une souris avec les consoles Linux en mode caractère, il faut que gpm(8) tourne en tant que démon.

1.4.5 Le visualisateur de fichiers

La commande less(1) invoque le visualisateur moderne (afficheur de contenu de fichier). Il affiche le contenu du fichier passé en argument ou de l'entrée standard. Pressez « h » si vous avez besoin d'une aide lors du parcours avec la commande `less`. Il peut faire beaucoup plus de choses que more(1) et peut être surchargé en exécutant « `eval $(lesspipe)` » ou « `eval $(lessfile)` » dans le script de démarrage de l'interpréteur de commandes. Vous trouverez davantage d'informations dans « `/usr/share/doc/less/LESSOPEN` ». L'option « `-R` » permet la sortie en mode caractères bruts et permet les séquences d'échappement de couleurs ANSI. Consultez less(1).

ASTUCE
Dans la commande `less`, saisir « h » pour voir l'écran d'aide, saisir « `/` » ou « ? » pour rechercher une chaine et saisir « `-i` » pour changer la sensibilité à la casse.

1.4.6 L'éditeur de texte

Il faudrait que vous soyez compétant avec l'une des variantes des programmes Vim ou Emacs qui sont très populaires sur les systèmes semblables à UNIX.

Je pense que s'habituer aux commandes de Vim est une bonne chose, car l'éditeur Vi est toujours présent dans le monde Linux et UNIX. (En pratique, le `vi` d'origine ou le nouveau `nvi` sont des programmes que vous trouvez partout. Pour les débutants, j'ai plutôt choisi Vim parce qu'il propose de l'aide par l'intermédiaire de la touche `F1` tout en restant assez semblable et plus puissant.)

Si vous choisissez plutôt Emacs ou XEmacs comme éditeur, c'est aussi un bon choix évidemment, particulièrement pour la programmation. Emacs possède une pléthore d'autres fonctionnalités, y compris un lecteur de nouvelles, un éditeur de répertoires, un programme de courriel, etc. Lorsqu'il est utilisé pour programmer ou éditer des scripts en shell, il reconnaît de manière intelligente le format de ce sur quoi vous être en train de travailler et il essaie de vous aider. Certaines personnes affirment que le seul programme dont ils ont besoin sous Linux est Emacs. Dix minutes d'apprentissage d'Emacs maintenant vous économiseront des heures plus tard. Il est grandement recommandé d'avoir le manuel de GNU Emacs comme référence lors de son apprentissage.

Tous ces programmes sont habituellement accompagnés d'un programme d'apprentissage pour vous aider à les utiliser par la pratique. Lancez Vim en entrant « `vim` » et en pressant la touche F1. Vous devriez au moins en lire

les 35 premières lignes. Suivez ensuite le cours en ligne en déplaçant le curseur sur « `|tutor|` » et en pressant `Ctrl-]`.

Note
De bons éditeurs, tels que Vim et Emacs, traitent correctement les textes codés en UTF-8 et autres codages exotiques. Il est préconisé d'utiliser l'environnement graphique en utilisant les paramètres linguistiques UTF-8 locaux et d'installer les programmes requis ainsi que les fontes assorties. Les éditeurs permettent de régler le codage des fichiers indépendamment de celui de l'environnement graphique. Veuillez vous référer à leur documentation relative au texte multi-octets.

1.4.7 Définir un éditeur de texte par défaut

Debian est fourni avec de nombreux éditeurs différents. Nous recommandons d'installer le paquet `vim`, comme indiqué ci-dessus.

Debian offre un accès unifié à l'éditeur par défaut du système par l'intermédiaire de la commande « `/usr/bin/editor` ». Cela permet à d'autres programmes (par exemple reportbug(1)) de pourvoir l'appeler. Vous pouvez le modifier par la commande qui suit :

```
$ sudo update-alternatives --config editor
```

Pour les débutants, je recommande de choisir « `/usr/bin/vim.basic` » plutôt que « `/usr/bin/vim.tiny` » car il prend en charge la mise en évidence de la syntaxe.

ASTUCE
De nombreux programmes utilisent les variables d'environnement « `$EDITOR` » ou « `$VISUAL` » pour décider de l'éditeur à utiliser (consultez Section 1.3.5 et Section 9.4.11). Pour des raisons de cohérence sur le système Debian, définissez-les à « `/usr/bin/editor` ». (Historiquement « `$EDITOR` » était défini à « `ed` » et « `$VISUAL` » était défini à « `vi` ».)

1.4.8 Utilisation de vim

Le vim(1) récent démarre de lui-même avec l'option raisonnable « `nocompatible` » et entre dans le mode `NORMAL`. 1

Veuillez utiliser le programme « `vimtutor` » pour apprendre à utiliser `vim` à l'aide d'un tutoriel interactif.

Le programme `vim` modifie son comportement lors de frappes de clavier selon le **mode**. La saisie de clavier dans un tampon se fait principalement dans les modes `INSERTION` et `REMPLACEMENT`. Le déplacement de curseur se fait principalement dans le mode `NORMAL`. La sélection interactive est faite dans le mode `VISUEL`. Saisir « `:` » dans le mode `NORMAL` change son **mode** au mode `Ex`. Le mode `Ex` accepte des commandes.

ASTUCE
Vim est fourni avec le paquet **Netrw**. Netrw gère la lecture et l'écriture de fichier, le parcours de répertoires à travers un réseau, et la navigation locale ! Essayez Netrw avec « `vim .` » (un point comme argument) et lisez son manuel avec « `:help netrw` ».

Pour une configuration plus poussée de `vim`, consultez Section 9.2.

1. Même l'ancien `vim` peut démarrer dans le mode sécurisé « `nocompatible` » en démarrant avec l'option « `-N` ».

mode	saisies de clavier	action
NORMAL	`:help|only`	affichage de l'aide
NORMAL	`:e filename.ext`	ouverture d'un nouveau tampon pour éditer `fichier.ext`
NORMAL	`:w`	écraser le fichier originel avec le tampon actuel
NORMAL	`:w filename.ext`	enregistrer le tampon actuel dans `fichier.ext`
NORMAL	`:q`	quitter `vim`
NORMAL	`:q!`	forcer l'abandon de `vim`
NORMAL	`:only`	clore toutes les autres fenêtres ouvertes
NORMAL	`:set nocompatible?`	vérifier si `vim` est dans le mode raisonnable `nocompatible`
NORMAL	`:set nocompatible`	utiliser `vim` avec le mode raisonnable `nocompatible`
NORMAL	`i`	entrer dans le mode `INSERTION`
NORMAL	`R`	entrer dans le mode `REMPLACEMENT`
NORMAL	`v`	entrer dans le mode `VISUEL`
NORMAL	`V`	entrer dans le mode `VISUEL LIGNE`
NORMAL	Ctrl-V	entrer dans le mode `VISUEL BLOC`
sauf `TERMINAL-JOB`	ESC-key	entrer dans le mode `NORMAL`
NORMAL	`:term`	entrer dans le mode `TERMINAL-JOB`
`TERMINAL-NORMAL`	`i`	entrer dans le mode `TERMINAL-JOB`
`TERMINAL-JOB`	Ctrl-`W N` (ou Ctrl-`\` Ctrl-`N`)	entrer dans le mode `TERMINAL-NORMAL`
`TERMINAL-JOB`	Ctrl-`W :`	entrer dans le mode `Ex` dans le mode `TERMINAL-NORMAL`

Table 1.16 – Liste des saisies de clavier basiques pour vim

1.4.9 Enregistrer les actions de l'interpréteur de commandes

La sortie d'une commande de l'interpréteur peut défiler, quitter votre écran et être définitivement perdue. C'est une bonne habitude d'enregistrer l'activité de l'interpréteur de commandes dans un fichier afin de la consulter plus tard. Ce type d'enregistrement est essentiel lorsque vous effectuez des tâches d'administration quelconques.

ASTUCE
Le nouveau Vim (version>=8.2) peut être utilisé pour enregistrer l'activité de l'interpréteur proprement en utilisant le mode `TERMINAL-JOB`. Consultez Section 1.4.8.

La méthode de base pour enregistrer l'activité de l'interpréteur de commandes est de la lancer sous script(1).

Essayez, par exemple, ce qui suit :

```
$ script
Script started, file is typescript
```

Lancez une commande quelconque sous `script`.

Pressez `Ctrl-D` pour quitter le `script`.

```
$ vim typescript
```

Consultez Section 9.1.1 .

1.4.10 Commandes UNIX de base

Apprenons les commandes UNIX de base. J'utilise ici « UNIX » dans son sens générique. Tous les clones d'UNIX proposent habituellement des commandes équivalentes. Le système Debian ne fait pas exception. Ne vous inquiétez pas si certaines commandes ne fonctionnent pas comme vous le voudriez maintenant. Si un `alias` est utilisé dans le shell, la sortie correspondante sera différente. Ces exemples ne sont pas destinés à être exécutés dans cet ordre.

Essayez toutes les commandes qui suivent en utilisant un compte non privilégié :

Note
UNIX a pour tradition de cacher les fichiers dont le nom commence par un « `.` ». Ce sont traditionnellement des fichiers qui contiennent des informations de configuration et des préférences de l'utilisateur.
Pour la commande `cd`, consultez bash-builtins(7).
Le visualisateur (« pager ») par défaut d'un système Debian non personnalisé est more(1) qui ne permet pas le défilement vers l'arrière. En installant le paquet `less` à l'aide de la ligne de commandes « `apt-get install less` », less(1) deviendra le visualisateur par défaut et vous pourrez faire défiler le texte vers l'arrière à l'aide des touches de curseur.
« `[` » et « `]` » dans l'expression rationnelle de la commande « `ps aux | grep -e "[e]xim4*"` » ci-dessus permet d'éviter une correspondance de `grep` avec lui-même. Le « `4*` » de l'expression rationnelle signifie 0 ou plusieurs instances du caractère « `4` » et permet donc à `grep` de trouver la correspondance à la fois avec « `exim` » et « `exim4` ». Bien que « `*` » soit utilisé dans le motif générique (« glob ») des noms de fichiers de l'interpréteur de commandes et dans l'expression rationnelle, leurs significations sont différentes. Vous pourrez apprendre les expressions rationnelles dans grep(1).

À titre d'exercice, parcourez les répertoires et jetez un coup d'œil au système en vous servant des commandes ci-dessus. Si vous avez des questions sur ces commandes de la console, veuillez consulter la page de manuel.

Essayez, par exemple, ce qui suit :

commande	description
`pwd`	afficher le nom du répertoire actuel ou de travail
`whoami`	afficher le nom de l'utilisateur actuel
`id`	afficher l'identité de l'utilisateur actuel (nom, uid, gid, et groupes associés)
`file foo`	afficher le type de fichier du fichier « *toto* »
`type -p commandname`	afficher l'emplacement du fichier de la commande « *nom-de-commande* »
`which commandname`	, ,
`type commandname`	afficher des informations sur la commande « *nom-de-commande* »
`apropos key-word`	rechercher les commandes ayant un rapport avec « *mot-clé* »
`man -k key-word`	, ,
`whatis commandname`	afficher une ligne d'explication sur la commande « *nom-de-commande* »
`man -a commandname`	afficher une explication sur la commande « *nom-de-commande* » (style UNIX)
`info commandname`	afficher une explication assez longue de la commande « *nom-de-commande* » (style GNU)
`ls`	afficher le contenu du répertoire (tous les fichiers et répertoires non cachés)
`ls -a`	afficher le contenu du répertoire (tous les fichiers et répertoires)
`ls -A`	afficher le contenu du répertoire (presque tous les fichiers et répertoires, par exemple sauter « `..` » et « `.` »)
`ls -la`	afficher tout le contenu du répertoire de façon détaillée
`ls -lai`	afficher tout le contenu du répertoire avec les numéros d'inœuds et les informations détaillées
`ls -d`	afficher tous les sous-répertoires du répertoire actuel
`tree`	afficher le contenu de l'arborescence des fichiers
`lsof foo`	afficher l'état d'ouverture du fichier « *toto* »
`lsof -p pid`	afficher les fichiers ouverts par le processus de numéro : « *pid* »
`mkdir foo`	créer le nouveau répertoire « *toto* » dans le répertoire en cours
`rmdir foo`	supprimer le répertoire « *toto* » du répertoire actuel
`cd foo`	allez au répertoire « *toto* » se trouvant dans le répertoire actuel ou dans le répertoire figurant dans la variable « `$CDPATH` »
`cd /`	aller au répertoire racine
`cd`	aller au répertoire personnel de l'utilisateur actuel
`cd /foo`	aller au répertoire de chemin absolu « */toto* »
`cd ..`	aller au répertoire parent
`cd ~foo`	aller au répertoire « *foo* » se trouvant dans le répertoire personnel de l'utilisateur
`cd -`	aller au répertoire précédent
`</etc/motd pager`	afficher le contenu de « `/etc/motd` » en utilisant le visualisateur (« pager ») par défaut
`touch junkfile`	créer un fichier vide « *fichier-poubelle* »
`cp foo bar`	copier le fichier « *toto* » existant dans le nouveau fichier « *titi* »
`rm junkfile`	supprimer le fichier « *fichier-poubelle* »
`mv foo bar`	renommer le fichier existant « *toto* » avec le nouveau nom « *titi* » (« *titi* » ne doit pas exister)
`mv foo bar`	déplacer le fichier existant « *toto* » vers le nouvel emplacement « *titi/toto* » (le répertoire « *titi* » doit exister)
`mv foo bar/baz`	déplacer le fichier existant « *toto* » vers un nouvel emplacement avec le nouveau nom « *titi/tutu* » (le répertoire « *titi* » doit exister mais le répertoire « *titi/tutu* » ne doit pas exister)
`chmod 600 foo`	rendre le fichier « *toto* » non lisible et non modifiable par les autres personnes (non exécutable pour tous)
`chmod 644 foo`	rendre un fichier existant « *toto* » accessible en lecture mais non modifiable par les autres personnes (non exécutable pour tous)
`chmod 755 foo`	rendre un fichier existant « *toto* » accessible en lecture mais non modifiable par les autres utilisateurs (exécutable pour tous)
`find / -name motif -print`	rechercher les noms de fichier contenant le « *motif* » de

```
$ man man
$ man bash
$ man builtins
$ man grep
$ man ls
```

Il peut être un peu difficile de s'habituer au style des pages de manuel parce qu'elles sont plutôt succinctes, particulièrement les plus anciennes, celles qui sont vraiment traditionnelles. Mais une fois que vous y serez familiarisé, vous apprécierez leur concision.

Remarquez que beaucoup de commandes UNIX, y compris celles de GNU et BSD, affichent une information d'aide courte si vous les exécutez de l'une des façons suivantes (ou parfois sans paramètre) :

```
$ commandname --help
$ commandname -h
```

1.5 La commande simple de l'interpréteur de commandes

Vous avez maintenant une certaine sensation sur la manière d'utiliser un système Debian. Nous allons regarder plus profondément le mécanisme d'exécution des commandes sous le système Debian. J'ai ici, pour les débutants, simplifié la réalité. Consultez bash(1) pour l'explication exacte.

Une simple commande est une séquence de :

1. assignations de variables (optionnelles) ;
2. nom de la commande ;
3. paramètres (optionnels) ;
4. redirections (optionnelles : > , >> , < , <<, etc.) ;
5. opérateurs de contrôle (optionnels : &&, || , *nouvelle ligne* , ;, &, (,)).

1.5.1 Exécution d'une commande et variables d'environnement

Les valeurs de certaines variables d'environnement modifient le comportement de certaines commandes UNIX.

Les valeurs par défaut des variables d'environnement sont définies initialement par le système PAM, certaines d'entre-elles peuvent donc être réinitialisées par certains programmes d'application :

— le système PAM tel que `pam_env` peut définir des variables d'environnement à l'aide de « `/etc/pam.conf` », « `/etc/environment` » et « `/etc/default/locale` » ;
— The modern GUI environment run from the user mode systemd unit (see systemd.unit(5)) sets its environment variables by a file in the "`~/.config/environment.d/`" directory.
— l'initialisation de programme spécifique à l'utilisateur peut réinitialiser les variables d'environnement à l'aide de « `~/.profile` », « `~/.bash_profile` » et `~/.bashrc` ».

1.5.2 La variable « `$LANG` »

Les paramètres régionaux par défaut sont définis dans la variable d'environnement « `$LANG` » et sont configurés comme « `LANG=xx_YY. UTF-8` » par l'installateur ou par la configuration ultérieure de l'interface graphique, par exemple, « Paramètres » → « Région & Langue » → « Langue » / « Formats » pour GNOME.

Note
Je vous recommande de configurer l'environnement du système en ne touchant qu'à la variable « `$LANG` » pour l'instant en laissant de côté les variables « `$LC_*` » à moins que ce ne soit absolument nécessaire.

---

valeur des paramètres linguistiques	signification
xx	codes de langue ISO 639 (en minuscules) tel que « fr »
YY	codes de pays ISO 3166 (en majuscules), par exemple « FR »
ZZZZ	le jeu de caractères, toujours défini à « UTF-8 »

Table 1.18 – Les trois parties des paramètres linguistiques

recommandation de paramètres linguistiques	Langue (zone)
en_US.UTF-8	anglais (USA)
en_GB.UTF-8	anglais (Grande-Bretagne)
fr_FR.UTF-8	français (France)
de_DE.UTF-8	allemand (Allemagne)
it_IT.UTF-8	italien (Italie)
es_ES.UTF-8	espagnol (Espagne)
ca_ES.UTF-8	catalan (Espagne)
sv_SE.UTF-8	suédois (Suède)
pt_BR.UTF-8	portugais (Brésil)
ru_RU.UTF-8	russe (Russie)
zh_CN.UTF-8	chinois (RP de Chine)
zh_TW.UTF-8	chinois (Taiwan)
ja_JP.UTF-8	japonais (Japon)
ko_KR.UTF-8	coréen (République de Corée)
vi_VN.UTF-8	vietnamien (Vietnam)

Table 1.19 – Liste des recommandations de paramètres linguistiques

La valeur complète des paramètres linguistiques indiqués par la variable « $LANG est constituée de trois parties « xx_YY.ZZZZ ».

L'exécution typique d'une commande utilise une séquence de lignes telle que la suivante :

```
$ echo $LANG
en_US.UTF-8
$ date -u
Wed 19 May 2021 03:18:43 PM UTC
$ LANG=fr_FR.UTF-8 date -u
mer. 19 mai 2021 15:19:02 UTC
```

Ici, le programme date(1) est exécuté avec différentes valeurs de paramètres régionaux :

— Avec la première commande, « $LANG » est définie à la valeur des paramètres linguistiques par défaut du système « fr_FR.UTF-8 ».

— Avec la deuxième commande, « $LANG » est définie à la valeur des paramètres linguistiques UTF-8 anglais des États-Unis « en_US.UTF-8 ».

Habituellement, la plupart des exécutions de commandes ne sont pas précédées de la définition de variables d'environnement. Pour les exemples ci-dessus, vous pouvez aussi exécuter :

```
$ LANG=fr_FR.UTF-8
$ date -u
mer. 19 mai 2021 15:19:24 UTC
```

ASTUCE
En remplissant un rapport de bogue, c'est une bonne idée de lancer et de vérifier la commande avec les paramètres régionaux « LANG=en_US.UTF-8 » si vous utilisez un environnement autre que l'environnement anglais.

Pour des informations détaillées sur la configuration des paramètres linguistiques, consultez Section 8.1.

1.5.3 La variable « $PATH »

Lorsque vous entrez une commande dans l'interpréteur, il recherche la commande dans la liste des répertoires contenus dans la variable d'environnement « $PATH ». La valeur de la variable d'environnement « $PATH » est aussi appelée « chemin de recherche de l'interpréteur de commandes ».

Dans une installation Debian par défaut, la variable d'environnement « $PATH » des comptes d'utilisateurs peut ne pas inclure « /usr/sbin » ni « /usr/sbin ». Par exemple, la commande `ifconfig` doit être lancée avec son chemin complet « /usr/sbin/ifconfig ». (La commande similaire `ip` est située dans « /usr/bin ».)

Vous pouvez modifier la variable d'environnement « $PATH » de l'interpréteur de commandes Bash par l'intermédiaire des fichiers« ~/.bash_profile » ou « ~/.bashrc ».

1.5.4 La variable « $HOME »

De nombreuses commandes enregistrent la configuration spécifique à un utilisateur dans son répertoire personnel et modifient leur comportement en fonction de son contenu. Le répertoire personnel est identifié par la variable d'environnement « $HOME ».

valeur de « $HOME »	situation d'exécution d'un programme
/	programme lancé par le processus init (démon)
/root	programme lancé depuis l'interpréteur de commandes normal de l'administrateur (« root »)
/home/*normal_user*	programme lancé depuis l'interpréteur de commandes d'un utilisateur normal
/home/*normal_user*	programme lancé depuis le menu du bureau de l'interface graphique de l'utilisateur
/home/*normal_user*	programme lancé en tant qu'administrateur avec « sudo programme »
/root	programme lancé en tant qu'administrateur avec « sudo -H programme »

Table 1.20 – Afficher les valeurs de la variable « $HOME »

ASTUCE
L'interpréteur de commandes étend « ~/ » pour former le répertoire personnel réel de l'utilisateur, par exemple, « $HOME/ ». L'interpréteur de commandes étend « ~foo/ » sous la forme du répertoire personnel de foo, par exemple, « /home/foo/ ».

Consultez Section 12.1.5 si $HOME n'est pas disponible pour votre programme.

1.5.5 Options de la ligne de commandes

Certaines commandes prennent des paramètres. Les paramètres qui commencent par « - » ou « -- » sont appelés options et contrôlent le comportement de la commande.

```
$ date
Thu 20 May 2021 01:08:08 AM JST
$ date -R
Thu, 20 May 2021 01:08:12 +0900
```

Ici, le paramètre de la ligne de commandes « -R » modifie le comportement de la commande date(1) afin qu'elle donne en sortie une chaîne de date conforme à la RFC2822.

1.5.6 Motifs génériques (« glob ») de l'interpréteur de commandes

Souvent, vous voudrez utiliser une commande sur un groupe de fichiers sans avoir à tous les entrer. C'est facilité par l'utilisation des motifs génériques d'expansion du nom de fichier de l'interpréteur de commandes (**glob**), (on les appelle parfois **jokers**).

motif générique de l'interpréteur	description de la règle de correspondance
*	nom de fichier ne commençant pas par « . »
.*	filename started with "." (in Bash 5.2 and later, "." and ".." are excluded by default)
?	exactement un caractère
[…]	exactement un caractère, chaque caractère étant l'un de ceux entre crochets
[a-z]	exactement un caractère, chaque caractère étant compris entre « a » et « z »
[!…]	exactement un caractère devant être différent de tous les caractères entre crochets (sauf « ! »)

Table 1.21 – Motifs génériques d'expansion du nom de fichier de l'interpréteur de commandes

Note
A "/" in a pathname cannot be matched by the shell glob pattern listed in the above table.

Essayez, par exemple, ce qui suit :

```
$ mkdir foo; cd foo; touch b1.txt b2.txt b3.c b4.h .b5.txt ..b6.txt
$ mkdir bar; cd bar; touch b1.txt b2.txt b3.c b4.h .b5.txt ..b6.txt ; cd -
$ echo b*.txt
b1.txt b2.txt
$ echo b*
b1.txt b2.txt b3.c b4.h bar
$ echo */*.txt
bar/b1.txt bar/b2.txt
$ echo *.[hc]
b3.c b4.h
$ echo .*
.b5.txt ..b6.txt
$ echo b[!1-3]*
b4.h bar
$ echo b*/b[!1-3]*
bar/b4.h
$ cd ..; rm -rf junk
```

Consultez glob(7).

Note
Contrairement à l'expansion normale du nom de fichier par l'interpréteur de commandes, le motif « * » de l'interpréteur testé par find(1) avec « -name » test etc., correspond au « . » du nom de fichier. (Nouvelle fonctionnalité POSIX.)

Note
On peut modifier le comportement d'expansion de fichiers selon des motifs génériques de BASH (« glob ») avec ses options « shopt » incluses telles que « dotglob », « noglob », « nocaseglob », « nullglob », « extglob », etc. Consultez bash(1).

1.5.7 Valeur de retour d'une commande

Toutes les commandes retournent comme valeur de retour leur état de fin d'exécution (variable : « $? »).

état de sortie de la commande	valeur numérique de retour	valeur logique de retour
succès	zéro, 0	**VRAI**
erreur	non-nulle, -1	**FAUX**

Table 1.22 – Codes de retour de la commande

Essayez, par exemple, ce qui suit :

```
$ [ 1 = 1 ] ; echo $?
0
$ [ 1 = 2 ] ; echo $?
1
```

Note
Vous remarquerez que, dans le contexte logique de l'interpréteur de commandes, un **succès** est traité comme la valeur logique **VRAIE** qui possède la valeur 0 (zéro). C'est parfois un peu contre-intuitif et il fallait le rappeler ici.

1.5.8 Séquences de commandes typiques et redirection de l'interpréteur de commandes

Essayez de retenir les idiomes suivants de l'interpréteur de commandes entrés sur une seule ligne en tant qu'extrait d'une commande de l'interpréteur.

Le système Debian est un système multi-tâches. Les travaux s'exécutant en arrière-plan permettent aux utilisateurs de faire tourner plusieurs programmes depuis un seul interpréteur de commandes. La gestion des processus en arrière-plan fait appel aux commandes internes de l'interpréteur : `jobs`, `fg`, `bg` et `kill`. Veuillez lire les sections de bash(1) se trouvant sous « SIGNAUX » et « CONTRÔLE DES TÂCHES » ainsi que bash-builtins(7).

Essayez, par exemple, ce qui suit :

```
$ </etc/motd pager
```

```
$ pager </etc/motd
```

```
$ pager /etc/motd
```

```
$ cat /etc/motd | pager
```

Bien que ces 4 exemples de redirections d'interpréteur de commande affichent la même chose, le dernier exemple utilise la commande supplémentaire `cat` et gaspille des ressources sans raison.

L'interpréteur de commande vous permet d'ouvrir des fichiers en utilisant la commande interne `exec` avec un descripteur de fichier arbitraire.

```
$ echo Hello >foo
$ exec 3<foo 4>bar  # open files
$ cat <&3 >&4       # redirect stdin to 3, stdout to 4
$ exec 3<&- 4>&-    # close files
$ cat bar
Hello
```

Les descripteurs de fichiers 0-2 sont prédéfinis.

idiome de commande	description
`command &`	exécuter la `commande` en **arrière-plan** dans le sous-shell
`command1 | command2`	la sortie standard de `commande1` est passée (« **pipe** ») à l'entrée standard de `commande2`. Les deux commandes peuvent tourner **simultanément**
`command1 2>&1 | command2`	La sortie standard et la sortie d'erreur standard de `commande1` sont toutes les deux passées à l'entrée standard de `commande2`. Les deux commandes peuvent tourner **simultanément**
`command1 ; command2`	exécuter `commande1` et `commande2` **séquentiellement**
`command1 && command2`	exécuter `commande1`, en cas de succès, exécuter `commande2` **séquentiellement** (retourne un succès si à la fois `commande1` **et** `commande2` ont été réussies)
`command1 || command2`	exécuter `commande1`, en cas d'échec, exécuter `commande2` **séquentiellement** (retourne un succès si `commande1` **ou** `commande2` a été réussie)
`command > foo`	rediriger la sortir standard de `commande` vers le fichier `toto` (l'écraser)
`command 2> foo`	rediriger la sortie d'erreur standard de la `commande` vers le fichier `toto` (et l'écraser)
`command >> foo`	rediriger la sortie standard de la `commande` vers le fichier `toto` (ajouter à la fin du fichier)
`command 2>> foo`	rediriger la sortie d'erreur standard de la `commande` vers le fichier `toto` (ajouter à la fin du fichier)
`command > foo 2>&1`	rediriger à la fois la sortie standard et l'erreur standard de la `commande` vers le fichier « `toto` »
`command < foo`	rediriger l'entrée standard de la `commande` vers le fichier `toto`
`command << delimiter`	rediriger l'entrée standard de la `commande` vers les lignes suivantes jusqu'à ce que le « `délimiteur` » soit rencontré (ce document)
`command <<- delimiter`	rediriger l'entrée standard de `commande` vers les lignes qui suivent jusqu'à ce que le « `délimiteur` » soit rencontré, les caractères de tabulation de tête sont supprimés des lignes d'entrée)

Table 1.23 – Idiomes des commandes de l'interpréteur

périphérique	description	descripteur de fichier
`stdin`	entrée standard	0
`stdout`	sortie standard	1
`stderr`	erreur standard	2

Table 1.24 – Descripteurs de fichier prédéfinis

1.5.9 Alias de commande

Vous pouvez définir des alias pour les commandes fréquemment utilisées.

Essayez, par exemple, ce qui suit :

```
$ alias la='ls -la'
```

Maintenant, « `la` » fonctionnera comme un raccourci pour « `ls -la` » qui donne la liste de tous les fichier dans le format de liste long.

Vous pouvez afficher la liste de tous les alias existants par la commande `alias` (consultez bash(1) sous « COMMANDES INTERNES DU SHELL »).

```
$ alias
...
alias la='ls -la'
```

Vous pouvez identifier le chemin exact ou identifier la commande par `type` (consultez bash(1) sous « COMMANDES INTERNES DU SHELL »).

Essayez, par exemple, ce qui suit :

```
$ type ls
ls is hashed (/bin/ls)
$ type la
la is aliased to ls -la
$ type echo
echo is a shell builtin
$ type file
file is /usr/bin/file
```

Ici, `ls` a été récemment recherché alors que « `file` » ne l'a pas été, donc « `ls` » est « hachée », c'est-à-dire que l'interpréteur de commandes possède un enregistrement interne permettant un accès rapide à l'emplacement de la commande « `ls` ».

ASTUCE
Consultez Section 9.3.6.

1.6 Traitement des données textuelles à la UNIX

Dans un environnement de travail à la UNIX, le traitement du texte est effectué en passant le texte par des tubes au travers d'une chaîne d'outils standards de traitement de texte. C'est une autre innovation cruciale d'UNIX.

1.6.1 Outils de traitement de texte d'UNIX

Il existe quelques outils standard de traitement de texte qui sont très souvent utilisés sur les systèmes « UNIX-like ».

- Aucune expression rationnelle n'est utilisée :
  - cat(1) concatène des fichiers et en affiche le contenu complet ;
  - tac(1) concatène des fichiers et les affiche en ordre inverse ;
  - cut(1) sélectionne des parties de lignes et les affiche ;
  - head(1) affiche le début d'un fichier ;
  - tail(1) affiche la fin d'un fichier ;
  - sort(1) trie des lignes de texte ;

  - uniq(1) supprime les lignes dupliquées d'un fichier trié ;
  - tr(1) traduit ou supprime des caractères ;
  - diff(1) compare des fichiers ligne par ligne.
- Une expression rationnelle basique (**BRE**) est utilisée par défaut :
  - ed(1) est un éditeur par ligne primitif ;
  - sed(1) est un éditeur de flux ;
  - grep(1) analyse la correspondance d'un texte avec des motifs ;
  - vim(1) est un éditeur en mode écran ;
  - emacs(1) est un éditeur en mode écran (un peu étendu **BRE**).
- Une expression rationnelle étendue (**ERE**) est utilisée :
  - awk(1) effectue un traitement simple du texte ;
  - egrep(1) fait correspondre du texte avec des motifs ;
  - `tcl`(3tcl) peut effectuer tous les traitements possibles du texte : consulter re_syntax(3). Souvent utilisé avec `tk`(3tk) ;
  - perl(1) peut effectuer tous les traitements imaginables sur du texte. Consulter perlre(1) ;
  - pcre2grep(1) du paquet `pcre2-utils` fait la correspondance de texte avec des motifs d'expressions rationnelles compatibles avec Perl (PCRE) (« Perl Compatible Regular Expressions ») ;
  - python3(1) avec le module `re` peut faire tous les traitements imaginables sur du texte. Consultez « `/usr/share/doc/`

Si vous n'êtes pas certain de ce que font exactement ces commandes veuillez utiliser la commande « `man` » pour vous en faire une idée par vous-même.

Note
L'ordre de tri et l'étendue de l'expression dépendent de la locale. Si vous souhaitez obtenir un comportement traditionnel pour une commande, utilisez la locale **C** ou **C.UTF-8** au lieu des locales normales **UTF-8** (voir Section 8.1).

Note
Les expressions rationnelles de Perl (perlre(1)), Expressions rationnelles compatible avec Perl (PCRE) et les expressions rationnelles de Python proposées par le module `re` ont de nombreuses extensions courantes par rapport aux expressions rationnelles étendues **ERE**.

1.6.2 Expressions rationnelles

Les expressions rationnelles sont utilisées avec de nombreux outils de traitement du texte. Elles sont analogues aux motifs génériques « globs » du shell mais elles sont plus compliquées et plus puissantes.

L'expression rationnelle décrit le motif de correspondance, elle est constituée de caractères de texte et de **métacaractères**.

Un **métacaractère** est simplement un caractère ayant une signification particulière. Il en existe deux styles principaux, **BRE** et **ERE** suivant les outils de texte décrits ci-dessus.

Une expression rationnelle d'**`emacs`** est **BRE de base** mais elle a été étendue afin de traiter « + » et « ? » comme des **métacaractères** comme dans les **ERE**. Il n'est dont pas nécessaire de les échapper avec « `\` » dans les expressions rationnelles d'`emacs`.

grep(1) peut être utilisé pour effectuer de la recherche de texte en utilisant une expression rationnelle.

Essayez, par exemple, ce qui suit :

```
$ egrep 'GNU.*LICENSE|Yoyodyne' /usr/share/common-licenses/GPL
GNU GENERAL PUBLIC LICENSE
GNU GENERAL PUBLIC LICENSE
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
```

BRE	ERE	description de l'expression rationnelle
`\ . [ ] ^ $ *`	`\ . [ ] ^ $ *`	**métacaractères** courants
`\+ \? \( \) \{ \} |`		**métacaractères** uniquement BRE, déspécifiés par « \ »
	`+ ? ( ) { } |`	**métacaractères** uniquement ERE, non déspécifiés par « \ »
c	c	correspond au **non métacaractère** « c »
`\c`	`\c`	correspond au caractère littéral « c » même si « c » est un **métacaractère** en lui-même
`.`	`.`	correspond à n'importe quel caractère y compris le saut de ligne
`^`	`^`	position au début d'une chaîne de caractères
`$`	`$`	position à la fin d'une chaîne de caractères
`\<`	`\<`	position au début d'un mot
`\>`	`\>`	position à la fin d'un mot
`[abc…]`	`[abc…]`	correspond à n'importe quel caractère dans « `abc…` »
`[^abc…]`	`[^abc…]`	correspond à n'importe quel caractère sauf ceux se trouvant dans « `abc…` »
`r*`	`r*`	correspond à aucune ou plusieurs instances de l'expression rationnelle identifiée par « `r` »
`r\+`	`r+`	correspond à une ou plusieurs instances de l'expression rationnelle identifiée par « `r` »
`r\?`	`r?`	correspond à aucune ou une instance de l'expression rationnelle identifiée par « `r` »
`r1|r2`	`r1|r2`	correspond à une instance de l'expression rationnelle identifiée par « `r1` » ou« `r2` »
`\(r1|r2\)`	`(r1|r2)`	correspond à une des expressions rationnelles identifiées par « `r1` » ou « `r2` » et la traite comme une expression rationnelle **entre crochets**

Table 1.25 – Métacaractères pour BRE et ERE

ASTUCE
Consultez Section 9.3.6.

1.6.3 Expressions de remplacement

Pour l'expression de remplacement, certains caractères ont une signification particulière.

expressions de remplacement	description du texte destiné à remplacer l'expression de replacement
`&`	ce à quoi correspond l'expression rationnelle (utilisez `\&` avec `emacs`)
`\n`	ce à quoi la nième expression rationnelle **entre crochets** correspond (« n » étant un nombre)

Table 1.26 – Expressions de remplacement

Pour chaîne de caractères Perl de remplacement, "$&" est utilisé au lieu de "&" et "$n" est utilisé au lieu de "\n".

Essayez, par exemple, ce qui suit :

```
$ echo zzz1abc2efg3hij4 | \
sed -e 's/\(1[a-z]*\)[0-9]*\(.*\)$/=&=/'
zzz=1abc2efg3hij4=
$ echo zzz1abc2efg3hij4 | \
sed -E -e 's/(1[a-z]*)[0-9]*(.*)$/=&=/'
zzz=1abc2efg3hij4=
$ echo zzz1abc2efg3hij4 | \
perl -pe 's/(1[a-z]*)[0-9]*(.*)$/=$&=/'
zzz=1abc2efg3hij4=
$ echo zzz1abc2efg3hij4 | \
sed -e 's/\(1[a-z]*\)[0-9]*\(.*\)$/\2===\1/'
zzzefg3hij4===1abc
$ echo zzz1abc2efg3hij4 | \
sed -E -e 's/(1[a-z]*)[0-9]*(.*)$/\2===\1/'
zzzefg3hij4===1abc
$ echo zzz1abc2efg3hij4 | \
perl -pe 's/(1[a-z]*)[0-9]*(.*)$/$2===$1/'
zzzefg3hij4===1abc
```

Vous prêterez ici une attention particulière au style de l'expression rationnelle **entre crochets** et à la manière dont les chaînes qui correspondent sont utilisées dans le processus de remplacement du texte avec les différents outils.

Ces expressions rationnelles peuvent aussi être utilisées pour les déplacements du curseur et des actions de remplacement de texte dans certains éditeurs.

Un « back slash » (« \ ») en fin de ligne sur la ligne de commandes du shell déspécifie le saut de ligne en un caractère d'espacement et permet de poursuivre la ligne de commandes de l'interpréteur sur la ligne suivante.

Pour apprendre ces commandes, veuillez lire les pages de manuel correspondantes.

1.6.4 Substitution globale avec des expressions rationnelles

La commande ed(1) peut remplacer toutes les instances de « `REGEX_SOURCE` » par « `TEXTE_DESTINATION` dans « `fichier` » :

```
$ ed file <<EOF
,s/FROM_REGEX/TO_TEXT/g
w
q
EOF
```

La commande sed(1) peut remplacer toutes les instances de « REGEX_SOURCE » par « TEXTE_DESTINATION dans « fichier » :

```
$ sed -i -e 's/FROM_REGEX/TO_TEXT/g' file
```

La commande vim(1) peut remplacer toutes les instances de « REGEX_SOURCE » avec « TEXTE_DESTINATION » dans « file » en utilisant les commandes « ex(1) :

```
$ vim '+%s/FROM_REGEX/TO_TEXT/gc' '+update' '+q' file
```

ASTUCE
L'indicateur « c » dans l'exemple qui précède permet une confirmation interactive de chaque substitution.

Plusieurs fichiers (fichier1 », « fichier2 » et « fichier3 ») peuvent être traités de manière similaire par une expression rationnelle avec vim(1) ou perl(1) :

```
$ vim '+argdo %s/FROM_REGEX/TO_TEXT/gce|update' '+q' file1 file2 file3
```

ASTUCE
L'indicateur « e » de l'exemple précédent évite qu'une erreur « No match » (pas de correspondance) ne casse un mapping :

```
$ perl -i -p -e 's/FROM_REGEX/TO_TEXT/g;' file1 file2 file3
```

Dans l'exemple en perl(1) ci-dessus, le « -i » force l'édition directe de chaque fichier cible, et le « -p » induit une boucle implicite sur tous les fichiers.

ASTUCE
L'utilisation du paramètre « -i.bak » à la place de « -i » conserve le fichier d'origine en ajoutant « .bak » à son nom de fichier. Cela permet la récupération plus facile d'erreurs lors de substitutions complexes.

Note
ed(1) et vim(1) sont **BRE** ; perl(1) est **ERE**.

1.6.5 Extraire des données d'un tableau contenu dans un fichier texte

Considérons un fichier texte « DPL » dans lequel les noms de certains des responsables du projet Debian d'avant 2004 et leur date d'investiture sont présentés dans un format séparés par des espaces :

```
Ian     Murdock   August  1993
Bruce   Perens    April   1996
Ian     Jackson   January 1998
Wichert Akkerman  January 1999
Ben     Collins   April   2001
Bdale   Garbee    April   2002
Martin  Michlmayr March   2003
```

ASTUCE
Consultez « Bref historique de Debian » pour la dernière Debian leadership history.

Awk est fréquemment utilisé pour extraire des données de ce type de fichiers.

Essayez, par exemple, ce qui suit :

```
$ awk '{ print $3 }' <DPL                  # month started
August
April
January
January
April
April
March
$ awk '($1=="Ian") { print }' <DPL         # DPL called Ian
Ian     Murdock   August  1993
Ian     Jackson   January 1998
$ awk '($2=="Perens") { print $3,$4 }' <DPL # When Perens started
April 1996
```

Des interpréteurs de commandes comme Bash peuvent aussi être utilisés pour analyser ce genre de fichiers.

Essayez, par exemple, ce qui suit :

```
$ while read first last month year; do
    echo $month
  done <DPL
... same output as the first Awk example
```

Ici, la commande interne `read` utilise les caractères de « `$IFS` » (séparateurs de champs internes) pour scinder les lignes en mots.

Si vous changez « `$IFS` » en « `:` », vous pouvez analyser facilement le fichier « `/etc/passwd` » :

```
$ oldIFS="$IFS"   # save old value
$ IFS=':'
$ while read user password uid gid rest_of_line; do
    if [ "$user" = "bozo" ]; then
      echo "$user's ID is $uid"
    fi
  done < /etc/passwd
bozo's ID is 1000
$ IFS="$oldIFS"   # restore old value
```

(Si Awk est utilisé pour faire la même chose, utilisez « `FS=':'` » pour définir le séparateur de champs.)

IFS est aussi utilisé par l'interpréteur de commandes pour scinder le résultat de l'expansion des paramètres, de la substitution de commande et de l'expansion arithmétique. Cela ne se produit pas pour les mots entre double ou simple apostrophes. La valeur de IFS par défaut est *space*, *tab* et *newline* combinés.

Faites attention en utilisant cette astuce IFS dans un shell. Des choses étranges peuvent survenir lorsque le shell interprète certaines parties du script comme son **entrée**.

```
$ IFS=":,"                        # use ":" and "," as IFS
$ echo IFS=$IFS,   IFS="$IFS"     # echo is a Bash builtin
IFS=  , IFS=:,
$ date -R                         # just a command output
Sat, 23 Aug 2003 08:30:15 +0200
$ echo $(date -R)                 # sub shell --> input to main shell
Sat  23 Aug 2003 08 30 36 +0200
$ unset IFS                       # reset IFS to the default
$ echo $(date -R)
Sat, 23 Aug 2003 08:30:50 +0200
```

1.6.6 Bouts de scripts pour les tubes

Les scripts suivants font des choses sympas avec les tubes.

fragment de script (à entrer sur une seule ligne)	**effet de la commande**
`find /usr -print`	rechercher tous les fichiers se trouvant sous « `/usr` »
`seq 1 100`	imprimer 1 à 100
`| xargs -n 1 command`	lancer la commande de manière répétitive en utilisant chaque élément provenant du tube (pipe) comme paramètre
`| xargs -n 1 echo`	scinder les éléments séparés par des espaces provenant du tube (pipe) en différentes lignes
`| xargs echo`	concaténer les lignes provenant du tube en une seule ligne
`| grep -e regex_pattern`	extraire du tube les lignes contenant *motif_expression_rationnelle*
`| grep -v -e regex_pattern`	extraire du tube les lignes ne contenant pas *motif_expression_rationnelle*
`| cut -d: -f3 -`	extraire du tube le troisième champ, séparé par « `:` » (fichier passwd, etc.)
`| awk '{ print $3 }'`	extraire du tube le troisième champ séparé par des caractères d'espacement
`| awk -F'\t' '{ print $3 }'`	extraire du tube le troisième champ séparé par une tabulation
`| col -bx`	supprimer le retour arrière (backspace) et convertir les tabulations en espaces
`| expand -`	convertir les tabulations en espaces
`| sort| uniq`	trier et supprimer les doublons
`| tr 'A-Z' 'a-z'`	convertir de majuscules en minuscules
`| tr -d '\n'`	concaténer les lignes en une seule ligne
`| tr -d '\r'`	supprimer le retour à la ligne (CR)
`| sed 's/^/# /'`	ajouter « # » au début de chaque ligne
`| sed 's/\.ext//g'`	supprimer « `.ext` »
`| sed -n -e 2p`	afficher la seconde ligne
`| head -n 2 -`	afficher les deux premières lignes
`| tail -n 2 -`	afficher les deux dernières lignes

Table 1.27 – Liste de parties de scripts pour enchaîner (piping) les commandes

Un script de l'interpréteur d'une seule ligne peut reboucler sur de nombreux fichiers en utilisant find(1) et xargs(1) afin d'effectuer des tâches assez complexes. Consultez Section 10.1.5 et Section 9.4.9.

Lorsque l'utilisation de l'interpréteur de commandes en mode interactif devient trop compliquée, pensez à écrire un script en shell (consultez Section 12.1).

Chapitre 2

Gestion des paquets Debian

Note
Ce chapitre a été écrit en supposant que le nom de code de la dernière version stable est `Trixie`.
La source des données du système APT est collectivement appelée **la liste des sources** dans ce document. Elle peut être définie n'importe où dans le fichier "`/etc/apt/sources.list`", les fichiers "`/etc/apt/sources.list.d/*.list`" ou les fichiers "`/etc/apt/sources.list.d/*.sources`".

2.1 Prérequis pour la gestion des paquets Debian

2.1.1 Système de gestion des paquets Debian

Debian est une association de volontaires qui construit des distributions **cohérentes** de paquets binaires de logiciels libres pré-compilés et les distribue depuis son archive.

L'archive Debian est proposée depuis de nombreux sites-miroirs distants, on peut y accéder par les méthodes HTTP et FTP. Elle est aussi disponible sous forme de CD-ROM/DVD.

Le système actuel de gestion des paquets de Debian qui peut utiliser toutes ces ressources est Advanced Packaging Tool (APT).

Le système de gestion des paquets Debian, **lorsqu'il est proprement utilisé**, permet à l'utilisateur d'installer sur le système des **ensembles cohérents de paquets binaires** à partir de l'archive. Il y a actuellement 78709 paquets disponibles pour l'architecture amd64.

Le système de gestion des paquets de Debian possède un riche historique et de nombreux choix de programmes d'interface pour l'utilisateur final et de méthodes de fond pour l'accès aux archives. Actuellement, nous recommandons ce qui suit :

— apt(8) pour toutes les opérations interactives en ligne de commande, y compris l'installation et la suppression de paquets et la mise à niveau de la distribution.
— apt-get(8) pour appeler le système de gestion de paquet Debian depuis des scripts. C'est également une option de rechange lorsqu'`apt` n'est pas disponible (souvent le cas avec d'anciens systèmes Debian).
— aptitude(8) pour une interface interactive en mode texte permettant de gérer les paquets installés et de faire des recherches parmi les paquets disponibles.

2.1.2 Configuration de paquets

Voici quelques points-clés de la configuration des paquets sur un système Debian :

paquet	popularité	taille	description
dpkg	V:909, I:1000	5086	système de gestion des paquets de bas niveau pour Debian (basé sur les fichiers)
apt	V:882, I:1000	5025	frontaux d'APT pour gérer les paquets en ligne de commande : apt/apt-get/apt-cache
aptitude	V:32, I:154	4837	frontal d'APT pour gérer interactivement les paquets avec une console plein écran : aptitude(8)
tasksel	V:42, I:985	352	frontal d'APT pour installer les tâches sélectionnées : tasksel(8)
unattended-upgrades	V:142, I:190	374	paquet d'amélioration d'APT permettant une installation automatique des mises à niveau de sécurité
gnome-software	V:159, I:260	4732	gestionnaire de logiciels pour GNOME (interface graphique pour APT)
synaptic	V:31, I:262	7264	gestionnaire de paquets graphique (frontal en GTK pour APT)
apt-utils	V:383, I:996	1187	Programmes utilitaires d'APT : apt-extracttemplates(1), apt-ftparchive(1) et apt-sortpkgs(1)
apt-listchanges	V:451, I:895	563	outil de notification des modifications d'un paquet
apt-listbugs	V:4.7, I:6.4	512	affiche la liste des bogues critiques avant chaque installation par APT
apt-file	V:14, I:52	76	utilitaire de recherche de paquet d'APT — interface en ligne de commandes
apt-rdepends	V:0.4, I:4.2	39	afficher de manière récursive la liste des dépendances du paquet

Table 2.1 – Liste des outils de gestion des paquets de Debian

— For the modern Desktop environment, rebooting of the system after package configuration change and package upgrade is a good idea to ensure the system to function properly.
— la configuration manuelle effectuée par l'administrateur du système est respectée. En d'autres termes, le système de configuration des paquets effectue, pour des raisons de commodité, une configuration non intrusive ;
— chaque paquet possède son propre script de configuration avec une interface utilisateur standardisée appelée debconf(7) qui permet de faciliter le processus initial d'installation du paquet ;
— les développeurs Debian font de leur mieux pour que vos mises à jour se fassent de manière impeccable avec les scripts de configuration du paquet ;
— l'administrateur du système peut utiliser toutes les fonctionnalités des paquets de logiciels. Cependant, celles qui présentent un risque de sécurité sont désactivées lors de l'installation par défaut ;
— si vous activez vous-même un service qui présente certains risques de sécurité, vous êtes responsable du confinement du risque ;
— Des configurations ésotériques peuvent être activées manuellement par l'administrateur du système. Cela peut créer une interférence avec les programmes génériques d'assistance à la configuration du système.

2.1.3 Précautions de base



AVERTISSEMENT
Ne pas installer de paquets provenant d'un mélange aléatoire de suites. Cela va probablement casser la cohérence des paquets et demande une connaissance en profondeur de la gestion du système, comme l'ABI, d'un compilateur, la version d'une bibliothèque, les fonctionnalités d'un interpréteur, etc.

Le nouveau administrateur du système Debian devrait rester avec la **stable** version de Debian tout en appliquant seulement les mises à jour de sécurité. Jusqu'à ce que vous compreniez très bien le système Debian, vous devriez suivre les précautions suivantes.

— Ne pas inclure **`testing`** ou **`unstable`** dans **la liste des sources**.
— Ne mélangez pas les archives Debian standard avec d'autres archives non Debian comme Ubuntu dans **la liste des sources** .
— ne pas créer le fichier « `/etc/apt/preferences` » ;
— ne pas modifier le comportement par défaut des outils de gestion des paquets au travers des fichiers de configuration sans en connaître toutes les conséquences ;
— ne pas installer de paquets quelconques à l'aide de « `dpkg -i` *`paquet_quelconque`* » ;
— ne jamais installer de paquets quelconques à l'aide de « `dpkg --force-all -i` *`paquet_quelconque`* » ;
— ne pas effacer ni modifier les fichiers se trouvant dans « `/var/lib/dpkg/` » ;
— ne pas écraser les fichiers systèmes en installant des logiciels directement depuis les sources.
  — Au besoin, les installer dans « `/usr/local` » ou « `/opt` ».

Les effets de non-compatibilité provoqués par la violation des précautions ci-dessus concernant le système de gestion des paquets Debian peuvent rendre votre système inutilisable.

L'administrateur système Debian sérieux, qui s'occupe de serveurs dont la mission est critique, devra prendre des précautions supplémentaires :

— ne pas installer de paquets, y compris les mises à jour de sécurité provenant de Debian sans les avoir testés soigneusement, avec votre configuration particulière, dans des conditions sûres.
  — Vous êtes finalement, en tant qu'administrateur système, responsable de votre système.
  — La longue histoire de stabilité du système Debian n'est pas, en elle-même, une garantie.

2.1.4 La vie avec d'éternelles mises à jour



Attention
Pour votre **serveur de production**, la suite `stable` avec les mises à jour de sécurité est recommandée. On peut dire la même chose des PC de bureau sur lesquels vous ne pouvez dépenser que des efforts limités d'administration.

Malgré mes avertissements ci-dessus, je sais que de nombreux lecteurs de ce document peuvent souhaiter exécuter les nouvelles distributions `testing` ou `unstable`.

La Lumière provenant de ce qui suit sauvera une personne de l'éternelle lutte karmique de l'enfer des mises à jour et lui permettra d'atteindre le nirvana de Debian.

Cette liste est faite pour un environnement de bureau **auto-administré**.

— Utilisez la publication `testing` car il s'agit pratiquement de la version continue gérée automatiquement par l'infrastructure d'assurance qualité de l'archive Debian telle que l'intégration continue de Debian , les pratiques de téléversement de sources uniquement et le suivi des transitions de bibliothèque. Les packages de la publication `testing` sont mis à jour assez fréquemment pour offrir toutes les fonctionnalités les plus récentes.
— Définir le nom de code correspondant à la suite `testing` ("`forky`" pendant le cycle de publication `trixie-as-stable`) dans **le répertoire des sources**.
— Mettez manuellement à jour ce nom de code dans **la liste des sources** vers le nouveau nom seulement après avoir évalué la situation par vous-même pendant environ un mois après la publication de la suite majeure. Les listes de diffusion des utilisateurs et des développeurs de Debian sont de bonnes sources d'information pour cela aussi.

L'utilisation de la publication `unstable` n'est pas recommandée. La publication `unstable` est **destinée au débogage des paquets** en tant que développeur, mais vous expose à des risques inutiles pour un utilisation normale comme bureau. Même si la suite `unstable` du système Debian semble très stable la plupart du temps, il y a eu quelques problèmes avec des paquets, et certains d'entre eux n'étaient pas si faciles à résoudre.

Voici quelques idées de mesures de précaution basiques pour assurer une récupération rapide et facile lors de bogues dans les paquets Debian :

— Make the rescue system available by following Section 3.2.
— faites un système avec un **double démarrage** en installant la suite `stable` du système Debian sur une autre partition ;
— pensez à installer `apt-listbugs` afin de vérifier les informations du Système Debian de suivi des bogues (BTS) avant de faire une mise à jour ;
— apprenez suffisamment l'infrastructure du système de paquets pour contourner le problème ;



Attention
Si vous ne savez pas faire l'une quelconque de ces actions de précaution, vous n'êtes probablement pas prêt pour les versions `testing` et `unstable`.

2.1.5 Bases concernant l'archive Debian

ASTUCE
La charte officielle de l'archive Debian est définie dans la Charte Debian, chapitre 2 - l'archive Debian.

Jetez un œil sur l'archive Debian avec le point de vue d'un utilisateur du système.

Pour un utilisateur du système, l'archive Debian est accessible à l'aide du système APT.

Le système APT spécifie sa source de données comme **la liste des sources** et elle est décrite dans sources.list(5).

For the `trixie` system with the typical HTTP access, **the source list** is provided in the modern deb822-style in "`/etc/apt/sources.list.d/debian.sources`" as the following:

```
Types: deb deb-src
URIs: http://deb.debian.org/debian/
Suites: trixie trixie-updates
Components: main non-free-firmware contrib non-free

Types: deb deb-src
URIs: http://security.debian.org/debian-security/
Suites: trixie-security
Components: main non-free-firmware contrib non-free
```

ASTUCE
If **the source list** is provided in the older deprecated one-line-style in "`/etc/apt/sources.list`" or "`/etc/apt/sources.list.d/*.list`" files, update them with:

```
$ sudo apt modernize-sources
```

Les points clés de **la liste des sources** au format deb822 sont les suivants.

— Ses fichiers de définition se trouvent dans les fichiers "`/etc/apt/sources.list.d/*.sources`".
— Chaque bloc de lignes séparé par une ligne vierge définit la source de données pour le système APT.
— La strophe "`Types:`" définit la liste des types tels que "`deb`" et "`deb-src`".
— La strophe "`URIs:`" définit la liste des URIs racine de l'archive Debian.
— La section "`Suites:`" définit la liste des noms de distribution en utilisant soit le nom de la suite, soit le nom de code.
— La section " `Composants:` " définit la liste des noms de zones d'archivage valides de l'archive Debian.

La définition de "deb-src" peut être omise sans risque si elle ne concerne que aptitude qui n'accède pas aux métadonnées relatives aux sources. Il accélère la mise à jour des métadonnées de l'archive.

L'URL peut être "https://", "http://", "ftp://", "file://", ...

Les lignes commençant par "#" sont des commentaires et sont ignorées.

Ici, j'ai tendance à utiliser le nom de code "trixie" ou "forky" au lieu du nom de suite "stable" ou "testing" pour éviter les surprises lors de la sortie de la prochaine stable.

ASTUCE
Si "sid" est utilisé dans l'exemple ci-dessus au lieu de "trixie", la ligne "deb: http://security.debian.org/ …" ou son contenu équivalent deb822 pour les mises à jour de sécurité dans **la liste des sources** n'est pas nécessaire. En effet, il n'existe pas d'archive de mise à jour de sécurité pour "sid" (unstable).

Voici la liste des URL des sites de l'archive Debian et du nom de la suite ou du nom de code utilisé dans le fichier de configuration après la publication de trixie.

URL de l'archive	nom de la suite	nom de code	but du dépôt
http://deb.debian.org/debian/	stable	trixie	publication stable quasi statique après des vérifications intensives
http://deb.debian.org/debian/	stable-updates	trixie-updates	Updates for the stable suite (important)
http://security.debian.org/debian-security/	stable-security	trixie-security	Security updates for the stable suite (important)
http://deb.debian.org/debian/	testing	forky	publication testing évolutive avec de sérieuses vérifications et un délai d'attente
http://deb.debian.org/debian/	unstable	sid	distribution unstable évolutive avec des vérifications minimales et sans délai d'attente
http://deb.debian.org/debian/	experimental	N/A	expérimentation d'une pré-publication par des développeurs (facultative et réservée aux développeurs)
http://deb.debian.org/debian/	stable-proposed-updates	trixie-proposed-updates	mises à jour pour la prochaine version stable intermédiaire (facultatif)
http://deb.debian.org/debian/	stable-backports	trixie-backports	collection arbitraire de paquets recompilés, la plupart issus de la publication testing (facultatif)
http://security.debian.org/debian-security/	testing-security	forky-security	non géré activement ni utilisé par l'équipe de sécurité

Table 2.2 – Liste des sites d'archive de Debian

Attention
Seule la version **stable** pure avec les mises à jour de sécurité présente la meilleure stabilité. Faire tourner une version principalement **stable** mélangée à quelques paquets venant des versions **testing** ou **unstable** est plus risqué que d'utiliser une version **unstable** pure parce que des versions de bibliothèques peuvent ne pas correspondre, etc. Si vous avez réellement besoin de la dernière version de certains programmes sous la version **stable**, utilisez alors les paquets venant de stable-updates et backports (consulter Section 2.7.4). Ces services doivent être utilisés avec des précautions supplémentaires.



Attention
De base, vous ne devriez avoir qu'une seule des suites `stable`, `testing` ou `unstable` sur la ligne « `deb` ». Si vous avez une combinaison des suites `stable`, `testing` et `unstable` sur la ligne « `deb` », les programmes APT vont être ralentis bien que seule la dernière archive soit utilisée. Des mentions multiples ont un intérêt lorsqu'on utilise le fichier « `/etc/apt/preferences` » avec des objectifs clairs (consultez Section 2.7.7).

ASTUCE
Pour un système Debian avec la suite `stable`, c'est une bonne idée d'inclure "`http://security.debian.org/`" dans **la liste des source** pour activer les mises à jour de sécurité comme dans l'exemple ci-dessus.

Note
Les bogues de sécurité de l'archive `stable` sont corrigés par l'équipe de sécurité de Debian. Cette activité a été assez rigoureuse et fiable. Ceux de l'archive `testing` peuvent être corrigés par l'équipe de sécurité de Debian. Pour diverses raisons, cette activité n'est pas aussi rigoureuse que pour `stable` et vous pouvez avoir à attendre la migration de paquets corrigés de `unstable` vers l'archive `testing`. Ceux de l'archive `unstable` sont corrigés par les responsables individuels. Les paquets de `unstable` maintenus de manière active sont habituellement maintenus dans un assez bon état par mise à niveau avec les dernières corrections de sécurité des développeurs amonts. Consultez FAQ de sécurité de Debian concernant la manière dont Debian gère les bogues de sécurité.

section	nombre de paquets	critères de composant du paquet
`main`	77218	conforme à DFSG sans dépendance vers `non-free`
`non-free-firmware`	85	non compatible DFSG, microprogramme requis pour une expérience satisfaisante de l'installation du système
`contrib`	365	conforme à DFSG mais avec des dépendances vers `non-free`
`non-free`	1041	non conforme à DFSG et non dans `non-free-firmware`

Table 2.3 – Liste des sections de l'archive de Debian

Ici, le nombre de paquets est celui de l'architecture amd64. La section `main` fournit le système Debian (consultez Section 2.1.6).

La meilleure manière d'étudier l'organisation de l'archive Debian est de pointer votre navigateur vers chacune des URL des archives en y ajoutant `dists` ou `pool`.

On se réfère à la distribution de deux manières, la version ou le nom de code. Le mot « distribution » est aussi utilisé comme synonyme de version dans de nombreuses documentations. La relation entre la version et le nom de code peut être résumée comme suit :

calendrier	version = `stable`	version = `testing`	version = `unstable`
après la diffusion de `trixie`	nom de code = `trixie`	nom de code = `forky`	nom de code = `sid`
après la diffusion de `forky`	nom de code = `forky`	nom de code = `duke`	nom de code = `sid`

Table 2.4 – Relation entre version et nom de code

L'histoire des noms de code a été décrite dans la FAQ Debian : 6.2.1 Quels noms de code ont déjà été utilisés ?

Dans la terminologie la plus stricte de l'archive Debian, le mot « section » est spécifiquement utilisé pour la catégorisation des paquets par zone d'application. (Cependant l'expression « section principale » peut parfois être utilisée pour décrire la section de l'archive Debian qui fournit la zone « main ».)

Chaque fois qu'un nouveau chargement est fait par un développeur Debian (DD) vers l'archive `unstable` (par l'intermédiaire du traitement d'incoming), le DD doit s'assurer que les paquets envoyés sont compatibles avec le dernier ensemble de paquets de l'archive `unstable`.

Si le DD casse intentionnellement cette compatibilité en raison de la mise à jour d'une bibliothèque importante, etc., il y a habituellement une annonce sur la liste de diffusion debian-devel, etc.

Avant qu'un ensemble de paquets ne soit déplacé par le script de maintenance de l'archive Debian depuis l'archive `unstable` vers l'archive `testing`, le script de maintenance de l'archive ne se contente pas vérifier sa maturité (environ 2-10 jours) et l'état des rapports de bogues pour ces paquets mais essaie aussi de s'assurer qu'ils sont compatibles avec le dernier ensemble des paquets de l'archive `testing`. Ce processus rend l'archive `testing` très actuelle et utilisable.

Par le processus de gel progressif de l'archive dirigé par l'équipe de diffusion (« release team »), l'archive `testing` est mûrie afin de la rendre entièrement cohérente et sans bogue avec quelques interventions manuelles. Ensuite, la nouvelle version `stable` est créée en assignant le nom de code de l'ancienne archive `testing` à la nouvelle archive `stable` et en créant un nouveau nom de code pour la nouvelle archive `testing`. Le contenu initial de la nouvelle archive `testing` est exactement le même que celui de l'archive `stable` qui vient d'être diffusée.

Les archives `unstable` et `testing` peuvent toutes les deux souffrir temporairement de problèmes en raison de divers facteurs :

— chargement vers l'archive cassé (la plupart du temps, cela concerne `unstable`) ;
— délai pour accepter un nouveau paquet dans l'archive (la plupart du temps, cela concerne `unstable`) ;
— problème de temps de synchronisation de l'archive (à la fois pour `testing` et `unstable`) ;
— intervention manuelle sur l'archive comme la suppression d'un paquet (davantage pour `testing`), etc.

Si vous décidez donc d'utiliser ces archives, vous devriez être capable de corriger ou de contourner ces types de problèmes.



Attention
Pendant les quelques mois qui suivent la diffusion d'une nouvelle version `stable`, la plupart des utilisateur de machines de bureau devraient utiliser l'archive `stable` avec ses mises à jour de sécurité même s'ils utilisent habituellement les archives `unstable` ou `testing`. Pendant cette période de transition, les archives `unstable` et `testing` ne sont pas bonnes pour la plupart des gens. Votre système sera difficile à conserver dans un bon état de fonctionnement avec l'archive `unstable` car elle souffre de pics d'importantes mises à jour de paquets fondamentaux. L'archive `testing` n'est pas utile non plus car elle a sensiblement le même contenu que l'archive `stable` sans la prise en compte de la sécurité (Debian testing-security-announce 2008-12). Après environ un mois, les archives `unstable` et `testing`peuvent être utilisées en prenant des précautions.

ASTUCE
Lors du suivi de l'archive `testing`, un problème causé par la suppression d'un paquet est habituellement contournée en installant le paquet correspondant de l'archive `unstable` qui est envoyé pour la correction du bogue.

Consultez la Charte Debian pour la définition des archives.

— « Sections »
— « Priorités »
— « Système de base »
— « Paquets essentiels »

2.1.6 Debian est totalement libre

Debian est totalement libre pour les raisons suivantes :

— Debian n'installe que des logiciels libres par défaut pour respecter les libertés des utilisateurs ;
— Debian ne fournit que des logiciels libres dans `main` ;
— Debian recommande de n'utiliser que des logiciels libres de `main` ;
— Aucun paquet de `main` ne dépend ou ne recommande de paquets de `non-free`, de `non-free-firmware` ou de `contrib`.

Certaines personnes se demandent si les deux faits suivants sont contradictoires ou non.

— « Debian demeurera totalement libre » (premier point du contrat social Debian).
— Les serveurs Debian hébergent certains paquets `non-free-firmware`, `non-free` et `contrib`.

Ce n'est pas contradictoire pour les raisons suivantes.

— Le système Debian est totalement libre et ses paquets sont hébergés par les serveurs Debian dans la section `main` de l'archive.
— Des paquets hors du système Debian sont hébergés par les serveurs Debian dans les sections `non-free`, `non-free-firmware` et `contrib` de l'archive.

C'est précisément expliqué dans les quatrième et cinquième points du contrat social Debian :

— Nos priorités sont nos utilisateurs et les logiciels libres.
  — Les besoins de nos utilisateurs et de la communauté des logiciels libres nous guideront. Nous placerons leurs intérêts en tête de nos priorités. Nous répondrons aux besoins de nos utilisateurs dans de nombreux types d'environnements informatiques différents. Nous ne nous opposerons pas aux travaux non libres prévus pour fonctionner sur les systèmes Debian. Nous permettrons, sans réclamer rétribution, que d'autres créent des distributions contenant conjointement des logiciels Debian et d'autres travaux. Pour servir ces objectifs, nous fournirons un système intégrant des composants de grande qualité sans restrictions légales incompatibles avec ces modes d'utilisation.
— Travaux non conformes à nos standards sur les logiciels libres.
  — Nous reconnaissons que certains de nos utilisateurs ont besoin d'utiliser des œuvres qui ne sont pas conformes aux Lignes directrices du logiciel libre Debian. Nous avons créé des sections "`non-free`", "`non-firmware`" et "`contrib`" dans notre archive pour ces travaux. Les paquets dans ces sections ne font pas partie du système Debian, bien qu'ils aient été configurés pour être utilisés avec Debian. Nous encourageons les fabricants de CD à lire les licences des paquets dans ces sections et à déterminer s'ils peuvent distribuer ces paquets dans leurs CD. Ainsi, même si les travaux non libres ne font pas partie de Debian, nous soutenons leur utilisation et fournissons l'infrastructure pour les paquets non libres (comme notre système de suivi des bogues et nos listes de diffusion). Les médias officiels de Debian peuvent inclure un micrologiciel qui ne fait pas partie du système Debian pour permettre l'utilisation de Debian avec du matériel qui nécessite ce micrologiciel.

Note
Le texte du cinquième paragraphe dans la version 1.2 actuelle du Contrat social de Debian est légèrement différent du texte ci-dessus. Cette déviation éditoriale est intentionnelle pour rendre ce document utilisateur cohérent sans modifier le contenu réel du Contrat social.

Les utilisateurs devraient être conscients des risques d'utilisation de paquets des sections `non-free`, `non-free-firmware` et `contrib` :

— l'absence de liberté pour de tels paquets de logiciel ;
— l'absence de suivi de la part de Debian pour de tels paquets de logiciel (Debian ne peut pas suivre correctement un logiciel sans avoir accès à son code source) ;
— la contamination de votre système Debian totalement libre.

Les principes du logiciel libre selon Debian sont les normes du logiciel libre pour Debian. Debian interprète « logiciel » de la façon la plus large possible, y compris la documentation, les microprogrammes, les logos et données artistiques des paquets. Cela rend les normes du logiciel libre de Debian très strictes.

Les paquets classiques de `non-free`, `non-free-firmware` et `contrib` incluent des paquets distribuables librement des types suivants :

— les paquets de documentation sous licence de documentation libre GNU avec des sections invariables comme celles de GCC et Make (la plupart sont dans la section `non-free/doc`) ;
— les paquets de micrologiciel contenant des données binaires sans source telles que celles listées dans Section 9.10.5 comme `non-free-firmware` (principalement trouvés dans la section `non-free-firmware/kernel`) ;
— les paquets de jeu ou de fonte avec des restrictions sur l'utilisation commerciale ou la modification de contenu.

Veuillez remarquer que le nombre de paquets de `non-free`, `non-free-firmware` et `contrib` est inférieur à 2 % de ceux dans `main`. Activer l'accès aux sections `non-free`, `non-free-firmware` et `contrib` ne cache pas la provenance des paquets. L'utilisation interactive d'aptitude(8) fournit une visibilité totale et un contrôle complet des paquets installés et de leur section pour garder le système aussi libre que voulu.

2.1.7 Dépendances des paquets

Le système Debian offre un ensemble cohérent de paquets binaires par l'intermédiaire de son mécanisme de déclaration de dépendances binaires versionnées dans les champs du fichier « control ». En voici une définition un peu simplifiée :

— « Depends »
  — Cela déclare une dépendance absolue du paquet et tous les paquets listés dans ce champ doivent être installés en même temps ou à l'avance.
— « Pre-Depends »
  — Comme pour « Depends » excepté que cela demande une installation complète et à l'avance des paquets cités.
— « Recommends »
  — Cela déclare une dépendance forte mais non absolue. La plupart des utilisateurs n'installeront pas le paquet si tous les paquets cités dans ce champ ne sont pas installés.
— « Suggests »
  — Cela déclare une dépendance faible. De nombreux utilisateurs de ce paquet pourront tirer profit de l'installation des paquets cités dans ce champ, mais auront cependant des fonctionnalités acceptables sans eux.
— « Enhances »
  — Cela déclare une dépendance faible comme Suggests mais fonctionne dans la direction opposée.
— « Casse »
  — Cela déclare une incompatibilité de paquet avec habituellement une indication de version. La solution est en général de mettre à jour tous les paquets indiqués dans ce champ.
— « Conflicts »
  — Cela déclare une incompatibilité absolue. Tous les paquets cités dans ce champs doivent être supprimés pour installer ce paquet.
— « Replaces »
  — C'est déclaré lorsque les fichiers installés par ce paquet remplacent des fichiers des paquets cités.
— « Provides »
  — C'est déclaré lorsque ce paquet fournit tous les fichiers et les fonctionnalités des paquets cités.

Note
Remarquez que définir simultanément « Provides », « Conflicts » et « Replaces » pour un paquet virtuel est une configuration saine. Cela permet de s'assurer qu'un seul paquet réel fournissant ce paquet virtuel puisse être installé à un moment donné.

La définition officielle, y compris les dépendances de sources, se trouve dans la Charte Debian ; Chapitre 7 - Déclaration des dépendances entre paquets.

2.1.8 Flux des événements dans la gestion d'un paquet

Voici un résumé du flux simplifié des événements de la gestion d'un paquet par APT.

— **Mettre à jour (« update »)** («`apt update`», « `aptitude update` » ou « `apt-get update` ») :

1. Rechercher les métadonnées d'une archive depuis l'archive distante
2. Reconstruire et mettre à jour les métadonnées locales pour qu'elles puissent être utilisées par APT

— **Mettre à niveau (« upgrade »)** («`apt upgrade`» et «`apt full-upgrade`», ou « `aptitude safe-upgrade` » et « `aptitude full-upgrade` » ou « `apt-get upgrade` » et « `apt-get dist-upgrade` ») :

1. Choisir la version candidate, qui est habituellement la dernière version disponible, pour tous les paquets installés (consultez Section 2.7.7 pour les exceptions) ;
2. Effectuer la résolution des dépendances du paquet
3. Rechercher le paquet binaire sélectionné depuis l'archive distante si la version candidate est différente de la version installée
4. Dépaqueter les paquets binaires ayant été téléchargés
5. Lancer le script **preinst**
6. Installer les fichiers binaires
7. Lancer le script **postinst**

— **Installer** («`apt install …`», `aptitude install …` » ou « `apt-get install …` ») :

1. Choisir les paquets indiqués sur la ligne de commandes
2. Effectuer la résolution des dépendances du paquet
3. Récupérer les paquets binaires sélectionnés depuis l'archive distante
4. Dépaqueter les paquets binaires ayant été téléchargés
5. Lancer le script **preinst**
6. Installer les fichiers binaires
7. Lancer le script **postinst**

— **Supprimer** («`apt remove …`», « `aptitude remove …` » ou « `apt-get remove …` ») :

1. Choisir les paquets indiqués sur la ligne de commandes
2. Effectuer la résolution des dépendances du paquet
3. Lancer le script **prerm**
4. Supprimer les fichiers installés **à l'exception** des fichiers de configuration
5. Lancer le script **postrm**

— **Purger** (« `apt purge` », « `aptitude purge …` » ou « `apt-get purge …` ») :

1. Choisir les paquets indiqués sur la ligne de commandes
2. Effectuer la résolution des dépendances du paquet
3. Lancer le script **prerm**
4. Supprimer les fichiers installés **y compris** leurs fichiers de configuration
5. Lancer le script **postrm**

J'ai ici intentionnellement sauté des détails techniques dans le souci d'avoir une vue d'ensemble.

2.1.9 Première réponse aux problèmes de gestion de paquets

Vous devriez lire l'excellente documentation officielle. Le premier document à lire est « `/usr/share/doc/`*`nom_paquet`*`/RE` qui est spécifique à Debian. Les autres documents dans « `/usr/share/doc/`*`nom_paquet`*`/` » devraient aussi être consultés. Si vous avez configuré l'interpréteur de commande comme dans Section 1.4.2, entrez ce qui suit :

```
$ cd package_name
$ pager README.Debian
$ mc
```

site web	commande
Page d'accueil du système de suivi des bogues Debian (BTS)	`sensible-browser "https://bugs.debian.org/"`
Signalement de bogue d'un nom de paquet connu	`sensible-browser "https://bugs.debian.org/`*`package_name`*`"`
Rapport de bogue concernant un numéro de bogue connu	`sensible-browser "https://bugs.debian.org/`*`bug_number`*`"`

Table 2.5 – Liste de sites web clés pour résoudre les problèmes avec un paquet particulier

Vous aurez besoin d'installer le paquet de documentation correspondant au paquet dont le nom possède le suffixe « `-doc` » pour des informations détaillées.

Si vous rencontrez des problèmes avec un paquet particulier, faites d'abord une recherche sur le site du système de suivi des bogues Debian (BTS).

Rechercher sur Google avec des mots de recherche comprenant « `site:debian.org` », « `site:wiki.debian.org` », « `site:lists.debian.org` », etc.

Pour déposer un signalement de bogue, veuillez utiliser la commande reportbug(1).

2.1.10 Comment obtenir des paquets Debian

Si vous trouvez deux paquets similaires et que vous vous demandez lequel installer sans faire des efforts d'« essais et erreurs », vous pouvez user de **bon sens**. Je considère que les points suivants constituent de bonnes indications pour les paquets à privilégier :

— Essential : yes > no
— Section : main > contrib > non-free
— Priorité : required > important > standard > optional > extra
— Tâches : paquets affichés dans les tâches tels que « Environnement de bureau »
— Paquets sélectionnés par le paquet de dépendance (par exemple, `gcc-16` par `gcc`)
— Popcon : les votes et le nombre d'installations les plus élevés
— Journaux des modifications (« Changelog ») : mises à jour régulières par le responsable
— BTS : pas de bogue RC (pas de bogue critique, grave ou sérieux)
— BTS : réactivité du responsable aux signalements de bogues
— BTS : le plus grand nombre de bogues réglés récemment
— BTS : le plus faible nombre de bogues restants qui ne soient pas dans la liste des vœux

Debian étant un projet basé sur le volontariat avec un modèle de développement distribué, son archive contient de nombreux paquets avec des cibles différentes et de qualité variable. Vous devrez choisir vous-même ce que vous voulez en faire.

2.1.11 Comment faire face à des exigences conflictuelles

Quelle que soit la suite du système Debian que vous pouvez décider d'utiliser, vous pouvez toujours vouloir exécuter des versions de programmes qui ne sont pas disponibles dans cette suite. Même si vous trouvez des paquets binaires de tels programmes dans d'autres suites Debian ou dans d'autres ressources non Debian, leurs exigences peuvent être incompatibles avec votre système Debian actuel.

Bien qu'il soit possible d'ajuster votre système de gestion de paquets avec la technique **apt-pinning** etc., comme décrit dans Section 2.7.7, pour installer de tels paquets binaires hors synchronisation, ces approches de peaufinage ont seulement des cas d'utilisation limités, car elles peuvent briser ces programmes et votre système.

Avant d'installer brutalement ces paquets hors synchronisation, vous devriez chercher toutes les solutions techniques alternatives plus sûres qui sont compatibles avec votre système Debian actuel :

— installez de tels programmes en utilisant les paquets binaires correspondants dans un bac à sable (consultez Section 7.7),
  — beaucoup de programmes graphiques tels que les applications LibreOffice et GNOME sont disponibles sous forme de paquets Flatpak, Snap ou AppImage ;
— créez un environnement isolé (« chroot ») ou similaire et faites-y tourner de tels programmes (consulter la Section 9.11),
  — les commandes en ligne de commande peuvent être exécutées facilement sous son chroot compatible (consulter Section 9.11.4),
  — plusieurs environnements de bureau complets peuvent être essayés facilement sans redémarrage (voir Section 9.11.5) ;
— construisez des versions souhaitées de paquets binaires qui sont compatibles avec votre système Debian actuel par vous-même,
  — il s'agit d'une tâche non triviale (voir Section 2.7.13).

2.2 Opérations de base de la gestion des paquets

Sur le système Debian, les opérations de gestion des paquets basées sur les dépôts peuvent être réalisées à l'aide de nombreux outils de gestion de paquets basés sur APT et disponibles dans le système Debian. Nous décrirons ici les outils de base de gestion des paquets : `apt`, `apt-get`/`apt-cache` et `aptitude`.

Pour les opérations de gestion des paquets qui concernent l'installation des paquets ou les mises à jour des métadonnées des paquets, vous aurez besoin des privilèges de l'administrateur.

2.2.1 `apt` vs. `apt-get` / `apt-cache` vs. `aptitude`

Bien qu'`aptitude` soit un très bon outil interactif et que l'auteur l'utilise, voici quelques avertissements que vous devriez connaître :

— La commande `aptitude` n'est pas recommandée pour une mise à niveau du système entre versions sur le système Debian `stable` après la sortie d'une nouvelle version.
  — L'utilisation de "`apt full-upgrade`" ou de "`apt-get dist-upgrade`" est recommandée pour cela. Voir Bug #411280.
— La commande `aptitude` suggère parfois la suppression massive de paquets lors de la mise à niveau du système sur des systèmes Debian en `testing` ou `unstable`.
  — Cette situation a effrayé de nombreux administrateurs système. Pas de panique.
  — Il semblerait que cela soit principalement causé par un biais de version parmi des paquets dépendants de, ou recommandés par, un méta-paquet tel que `gnome-core`.
  — Cela peut être résolu en sélectionnant « Annuler les actions en attente » dans le menu de commande d'`aptitude`, en quittant `aptitude` et en utilisant la commande « `apt full-upgrade` ».

Les commandes `apt-get` et `apt-cache` sont les outils les plus **basiques** de gestion des paquets basés sur APT.

— `apt-get` et `apt-cache` n'offre qu'une interface utilisateur en ligne de commandes.
— `apt-get` est le mieux adapté pour les **mises à jour majeures du système** entre les versions, etc.
— `apt-get` offre un système de résolution des dépendances entre paquets **robuste**.
— `apt-get` nécessite moins de ressources matérielles. Il consomme moins de mémoire et fonctionne plus rapidement.
— `apt-cache` offre une recherche basée sur des expressions rationnelles **standard** sur les noms et les descriptions des paquets.
— `apt-get` et `apt-cache` peuvent gérer des versions multiples des paquets en utilisant `/etc/apt/preferences` mais est assez lourd.

La commande `apt` est une interface de haut niveau en ligne de commande pour la gestion de paquets. C'est basiquement une enveloppe d'`apt-get`, d'`apt-cache` et de commandes similaires, originellement destinée comme interface d'utilisateur final, et active quelques options mieux adaptées par défaut à un usage interactif.

— `apt` fournit une barre de progression plaisante lors de l'installation de paquets en utilisant `apt install`.
— `apt` **supprimera** par défaut les paquets `.deb` mis en cache après une utilisation réussie de paquets téléchargés.

ASTUCE
Il est recommandé aux utilisateurs d'utiliser la nouvelle commande apt(8) pour un usage **interactif** et d'utiliser les commandes apt-get(8) et apt-cache(8) dans des scripts d'interpréteur.

La commande `aptitude` est l'outil de gestion des paquets basé sur APT le plus **flexible**.

— `aptitude` offre une interface utilisateur interactive en plein écran en mode texte.
— `aptitude` offre aussi une interface utilisateur en ligne de commandes.
— `aptitude` est le mieux adapté pour la **gestion interactive journalière des paquets** comme, par exemple, la vérification des paquets installés et la recherche de paquets disponibles.
— `aptitude` nécessite plus de ressources matérielles. Il consomme plus de mémoire et fonctionne moins rapidement.
— `aptitude` offre une recherche **avancée** basée sur des expressions rationnelles pour la recherche sur toutes les métadonnées des paquets.
— `aptitude` peut gérer des versions multiples des paquets sans utiliser `/etc/apt/preferences` et est assez intuitif.

2.2.2 Opérations de base de gestion des paquets en ligne de commandes

Voici les opérations de base de gestion des paquets en ligne de commandes en utilisant apt(8), aptitude(8), apt-get(8) et apt-cache(8) .

`apt`/`apt-get` et `aptitude` peuvent mélangées sans inconvénients majeurs.

« `aptitude why` *`expression-rationnelle`* » peut afficher plus d'informations par « `aptitude -v why` *`expressio`*
On peut obtenir des informations similaires par « `apt rdepends` *`paquet`*" ou "`apt-cache rdepends` *`paquet`* ».

Lorsque la commande `aptitude` est lancée en mode ligne de commande, et rencontre des problèmes tels que des conflits de paquets, vous pouvez passez en mode plein écran en pressant ensuite la touche « e » à l'invite de commande.

Note
Bien que la commande `aptitude` soit disponible avec de riches fonctionnalités comme son solveur de paquets avancé, cette complexité a causé (et peut encore causer) certaines régressions comme le bogue #411123, le bogue #514930 et le bogue #570377. En cas de doute, veuillez utiliser les commandes `apt`, `apt-get` et `apt-cache` plutôt que la commande `aptitude`.

Vous pouvez indiquer les options de commande juste après « `aptitude` ».

Consultez aptitude(8) et le « manuel de l'utilisateur d'aptitude » à « `/usr/share/doc/aptitude/README` » pour en apprendre davantage.

2.2.3 Utilisation interactive d'aptitude

Pour une gestion interactive des paquets, lancez `aptitude` en mode interactif depuis l'invite de l'interpréteur de commandes de la console comme suit :

syntaxe d'`apt`	syntaxe d'`aptitude`	syntaxe d'`apt-get` et `apt-cache`	description
`apt update`	`aptitude update`	`apt-get update`	mettre à jour les métadonnées de l'archive du paquet
`apt install foo`	`aptitude install foo`	`apt-get install foo`	installer la version candidate du paquet « `toto` » ainsi que ses dépendances
`apt upgrade`	`aptitude safe-upgrade`	`apt-get upgrade`	installer les versions candidates des paquets installés sans supprimer aucun autre paquet
`apt full-upgrade`	`aptitude full-upgrade`	`apt-get dist-upgrade`	installer les versions candidates des paquets installés en supprimant d'autres paquets si nécessaire
`apt remove foo`	`aptitude remove foo`	`apt-get remove foo`	supprimer le paquet « `toto` » en laissant ses fichiers de configuration
`apt autoremove`	N/A	`apt-get autoremove`	supprimer les paquets installés automatiquement lorsqu'ils ne sont plus nécessaires
`apt purge foo`	`aptitude purge foo`	`apt-get purge foo`	purger le paquet « `toto` » ainsi que ses fichiers de configuration
`apt clean`	`aptitude clean`	`apt-get clean`	nettoyer complètement le dépôt local des fichiers de paquets récupérés
`apt autoclean`	`aptitude autoclean`	`apt-get autoclean`	nettoyer le dépôt local des fichiers des paquets périmés
`apt show foo`	`aptitude show foo`	`apt-cache show foo`	afficher des informations détaillées concernant le paquet « `toto` »
`apt search` *`regex`*	`aptitude search` *`regex`*	`apt-cache search` *`regex`*	rechercher les paquets qui correspondent à l'*expression rationnelle*
N/A	`aptitude why` *`regex`*	N/A	expliquer les raisons qui font que les paquets correspondant à l'*expression rationnelle* devront être installés
N/A	`aptitude why-not` *`regex`*	N/A	expliquer les raisons pour lesquels les paquets qui correspondent à l'*expression rationnelle* ne peuvent pas être installés
`apt list --manual-installed`	`aptitude search '~i!~M'`	`apt-mark showmanual`	lister les paquets installés manuellement

Table 2.6 – Opérations de base de gestion des paquets avec la ligne de commandes en utilisant apt(8), aptitude(8), apt-get(8) et apt-cache(8)

option de la commande	description
`-s`	simuler le résultat de la commande
`-d`	télécharger seulement les paquets sans les installer ni les mettre à jour
`-D`	afficher une courte explication avant les installations ou les suppressions automatiques

Table 2.7 – Options importantes de la commande aptitude(8)

```
$ sudo aptitude -u
Password:
```

Cela va mettre à jour la copie locale des informations de l'archive et afficher la liste des paquets en plein écran avec un menu. On trouvera la configuration d'aptitude dans « `~/.aptitude/config` ».

ASTUCE
Si vous désirez utiliser la configuration de l'administrateur (root) plutôt que celle de l'utilisateur, utilisez la commande « `sudo -H aptitude …` » en remplacement de « `sudo aptitude …` » dans l'expression précédente.

ASTUCE
`Aptitude` définit automatiquement les **actions en attente** lorsqu'il est lancé de manière interactive. Si elles ne vous conviennent pas, vous pouvez le réinitialiser depuis le menu : « Action » → « Annuler les opérations en attente ».

2.2.4 Raccourcis clavier d'aptitude

Les raccourcis clavier principaux pour parcourir l'état des paquets et pour définir les « actions prévues » sur ces paquets dans le mode plein écran sont les suivants :

touche	**affectation**
`F10` ou `Ctrl-t`	menu
`?`	afficher l'**aide** pour les raccourcis clavier (liste plus complète)
`F10` → Aide → Manuel de l'utilisateur	afficher le Manuel de l'utilisateur
`u`	mettre à jour les informations de l'archive des paquets
`+`	marquer le paquet pour **mise à niveau** ou **installation**
`-`	marquer le paquet pour **suppression** (conserver ses fichiers de configuration)
`_`	marquer le paquet pour être **purgé** (supprimer ses fichiers de configuration)
`=`	mettre le paquet dans l'état « **conservé** »
`U`	marquer tous les paquets susceptibles de mise à niveau (fonctionne comme **full-upgrade**)
`g`	lancer le **téléchargement** et l'**installation** des paquets sélectionnés
`q`	quitter l'écran actuel et enregistrer les modifications
`x`	quitter l'écran actuel en abandonnant les modifications
`Enter`	afficher les informations concernant un paquet
`C`	afficher le journal des modifications (« changelog ») d'un paquet
`l`	modifier les limites pour les paquets affichés
`/`	rechercher la première correspondance
`\`	répéter la dernière recherche

Table 2.8 – Liste des raccourcis clavier d'aptitude

L'indication du nom de fichier sur la ligne de commandes et à l'invite du menu après avoir pressé « `l` » et « `//` » prend l'expression rationnelle d'aptitude telle que décrite ci-dessous. Une expression rationnelle d'aptitude peut correspondre explicitement à un nom de paquet en utilisant une chaîne de caractères commençant par « ~n » et suivie du nom de paquet.

ASTUCE
Vous devrez presser « U » pour obtenir la mise à niveau de tous les paquets installés vers la **version candidate** de l'interface visuelle. Sinon, seuls les paquets sélectionnés et certains paquets ayant des dépendances versionnées sur ces paquets seront mis à niveau vers la **version candidate**.

2.2.5 Vues des paquets sous aptitude

Dans le mode interactif en plein écran d'aptitude(8), les paquets de la liste des paquets sont affichés comme dans l'exemple suivant.

```
idA   libsmbclient                          -2220kB 3.0.25a-1  3.0.25a-2
```

Cette ligne signifie, en partant de la gauche :

— Indicateur d'« état actuel » (la première lettre)
— Indicateur d'« action prévue » (la seconde lettre)
— Indicateur « automatique » (la troisième lettre)
— Nom du paquet
— Modification de l'utilisation du disque attribuée à l'« action prévue »
— Version actuelle du paquet
— Version candidate du paquet

ASTUCE
La liste complète des indicateurs est donnée en bas de l'écran d'**Aide** affiché en pressant « ? ».

La **version candidate** est choisie en fonction des préférences locales actuelles (consultez apt_preferences(5) et Section 2.7.7).

Plusieurs types de vues de paquets sont disponibles depuis le menu « Vues ».

vue	**description de la vue**
Package View	consultez Tableau 2.10 (défaut)
Audit Recommendations	liste des paquets qui sont recommandés par certains paquets installés mais qui ne sont pas encore installés sur le système
Flat Package List	liste des paquets sans regroupement par catégories (pour l'utilisation avec des expressions rationnelles)
Source Package View	liste de paquets groupés par paquet source

Table 2.9 – Liste des vues d'aptitude

La « Vue des paquets » standard classe les paquets un peu comme le fait dselect avec quelques fonctionnalités supplémentaires.

ASTUCE
La vue des tâches peut être utilisée pour choisir les paquets nécessaires à votre tâche.

catégorie	description de la vue
`Upgradable Packages`	liste des paquets organisée sous la forme `section` → `zone` → `paquet`
`New Packages`	,,
`Installed Packages`	,,
`Not Installed Packages`	,,
`Obsolete and Locally Created Packages`	,,
`Virtual Packages`	liste des paquets ayant la même fonction
`Tasks`	liste des paquets ayant les différentes fonctions généralement nécessaires à une tâche

Table 2.10 – Classement par catégories des vues de paquets standard

2.2.6 Options de la méthode de recherche avec aptitude

Aptitude vous offre différentes options pour rechercher des paquets en utilisant sa formule d'expressions rationnelles.

— Ligne de commande du shell :
  — « `aptitude search '`*`expression_rationnelle_aptitude`*`'` » afin d'afficher l'état d'installation, le nom du paquet et une courte description des paquets correspondants
  — « `aptitude show '`*`nom_paquet`*`'` » pour afficher la description détaillée du paquet
— Mode interactif plein écran :
  — « `l` » pour limiter la vue des paquets à ceux qui correspondent
  — « `/` » pour rechercher un paquet correspondant
  — « `\` » pour rechercher en arrière un paquet correspondant
  — « `n` » pour rechercher le suivant
  — « `N` » pour rechercher le suivant (en arrière)

ASTUCE
La chaîne du *nom_paquet* est traitée comme la correspondance exacte de chaîne pour le nom de paquet à moins qu'il ne soit lancé explicitement avec « ~ » pour être la formule d'expression rationnelle.

2.2.7 Les formules d'expressions rationnelles d'aptitude

La formule des expressions rationnelles d'aptitude est étendue **ERE** de manière similaire à mutt (consultez Section 1.6.2) et la signification des extensions de règles de correspondance spécifiques à `aptitude` est la suivante :

— La partie expression rationnelle est la même **ERE** que celle utilisée dans les outils UNIX typiques en utilisant « ^ », « `.*` », « $ » etc. comme dans egrep(1), awk(1) et perl(1).
— La dépendance *type* est comprise dans la liste (depends, predepends, recommends, suggests, conflicts, replaces, provides) et spécifie les relations du paquet avec d'autres paquets.
— Le *type* de relation par défaut est « depends ».

ASTUCE
Lorsqu'un *motif d'expression rationnelle* est une chaîne de caractères vide, placez « ~T » directement après la commande.

Voici quelques raccourcis.

description des règles étendues de correspondance	**formules d'expressions rationnelles**
correspond au nom du paquet	`~n`*`regex_name`*
correspond à la description	`~d`*`regex_description`*
correspond au nom de la tâche	`~t`*`regex_task`*
correspond à l'étiquette debtag	`~G`*`regex_debtag`*
correspond au responsable du paquet	`~m`*`regex_maintainer`*
correspond à la section du paquet	`~s`*`regex_section`*
correspond à la version du paquet	`~V`*`regex_version`*
correspond à l'archive	`~A{trixie,forky,sid}`
correspond à l'origine	`~O{debian,…}`
correspond à la priorité	`~p{extra,important,optional,required,standard}`
correspond aux paquets essentiels	`~E`
correspond aux paquets virtuels	`~v`
correspond aux nouveaux paquets	`~N`
correspond aux actions en attente	`~a{install,upgrade,downgrade,remove,purge,hold,keep}`
correspond aux paquets installés	`~i`
correspond aux paquets installés ayant la marque **A** (paquets installés automatiquement)	`~M`
correspond aux paquets installés n'ayant pas la marque **A** (paquets sélectionnés par l'administrateur)	`~i!~M`
correspond aux paquets installés et pouvant être mis à jour	`~U`
correspond aux paquets supprimés mais non purgés	`~c`
correspond aux paquets supprimés, purgés ou pouvant être supprimés	`~g`
correspond aux paquets ayant une dépendance cassée	`~b`
correspond aux paquets ayant une dépendance cassée de *type*	`~B`*`type`*
correspond aux paquets filtrés par *motif* ayant une dépendance de *type*	`~D[`*`type`*`:]`*`pattern`*
correspond aux paquets filtrés par *motif* ayant une dépendance cassée de *type*	`~DB[`*`type`*`:]`*`pattern`*
correspond aux paquets vers lesquels le paquet filtré par *motif* déclare une dépendance de *type*	`~R[`*`type`*`:]`*`pattern`*
correspond aux paquets vers lesquels le paquet filtré par *motif* a une dépendance cassée de *type*	`~RB[`*`type`*`:]`*`pattern`*
correspond aux paquets desquels dépendent d'autres paquets installés	`~R~i`
correspond aux paquets desquels ne dépend aucun autre paquet	`!~R~i`
correspond aux paquets vers lesquels d'autres paquets installés dépendent ou qu'ils recommandent	`~R~i|~Rrecommends:~i`
correspond au paquet *motif* dont la version est filtrée	`~S filter `*`pattern`*
correspond à tous les paquets (vrai)	`~T`
ne correspond à aucun paquet (faux)	`~F`

Table 2.11 – Liste des formules d'expressions rationnelles d'aptitude

— « ~P*term* » == « ~Dprovides:*term* »
— « ~C*term* » == « ~Dconflicts:*term* »
— « …~W term » == « (…|term) »

Les utilisateurs familiers avec mutt comprendront rapidement car mutt a été la source d'inspiration pour la syntaxe des expressions. Consultez « SEARCHING, LIMITING, AND EXPRESSIONS » dans le manuel de l'utilisateur (« /usr/share/doc/aptitude/README »).

Note
Avec la version lenny d'aptitude(8), la nouvelle **forme longue** de la syntaxe comme « ?broken » peut être utilisée pour la correspondance des expressions rationnelles en remplacement de l'ancienne **forme courte** équivalente « ~b ». Le caractère espace « » est maintenant considéré comme l'un des caractères de terminaison d'une expression rationnelle en plus du caractère tilde « ~ ». Consultez la syntaxe de la nouvelle **forme longue** dans le « Manuel de l'utilisateur ».

2.2.8 Résolution des dépendances par aptitude

La sélection d'un paquet dans aptitude récupère non seulement les paquets définis dans son champ « Depends: » mais aussi ceux définis dans le champ « Recommends: » si la configuration a été faite dans ce sens dans le menu « F10 → Options → Préférences → Gestion des dépendances ». Ces paquets installés automatiquement seront supprimés automatiquement s'ils ne sont plus nécessaires sous aptitude.

Le drapeau contrôlant le comportement « auto install » de la commande aptitude peut aussi être manipulé en utilisant la commande apt-mark(8) du paquet apt.

2.2.9 Journaux d'activité des paquets

Vous pouvez vérifier l'activité de l'historique des paquets dans les fichiers journaux.

fichier	contenu
/var/log/dpkg.log	Enregistrement des actions au niveau de dpkg pour l'activité de tous les paquets
/var/log/apt/term.log	Journal de l'activité générique d'APT
/var/log/aptitude	Journal des actions de la commande aptitude

Table 2.12 – Fichiers journaux de l'activité des paquets

En réalité, il n'est pas aussi facile de comprendre la signification de ces journaux. Consultez Section 9.3.9 pour une façon de faire plus simple.

2.3 Exemples d'opérations avec aptitude

Voici quelques exemples d'opérations d'aptitude(8).

2.3.1 Rechercher des paquets intéressants

Vous pouvez rechercher les paquets qui satisfont à vos besoins avec aptitude à partir de la description du paquet ou depuis la liste se trouvant dans « Tasks ».

2.3.2 Afficher les paquets dont les noms correspondent à une expression rationnelle

La commande suivante affiche les paquets dont les noms correspondent à une expression rationnelle.

```
$ aptitude search '~n(pam|nss).*ldap'
v   libnss-ldap                                                -
p   libnss-ldapd                                               - NSS module for using LDAP  ↩
    as a naming service
v   libpam-ldap                                                -
p   libpam-ldapd                                               - PAM module for using LDAP  ↩
    as an authentication service
```

Il vous est assez facile de trouver le nom exact d'un paquet.

2.3.3 Parcours en correspondance avec une expression rationnelle

L'expression rationnelle « `~dipv6` » entrée dans la vue « Nouvelle liste des paquets » (« New Flat Package List » depuis l'invite « `l` » limite la vue aux paquets dont la description correspond à cette expression rationnelle et vous permet de parcourir les informations de manière interactive.

2.3.4 Purger pour de bon les paquets supprimés

Vous pouvez supprimer tous les fichiers de configuration subsistant des paquets supprimés.

Vérifiez le résultat de la commande suivante :

```
# aptitude search '~c'
```

Si vous pensez que les paquets affichés doivent être purgés, exécutez la commande suivante :

```
# aptitude purge '~c'
```

Vous pouvez avoir envie de faire la même chose en mode interactif avec un contrôle plus fin.

Indiquez l'expression rationnelle « `~c` » dans la « Nouvelle liste des paquets » en utilisant l'invite « `l` ». Cela limite la vue des paquets à ceux qui correspondent à l'expression rationnelle, c'est-à-dire « supprimé mais non purgé ». On peut visualiser tous les paquets correspondant à cette expression rationnelle en pressant « `[` » depuis une section de haut niveau.

Pressez ensuite « `_` » depuis une fenêtre de haut niveau comme « Paquets non installés ». Seuls les paquets correspondants à l'expression rationnelle se trouvant dans cette section seront marqués comme devant être purgés par cette commande. Vous pouvez exclure certains paquets de cette opération en pressant de manière interactive la touche « = » en face de chacun d'eux.

Cette technique est assez pratique et fonctionne avec de nombreuses autres touches de commande.

2.3.5 Toilettage de l'état d'installation automatique/manuel

Voici comment je nettoie l'état d'installation automatique/manuel des paquets (après avoir utilisé un installateur de paquets autre qu'aptitude, etc.).

1. Démarrer `aptitude` en mode interactif en tant qu'administrateur.
2. Entrer « u », « `U` », « `f` » et « g » pour mettre à jour la liste des paquets et mettre à niveau les paquets.
3. Entrer « `l` » afin de définir la limite d'affichage des paquets avec « `~i(~R~i|~Rrecommends:~i)` » et entrez « `M` » sur « `Paquets installés` » automatiquement.
4. Entrer « `l` » afin de définir la limite d'affichage des paquets avec « `~prequired|~pimportant|~pstandard|~E` » et entrez « `m` » sur les « `Paquets installés` » manuellement.

5. Entrer « `l` » pour définir la limite d'affichage des paquets avec « `~i!~M` » et supprimez tous les paquets inutilisés en entrant `-` » sur chacun d'eux après les avoir affichés en entrant « `[` » sur « `Paquets installés` ».
6. Entrer « `l` » pour définir la limite d'affichage des paquets avec « `~i` » puis entrez `m` » sur les « `Tâches` » pour attribuer un marquage « manuellement installé » aux paquets.
7. Quitter `aptitude`.
8. Lancer « `apt-get -s autoremove|less` » en tant qu'administrateur pour vérifier les paquets non utilisés.
9. Redémarrer `aptitude` en mode interactif et marquer les paquets nécessaires comme « `m` ».
10. Redémarrer « `apt-get -s autoremove|less` » en tant qu'administrateur et vérifier à nouveau que « REMOVED » ne contient que les paquets voulus.
11. Lancer « `apt-get autoremove|less` » en tant qu'administrateur pour supprimer automatiquement les paquets inutilisés.

L'action « `m` » sur les « `Tâches` » est facultative pour éviter une situation de suppression en masse de paquets dans le futur.

2.3.6 Mise à jour pour l'ensemble du système

Note
Lors du changement vers une nouvelle version, etc., vous devriez envisager d'effectuer une installation propre d'un nouveau système même si Debian peut être mis à niveau comme décrit ci-dessous. Cela vous donne une chance de supprimer les résidus amassés et vous présente la meilleure combinaison des derniers paquets. Bien entendu, vous devrez effectuer une sauvegarde totale de votre système vers un endroit sûr (consultez Section 10.2) avant de faire cela. Je vous recommande de faire une configuration dual boot en utilisant des partitions différentes afin d'effectuer une transition en douceur.

Vous pouvez effectuer une mise à niveau du système vers une nouvelle version en changeant le contenu de la **liste des sources** pointant vers une nouvelle publication et en exécutant la commande "`apt update ; apt dist-upgrade`".

Pour mettre à niveau depuis `stable` vers `testing` ou `unstable` durant le cycle de publication de `trixie` en tant que `stable`, il faut remplacer « `trixie` » dans l'exemple de **liste des sources** de Section 2.1.5 avec « `forky` » ou « `sid` ».

En réalité, vous pouvez rencontrer quelques complications en raison de problèmes de transition de paquets, le plus souvent pour des problèmes de dépendances de paquets. Plus la mise à jour est importante, plus vous avez de chances de rencontrer des problèmes importants. Lors de la transition de l'ancienne version `stable` vers la nouvelle version `stable` après sa diffusion, afin de minimiser les problèmes vous pouvez lire ses nouvelles Notes de diffusion et suivre la procédure exacte qui y est décrite.

Lorsque vous décidez de changer de la version `stable` vers la version `testing` avant sa diffusion formelle, il n'y a pas de Notes de diffusion pour vous aider. La différence entre `stable` et `testing` peut être devenue assez importante depuis la diffusion de la version `stable` précédente et rendre compliquée la situation de la mise à jour.

Vous devriez aller vers la mise à niveau complète avec précaution tout en récupérant les dernières informations depuis les listes de diffusion et en usant de bon sens.

1. Lire les « Notes de diffusion » précédentes.
2. Faire la sauvegarde de l'ensemble du système (particulièrement les données et les informations de configuration).
3. Avoir un support amorçable prêt au cas où le chargeur initial serait cassé.
4. Informer les utilisateurs du système bien à l'avance.
5. Enregistrer l'activité de mise à jour avec script(1).
6. Appliquer « unmarkauto » aux paquets nécessaires, par exemple « `aptitude unmarkauto vim` », afin d'en éviter la suppression.
7. Minimiser les paquets installés pour réduire les chances de conflits de paquets, par exemple supprimer les paquets de la tâche « bureau ».

8. Supprimer le fichier « `/etc/apt/preferences` » (désactiver l'épinglage apt « **apt-pinning** »).
9. Essayer de mettre à jour par étapes : `oldstable` → `stable` → `testing` → `unstable`.
10. Mettre à jour la **liste des sources** pour pointer vers une nouvelle archive seulement et exécuter "`aptitude update`".
11. Installer d'abord, de manière facultative, les nouveaux **paquets essentiels**, par exemple « `aptitude install perl` ».
12. Lancer la commande « `apt-get -s dist-upgrade` » pour contrôler quel en sera l'impact.
13. Et enfin lancer la commande « `apt-get dist-upgrade` ».



Attention
Il n'est pas sage de sauter une version majeure de Debian lors de la mise à niveau entre versions `stable`.



Attention
Dans les « Notes de diffusion » précédentes, GCC, Linux Kernel, initrd-tools, Glibc, Perl, APT tool chain, etc. ont demandé une attention particulière pour une mise à niveau de l'ensemble du système.



Attention
"Release Notes" may not cover all possible cases. If you change low level configurations, your next upgrade may fail badly as "... segfault after upgrade ...".

Pour une mise à jour quotidienne d'`unstable`, consultez Section 2.4.3.

2.4 Opérations avancées de gestion des paquets

2.4.1 Opérations avancées de gestion des paquets en ligne de commandes

Voici la liste des autres opérations de gestion des paquets pour lesquelles `aptitude` est de trop haut niveau ou n'a pas la fonctionnalité requise.

Note
Pour les paquets disponibles en multi-arch(itectures), certaines commandes devront parfois être complétées du nom de l'architecture cible. Il faut, par exemple, utiliser « `dpkg -L libglib2.0-0:amd64` » pour obtenir une liste du contenu du paquet `libglib2.0-0` dans son architecture `amd64`.



Attention
Un outil de plus bas niveau tel que « `dpkg -i …` » et « `debi …` » devra être utilisé avec précautions par l'administrateur du système. Il ne s'assure pas automatiquement des dépendances exigées par le paquet. Les options « `--force-all` » et similaires de la ligne de commandes de dpkg (consultez dpkg(1)) ne sont prévues pour être utilisées que par des experts. Les utiliser sans comprendre entièrement leurs effets peut casser l'ensemble de votre système.

Veuillez noter ce qui suit :

commande	action
`COLUMNS=120 dpkg -l` *`package_name_pattern`*	afficher l’état d’un paquet installé pour le signalement de bogue
`dpkg -L` *`package_name`*	afficher le contenu d’un paquet installé
`dpkg -L` *`package_name`* `| egrep '/usr/share/man/man.*/.+'`	afficher les pages de manuel d’un paquet installé
`dpkg -S` *`file_name_pattern`*	afficher les paquets installés dont le nom correspond
`apt-file search` *`file_name_pattern`*	afficher les paquets de l’archive dont le nom correspond
`apt-file list` *`package_name_pattern`*	afficher le contenu d’un paquet correspondant de l’archive
`dpkg-reconfigure` *`package_name`*	reconfigurer le paquet exact
`dpkg-reconfigure -plow` *`package_name`*	reconfigurer le paquet exact avec la question la plus détaillée
`configure-debian`	reconfigurer les paquets depuis le menu en plein écran
`dpkg --audit`	système de vérification des paquets partiellement installés
`dpkg --configure -a`	configurer tous les paquets partiellement installés
`apt-cache policy` *`binary_package_name`*	afficher la version disponible, la priorité et les informations concernant l’archive du paquet binaire
`apt-cache madison` *`package_name`*	afficher la version disponible et les informations de l’archive concernant un paquet
`apt-cache showsrc` *`binary_package_name`*	afficher les informations concernant le paquet source d’un paquet binaire
`apt-get build-dep` *`package_name`*	installer les paquets nécessaires à la construction d’un paquet
`aptitude build-dep` *`package_name`*	installer les paquets nécessaires à la construction d’un paquet
`apt-get source` *`package_name`*	télécharger une source (depuis l’archive standard)
`dget` *`URL for dsc file`*	télécharger un paquet source (depuis une autre archive)
`dpkg-source -x` *`package_name_version-debian.revision`*`.dsc`	construire une arborescence des sources à partir d’un ensemble de paquets source (« `*.orig.tar.gz` » et « `*.debian.tar.gz`”/”`*.diff.gz` »)
`debuild binary`	construire des paquets depuis une arborescence source locale
`make-kpkg kernel_image`	construire un paquet du noyau à partir de l’arborescence source du noyau
`make-kpkg --initrd kernel_image`	construire un paquet du noyau à partir de l’arborescence source du noyau avec initramfs activé
`dpkg -i` *`package_name_version-debian.revision_arch`*`.deb`	installer un paquet local sur le système
`apt install /path/to/`*`package_filename`*`.deb`	installer un paquet local sur le système, tout en essayant de résoudre automatiquement les dépendances
`debi` *`package_name_version-debian.revision_arch`*`.dsc`	installer des paquets locaux sur le système
`dpkg --get-selections '*' >selection.txt`	enregistrer l’information d’état de la sélection des paquets au niveau de `dpkg`
`dpkg --set-selections <selection.txt`	définir l’information d’état de sélection des paquets au niveau de `dpkg`
`echo` *`package_name`* `hold | dpkg --set-selections`	définir l’information d’état de sélection des paquets au niveau de `dpkg` à **hold** (gelé, équivalent à `aptitude hold` *`nom_paquet`*)

Table 2.13 – Liste des opérations avancées de gestion des paquets

— Toutes les commandes de configuration et d'installation doivent être lancées avec le compte de l'administrateur.
— Au contraire d'`aptitude` qui utilise des expressions rationnelles (consultez Section 1.6.2), les autres commandes de gestion des paquets utilisent des motifs semblables aux motifs génériques (« glob ») de l'interpréteur de commandes (consultez Section 1.5.6).
— La commande apt-file(1) fournie par le paquet `apt-file` doit être précédée de l'exécution de la commande « `apt-file update` ».
— configure-debian(8) fourni par le paquet `configure-debian` exécute en fond dpkg-reconfigure(8).
— dpkg-reconfigure(8) exécute les scripts du paquet en utilisant en fond debconf(1).
— Les commandes « `apt-get build-dep` », « `apt-get source` » et « `apt-cache showsrc` » nécessitent une entrée « `deb-src` » dans la **liste des sources**.
— dget(1), debuild(1) etdebi(1) ont besoin du paquet `devscripts`.
— Consultez la procédure de (re)paquetage en utilisant « `apt-get source` » dans Section 2.7.13.
— La commande `make-kpkg` exige le paquet `kernel-package` (consultez Section 9.10).
— Consultez Section 12.9 pour la réalisation de paquets en général.

2.4.2 Vérifier les fichiers de paquets installés

L'installation de `debsums` permet, avec debsums(1), la vérification des fichiers des paquets installés d'après les valeurs de MD5sum se trouvant dans le fichier « `/var/lib/dpkg/info/*.md5sums` » Consultez Section 10.3.5 pour le fonctionnement de MD5sum.

Note
Comme la base de données MD5sum peut être trafiquée par un intrus, debsums(1) est d'une utilité restreinte en tant qu'outil de sécurité. Il n'est bon que pour la vérification locale des modifications de l'administrateur ou des défectuosités en raison de problèmes de support.

2.4.3 Protection contre les problèmes de paquets

De nombreux utilisateurs préfèrent suivre les versions **testing** (ou **unstable**) du système Debian pour ses nouvelles fonctionnalités et paquets. Cela rend le système plus sensible aux bogues critiques des paquets.

L'installation du paquet `apt-listbugs` protège votre système contre les bogues critiques en recherchant automatiquement dans le BTS de Debian les bogues critiques lors de la mise à jour par l'intermédiaire du système APT.

L'installation du paquet `apt-listchanges` indique les nouveautés importantes se trouvant dans « `NEWS.Debian` » lors de la mise à jour du système avec APT.

2.4.4 Rechercher dans les métadonnées du paquet

Bien que visiter le site Debian à https://packages.debian.org/ permette aujourd'hui de rechercher facilement les métadonnées des paquets, voyons les méthodes plus traditionnelles .

Les commandes grep-dctrl(1), grep-status(1) et grep-available(1) peuvent être utilisées pour effectuer des recherches dans tous les fichiers dont le format général est celui d'un fichier de contrôle de Debian.

« `dpkg -S` *`motif_nom_fichier`* » peut être utilisé pour rechercher les noms de paquets installés par `dpkg` qui contiennent des fichiers dont le nom correspond au motif. Mais les fichiers créés par les scripts du responsable du paquet ne sont pas pris en compte.

Si vous devez faire des recherches plus élaborées sur les métadonnées de dpkg, il vous faudra lancer la commande « `grep -e motif_expression_rationnelle *` » dans le répertoire « `/var/lib/dpkg/info/` ». Cela vous permet de rechercher des mots mentionnés dans les scripts des paquets et les textes des requêtes d'installation.

Pour rechercher de manière récursive les dépendances de paquets, vous devrez utiliser apt-rdepends(8).

2.5 Fonctionnement interne de la gestion des paquets Debian

Voyons comment le système Debian de gestion des paquets fonctionne de manière interne. Cela vous permettra de créer votre propre solution à certains problèmes de paquets.

2.5.1 Métadonnées de l'archive

Les fichiers de métadonnées de chaque distribution se trouvent sur chaque miroir Debian, dans « `dist/`*`nom_de_code`* », par exemple, « `http://deb.debian.org/debian/` ». On peut parcourir la structure de son archive à l'aide d'un navigateur web. Il existe 6 types de métadonnées clés :

fichier	emplacement	contenu
`Release`	sommet de la distribution	description de l'archive et informations d'intégrité
`Release.gpg`	sommet de la distribution	fichier signature du fichier « `Release` » signé avec la clé de l'archive
`Contents-`*`architecture`*	sommet de la distribution	liste de tous les fichiers pour tous les paquets dans l'archive pertinente
`Release`	sommet de chaque combinaison de distribution/section/architecture	description de l'archive utilisée pour la règle de apt_preferences(5)
`Packages`	sommet de chaque combinaison de distribution/section/architecture binaire	`debian/control` concaténés des paquets binaires
`Sources`	sommet de chaque combinaison distribution/section/source	`debian/control` concaténés des paquets sources

Table 2.14 – Contenu des métadonnées de l'archive Debian

Dans les archives récentes, ces métadonnées sont enregistrées sous forme compressée et différentielle afin de limiter le trafic réseau.

2.5.2 Fichier « Release » de plus haut niveau et authenticité

ASTUCE
Le fichier « `Release` » de plus haut niveau est utilisé pour signer l'archive au moyen du système **secure APT**.

Chaque version de l'archive Debian possède un fichier « `Release` » de plus haut niveau, par exemple, « `http://deb.debi` comme ci-dessous :

```
Origin: Debian
Label: Debian
Suite: unstable
Codename: sid
Date: Sat, 14 May 2011 08:20:50 UTC
Valid-Until: Sat, 21 May 2011 08:20:50 UTC
Architectures: alpha amd64 armel hppa hurd-i386 i386 ia64 kfreebsd-amd64 kfreebsd-i386 mips ↩
    mipsel powerpc s390 sparc
Components: main contrib non-free
Description: Debian x.y Unstable - Not Released
```

```
MD5Sum:
 bdc8fa4b3f5e4a715dd0d56d176fc789 18876880 Contents-alpha.gz
 9469a03c94b85e010d116aeeab9614c0 19441880 Contents-amd64.gz
 3d68e206d7faa3aded660dc0996054fe 19203165 Contents-armel.gz
...
```

Note
Vous pouvez trouver ici ma justification pour l'utilisation de « suite », « nom de code » dans Section 2.1.5. La « distribution » est utilisée pour désigner à la fois « suite » et « nom de code ». Tous les noms possibles de « sections » de l'archive sont indiqués dans l'archive sous « Composants ».

L'intégrité du fichier « `Release` » de premier niveau est vérifiée par l'infrastructure cryptographique appelée secure apt comme décrit dans apt-secure(8).

— Le fichier de signature chiffré « `Release.gpg` » est créé à partir du fichier « `Release` » réel de plus haut niveau et de la clé secrète de l'archive Debian.
— Les clés publiques des archives de Debian sont installées localement par le paquet récent `debian-archive-keyring`.
— Le système **secure APT** vérifie automatiquement l'intégrité du fichier téléchargé « `Release` » à l'aide du fichier « `Release.gpg` » de signature et des clés publiques d'archive installées localement.
— L'intégrité de tous les fichiers « `Packages` » et « `Sources` » est vérifiée en utilisant les valeurs des sommes MD5 se trouvant dans son fichier de plus haut niveau « `Release` ». L'intégrité de tous les fichiers de paquets est vérifiée en utilisant les valeurs des sommes MD5 se trouvant dans les fichiers « `Packages` » et « `Sources` ». Consultez debsums(1) et Section 2.4.2.
— Comme la vérification de la signature cryptographique est un processus beaucoup plus consommateur de processeur que les calculs de sommes MD5, l'utilisation d'une somme MD5 pour chacun des paquets tout en utilisant une signature cryptographique pour le fichier « `Release` » de plus haut niveau allie une bonne sécurité avec de bonnes performances (consultez Section 10.3).

Si une entrée dans la **liste des sources** spécifie l'option "`signed-by`", l'intégrité de son fichier téléchargé "`Release`" est vérifiée en utilisant la clé publique spécifiée. Cela est utile lorsque la **liste des sources** contient des archives non Debian.

Vous pouvez également vérifier manuellement l'intégrité du fichier « `Release` » avec le fichier « `Release.gpg` » et la clé publique d'archive de Debian publiée sur ftp-master.debian.org en utilisant `gpg`.

2.5.3 Fichiers « Release » de niveau de l'archive

ASTUCE
Les fichiers « `Release` » de niveau de l'archive sont utilisés pour la règle d'apt_preferences(5).

Il existe des fichiers « `Release` » de niveau de l'archive pour tous les sites d'archives spécifiés par la **liste des sources**, tels que « `http://deb.debian.org/debian/dists/binstable/main/binary-amd64/Release` » ou « `http://deb.debian.org/debian/dists/sid/main/binary-amd64/Release` » comme suit :

```
Archive: unstable
Origin: Debian
Label: Debian
Component: main
Architecture: amd64
```



Attention
Pour l'entrée « `Archive:` », les noms de version (« `stable` », « `testing` », « `unstable` », ...) sont utilisés dans l'archive Debian alors que les noms de code (« `trusty` », « `xenial` », « `artful` », ...) sont utilisés dans l'archive Ubuntu.

Pour certaines archives, comme `experimental` et `trixie-backports`, qui contiennent des paquets qui ne devraient pas être installés automatiquement, il y a une ligne supplémentaire, par exemple « `http://deb.debian.org/debi` comme suit :

```
Archive: experimental
Origin: Debian
Label: Debian
NotAutomatic: yes
Component: main
Architecture: amd64
```

Remarquez que les archives normales, sans « `NotAutomatic: yes` », la valeur par défaut de « Pin-Priority » est de 500, alors que pour les archives spéciales avec « `NotAutomatic: yes` » la valeur par défaut de « Pin-Priority » est de 1 (consultez apt_preferences(5) et Section 2.7.7).

2.5.4 Récupérer les métadonnées d'un paquet

Lorsque des outils d'APT, comme `aptitude`, `apt-get`, `synaptic`, `apt-file`, `auto-apt`, etc., sont utilisés, il faut mettre à jour les copies locales des métadonnées contenant les informations de l'archive Debian. Ces copies locales ont les noms de fichier suivants, correspondant aux noms de la `distribution`, de la `section` et de l'`architecture` indiquées dans la **liste des sources** (consulter Section 2.1.5).

— « `/var/lib/apt/lists/deb.debian.org_debian_dists_`*`distribution`*`_Release` »
— « `/var/lib/apt/lists/deb.debian.org_debian_dists_`*`distribution`*`_Release.gpg` »
— « `/var/lib/apt/lists/deb.debian.org_debian_dists_`*`distribution_area`*`_binary-`*`architecture`*`_P`
— « `/var/lib/apt/lists/deb.debian.org_debian_dists_`*`distribution_area`*`_source_Sources` »
— « `/var/cache/apt/apt-file/deb.debian.org_debian_dists_`*`distribution`*`_Contents-`*`architecture`*
(pour `apt-file`)

Les quatre premiers types de fichiers sont partagés par toutes les commandes APT pertinentes et mis à jour depuis la ligne de commande par « `apt-get update` » ou « `aptitude update` ». Les métadonnées « `Packages` » sont mises à jour si une ligne « `deb` » est spécifiée dans la **liste des sources**. Les métadonnées « `Sources` » sont mises à jour si une ligne « `deb-src` » est spécifiée dans la **liste des sources**.

Les métadonnées « `Packages` » et « `Sources` » contiennent une entrée « `Filename:` » pointant vers l'emplacement du paquet binaire et du paquet source. Actuellement, ces paquets sont situés dans l'arborescence du répertoire « `pool/` » afin d'améliorer le passage d'une version à l'autre.

On peut effectuer des recherches interactivement dans les copies locales des métadonnées « `Packages` » à l'aide d'`aptitude`. La commande de recherche spécialisée grep-dctrl(1) peut effectuer des recherches dans les copies locales des métadonnées « `Packages` » et « `Sources` ».

La copie locale des métadonnées « `Contents-`*`architecture`* » peut être mise à jour par « `apt-file update` », son emplacement est différent des quatre autres. Consultez apt-file(1). (`auto-apt` utilise par défaut un emplacement différent pour la copie locale de « `Contents-`*`architecture`*`.gz` ».)

2.5.5 État des paquets pour APT

En plus des métadonnées récupérées par téléchargement, l'outil APT des versions ultérieures à Lenny enregistre l'état de l'installation généré localement dans « `/var/lib/apt/extended_states` » qui est utilisé par tous les outils APT afin de suivre tous les paquets installés automatiquement.

2.5.6 État des paquets pour aptitude

En plus des métadonnées récupérées par téléchargement, la commande `aptitude` enregistre l'état de l'installation généré localement dans « `/var/lib/aptitude/pkgstates` » qu'il est le seul à utiliser.

2.5.7 Copies locales des paquets téléchargés

Tous les paquets ayant été téléchargés au moyen du mécanisme APT sont enregistrés dans le répertoire « `/var/cache/ap` jusqu'à ce qu'ils en soient supprimés.

Cette politique de nettoyage des fichiers de cache pour `aptitude` peut être spécifiée sous « `Options` » → « `Préférences` et peut être déclenchée manuellement par le menu « `Nettoyer le cache des paquets` » ou « `Enlever les fichiers périmés` » sous « `Actions` ».

2.5.8 Nom de fichier d'un paquet Debian

Les fichiers de paquets Debian ont une structure de nom particulière.

type de paquet	structure du nom
Paquet binaire (encore appelé deb)	*package-name_upstream-version-debian.revision_architec*
Le paquet binaire de l'installateur de Debian (connu sous le nom udeb)	*package-name_upstream-version-debian.revision_architec*
paquet source (source amont)	*package-name_upstream-version-debian.revision*.orig.tar
Paquet source `1.0` (modifications Debian)	*package-name_upstream-version-debian.revision*.diff.gz
Paquet source `3.0 (quilt)` (modifications Debian)	*package-name_upstream-version-debian.revision*.debian.ta
paquet source (description)	*package-name_upstream-version-debian.revision*.dsc

Table 2.15 – Structure du nom des paquets Debian :

ASTUCE
Seuls les formats de paquets source de base sont décrits ici. Consultez dpkg-source(1) pour davantage d'informations.

nom de la composante	caractères utilisables (expressions rationnelles étendues)	existence
package-name	`[a-z0-9][-a-z0-9.+]+`	nécessaire
epoch:	`[0-9]+:`	optionnel
upstream-version	`[-a-zA-Z0-9.+:]+`	nécessaire
debian.revision	`[a-zA-Z0-9.+~]+`	optionnel

Table 2.16 – Caractères utilisables pour chacune des composantes des noms de paquets Debian

Note
Vous pouvez vérifier l'ordre des versions d'un paquet à l'aide de dpkg(1), par exemple, « `dpkg --compare-versions 7.0 gt 7.~pre1 ; echo $?` ».

Note
L'installateur Debian (d-i) utilise `udeb` comme extension du nom de fichier de ses paquets binaires plutôt que le `deb` normal. Un paquet `udeb` est un paquet `deb` allégé dont certaines parties non essentielles du contenu, comme la documentation, sont supprimées afin d'économiser de la place en relâchant les exigences de la charte des paquets. Les paquet `deb` et `udeb` partagent la même structure de paquet. Le « u » signifie micro.

2.5.9 La commande dpkg

dpkg(1) est l'outil de plus bas niveau pour la gestion des paquets de Debian. C'est un outil très puissant et il faut l'utiliser avec précaution.

Lors de l'installation d'un paquet appelé « *nom_paquet* », `dpkg` le traite selon l'ordre suivant :

1. dépaquetage du fichier deb (équivalent à « `ar -x` ») ;
2. exécution de « *nom_paquet*`.preinst` » en utilisant debconf(1) ;
3. installation du contenu du paquet sur le système (équivalent à « `tar -x` » ;
4. exécution de « *nom_paquet*`.postinst` » en utilisant debconf(1).

Le système `debconf` fournit une interaction standardisée avec l'utilisateur avec la prise en charge de I18N and L10N (Chapitre 8).

fichier	description du contenu
`/var/lib/dpkg/info/`*package_name*`.conffiles`	liste des fichiers de configuration. (modifiables par l'utilisateur)
`/var/lib/dpkg/info/`*package_name*`.list`	liste des fichiers et répertoires installés par le paquet
`/var/lib/dpkg/info/`*package_name*`.md5sums`	liste des valeurs de hachage MD5 pour les fichiers installés par le paquet
`/var/lib/dpkg/info/`*package_name*`.preinst`	script du paquet à exécuter avant l'installation du paquet
`/var/lib/dpkg/info/`*package_name*`.postinst`	script du paquet à exécuter après l'installation du paquet
`/var/lib/dpkg/info/`*package_name*`.prerm`	script du paquet à exécuter avant la suppression du paquet
`/var/lib/dpkg/info/`*package_name*`.postrm`	script du paquet à exécuter après la suppression du paquet
`/var/lib/dpkg/info/`*package_name*`.config`	script du paquet pour le système `debconf`
`/var/lib/dpkg/alternatives/`*package_name*	information d'alternative utilisée par la commande `update-alternatives` command
`/var/lib/dpkg/available`	information de disponibilité de tous les paquets
`/var/lib/dpkg/diversions`	information sur les détournements utilisés par dpkg(1) et établis avec dpkg-divert(1)
`/var/lib/dpkg/statoverride`	information de remplacement de statut utilisé par dpkg(1) et établi avec dpkg-statoverride(1)
`/var/lib/dpkg/status`	informations d'état pour tous les paquets
`/var/lib/dpkg/status-old`	fichier de sauvegarde de première génération du fichier « `var/lib/dpkg/status` »
`/var/backups/dpkg.status*`	fichier de sauvegarde de seconde génération du fichier « `var/lib/dpkg/status` »

Table 2.17 – Fichiers particuliers créés par `dpkg`

Le fichier « `status` » est aussi utilisé par des outils comme dpkg(1), « `dselect update` » et « `apt-get -u dselect-upgrade` ».

La commande de recherche spécialisée grep-dctrl(1) peut rechercher des copies locales des métadonnées « `status` » et « `available` ».

ASTUCE
Dans l'environnement de l'installateur debian, la commande `udpkg` est utilisée pour ouvrir les paquets `udeb`. La commande `udpkg` est une version allégée de la commande `dpkg`.

2.5.10 La commande update-alternatives

Le système Debian possède un mécanisme pour installer paisiblement des paquets qui présentent un certain recouvrement en utilisant update-alternatives(1). Par exemple, vous pouvez faire que la commande `vi` choisisse de lancer `vim` alors que les paquets `vim` et `nvi` sont tous deux installés.

```
$ ls -l $(type -p vi)
lrwxrwxrwx 1 root root 20 2007-03-24 19:05 /usr/bin/vi -> /etc/alternatives/vi
$ sudo update-alternatives --display vi
...
$ sudo update-alternatives --config vi
  Selection    Command
 -----------------------------------------------
      1        /usr/bin/vim
*+    2        /usr/bin/nvi

Enter to keep the default[*], or type selection number: 1
```

Le système d'alternatives de Debian utilise des liens symboliques dans « `/etc/alternatives/` » pour enregistrer ses sélections. Le processus de sélection utilise le fichier correspondant de « `/var/lib/dpkg/alternatives/` ».

2.5.11 Commande dpkg-statoverride

Stat overrides, fournie par la commande dpkg-statoverride(1) est un moyen d'indiquer à dpkg(1) d'utiliser un propriétaire ou un mode différent pour un **fichier** lorsqu'un paquet est installé. Si « `--update` » est indiqué et que le fichier existe, il est immédiatement configuré avec le nouveau propriétaire et le nouveau mode.



Attention
Une modification directe par l'administrateur du propriétaire ou du mode d'un **fichier** dont le propriétaire est le paquet en utilisant les commandes `chmod` ou `chown` sera réinitialisée lors d'une nouvelle mise à niveau du paquet.

Note
J'utilise ici le mot **fichier**, mais en réalité, ce peut être n'importe quel objet d'un système de fichiers que gère `dpkg`, y compris les répertoires, les périphériques, etc.

2.5.12 Commande dpkg-divert

Les fichiers **diversions** fournis par la commande dpkg-divert(1) sont un moyen de forcer dpkg(1) à ne pas installer un fichier à son emplacement par défaut, mais à un emplacement **détourné** (« diverted »). L'utilisation de `dpkg-divert` est destinée à la maintenance de paquets par des scripts. Son utilisation occasionnelle par l'administrateur du système est obsolète.

2.6 Récupérer un système cassé

En utilisant la distribution `testing` ou `unstable`, l'administrateur peut avoir à restaurer le système à partir d'une situation où la gestion des paquets est défectueuse.



Attention
Certaines des méthodes décrites ici sont des actions très risquées. Vous avez été prévenu !

2.6.1 Incompatibilité avec une ancienne configuration de l'utilisateur

Si un programme avec interface graphique présente une instabilité après une mise à niveau amont importante, vous devriez songer à des interférences avec les anciens fichiers de configuration locaux qu'il avait créés. S'il est stable avec un compte d'utilisateur fraîchement créé, cette hypothèse est confirmée. (C'est un bogue d'empaquetage du paquet et c'est le plus souvent évité par le responsable du paquet.)

Pour retrouver la stabilité, vous devrez déplacer les fichiers de configuration locaux et redémarrer le programme ayant une interface graphique. Il vous faudra peut-être lire le contenu de l'ancien fichier de configuration pour retrouver plus tard vos informations de configuration. (Ne les effacez pas trop rapidement.)

2.6.2 Erreurs de mise en cache des données du paquet

Les erreurs de mise en cache des données du paquet provoquent des erreurs intrigantes, telles que "GPG error: ... invalid: BADSIG ..." avec APT.

Vous devez supprimer toutes les données mises en cache avec « `sudo rm -rf /var/lib/apt/*` » et réessayer. (Si `apt-cacher-ng` est utilisé, vous devez également exécuter « `sudo rm -rf /var/cache/apt-cacher-ng/*` ».)

2.6.3 Récupération avec la commande dpkg

Comme `dpkg` est un outil de gestion de paquets de très bas niveau, il peut fonctionner sans connexion réseau.

Supposons que le paquet « `toto` » soit défectueux et qu'il faille le corriger.

Vous pouvez trouver des copies de l'ancienne version du paquet `toto` sans bogue dans le répertoire de cache des paquets « `/var/cache/apt/archives/` ». (Si ce n'est pas le cas, vous pouvez télécharger l'archive depuis https://snapshot.debian.org/ ou la copier depuis le cache des paquets d'une machine qui fonctionne.)

Si vous pouvez démarrer le système, vous pouvez l'installer avec la commande suivante :

```
# dpkg -i /path/to/foo_old_version_arch.deb
```

Si la tentative d'installation d'un paquet de cette manière échoue en raison de la violation de certaines dépendances et que vous voulez vraiment le faire, vous pouvez, en dernier ressort, outrepasser les dépendances en utilisant les options « `--ignore-depends` », « `--force-depends` » de `dpkg` et d'autres options. Si vous le faites, vous aurez un sérieux effort à faire pour restaurer les dépendances correctes par la suite. Consultez dpkg(8) pour davantage d'informations.

Note
Si votre système est sérieusement cassé, vous devriez faire une sauvegarde complète du système dans un endroit sûr (consultez Section 10.2) et effectuer une installation propre. Cela demande moins de temps et donne, en fin de compte, de meilleurs résultats.

ASTUCE
Si la casse du système est minime, vous pouvez faire un retour en arrière (downgrade) de tout le système comme dans Section 2.7.11 en utilisant le système de plus haut niveau APT.

2.6.4 Échec d'installation à cause de dépendances manquantes

Si vous forcez l'installation d'un paquet avec « `sudo dpkg -i ...` » sur un système sans que tous les paquets de dépendance soient installés, l'installation du paquet échouera puisque partielle.

Vous devriez installer tous les paquets dépendants en utilisant à plusieurs reprises « `sudo dpkg -i ...` » ou en ayant recours à :

```
# apt --fix-broken install
```

Puis, configurez tous les paquets partiellement installés à l'aide de la commande suivante :

```
# dpkg --configure -a
```

2.6.5 Différents paquets ayant des fichiers communs

Les systèmes de gestion d'archive au niveau du paquet, tels qu'aptitude(8) ou apt-get(8) ne tenteront même pas, grâce aux dépendances des paquets, d'installer des paquets ayant des fichiers qui se superposent (consultez Section 2.1.7).

Des erreurs du responsable du paquet ou le déploiement de sources d'archives mélangées et incohérentes (consultez Section 2.7.6) par l'administrateur du système peuvent créer une situation où les dépendances des paquets sont décrites de manière incorrecte. Lorsque, dans une telle situation, vous installez un paquet qui écrase des fichiers en utilisant aptitude(8) ou apt-get(8), dpkg(1) qui dépaquette le paquet va retourner une erreur au programme appelant sans écraser les fichiers existants.



Attention
L'utilisation de programmes tierce partie introduit un risque significatif par l'intermédiaire des scripts du responsable qui sont lancés avec les privilèges de l'administrateur et peuvent effectuer n'importe quoi sur votre système. La commande dpkg(1) ne protège que contre l'écrasement des fichiers lors du dépaquetage.

Vous pouvez contourner un tel problème d'installation cassée en supprimant d'abord l'ancien paquet, *`ancien_paquet`*, qui pose des problèmes.

```
$ sudo dpkg -P old-package
```

2.6.6 Corriger les scripts cassés des paquets

Lorsqu'une commande dans le script du paquet retourne une erreur pour une raison quelconque et que le script retourne une erreur, le système de gestion des paquets arrête son action et se termine en laissant des paquets partiellement installés. Lorsqu'un paquet comporte des bogues dans les scripts de suppression, le paquet peut devenir impossible à supprimer et assez déplaisant.

Pour les problèmes avec le script de paquet de « *`nom_paquet`* », il vous faudra regarder dans les scripts du paquet suivants :

— « `/var/lib/dpkg/info/`*`nom_paquet`*`.preinst` »
— « `/var/lib/dpkg/info/`*`nom_paquet`*`.postinst` »
— « `/var/lib/dpkg/info/`*`nom_paquet`*`.prerm` »
— « `/var/lib/dpkg/info/`*`nom_paquet`*`.postrm` »

Éditez le script du paquet posant problème avec le compte de l'administrateur en utilisant les techniques suivantes :

— désactiver la ligne posant problème avec un « # » en tête de ligne ;
— forcer un retour avec succès en ajoutant à la fin de la ligne qui pose problème « `|| true` ».

Puis, configurez tous les paquets partiellement installés à l'aide de la commande suivante :

```
# dpkg --configure -a
```

2.6.7 Récupérer les données de sélection des paquets

Si le fichier « /var/lib/dpkg/status » est corrompu pour une raison quelconque, le système Debian perd les données de paquets sélectionnés et est sérieusement endommagé. Regardez l'ancienne version du fichier « /var/lib/dpkg/s dans « /var/lib/dpkg/status-old » ou « /var/backups/dpkg.status.* ».

Conserver « /var/backups/ » sur une partition séparée peut être une bonne idée car ce répertoire contient de nombreuses données importantes du système.

Pour les casses sévères, je recommande de faire une réinstallation propre après avoir fait une sauvegarde du système. Même si vous avez perdu tout ce qui se trouve dans « /var/ », vous pouvez encore récupérer certaines informations depuis les répertoires qui se trouvent dans « /usr/share/doc/ » afin de vous guider dans votre nouvelle installation.

Réinstaller un système (de bureau) minimum.

```
# mkdir -p /path/to/old/system
```

Montez l'ancien système sur « /chemin/vers/ancien/système/ ».

```
# cd /path/to/old/system/usr/share/doc
# ls -1 >~/ls1.txt
# cd /usr/share/doc
# ls -1 >>~/ls1.txt
# cd
# sort ls1.txt | uniq | less
```

Le système vous présentera alors les noms de paquets à installer. (Il peut y avoir des noms qui ne soient pas des noms de paquets, comme, par exemple, « texmf ».)

2.7 Astuces pour la gestion des paquets

Pour la simplicité, les exemples de **liste des sources** dans cette section sont présentés sous « /etc/apt/sources.list » dans style ligne par ligne après la publication de trixie.

2.7.1 Qui a envoyé le paquet ?

Bien que le nom du responsable figure dans « /var/lib/dpkg/available » et « /usr/share/doc/nom_paquet/cha procure quelques informations sur « qui se trouve derrière l'activité de construction des paquets », celui qui a réellement envoyé le paquet est un peu obscur. who-uploads(1) dans le paquet devscripts identifie celui qui a réellement envoyé les paquets sources Debian.

2.7.2 Diminuer la bande passante utilisée par APT

Si vous désirez limiter la bande passante utilisée par APT à, par exemple, 800Kib/sec (=100kio/sec), vous devrez configurer APT avec son paramètre de configuration comme suit :

```
APT::Acquire::http::Dl-Limit "800";
```

2.7.3 Chargement et mise à niveau automatique de paquets

Le paquet `apt` est diffusé avec son propre script d'événements planifiés (cron) « `/etc/cron.daily/apt` » afin de gérer le téléchargement automatique de paquets. Ce script peut être amélioré afin d'effectuer la mise à niveau automatique des paquets en installant le paquet `unattended-upgrades`. Cela peut être personnalisé à l'aide de paramètres se trouvant dans « `/etc/apt/apt.conf.d/02backup` » et « `/etc/apt/apt.conf.d/50unattended-upgrades` » comme décrit dans « `/usr/share/doc/unattended-upgrades/README` ».

Le paquet `unattended-upgrades` est principalement destiné à des mises à jour de sécurité des systèmes `stable`. Si le risque de casser un système `stable` existant par la mise à niveau automatique est plus faible que celui d'avoir un système cassé par un intrus utilisant une de ses failles de sécurité qui a été fermée par une mise à jour de sécurité, vous devriez envisager d'utiliser cette mise à niveau automatique avec les paramètres de configuration suivants :

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::Unattended-Upgrade "1";
```

Si vous faites tourner un système `testing` ou `unstable`, vous ne devriez pas utiliser les mises à niveau automatiques car cela cassera probablement votre système un jour ou l'autre. Même dans le cas de `testing` ou `unstable`, vous pourrez télécharger des paquets à l'avance afin de gagner du temps pour la mise à niveau interactive avec les paramètres de configuration suivants :

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::Unattended-Upgrade "0";
```

2.7.4 Mises à jour et rétroportages

Il existe des stable-updates (« `trixie`-updates » pendant le cycle de publication de `trixie`-comme-`stable`) et des archives de backports.debian.org fournissent des paquets de mise à niveau pour `stable`.

Afin d'utiliser ces archives, placez la liste de toutes les archives nécessaires dans le fichier « `/etc/apt/sources.list` » de la manière suivante :

```
deb http://deb.debian.org/debian/ trixie main non-free-firmware contrib non-free
deb http://security.debian.org/debian-security trixie-security main non-free-firmware  ↩
    contrib non-free
deb http://deb.debian.org/debian/ trixie-updates main non-free-firmware contrib non-free
deb http://deb.debian.org/debian/ trixie-backports main non-free-firmware contrib non-free
```

Il n'est pas nécessaire de définir explicitement la valeur de Pin-Priority dans le fichier « `/etc/apt/preferences` ». Quand de nouveaux paquets sont disponibles, la configuration par défaut fournit les mises à niveau les plus saines (consultez Section 2.5.3).

— Tous les paquets installés les plus anciens sont mis à niveau vers les plus récents à partir de `trixie-updates`.
— Seuls les paquets les plus anciens installés à partir de `trixie-backports` sont mis à niveau vers les plus récents à partir de `trixie-backports`.

Chaque fois que vous désirerez installer un paquet nommé « *`nom_paquet`* » avec ses dépendances depuis l'archive `trixie-backports` vous-même, vous utiliserez la commande suivante en changeant de version cible avec l'option « `-t` » :

```
$ sudo apt-get install -t trixie-backports package-name
```



AVERTISSEMENT
N'installez pas trop de paquets à partir des archives backports.debian.org. Cela peut causer des complications de dépendance. Voir Section 2.1.11 pour les solutions alternatives.

2.7.5 Archives de paquets externes



AVERTISSEMENT
Vous devez être conscient que le paquet externe obtient les privilèges du superutilisateur sur votre système. Vous ne devriez utiliser que l'archive de confiance de paquets externes. Voir Section 2.1.11 pour les solutions alternatives.

Vous pouvez utiliser APT sécurisé avec l'archive de paquets externes compatible Debian en l'ajoutant à la **liste des sources** et son fichier de clé d'archive dans le répertoire "`/etc/apt/trusted.gpg.d/`". Voir sources.list(5) et apt-secure(8).

2.7.6 Paquets de sources mixtes d'archives sans apt-pinning



Attention
Installer des paquets provenant de sources d'archives mélangées n'est pas pris en charge par la distribution officielle de Debian sauf pour la prise en charge officielle d'une combinaison particulière d'archives telle que `stable` avec security updates et stable-updates.

Voici un exemple des opérations pour inclure des paquets d'une version amont spécifique plus récente se trouvant dans `unstable` pour une seule occasion tout en suivant `testing` :

1. modifier le fichier « `/etc/apt/sources.list` » de manière temporaire avec la seule entrée « `unstable` » ;
2. lancer « `aptitude update` » ;
3. lancer « `aptitude install` *`nom_paquet`* » ;
4. rétablir le fichier « `/etc/apt/sources.list` » pour `testing`.
5. lancer « `aptitude update` » ;

Vous ne créez pas le fichier « `/etc/apt/preferences` » et vous n'avez pas besoin de vous préoccuper de l'épinglage apt (« **apt-pinning** ») avec cette approche manuelle. Mais c'est très lourd.



Attention
En utilisant une source d'archive mixte, vous devez vous assurer par vous-même de la compatibilité des paquets car Debian ne la garantit pas. S'il existe des incompatibilités de paquets, vous pouvez casser votre système. Vous devrez être capable d'apprécier ces exigences techniques. L'utilisation de sources mixtes d'archives aléatoires est une opération entièrement facultative et son utilisation est quelque chose que je ne vous encourage pas à faire.

Les règles générales pour l'installation de paquets de différentes archives sont les suivantes :

— Les paquets non binaires (« `Architecture: all` ») sont plus **sûrs** à installer :
  — paquets de documentation : pas d'exigence particulière
  — paquet de programmes d'interpréteur : un interpréteur compatible doit être disponible
— Les paquets binaires (qui ne sont pas « `Architecture: all` ») sont confrontés à de nombreux barrages et ne sont **pas sûrs** à installer
  — compatibilité des versions de bibliothèques (y compris « `libc` »)
  — compatibilité des versions des programmes utilitaires en rapport
  — compatibilité avec l'ABI du noyau
  — compatibilité avec l'ABI C++
  — …

Note
De manière à rendre un paquet **plus sûr** à installer, certains programmes commerciaux binaires et non libres peuvent être fournis liés avec des bibliothèques complètement statiques. Vous devrez quand même vérifier leurs problèmes de compatibilité avec l'ABI, etc.

Note
Sauf pour contourner pour un court terme un paquet cassé, l'installation de paquets binaires d'archives non Debian est généralement mauvaise idée. Vous devriez chercher toutes les solutions techniques alternatives plus sûres qui sont compatibles avec votre système Debian actuel (voir Section 2.1.11).

2.7.7 Ajustement de la version candidate avec apt-pinning



AVERTISSEMENT
Use of **apt-pinning** technique by a novice user is sure call for major troubles. You must avoid using this technique except when you absolutely need it. See Section 2.1.11 for alternative solutions.

Sans le fichier « `/etc/apt/preferences` », le système APT choisit, en utilisant la chaîne de version, la dernière version disponible comme **version candidate**. C'est l'état normal et l'utilisation la plus recommandée du système APT. Toutes les combinaisons d'archives officiellement prises en charge n'exigent pas le fichier « `/etc/apt/preferences` » car certaines archives qui ne peuvent pas être utilisées comme source des mises à jour automatiques sont marquées **NotAutomatic** et gérées proprement.

ASTUCE
La règle de comparaison de la chaîne de version peut être vérifiée avec, par exemple « `dpkg --compare-versions ver1.1 gt ver1.1~1; echo $?` » (consulter dpkg(1)).

Lorsque vous installez régulièrement des paquets depuis un mélange de sources d'archives (consulter Section 2.7.6), vous pouvez automatiser ces opérations compliquées en créant le fichier « `/etc/apt/preferences` » avec les entrées appropriées et en ajustant la règle de sélection des paquets pour la **version candidate** comme décrit dans apt_preferences(5). C'est appelé **apt-pinning** (épinglage avec apt).

Lorsque vous utilisez **apt-pinning**, vous devez assurer la compatibilité des paquets par vous-même puisque Debian ne le garantit pas. **apt-pinning** est une opération totalement facultative et son utilisation n'est pas quelque chose que je vous encourage à utiliser.

Les fichiers Release de niveau de l'archive (consulter Section 2.5.3) sont utilisés pour la règle de apt_preferences(5). Par conséquent **apt-pinning** fonctionne seulement avec le nom de « suite » pour les archives normales de Debian et les archives de sécurité de Debian, ce qui est différent des archives d'Ubuntu. Par exemple, dans le fichier « `/etc/apt/preferences` », il est possible de mettre « `Pin: release a=unstable` », mais pas « `Pin: release a=sid` ».

Lorsque vous utilisez une archive ne venant pas de Debian en tant que partie d'**apt-pinning**, vous devez vérifier ce pour quoi elles sont prévues et aussi vérifier leur crédibilité. Par exemple, Ubuntu et Debian ne sont pas prévues pour être mélangées.

Note
Même si vous ne créez pas le fichier « `/etc/apt/preferences` », vous pouvez effectuer des opérations assez complexes sur le système sans **apt-pinning** (consulter Section 2.6.3 et Section 2.7.6).

Voici une explication simplifiée de la technique d'**apt-pinning**.

Le système APT choisit la **mise à niveau** du paquet de plus haute priorité d'épinglage (« Pin-Priority ») dans la liste des sources de paquets disponible dans le fichier « `/etc/apt/sources.list` » comme paquet de « **version candidate** ». Si la priorité d'épinglage du paquet est supérieure à 1000, cette restriction de version pour la **mise à niveau** est levée afin de permettre le retour vers une version précédente (consulter Section 2.7.11).

La valeur de priorité d'épinglage de chaque paquet est définie par l'entrée « Pin-Priority » dans le fichier « `/etc/apt/prefe` ou utilise sa valeur par défaut.

priorité d'épinglage	**effet d'apt-pinning sur le paquet**
1001	installer le paquet même s'il s'agit d'un retour en arrière
990	utilisé par défaut pour l'archive **version cible**
500	utilisé par défaut pour l'archive **normale**
100	utilisé par défaut pour l'archive **non automatique mais mises à niveau automatiques**
100	utilisé pour le paquet **installé**
1	utilisé par défaut pour l'archive **non automatique**
-1	**ne jamais installer** le paquet même s'il est recommandé

Table 2.18 – Liste de valeurs remarquables des priorités d'épinglage pour la technique de**apt-pinning**

L'archive **version cible** peut être définie par l'option de ligne de commande, par exemple, "`apt-get install -t testing un_paquet`"

L'archive **non automatique mais mises à niveau automatiques** est définie par le serveur d'archive dont le fichier Release au niveau de l'archive (consultez Section 2.5.3) contient à la fois « `NotAutomatic: yes` » et « `ButAutomaticUpgra yes` ». L'archive **non automatique** est définie par le serveur d'archive dont le fichier Release au niveau de l'archive contient « `NotAutomatic: yes` ».

La **situation d'épinglage apt** de *paquet* provenant de sources d'archive multiple est affichée par « `apt-cache policy` *`paquet`* ».

— Une ligne commençant par « `Package pin:` » affiche la version d'**épinglage** du paquet si l'association n'est définie que pour *paquet* par exemple, « `Package pin: 0.190` ».

— Il n'existe pas de ligne avec « `Package pin:` » s'il n'y a pas d'association définie uniquement avec *paquet*.

— La valeur de Pin-Priority associée uniquement avec *paquet* est affichée sur la partie droite de toutes les chaînes de version, par exemple, « `0.181 700` ».

— « `0` » est affiché à droite de toutes les chaînes de version s'il n'y a pas d'association définie avec uniquement *paquet*, par exemple, « `0.181 0` ».

— Les valeurs de Pin-Priority des archives (définies par « `Package: *` » dans le fichier « `/etc/apt/preferences` ») sont affichées sur la gauche de tous les chemins vers les archives, par exemple, « `100 http://deb.debian.org/deb trixie-backports/main Packages` ».

2.7.8 Blocage des paquets installés par « Recommends »



AVERTISSEMENT
Use of **apt-pinning** technique by a novice user is sure call for major troubles. You must avoid using this technique except when you absolutely need it. See Section 2.1.11 for alternative solutions.

Si vous ne voulez pas installer des paquets particuliers automatiquement avec « Recommends », vous devez créer le fichier « `/etc/apt/preferences` » et y placer la liste explicite de tous ces paquets au début du fichier comme suit :

```
Package: package-1
Pin: version *
Pin-Priority: -1

Package: package-2
Pin: version *
Pin-Priority: -1
```

2.7.9 Suivre `testing` avec quelques paquets d'`unstable`



AVERTISSEMENT
Use of **apt-pinning** technique by a novice user is sure call for major troubles. You must avoid using this technique except when you absolutely need it. See Section 2.1.11 for alternative solutions.

Voici un exemple de technique d'**épinglage apt** permettant d'inclure de manière régulière une version amont plus récente de paquets spécifiques se trouvant dans `unstable` tout en suivant `testing`. Listez toutes les archives nécessaires dans le fichier « `/etc/apt/sources.list` » de la manière suivante :

```
deb http://deb.debian.org/debian/ testing main contrib non-free
deb http://deb.debian.org/debian/ unstable main contrib non-free
deb http://security.debian.org/debian-security testing-security main contrib
```

Configurez le fichier « `/etc/apt/preferences` » comme suit :

```
Package: *
Pin: release a=unstable
Pin-Priority: 100
```

Lorsque vous désirez installer un paquet appelé « *`nom_paquet`* » avec ses dépendances depuis l'archive `unstable` avec cette configuration, vous utilisez la commande suivante qui modifie la version cible avec l'option « `-t` » (la priorité d'épinglage de `unstable` devient 990).

```
$ sudo apt-get install -t unstable package-name
```

Avec cette configuration, l'exécution habituelle de « `apt-get upgrade` » et « `apt-get dist-upgrade` » (ou « `aptitud safe-upgrade` » et « `aptitude full-upgrade` » met à niveau les paquets qui avaient été installés depuis l'archive `testing` en utilisant l'archive `testing` actuelle et les paquets qui avaient été installés depuis l'archive `unstable` en utilisant l'archive `unstable` actuelle.



Attention
Faites bien attention à ne pas supprimer l'entrée « `testing` » du fichier « `/etc/apt/sources.list` ». Sans l'entrée « `testing` », le système APT mettra à niveau les paquets en utilisant la nouvelle archive `unstable`.

ASTUCE
J'édite habituellement le fichier « `/etc/apt/sources.list` » en commentant l'entrée correspondant à l'archive « `unstable` » juste après avoir effectué les opérations ci-dessus. Cela évite un processus de mise à jour lent en raison du nombre trop important d'entrées dans le fichier « `/etc/apt/sources.list` » bien que cela ne permette pas de mettre à niveau les paquets qui avaient été installés depuis l'archive `unstable` en utilisant l'archive `unstable` actuelle.

ASTUCE
Si on utilise « `Pin-Priority: 1` » à la place de « `Pin-Priority: 100` » dans le fichier « `/etc/apt/preferences` », les paquets déjà installés ayant une valeur de Pin-Priority de 100 ne seront pas mis à niveau depuis l'archive `unstable` même si l'entrée « `testing` » du fichier « `/etc/apt/sources.list` » est supprimée.

Si vous désirez suivre automatiquement un paquet particulier dans `unstable` sans une installation initiale « `-t unstable` », vous devrez créer le fichier « `/etc/apt/preferences` » et y placer la liste explicite de tous ces paquets au début du fichier de la manière suivante :

```
Package: package-1
Pin: release a=unstable
Pin-Priority: 700

Package: package-2
Pin: release a=unstable
Pin-Priority: 700
```

Cela définit la valeur de Pin-Priority pour chacun de ces paquets spécifiques. Par exemple, pour suivre la dernière version `unstable` de cette « Référence Debian » en français, vous devrez ajouter les entrées suivantes dans le fichier « `/etc/apt/preferences` ».

```
Package: debian-reference-en
Pin: release a=unstable
Pin-Priority: 700

Package: debian-reference-common
Pin: release a=unstable
Pin-Priority: 700
```

ASTUCE
Cette technique d'**apt-pinning** est valable même si vous suivez l'archive `stable`. Jusqu'à présent et selon mon expérience, les paquets de documentation ont toujours été sûrs à installer depuis l'archive `unstable`.

2.7.10 Suivre `unstable` avec quelques paquets d'`experimental`



AVERTISSEMENT
Use of **apt-pinning** technique by a novice user is sure call for major troubles. You must avoid using this technique except when you absolutely need it. See Section 2.1.11 for alternative solutions.

Voici un autre exemple de technique d'**épinglage apt** destinée à inclure une version amont plus récente de paquets spécifiques se trouvant dans `experimental` tout en suivant `unstable`. Vous donnez la liste de toutes les archives nécessaires dans fichier « `/etc/apt/sources.list` » de la manière suivante :

```
deb http://deb.debian.org/debian/ unstable main contrib non-free
deb http://deb.debian.org/debian/ experimental main contrib non-free
deb http://security.debian.org/ testing-security main contrib
```

La valeur de Pin-Priority par défaut pour l'archive `experimental` est toujours de 1 (<<100) car c'est une archive **non automatique** (consultez Section 2.5.3). Il n'y a pas besoin de définir explicitement de valeur de Pin-Priority dans le fichier « `/etc/apt/preferences` » simplement pour utiliser l'archive `experimental` à moins que vous ne désiriez suivre des paquets particuliers dans cette archive de manière automatique pour la mise à niveau suivante.

2.7.11 Retour d'urgence à une version précédente (downgrade)



AVERTISSEMENT
Use of **apt-pinning** technique by a novice user is sure call for major troubles. You must avoid using this technique except when you absolutely need it. See Section 2.1.11 for alternative solutions.



Attention
Le retour vers une version antérieure n'est pas officiellement géré par Debian dans sa conception. Ce ne devrait être fait qu'en tant que partie d'un processus de récupération d'urgence. Malgré cette situation, on sait que cela fonctionne bien pour de nombreux incidents. Avec les systèmes critiques vous devrez effectuer une sauvegarde des données importantes du système après l'opération de récupération et réinstaller le nouveau système depuis le départ.

Vous pouvez être assez chanceux pour revenir en arrière depuis une archive plus récente vers une archive plus ancienne afin de récupérer une mise à jour du système en manipulant la **version candidate** (consultez Section 2.7.7). Cette méthode est un remplacement de paresseux des nombreuses et fastidieuses commandes « `dpkg -i` *`paquet-cassé_ancienne-version`*`.deb` » (consultez Section 2.6.3).

Recherchez les lignes du fichier « `/etc/apt/sources.list` » permettant de suivre `unstable` ayant la forme suivante :

```
deb http://deb.debian.org/debian/ sid main contrib non-free
```

Remplacez-la avec la suivante pour suivre `testing` :

```
deb http://deb.debian.org/debian/ forky main contrib non-free
```

Configurez le fichier « `/etc/apt/preferences` » comme suit :

```
Package: *
Pin: release a=testing
Pin-Priority: 1010
```

Lancez « `apt-get update; apt-get dist-upgrade` » pour forcer l'installation à une version antérieure des paquets du système.

Supprimez ce fichier spécial « `/etc/apt/preferences` » après ce retour en arrière d'urgence.

ASTUCE
C'est une bonne idée de supprimer (sans purger) autant de paquets que possible afin de limiter les problèmes de dépendances. Vous devrez peut-être supprimer et installer manuellement un certain nombre de paquets afin de remettre le système dans un état antérieur. Le noyau de Linux, le gestionnaire d'amorçage, udev, PAM, APT, et les paquets relatifs au réseau ainsi que leurs fichiers de configuration demandent une attention particulière.

2.7.12 Paquet equivs

Si vous devez compiler un programme à partir de ses sources pour remplacer un paquet Debian, le mieux est d'en faire un paquet local réellement « debianisé » (`*.deb`) et d'utiliser une archive privée.

Si vous choisissez de compiler un programme depuis ses sources et de l'installer plutôt sous « `/usr/local` », vous pouvez avoir besoin d'utiliser `equivs` en dernier ressort pour satisfaire les dépendances des paquets manquants.

```
Package: equivs
Priority: optional
Section: admin
Description: Circumventing Debian package dependencies
 This package provides a tool to create trivial Debian packages.
 Typically these packages contain only dependency information, but they
 can also include normal installed files like other packages do.
 .
 One use for this is to create a metapackage: a package whose sole
 purpose is to declare dependencies and conflicts on other packages so
 that these will be automatically installed, upgraded, or removed.
 .
 Another use is to circumvent dependency checking: by letting dpkg
 think a particular package name and version is installed when it
 isn't, you can work around bugs in other packages' dependencies.
 (Please do still file such bugs, though.)
```

2.7.13 Porter un paquet vers le système stable



Attention
Il n'y a aucune garantie que la procédure décrite ici fonctionne sans efforts manuels supplémentaires pour faire face aux différences de systèmes.

Pour des mises à niveau partielles du système `stable`, il est souhaitable de reconstruire un paquet dans son environnement en utilisant le paquet source. Cela évite des mises à niveau massives de paquets en raison de leurs dépendances.

Ajoutez les entrées suivantes au fichier « `/etc/apt/sources.list` » d'un système `stable` :

```
deb-src http://deb.debian.org/debian unstable  main contrib non-free
```

Installez les paquets nécessaires à la compilation et téléchargez les sources comme suit :

```
# apt-get update
# apt-get dist-upgrade
# apt-get install fakeroot devscripts build-essential
# apt-get build-dep foo
$ apt-get source foo
$ cd foo*
```

Mettez à jour certains paquets de la chaîne d'outils tels que `dpkg`, et `debhelper` à partir des paquets rétroportés s'ils sont requis pour le rétroportage.

Exécutez ce qui suit :

```
$ dch -i
```

Incrémentez la version du paquet, en ajoutant, par exemple « `+bp1` » dans « `debian/changelog` »

Construisez les paquets et installez-les sur le système en faisant ce qui suit :

```
$ debuild
$ cd ..
# debi foo*.changes
```

2.7.14 Serveur mandataire (proxy) pour APT

Comme effectuer le miroir complet d'une sous-section d'une archive Debian gaspille de l'espace disque et de la bande passante du réseau, il est souhaitable, lorsque vous administrez de nombreux systèmes sur le LAN, de mettre en œuvre un serveur mandataire (« proxy ») local pour APT. APT peut être configuré pour utiliser un serveur mandataire web (http) générique comme `squid` (consultez Section 6.5) tel que décrit dans apt.conf(5) et dans « `/usr/share/doc/apt/examples/configure-index.gz` ». La variable d'environnement « `$http_proxy` » peut être utilisée pour outrepasser le serveur mandataire défini dans le fichier « `/etc/apt/apt.conf` ».

Il y a des outils de proxy spécifiques pour l'archive Debian. Vous devriez consulter le BTS avant de les installer.

paquet	popularité	taille	description
`apt-cacher`	V:0.27, I:0.32	265	proxy avec cache pour les paquets et les fichiers source Debian (programme Perl)
`apt-cacher-ng`	V:3.8, I:3.9	2063	proxy avec cache pour la distribution de paquets de logiciel (programme C++ compilé)

Table 2.19 – Liste des outils de proxy spécifiques à l'archive Debian



Attention
Lors que Debian réorganise la structure de son archive, ces outils de proxy spécialisés ont tendance à exiger que le code soit réécrit par le responsable du paquet et peuvent ne plus fonctionner pendant un certain temps. D'un autre côté, les serveurs mandataires web (http) génériques sont plus robustes et s'accommodent plus facilement de tels changements.

2.7.15 Autres lectures concernant la gestion des paquets

Vous pouvez en apprendre davantage sur la gestion des paquets dans les documentations suivantes :

- Documentations primaires sur la gestion des paquets :
  - aptitude(8), dpkg(1), tasksel(8), apt(8), apt-get(8), apt-config(8), apt-secure(8), sources.list(5), apt.conf(5) et apt_preferences(5) ;
  - « `/usr/share/doc/apt-doc/guide.html/index.html` » et « `/usr/share/doc/apt-doc/offline.html` » du paquet `apt-doc` ;
  - « `/usr/share/doc/aptitude/html/en/index.html` » du paquet `aptitude-doc-fr`.
- Documentations officielles et détaillées sur l'archive Debian :
  - « Charte Debian, chapitre 2 - L'archive Debian »,
  - « Manuel de référence du développeur Debian, chapitre 4 - Ressources pour les développeurs Debian 4.6 L'archive Debian » et
  - « FAQ de Debian GNU/Linux, chapitre 6 - Les archives FTP Debian ».
- Didacticiel pour la construction d'un paquet Debian pour les utilisateurs Debian :
  - « <Guide des responsables Debian> ».

Chapitre 3

Initialisation du système

En tant qu'administrateur du système, il est sage que vous sachiez en gros comment le système Debian est démarré et configuré. Bien que les détails exacts figurent dans les fichiers sources des paquets installés et dans leurs documentations, c'est un peu pénible pour la plupart d'entre-nous.

Voici un aperçu succinct des points clé de l'initialisation du système Debian. Comme le système Debian évolue constamment, vous devriez vous référer à la dernière documentation.

— Le manuel Debian Linux Kernel Handbook est la principale source d'informations sur le noyau Debian.
— bootup(7) décrit le processus de démarrage du système basé sur `systemd` (dernières versions de Debian).
— boot(7) décrit le processus de démarrage du système basé sur UNIX System V version 4 (anciennes versions de Debian).

3.1 Aperçu du processus d'amorçage du système

Le système informatique subit plusieurs phases de processus d'amorçage (« boot strap process ») depuis l'événement de mise sous tension jusqu'à ce qu'il offre à l'utilisateur un système d'exploitation (OS) pleinement fonctionnel.

Pour des raison de simplicité, je limiterai la discussion à une plateforme PC typique avec l'installation par défaut.

Le processus d'amorçage typique est comme une fusée à quatre étages. Chaque étage de la fusée passe le contrôle du système à l'étage suivant.

— Section 3.1.1
— Section 3.1.2
— Section 3.1.3
— Section 3.1.4

Bien entendu, elles peuvent être configurées de manière différente. Par exemple, si vous avez compilé votre propre noyau, vous pouvez sautez l'étape avec le système mini-Debian. Ne supposez donc pas que c'est le cas sur votre système avant de l'avoir vérifié vous-même.

3.1.1 Étage 1 : UEFI

L' interface micrologicielle extensible unifiée UEFI définit un gestionnaire de démarrage dans le cadre de la spécification UEFI. Lorsqu'un ordinateur est démarré, le gestionnaire de démarrage est le 1er étage du processus d'amorçage qui vérifie la configuration de démarrage et, en fonction de ses réglages, exécute alors le chargeur de démarrage ou le noyau du système d'exploitation spécifié (généralement le chargeur de démarrage). La configuration de démarrage est définie par les variables stockées dans la mémoire RAM non volatile, y compris les variables qui indiquent le chemin des systèmes de fichiers vers les chargeurs de démarrage ou les noyaux de système d'exploitation.

Une partition système EFI (ESP) est une partition de périphérique de stockage de données qui est utilisée dans les ordinateurs adhérant à la spécification UEFI. Accédée par le micrologiciel UEFI quand un ordinateur est démarré, elle stocke les applications UEFI et les fichiers dont ces applications ont besoin pour fonctionner, y compris les chargeurs de démarrage du système d'exploitation. (Sur le système patrimonial de PC, BIOS stocké dans le MBR peut être utilisé à la place.)

3.1.2 Étage 2 : le chargeur initial

Le chargeur de démarrage est le 2ème étage du processus de démarrage qui est lancé par l'UEFI. Il charge l'image du noyau du système et l'image initrd dans la mémoire et leur donne le contrôle. Cette image initrd est l'image du système de fichiers racine et sa prise en charge dépend du chargeur de démarrage utilisé.

Le système Debian utilise normalement le noyau Linux comme noyau système par défaut. L'image initrd pour le noyau Linux 5.x actuel est techniquement l'image initramfs (système de fichiers initial en RAM).

Il existe de nombreux chargeurs de démarrage et de nombreuses options de configuration disponibles.

paquet	popularité	taille	initrd	chargeur initial	description
grub-efi-amd64	I:472	142	Pris en charge	GRUB UEFI	assez intelligent pour comprendre le partitionnement du disque et des systèmes de fichiers tels que vfat, ext4, …. (UEFI)
grub-pc	V:17, I:504	479	Pris en charge	GRUB 2	assez intelligent pour comprendre le partitionnement du disque et des systèmes de fichiers tels que vfat, ext4, …. (BIOS)
grub-rescue-pc	V:0.04, I:0.55	7269	Pris en charge	GRUB 2	images de secours amorçables de GRUB 2 (CD et disquettes) (versions PC/BIOS)
grml-rescueboot	V:0.1, I:1.1	39	N/A	GRUB 2 plug-in	grml-rescueboot adds live Linux ISO images to the grub2 boot menu
syslinux	V:3, I:30	325	Pris en charge	Isolinux	il comprend le système de fichiers ISO9660. C'est utilisé pour le CD d'amorçage.
syslinux	V:3, I:30	325	Pris en charge	Syslinux	il comprend le système de fichiers MSDOS (FAT). Il est utilisé par la disquette d'amorçage.
loadlin	V:0.02, I:0.36	87	Pris en charge	Loadlin	un nouveau système est démarré depuis le système FreeDOS/MSDOS.
mbr	V:0.2, I:2.6	47	Non pris en charge	MBR par Neil Turton	c'est un logiciel libre qui se substitue au MBR de MSDOS. Il ne comprend que les partitions sur disques.

Table 3.1 – Liste des chargeurs initiaux

Pour le système UEFI, GRUB2 lit d'abord la partition ESP et utilise l'UUID spécifié pour `search.fs_uuid` dans « `/boot/efi/EFI/debian/grub.cfg` » pour déterminer la partition du fichier de configuration du menu GRUB2 « `/boot/grub/grub.cfg` ».

La partie principale du fichier de configuration du menu GRUB2 ressemble à ceci :

```
menuentry 'Debian GNU/Linux' ... {
        load_video
        insmod gzio
        insmod part_gpt
```

```
        insmod ext2
        search --no-floppy --fs-uuid --set=root fe3e1db5-6454-46d6-a14c-071208ebe4b1
        echo    'Loading Linux 5.10.0-6-amd64 ...'
        linux   /boot/vmlinuz-5.10.0-6-amd64 root=UUID=fe3e1db5-6454-46d6-a14c-071208ebe4b1 ↩
             ro quiet
        echo    'Loading initial ramdisk ...'
        initrd  /boot/initrd.img-5.10.0-6-amd64
}
```

Pour cette partie de `/boot/grub/grub.cfg`, cette entrée de menu signifie ce qui suit.

réglage	**valeur**
modules GRUB2 chargés	`gzio`, `part_gpt`, `ext2`
partition du système de fichiers racine utilisée	partition identifiée par `UUID=fe3e1db5-6454-46d6-a14c-071208ebe4b1`
chemin d'image du noyau dans le système de fichiers racine	`/boot/vmlinuz-5.10.0-6-amd64`
paramètre de démarrage du noyau utilisé	"`root=UUID=fe3e1db5-6454-46d6-a14c-071208ebe4b1 ro quiet`"
chemin de l'image initrd dans le système de fichiers racine	`/boot/initrd.img-5.10.0-6-amd64`

Table 3.2 – La signification de l'entrée de menu de la partie ci-dessus de `/boot/grub/grub.cfg`

On Debian system, "`/boot/grub/grub.cfg`" is managed by the installed GRUB package (e.g. `grub-efi-amd64`) and direct user modification to this file is deprecated. You should customize configuration files in "`/etc/grub.d/`" and "`/etc/default/grub`", instead. Then configure the GRUB configuration files and update NVRAM variables to automatically boot into Debian by:

```
# dpkg-reconfigure grub-efi-amd64
```

ASTUCE
You can show kernel boot log messages by removing "`quiet`" from the "`GRUB_CMDLINE_LINUX_DEFAULT`" value in "`/etc/default/grub`".

ASTUCE
You can add GRUB splash background by placing its image file in "`/boot/grub/`".

Consultez « `info grub` », grub-install(8) et grub-mkconfig(8).

3.1.3 Étage 3 : le système mini-Debian

Le système mini-Debian est le troisième étage du processus d'amorçage lancée par le chargeur d'amorçage. Il lance le noyau du système avec son système de fichiers racine en mémoire. C'est un étage préparatoire facultatif du processus de démarrage.

Note
Le terme « système mini-Debian » est utilisé par l'auteur pour décrire dans ce document ce 3ème étage du processus de démarrage. On désigne souvent ce système par système initrd. Un système semblable en mémoire est utilisé par l'installateur Debian.

Le script « `/init` » est exécuté en tant que premier programme sur le système de fichiers racine en mémoire. C'est un script de l'interpréteur de commandes qui initialise le noyau dans l'espace utilisateur et passe le contrôle au prochain étage. Ce système mini-Debian offre au système d'amorçage une flexibilité comme l'ajout de modules du noyau avant le processus de démarrage principal ou le montage du système de fichiers racine en mode chiffré.

— Le programme « `/init` » est un programme de script d'interpréteur si initramfs a été créé par `initramfs-tools`.
  — Vous pouvez interrompre cette partie du processus d'amorçage afin d'obtenir l'invite de l'interpréteur de l'administrateur en indiquant « `break=init` » etc. comme paramètre de démarrage du noyau. Consultez le script « `/init` » pour d'autres conditions d'interruption. Cet environnement d'interpréteur de commandes est suffisamment sophistiqué pour effectuer une bonne inspection du matériel de votre machine.
  — Les commandes disponibles avec ce système mini-Debian sont des commandes réduites et sont principalement fournies par un outil GNU appelé busybox(1).
— Le programme « `/init` » est un programme exécutable de `systemd` si initramfs a été créé par `dracut`.
  — Les commandes disponibles dans ce mini-système Debian forment un environnement systemd(1) réduit à l'essentiel.

Attention
Vous devrez utiliser l'option « `-n` » de la commande `mount` lorsque vous êtes sur le système de fichiers en lecture seule.

3.1.4 Étage 4 : le système Debian normal

The normal Debian system is the 4th stage of the boot process which is started by the mini-Debian system. The system kernel for the mini-Debian system continues to run in this environment. The root filesystem is switched from the one on the memory to the one on the physical storage device.

Le programme init est le premier à être exécuté, assorti du PID=1, afin qu'il accomplisse son rôle de processus principal de démarrage de plusieurs programmes. Le chemin par défaut du programme init est « `/usr/sbin/init` » mais il peut être modifié en passant un paramètre de démarrage au noyau, comme suit : « `init=/chemin/vers/programme_i`

« `/usr/sbin/init` » est lié symboliquement à « `/lib/systemd/systemd` » depuis Debian 8 Jessie (publiée en 2015).

ASTUCE
Il est possible de vérifier le niveau d'exécution actuel de la commande init du système avec la commande « `ps --pid 1 -f` ».

ASTUCE
Vous trouverez des conseils actualisés pour accélérer le processus de démarrage sur Debian wiki:BootProcessSpeedup.

3.2 Rescue system

AVERTISSEMENT
Do not perform system administration tasks around the boot strap process without having a rescue system.

paquet	popularité	taille	description
systemd	V:925, I:981	11401	démon init(8) basé sur des événements pour la concurrence (remplacement de sysvinit)
cloud-init	V:4.6, I:6.4	3267	Systèmes d'initialisation pour les instances d'infrastructure de nuage
systemd-sysv	V:921, I:984	77	les pages de manuel et liens nécessaires à systemd pour remplacer sysvinit
init-system-helpers	V:754, I:989	132	outils d'aide pour basculer entre sysvinit et systemd
initscripts	V:14, I:49	197	scripts pour initialiser et arrêter le système
sysvinit-core	V:3.1, I:3.5	368	Utilitaires init(8) de type System-V
sysv-rc	V:27, I:52	85	Mécanisme de changement de niveau de fonctionnement de type System-V
sysvinit-utils	V:734, I:1000	100	Utilitaires de type System-V (startpar(1), bootlogd(8), …)
lsb-base	V:193, I:281	12	Linux Standard Base 3.2 fonctionnalité de script d'initialisation
insserv	V:32, I:52	132	outil pour organiser la séquence de démarrage en utilisant les dépendances du script LSB de init.d
kexec-tools	V:1.6, I:5.3	320	outil kexec pour le redémarrage par kexec(8) (redémarrage à chaud)
systemd-bootchart	V:0.17, I:0.49	131	analyseur des performances du processus de démarrage
mingetty	V:0.1, I:2.3	36	getty(8) en mode console uniquement
mgetty	V:0.13, I:0.44	318	modem intelligent (« smart modem ») remplaçant de getty(8)

Table 3.3 – Liste d'utilitaires d'amorçage initial pour le système Debian :

Availability of a rescue system enables us to perform challenging tasks such as:

— Booting a system from a broken boot loader installation
— Fixing a broken boot loader installation
— Extracting data from a broken unbootable system
— Editing filesystems, disk partitions, and LVM volumes involving the root filesystem

Typically, a rescue system is provided as a ISO image file and written to a removable storage media such as:

— USB flash drive prepared as Section 9.7.2
— CD / DVD prepared as Section 9.7.7

For simplicity, USB flash drive cases are mentioned as examples below but CD or DVD may be used as well.

ASTUCE
You may need to change some UEFI NVRAM variables to boot arbitrary boot loaders on the removable storage media.

3.2.1 GRUB UEFI rescue system on USB

The GRUB UEFI rescue system with menu can be started by turning on system power with the "GRUB UEFI rescue system on USB" inserted.

This "GRUB UEFI rescue system on USB" is prepared by writing the Super Grub2 Disk ISO image to USB flash drive in advance.

In case of broken boot loader configuration by the installation of another operating system etc., you can fix this by:

— Start the GRUB UEFI rescue system to discover bootable installed systems automatically.

— Start the installed Debian system from the GRUB menu.
— At Linux root shell prompt:

```
# dpkg-reconfigure grub-efi-amd64
```

Note
The bootable GRUB rescue ISO image can be generated by following "`info grub-mkrescue`", too. It offers a CLI GRUB shell prompt but doesn't offer automatic discovery of bootable installed systems.

3.2.2 Linux live rescue system on USB

The Linux live rescue system can be started by turning on system power with the "Linux live rescue system on USB" inserted.

This "Linux live rescue system on USB" can be prepared by writing one of Linux live ISO images based on Debian to a USB flash drive in advance. Here are some examples of such Linux live images.

— Debian Live images
— Kali Linux Live
— Grml Live Linux

Here are some use cases of this Linux live rescue system:

— Fix the broken boot loader configuration caused by the installation of another operating system etc.:
  — Start the Linux live rescue system.
  — Mount the partition containing the root filesystem of the unbootable installed Debian system to "`/mnt`".
  — At Linux root shell prompt:

```
# chroot /mnt; dpkg-reconfigure grub-efi-amd64
```

— Fix the broken `dpkg` package:
  — Start the Linux live rescue system.
  — Mount the partition containing the root filesystem of the installed Debian system with the broken `dpkg` package to "`/mnt`".
  — At Linux root shell prompt:

```
# dpkg --root /mnt -i /mnt/var/cache/apt/archives/dpkg_old_version_amd64.deb
```

— Perform normally prohibited changes such as filesystem operations to the installed system (see Section 9.6).



AVERTISSEMENT
Your GUI screen of the Linux live system may be locked after the inactivity.

ASTUCE

— It's a good idea to set your password as soon as you start a Linux live system.
— You may set your password for example by "`sudo passwd` *`user`*" from the user's shell prompt on the Linux virtual consoles (see Section 1.1.6).
— Some Linux live systems may set their default "*user*/*password*": "Debian Live" = "user/live", "Kali Linux Live" = "kali/kali"

3.2.3 Linux live rescue system from GRUB

The Linux live rescue system can be started from the GRUB menu entry. The GRUB configuration for this is prepared by the following:

— Installez le paquet `grml-rescueboot`.
— Find a Linux live ISO image which has "`/boot/grub/loopback.cfg`" in its image.
  — ISO images mentioned in Section 3.2.2 have "`/boot/grub/loopback.cfg`" in their image.
— Copy some of these Linux live ISO images to the "`/boot/grml/`" directory.
— Update GRUB menu by:

```
# dpkg-reconfigure grub-efi-amd64
```

Upon turning on system power, the GRUB displays menu with candidates for Linux live rescue systems.

3.3 Systemd

3.3.1 Initialisation avec Systemd

Quand le système Debian démarre, `/usr/sbin/init` lié symboliquement avec `/usr/lib/systemd` est démarré comme processus de système init (`PID=1`) propriété du superutilisateur (`UID=0`). Consulter systemd(1).

Le processus d'initialisation de `systemd` génère des processus en parallèle en se basant sur les fichiers de configuration des unités (consulter systemd.unit(5)) qui sont écrits dans un style déclaratif au lieu du style procédural de type SysV.

Les processus engendrés sont placés dans des groupes de contrôle Linux individuels nommés d'après l'unité à laquelle ils appartiennent dans la hiérarchie privée de systemd (consulter cgroups et Section 4.7.5).

Les unités pour le mode système sont chargés à partir du « Chemin de recherche d'unités système » décrit dans systemd.unit(5). Les principales sont les suivantes par ordre de priorité :

— « `/etc/systemd/system/*` » : unités du système créés par l'administrateur ;
— « `/run/systemd/system/*` » : unités d'exécution ;
— « `/lib/systemd/system/*` » : unités du système installées par le gestionnaire de paquets de la distribution.

Leurs interdépendances sont spécifiées par les directives « `Wants=` », « `Requires=` », « `Before=` », « `After=` », ... (consulter « MAPPING OF UNIT PROPERTIES TO THEIR INVERSES » dans systemd.unit(5)). Les contrôles de ressources sont également définis (consulter systemd.resource-control(5)).

Le suffixe du fichier de configuration de l'unité encode leurs types comme :

— ***.service** décrit le processus contrôlé et supervisé par `systemd`. Consulter systemd.service(5) ;
— ***.device** décrit le périphérique exposé dans sysfs(5) en tant qu'arborescence de périphériques udev(7). Consulter systemd.device(5) ;
— ***.mount** décrit le point de montage du système de fichiers contrôlé et supervisé par `systemd`. Consulter systemd.mount(5) ;
— ***.automount** décrit le point de montage automatique du système de fichiers contrôlé et supervisé par `systemd`. Consulter systemd.automount(5) ;
— ***.swap** décrit le périphérique ou le fichier d'échange contrôlé et supervisé par `systemd`. Consulter systemd.swap(5) ;
— ***.path** décrit le chemin supervisé par `systemd` pour l'activation basée sur le chemin. Consulter systemd.path(5) ;
— ***.socket** décrit le socket contrôlé et supervisé par `systemd` pour l'activation basée sur un socket. Consulter systemd.socket(5) ;
— ***.timer** décrit la minuterie contrôlée et supervisée par `systemd` pour l'activation basée sur une minuterie. Consulter systemd.timer(5) ;
— ***.slice** gère les ressources avec cgroups(7). Consulter systemd.slice(5) ;

— ***.scope** est créé de manière programmatique à l'aide des interfaces de bus de `systemd` pour gérer un ensemble de processus système. Consulter systemd.scope(5) ;
— ***.target** regroupe d'autres fichiers de configuration d'unité pour créer le point de synchronisation au démarrage. Consulter systemd.target(5).

Au démarrage du système (c'est-à-dire init), le processus `systemd` tente de démarrer le « `/lib/systemd/system/defau` (normalement lié symboliquement lié à « `graphical.target` »). Tout d'abord, certaines unités cibles spéciales (consulter systemd.special(7)) telles que « `local-fs.target` », « `swap.target` » et « `cryptsetup.target` » sont extraites pour monter les systèmes de fichiers. Ensuite, d'autres unités cibles sont également extraites par les dépendances de l'unité cible. Pour plus d'informations, lisez bootup(7).

`systemd` offre des fonctionnalités de rétrocompatibilité. Les scripts de démarrage de style SysV dans « `/etc/init.d/rc[0` sont toujours analysés et telinit(8) est traduit en demandes d'activation d'unité systemd.



Attention
Les niveaux d'exécution émulés 2 à 4 sont tous liés symboliquement au même « `multi-user.target` ».

3.3.2 Connexion avec Systemd

Quand un utilisateur se connecte sur un système Debian à l'aide de gdm3(8), sshd(8), etc., `/lib/systemd/system --use` est démarré comme processus de gestionnaire de services de l'utilisateur, propriété de l'utilisateur correspondant (consulter systemd(1)).

Le gestionnaire `systemd` de services de l'utilisateur génère des processus en parallèle en se basant sur les fichiers de configuration des unités (consulter systemd.unit(5)) de style déclaratif et `user@.service`(5)).

Les unités pour le mode utilisateur sont chargées à partir du « Chemin de recherche d'unités de l'utilisateur » décrit dans systemd.unit(5). Les principales sont les suivantes par ordre de priorité :

— « `~/.config/systemd/user/*` » : unités de configuration de l'utilisateur ;
— « `/etc/systemd/user/*` » : unités de l'utilisateur créés par l'administrateur ;
— « `/run/systemd/user/*` » : unités d'exécution ;
— « `/lib/systemd/user/*` » : unités de l'utilisateur installées par le gestionnaire de paquets de la distribution.

Elles sont gérées de la même façon que dans la Section 3.3.1.

3.4 Messages du noyau

Le message d'erreur du noyau affiché sur la console peut être configuré en définissant son seuil :

```
# dmesg -n3
```

3.5 Messages du système

Sous `systemd`, les messages du noyau et du système sont consignés par le service de journal `systemd-journald.servi` (alias `journald`) soit comme données binaires persistantes dans « `/var/log/journal` » ou comme données binaires volatiles dans « `/run/log/journal/` ». Ces données binaires de journal sont accessibles avec la commande journalctl(1). Par exemple, vous pouvez afficher le journal du dernier démarrage avec :

```
$ journalctl -b
```

valeur du niveau d'erreur	nom du niveau d'erreur	signification
0	KERN_EMERG	le système est inutilisable
1	KERN_ALERT	une action doit être entreprise immédiatement
2	KERN_CRIT	conditions critiques
3	KERN_ERR	conditions d'erreur
4	KERN_WARNING	conditions d'avertissement
5	KERN_NOTICE	condition normale mais significative
6	KERN_INFO	information
7	KERN_DEBUG	messages du niveau de débogage

Table 3.4 – Liste des niveaux d'erreur du noyau

Opération	bribe de commande
Afficher le journal des services système et du noyau depuis le dernier démarrage	"`journalctl -b --system`"
Afficher le journal des services de l'utilisateur actuel depuis le dernier démarrage	"`journalctl -b --user`"
Afficher le journal des tâches de « `$unit` » depuis le dernier démarrage	"`journalctl -b -u $unit`"
Afficher le journal des tâches de « `$unit` » (style « `tail -f` ») depuis le dernier démarrage	"`journalctl -b -u $unit -f`"

Table 3.5 – Liste de bribes de commande utilisant `journalctl`

Sous `systemd`, l'utilitaire de journalisation du système rsyslogd(8) peut n'être pas installé. S'il est installé, il modifie son comportement pour lire les données binaires volatiles du journal (au lieu de la valeur par défaut avant systemd « `/dev/log` ») et pour créer des données ASCII permanentes traditionnelles de journal du système . Cela peut être personnalisé avec « `/etc/default/rsyslog` » et « `/etc/rsyslog.conf` » pour à la fois le fichier journal et l'affichage à l'écran. Voir rsyslogd(8) et rsyslog.conf(5). Voir aussi Section 9.3.2.

3.6 Gestion du système

`systemd` offre non seulement un système d'initialisation, mais aussi des opérations génériques de gestion du système avec la commande systemctl(1).

Ici, « `$unit` » dans les exemples ci-dessus peut être un nom d'unité unique (les suffixes tels que `.service` et `.cible` sont facultatifs) ou, dans de nombreux cas, des spécifications d'unité multiples (modèles génériques de type interpréteur « * », « ? », « `[]` » en utilisant fnmatch(3) qui seront comparés aux noms principaux de toutes les unités actuellement en mémoire).

Les commandes de modification d'état du système dans les exemples ci-dessus sont classiquement précédées par la commande « `sudo` » pour obtenir les droits d'administration nécessaires.

La sortie de « `systemctl status $unit|$PID|$device` » utilise des points de couleur (« ● ») pour examiner l'état de l'unité d'un seul coup d'œil.

— Un « ● » blanc indique un état « inactif » ou « désactivation ».
— Un « ● » rouge indique un état « échec » ou « erreur ».
— Un « ● » vert indique un état « actif », « rechargement » ou « activation ».

Opération	bribe de commande
Lister toutes les types d'unités disponibles	"systemctl list-units --type=help"
Lister toutes les unités cibles en mémoire	"systemctl list-units --type=target"
Lister toutes les unités de service en mémoire	"systemctl list-units --type=service"
Lister toutes les unités de périphérique en mémoire	"systemctl list-units --type=device"
Lister toutes les unités de montage en mémoire	"systemctl list-units --type=mount"
Lister toutes les unités de sockets en mémoire	"systemctl list-sockets"
Lister toutes les unités de minuterie en mémoire	"systemctl list-timers"
Démarrer « $unit »	"systemctl start $unit"
Stopper « $unit »	"systemctl stop $unit"
Recharger une configuration particulière à un service	"systemctl reload $unit"
Arrêter et démarrer tous les « $unit »	"systemctl restart $unit"
Démarrer « $unit » et arrêter tous les autres	"systemctl isolate $unit"
Basculer vers « graphical » (système GUI)	"systemctl isolate graphical"
Basculer vers « multi-user » (système CLI)	"systemctl isolate multi-user"
Basculer vers « rescue » (système CLI mono-utilisateur)	"systemctl isolate rescue"
Envoyer le signal kill à « $unit »	"systemctl kill $unit"
Vérifier si le service « $unit » est actif	"systemctl is-active $unit"
Vérifier si le service « $unit » est défaillant	"systemctl is-failed $unit"
Vérifier l'état de « $unit\|$PID|périphérique »	"systemctl status $unit\|$PID|$device"
Afficher les propriétés de « $unit\|$job »	"systemctl show $unit\|$job"
Réinitialiser « $unit » défaillant	"systemctl reset-failed $unit"
Lister les dépendances de tous les services d'unités	"systemctl list-dependencies --all"
Lister les fichiers d'unités installés sur le système	« systemctl list-unit-files »
Activer « $unit » (ajout de lien symbolique)	"systemctl enable $unit"
Désactiver « $unit » (suppression de lien symbolique)	"systemctl disable $unit"
Démasquer « $unit » (suppression de lien symbolique vers « /dev/null »)	"systemctl unmask $unit"
Masquer « $unit » (ajout de lien symbolique vers « /dev/null »)	"systemctl mask $unit"
Obtenir le réglage de la cible par défaut	"systemctl get-default"
Définir la cible par défaut à « graphical » (système à interface graphique)	"systemctl set-default graphical"
Définir la cible par défaut à « multi-user » (système en ligne de commande)	"systemctl set-default multi-user"
Afficher l'environnement de travail	"systemctl show-environment"
Définir la « variable » d'environnement de travail à « valeur »	"systemctl set-environment variable=value"
Rendre indéfini la « variable » d'environnement de travail	"systemctl unset-environment variable"
Recharger tous les démons et fichiers d'unité	"systemctl daemon-reload"
Arrêter le système	"systemctl poweroff"
Arrêter et redémarrer le système	"systemctl reboot"
Suspendre le système	"systemctl suspend"
Hiberner le système	"systemctl hibernate"

3.7 Autres moniteurs du système

Voici une liste d'autres bribes de commandes de surveillance sous `systemd`. Veuillez lire les pages de manuel pertinentes, y compris cgroups(7).

Opération	bribe de commande
Afficher la durée de toutes les étapes d'initialisation	`"systemd-analyze time"`
Lister toutes les unités selon leurs temps d'initialisation	`"systemd-analyze blame"`
Charger et détecter les erreurs dans le fichier « `$unit` »	`"systemd-analyze verify $unit"`
Afficher des informations d'état d'exécution succinctes de l'utilisateur de la session de l'appelant	`"loginctl user-status"`
Afficher des informations succinctes sur l'état d'exécution de la session de l'appelant	`"loginctl session-status"`
Suivre le processus de démarrage selon les cgroups	`"systemd-cgls"`
Suivre le processus de démarrage selon les cgroups	`"ps xawf -eo pid,user,cgroup,args"`
Suivre le processus de démarrage selon les cgroups	Lire sysfs sous `"/sys/fs/cgroup/"`

Table 3.7 – Liste d'autres bribes de commandes de surveillance sous `systemd`

3.8 Configuration du système

3.8.1 Nom de machine (« hostname »)

Le noyau gère le système **nom_machine**. L'unité centrale démarrée par `systemd-hostnamed.service` définit le nom de machine du système au démarrage au nom stocké dans « `/etc/hostname` ». Ce fichier doit contenir **uniquement** le nom de machine du système et non un nom de domaine pleinement qualifié.

Pour afficher le nom de la machine utilisée, lancez la commande `hostname` (1) sans paramètre.

3.8.2 Le système de fichiers

Les options de montage des systèmes ordinaires de fichiers de disque et en réseau sont définies dans « `/etc/fstab` ». Consulter fstab(5) et Section 9.6.7.

La configuration du système de fichiers chiffré est définie dans « `/etc/crypttab` ». Consulter crypttab(5).

La configuration du RAID logiciel avec mdadm(8) est définie dans « `/etc/mdadm/mdadm.conf` ». Consulter mdadm.conf(5)



AVERTISSEMENT
Une fois tous les systèmes de fichiers montés, les fichiers temporaires se trouvant dans « `/tmp` », « `/var/lock` » et « `/var/run` » sont effacés lors de chaque démarrage du système.

3.8.3 Initialisation de l'interface réseau

Les interfaces réseau sont ordinairement initialisées dans « `networking.service` » pour l'interface `lo` et « `NetworkMana` pour les autres interfaces sur les systèmes de bureau modernes de Debian sous `systemd`.

Voir Chapitre 5 pour savoir comment les configurer.

3.8.4 Initialisation du système d'infonuagique

L'instance du système d'infonuagique peut être lancée comme un clone des « Images officielles de Debian pour l'infonuagique » ou des images similaires. Pour une telle instance de système, les personnalités telles que le nom d'hôte, le système de fichiers, le réseautage, la régionalisation, les clés SSH, les utilisateurs et les groupes peuvent être configurées en utilisant des fonctionnalités fournies par les paquets `cloud-init` et `netplan.io` avec plusieurs sources de données telles que les fichiers placés dans l'image originelle du système et des données externes fournies lors de son amorçage. Ces paquets activent la configuration déclarative du système utilisant des données YAML.

Pour plus de détails, consulter « L'infonuagique avec Debian et ses descendants », la « documentation de Cloud-init » et la Section 5.4.

3.8.5 Exemple de personnalisation pour ajuster le service sshd

Avec l'installation par défaut, de nombreux services réseau (consulter Chapitre 6) sont démarrés comme processus démon après `network.target` au démarrage par `systemd`. Le démon « `sshd` » ne fait pas exception. Changeons donc cela pour un démarrage sur demande de « `sshd` » comme exemple de personnalisation.

Tout d'abord, désactivons l'unité de service du système installée.

```
$ sudo systemctl stop sshd.service
$ sudo systemctl mask sshd.service
```

Le système d'activation de socket à la demande des services Unix classiques passait par le superserveur `inetd` (ou `xinetd`). Sous `systemd`, l'équivalent peut être activé en ajoutant les fichiers de configuration d'unité ***.socket** et ***.service**.

`sshd.socket` pour indiquer un socket sur lequel écouté

```
[Unit]
Description=SSH Socket for Per-Connection Servers

[Socket]
ListenStream=22
Accept=yes

[Install]
WantedBy=sockets.target
```

`sshd@.service` comme fichier de service correspondant de `sshd.socket`

```
[Unit]
Description=SSH Per-Connection Server

[Service]
ExecStart=-/usr/sbin/sshd -i
StandardInput=socket
```

Puis rechargez.

```
$ sudo systemctl daemon-reload
```

3.9 Le système udev

Le système udev fournit un mécanisme de découverte et d'initialisation automatique du matériel (consulter udev(7)) depuis le noyau 2.6 de Linux. Lors de la découverte de chaque périphérique par le noyau, le système udev lance un processus utilisateur qui utilise les informations provenant du système de fichiers sysfs (consulter Section 1.2.12), charge les modules du noyau nécessaires pour sa prise en charge en utilisant le programme modprobe(8) (consulter la Section 3.10) et crée les nœuds de périphérique en conséquence.

ASTUCE
Si « `/lib/modules/`*`kernel-version`*`/modules.dep` » n'a pas été proprement créé par depmod(8) pour quelque raison, les modules peuvent ne pas être chargés par le système udev comme on le souhaiterait. Lancez « `depmod -a` » pour corriger ce problème.
Les nœuds de périphériques n'ont pas besoin d'être statiques pour les règles de montage se trouvant dans « `/etc/fstab` ». Vous pouvez utiliser UUID à la place de leur nom de périphérique tel que« `/dev/sda` » pour monter les périphériques. Consultez Section 9.6.3.

Comme le système udev est une cible quelque peu mouvante, je laisse les détails pour d'autres documentations et je ne donnerai ici qu'un minimum d'informations.



AVERTISSEMENT
N'essayez pas d'exécuter des programmes d'exécution longue tels qu'un script de sauvegarde avec `RUN` dans les règles d'udev comme mentionné dans udev(7). Veuillez créer un fichier systemd.service(5) adéquat et l'activer (consulter la Section 10.2.3.2).

3.10 Initialisation des modules du noyau

Le programme modprobe(8) nous permet de configurer, depuis un processus utilisateur, un noyau Linux en cours d'exécution en ajoutant ou en supprimant des modules du noyau. Le système udev (consultez Section 3.9) en automatise l'appel afin d'aider à l'initialisation du module du noyau.

There are non-hardware modules and special hardware driver modules as the following which need to be pre-loaded by listing them, one per line, in a file with `.conf` extension in the "`/etc/modules-load.d/`" directory (see modules-load.d(5)).

— les modules TUN/TAP fournissent un périphérique de réseau virtuel point-à-point (TUN) et un périphérique de réseau virtuel Ethernet (TAP) ;
— les modules netfilter fournissent les fonctions de pare-feu netfilter (iptables(8), Section 5.7) ;
— le module du pilote du temporisateur de chien de garde.

Les fichiers de configuration du programme modprobe(8) se trouvent dans le répertoire « `/etc/modprobes.d/` » comme c'est expliqué dans modprobe.conf(5). (Si vous souhaitez que certains modules du noyau ne soient pas chargés automatiquement, vous pouvez les mettre en liste noire dans le fichier« `/etc/modprobes.d/blacklist` ».)

Le fichier « `/lib/modules/`*`version`*`/modules.dep` » généré par le programme depmod(8) décrit les dépendances des modules utilisés par le programme modprobe(8).

Note
Si vous rencontrez des problèmes de chargement de modules lors du chargement des modules au démarrage ou avec modprobe(8), « `depmod -a` » peut résoudre ces problèmes en reconstruisant « `modules.dep` ».

Le programme modinfo(8) affiche des informations concernant les modules du noyau.

Le programme lsmod(8) formate de manière agréable le contenu de « `/proc/modules` », affichant quels sont les modules du noyau actuellement chargés.

ASTUCE
Vous pouvez identifier le matériel exact installé sur votre système. Consultez Section 9.5.3.
Vous pouvez configurer le matériel au moment du démarrage pour activer les fonctionnalités désirées de ce matériel. Consultez Section 9.5.4.
Vous pouvez probablement ajouter la prise en charge d'un périphérique particulier en recompilant le noyau. Consultez Section 9.10.

Chapitre 4

Contrôle d’authentification et d’accès

Lorsqu’une personne (ou un programme) demande l’accès au système, l’authentification confirme que l’identité est autorisée.



AVERTISSEMENT
Des erreurs de configuration de PAM peuvent vous mettre à la porte de votre propre système. Vous devez avoir un CD de secours prêt ou une partition de démarrage de remplacement. Pour restaurer, démarrez le système depuis l’un de ces moyens de secours et corrigez les choses depuis là.

4.1 Authentification normale d’UNIX

L’authentification normale d’UNIX est fournie par le module pam_unix(8) avec PAM (Pluggable Authentication Modules : « Modules attachables d’authentification ») . Il y a trois fichiers de configuration importants, dont les entrées sont séparées par des « `:` », ce sont :

fichier	autorisation	utilisateur	groupe	description
`/etc/passwd`	`-rw-r--r--`	`root`	`root`	informations des comptes utilisateurs (aseptisée)
`/etc/shadow`	`-rw-r-----`	`root`	`shadow`	informations sécurisées des comptes utilisateurs
`/etc/group`	`-rw-r--r--`	`root`	`root`	informations des groupes

Table 4.1 – 3 fichiers de configuration importants pour pam_unix(8)

« `/etc/passwd` » contient ce qui suit :

```
 ...
user1:x:1000:1000:User1 Name,,,:/home/user1:/bin/bash
user2:x:1001:1001:User2 Name,,,:/home/user2:/bin/bash
 ...
```

Comme il est expliqué dans passwd(5), les entrées de ce fichier, séparées par des « `:` », ont la signification suivante :

— nom de l’utilisateur pour la connexion ;
— entrée de spécification du mot de passe ;
— identifiant numérique de l’utilisateur ;
— identifiant numérique du groupe ;

— nom de l'utilisateur ou champ de commentaire ;
— répertoire personnel de l'utilisateur ;
— interpréteur de commandes facultatif de l'utilisateur.

La seconde entrée de « /etc/passwd » était autrefois utilisée comme entrée de mot de passe chiffré. Depuis l'introduction de « /etc/shadow », cette entrée est utilisée comme entrée de spécification du mot de passe.

contenu	signification
(vide)	compte sans mot de passe
x	le mot de passe chiffré se trouve dans « /etc/shadow »

Table 4.2 – Contenu de la seconde entrée de « /etc/passwd »

« /etc/shadow » contient ceci :

```
 ...
user1:$1$Xop0FYH9$IfxyQwBe9b8tiyIkt2P4F/:13262:0:99999:7:::
user2:$1$vXGZLVbS$ElyErNf/agUDsm1DehJMS/:13261:0:99999:7:::
 ...
```

Comme c'est expliqué dans shadow(5), les différentes entrées de ce fichier, séparées par des « : », ont les significations suivantes :

— nom de l'utilisateur pour la connexion ;
— mot de passe chiffré (le « 1 » du début indique l'utilisation d'un chiffrement MD5. Le signe « * » indique que le compte ne peut pas se connecter) ;
— Date du dernier changement de mot de passe, exprimé en nombre de jours passés depuis le premier janvier 1970.
— nombre de jours avant qu'un utilisateur ne soit autorisé à changer à nouveau son mot de passe ;
— nombre de jours avant que l'utilisateur ne soit tenu de changer son mot de passe ;
— nombre de jours avant qu'un mot de passe n'arrive à expiration et durant lesquels l'utilisateur doit être averti ;
— nombre de jours durant lesquels un mot de passe devrait encore être accepté, passé sa date d'expiration ;
— date d'expiration du compte, exprimée en nombre de jours passés depuis le premier janvier 1970 ;
— …

« /etc/group » contient ce qui suit :

```
group1:x:20:user1,user2
```

Comme il est expliqué dans group(5), les entrées de ce fichier, séparées par des « : », ont la signification suivante :

— nom du groupe ;
— mot de passe chiffré (non utilisé en pratique) ;
— identifiant numérique du groupe ;
— liste des noms d'utilisateurs séparés par des « , ».

Note
« /etc/gshadow » fournit les mêmes fonctions que « /etc/shadow » pour « /etc/group » mais n'est pas réellement utilisé.

Note
Le groupe d'appartenance réel d'un utilisateur peut être ajouté dynamiquement si la ligne « auth optional pam_group.so » est ajoutée à « /etc/pam.d/common-auth » et défini dans « /etc/security/group.conf ». Consultez pam_group(8).

Note
Le paquet base-passwd contient une liste faisant autorité d'utilisateurs et de groupes : « /usr/share/doc/base-passwd/users-and-groups.html ».

4.2 Gestion des informations des comptes et des mots de passes

Voici quelques commandes importantes pour gérer les informations des comptes :

commande	fonction
`getent passwd` *`user_name`*	consulter les informations du compte « *`nom_utilisateur`* »
`getent shadow` *`user_name`*	consulter les informations cachées du compte « *`nom_utilisateur`* »
`getent group` *`group_name`*	consulter les informations du groupe « *`nom_ groupe`* »
`passwd`	gérer le mot de passe de ce compte
`passwd -e`	définir un mot de passe à usage unique pour l'activation du compte
`chage`	gérer les informations de durée de validité du mot de passe

Table 4.3 – Liste des commandes servant à gérer les informations des comptes

Vous pouvez avoir besoin des droits de l'administrateur pour certaines fonctions. Consultez crypt(3) pour le chiffrement des mots de passe et des données.

Note
Sur les systèmes configurés avec PAM et NSS comme la machine salsa de Debian, le contenu des fichiers locaux « `/etc/passwd` », « `/etc/group` » et « `/etc/shadow` » peut ne pas être utilisé de manière active par le système. Le commandes ci-dessus restent valables même sous un tel environnement.

4.3 Mot de passe de qualité

Lors de la création d'un mot de passe à l'installation de votre système ou avec la commande passwd(1), il vous faudra choisir un bon mot de passe composé d'au moins 6 à 8 caractères, comprenant au moins un des caractères appartenant à l'ensemble suivant conformément à passwd(1) :

— caractères alphabétiques en minuscules ;
— chiffres de 0 à 9 ;
— marques de ponctuation.



AVERTISSEMENT
Ne choisissez pas des mots qui se devinent aisément pour le mot de passe. Les noms de compte, les numéros de sécurité sociale, les numéros de téléphone, les adresses, les dates anniversaire, les noms des membres de votre famille ou de vos animaux domestiques, les mots du dictionnaire, les suites simples de caractères telles que « 12345 » ou « qwerty »… constituent tous de mauvais choix pour le mot de passe.

4.4 Créer un mot de passe chiffré

Il existe des outils autonomes permettant de créer des mots de passe chiffrés à partir d'une « semence ».

paquet	popularité	taille	commande	fonction
whois	V:25, I:199	384	mkpasswd	frontal de la bibliothèque crypt(3) avec fonctionnalités surabondantes
openssl	V:853, I:996	2564	openssl passwd	calculer le hachage du mot de passe (OpenSSL). passwd(1ssl)

Table 4.4 – Liste d'outils permettant de générer des mots de passe

4.5 PAM et NSS

De nombreux systèmes modernes semblable à UNIX, comme le système Debian, fournissent les mécanismes PAM (Pluggable Authentication Modules) et NSS (Name Service Switch) pour la configuration du système par l'administrateur local. Leur rôle peut être résumé de la manière suivante :

— PAM offre un mécanisme d'authentification souple qui est utilisé par les logiciels applicatifs lorsqu'ils ont besoins d'échanger des mots de passe.

— NSS fournit un mécanisme souple de service de noms qui est fréquemment utilisé par la bibliothèque standard C pour obtenir le nom de groupe de programmes comme ls(1) et id(1).

Ces systèmes PAM et NSS doivent être configurés de manière cohérente.

Les paquets importants des systèmes PAM et NSS sont les suivants :

paquet	popularité	taille	description
libpam-modules	V:945, I:1000	892	Modules attachables d'authentification (PAM) (service de base)
libpam-ldapd	V:8, I:13	82	Modules attachables d'authentification (PAM) permettant l'utilisation d'interfaces LDAP
libpam-systemd	V:788, I:972	849	Modules attachables d'authentification (PAM) pour enregistrer des sessions d'utilisateur pour logind
libpam-doc	I:7.7	1492	Modules attachables d'authentification (PAM) (documentation en HTML et texte)
libc6	V:924, I:1000	5483	bibliothèque GNU C Library : bibliothèques partagées qui fournissent aussi le service « Name Service Switch »
glibc-doc	I:4.9	3927	bibliothèque GNU C : pages de manuel
glibc-doc-reference	I:3.0	14764	bibliothèque GNU C : manuel de référence dans les formats info, pdf et html (non libre)
libnss-mdns	V:314, I:505	142	module NSS pour la résolution des noms DNS Multicast
libnss-ldapd	V:9, I:16	138	module NSS pour l'utilisation de LDAP comme service de nommage

Table 4.5 – Liste des paquets importants des systèmes PAM et NNS

— « The Linux-PAM System Administrators' Guide » de libpam-doc est essentiel à l'apprentissage de la configuration de PAM.

— La section « System Databases and Name Service Switch » de glibc-doc-reference est essentielle pour l'apprentissage de la configuration de NSS.

Note
Vous en trouverez une liste plus complète et actuelle avec la commande « aptitude search 'libpam-|libnss-' ». L'acronyme NSS peut aussi signifier « Network Security Service » qui est différent de « Name Service Switch ».

Note
PAM est la manière la plus élémentaire d'initialiser des variables d'environnement pour tous les programmes avec des valeurs par défaut valables pour l'ensemble du système.

Sous systemd, le paquet `libpam-systemd` est installé pour gérer les connexions utilisateur en enregistrant les sessions utilisateur dans la hiérarchie de groupes de contrôle de `systemd` pour logind. Voir systemd-logind(8), logind.conf(5) et pam_systemd(8).

4.5.1 Fichiers de configuration auxquels accèdent PAM et NSS

Voici quelques fichiers de configuration importants auxquels PAM et NSS peuvent accéder :

fichier de configuration	fonction
`/etc/pam.d/`*`program_name`*	définir la configuration de PAM pour le « *`nom_programme`* », consultez pam(7) et pam.d(5)
`/etc/nsswitch.conf`	définir la configuration de NSS avec une entrée pour chaque service. Consultez nsswitch.conf(5)
`/etc/nologin`	limiter la connexion de l'utilisateur à l'aide du module pam_nologin(8)
`/etc/securetty`	limiter l'accès de l'administrateur à certains tty à l'aide du module pam_securetty(8)
`/etc/security/access.conf`	limiter les accès à l'aide du module pam_access(8)
`/etc/security/group.conf`	définir les limitations en fonction du groupe à l'aide du module pam_group(8)
`/etc/security/pam_env.conf`	définir des variables d'environnement à l'aide du module pam_env(8)
`/etc/environment`	définir des variables d'environnement supplémentaires à l'aide du module pam_env(8) avec le paramètre « `readenv=1` »
`/etc/default/locale`	définir les paramètres linguistiques (« locale ») à l'aide du module pam_env(8) avec le paramètre « `readenv=1 envfile=/etc/default/locale` » (Debian)
`/etc/security/limits.conf`	définir les limitations de ressources (ulimit, core, …) à l'aide du module pam_limits(8)
`/etc/security/time.conf`	définir les restrictions de temps à l'aide du module pam_time(8)
`/etc/systemd/logind.conf`	définir la configuration du gestionnaire de login de `systemd` (voir logind.conf(5) et systemd-logind.service(8))

Table 4.6 – Liste des fichiers de configuration auxquels PAM et NSS accèdent

Les limitations dans la sélection des mots de passe est implémentée par les modules PAM pam_unix(8) et pam_cracklib(8). Ils peuvent être configurés à l'aide de leurs paramètres.

ASTUCE
Les noms de fichiers des modules PAM ont le suffixe « `.so` ».

4.5.2 Le système de gestion centralisée moderne

La gestion centralisée du système peut être mise en œuvre en utilisant le serveur centralisé LDAP Protocole léger d'accès aux répertoires (« Lightweight Directory Access Protocol ») pour administrer de nombreux systèmes semblables à UNIX ou autres sur le réseau. Le logiciel OpenLDAP est l'implémentation à sources ouvertes du protocole LDAP.

Sur un système Debian, le serveur LDAP fournit les informations de compte en utilisant PAM et de NSS avec les paquets `libpam-ldapd` et `libnss-ldapd`. Un certain nombre d'actions sont nécessaires pour l'activer (je n'ai pas utilisé cette configuration et ce qui suit est une information secondaire, veuillez la lire dans ce contexte) :

— définissez un serveur LDAP centralisé en faisant tourner un programme tel que le démon LDAP, slapd(8) ;
— modifiez les fichiers de configuration de PAM dans le répertoire « `/etc/pam.d/` » pour utiliser « `pam_ldap.so` » plutôt que le module par défaut « `pam_unix.so` » ;
— modifiez la configuration de NSS dans le fichier « `/etc/nsswitch.conf` » pour utiliser « `ldap` » plutôt que ce qui s'y trouve par défaut (« `compat` » ou « `file` ») ;
— vous devez configurer `libpam-ldapd` de manière à ce qu'il utilise une connexion SSL (ou TLS) pour la sécurité du mot de passe ;
— vous pouvez configurer `libnss-ldapd` de manière à ce qu'il utilise une connexion SSL (ou TLS) afin d'assurer l'intégrité des données au prix d'une surcharge du réseau LDAP ;
— Afin de réduire le trafic réseau de LDAP, vous devrez faire tourner nscd(8) localement pour mettre en cache les résultats de recherche de LDAP .

See documentations in nsswitch.conf(5), pam.conf(5), ldap.conf(5), and "`/usr/share/doc/libpam-doc/html/`" offered by the `libpam-doc` package and "`info libc 'Name Service Switch'`" offered by the `glibc-doc` package.

De manière similaire, vous pouvez mettre en œuvre des systèmes centralisés de remplacement avec d'autres méthodes.

— Intégration d'utilisateur et de groupe au système Windows.
  — Accès aux services de Domaine (Microsoft) avec les paquets `winbind` et `libpam_winbind`.
  — Consultez winbindd(8) et Intégration de réseaux MS Windows avec Samba.
— Intégration d'utilisateur et de groupe à l'ancien système de type UNIX.
  — Accès à NIS (appelé initialement YP) ou NIS+ avec le paquet `nis`.
  — Consultez Le Linux NIS(YP)/NYS/NIS+ HOWTO.

4.5.3 « Pourquoi la commande su de GNU ne gère-t-elle pas le groupe wheel »

C'est la célèbre phrase de Richard M. Stallman en bas de l'ancienne page `info su`. Ne pas s'inquiéter : la commande `su` actuelle de https://www.debian.org utilise PAM, on peut donc restreindre l'accès de `su` au groupe `root` en activant la ligne de « `/etc/pam.d/su` »comportant « `pam_wheel.so` ».

4.5.4 Règle de mots de passe plus stricte

L'installation du paquet `libpam-pwquality` permet d'obliger à suivre une règle plus sévère pour les mots de passe.

Sur un système GNOME typique qui installe automatiquement `libpam-gnome-keyring`, « `/etc/pam.d/common-passw` ressemble à :

```
# here are the per-package modules (the "Primary" block)
password  requisite     pam_pwquality.so retry=3 minlen=8 difok=3
password  [success=1 default=ignore]  pam_unix.so obscure use_authtok try_first_pass  ↩
    yescrypt
# here's the fallback if no module succeeds
password  requisite     pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password  required      pam_permit.so
# and here are more per-package modules (the "Additional" block)
password  optional  pam_gnome_keyring.so
# end of pam-auth-update config
```

4.6 Sécurité de l'authentification

Note
Les informations données ici pourraient ne répondre **que partiellement** à vos besoins en matière de sécurité mais elles devraient néanmoins constituer **un bon point de départ**.

4.6.1 Mot de passe sûr avec Internet

La couche de transport de nombreux services populaires communique les messages, y compris les mots de passe d'authentification, en texte clair. C'est une très mauvaise idée de transmettre un mot de passe en texte clair dans la jungle d'Internet où il peut être intercepté. Vous pouvez faire tourner des services sur une couche de transport sécurisée « Sécurité de la couche de transport (« Transport Layer Security ») » (TLS) ou son prédécesseur « Secure Sockets Layer » (SSL) pour sécuriser par chiffrement la communication dans son ensemble, y compris le mot de passe.

nom de service non sûr	port	nom de service sûr	port
`www (http)`	80	`https`	443
`smtp (mail)`	25	`ssmtp (smtps)`	465
`ftp-data`	20	`ftps-data`	989
`ftp`	21	`ftps`	990
`telnet`	23	`telnets`	992
`imap2`	143	`imaps`	993
`pop3`	110	`pop3s`	995
`ldap`	389	`ldaps`	636

Table 4.7 – Liste des services et ports sûrs et non sûrs

Le chiffrement coûte du temps processeur. Comme solution de remplacement pour économiser du temps processeur, vous pouvez continuer à effectuer les communications en texte clair tout en ne sécurisant que le mot de passe avec un protocole d'authentification sécurisé comme « Authenticated Post Office Protocol » (APOP) pour POP et « Challenge-Response Authentication Mechanism MD5 » (CRAM-MD5) pour SMTP et IMAP. (Depuis peu, pour envoyer des messages de courrier électronique au travers d'internet à votre serveur de courrier depuis votre client de courrier, il est devenu habituel d'utiliser le port 587 en remplacement du port SMTP 25 habituel pour soumettre le courrier afin d'éviter le blocage du port 25 par le fournisseur d'accès au réseau tout en vous authentifiant avec CRAM-MD5.)

The traditional password-based authentication still suffers inherent security issues such as password reuse across multiple websites. See other authentication methods for the secure solution: Time-based one-time password (TOTP), Universal 2nd Factor (U2F), Web Authentication (WebAuthn, commonly referred as "passkey")

4.6.2 Le shell sûr (Secure Shell)

Le programme SSH, Shell sûr (« Secure Shell ») permet une communication chiffrée sûre entre deux machines qui ne sont pas « de confiance » au travers d'un réseau non sûr avec une authentification sûre. Il est constitué du client OpenSSH, de ssh(1), et du démon OpenSSH, `sshd` (8). SSH peut être utilisé pour « tunneler » de manière sécurisée un protocole de communications non sûr tel que POP et X au travers d'Internet à l'aide de la fonctionnalité de transfert de port.

Le client essaie de s'authentifier en utilisant l'authentification basée sur l'hôte, une clé publique d'authentification, une authentification par question-réponse ou une authentification par mot de passe. L'utilisation d'une authentification par clé publique permet la connexion à distance sans mot de passe. Consultez Section 6.3.

4.6.3 Mesures de sécurité supplémentaires pour Internet

Même lorsque vous exécutez des services sécurisés tels que des serveurs Secure Shell (SSH) ou PPTP (protocole de tunnel point-à-point), il existe toujours des risques d'intrusion à l'aide d'une attaque par force brute pour deviner le mot de passe, etc., à partir d'Internet. L'utilisation d'une politique de pare-feu (voir Section 5.7) avec les outils de sécurité suivants peut améliorer la situation en matière de sécurité.

paquet	popularité	taille	description
knockd	V:0.7, I:1.5	110	petit démon de « tocage à la porte » knockd(1) et client knock(1)
fail2ban	V:101, I:112	2275	bannir les IP qui provoquent des erreurs d'authentification multiples

Table 4.8 – Liste des outils fournissant des mesures de sécurité supplémentaires

4.6.4 sécuriser le mot de passe de l'administrateur

Afin d'éviter que des personnes accèdent à votre machine avec les privilèges de l'administrateur, vous devez prendre les mesures suivantes :

— Prevent physical access to the system storage device (HDD / SSD / ...)
— verrouiller l'UEFI/BIOS et empêcher le démarrage à partir du support amovible
— définir un mot de passe pour la session interactive de GRUB ;
— verrouiller l'édition du menu de GRUB.

With physical access to the system storage device, resetting the password is relatively easy with following steps.

1. Move the system storage device to a PC with USB bootable UEFI/BIOS.
2. Boot system with a rescue media (see Section 3.2.2).
3. Monter la partition racine avec les droits en lecture et écriture.
4. Éditer « `/etc/passwd` » de la partition racine et vider la seconde entrée du compte `root`.

Si vous avez l'accès en édition au menu GRUB (consultez Section 3.1.2), avec `grub-rescue-pc`, il est encore plus simple d'effectuer les étapes suivantes au moment du démarrage :

1. Démarrer le système avec les paramètres du noyau modifiés en quelque chose qui ressemble à « `root=/dev/sda6 rw init=/bin/sh` ».
2. Éditer « `/etc/passwd` » et vider la seconde entrée du compte `root`.
3. Redémarrer le système.

L'interpréteur de commandes de l'administrateur est maintenant accessible sans mot de passe.

Note
Une fois que quelqu'un a accès à l'interpréteur de commandes de l'administrateur, il peut accéder à l'ensemble du système et en réinitialiser tous les mots de passe. De plus, il peut compromettre le mot de passe de n'importe quel utilisateur en utilisant des outils de cassage de mots de passe par force brute tels que les paquets `john` et `crack` (consultez Section 9.5.10). Ces mots de passes cassés peuvent permettre de compromettre d'autres systèmes.

La seule solution logicielle raisonnable pour éviter tout ça est d'utiliser une partition racine (ou une partition « `/etc` ») chiffrée par logiciel en utilisant dm-crypt et initramfs (consultez Section 9.9). Vous aurez alors toujours besoin d'un mot de passe pour vous connecter au système.

4.7 Autres contrôles d'accès

Il existe des contrôles d'accès au système autres que l'authentification par mot de passe et les autorisations de fichier.

Note
Consultez Section 9.4.16 pour restreindre la fonctionnalité de touche d'appel sécurisée (SAK) (« secure attention key ») du noyau.

4.7.1 Listes de contrôle d'accès (ACL)

Les ACL sont un sur-ensemble des permissions normales comme expliqué dans la Section 1.2.3.

Les ACL sont utilisées dans les environnements de bureau modernes. Quand un périphérique USB formaté de stockage est auto-monté, comme par exemple « `/media/penguin/SBSTICK` », un utilisateur `penguin` normal peut exécuter :

```
 $ cd /media/penguin
 $ ls -la
total 16
drwxr-x---+ 1 root    root    16 Jan 17 22:55 .
drwxr-xr-x  1 root    root    28 Sep 17 19:03 ..
drwxr-xr-x  1 penguin penguin 18 Jan  6 07:05 USBSTICK
```

« + » dans la 11ème colonne indique que les ACL sont actives. Sans ACL, un utilisateur `penguin` normal ne pourrait réaliser une telle liste puisque `penguin` n'est pas dans le groupe `root`. Les ACL peuvent être vues comme :

```
 $ getfacl .
# file: .
# owner: root
# group: root
user::rwx
user:penguin:r-x
group::---
mask::r-x
other::---
```

Ici :

— "`user::rwx`", "`group::---`", and "`other::---`" correspond au propriétaire normal, au groupe et aux autres permissions ;
— l'ACL « `user:penguin:r-x` » permet à un utilisateur normal `penguin` d'avoir les permissions « `r-x` ». Cela permet à « `ls -la` » de lister le contenu d'un répertoire ;
— l'ACL « `mask::r-x` » définit les permissions maximales.

Consultez « POSIX Access Control Lists on Linux », acl(5), getfacl(1) et setfacl(1) pour plus de renseignements.

4.7.2 sudo

sudo(8) est un programme conçu pour permettre à un administrateur système de donner des privilèges d'administration limités aux utilisateurs et d'enregistrer dans un journal les actions de l'administrateur (« root ». `sudo` ne demande que le mot de passe d'un utilisateur normal. Installez le paquet `sudo` et activez-le en définissant les options dans « `/etc/sudoers` ». Consultez l'exemple de configuration dans « `/usr/share/doc/sudo/examples/sudoers` » et Section 1.1.12.

Mon utilisation de `sudo` sur un système avec un seul utilisateur (consultez Section 1.1.12) est destinée à me protéger moi-même contre ma propre stupidité. Personnellement, je considère que l'utilisation de `sudo` est une meilleure alternative que l'utilisation permanente du système depuis le compte de l'administrateur. Par exemple, les modifications suivantes du propriétaire de « *`un_fichier`* » par « *`mon_nom`* » :

```
$ sudo chown my_name some_file
```

Bien sûr, si vous connaissez le mot de passe de root (comme beaucoup d'utilisateurs de Debian qui ont installé eux-mêmes leur système), n'importe quelle commande peut être lancée en tant qu'administrateur depuis un compte utilisateur par « `su -c` ».

4.7.3 PolicyKit

PolicyKit est un composant du système d'exploitation permettant de contrôler les droits globaux sur les systèmes de type UNIX.

Les applications graphiques les plus récentes ne sont pas conçues pour fonctionner comme des processus privilégiés. Elles échangent avec les processus privilégiés par l'intermédiaire de PolicyKit pour réaliser les opérations d'administration.

PolicyKit limite de telles opérations aux comptes d'utilisateurs appartenant au groupe `sudo` sur le système Debian.

Consultez polkit(8).

4.7.4 Restreindre l'accès à certains services du serveur

Pour la sécurité du système, il est préférable de désactiver autant de programmes de serveurs que possible. Cela devient critique pour les services par l'intermédiaire du réseau. Avoir des services réseau inutilisés, qu'ils soient activés directement en tant que démon ou par l'intermédiaire du programme super-serveur, est considéré comme un risque de sécurité.

De nombreux programmes, tels que sshd(8), utilisent un contrôle d'accès basé sur PAM. Il y a de nombreuses manières de restreindre l'accès à certains serveurs de services :

— fichiers de configuration : « `/etc/default/`*`nom_programme`* » ;
— configuration d'unité de service systemd pour un démon« ;
— PAM (Modules d'authentification attachables) (« Pluggable Authentication Modules ») ;
— « `/etc/inetd.conf` » pour le super-serveur ;
— « `/etc/hosts.deny` » et « `/etc/hosts.allow` » pour l'enrobeur TCP, tcpd(8) ;
— « `/etc/rpc.conf` » pour Sun RPC ;
— « `/etc/at.allow` » et « `/etc/at.deny` » pour atd(8) ;
— « `/etc/cron.allow` » et « `/etc/cron.deny` » pour crontab(1) ;
— un pare-feu réseau de l'infrastructure netfilter.

Consulter Section 3.6, Section 4.5.1 et Section 5.7.

ASTUCE
Les services Sun RPC doivent être actif pour NFS et les autres programmes basés sur RPC.

ASTUCE
Si vous avez des problèmes pour les accès à distance sur un système Debian récent, commentez la ligne de configuration posant problème, comme « `ALL: PARANOID` » de « `/etc/hosts.deny` » si elle existe. (Mais vous devrez faire attention au risque de sécurité induit par ce type d'action)

4.7.5 Caractéristiques de sécurité de Linux

Le noyau Linux a évolué et prend en charge des fonctionnalités de sécurité introuvables dans les implémentations UNIX traditionnelles.

Linux prend en charge les attributs étendus qui accroissent les attributs UNIX traditionnels (Consulter xattr(7)).

Linux divise les privilèges traditionnellement associés au superutilisateur en unités distinctes, appelées `capacités`(7), qui peuvent être activées et désactivées indépendamment. Les capacités sont un attribut par fil d'exécution depuis la version 2.2 du noyau.

Le cadriciel Linux Security Module (LSM) fournit un mécanisme pour que divers contrôles de sécurité puissent être imbriquées dans de nouvelles extensions de noyau. Par exemple :

— AppArmor
— Security-Enhanced Linux (SELinux) ;
— Smack (Simplified Mandatory Access Control Kernel) ;
— Tomoyo Linux

Étant donné que ces extensions peuvent restreindre davantage le modèle de privilège que les politiques ordinaires de modèle de sécurité de type Unix, même le pouvoir du superutilisateur peut être restreint. Il est conseillé de lire le document sur le cadriciel Linux Security Module (LSM) sur kernel.org.

Les espaces de noms de Linux enveloppent une ressource système globale dans une abstraction qui donne l'impression aux processus de l'espace de noms qu'ils ont leur propre instance isolée de la ressource globale. Les modifications apportées à la ressource globale sont visibles pour les autres processus membres de l'espace de noms, mais invisibles pour les autres processus. Depuis la version 5.6 du noyau, il existe 8 types d'espaces de noms (Consulter namespaces(7), unshare(1), nsenter(1)).

À partir de Debian 11 Bullseye (2021), Debian utilise une hiérarchie de cgroups unifiée (c'est-à-dire cgroups-v2).

Voici des exemples d'utilisation des espaces de noms avec cgroups pour isoler leurs processus et permettre le contrôle des ressources :

— Systemd (consulter Section 3.3.1) ;
— environnement de bac sable (consulter Section 7.7) ;
— Conteneurs Linux tels que Docker, LXC (consulter Section 9.11).

Ces fonctionnalités ne peuvent pas être réalisées par la Section 4.1. Ces sujets avancés sont pour la plupart hors du périmètre de ce document d'introduction.

Chapitre 5

Configuration du réseau

ASTUCE
Pour un guide actualisé de la gestion réseau sous Debian, lire le Guide de l'administrateur Debian - configurer le réseau (« The Debian Administrator's Handbook — Configuring the Network »).

ASTUCE
Sous systemd, networkd peut être utilisé pour gérer les réseaux. Consultez systemd-networkd(8).

5.1 L'infrastructure de base du réseau

Passons en revue l'infrastructure de base du réseau sur un système Debian moderne.

5.1.1 Résolution du nom d'hôte

La résolution du nom d'hôte est actuellement prise en charge aussi par le mécanisme NSS (Name Service Switch). Le flux de cette résolution est le suivant :

1. Le fichier « `/etc/nsswitch.conf` » avec une entrée comme « `hosts: files dns` » donne l'ordre de la résolution du nom d'hôte (cela remplace l'ancienne fonctionnalité de l'entrée « `order` » dans « `/etc/host.conf` »).
2. La méthode `files` est d'abord appelée. Si le nom d'hôte est trouvé dans le fichier « `/etc/hosts` », elle retourne toutes les adresses valables qui y correspondent et quitte. (Le fichier « `/etc/host.conf` » contient « `multi on` ».)
3. La méthode `dns` est appelée. Si le nom d'hôte est trouvé par une requête au Système de noms de domaine Internet (DNS) (« Internet Domain Name System ») identifié par le fichier « `/etc/resolv.conf` », elle retourne toutes les adresses valables correspondantes et quitte.

Une station de travail peut être installée avec comme nom d'hôte défini par exemple à « *host_name* » et comme nom de domaine facultatif défini à une chaine vide. Alors, « `/etc/hosts` ressemble à ceci :

```
127.0.0.1 localhost
127.0.1.1 host_name

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

paquets	popularité	taille	type	description
network-manager	V:429, I:472	7366	config::NM	NetworkManager (démon) : gère automatiquement le réseau
network-manager-gnome	V:33, I:136	19	config::NM	NetworkManager (frontal de GNOME)
netplan.io	V:2.4, I:8.7	340	config::NM+networkd	Netplan (generator) : interface unifiée déclarative vers les backends NetworkManager et systemd-networkd
ifupdown	V:686, I:973	204	config::ifupdown	outil standard pour activer ou désactiver le réseau (spécifique à Debian)
pppoeconf	V:0.2, I:3.6	176	config::helper	assistant de configuration d'une connexion PPPoE
wpasupplicant	V:431, I:520	4337	config::helper	client prenant en charge WPA et WPA2 (IEEE 802.11i)
wpagui	V:0.2, I:1.2	780	config::helper	client graphique (Qt) pour wpa_supplicant
wireless-tools	V:196, I:263	232	config::helper	outils pour manipuler les « Extensions Linux sans fil » (Linux Wireless Extensions)
iw	V:42, I:478	332	config::helper	outil de configuration des périphériques sans fil de Linux
iproute2	V:802, I:988	4279	config::iproute2	iproute2, IPv6 et autres configurations avancées du réseau : ip(8), tc(8), etc
iptables	V:402, I:635	2408	config::Netfilter	outils d'administration pour le filtrage des paquets et NAT (Netfilter)
nftables	V:301, I:889	189	config::Netfilter	outils d'administration pour le filtrage des paquets et NAT (Netfilter) (successeur à {ip,ip6,arp,eb}tables)
iputils-ping	V:222, I:998	188	test	tester l'accessibilité d'une machine distante par nom de machine ou adresse IP (iproute2)
iputils-arping	V:2, I:17	53	test	tester l'accessibilité réseau d'une machine distante spécifiée par une adresse ARP
iputils-tracepath	V:2, I:19	50	test	tracer le chemin du réseau vers une machine distante
ethtool	V:103, I:252	1106	test	afficher ou modifier les paramètres d'un périphérique Ethernet
mtr-tiny	V:5, I:38	182	test::low-level	tracer le chemin réseau vers une machine distante (curses)
mtr	V:4, I:39	231	test::low-level	tracer le chemin réseau vers une machine distante (curses et GTK)
gnome-nettool	V:0.5, I:7.9	2480	test::low-level	outils pour des opérations d'informations habituelles sur le réseau (GNOME)
nmap	V:27, I:180	4946	test::low-level	cartographie réseau / balayage de ports (Nmap, console)
tcpdump	V:24, I:181	1418	test::low-level	analyseur de trafic réseau (Tcpdump, console)
wireshark	V:4, I:38	17442	test::low-level	analyseur de trafic réseau (Wireshark, GTK)
tshark	V:3, I:23	615	test::low-level	analyseur de trafic réseau (console)
tcptrace	V:0.2, I:1.5	407	test::low-level	produit un résumé des connexions à partir d'une sortie de `tcpdump`
dnsutils	V:4, I:137	23	test::low-level	clients réseau fournis par BIND : nslookup(1), nsupdate(1), dig(1)
dlint	V:0.1, I:1.9	52	test::low-level	vérifier les zones d'information DNS en utilisant des requêtes du serveur de noms
dnstracer	V:0.1, I:1.0	63	test::low-level	tracer une chaîne de serveurs DNS jusqu'à la source

Table 5.1 – Liste des outils de configuration du réseau

Chaque ligne commence par une adresse IP et est suivie du nom d'hôte associé.

L'adresse IP `127.0.1.1` en deuxième ligne de cet exemple pourrait ne pas être présente sur d'autres systèmes de type UNIX. L'installateur Debian ajoute cette entrée pour les systèmes sans adresse IP permanente en tant que contournement pour certains programmes (par exemple GNOME) comme expliqué dans le bogue nº 719621.

Le *nom_hote* correspond au nom d'hôte défini dans « `/etc/hostname` » (Consulter Section 3.8.1).

Pour un système avec une adresse IP permanente, cette adresse IP devrait être utilisée à la place de `127.0.1.1`.

Pour un système avec une adresse IP permanente et un nom de domaine complètement qualifié (FQDN) fourni par le système de noms de domaine (DNS), les *nom_hote* et *nom_domaine* canoniques devraient être utilisés ici, plutôt que le simple *nom_hote*.

« `/etc/resolv.conf` » est un fichier statique si le paquet `resolvconf` n'est pas installé. S'il est installé, c'est un lien symbolique. Dans tous les cas, il contient des informations qui initialisent les routines du résolveur. Si le DNS est trouvé à l'IP=« `192.168.11.1` », il contient ce qui suit :

```
nameserver 192.168.11.1
```

Le paquet `resolvconf` fait de ce « `/etc/resolv.conf` » un lien symbolique et gère son contenu automatiquement par le script hook.

Pour une station de travail compatible PC sur un réseau local ad hoc typique, le nom d'hôte peut être résolu à l'aide du Multicast DNS (mDNS en plus des méthodes basiques par `files` et `dns`.

— Sur les systèmes Debian, Avahi fournit un cadre pour le « Multicast DNS Service Discovery ».
— Il est l'équivalent de Apple Bonjour/Apple Rendezvous.
— Le greffon de la bibliothèque `libnss-mdns` fournit une résolution de nom d'hôte à l'aide de mDNS pour la fonction « GNU Name Service Switch (NSS) » de « GNU C Library (glibc) ».
— Le fichier « `/etc/nsswitch.conf` » devrait contenir une section telle que « `hosts: files mdns4_minimal [NOTFOUND=return] dns` » (consulter `usr/share/doc/libnss-mdns/README.Debian` pour d'autres configurations).
— Un nom d'hôte suffixé avec un pseudo domaine de premier niveau « .local » est résolu en envoyant un message de requête mDNS dans un paquet UDP multicast en utilisant une adresse IPv4 « `224.0.0.251` » ou une adresse IPv6 « `FF02::FB` ».

Note
L'expansion de nom de domaine de premier niveau générique « generic Top-Level Domains (gTLD) » dans le système de noms de domaine est en cours de réalisation. Soyez attentifs aux conflits de noms lors du choix d'un nom de domaine utilisé uniquement au sein d'un réseau local « LAN ».

Note
L'utilisation de paquets tels que `libnss-resolve` avec `systemd-resolved`, ou `libnss-myhostname`, ou `libnss-mymachine`, avec les listes correspondantes sur la ligne « `hosts` » dans le fichier « `/etc/nsswitch.conf` » peut outrepasser la configuration traditionnelle du réseau expliquée ci-dessus (consulter nss-resolve(8), systemd-resolved(8), nss-myhostname(8) et nss-mymachines(8) pour plus de détails).

5.1.2 Nom de l'interface réseau

systemd utilise "Noms prévisibles d'interface réseau" comme "`enp0s25`".

5.1.3 Plage d'adresses réseau du réseau local (« LAN »)

Un rappel des plages d'adresses IPv4 32 bits de chacune des classes réservées à l'utilisation sur un réseau local (LAN) par la rfc1918. Ces adresses garantissent qu'aucun conflit ne sera créé avec aucune des adresses présentes sur Internet proprement dit.

Note
Les adresses IP écrites avec des deux-points sont des adresses IPv6, par exemple, « `::1` » pour `localhost`.

Classe	adresses de réseau	masque de réseau	masque de réseau /bits	nombre de sous-réseaux
A	10.x.x.x	255.0.0.0	/8	1
B	172.16.x.x — 172.31.x.x	255.255.0.0	/16	16
C	192.168.0.x — 192.168.255.x	255.255.255.0	/24	256

Table 5.2 – Liste des plages d'adresses de réseau

Note
Si une de ces adresses est assignée à une machine, cette machine ne doit alors pas accéder directement à Internet mais passer par une passerelle qui agit en tant que serveur mandataire (« proxy ») pour les services individuels ou sinon effectuer une traduction d'adresse réseau (NAT) (« Network Address Translation ». Un routeur à large bande effectue en général la NAT pour l'environnement du LAN de l'utilisateur grand public.

5.1.4 La gestion du périphérique réseau

La plupart des périphériques matériels sont pris en charge par le système Debian, il y a quelques périphériques de réseau qui exigent, pour les gérer, des microprogrammes non libres d'après les principes du logiciel libre selon Debian. Veuillez consulter Section 9.10.5.

5.2 Configuration moderne de réseau pour ordinateur de bureau

Les interfaces réseau sont ordinairement initialisées dans « `networking.service` » pour l'interface `lo` et « `NetworkMana` pour les autres interfaces sur les systèmes de bureau modernes de Debian sous `systemd`.

Debian peut gérer la connexion réseau via un logiciel démon de gestion tel que NetworkManager (NM) (gestionnaire de réseau et paquets associés).
— Ils sont fournis avec leur propre interface utilisateur graphique (GUI) et en ligne de commandes.
— Ils ont leur propre démon en tant que sytème dorsal.
— Ils permettent une connexion facile de votre système à Internet.
— Ils permettent une gestion facile de la configuration du réseau filaire ou sans fil.
— Ils nous permettent de configurer le réseau indépendamment de l'ancien paquet « `ifupdown` ».

Note
Ne pas utiliser ces outils de configuration automatique du réseau sur un serveur. Ils ont été prévus principalement pour les utilisateurs de système de bureau tournant sur des ordinateurs portables.

Ces outils modernes de configuration du réseau doivent être configurés correctement afin d'éviter des conflits avec l'ancien paquet `ifupdown` et son fichier de configuration « `/etc/network/interfaces` ».

5.2.1 Outils graphiques de configuration du réseau

La documentation officielle de NM sous Debian sont fournies par « `/usr/share/doc/network-manager/README.Debi` »

Essentiellement, la configuration réseau pour un ordinateur de bureau est faite de la manière suivante :

1. Rendez l'utilisateur du bureau, par exemple `toto`, membre du groupe « `netdev` » à l'aide de la commande suivante (vous pouvez aussi le faire automatiquement à l'aide de D-bus sous les environnements de bureau modernes comme GNOME et KDE) :

```
$ sudo usermod -a -G netdev foo
```

2. Gardez la configuration de « `/etc/network/interfaces` » aussi simple que possible comme ce qui suit :

```
auto lo
iface lo inet loopback
```

3. Redémarrez NM de la manière suivante :

```
$ sudo systemctl restart NetworkManager
```

4. Configurez votre réseau à l'aide d'une interface graphique.

Note
Afin d'éviter les conflits avec `ifupdown`, seules les interfaces qui ne sont **pas** listées dans « `/etc/network/interfaces` » sont gérées par NM.

ASTUCE
Si vous désirez étendre les possibilités de configuration de NM, veuillez récupérer les modules d'extension appropriés et les paquets supplémentaires tels que `network-manager-openconnect`, `network-manager-openvpn-gnome`, `network-manager-pptp-gnome`, `mobile-broadband-provider-info`, etc.

5.3 Configuration moderne de réseau sans interface graphique

Sous systemd, le réseau peut aussi être configuré dans `/etc/systemd/network/`. Voir : systemd-resolved(8), resolved.conf(5), et systemd-networkd(8).

Cela permet une configuration moderne de réseau sans interface graphique.

Une configuration de client DHCP peut être réglée en créant « `/etc/systemd/network/dhcp.network` ». Par exemple :

```
[Match]
Name=en*

[Network]
DHCP=yes
```

Une configuration de réseau statique peut être réglée en créant « `/etc/systemd/network/static.network` ». Par exemple :

```
[Match]
Name=en*

[Network]
Address=192.168.0.15/24
Gateway=192.168.0.1
```

5.4 Configuration moderne de réseau pour l'infonuagique

La configuration moderne de réseau pour l'infonuagique peut utiliser les paquets `cloud-init` et `netplan.io` (consulter Section 3.8.4).

Le paquet `netplan.io` prend en charge `systemd-networkd` et `NetworkManager` comme dorsaux de configuration réseau et active la configuration déclarative de réseau en utilisant des données YAML . Si YAML est modifié :

— exécutez la commande « `netplan generate` » pour générer toute la configuration nécessaire du dorsal à partir de YAML ;

— exécuter la commande « `netplan apply` » pour appliquer la configuration aux dorsaux ;

Consultez la « documentation de Netplan », netplan(5), netplan-generate(8) et netplan-apply(8).

Consultez aussi la « documentation de Cloud-init » (particulièrement vers « Sources de configuration » et « Netplan Passthrough ») pour savoir comment `cloud-init` peut intégrer la configuration de `netplan.io` avec des sources de données de remplacement.

5.4.1 Configuration moderne de réseau pour l'infonuagique avec DHCP

Une configuration DHCP de client peut être définie en créant un fichier de données source « `/etc/netplan/50-dhcp.yam`

```
network:
  version: 2
  ethernets:
    all-en:
      match:
        name: "en*"
      dhcp4: true
      dhcp6: true
```

5.4.2 Configuration moderne de réseau pour l'infonuagique avec IP statique

Une configuration de réseau statique peut être définie en créant un fichier de données source « `/etc/netplan/50-static`

```
network:
  version: 2
  ethernets:
    eth0:
      addresses:
        - 192.168.0.15/24
      routes:
        - to: default
          via: 192.168.0.1
```

5.4.3 Configuration moderne de réseau pour l'infonuagique avec Network Manager

Une configuration de réseau de client en utilisant une infrastructure Network Manager peut être définie en créant un fichier de données source « `/etc/netplan/00-network-manager.yaml` » :

```
network:
  version: 2
  renderer: NetworkManager
```

5.5 Configuration réseau de bas niveau

Pour la configuration réseau de bas niveau sous Linux, utilisez les programmes iproute2 (ip(8), ...) .

5.5.1 Commandes Iproute2

Les commandes Iproute2 offrent des possibilités complètes de configuration de bas niveau du réseau. Voici une table de conversion des commandes obsolètes net-tools vers les nouvelles commandes iproute2, etc.

net-tools obsolètes	nouveau iproute2, etc.	manipulation
ifconfig(8)	`ip addr`	adresse de protocole (IP ou IPv6) d'un périphérique
route(8)	`ip route`	entrée de la table de routage
arp(8)	`ip neigh`	entrée de cache ARP ou NDISC
`ipmaddr`	`ip maddr`	adresse multicast
`iptunnel`	`ip tunnel`	tunnel sur IP
nameif(8)	ifrename(8)	nommer les interfaces réseau en se basant sur l'adresse MAC
mii-tool(8)	ethtool(8)	paramétrage du périphérique Ethernet

Table 5.3 – Table de conversion depuis les commandes obsolètes `net-tools` vers les nouvelles commandes `iproute2`

Consulter ip(8) et Linux Advanced Routing & Traffic Control.

5.5.2 Opérations sûres de bas niveau sur le réseau

Vous pouvez utiliser de manière sûre les commandes de réseau de bas niveau de la manière suivante car elles ne modifient pas la configuration du réseau.

ASTUCE
Certains de ces outils de configuration du réseau se trouvent dans « `/usr/sbin/` ». Il vous faudra peut-être utiliser le chemin complet vers la commande comme « `/usr/sbin/ifconfig` » ou ajouter « `/usr/sbin` » à la liste « `$PATH` » dans votre fichier « `~/.bashrc` ».

5.6 Optimisation du réseau

L'optimisation générique du réseau est en dehors des buts de cette documentation. Je ne parle que des sujets pertinents pour une connexion de l'utilisateur grand public.

5.6.1 Rechercher le MTU optimal

NM fixe normalement le MTU (Maximum Transmission Unit) .

Dans certains cas, vous souhaiterez peut-être définir le MTU manuellement après des essais de ping(8) avec l'option « `-M do` » pour envoyer un paquet ICMP avec différentes tailles de paquet de données. Le MTU est la taille maximale possible de paquet de données sans fragmentation IP, plus 28 octets pour l'IPv4 et plus 48 octets pour l'IPv6. Par exemple, ce qui suit trouve un MTU pour la connexion IPv4 de 1460 et un MTU pour la connexion IPv6 de 1500.

commande	description
`ip addr show`	afficher l'état et l'adresse du lien des interfaces actives
`route -n`	afficher toutes les tables de routage sous forme d'adresses numériques
`ip route show`	afficher toutes les tables de routage sous forme d'adresses numériques
`arp`	afficher le contenu actuel des tables de cache d'ARP
`ip neigh`	afficher le contenu actuel des tables de cache d'ARP
`plog`	afficher le journal du démon ppp
`ping yahoo.com`	vérifier la connexion internet vers « `yahoo.com` »
`whois yahoo.com`	vérifier qui a enregistré « `yahoo.com` » dans la base de données des domaines
`traceroute yahoo.com`	tracer la connexion Internet vers « `yahoo.com` »
`tracepath yahoo.com`	tracer la connexion Internet vers « `yahoo.com` »
`mtr yahoo.com`	tracer la connexion Internet vers « `yahoo.com` » (de manière répétitive)
`dig [@dns-server.com] example.com [{a|mx|any}]`	vérifier les enregistrements DNS de « `example.com` » par « `dns-serveur.com` » pour un enregistrement « `a` », « `mx` » ou « `any` »
`iptables -L -n`	vérifier le filtre de paquets
`ss -a`	rechercher tous les ports ouverts
`ss -l`	rechercher les ports à l'écoute
`ss -l -t -n`	rechercher les ports TCP à l'écoute (numérique)
`dlint example.com`	vérifier les informations de zone DNS de « `example.com` »

Table 5.4 – Liste des commandes de réseau de bas niveau

paquets	popularité	taille	description
`iftop`	V:6, I:83	93	afficher l'utilisation de la bande passante d'une interface réseau
`iperf`	V:2, I:31	23	outil de mesure de la bande passante du protocole Internet
`ifstat`	V:0.5, I:5.1	52	InterFace STATistics Monitoring (surveillance des statistiques de l'interface)
`bmon`	V:2, I:19	154	surveillance portable de la bande passante et estimation du débit
`ethstatus`	V:0.2, I:2.3	41	script qui mesure rapidement le débit d"une interface réseau
`bwm-ng`	V:1.2, I:8.8	95	moniteur de bande passante simple en mode console
`ethstats`	V:0.06, I:0.35	21	moniteur de statistiques Ethernet en mode console

Table 5.5 – Liste des outils d'optimisation du réseau

```
$ ping -4 -c 1 -s $((1500-28)) -M do www.debian.org
PING  (149.20.4.15) 1472(1500) bytes of data.
ping: local error: message too long, mtu=1460

---  ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

$ ping -4 -c 1 -s $((1460-28)) -M do www.debian.org
PING  (130.89.148.77) 1432(1460) bytes of data.
1440 bytes from klecker-misc.debian.org (130.89.148.77): icmp_seq=1 ttl=50 time=325 ms

---  ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 325.318/325.318/325.318/0.000 ms
$ ping -6 -c 1 -s $((1500-48)) -M do www.debian.org
PING www.debian.org(mirror-csail.debian.org (2603:400a:ffff:bb8::801f:3e)) 1452 data bytes
1460 bytes from mirror-csail.debian.org (2603:400a:ffff:bb8::801f:3e): icmp_seq=1 ttl=47  ↩
    time=191 ms

--- www.debian.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 191.332/191.332/191.332/0.000 ms
```

Ce processus est la découverte du chemin MTU (PMTU) (RFC1191) et la commande tracepath(8) peut l'automatiser.

environnement de réseau	**MTU**	**justification**
Lien commuté (IP : PPP)	576	standard
Lien Ethernet (IP : DHCP ou fixe)	1500	standard et par défaut

Table 5.6 – Lignes directrices pour une valeur optimale de MTU

En plus de ces lignes directrices basiques, vous devriez connaitre ce qui suit :

— Toute utilisation d'une méthode de tunneling (VPN, etc.) peut réduire le MTU optimal en raison de la surcharge qu'elle engendre.
— La valeur de MTU ne doit pas excéder la valeur expérimentale déterminée de PMTU .
— La valeur de MTU la plus élevée est généralement meilleure lors que les autres limitations sont remplies.

La taille maximum de segment (MSS : « maximum segment size ») est utilisée comme mesure de remplacement de la taille des paquets. La relation entre MSS et MTU est la suivante :

— MSS = MTU - 40 pour IPv4
— MSS = MTU - 60 pour IPv6

Note
Les optimisations basées sur iptables(8) (consultez Section 5.7) peuvent limiter la taille des paquets au MSS, ce qui est utile pour le routeur. Consultez "TCPMSS" dans iptables(8).

5.6.2 Optimisation de TCP sur le réseau Internet

Le débit TCP peut être maximisé en ajustant les paramètres de taille de tampon TCP comme cela est décrit dans « Ajustement de TCP » pour les réseaux WAN modernes de haut débit et de faible latence. À ce jour, les paramètres par défaut de Debian fonctionnent bien même pour mon réseau local connecté à l'aide du service rapide FTTP à 1Gb/s.

5.7 Infrastructure de netfilter

Netfilter fournit l'infrastructure pour unpare-feu dynamique (« stateful firewall ») et la traduction d'adresses réseau (NAT) (« network address translation ») avec des modules du noyau de Linux (consultez Section 3.10).

paquets	popularité	taille	description
nftables	V:301, I:889	189	outils d'administration pour le filtrage des paquets et NAT (Netfilter) (successeur à {ip,ip6,arp,eb}tables)
iptables	V:402, I:635	2408	outils d'administration pour netfilter (iptables(8) pour IPv4, ip6tables(8) for IPv6)
arptables	V:0.1, I:1.6	102	outils d'administration pour netfilter (arptables(8) pour ARP)
ebtables	V:14, I:22	276	outils d'administration pour netfilter (ebtables(8) pour le pontage Ethernet)
iptstate	V:0.1, I:1.6	130	surveillance continue de l'état de netfilter (semblable à top(1))
ufw	V:87, I:110	862	Uncomplicated Firewall (UFW) est un programme pour gérer un pare-feu netfilter
gufw	V:5, I:10	3682	interface graphique pour Uncomplicated Firewall (UFW)
firewalld	V:16, I:22	2581	firewalld est un programme de pare-feu géré dynamiquement avec gestion des zones de réseau
firewall-config	V:0.8, I:3.0	1096	interface graphique pour firewalld
shorewall-init	V:0.13, I:0.31	88	initialisation de Shoreline Firewall
shorewall	V:2.2, I:4.5	3090	Shoreline Firewall, générateur de fichier de configuration pour netfilter
shorewall-lite	V:0.03, I:0.05	71	Shoreline Firewall, générateur de fichier de configuration pour netfilter (version légère)
shorewall6	V:0.7, I:1.2	1334	Shoreline Firewall, générateur de fichier de configuration pour netfilter (version IPv6)
shorewall6-lite	V:0.01, I:0.01	71	Shoreline Firewall, générateur de fichier de configuration pour netfilter (version légère, IPv6)

Table 5.7 – Liste d'outils de pare-feu

See The netfilter.org "nftables" project for the newer user space nft(8) command in the `nftables` package. (Linux kernel >= 3.13)

See The netfilter.org "iptables" project for the older user space iptables(8) and similar commands in the `iptables` and similar packages. (Linux kernel >= 2.4.x)

Vous pouvez configurer vous-même netfilter de manière interactive depuis l'interpréteur de commandes, enregistrer son état avec iptables-save(8) et le restaurer par l'intermédiaire d'un script d'init avec iptables-restore(8) lors du redémarrage du système.

Des scripts d'assistant tels que shorewall facilitent ce processus.

Consultez les documentations se trouvant dans la documentation de Netfilter (ou dans « `/usr/share/doc/iptables/htm`

— Linux Networking-concepts HOWTO
— Linux 2.4 Packet Filtering HOWTO
— Linux 2.4 NAT HOWTO
— nftables HOWTO

Chapitre 6

Applications réseau

Après avoir établi une connexion réseau (consultez Chapitre 5), vous pouvez faire tourner diverses applications réseau.

ASTUCE
Pour un guide spécifique de Debian moderne sur les infrastructures réseaux, lisez Le Livre de l'Administrateur Debian - Infrastructure réseau.

ASTUCE
Si vous avez activé la « validation en deux étapes » avec certains fournisseurs de services Internet, vous devez obtenir un mot de passe d'application pour accéder aux services POP et SMTP à partir de votre programme. Vous devrez peut-être valider votre adresse IP d'hôte à l'avance.

6.1 Navigateurs Web

Il y a de nombreux paquets de navigateurs web permettant d'accéder à des contenus distants avec le protocole de transfert hypertexte (« Hypertext Transfer Protocol (HTTP) »).

paquet	popularité	taille	type	description du navigateur web
`chromium`	V:37, I:101	324490	X	Chromium (navigateur libre de Google)
`firefox`	V:15, I:21	317809	, ,	Firefox (navigateur libre de Mozilla, uniquement disponible dans Debian Unstable)
`firefox-esr`	V:188, I:423	299668	, ,	Firefox ESR (Firefox Extended Support Release = Firefox Support à Long Terme)
`epiphany-browser`	V:2.0, I:9.7	6737	, ,	GNOME, Epiphany respectant HIG
`konqueror`	V:26, I:110	8191	, ,	KDE, Konqueror
`dillo`	V:0.5, I:3.8	2203	, ,	Dillo (navigateur léger, basé sur FLTK)
`w3m`	V:11, I:125	2853	texte	w3m
`lynx`	V:30, I:454	2031	, ,	Lynx
`elinks`	V:2, I:14	1839	, ,	ELinks
`links`	V:2, I:19	2344	, ,	Links (texte uniquement)
`links2`	V:0.8, I:9.2	5497	graphique	Links (graphique en mode console sans X)

Table 6.1 – Liste de navigateurs web

6.1.1 Usurpation de la chaîne User-Agent

Pour accéder à certains sites web excessivement restrictifs, vous devrez peut-être usurper la chaîne User-Agent renvoyée par le programme de navigation web. Consulter :

— documentation web de MDN : userAgent ;
— développeurs de Chrome : Override the user agent string ;
— comment changer votre agent utilisateur ;
— comment modifier l'agent utilisateur dans Chrome, Firefox, Safari, etc. ;
— comment changer l'agent utilisateur de votre navigateur sans installer d'extensions ;
— comment changer l'agent utilisateur dans Gnome Web (epiphany).

6.1.2 Extension de navigateur

Tous les navigateurs graphiques modernes gèrent le code source basé sur les extensions de navigateur et cela devient normalisé comme extensions web.

6.2 Le système de courrier électronique

Cette section se concentre sur les postes de travail mobiles typiques utilisant des connexions Internet de qualité grand public.



Attention
Si vous êtes sur le point de configurer le serveur de courrier pour échanger directement du courrier avec Internet, vous feriez mieux de lire ce document élémentaire.

6.2.1 Bases du courrier électronique

Un courrier électronique est composé de trois parties : l'enveloppe, l'en-tête et le corps du message.

— Les renseignements « To » et « From » de l'enveloppe sont utilisés par le SMTP pour délivrer le courrier électronique (« From » dans l'enveloppe indique l'adresse de rebond, « From_ », etc.).
— Les renseignements « To » et « From » de l'en-tête sont affichés par le client de messagerie (même s'ils sont généralement identiques à ceux de l'enveloppe, ce n'est pas toujours le cas).
— Le format de courriel pour les données d'en-tête et de corps est étendu par Multipurpose Internet Mail Extensions (MIME) du texte ASCII brut à d'autres codages de caractères, ainsi qu'aux pièces jointes audio, vidéo, images et programmes d'application.

Les clients de messagerie, basés sur une interface graphique complète, offrent toutes les fonctions suivantes en utilisant une configuration intuitive.

— Il crée et interprète les données de l'en-tête et du corps du message en utilisant Multipurpose Internet Mail Extensions (MIME) pour traiter le type et l'encodage des données du contenu.
— Il s'authentifie auprès des serveurs SMTP et IMAP du FAI à l'aide de l'ancienne authentification d'accès basic ou celle moderne OAuth 2.0. (Pour OAuth 2.0, définissez-le à l'aide des paramètres de l'environnement du bureau. Par exemple, « Paramètres » -> « Comptes en ligne ».)
— Il envoie le message au serveur SMTP hôte du FAI écoutant le port de soumission de message (587).
— Il reçoit le message stocké sur le serveur du FAI à partir du port TLS/IMAP4 (993).
— Il peut filtrer les courriels en fonction de leurs attributs.
— Il peut offrir des fonctionnalités supplémentaires : contacts, agenda, tâches, mémos.

paquet	popularité	taille	type
evolution	V:27, I:227	521	programme X avec une interface graphique (GNOME, suite « groupware »)
thunderbird	V:39, I:100	304278	programme d'interface graphique X (GTK, Mozilla Thunderbird)
kmail	V:43, I:101	25384	programme X avec une interface graphique (KDE)
mutt	V:11, I:78	7378	programme de terminal en mode caractère, probablement utilisé avec vim
mew	V:0.01, I:0.14	2319	programme de terminal en mode caractères sous (x)emacs

Table 6.2 – Liste d'agents de courrier électronique de l'utilisateur (MUA)

6.2.2 Limite du service de courriels moderne

Les services de courriel modernes sont soumis à certaines limitations afin de minimiser l'exposition aux problèmes de pourriel (courriel non désiré et non sollicité).

— Il n'est pas réaliste de mettre en œuvre un serveur SMTP sur un réseau grand public pour envoyer de manière fiable un courriel directement vers une machine distante.
— Un courriel peut être rejeté par n'importe quel hôte sur le chemin vers la destination silencieusement, à moins qu'il ne paraisse aussi authentique que possible.
— Il n'est pas réaliste d'espérer qu'un smarthost (relais) unique puisse envoyer des courriels provenant d'adresses de sources de courriel sans lien vers une machine distante de manière fiable.

Cela est dû au fait que :

— les connexions au port SMTP (25) des hôtes desservis par le réseau grand public vers Internet sont bloquées ;
— les connexions du port SMTP (25) vers les hôtes desservis par le réseau grand public à partir d'Internet sont bloquées ;
— les messages sortants des hôtes desservis par le réseau grand public vers Internet ne peuvent être envoyés qu'à travers le port de soumission de message (587) ;
— les techniques anti-pourriels telles queDomainKeys Identified Mail (DKIM), Sender_Policy_Framework (SPF) et Domain-based Message Authentication, Reporting and Conformance (DMARC) sont couramment utilisées pour le filtrage du courrier ;
— le service de DomainKeys Identified Mail peut être fourni pour vos messages envoyés par l'intermédiaire du smarthost ;
— Le relais de courriel peut réécrire l'adresse du courriel source dans l'en-tête du message à celle de votre compte de messagerie sur le relais pour empêcher l'usurpation d'adresse de courriel.

6.2.3 Attente du service de courriels historique

Certains programmes sur Debian s'attendent à accéder à la commande `/usr/sbin/sendmail` pour envoyer des courriels comme comportement par défaut ou personnalisé puisque le service de messagerie sur un système UNIX fonctionnait historiquement comme :

— un courriel est créé sous forme de fichier texte ;
— le courriel est transmis à la commande `/usr/sbin/sendmail` ;
— pour l'adresse de destination sur le même hôte, la commande `/usr/sbin/sendmail` effectue la livraison locale du courriel en l'ajoutant au fichier `/var/mail/$username`,
  — commandes qui attendent cette fonctionnalité : `apt-listchanges`, `cron`, `at`, ... ;
— pour l'adresse de destination sur l'hôte distant, la commande `/usr/sbin/sendmail` effectue le transfert à distance du courriel vers l'hôte de destination trouvé par l'enregistrement MX du DNS en utilisant SMTP,
  — commandes qui attendent cette fonctionnalité : `popcon`, `reportbug`, `bts`, ...

6.2.4 Agent de transport de courrier électronique (« MTA »)

Les stations de travail mobiles Debian pourront être configurées uniquement avec des clients de messageriecomplets basés sur une interface graphique sans le programme d'agent de transfert de courrier (MTA) après Debian 12 Bookworm.

Debian installait traditionnellement un programme MTA pour prendre en charge les programmes attendant la commande `/usr/sbin/sendmail`. Un tel MTA sur les postes de travail mobiles doit respecter la Section 6.2.2 et la Section 6.2.3.

Pour les postes de travail mobiles, le choix typique de MTA est soit `exim4-daemon-light` ou `postfix` avec son option d'installation telle que « Courriel envoyé par smarthost, reçue par SMTP ou fetchmail » sélectionnée. Ce sont des MTA légers qui respectent "`/etc/aliases`".

ASTUCE
Configurer `exim4` pour envoyer le courrier Internet par l'intermédiaire de plusieurs smarthost correspondants pour plusieurs sources d'adresses de courriel n'est pas trivial. Si vous avez besoin d'une telle capacité pour certains programmes, configurez-les pour utiliser `msmtp` qui est facile à configurer pour plusieurs sources d'adresses de courriel. Ensuite, laissez le MTA principal seulement pour une seule adresse e-mail.

paquet	popularité	taille	description
`exim4-daemon-light`	V:197, I:206	1777	Agent de transport de courrier électronique Exim4 (MTA : par défaut dans Debian)
`exim4-daemon-heavy`	V:4.5, I:4.8	1955	Agent de transport de courriel Exim4 (MTA : alternative flexible)
`exim4-base`	V:201, I:211	1687	Documentation d'Exim4 (texte) et fichiers communs
`exim4-doc-html`	I:0.94	3809	Documentation d'Exim4 (html)
`exim4-doc-info`	I:0.51	649	Documentation d'Exim4 (info)
`postfix`	V:99, I:104	4277	Agent de transport de courriel Postfix (MTA : alternative sécurisée)
`postfix-doc`	I:4.0	5030	Documentation de Postfix (html+texte)
`sasl2-bin`	V:4.5, I:9.4	370	Implémentation de l'API Cyrus SASL (complément à Postfix pour SMTP AUTH)
`cyrus-sasl2-doc`	I:0.68	2140	Cyrus SASL - documentation
`msmtp`	V:10, I:15	822	MTA léger
`msmtp-mta`	V:7.8, I:9.8	139	MTA léger (extension de compatibilité de sendmail pour `msmtp`)
`nullmailer`	V:7.2, I:7.6	483	MTA simple, pas de courrier local
`ssmtp`	V:3.8, I:5.6	134	MTA simple, pas de courrier local
`sendmail-bin`	V:9.4, I:9.7	1954	MTA complet (seulement si vous avez déjà des connaissances)
`git-email`	V:0.4, I:8.6	1229	programme git-send-email(1) pour envoyer une série de courriels de correctif

Table 6.3 – Liste de paquets basiques concernant des agents de transport du courriel

6.2.4.1 Configuration d'exim4

Pour le courrier d'Internet par l'intermédiaire d'un smarthost, vous (re)configurerez les paquets `exim4-*` comme suit :

```
$ sudo systemctl stop exim4
$ sudo dpkg-reconfigure exim4-config
```

Choisir « envoi via relais (« smarthost ») - réception SMTP ou fetchmail » : pour « Configuration du serveur de courriel ».

Définir « Nom de courriel du système » à sa valeur par défaut qui est le nom pleinement qualifié (FQDN, consultez Section 5.1.1).

Définir « Liste d'adresses IP où Exim sera en attente de connexions SMTP entrantes » à sa valeur par défaut qui est « 127.0.0.1 ; ::1 ».

Supprimer le contenu de « Autres destinations dont le courriel doit être accepté ».

Supprimer le contenu de « Machines à relayer :».

Définir « dresse IP ou nom d'hôte du relais sortant : à « smtp.hostname.dom:587 ».

Sélectionner « Non » pour « Faut-il cacher le nom local de courriel dans les courriels sortants ? ». (Utiliser plutôt « `/etc/email-addresses` » comme dans Section 6.2.4.3.)

Donner à « Faut-il optimiser les requêtes DNS (connexion à la demande) ? » l'une des réponses suivantes :

— « Non » si le système est connecté à Internet au démarrage.
— « Oui » si le système n'est **pas** connecté à Internet au démarrage.

Définir « Méthode de distribution du courrier local : » à « Format « mbox » dans /var/mail ».

Sélectionner « Oui » pour « Faut-il séparer la configuration dans plusieurs fichiers ? :».

Créer les entrées de mots de passe pour le smarthost en éditant « `/etc/exim4/passwd.client` ».

```
$ sudo vim /etc/exim4/passwd.client
 ...
$ cat /etc/exim4/passwd.client
^smtp.*\.hostname\.dom:username@hostname.dom:password
```

Configurez exim4(8) avec « `QUEUERUNNER='queueonly'` », « `QUEUERUNNER='nodaemon'` », etc., dans « `/etc/defau` pour minimiser l'utilisation des ressources système (facultatif).

Lancer `exim4` par la commande suivante :

```
$ sudo systemctl start exim4
```

Le nom de machine dans « `/etc/exim4/passwd.client` » ne doit pas être un alias. Vérifiez le nom de machine réel comme suit :

```
$ host smtp.hostname.dom
smtp.hostname.dom is an alias for smtp99.hostname.dom.
smtp99.hostname.dom has address 123.234.123.89
```

J'utilise une expression rationnelle dans « `/etc/exim4/passwd.client` » pour contourner le problème d'alias. SMTP AUTH fonctionne probablement même dans le cas où le FAI déplace la machine pointée par l'alias.

Vous pouvez mettre à jour vous-même la configuration d'`exim4` de la façon suivante :

— Mettre à jour les fichiers de configuration d'`exim4` dans « `/etc/exim4/` ».
  — Créer « `/etc/exim4/exim4.conf.localmacros` » pour configurer les macros et éditer « `/etc/exim4/exim4.c` (configuration en un seul fichier).
  — Créer de nouveaux fichiers ou éditer des fichiers existants dans les sous-répertoires de « `/etc/exim4/exim4.conf` (configuration séparée en plusieurs fichiers).
— Exécutez « `systemctl reload exim4` ».



Attention
Le lancement d'`exim4` est long si on a choisi « Non » (valeur par défaut) à la demande « Faut-il optimiser les requêtes DNS (connexion à la demande) ? » lors de la configuration debconf et que le système n'est **pas** connecté à Internet lors du démarrage.

Veuillez lire le guide officiel se trouvant à « `/usr/share/doc/exim4-base/README.Debian.gz` » et update-exim4.conf(8).



AVERTISSEMENT
En pratique, utilisez SMTP avec STARTTLS sur le port 587 ou SMTPS (SMTP au-dessus de SSL) sur le port 465, au lieu de SMTP simple sur le port 25.

6.2.4.2 Configuration de postfix avec SASL

Pour utiliser le courrier électronique d'Internet par l'intermédiaire d'un smarthost, vous devrez d'abord lire la documentation postfix et les pages de manuel importantes.

commande	fonction
postfix(1)	Programme de contrôle de postfix
postconf(1)	Utilitaire de configuration de postfix
postconf(5)	Paramètres de configuration de postfix
postmap(1)	Maintenance des tables de consultation de postfix
postalias(1)	Maintenance de la base de données des alias de postfix

Table 6.4 – Liste des pages de manuel importantes de postfix

Vous (re)configurez les paquets `postfix` et `sasl2-bin` comme suit :

```
$ sudo systemctl stop postfix
$ sudo dpkg-reconfigure postfix
```

Choisir « Internet avec smarthost ».

Définissez « machine de relais SMTP (blanc pour aucun): » à « `[smtp.hostname.dom]:587` » et configurez-le de la manière suivante :

```
$ sudo postconf -e 'smtp_sender_dependent_authentication = yes'
$ sudo postconf -e 'smtp_sasl_auth_enable = yes'
$ sudo postconf -e 'smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd'
$ sudo postconf -e 'smtp_sasl_type = cyrus'
$ sudo vim /etc/postfix/sasl_passwd
```

Créez les entrées de mots de passe pour le smarthost :

```
$ cat /etc/postfix/sasl_passwd
[smtp.hostname.dom]:587     username:password
$ sudo postmap hush:/etc/postfix/sasl_passwd
```

Lancez `postfix` comme suit :

```
$ sudo systemctl start postfix
```

Ici, l'utilisation de « [» et «] » dans le dialogue de `dpkg-reconfigure` et « `/etc/postfix/sasl_passwd` » permet de s'assurer de ne pas vérifier l'enregistrement MX mais d'utiliser directement le nom exact de la machine indiquée. Consultez « Enabling SASL authentication in the Postfix SMTP client » dans « `/usr/share/doc/postfix/html/S`

fichier	fonction	application
`/etc/mailname`	nom de machine par défaut pour le courrier (sortant)	Spécifique à Debian, mailname(5)
`/etc/email-addresses`	usurpation du nom de machine pour le courriel sortant	Spécifique à exim(8), exim4-config_files(5)
`/etc/postfix/generic`	usurpation du nom de machine pour le courriel sortant	Spécifique à postfix(1) specific, activé après l'exécution de la commande postmap(1).
`/etc/aliases`	alias du nom de compte pour le courrier entrant	général, activé après l'exécution de la commande newaliases(1).

Table 6.5 – Liste des fichiers de configuration liés aux adresses de courriel

6.2.4.3 Configuration de l'adresse de courriel

Il existe plusieurs fichiers de configuration de l'adresse de courriel pour l'acheminement du courriel, sa diffusion et les agents d'utilisateur.

Le **nom de courriel** (mailname » dans le fichier « `/etc/mailname` » est habituellement un nom de domaine entièrement qualifié (FQDN) qui est résolu vers l'une des adresses IP de la machine. Pour les stations de travail mobiles qui n'ont pas de nom de machine pouvant être résolu par une adresse IP, définissez ce **mailname** à la valeur donnée par « `hostname -f` ». (C'est un choix sûr et qui fonctionne à la fois avec `exim4-*` et `postfix`.)

ASTUCE
Le contenu de « `/etc/mailname` » est utilisé par de nombreux programmes autres que les MTA pour définir leur comportement par défaut. Pour `mutt`, définissez les variables « `hostname` » et « `from` » dans le fichier `~/muttrc` pour passer outre la valeur de **mailname**. Pour les programmes du paquet `devscripts`, comme bts(1) et dch(1), exportez les variables d'environnement « `$DEBFULLNAME` » et « `$DEBEMAIL` » afin de passer outre cette définition.

ASTUCE
Le paquet `popularity-contest` envoie normalement un courriel depuis le compte de l'administrateur avec un nom de domaine pleinement qualifié (FDQN). Vous devez définir `MAILFROM` dans `/etc/popularity-contest.conf` comme c'est décrit dans le fichier `/usr/share/popularity-contest/default.conf`. Sinon, votre courriel sera rejeté par le serveur SMTP sur « smarthost ». Bien que ce soit fastidieux, cette approche est plus sûre que la réécriture par le MTA de l'adresse source pour tous les courriels en provenance de l'administrateur (« root ») et devrait être utilisé pour les autres démons et les scripts des tâches planifiées (« cron ».

Lors de la définition de **mailname** avec la valeur donnée par « `hostname -f` », l'usurpation de l'adresse source du courrier par le MTA peut être réalisée par l'intermédiaire :

— du fichier « `/etc/email-addresses` » pour exim4(8) comme expliqué dans exim4-config_files(5)
— du fichier « `/etc/postfix/generic` » pour postfix(1) comme expliqué dans generic(5)

Pour `postfix`, les étapes suivantes sont nécessaires :

```
# postmap hash:/etc/postfix/generic
# postconf -e 'smtp_generic_maps = hash:/etc/postfix/generic'
# postfix reload
```

Vous pouvez tester la configuration de l'adresse de courriel de la manière suivante :

— exim(8) avec les options `-brw, -bf, -bF, -bV,` …
— postmap(1) avec l'option `-q`.

ASTUCE
Il existe, avec Exim, un certain nombre de programmes utilitaires tels qu'exiqgrep(8) et exipick(8). Consultez « `dpkg -L exim4-base|grep man8/` » pour les commandes disponibles.

6.2.4.4 Opération de base du MTA

Il y a quelques opérations de base du MTA. Certaines peuvent être effectuées à l'aide de l'interface de compatibilité avec sendmail(1).

commande exim	commande postfix	description
`sendmail`	`sendmail`	lire les courriels depuis l'entrée standard et les classer pour la diffusion (`-bm`)
`mailq`	`mailq`	afficher la file d'attente des courriels avec leur état et leur identifiant de file d'attente (« queue ID ») (`-bp`)
`newaliases`	`newaliases`	initialiser la base de données des alias (`-I`)
`exim4 -q`	`postqueue -f`	supprimer les courriels en attente (`-q`)
`exim4 -qf`	`postsuper -r ALL deferred; postqueue -f`	supprimer tous les courriels
`exim4 -qff`	`postsuper -r ALL; postqueue -f`	supprimer tous les courriels gelés
`exim4 -Mg queue_id`	`postsuper -h queue_id`	geler un message d'après son identifiant de file d'attente
`exim4 -Mrm queue_id`	`postsuper -d queue_id`	supprimer un message d'après son identifiant de file d'attente
N/A	`postsuper -d ALL`	supprimer tous les messages

Table 6.6 – Liste des opérations de base du MTA

ASTUCE
Ce peut être une bonne idée de supprimer tous les messages à l'aide d'un script placé dans « `/etc/ppp/ip-up.d/*` ».

6.3 Le serveur et les utilitaires d'accès à distance (SSH)

SSH, le « Secure SHell », est la manière **sûre** de se connecter au travers d'Internet. Une version libre de SSH, appelée OpenSSH, est disponible sous Debian sous forme des paquets `openssh-client` et `openssh-server`.

Pour l'utilisateur, ssh(1) fonctionne comme un telnet(1) intelligent et plus sûr. Contrairement à la commande `telnet`, la commande `ssh` ne s'arrête pas avec le caractère d'échappement de `telnet` (valeur initiale par défaut Ctrl-]).

Bien que `shellinabox` ne soit pas un programme SSH, il est répertorié ici comme une alternative intéressante pour l'accès au terminal distant.

Consulter également Section 7.9 pour la connexion aux programmes clients X distants.



Attention
Consultez Section 4.6.3 si votre serveur SSH est accessible depuis Internet.

paquet	popularité	taille	outil	description
openssh-client	V:709, I:997	3611	ssh(1)	client de l'interpréteur de commandes sécurisé
openssh-server	V:781, I:824	3615	sshd(8)	serveur de l'interpréteur de commandes sécurisé
ssh-askpass	I:15	103	ssh-askpass(1)	demande à l'utilisateur une phrase de passe pour ssh-add (X natif)
ssh-askpass-gnome	V:0.4, I:2.9	46	gnome-ssh-askpass(1)	demande à l'utilisateur d'une phrase de passe pour ssh-add (GNOME)
ssh-askpass-fullscreen	V:0.08, I:0.36	41	ssh-askpass-fullscreen(1)	demande à l'utilisateur d'une phrase de passe pour ssh-add (GNOME) de manière jolie
shellinabox	V:0.60, I:0.88	528	shellinaboxd(1)	serveur web pour émulateur de terminal VT100 accessible dans un navigateur

Table 6.7 – Liste des serveurs et des utilitaires d'accès à distance

ASTUCE
Utilisez le programme screen(1) pour qu'un processus de l'interpréteur de commandes distant survive à une interruption de la connexion (consultez Section 9.1.2).

6.3.1 Bases de SSH

Le démon SSH OpenSSH ne prend en charge que le protocole SSH 2.

Veuillez lire « `/usr/share/doc/openssh-client/README. Debian.gz` », ssh(1), sshd(8), ssh-keygen(1), ssh-add(1) et ssh-agent(1).



AVERTISSEMENT
Il ne faut pas que « `/etc/ssh/sshd_not_to_be_run` » soit présent si l'on souhaite faire tourner le serveur OpenSSH.
N'activez pas l'authentification basée sur rhost (`HostbasedAuthentication` dans `/etc/ssh/sshd_config`).

fichier de configuration	description du fichier de configuration
`/etc/ssh/ssh_config`	valeurs par défauts des paramètres du client SSH, consultez ssh_config(5)
`/etc/ssh/sshd_config`	valeurs par défauts des paramètres du serveur SSH, consultez sshd_config(5)
`~/.ssh/authorized_keys`	clés publiques SSH par défaut utilisées pour se connecter à ce compte sur ce serveur SSH
`~/.ssh/id_rsa`	clé secrète SSH-2 RSA de l'utilisateur
`~/.ssh/id_`*`key-type-name`*	clé secrète SSH-2 *clé-type-nom* telle que `ecdsa`, `ed25519`, ... de l'utilisateur

Table 6.8 – Liste des fichiers de configuration de SSH

Ce qui suit permettra de démarrer un connexion ssh(1) depuis un client :

commande	description
`ssh username@hostname.domain.ext`	connexion avec le mode par défaut
`ssh -v username@hostname.domain.ext`	connexion avec le mode par défaut et les messages de débogage
`ssh -o PreferredAuthentications=password username@hostname.domain.ext`	forcer l'utilisation d'un mot de passe avec SSH version 2
`ssh -t username@hostname.domain.ext passwd`	exécuter le programme `passwd` pour mettre à jour le mot de passe sur un hôte distant

Table 6.9 – Liste d'exemples de démarrage du client SSH

6.3.2 Nom d'utilisateur sur l'hôte distant

Si vous utilisez le même nom d'utilisateur sur l'hôte local et l'hôte distant, vous pouvez éviter de taper « `nomutilisateur@` »

Même si vous utilisez un nom d'utilisateur différent sur la machine locale et la machine distante, vous pouvez l'éliminer en utilisant « `~/.ssh/config` ». Pour le service Salsa de Debian avec le nom de compte « `toto-guest` », vous devrez configurer « `~/.ssh/config` » afin qu'il contienne ceci :

```
Host salsa.debian.org people.debian.org
User foo-guest
```

6.3.3 Se connecter sans mot de passe distant

Il est possible d'éviter de se rappeler les mots de passe des systèmes distants en utilisant « `PubkeyAuthentication` » (protocole SSH-2).

Sur le système distant, définissez les entrées respectives, « `PubkeyAuthentication yes` », dans « `/etc/ssh/sshd_c`

Générez ensuite localement les clés d'identification et installez la clé publique sur le système distant en faisant ce qui suit :

```
$ ssh-keygen -t rsa
$ cat .ssh/id_rsa.pub | ssh user1@remote "cat - >>.ssh/authorized_keys"
```

Vous pouvez ajouter des options aux entrées dans « `~/.ssh/authorized_keys` » pour limiter les hôtes et pour exécuter des commandes spécifiques. Consulter sshd(8) « AUTHORIZED_KEYS FILE FORMAT ».

6.3.4 Clients SSH exotiques

Il existe quelques clients SSH libres disponibles pour d'autres plateformes.

environnement	programme SSH libre
Windows	puTTY (PuTTY : un client libre SSH et Telnet) (GPL)
Windows (cygwin)	SSH dans cygwin (Cygwin : obtenez cette sensation Linux sur Windows) (GPL)
Mac OS X	OpenSSH ; utilise `ssh` dans l'application Terminal (GPL)

Table 6.10 – Liste des clients SSH libres pour d'autres plateformes

6.3.5 Configurer ssh-agent

Il est plus sûr de protéger les clés secrètes de votre authentification SSH avec une phrase de passe. Si la phrase de passe n'a pas été définie, utilisez « `ssh-keygen -p` » pour le faire.

Placez votre clé publique SSH (par exemple « `~/.ssh/id_rsa.pub` ») dans « `~/.ssh/authorized_keys` » sur la machine distante en utilisant une connexion basée sur un mot de passe comme décrit ci-dessus.

```
$ ssh-agent bash
$ ssh-add ~/.ssh/id_rsa
Enter passphrase for /home/username/.ssh/id_rsa:
Identity added: /home/username/.ssh/id_rsa (/home/username/.ssh/id_rsa)
```

Il n'y a plus besoin de mot de passe distant, à partir de maintenant, pour la commande suivante :

```
$ scp foo username@remote.host:foo
```

Pressez ^D pour quitter la session de l'agent ssh.

Pour le serveur X, le script de démarrage normal de Debian exécute `ssh-agent` comme processus-père. Vous n'aurez donc à exécuter `ssh-add` qu'une seule fois. Pour davantage d'informations, veuillez lire ssh-agent(1) et ssh-add(1).

6.3.6 Envoi d'un courriel à partir d'un hôte distant

Si vous avez un compte d'interpréteur SSH sur un serveur avec des paramètres DNS appropriés, vous pouvez envoyer un courriel généré sur votre poste de travail sous la forme d'un courriel véritablement envoyé à partir du serveur distant.

```
$ ssh username@example.org /usr/sbin/sendmail -bm -ti -f "username@example.org" < mail_data ↩
    .txt
```

6.3.7 Redirection de port pour un tunnel SMTP/POP3

Pour mettre en place un tube pour se connecter au port 25 du `serveur-distant` depuis le port 4025 de `localhost`, et au port 110 du `serveur-distant` depuis le port 4110 de `localhost` au travers de `ssh`, exécutez ce qui suit sur la machine locale :

```
# ssh -q -L 4025:remote-server:25 4110:remote-server:110 username@remote-server
```

C'est une manière sécurisée d'effectuer une connexion à des serveurs SMTP / POP3 par Internet. Définissez l'entrée « `AllowTcpForwarding` » à « `yes` » dans « `/etc/ssh/sshd_config` » sur la machine distante.

6.3.8 Comment arrêter le système distant par SSH

Vous devez protéger le processus qui effectue « `shutdown -h now` » (consultez Section 1.1.8) de l'arrêt de SSH en utilisant la commande at(1) (consultez Section 9.4.13) comme suit :

```
# echo "shutdown -h now" | at now
```

Lancer « `shutdown -h now` » dans une sessionscreen(1) (consultez Section 9.1.2) est une autre manière d'effectuer la même chose.

6.3.9 Résoudre les problèmes avec SSH

Si vous rencontrez des problèmes, vérifiez les permissions des fichiers de configuration et lancez `ssh` avec l'option « `-v` ».

Utilisez l'option « `-p` » si vous êtes administrateur et que vous rencontrez des problèmes avec un pare-feu. Cela évite l'utilisation des ports 1 — 1023 du serveur.

Si les connexions `ssh` vers un site distant s'arrêtent subitement de fonctionner, cela peut être suite à des bidouilles de l'administrateur, le plus probablement un changement de « `host_key` » pendant une maintenance du système. Après s'être assuré que c'est bien le cas et que personne n'essaie de se faire passer pour la machine distante par une habile bidouille, on peut se reconnecter en supprimant sur la machine locale l'entrée « `host_key` » de « `~/.ssh/known_hosts` ».

6.4 Le serveur et les utilitaires d'impression

Dans un ancien système de type Unix, lpd (Line printer daemon) de BSD était la norme et le format d'impression standard des logiciels libres classiques était PS (PostScript). Un système de filtre était utilisé en plus de Ghostscript pour permettre l'impression sur une imprimante non PostScript. Consulter la Section 11.4.1.

Dans un système Debian moderne, le système d'impression CUPS (Common UNIX Printing System) est la norme de facto, et le format d'impression standard des logiciels libres modernes est PDF (Portable Document Format).

The CUPS uses Internet Printing Protocol (IPP). The IPP is the cross-platform de facto standard for remote printing with bi-directional communication capability.

Grâce à la fonctionnalité d'autoconversion dépendante du format du fichier du système CUPS, passer simplement les données à la commande `lpr` devrait créer la sortie imprimable souhaitée. (Dans CUPS, `lpr` peut être activé en installant la paquet `cups-bsd`.)

Le système Debian possède certains paquets notables de serveurs et d'utilitaires d'impression :

paquet	popularité	taille	port	description
`lpr`	V:1.9, I:2.2	378	imprimante (515)	BSD lpr/lpd (démon d'impression)
`cups`	V:141, I:444	1102	IPP (631)	Serveur Internet d'impression CUPS
`cups-client`	V:149, I:455	432	, ,	commandes d'impression System V pour CUPS : lp(1), lpstat(1), lpoptions(1), cancel(1), lpmove(8), lpinfo(8), lpadmin(8), …
`cups-bsd`	V:33, I:178	131	, ,	commandes d'impression BSD pour CUPS : lpr(1), lpq(1), lprm(1), lpc(8)
`printer-driver-gutenprint`	V:10, I:49	1117	Non applicable	pilotes d'impression pour CUPS

Table 6.11 – Liste des serveurs et utilitaires d'impression

ASTUCE
Vous pouvez configurer le système CUPS en pointant votre navigateur web sur « http://localhost:631/ ».

6.5 Autres serveurs d'applications réseau

Voici d'autres serveurs d'applications réseau :

paquet	popularité	taille	protocole	description
inetutils-telnetd	V:0.2, I:1.3	200	TELNET	Serveur TELNET
nfs-kernel-server	V:46, I:53	886	NFS	Partage de fichiers UNIX
samba	V:112, I:123	5190	SMB	Partage de fichiers et d'imprimantes Windows
netatalk	V:0.57, I:0.79	933	ATP	Partage de fichiers et d'imprimantes Apple/Mac (AppleTalk)
proftpd-core	V:7.6, I:8.4	5041	FTP	Téléchargement généraliste de fichiers
apache2	V:168, I:205	590	HTTP	Serveur Web généraliste
squid	V:8.8, I:9.3	9585	, ,	Serveur mandataire (proxy) web généraliste
bind9	V:32, I:35	905	DNS	adresses IP des autres machines
kea	I:0.51	245	DHCP	adresse IP du client lui-même

Table 6.12 – Liste d'autres serveurs d'applications réseau

Le protocole « Common Internet File System Protocol » (CIFS) est le même protocole que Server Message Block (SMB), il est largement utilisé par Microsoft Windows.

ASTUCE
Consultez Section 4.5.2 pour l'intégration de systèmes de type serveur.

ASTUCE
La résolution de nom d'hôte est normalement fournie par le serveur DNS. Pour l'affectation dynamique d'adresse IP hôte par DHCP, le DNS dynamique peut être configuré pour la résolution de nom d'hôte en utilisant `bind9` et `kea` comme décrit sur la page DDNS du wiki Debian.

ASTUCE
L'utilisation d'un serveur mandataire tel que `squid` est bien plus efficace pour économiser de la bande passante que l'utilisation d'un serveur miroir local comportant tout le contenu de l'archive Debian.

6.6 Autres clients d'applications réseau

Voici d'autres clients d'applications réseau :

6.7 Le diagnostic des démons du système

Le programme `telnet` permet la connexion manuelle aux démons du système et leur diagnostic.

Pour tester le service POP3 brut, essayez ce qui suit :

```
$ telnet mail.ispname.net pop3
```

Pour tester le service POP3, ayant TLS/SSL activé, de certains fournisseurs d'accès Internet (FAI), vous devrez avoir un client `telnet` ayant TLS/SSL activé en utilisant l'un des paquets `telnet-ssl` ou `openssl`.

```
$ telnet -z ssl pop.gmail.com 995
```

paquet	popularité	taille	protocole	description
netcat-traditional	V:61, I:905	139	TCP/IP	couteau de l'armée Suisse pour TCP/IP
netcat-openbsd	V:23, I:118	105	TCP/IP	TCP/IP swiss army knife with support for IPv6, proxies, and Unix sockets
openssl	V:853, I:996	2564	SSL	binaire Secure Socket Layer (SSL) et outils de chiffrement associés
stunnel4	V:7, I:10	24	, ,	enrobeur SSL universel
inetutils-telnet	V:48, I:778	282	TELNET	Client TELNET
nfs-common	V:151, I:188	1259	NFS	Partage de fichiers UNIX
smbclient	V:29, I:204	2117	SMB	Client de partage de fichiers et imprimantes MS Windows
cifs-utils	V:34, I:117	369	, ,	commande de montage et de démontage de fichiers MS Windows distants
wget	V:186, I:982	3784	HTTP et FTP	téléchargement web
curl	V:296, I:739	526	, ,	, ,
transmission-gtk	V:10, I:135	6639	BitTorrent	BitTorrent client (GTK)
transmission-qt	V:0.6, I:2.5	6522	, ,	BitTorrent client (Qt)
ktorrent	V:1.3, I:4.8	5031	, ,	BitTorrent client (Qt)
qbittorrent	V:8, I:23	16666	, ,	BitTorrent client (Qt)
bind9-host	V:143, I:940	137	DNS	host(1) de bind9, « `Priority: standard` »
bind9-dnsutils	V:116, I:836	521	, ,	dig(1) de bind, « `Priority: standard` »
ldap-utils	V:11, I:56	791	LDAP	obtenir des données d'un serveur LDAP

Table 6.13 – Liste de clients d'applications réseau

```
$ openssl s_client -connect pop.gmail.com:995
```

Les RFC suivantes proposent les connaissances nécessaires pour chaque démon :

RFC	description
rfc1939 et rfc2449	service POP3
rfc3501	service IMAP4
rfc2821 (rfc821)	service SMTP
rfc2822 (rfc822)	Format de fichier de courrier électronique
rfc2045	Extensions multifonctions du courrier Internet « Multipurpose Internet Mail Extensions (MIME) »
rfc819	service DNS
rfc2616	service HTTP
rfc2396	définition d'une URI

Table 6.14 – Liste des RFC courantes

L'utilisation des ports est décrite dans « `/etc/services` ».

Chapitre 7

Système d’interface graphique

7.1 Environnement de bureau avec interface graphique

Plusieurs choix sont possibles pour un environnement de bureau graphique (GUI) complet pour un système Debian.

paquet de tâche	popularité	taille	description
task-gnome-desktop	I:191	9	environnement de bureau GNOME
task-xfce-desktop	I:88	9	environnement de bureau Xfce
task-kde-desktop	I:93	6	environnement de bureau KDE Plasma
task-mate-desktop	I:30	9	environnement de bureau MATE
task-cinnamon-desktop	I:35	9	environnement de bureau Cinnamon
task-lxde-desktop	I:18	9	environnement de bureau LXDE
task-lxqt-desktop	I:16	9	environnement de bureau LXQt
task-gnome-flashback-desktop	I:9.4	6	environnement de bureau GNOME Flashback

Table 7.1 – Liste des environnements de bureau

ASTUCE
Les paquets de dépendance sélectionnés par un métapaquet de tâche peuvent ne pas être synchronisés avec le dernier état de transition de paquet dans l'environnement Debian `unstable`/`testing`. Pour `task-gnome-desktop`, vous devrez peut-être ajuster les sélections de package comme suit :

— démarrez aptitude(8) en tant que `sudo aptitude -u` ;
— déplacez le curseur sur « Tâches » et appuyez sur « Entrée » ;
— déplacez le curseur sur « Utilisateur final », appuyez sur « Entrée » ;
— déplacez le curseur sur « GNOME » puis appuyez sur « Entrée » ;
— déplacez le curseur sur `task-gnome-desktop` puis appuyez sur « Entrée » ;
— déplacez le curseur sur « Dépend » et appuyez sur « m » (sélectionné manuellement) ;
— déplacez le curseur sur « Recommande » puis appuyez sur « m » (sélectionné manuellement) ;
— déplacez le curseur sur `task-gnome-desktop` et appuyez sur « - » (abandon) ;
— ajustez les paquets sélectionnés tout en abandonnant ceux problématiques qui provoquent des conflits de paquets ;
— appuyez sur « g » pour lancer l'installation.

Ce chapitre porte principalement sur l'environnement de bureau par défaut de Debian : `task-gnome-desktop` proposant GNOME avec wayland.

7.2 Protocole de communication graphique

Le protocole de communication graphique utilisé pour le bureau GNOME peut être :

— Wayland (protocole du serveur d'affichage) (natif) ;
— protocole central du système X Window (à l'aide de `xwayland`).

Veuillez consulter le site freedesktop.org pour savoir en quoi l'architecture de Wayland est différente de celle de X Window.

Du point de vue de l'utilisateur, les différences peuvent être résumées simplement comme suit :

— Wayland est un protocole de communication d'interface graphique pour le même hôte : nouveau, plus simple, plus rapide, sans binaire racine setuid ;
— X Window est un protocole de communication d'interface graphique compatible avec le réseau : binaire traditionnel, complexe, plus lent, setuid root.

Pour les applications utilisant le protocole Wayland, l'accès à leur contenu d'affichage à partir d'un hôte distant est pris en charge par VNC ou RDP. Consulter la Section 7.8.

Les serveurs X modernes possèdent l'extension de mémoire partagée du MIT et communiquent avec leurs clients X locaux en utilisant la mémoire partagée locale. Cela permet de contourner le canal de communication d'Xlib interprocessus transparent du réseau et de gagner en performance. Cette situation a été le contexte de la création de Wayland en tant que protocole de communication uniquement locale d'interface graphique.

En utilisant le programme `xeyes` lancé à partir du terminal GNOME, vous pouvez tester le protocole de communication d'interface graphique utilisé par chaque application graphique.

```
$ xeyes
```

— si le curseur de la souris se trouve sur une application, telle que « GNOME terminal », qui utilise le protocole de serveur d'affichage Wayland, les yeux ne se déplacent pas avec le curseur de la souris ;
— si le curseur de la souris se trouve sur une application, telle que « xterm », qui utilise le protocole de base du système X Window, les yeux se déplacent avec le curseur de la souris, révélant la nature pas si isolée de l'architecture X Window.

En avril 2021, de nombreuses applications GUI populaires telles que GNOME et LibreOffice (LO) ont migré vers le protocole de serveur d'affichage Wayland. Je constate que `xterm`, `gitk`, `chromium`, `firefox`, `gimp`, `dia` et les applications du système KDE utilisent toujours le protocole X Window.

Note
Que ce soit pour xwayland sur Wayland ou le système X Window natif, l'ancien fichier de configuration du serveur X « `/etc/X11/xorg.conf` » ne devrait pas exister sur le système. Les périphériques graphiques et d'entrée sont maintenant configurés par le noyau avec DRM, KMS et udev. Le serveur X natif a été réécrit pour les utiliser. Consulter « modedb default video mode support » dans la documentation du noyau Linux.

7.3 Infrastructure d'interface graphique

Voici des paquets d'infrastructure graphique notables pour l'environnement GNOME on Wayland.

paquet	popularité	taille du paquet	description
`mutter`	V:1, I:21	218	gestionnaire de fenêtres de GNOME mutter **[auto]**
`xwayland`	V:256, I:332	2653	serveur X fonctionnant au-dessus de wayland **[auto]**
`gnome-remote-desktop`	V:154, I:239	2743	démon de bureau à distance pour GNOME utilisant PipeWire **[auto]**
`gnome-tweaks`	V:17, I:230	1145	paramètres de configuration avancée pour GNOME
`gnome-shell-extension-prefs`	V:7, I:109	33	Outil pour activer / désactiver les extensions du GNOME Shell

Table 7.2 – Liste de paquets d'infrastructure graphique notables

Ici, "**[auto]**" signifie que ces paquets sont automatiquement installés lorsque `task-gnome-desktop` est installé.

ASTUCE
`gnome-tweaks` est l'utilitaire de configuration indispensable. Par exemple :

— vous pouvez forcer la « sur-amplification » du volume sonore à partir de « Général » ;

— vous pouvez forcer la transformation de « Caps » en « Esc » à partir de « Clavier & souris » -> « Clavier » -> «Options supplémentaires de disposition ».

ASTUCE
Les fonctionnalités détaillées de l'environnement de bureau GNOME peuvent être configurées à l'aide d'utilitaires lancés en tapant "`settings`", "`tweaks`" ou "`extensions`" après avoir appuyé sur la touche `Super-`.

7.4 Applications graphiques

De nombreuses applications graphiques utiles sont disponibles avec Debian maintenant. L'installation de paquets logiciels tels que `scribus` (KDE) dans l'environnement de bureau GNOME est tout à fait acceptable puisque les fonctionnalités correspondantes ne sont pas disponibles dans l'environnement de bureau GNOME. Mais installer trop de paquets avec des fonctionnalités dupliquées peut encombrer votre système.

Voici une liste d'applications de base qui ont attiré mon attention :

paquet	popularité	taille du paquet	type	description
evolution	V:27, I:227	521	GNOME	Gestion d'informations personnelles (logiciel de travail collaboratif et courriel)
thunderbird	V:39, I:100	304278	GTK	client de messagerie (Mozilla Thunderbird)
kontact	V:1.0, I:9.7	2416	KDE	Gestion d'informations personnelles (logiciel de travail collaboratif et courriel)
libreoffice-writer	V:92, I:417	34243	LO	traitement de texte
abiword	V:0.8, I:4.5	3596	GNOME	traitement de texte
calligrawords	V:0.3, I:2.6	7144	KDE	traitement de texte
scribus	V:1, I:12	32942	KDE	éditeur de PAO pour modifier les fichiers PDF
glabels	V:0.4, I:2.4	1295	GNOME	éditeur d'étiquettes
libreoffice-calc	V:89, I:414	29622	LO	feuille de calcul
gnumeric	V:2.8, I:9.7	10053	GNOME	feuille de calcul
calligrasheets	V:0.2, I:1.6	14014	KDE	feuille de calcul
libreoffice-impress	V:78, I:412	2501	LO	présentation
calligrastage	V:0.1, I:1.6	6237	KDE	présentation
libreoffice-base	V:18, I:65	5136	LO	gestion de base de données
kexi	V:0.04, I:0.73	7565	KDE	gestion de base de données
libreoffice-draw	V:78, I:412	11424	LO	éditeur de graphiques vectoriels (draw)
inkscape	V:12, I:73	113357	GNOME	éditeur de graphiques vectoriels (draw)
karbon	V:0.2, I:2.0	4029	KDE	éditeur de graphiques vectoriels (draw)
dia	V:1, I:16	3846	GTK	éditeur d'organigrammes et de diagrammes
gimp	V:40, I:206	33386	GTK	éditeur de graphiques en champs de bits (« bitmap ») (paint)
shotwell	V:15, I:244	6334	GTK	gestionnaire de photos numériques
digikam	V:1.5, I:8.2	333	KDE	gestionnaire de photos numériques
darktable	V:3, I:10	36092	GTK	table lumineuse et chambre noire pour photographe
gnome-video-trimmer	V:0.16, I:0.97	1665	GNOME	lossless cutting of video files
kdenlive	V:3, I:21	22233	KDE	non-linear video editor
shotcut	V:1.2, I:7.5	8488	app. QT	non-linear video editor
openshot-qt	V:0.8, I:8.3	196007	app. QT	non-linear video editor
flowblade	V:0.2, I:2.1	42213	GTK	non-linear video editor
planner	V:0.2, I:4.3	1400	GNOME	gestion de projets
calligraplan	V:0.1, I:2.9	23545	KDE	gestion de projets
gnucash	V:2.0, I:6.4	29447	GNOME	gestion financière personnelle
homebank	V:0.3, I:1.5	3277	GTK	gestion financière personnelle
kmymoney	V:0.4, I:1.8	19313	KDE	gestion financière personnelle
frescobaldi	V:0.3, I:1.4	14150	app. QT	LilyPond sheet music text editor
librecad	V:1, I:13	12361	app. QT	système de CAO (2D)
freecad	V:1, I:20	112	app. QT	système de CAO (3D)
kicad	V:3, I:14	226336	GTK	création de schémas électroniques et de circuits imprimés
xsane	V:9, I:127	1512	GTK	interface pour dispositifs de numérisation (scanner)
libreoffice-math	V:73, I:415	1980	LO	éditeur d'équations et de formules mathématiques
calibre	V:7, I:24	76289	KDE	convertisseur de livre numérique et gestion de bibliothèque
fbreader	V:0.7, I:5.2	4204	GTK	lectrice de e-book
evince	V:76, I:280	953	GNOME	afficheur de documents (pdf)
okular	V:43, I:128	4544	KDE	afficheur de documents (pdf)
x11-apps	V:32, I:440	2461	pure app. X	xeyes(1), etc.
	V:233,			

7.5 Répertoires de l'utilisateur

Les noms par défaut de répertoires de l'utilisateur, tels que « `~/Desktop` », « `~/Documents` »…, utilisés par l'environnement de bureau dépendent de la régionalisation (locale) utilisée pour l'installation du système. Ils peuvent être redéfinis en noms anglais avec :

```
$ LANGUAGE=C xdg-user-dirs-update --force
```

Ensuite toutes les données doivent être déplacées manuellement dans les nouveaux répertoires (consulter xdg-user-dirs-update(1)).

Ils peuvent aussi être définis à n'importe quel nom en éditant « `~/.config/user-dirs.dirs` » (consulter user-dirs.dirs(5)).

7.6 Fontes de caractères

De nombreuses fontes utiles à taille variable sont disponibles pour les utilisateurs de Debian. La préoccupation de l'utilisateur est de comment éviter la redondance et de comment configurer des parties de fontes installées pour leur désactivation. Sinon, les choix de fontes inutiles peuvent encombrer vos menus d'application graphique.

Le système Debian utilise la bibliothèque FreeType 2.0 pour la matricialisation de nombreux formats de fontes à taille variable pour l'écran et l'impression :

— les fontes Type 1 (PostScript) qui utilisent des courbes de Bézier cubiques (format presque obsolète) :
— les fontes TrueType qui utilisent des Courbes de Bézier quadratiques (bon choix de format) ;
— les fontes OpenType qui utilisent des courbes de Bézier cubiques (meilleur choix de format).

7.6.1 Fontes de base

Le tableau suivant est compilé dans l'espoir d'aider les utilisateurs à choisir des fontes proportionnelles appropriées avec une compréhension claire de la compatibilité métrique et de la couverture des glyphes. La plupart des fontes couvrent tous les caractères latins, grecs et cyrilliques. Le choix final des fontes activées peut également être influencé par votre esthétique. Ces fontes peuvent être utilisées pour l'affichage à l'écran ou pour l'impression sur papier.

Ici :

— « MCM » signifie « métriques compatibles avec les fontes fournies par Microsoft » ;
— « MCMATC » signifie « métrique compatible avec les fontes fournies par Microsoft : Arial, Times New Roman, Courier New » ;
— « MCAHTC » signifie « métriques compatibles avec les fontes fournies par Adobe : Helvetica, Times, Courier » ;
— les chiffres dans les colonnes de type de fonte représentent la largeur approximative relative du « M » pour une même taille de fonte ;
— le « P » dans les colonnes de type de fonte mono représente sa facilité d'utilisation pour la programmation, ayant des « 0 »/« O » et « 1 »/« I »/« l » clairement distinguables ;
— le paquet `ttf-mscorefonts-installer` télécharge les « fontes de base pour le Web » de Microsoft et installe Arial, Times New Roman, Courier New, Verdana, ... Ces données de fontes installées sont des données non libres.

De nombreuses fontes latines libres ont leur origine dans la famille URW Nimbus ou Bitstream Vera.

ASTUCE
Si vos paramètres régionaux ont besoin de fontes qui ne sont pas bien couvertes par les fontes ci-dessus, veuillez utiliser aptitude pour vérifier les paquets de tâches répertoriés sous « Tâches » -> « Régionalisation ». Les paquets répertoriés comme « Depends: » ou « Recommends: » dans les paquets de tâches de régionalisation sont les principaux candidats.

paquet	popularité	taille	sans	serif	mono	remarque
fonts-cantarell	V:173, I:284	224	59	-	-	Cantarell (GNOME 3, affichage)
fonts-noto	I:146	30	61	63	40	fontes Noto (Google, multilingue avec CJK)
fonts-dejavu	I:381	34	58	68	40	DejaVu (GNOME 2, MCM :Verdana, Bitstream Vera étendu)
fonts-liberation2	V:36, I:143	15	56	60	40	fontes Liberation pour LibreOffice (Red Hat, MCMATC)
fonts-croscore	V:19, I:35	5260	56	60	40	Chrome OS : Arimo, Tinos et Cousine (Google, MCMATC)
fonts-crosextra-carlito	V:21, I:88	2696	57	-	-	Chrome OS : Carlito (Google, MCM :Calibri)
fonts-crosextra-caladea	V:13, I:81	347	-	55	-	Chrome OS : Caladea (Google, MCM :Cambria) (Latin uniquement)
fonts-freefont-ttf	V:77, I:200	14461	57	59	40	GNU FreeFont (URW Nimbus étendu)
fonts-quicksand	V:210, I:450	392	56	-	-	task-desktop de Debian, Quicksand (affichage, Latin uniquement)
fonts-hack	V:33, I:135	2509	-	-	40 P	police de caractères conçue pour le code source de Hack (Facebook)
fonts-sil-gentiumplus	V:1, I:27	14345	-	54	-	Gentium SIL
fonts-sil-charis	V:1, I:27	6704	-	59	-	Charis SIL
fonts-urw-base35	V:187, I:533	15558	56	60	40	URW Nimbus (Nimbus Sans, Roman No. 9 L, Mono L, MCAHTC)
fonts-ubuntu	V:2.1, I:4.4	4312	58	-	33 P	fontes d'Ubuntu (affichage)
fonts-terminus	V:0.3, I:3.8	451	-	-	33	Cool retro terminal fonts
ttf-mscorefonts-installer	V:1, I:40	85	56 ?	60	40	téléchargeur de fontes non libres de Microsoft (voir ci-dessous)

Table 7.4 – Liste de fontes notables TrueType et OpenType

7.6.2 Matricialisation des fontes

Debian utilise FreeType pour le tramage des fontes. Son infrastructure de choix des fontes est fournie par la bibliothèque de configuration des fontes Fontconfig.

paquet	popularité	taille	description
libfreetype6	V:582, I:996	1098	bibliothèque de tramage des fontes FreeType
libfontconfig1	V:580, I:801	344	bibliothèque Fontconfig générique de configuration des fontes
fontconfig	V:477, I:688	415	fc-* : commandes (CLI) pour Fontconfig
font-manager	V:2.2, I:6.3	1139	Font Manager : commandes (GUI) pour Fontconfig
nautilus-font-manager	V:0.09, I:0.28	43	extension de Nautilus pour Font Manager

Table 7.5 – Liste d'environnements de fontes notables et de paquets connexes

ASTUCE
Certains paquets de fontes tels que fonts-noto* installent beaucoup trop de fontes. Vous pouvez également conserver certains paquets de fontes installés mais désactivés dans des conditions d'utilisation normales. Des glyphes multiples sont attendus pour certains points de code Unicode en raison de l'Unification Han et des glyphes indésirables peuvent être choisis par la bibliothèque Fontconfig non configurée. L'un des cas les plus ennuyeux est « U+3001 IDEOGRAPHIC COMA » et « U+3002 IDEOGRAPHIC FULL STOP » pour les pays CJK. Vous pouvez éviter facilement cette situation problématique en configurant la disponibilité des fontes à l'aide de l'interface graphique du gestionnaire de fontes (font-manager).

Vous pouvez aussi vérifier les informations de configuration des fontes à partir de la ligne de commande comme suit :

— « fc-match(1) » pour la fonte par défaut de fontconfig ;
— « fc-list » pour les fontes disponibles de fontconfig.

Vous pouvez paramétrer l'état de configuration des fontes à partir d'un éditeur de texte, mais cela n'est pas trivial. Consulter fonts.conf(5).

7.7 Bac à sable

De nombreuses applications sous Linux, principalement basées sur une interface graphique, sont disponibles dans des formats binaires à partir de sources autres que Debian :

— AppImage – applications Linux qui s'exécutent partout ;
— FLATHUB – applications pour Linux, fonctionnant pour nous ;
— snapcraft – magasin d'applications pour Linux .

AVERTISSEMENT
Les binaires de ces sites peuvent inclure des logiciels propriétaires non libres.

Il existe quelques raisons d'être pour ces distributions au format binaire pour les aficionados du logiciel libre qui utilisent Debian, car elles peuvent accueillir un ensemble sain de bibliothèques utilisées pour chaque application par leur développeur amont, indépendamment de celles fournies par Debian.

Le risque inhérent à l'exécution de fichiers binaires externes peut être réduit en utilisant l'environnement sandbox (bac à sable) qui exploite les fonctionnalités de sécurité Linux modernes (consulter laSection 4.7.5) :

— pour les binaires provenant d'AppImage et de certains sites amont, exécutez-les dans firejail avec une configuration manuelle ;
— pour les binaires de FLATHUB, exécutez-les dans Flatpak (aucune configuration manuelle requise) ;
— pour les fichiers binaires de snapcraft, exécutez-les dans Snap (aucune configuration manuelle requise, compatible avec les programmes démon).

Le paquet `xdg-desktop-portal` fournit une API normalisée pour les fonctionnalités de bureau communes. Consulter xdg-desktop-portal (flatpak) et xdg-desktop-portal (snap) .

paquet	popularité	taille	description
`flatpak`	V:108, I:116	9297	cadriciel de déploiement d'application Flatpak pour les applications de bureau
`gnome-software-plugin-flatpak`	V:28, I:39	291	prise en charge de Flatpak pour les logiciels de GNOME
`snapd`	V:56, I:59	62738	démon et outils permettant d'activer les paquets snap
`gnome-software-plugin-snap`	V:1.4, I:2.0	144	prise en charge de Snap par GNOME Software
`xdg-desktop-portal`	V:376, I:438	2291	portail d'intégration au bureau pour Flatpak et Snap
`xdg-desktop-portal-gtk`	V:340, I:436	715	dorsal de xdg-desktop-portal pour gtk (GNOME)
`xdg-desktop-portal-kde`	V:81, I:111	3333	dorsal de xdg-desktop-portal pour Qt (KDE)
`xdg-desktop-portal-wlr`	V:2.4, I:6.7	167	dorsal de xdg-desktop-portal pour wlroots (Wayland)
`firejail`	V:1.1, I:4.2	1725	programme de bac à sable de sécurité SUID firejail pour utiliser avec AppImage

Table 7.6 – Liste des environnements bac à sable notables et des packages connexes

Cette technologie d'environnement de bac à sable (sandbox) ressemble beaucoup aux applications sur les systèmes d'exploitation d'ordiphone où elles sont exécutées en ayant des accès contrôlés aux ressources.

Certaines grosses applications graphiques, telles que les navigateurs web, sur Debian utilisent également la technologie d'environnement de bac à sable en interne pour les rendre plus sécurisées.

7.8 Bureau à distance

7.9 Connexion au serveur X

Il existe plusieurs façons de se connecter au serveur X à partir d'une application sur un hôte distant, dont `xwayland` sur l'hôte local.

7.9.1 Connexion locale au serveur X

L'accès au serveur X local par les applications locales qui utilisent le protocole de base X peut être réalisé localement à l'aide d'un socket local de domaine UNIX. Cela peut être autorisé par le fichier d'authentification contenant les cookies d'accès. L'emplacement du fichier d'authentification est déterminé par la variable d'environnement « `$XAUTHORITY` » et l'affichage X est identifié par la variable d'environnement « `$DISPLAY` ». Étant donné que ceux-ci sont normalement définis automatiquement, aucune action spéciale n'est nécessaire, par exemple « `gitk` » comme suit.

```
username $ gitk
```

paquet	popularité	taille	protocoles	description
gnome-remote-desktop	V:15, I:239	2743	RDP	serveur GNOME Remote Desktop
xrdp	V:29, I:31	4757	RDP	xrdp, serveur Remote Desktop Protocol (RDP)
x11vnc	V:10, I:48	1863	RFB (VNC)	x11vnc, serveur Remote Framebuffer Protocol (VNC)
tigervnc-standalone-server	V:6, I:15	3176	RFB (VNC)	TigerVNC, serveur Remote Framebuffer Protocol (VNC)
gnome-connections	V:7, I:154	1726	RDP, RFB (VNC)	client de bureau distant pour GNOME
remmina	V:16, I:59	1046	RDP, RFB (VNC), SPICE, SSH, ...	Remmina : client de bureau à distance en GTK
krdc	V:2, I:15	4144	RDP, RFB (VNC)	KRDC : client de bureau à distance de KDE
virt-viewer	V:5, I:36	1278	RFB (VNC), SPICE	client d'affichage graphique de Virtual Machine Manager du système d'exploitation invité

Table 7.7 – Liste des serveurs et des utilitaires notables d'accès à distance

paquet	popularité	taille	commande	description
openssh-server	V:781, I:824	3615	`sshd` avec l'option `X11-forwarding`	serveur SSH (sécurisé)
openssh-client	V:709, I:997	3611	`ssh -X`	client SSH (sécurisé)
xauth	V:197, I:975	81	`xauth`	utilitaire de fichier d'authentification X
x11-xserver-utils	V:376, I:532	559	`xhost`	contrôle d'accès au serveur pour X

Table 7.8 – Liste des méthodes de connexion au serveur X

Note
Pour `xwayland`, `XAUTHORITY` contient une valeur comme « `/run/user/1000/.mutter-Xwaylandauth.YVSU30` ».

7.9.2 Connexion à distance au serveur X

L'accès à l'écran du serveur X local à partir des applications distantes qui utilisent le protocole de base de X est pris en charge par la fonction de transfert X11 :

— ouvrez un `xterm` sur la machine locale ;
— lancez ssh(1) avec `-X` pour établir une connexion avec un site distant comme suit :

```
localname @ localhost $ ssh -q -X loginname@remotehost.domain
Password:
```

— exécutez une commande d'application X, par exemple « `gitk` », sur le site distant comme suit :

```
loginname @ remotehost $ gitk
```

Cette méthode permet l'affichage du client X distant comme s'il était connecté par une socket UNIX locale.

Consulter la Section 6.3 pour SSH/SSHD.



AVERTISSEMENT
Une connexion distante TCP/IP au serveur X est désactivée par défaut sur un système Debian pour des raisons de sécurité. Ne les activez pas en définissant simplement « `xhost +` » ni en activant connexion XDMCP, si vous pouvez l'éviter.

7.9.3 Connexion avec chroot au serveur X

L'accès au serveur X par les applications qui utilisent le protocole X de base et s'exécutent sur le même hôte mais dans un environnement tel que chroot où le fichier d'authentification n'est pas accessible, peut être autorisé de manière sécurisée avec `xhost` en utilisant l'accès basé sur l'utilisateur , par exemple « `gitk` » comme suit :

```
username $ xhost + si:localuser:root ; sudo chroot /path/to
# cd /src
# gitk
# exit
username $ xhost -
```

7.10 Presse-papier

Pour copier du texte dans le presse-papiers, consulter la Section 1.4.4.

Pour copier des graphismes dans le presse-papiers, consulter la Section 11.6.

Certaines commandes en ligne de commande peuvent également manipuler le presse-papiers (PRIMARY et CLIPBOARD).

paquet	popularité	taille du paquet	cible	description
`xsel`	V:8, I:42	55	X	interface en ligne de commande pour les sélections X (presse-papiers)
`xclip`	V:19, I:78	62	X	interface en ligne de commande pour les sélections X (presse-papiers)
`wl-clipboard`	V:11, I:28	174	Wayland	`wl-copy` `wl-paste` : interface en ligne de commande pour le presse-papiers de Wayland
`gpm`	V:8.0, I:8.7	524	console Linux	démon qui capture les événements de souris sur la console Linux

Table 7.9 – Liste de programmes en rapport avec la manipulation du presse-papiers « caractères »

Chapitre 8

I18N et L10N

Le multilinguisme (M17N) ou la gestion de la langue natale (« Native Language Support ») d'un logiciel applicatif est réalisé en deux étapes :

— L'internationalisation (I18N) : donne la possibilité à un logiciel de gérer plusieurs paramètres linguistiques ;
— La localisation (L10N) : permet au logiciel de prendre en charge des paramètres linguistiques particuliers.

ASTUCE
Il y a 17, 18, ou 10 lettres entre « m » et « n », « i » et « n » ou « l » et « n » dans « multilingualization », « internationalization » et « localization » ce qui correspond à M17N, I18N et L10N. Consulter Internationalisation et localisation pour plus de détails.

8.1 Les paramètres linguistiques (« locale »)

Le comportement des programmes prenant en charge l'internationalisation est configuré par la variable d'environnement « `$LANG` » pour prendre en charge la régionalisation. La prise en charge effective des fonctionnalités dépendantes des paramètres régionaux par la bibliothèque `libc` nécessite l'installation des paquets `locales` ou `locales-all`. Le paquet `locales` doit être initialisé correctement.

Si ni le paquet `locales` ni le paquet `locales-all` n'est installé, la prise en charge des fonctionnalités régionales est impossible et le système utilise des messages en anglais américain et gère les données en tant que **ASCII**. Ce comportement est le même que si « `$LANG` » est défini par « `LANG=` », « `LANG=C` » ou « `LANG= POSIX` ».

Des logiciels modernes, tels que GNOME et KDE, gèrent le multilinguisme. Ils sont internationalisés en les faisant gérer les données UTF-8 et régionalisés en leur donnant les messages traduits par l'intermédiaire de l'infrastructure gettext(1). Les messages traduits peuvent être fournis sous forme de paquets de régionalisation séparés.

Le système actuel d'interface graphique du bureau Debian définit normalement les paramètres régionaux pour l'environnement graphique comme « `LANG=xx_YY.UTF-8` ». Ici, « `xx` » correspond aux codes de langue ISO 639 et « YY » correspond aux codes de pays ISO 3166. Ces valeurs sont définies par la boîte de dialogue de configuration de l'interface graphique de bureau et modifient le comportement du programme. Consulter la Section 1.5.2.

8.1.1 Justification de l'utilisation d'UTF-8 dans les paramètres linguistiques

La représentation la plus simple des données textuelles est en **ASCII**, ce qui est suffisant pour l'anglais et utilise moins de 127 caractères (représentables avec 7 bits).

Même le texte anglais en texte brut peut contenir des caractères non-ASCII, par exemple les guillemets apostrophes culbutés gauches et droits ne sont pas disponibles en ASCII.

```
b''“b''double quoted textb''”b'' is not "double quoted ASCII"
b''‘b''single quoted textb''’b'' is not 'single quoted ASCII'
```

Afin de prendre en charge un plus grand nombre de caractères, de nombreux jeux de caractères et systèmes de codage ont été utilisés pour prendre en charge beaucoup de langues (consulter le Tableau 11.2).

Le jeu de caractères Unicode peut représenter pratiquement tous les caractères humainement connus avec une plage de points de code de 21 bits (c'est-à-dire de 0 à 10FFFF en notation hexadécimale).

Le système de codage de texte UTF-8 adapte les points de code Unicode dans un flux de données pratique de 8 bits grandement compatible avec le système de traitement de données ASCII. Cela fait de **UTF-8** le choix moderne privilégié. **UTF** signifie Unicode Transformation Format. Lorsque les données en texte brut ASCII sont converties en données UTF-8, elles ont exactement le même contenu et la même taille que l'ASCII originel. Ainsi, vous ne perdez rien en déployant les paramètres régionaux UTF-8.

Sous les paramètres régionaux UTF-8 avec le programme d'application compatible, vous pouvez afficher et modifier toutes les données textuelles en langue étrangère tant que les fontes et les méthodes de saisie requises sont installées et activées. Par exemple avec le paramètre « `LANG=fr_FR. UTF-8` », gedit(1) (éditeur de texte pour le bureau GNOME) peut afficher et modifier des données de texte en caractères chinois tout en présentant des menus en français.

ASTUCE
Le nouveau paramètre régional standard "`en_US.UTF-8`" et celui ancien standard « `C`"/"`POSIX` » utilisent les messages standard en anglais américain. Ils ont des différences subtiles dans l'ordre de tri, etc. Si vous souhaitez gérer non seulement les caractères ASCII, mais également tous les caractères encodés en UTF-8 élégamment tout en conservant l'ancien comportement régional « `C` », utilisez le paramètre « `C.UTF-8` » non standard avec Debian.

Note
Certains programmes utilisent davantage de mémoire lors de l'utilisation de I18N. Cela parce qu'ils sont codés avec l'utilisation interne d'UTF-32(UCS4) pour la prise en compte d'Unicode afin d'optimiser la vitesse, ils utilisent 4 octets pour chaque caractère ASCII indépendamment de la « locale » sélectionnée. De nouveau, il n'y a rien à perdre en mettant en œuvre des paramètres linguistiques UTF-8.

8.1.2 Reconfiguration des paramètres linguistiques

Pour que le système puisse accéder à un paramètre régional particulier, les données de paramètres régionaux doivent être compilées à partir de la base de données de paramètres régionaux.

Le paquet `locales` n'est **pas** fourni avec des données de paramètres régionaux précompilées. Vous devez le configurer comme suit :

```
# dpkg-reconfigure locales
```

Ce processus se déroule en deux étapes :

1. Sélectionnez toutes les données de paramètres régionaux requis à compiler dans une forme binaire (assurez-vous d'inclure au moins un paramètre régional UTF-8) ;
2. Définissez la valeur des paramètres linguistiques par défaut pour l'ensemble du système dans « `/etc/default/loca` pour une utilisation par PAM (consulter la Section 4.5).

La valeur des paramètres régionaux par défaut pour l'ensemble du système définie dans « `/etc/default/locale` » peut être remplacée par la configuration de l'interface graphique pour les applications graphiques.

Note
Le système d'encodage traditionnel réel peut être identifié par « `/usr/share/i18n/SUPPORTED` ». Ainsi, « `LANG=en_US` » est « `LANG=en_US. ISO-8859-1` ».

Le paquet `locales-all` est livré avec les données de paramètres régionaux pré-compilées pour toutes les données de paramètres régionaux. Comme il ne crée pas « `/etc/default/locale` », vous devrez peut-être encore installer le paquet `locales`.

ASTUCE
Le paquet `locales` de certaines distributions dérivées de Debian sont livrées avec des données pré-compilées pour tous les paramètres régionaux existants. Vous devez installer les deux paquets `locales` et `locales-all` dans Debian pour imiter cet environnement de système.

8.1.3 Coder les noms de fichiers

Pour les échanges de données entre plateformes (consultez Section 10.1.7), il vous faudra peut-être monter certains systèmes de fichiers ayant un codage particulier. Par exemple, `, la commande mount`(8) pour un système de fichiers vfat suppose que l'on utilise CP437 si on l'utilise sans option. Vous devrez fournir les options explicites à mount pour utiliser des noms de fichiers codés en UTF-8 ou en CP932.

Note
When auto-mounting a hot-pluggable USB flash drive under modern desktop environment such as GNOME, you may provide such mount option by right clicking the icon on the desktop, click ”Drive” tab, click to expand ”Setting”, and entering ”utf8” to ”Mount options:”. The next time this USB flash drive is mounted, mount with UTF-8 is enabled.

Note
Si vous êtes en train de mettre à jour le système ou de déplacer des disques depuis un ancien système qui n'était pas UTF-8, les noms de fichiers avec des caractères non ASCII peuvent être codés avec des codages historiques et obsolètes tels que ISO-8859-1 ou eucJP. Veuillez consulter l'aide des outils de conversion de texte pour les convertir en UTF-8. Consultez Section 11.1.

Samba utilise Unicode pour les clients les plus récents (Windows NT, 200x, XP) mais utilise par défaut CP850 pour des clients plus anciens (DOS et Windows 9x/Me). Cette valeur par défaut pour les anciens clients peut être modifiée en utilisant « `dos charset` » dans le fichier « `/etc/samba/smb.conf` », par exemple, avecCP932 pour le japonais.

8.1.4 Messages et documentation traduits

Il existe des traductions de nombreux messages et documents affichés par le système Debian, comme les messages d'erreur, la sortie standard des programmes,les menus et les pages de manuel. La GNU gettext(1) command tool chain est utilisé comme outil de base pour la plupart des activités de traduction.

Dans « Tâches » → « Localisation » aptitude(8) fournit une liste exhaustive de paquets binaires utiles qui ajoutent les traductions de messages aux applications et fournissent de la documentation traduite.

Vous pouvez, par exemple, obtenir les messages traduits pour une page de manuel en installant le paquet `manpages-`*`LANG`*. Pour lire la page de manuel de *nom_programme* en italien depuis « `/usr/share/man/it/` », lancez le programme de la manière suivante :

```
LANG=it_IT.UTF-8 man programname
```

GNU gettext peut s'adapter à la liste de priorité des langues de traduction avec la variable d'environnement `$LANGUAGE`. Par exemple :

```
$ export LANGUAGE="pt:pt_BR:es:it:fr"
```

Pour en savoir plus, consultez `info gettext` et lisez la section « The LANGUAGE variable ».

8.1.5 Effet des paramètres linguistiques

L'ordre de tri des caractères par sort(1) et ls(1) est affecté par la régionalisation. L'export `LANG=en_US.UTF-8` trie dans l'ordre du dictionnaire `A->a->B->b...->Z->z`, tandis que l'export `LANG=C.UTF-8` trie dans l'ordre binaire ASCII `A->B->...->Z->a->b...`.

Le format de date de ls(1) est affecté par les paramètres linguistiques (consultez Section 9.3.4).

Le format de date(1) est affecté par la régionalisation. Par exemple :

```
 $ unset LC_ALL
 $ LANG=en_US.UTF-8 date
Thu Dec 24 08:30:00 PM JST 2023
 $ LANG=en_GB.UTF-8 date
Thu 24 Dec 20:30:10 JST 2023
 $ LANG=es_ES.UTF-8 date
jue 24 dic 2023 20:30:20 JST
 $ LC_TIME=en_DK.UTF-8 date
2023-12-24T20:30:30 JST
```

La ponctuation des nombres est différente selon les régions. Par exemple, dans la langue anglaise, mille virgule un est affiché comme « `1,000.1` » alors que dans la langue allemande, il est affiché comme « `1.000,1` ». Vous pouvez constater cette différence dans un tableur.

Chaque caractéristique détaillée de la variable d'environnement « `$LANG` » peut être remplacée en réglant les variables à « `$LC_*` ». Ces variables d'environnement peuvent être remplacées à nouveau par la variable réglée à « `$LC_ALL` ». Consultez la page de manuel locale(7) pour plus de détails. À moins que vous n'ayez de bonnes raisons de créer une configuration compliquée, n'utilisez pas ces variables et utilisez uniquement la variable « `$LANG` » définie à un des paramètres régionaux UTF-8.

8.2 L'entrée clavier

The keyboard system can be configured at different layers of the system.

— Linux kernel: keyboard(5)
— X server: setxkbmap(1), xkeyboard-config(5), environment variable `XMODIFIERS`
— GUI desktop environment: Input Method framework: `ibus`, `fcitx5`
— Application: environment variables to set its input source: `GTK_IM_MODULE`, `QT_IM_MODULE`, `QT_IM_MODULES`, ...

Input method framework (IM) consists of:

— Input method engine (IME): Actual input method
— Configuration: Handles the configuration for IBus and other services such as IME plugins
— Panel: User interface such as language bar and candidate selection table

Multilingual input to the application is processed roughly as:

```
Keyboard        UI panel    Configuration          Application
|                  ^ |           |                       ^  ^
v                  | v           v                       |  |
Linux kernel -> Input method engine (IME) -+-> Gtk, Qt ---+  |
                  | ^                     |                 |
                  | |                     +-> X11, Wayland -+
                  v |
                IME plugin (ibus-mozc, ...)
```

8.2.1 La saisie avec le clavier pour la console Linux et X Window

Le système Debian peut être configuré pour travailler avec de nombreuses configurations de clavier internationales en utilisant le paquet `keybord-configuration`.

```
# dpkg-reconfigure keyboard-configuration
```

Pour la console Linux et le système X Window, cela met à jour les paramètres de configuration dans « `/etc/default/keyb`
De nombreux caractères non ASCII, y compris les caractères accentués utilisés par de nombreuses langues européennes, peuvent être disponibles avec les touches mortes, la touche AltGr et la touche compose.

Note
Si ibus est actif, votre configuration de clavier X classique avec `setxkbmap` peut être remplacée par `ibus`, même dans un environnement de bureau classique basé sur X. Vous pouvez désactiver `ibus` si installé à l'aide de `im-config` pour définir la méthode de saisie à « None ». Pour en savoir plus, consultez le wiki de Debian concernant le clavier.

8.2.2 La saisie avec le clavier pour Wayland

Unlike the X Window protocol, the Wayland core protocol doesn't even support the input of accented characters. Popular Wayland Compositors, such as Mutter for GNOME or KWin for KDE, implement extension protocols such as the `text-input-unstable-v3` for the text input (see "current Wayland protocols and their support status").

The `text-input-unstable-v3` protocol works well with Input methods for Wayland (see "Wayland input method project post-mortem").

— Most GUI applications are built with GUI libraries such as GTK or Qt which support this `text-input-unstable-v3`.
— Popular Input Method Engines (IME), such as IBus or Fcitx (version 5), can work with this `text-input-unstable-v3`.
— IMEs support text input for many languages with plugins.
— Recent IMEs integrate "X Keyboard Extension (XKB)" functionalities such as `setxkbmap` previously provided by the X Window to support accented character text input for European languages for Wayland.

8.2.3 Prise en charge de la méthode d'entrée avec iBus

For GNOME, "`ibus`" is the default IME which is automatically installed via its package dependency.

— Most multilingualized keyboard input features can be configured from "GNOME Settings" or "GNOME Tweaks".
— Some multilingualized keyboard input features may need to be configured from the ibus-setup(1) command.
— The emoji keyboard input is available by typing "`SUPER-.`" (Simultaneously type Windows and period keys) followed by a keyword for each emoji and SPACE-keys.

The list of IBus and its plugin packages are the following.

8.2.4 The input method support with Fcitx

The Fcitx (version 5) input method framework is popular with Chinese users and compatible with "`ibus`".

The list of "`fcitx5`" and its plugin packages are the following.

paquet	popularité	taille	paramètres linguistiques pris en charge
ibus	V:211, I:252	1884	infrastructure de méthode d'entrée utilisant dbus
ibus-mozc	V:2.0, I:3.7	872	Japonais
ibus-anthy	V:0.45, I:0.98	8740	Japonais
ibus-skk	V:0.04, I:0.15	242	Japonais
ibus-kkc	V:0.04, I:0.18	206	Japonais
ibus-libpinyin	V:1.0, I:4.7	125	Chinois (pour zh_CN)
ibus-chewing	V:0.11, I:0.73	298	Chinois (pour zh_TW)
ibus-libzhuyin	I:0.08	41021	Chinois (pour zh_TW)
ibus-rime	V:0.31, I:0.50	76	Chinois (pour zh_CN/zh_TW)
ibus-cangjie	V:0.01, I:0.09	235	Chinois (pour zh_HK)
ibus-hangul	V:0.3, I:1.9	261	Coréen
ibus-libthai	V:0.00, I:0.05	84	Thaï
ibus-table-thai	I:0.05	59	Thaï
ibus-unikey	V:0.21, I:0.42	286	Vietnamien
keyman	V:0.01, I:0.09	507	Multilingue : moteur Keyman pour plus de 2 000 langues
ibus-table	V:0.06, I:0.85	2482	moteur de tables pour iBus
ibus-m17n	V:0.2, I:1.7	373	Multilingue : Indien, Arabe et autres

Table 8.1 – List of IBus and its plugin packages

paquet	popularité	taille	paramètres linguistiques pris en charge
fcitx5	V:8, I:12	712	input method framework compatible with "`ibus`"
fcitx5-mozc	V:1.1, I:1.8	1578	Japonais
fcitx5-anthy	V:0.06, I:0.24	853	Japonais
fcitx5-skk	V:0.08, I:0.16	371	Japonais
fcitx5-kkc	V:0.00, I:0.07	418	Japonais
fcitx5-chinese-addons	I:8.7	17	Chinois (métapaquet pour zh_*)
fcitx5-pinyin	V:4.2, I:9.4	1000	Chinois (pour zh_CN)
fcitx5-chewing	V:0.29, I:0.92	228	Chinois (pour zh_TW)
fcitx5-zhuyin	I:0.07	108535	Chinois (pour zh_TW)
fcitx5-rime	V:0.57, I:0.99	393	Chinois (pour zh_CN/zh_TW)
fcitx5-table-cangjie-large	I:0.12	1292	Chinois (pour zh_HK)
fcitx5-hangul	V:0.13, I:0.28	239	Coréen
fcitx5-libthai	I:0.04	119	Thaï
fcitx5-table-thai	I:0.07	34	Thaï
fcitx5-unikey	V:0.13, I:0.26	624	Vietnamien
fcitx5-m17n	V:0.41, I:0.85	171	Multilingue : Indien, Arabe et autres
fcitx5-table	V:0.3, I:9.2	428	moteur de tables pour fcitx5
fcitx5-keyman	V:0.04, I:0.05	241	Multilingue : moteur Keyman pour plus de 2 000 langues

Table 8.2 – Liste des paquets Fcitx5 et de ses moteurs

8.2.5 Un exemple pour le japonais

I find the Japanese input method started under English environment (”`en_US.UTF-8`”) very useful. Here is how I did this with IBus:

1. Install the Japanese input tool package `ibus-mozc` (or `ibus-anthy`).
2. — Generic: Execute ibus-setup(1) → select ”Input Method” → click ”Add” → ”Japanese” → ”Mozc (or Anthy)” → click ”Add”
   — GNOME: Select ”Settings” → ”Keyboard” → ”Input Sources” → click ”+” in ”Input Sources” → ”Japanese” → ”Mozc (or Anthy)” → click ”Add”
3. Vous pouvez choisir autant de « Input Sources » que vous le souhaitez ;
4. Reconnectez-vous au compte utilisateur.
5. Configurez chaque « Input Source » avec un clic droit sur l’icône de la barre d’outils de l’interface graphique ;
6. Choisissez parmi les sources d’entrée installées avec SUPER+ESPACE. (en général, SUPER désigne la touche Windows)

ASTUCE
Si vous souhaitez avoir accès à l’environnement de clavier alphabétique uniquement avec le clavier japonais physique sur lequel shift-2 a " (guillemets doubles) gravé, vous sélectionnez « Japonais » dans la procédure ci-dessus. Vous pouvez entrer japonais en utilisant « Japanese mozc (ou anthy) » avec le clavier physique « US » sur lequel shift-2 a @ (marque arobase) gravé.

— For Wayland:
  — The `im-config` package does nothing and can be removed safely.
  — You probably don’t need to set environment variables except for the backward compatibility etc.
  — If you need to set environment variables, create a file such as ”`~/.config/environment.d/50-input-method.c`” to set them.
— For X Window:
  — Install the `im-config` package.
  — L’entrée du menu graphique pour im-config(8) est « Input method ».
  — Alternativement, exécutez « `im-config` ” à partir de l’interpréteur de commande de l’utilisateur.
  — im-config(8) se comporte différemment selon que la commande est exécutée depuis le compte de l’administrateur ou non.
  — im-config(8) active la meilleure méthode de saisie sur le système par défaut sans intervention de l’utilisateur.

8.3 L’affichage de sortie

La console Linux ne peut afficher qu’un nombre restreint de caractères. Vous devez avoir un programme de terminal particulier tel que jfbterm(1) pour afficher les langues non européennes sur des consoles autres que la console graphique.

L’environnement graphique (Chapitre 7) peut afficher n’importe quel caractère en UTF-8 tant que les fontes requises soient installées et activées. L’encodage des données de la fonte originelle est pris en compte et est transparent pour l’utilisateur.

8.3.1 Configuration du terminal

Le système Debian peut être configuré pour travailler avec de nombreuses configurations de clavier internationales en utilisant le paquet `console-setup`.

```
# dpkg-reconfigure console-setup
```

For the Linux console and the X Window system, this updates configuration parameters in "/etc/default/console-setu
to display many non-ASCII characters including accented characters used by many European languages can be made available.

Il existe plusieurs composants pour configurer la console en mode caractères et les fonctionnalités du système ncurses(3).

— Le fichier « /etc/terminfo/*/* » (terminfo(5))
— La variable d'environnement « $TERM »(term(7))
— setterm(1), stty(1), tic(1) et toe(1)

Si l'entrée terminfo pour xterm ne fonctionne pas avec un xterm non Debian, changez le type de terminal dans « $TERM » de « xterm » pour une version limitée en fonctionnalités comme « xterm-r6 » lorsque vous-vous connectez à distance à un système Debian. Consultez « /usr/share/doc/ncurses-base/FAQ » pour davantage d'informations. « dumb » est le plus petit dénominateur commun pour « $TERM ».

8.3.2 Largeur des caractères ambigus d'Asie orientale

Avec les paramètres régionaux d'Asie de l'Est, le dessin de la boîte, les caractères grecs et cyrilliques peuvent être affichés plus larges que la largeur souhaitée et provoquer une sortie sur le terminal non alignée (consulter Unicode Standard Annex #11, 4.2 Ambiguous Characters).

Vous pouvez contourner ce problème :

— gnome-terminal : Préférences → Profils → *Nom du profil (Sans nom)* → Compatibilité → Caractères de largeur ambigüe → Fins ;
— ncurses : paramètre l'environnement export NCURSES_NO_UTF8_ACS=0.

Chapitre 9

Astuces du système

Je décris ici les astuces de base pour configurer et gérer les systèmes, la plupart depuis la console.

9.1 Conseils pour la console

Quelques programmes utilitaires existent pour vous aider dans vos activités de console.

paquet	popularité	taille	description
mc	V:42, I:172	1590	consulter la Section 1.3
bsdutils	V:445, I:1000	338	script(1) commande permettant d'enregistrer une session de terminal
screen	V:49, I:187	1007	multiplexeur de terminal avec une émulation de terminal VT100/ANSI
tmux	V:114, I:172	1653	alternative de terminal multiple (utiliser « Control-B » à la place)
ripgrep	V:17, I:49	5284	recherche récursive rapide de chaines dans l'arborescence de code source avec filtrage automatique
fzf	V:10, I:37	5692	recherche approximative de texte
fzy	V:0.05, I:0.31	59	recherche approximative de texte
rlwrap	V:1, I:11	328	enveloppe de ligne de commande de la fonctionnalité readline
ledit	V:0.4, I:7.1	401 401	enveloppe de ligne de commande de la fonctionnalité readline
rlfe	V:0.03, I:0.49	45	enveloppe de ligne de commande de la fonctionnalité readline

Table 9.1 – Liste des programmes de prise en charge d'activités avec une console

9.1.1 Enregistrer proprement l'activité de la console

La simple utilisation de script(1) (consultez Section 1.4.9) pour enregistrer l'activité de l'interpréteur de commandes produit un fichier avec des caractères de contrôle. Cela peut être évité en utilisant col(1) comme suit :

```
$ script
Script started, file is typescript
```

faites quelque chose… et pressez `Ctrl-D` pour quitter `script`.

```
$ col -bx < typescript > cleanedfile
$ vim cleanedfile
```

Il existe d'autres méthodes pour enregistrer les activités des interpréteurs de commandes :

— utilisation de `tee` (utilisable pendant le processus de démarrage dans l'initramfs) :

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

— utilisation du `gnome-terminal` avec la mémoire tampon de lignes étendue pour le défilement ;
— utilisation de `screen` avec « `^A H` » (consulter la Section 9.1.2) pour effectuer l'enregistrement de la console ;
— utilisation de `vim` avec « `:terminal` » pour entrer en mode terminal (utiliser « `Ctrl-W N` » pour quitter le mode terminal vers le mode normal et « `:w nom_fichier` » pour écrire le tampon dans un fichier ;
— utilisation d'`emacs` avec « `M-x shell` », « `M-x eshell` » ou « `M-x term` » pour accéder à la console d'enregistrement (utiliser « `C-x C-w` » pour écrire la mémoire tampon dans un fichier).

9.1.2 Le programme screen

screen(1) ne permet pas uniquement de faire tourner plusieurs processus dans une fenêtre de terminal, mais aussi à un **processus de l'interpréteur de commandes distant de survivre à d'éventuelles interruptions de la connexion**. Voici un scénario typique de screen(1) :

1. vous-vous connectez à une machine distante ;
2. vous démarrez `screen` sur une seule console ;
3. vous exécutez plusieurs programmes dans les fenêtres `screen` créées avec `^A c` (« Ctrl-A » suivi de « c ») ;
4. vous passez d'une des fenêtres multiples de `screen` à l'autre avec `^A n` (« Ctrl-A » suivi de « n ») ;
5. vous avez alors besoin de quitter votre terminal, mais vous ne voulez pas perdre votre travail en cours perdant la connexion.
6. vous pouvez **détacher** la session `screen` par différentes méthodes :
   — débrancher brutalement votre connexion réseau ;
   — entrer `^A d` (« Ctrl-A » suivi de « d ») et en quittant manuellement la connexion distante :
   — entrer `^A DD` (« Ctrl-A » suivi de « DD ») pour détacher `screen` et vous déconnecter.
7. Vous vous reconnectez à la même machine distante (même depuis un autre terminal) ;
8. Vous lancez `screen` avec « `screen -r` » ;
9. `screen` **réattache** magiquement toutes les fenêtres `screen` précédentes avec tous les programmes qui y tournent.

ASTUCE
Avec `screen`, vous pouvez économiser des frais de connexion pour les connexions limitées, telles que les connexion commutées ou par paquets, parce que vous laissez un processus actif alors que vous êtes déconnecté. Vous pouvez le ré-attacher plus tard, lorsque vous vous reconnectez.

Dans une session `screen`, toutes les entrées clavier sont envoyées vers votre fenêtre actuelle sauf les séquences de touche de commande. Toutes les séquences de touche de commande `screen` sont entrées par `^A` (« Ctrl-A ») suivi d'un seule touche [plus les paramètres]. Voici celles dont il est important de se souvenir.

Consultez screen(1) pour davantage d'informations.

Consulter tmux(1) pour les fonctionnalités de la commande alternative.

affectation	signification
^A ?	afficher l'aide de screen (afficher les raccourcis clavier)
^A c	créer une nouvelle fenêtre et basculer vers celle-ci
^A n	aller à la fenêtre suivante
^A p	aller à la fenêtre précédente
^A 0	aller à la fenêtre 0
^A 1	aller à la fenêtre 1
^A w	afficher la liste des fenêtres
^A a	envoyer un Ctrl-A à la fenêtre actuelle en tant qu'entrée clavier
^A h	écrire dans un fichier une copie de la fenêtre actuelle
^A H	commencer et finir l'enregistrement de la fenêtre en cours vers un fichier
^A ^X	verrouiller le terminal (protégé par un mot de passe)
^A d	détacher la session screen du terminal
^A DD	détacher la session screen et se déconnecter

Table 9.2 – Liste des raccourcis clavier de screen

9.1.3 Navigation dans les répertoires

Dans la Section 1.4.2, deux astuces pour permettre une navigation rapide dans les répertoires sont décrites : `$CDPATH` et `mc`.

Si vous utilisez un programme de recherche approximative de texte, vous pouvez le faire sans taper le chemin exact. Pour `fzf`, incluez ce qui suit dans votre fichier `~/.bashrc` :

```
FZF_KEYBINDINGS_PATH=/usr/share/doc/fzf/examples/key-bindings.bash
FZF_COMPLETION_PATH=/usr/share/doc/fzf/examples/completion.bash
if [ -f $FZF_KEYBINDINGS_PATH ] && [ -f $FZF_COMPLETION_PATH ]; then
  . $FZF_KEYBINDINGS_PATH
  . $FZF_COMPLETION_PATH
fi
```

Par exemple :

— Vous pouvez accéder à un sous-répertoire très imbriqué avec un minimum d'efforts. Vous tapez d'abord « `cd **` » et appuyez sur `Tab`. Ensuite, vous recevrez une invite de commande avec les chemins des possibilités. La saisie de chaînes de chemin partiel, par exemple, `s/d/b toto`, réduira les chemins potentiels. Vous sélectionnez le chemin à utiliser avec `cd` et les curseur et touche Entrée ;

— Vous pouvez sélectionner une commande dans l'historique des commandes plus efficacement avec un minimum d'efforts. Vous appuyez sur `Ctrl-R` à l'invite de commandes. Ensuite, vous recevrez une invite avec les commandes potentielles. La saisie de chaînes de commande partielles, par exemple, `vim d`, réduira les possibilités. Vous sélectionnez celle à utiliser avec les curseur et touche Entrée.

9.1.4 Enveloppe pour Readline

Certaines commandes telles que `/usr/bin/dash`, qui ne disposent pas de la capacité d'édition de l'historique de lignes de commande, peuvent ajouter cette fonctionnalité de manière transparente en s'exécutant sous `rlwrap` ou ses équivalents.

```
$ rlwrap dash -i
```

Cela fournit une plateforme pratique pour tester des points subtils de `dash` avec un environnement convivial de type `bash`.

9.1.5 Analyse de l'arborescence du code source

La commande rg(1) du paquet `ripgrep` offre une alternative plus rapide que la commande grep(1) pour analyser l'arborescence de code source à la recherche d'une situation typique. Il tire parti des processeurs multicœurs modernes et applique automatiquement des filtres satisfaisants pour ignorer certains fichiers.

9.2 Personnaliser vim

Après avoir appris les bases de vim(1) grâce à la Section 1.4.8, veuillez lire « Seven habits of effective text editing (2000) » (sept pratiques pour une édition efficace de texte) de Bram Moolenaar pour connaitre comment `vim` devrait être utilisé.

9.2.1 Personnalisation de vim avec des fonctionnalités internes

Le comportement de `vim` peut être modifié de manière significative en activant ses fonctionnalités internes à l'aide des commandes en mode `Ex` telles que « `set ...` » pour définir les options vim.

Ces commandes en mode `Ex` peuvent être incluses dans le fichier traditionnel vimrc de l'utilisateur, « `~/.vimrc` » ou « `~/.vim/vimrc` » plus adapté à git. Voici un exemple très simple 1 :

```
""" Generic baseline Vim and Neovim configuration (~/.vimrc)
"""   - For NeoVim, use "nvim -u ~/.vimrc [filename]"
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
let mapleader = ' '             " :h mapleader
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
set nocompatible                " :h 'cp -- sensible (n)vim mode
syntax on                       " :h :syn-on
filetype plugin indent on       " :h :filetype-overview
set encoding=utf-8              " :h 'enc (default: latin1) -- sensible encoding
""" current vim option value can be verified by :set encoding?
set backspace=indent,eol,start  " :h 'bs (default: nobs) -- sensible BS
set statusline=%<%f%m%r%h%w%=%y[U+%04B]%2l/%2L=%P,%2c%V
set listchars=eol:¶,tab:b''▫b''\ ,extends:b''▫b'',precedes:b''▫b'',nbsp:b''▫b''
set viminfo=!,'100,<5000,s100,h " :h 'vi -- bigger copy buffer etc.
""" Pick "colorscheme" from blue darkblue default delek desert elflord evening
""" habamax industry koehler lunaperche morning murphy pablo peachpuff quiet ron
""" shine slate torte zellner
colorscheme industry
""" don't pick "colorscheme" as "default" which may kill SpellUnderline settings
set scrolloff=5                 " :h 'scr -- show 5 lines around cursor
set laststatus=2                " :h 'ls (default 1)  k
""" boolean options can be unset by prefixing "no"
set ignorecase                  " :h 'ic
set smartcase                   " :h 'scs
set autoindent                  " :h 'ai
set smartindent                 " :h 'si
set nowrap                      " :h 'wrap
"set list                        " :h 'list (default nolist)
set noerrorbells                " :h 'eb
set novisualbell                " :h 'vb
set t_vb=                       " :h 't_vb -- termcap visual bell
set spell                       " :h 'spell
set spelllang=en_us,cjk         " :h 'spl -- english spell, ignore CJK
set clipboard=unnamedplus       " :h 'cb -- cut/copy/paste with other app
set hidden                      " :h 'hid
set autowrite                   " :h 'aw
```

1. Exemples de personnalisation plus élaborée : « Vim Galore », « sensible.vim », ...

```
set timeoutlen=300              " :h 'tm
```

Le plan de codage clavier de `vim` peut être modifié dans le fichier vimrc de l'utilisateur. Par exemple :



Attention
N'essayez pas de modifier les combinaisons de touches par défaut sans de très bonnes raisons.

```
""" Popular mappings (imitating LazyVim etc.)
""" Window moves without using CTRL-W which is dangerous in INSERT mode
nnoremap <C-H> <C-W>h
nnoremap <C-J> <C-W>j
nnoremap <C-K> <C-W>k
silent! nnoremap <C-L> <C-W>l
""" Window resize
nnoremap <C-LEFT> <CMD>vertical resize -2<CR>
nnoremap <C-DOWN> <CMD>resize -2<CR>
nnoremap <C-UP> <CMD>resize +2<CR>
nnoremap <C-RIGHT> <CMD>vertical resize +2<CR>
""" Clear hlsearch with <ESC> (<C-L> is mapped as above)
nnoremap <ESC> <CMD>noh<CR><ESC>
inoremap <ESC> <CMD>noh<CR><ESC>
""" center after jump next
nnoremap n nzz
nnoremap N Nzz
""" fast "jk" to get out of INSERT mode (<ESC>)
inoremap  jk <CMD>noh<CR><ESC>
""" fast "<ESC><ESC>" to get out of TERM mode (CTRL-\ CTRL-N)
tnoremap <ESC><ESC> <C-\><C-N>
""" fast "jk" to get out of TERM mode (CTRL-\ CTRL-N)
tnoremap jk <C-\><C-N>
""" previous/next trouble/quickfix item
nnoremap [q <CMD>cprevious<CR>
nnoremap ]q <CMD>cnext<CR>
""" buffers
nnoremap <S-H> <CMD>bprevious<CR>
nnoremap <S-L> <CMD>bnext<CR>
nnoremap [b <CMD>bprevious<CR>
nnoremap ]b <CMD>bnext<CR>
""" Add undo break-points
inoremap  , ,<C-G>u
inoremap  . .<C-G>u
inoremap  ; ;<C-G>u
""" save file
inoremap <C-S> <CMD>w<CR><ESC>
xnoremap <C-S> <CMD>w<CR><ESC>
nnoremap <C-S> <CMD>w<CR><ESC>
snoremap <C-S> <CMD>w<CR><ESC>
""" better indenting
vnoremap < <gv
vnoremap > >gv
""" terminal (Somehow under Linux, <C-/> becomes <C-_> in Vim)
nnoremap <C-_> <CMD>terminal<CR>
"nnoremap <C-/> <CMD>terminal<CR>
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
if ! has('nvim')
""" Toggle paste mode with <SPACE>p for Vim (no need for Nvim)
set pastetoggle=<leader>p
""" nvim default mappings for Vim.  See :h default-mappings in nvim
```

```
""" copy to EOL (no delete) like D for d
noremap Y y$
""" sets a new undo point before deleting
inoremap <C-U> <C-G>u<C-U>
inoremap <C-W> <C-G>u<C-W>
""" <C-L> is re-purposed as above
""" execute the previous macro recorded with Q
nnoremap Q @@
""" repeat last substitute and *KEEP* flags
nnoremap & :&&<CR>
""" search visual selected string for visual mode
xnoremap * y/\V<C-R>"<CR>
xnoremap # y?\V<C-R>"<CR>
endif
```

Pour que les combinaisons de touches ci-dessus fonctionnent correctement, le programme du terminal doit être configuré de manière à générer un « DEL ASCII » pour la touche `Retour arrière` et une « séquence d'échappement » pour la touche `Suppression`.

Diverses autres configurations peuvent être modifiées dans le fichier vimrc de l'utilisateur. Par exemple :

```
""" Use faster 'rg' (ripgrep package) for :grep
if executable("rg")
  set grepprg=rg\ --vimgrep\ --smart-case
  set grepformat=%f:%l:%c:%m
endif
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
""" Retain last cursor position :h '"
augroup RetainLastCursorPosition
  autocmd!
  autocmd BufReadPost *
    \ if line("'\"") > 0 && line ("'\"") <= line("$") |
    \   exe "normal! g'\"" |
    \ endif
augroup END
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
""" Force to use underline for spell check results
augroup SpellUnderline
  autocmd!
  autocmd ColorScheme * highlight SpellBad term=Underline gui=Undercurl
  autocmd ColorScheme * highlight SpellCap term=Underline gui=Undercurl
  autocmd ColorScheme * highlight SpellLocal term=Underline gui=Undercurl
  autocmd ColorScheme * highlight SpellRare term=Underline gui=Undercurl
augroup END
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
""" highlight tailing spaces except when typing as red (set after colorscheme)
highlight TailingWhitespaces ctermbg=red guibg=red
""" \s\+     1 or more whitespace character: <Space> and <Tab>
""" \%#\@<!  Matches with zero width if the cursor position does NOT match.
match TailingWhitespaces /\s\+\%#\@<!$/
```

9.2.2 Personnalisation de vim avec des paquets externes

Des paquets d'extensions externes intéressantes sont disponibles :

— Vim - l'éditeur de texte très répandu – le site officiel de l'amont de vim et des scripts pour vim ;

— VimAwesome – la liste des extensions de vim ;

— vim-scripts – paquet Debian : une collection de scripts pour vim.

Des paquets de greffon dans le paquet vim-scripts peuvent être activés en utilisant le fichier vimrc de l'utilisateur. Par exemple :

```
packadd! secure-modelines
packadd! winmanager
" IDE-like UI for files and buffers with <space>w
nnoremap <leader>w         :WMToggle<CR>
```

Le nouveau système de paquets natif de Vim fonctionne bien avec « `git` » et « `git submodule` ». Un tel exemple de configuration se trouve à mon dépôt git : dot-vim. Cela réalise essentiellement :

— en utilisant « `git` » et « `git submodule` », les derniers paquets externes, tels que « *nom* », sont placés dans `~/.vim/pack/*/opt/`*nom* et similaires ;
— en ajoutant la ligne `:packadd!` *nom* au fichier vimrc de l'utilisateur, ces paquets sont placés dans `runtimepath` ;
— vim charge ces paquets dans `runtimepath` lors de son initialisation ;
— à la fin de son initialisation, les étiquettes des documents installés sont mises à jour avec « `helptags ALL` ».

Pour en savoir plus, démarrez `vim` avec « `vim --startuptime vimstart.log` » pour vérifier la séquence d'exécution réelle et le temps passé pour chaque étape.

Il est assez déroutant de voir trop de façons 2 de gérer et de charger ces paquets externes dans `vim`. Vérifier les informations originelles est la meilleure solution.

saisies de clavier	**information**
`:help package`	explication sur le mécanisme des paquets vim
`:help runtimepath`	explication sur le mécanisme `runtimepath`
`:version`	états internes, y compris les fonctionnalités, pour le fichier vimrc
`:echo $VIM`	variable d'environnement « `$VIM` » utilisée pour localiser le fichier vimrc
`:set runtimepath?`	liste des répertoires dans lesquels rechercher tous les fichiers de prise en charge de l'environnement d'exécution
`:echo $VIMRUNTIME`	variable d'environnement « `$VIMRUNTIME` » utilisée pour localiser divers fichiers de prise en charge de l'environnement d'exécution fournis par le système

Table 9.3 – Informations sur l'initialisation de `vim`

9.3 Enregistrer et présenter des données

9.3.1 Le démon de journal

De nombreux programmes traditionnels enregistrent leurs activités au format de fichier texte dans le répertoire « `/var/log/` ».

logrotate(8) est utilisée pour simplifier l'administration des fichiers journaux sur un système qui génère beaucoup de fichiers journaux.

De nombreux programmes récents enregistrent leurs activités au format de fichier binaire à l'aide du service Journal de systemd-journald(8) dans le répertoire « `/var/log/journal` ».

Vous pouvez consigner des données dans le Journal systemd-journald(8) à partir d'un script d'interpréteur à l'aide de la commande systemd-cat(1).

Consultez Section 3.5 et Section 3.4.

paquet	popularité	taille	description
fail2ban	V:101, I:112	2275	bannir les IP qui provoquent des erreurs d'authentification multiples
logwatch	V:9, I:10	2440	analyseur de journal avec une sortie sympathique en Perl
awstats	V:4.7, I:7.5	6934	analyseur des journaux du serveur web puissant ayant de nombreuses fonctionnalités
analog	V:3, I:81	3739	analyseur des journaux du serveur web
sarg	V:0.61, I:0.67	863	générateur de rapport d'analyse de squid
pflogsumm	V:1.4, I:3.4	173	résumer les entrées de journal de postfix
fwlogwatch	V:0.11, I:0.15	487	analyseur des journaux du pare-feu
squidview	V:0.04, I:0.38	224	surveiller et analyser les fichiers access.log de squid
swatch	V:0.07, I:0.28	99	visualisateur de fichier journal avec correspondance d'expressions rationnelles, mise en évidence et « hooks »
crm114	V:0.05, I:0.30	1371	analyseur et filtre de données diverses, dont les pourriels (CRM114)
icmpinfo	V:0.01, I:0.34	42	interpréter les messages ICMP

Table 9.4 – Liste des analyseurs de journaux système

9.3.2 Analyseur de journaux

Voici les principaux analyseurs de journaux (« `~Gsecurity::log-analyzer` » dans aptitude(8)).

Note
CRM114 fournit une architecture de vocabulaire pour écrire des filtres **fuzzy** avec la bibliothèque d'expressions rationnelles TRE. Une de ses utilisations courantes est le filtrage des pourriels mais il peut aussi être utilisé pour l'analyse de journaux.

9.3.3 Affichage personnalisé des données de texte

Bien que des outils de visualisation de texte (« pager » tels que more(1) et less(1) (consulter la Section 1.4.5) et des outils personnalisés de mise en évidence et de formatage (consulter la Section 11.1.8) peuvent afficher des données textuelles de manière agréable, les éditeurs généralistes (consulter la Section 1.4.6) sont plus souples et paramétrables.

ASTUCE
Pour vim(1) et ses alias de visualisation de texte view(1), « `:set hls` » active la recherche avec mise en évidence.

9.3.4 Affichage personnalisé de la date et de l'heure

Le format par défaut d'affichage de la date et de l'heure par la commande « `"ls -l` » dépend des **paramètres régionaux** (consulter la Section 1.2.6 pour la valeur). La variable « `$LANG` » est d'abord visée, elle peut être surchargée par les variables d'environnement exportées « `$LC_TIME` » ou « `$LC_ALL` ».

2. vim-pathogen était populaire.

Le format réel de l'affichage pour chaque paramètre linguistique dépend de la version de la bibliothèque C standard (paquet `libc6`) utilisée. Par exemple, les différentes versions de Debian ont des valeurs par défaut différentes. Pour les formats iso, consulter ISO 8601.

Si vous désirez vraiment personnaliser ce format d'affichage de la date et de l'heure plus loin que ne le fait **locale**, vous pouvez définir la **valeur de style d'heure** avec le paramètre « `--time-style` » ou par la valeur de « `$TIME_STYLE` » (consultez ls(1), date(1), « `info coreutils 'ls invocation'` »).

valeur de style pour l'heure	locale	affichage de la date et l'heure
`iso`	n'importe laquelle	`01-19 00:15`
`long-iso`	n'importe laquelle	`2009-01-19 00:15`
`full-iso`	n'importe laquelle	`2009-01-19 00:15:16.000000000 +0900`
`locale`	`C`	`Jan 19 00:15`
`locale`	`en_US.UTF-8`	`Jan 19 00:15`
`locale`	`es_ES.UTF-8`	`ene 19 00:15`
`+%d.%m.%y %H:%M`	n'importe laquelle	`19.01.09 00:15`
`+%d.%b.%y %H:%M`	`C` ou `en_US.UTF-8`	`19.Jan.09 00:15`
`+%d.%b.%y %H:%M`	`es_ES.UTF-8`	`19.ene.09 00:15`

Table 9.5 – Exemples d'affichage pour la commande « `ls -l` » avec la **valeur de style pour l'heure**

ASTUCE
Vous pouvez éviter de saisir une option longue sur la ligne de commande à l'aide de l'alias de commande (consulter la Section 1.5.9) :

```
alias ls='ls --time-style=+%d.%m.%y %H:%M'
```

9.3.5 Écho colorisé de l'interpréteur de commandes

L'écho de l'interpréteur de commandes sur la plupart des terminaux peut être colorisé en utilisant le code ANSI d'échappement (consultez « `/usr/share/doc/xterm/ctlseqs.txt.gz` »).

Essayez, par exemple, ce qui suit :

```
$ RED=$(printf "\x1b[31m")
$ NORMAL=$(printf "\x1b[0m")
$ REVERSE=$(printf "\x1b[7m")
$ echo "${RED}RED-TEXT${NORMAL} ${REVERSE}REVERSE-TEXT${NORMAL}"
```

9.3.6 Commandes colorisées

Les commandes colorisées sont pratiques pour examiner la sortie d'une commande dans l'environnement interactif. J'inclus ce qui suit dans mon fichier « `~/.bashrc` ».

```
if [ "$TERM" != "dumb" ]; then
    eval "`dircolors -b`"
    alias ls='ls --color=always'
    alias ll='ls --color=always -l'
    alias la='ls --color=always -A'
    alias less='less -R'
    alias ls='ls --color=always'
    alias grep='grep --color=always'
```

```
    alias egrep='egrep --color=always'
    alias fgrep='fgrep --color=always'
    alias zgrep='zgrep --color=always'
else
    alias ll='ls -l'
    alias la='ls -A'
fi
```

L'utilisation d'alias réserve les effets de couleurs à l'utilisation interactives des commandes. Il a l'avantage sur l'exportation de la variable d'environnement « `export GREP_OPTIONS='--color=auto'` » car la couleur peut être affichée avec des programmes de visualisation (« pager » tels que less(1). Si vous souhaitez supprimer la couleur lors du tubage (« piping » à destination d'autres programmes, utilisez plutôt « `--color=auto` » dans l'exemple ci-dessus pour « `~/.bashrc` ».

ASTUCE
Vous pouvez désactiver ces alias dans l'environnement interactif en appelant l'interpréteur de commandes par « `TERM=dumb bash` ».

9.3.7 Enregistrer l'activité de l'éditeur pour des répétitions complexes

Vous pouvez enregistrer l'activité de l'éditeur pour des répétitions complexes.

Pour Vim, de la manière suivante :

— « `qa` » : démarre l'enregistrement des caractères entrés dans le registre appelé « a ».
— … activité de l'éditeur
— « `q` » : termine l'enregistrement des caractères entrés.
— « `@a` » : exécute le contenu du registre « a ».

Pour Emacs, de la manière suivante :

— « `:C-x (` » : commencer la définition d'une macro clavier.
— … activité de l'éditeur
— « `C-x )` » : terminer la définition d'une macro clavier.
— « `C-x e` » : exécuter une macro clavier.

9.3.8 Enregistrer l'image graphique d'une application X

Il existe plusieurs manières d'enregistrer l'image graphique d'une application X, y compris un affichage `xterm`.

paquet	popularité	taille	écran	commande
`gnome-screenshot`	V:12, I:91	1115	Wayland	application de capture d'écran pour GNOME
`flameshot`	V:7, I:17	3542	Wayland	application de capture d'écran pour steroid
`gimp`	V:40, I:206	33386	Wayland + X	capture d'écran dans le menu de l'interface graphique
`x11-apps`	V:32, I:440	2461	X	xwd(1)
`imagemagick`	V:7, I:266	82	X	import(1)
`scrot`	V:4, I:47	144	X	scrot(1)

Table 9.6 – Liste des outils de manipulation d'images

paquet	popularité	taille	description
etckeeper	V:23, I:26	156	enregistrer les fichiers de configuration et leurs métadonnées avec Git (par défaut), Mercurial ou GNU Bazaar
timeshift	V:7, I:13	4700	utilitaire de restauration du système utilisant rsync ou des instantanés BTRFS
snapper	V:6.3, I:8.4	2461	outil de gestion d'instantané de systèmes de fichiers Linux

Table 9.7 – Liste des paquets pouvant enregistrer l'historique de configuration

9.3.9 Enregistrer les modifications dans des fichiers de configuration

Il existe des outils spécialisés pour enregistrer les modifications apportées aux fichiers de configuration à l'aide de DVCS et pour créer des instantanés système sur Btrfs.

Vous pouvez également envisager l'approche de la Section 10.2.3 avec un script local.

9.4 Surveiller, contrôler et démarrer l'activité des programmes

L'activité des programmes peut être surveillée et contrôlée à l'aide d'outils spécialisés.

paquet	popularité	taille	description
coreutils	V:914, I:1000	19751	nice(1) : lancer un programme avec une priorité d'ordonnancement modifiée
bsdutils	V:445, I:1000	338	renice(1) : modifier la priorité d'ordonnancement d'un programme en cours d'exécution
procps	V:845, I:998	2692	utilitaires du système de fichiers « `/proc` » : ps(1), top(1), kill(1), watch(1), …
psmisc	V:404, I:705	950	utilitaires du système de fichiers « `/proc` » : killall(1), fuser(1), peekfd(1), pstree(1)
time	V:7, I:73	148	time(1) : lancer un programme qui indique l'utilisation des ressources du système en fonction du temps
sysstat	V:112, I:163	1904	sar(1), iostat(1), mpstat(1), … : outils de mesure des performances du système pour Linux
isag	V:0.1, I:3.1	109	Générateur interactif de graphes de l'activité système pour sysstat
lsof	V:463, I:949	492	lsof(8) : afficher la liste des fichiers ouverts par un processus en cours d'utilisation en utilisant l'option « `-p`
strace	V:11, I:99	3250	strace(1) : tracer les appels système et les signaux
ltrace	V:1, I:11	420	ltrace(1) : tracer les appels de bibliothèque
xtrace	V:0.03, I:0.65	342	xtrace(1) : tracer la communication entre un client X11 et le serveur
powertop	V:38, I:231	696	powertop(1) : information concernant la puissance électrique utilisée
cron	V:929, I:997	248	faire tourner des processus en arrière plan selon un calendrier depuis le démon cron(8)
anacron	V:424, I:480	114	ordonnanceur de type cron pour les systèmes qui ne tournent pas 24 heures sur 24
at	V:68, I:87	158	at(1) ou batch(1) : lancer un travail à une heure déterminée ou en dessous d'un niveau de charge donné

Table 9.8 – Liste des outils de surveillance et de contrôle de l'activité des programmes

ASTUCE
Le paquet `procps` fournit des fonctions très basiques de surveillance, de contrôle et de lancement des activités du programme. Vous devriez toutes les apprendre.

9.4.1 Temps d'un processus

Afficher la durée du processus indiqué dans la commande.

```
# time some_command >/dev/null
real    0m0.035s       # time on wall clock (elapsed real time)
user    0m0.000s       # time in user mode
sys     0m0.020s       # time in kernel mode
```

9.4.2 La priorité d'ordonnancement

Une valeur de politesse (« nice » est utilisée pour contrôler la priorité d'ordonnancement du processus.

valeur de politesse	priorité d'ordonnancement
19	la plus basse priorité d'un processus (poli)
0	très haute priorité de processus pour un utilisateur
-20	très haute priorité d'un processus pour root (non poli)

Table 9.9 – Liste des valeurs de politesse pour la priorité d'ordonnancement

```
# nice  -19 top                                      # very nice
# nice --20 wodim -v -eject speed=2 dev=0,0 disk.img # very fast
```

Parfois, une valeur extrême de politesse (« nice ») fait plus de mal que de bien au système. Utilisez cette commande avec précaution.

9.4.3 La commande ps

La commande ps(1) sous Debian comporte à la fois les fonctionnalités de BSD et de SystemV, elle aide à identifier l'activité des processus de manière statique.

style	commande typique	fonction
BSD	`ps aux`	afficher %CPU %MEM
System V	`ps -efH`	afficher le PPID

Table 9.10 – Liste des styles de la commande ps

Les processus enfants zombies (« defunct ») peuvent être tués par l'identifiant du processus parent identifié dans le champ « `PPID` ».

La commande pstree(1) affiche une arborescence des processus.

9.4.4 La commande top

top(1) sous Debian a de riches fonctionnalités et aide à identifier de manière dynamique quels sont les processus qui ont une activité curieuse.

C'est un programme interactif en mode plein écran. Vous pouvez afficher l'aide sur son utilisation en tapant la touche « h » et la quitter en tapant la touche « q ».

9.4.5 Afficher les fichier ouverts par un processus

Vous pouvez afficher la liste des fichiers ouverts par un processus avec un identifiant de processus (PID), par exemple de 1 en faisant ce qui suit :

```
$ sudo lsof -p 1
```

PID=1 est habituellement le programme `init`.

9.4.6 Tracer l'activité d'un programme

Vous pouvez tracer l'activité d'un programme avec strace(1), ltrace(1) ou xtrace(1) pour les appels système, les appels bibliothèque ou la communication entre le serveur et le client X11.

Vous pouvez tracer les appels système de la commande `ls` de la manière suivante :

```
$ sudo strace ls
```

ASTUCE
Utilisez le script **strace-graph** trouvé dans **/usr/share/doc/strace/examples/** pour créer une belle arborescence.

9.4.7 Identification des processus qui utilisent des fichiers ou des sockets

Vous pouvez aussi identifier les processus qui utilisent des fichiers par fuser(1), par exemple pour « `/var/log/mail.log` » en faisant ce qui suit :

```
$ sudo fuser -v /var/log/mail.log
                     USER        PID ACCESS COMMAND
/var/log/mail.log:   root       2946 F.... rsyslogd
```

Vous voyez que le fichier « `/var/log/mail.log` » est ouvert en écriture par la commande rsyslogd(8).

Vous pouvez aussi identifier les processus qui utilisent des sockets par fuser(1), par exemple, pour « `smtp/tcp` » par ce qui suit :

```
$ sudo fuser -v smtp/tcp
                     USER        PID ACCESS COMMAND
smtp/tcp:            Debian-exim   3379 F.... exim4
```

Vous savez maintenant que exim4(8) tourne sur votre système pour gérer les connexions TCP du port SMTP (25).

9.4.8 Répéter une commande avec un intervalle constant

watch(1) execute un programme de manière répétitive avec un intervalle constant tout en affichant sa sortie en plein écran.

```
$ watch w
```

Cela permet l'affichage, mis à jour toutes les deux secondes, de qui est connecté au système.

9.4.9 Répéter une commande en bouclant entre des fichiers

Il existe plusieurs manières de répéter une boucle de commande entre des fichiers correspondant à une condition, par exemple, correspondant au motif « glob » « `*.ext` ».

— Méthode de la boucle « for» » de l'interpréteur de commandes (consultez Section 12.1.4) :

```
for x in *.ext; do if [ -f "$x"]; then command "$x" ; fi; done
```

— combinaison de find(1) et de xargs(1) :

```
find . -type f -maxdepth 1 -name '*.ext' -print0 | xargs -0 -n 1 command
```

— find(1) avec l'option « `-exec` » avec une commande :

```
find . -type f -maxdepth 1 -name '*.ext' -exec command '{}' \;
```

— find(1) avec l'option « `-exec` » avec un court script de l'interpréteur :

```
find . -type f -maxdepth 1 -name '*.ext' -exec sh -c "command '{}' && echo 'successful'" \;
```

Les exemple ci-dessus ont été écrits afin d'assurer une prise en compte correcte de noms de fichiers étranges tels que ceux qui comportent des espaces. Consultez Section 10.1.5 pour une utilisation avancée de find(1).

9.4.10 Lancer un programme depuis l'interface graphique

Pour l'interface en ligne de commande (CLI), le premier programme trouvé dans les répertoires spécifiés dans la variable d'environnement `$PATH`, et dont le nom correspond, est exécuté. Consulter Section 1.5.3.

Pour l'interface utilisateur graphique (GUI) conforme aux normes freedesktop.org, les fichiers `*.desktop` du dossier `/usr/share/applications/` fournissent les attributs nécessaires pour l'affichage du menu graphique de chaque programme. Chaque paquet conforme au système de menu xdg de Freedesktop.org installe ses données de menu fournies par « *.desktop » sous « /usr/share/applications/ ». Les environnements de bureau modernes conformes à la norme Freedesktop.org utilisent ces données pour générer leur menu à l'aide du paquet xdg-utils. Consulter « /usr/share/doc/xdg-utils/README ».

Par exemple, le fichier `chromium.desktop` définit les attributs pour le "Navigateur Web Chromium" tel que "Name" pour le nom du programme, "Exec" pour le chemin et les paramètres d'exécution du programme, "Icon" pour l'icône utilisée, etc. (voir la Spécification d'Entrée de fichier Desktop) comme suit :

```
[Desktop Entry]
Version=1.0
Name=Chromium Web Browser
GenericName=Web Browser
Comment=Access the Internet
Comment[fr]=Explorer le Web
Exec=/usr/bin/chromium %U
Terminal=false
X-MultipleArgs=false
Type=Application
Icon=chromium
Categories=Network;WebBrowser;
MimeType=text/html;text/xml;application/xhtml_xml;x-scheme-handler/http;x-scheme-handler/ ↩
    https;
StartupWMClass=Chromium
StartupNotify=true
```

C'est une description extrêmement simplifiée. Les fichiers `*.desktop` sont analysés comme suit :

L'environnement de bureau définit les variables d'environnement `$XDG_DATA_HOME` et `$XDG_DATA_DIR`. Par exemple, sous GNOME :

— « `$XDG_DATA_HOME` » n'est pas paramétré. (La valeur par défaut de « `$HOME/.local/share` » est utilisée.)
— « `$XDG_DATA_DIRS` » est paramétré à « `/usr/share/gnome:/usr/local/share/:/usr/share/` ».

Les répertoires de base (voir Spécification du répertoire de base XDG et les répertoires des « `applications` » sont donc les suivants :

— `$HOME/.local/share/` → `$HOME/.local/share/applications/`
— `/usr/share/gnome/` → `/usr/share/gnome/applications/`
— `/usr/local/share/` → `/usr/local/share/applications/`
— `/usr/share/` → `/usr/share/applications/`

Les fichiers `*.desktop` sont parcourus dans ces répertoires `applications` dans cet ordre.

ASTUCE
Une entrée de menu « graphique » personnalisée peut être créée en ajoutant un fichier « `*.desktop` » dans le dossier « `$HOME/.local/share/applications/` ».

ASTUCE
La ligne « `Exec=...` » n'est pas analysée par l'interpréteur. Utilisez la commande env(1) si des variables d'environnement doivent être définies.

ASTUCE
De la même façon, si un fichier « `*.desktop` » est créé dans le répertoire « `autostart` » dans ces répertoires de base, le programme spécifié dans le fichier « `*.desktop` » est exécuté automatiquement lorsque l'environnement de bureau est démarré. Consultez la Spécification de démarrage automatique d'applications.

ASTUCE
De la même façon, si un fichier « `*.desktop` » est créé dans le répertoire « `$HOME/Desktop` » et que l'environnement de bureau est configuré afin de prendre en charge l'exécution depuis une icône de bureau, le programme qui y est spécifié est exécuté lorsqu'on clique sur l'icône. Veuillez noter que le nom du répertoire « `$HOME/Desktop` » est dépendant de la localisation. Consultez xdg-user-dirs-update(1).

9.4.11 Personnaliser le programme à lancer

Certains programmes lancent automatiquement d'autres programmes. Voici des points-clés pour la personnalisation de ce processus :

— Menu de configuration des applications :
  — GNOME desktop: "Settings" → "Apps" → "Default Apps"
  — KDE desktop: "Application Launcher" → "System Settings" → "Default Applications"
  — navigateur Iceweasel : « Éditer » → « Préférences » → « Applications »
  — mc(1) : « `/etc/mc/mc.ext` »
— les variables d'environnement telles que « `$BROWSER` », « `$EDITOR` », « `$VISUAL` » et « `$PAGER` » (consulter environ(7))
— le système update-alternatives(1) pour des programmes tels que « `editor` », « `view` », « `x-www-browser` », « `gnome-www-browser` » et « `www-browser` » (consultez Section 1.4.7)

— le contenu des fichiers « `~/.mailcap` » et « `/etc/mailcap` » qui associe un type MIME avec un programme (consultez mailcap(5))
— le contenu des fichiers « `~/.mime.types` » et « `/etc/mime.types` » qui associe l'extension du nom de fichier avec un type MIME (consultez run-mailcap(1))

ASTUCE
update-mime(8) met à jour le fichier ≤ `/etc/mailcap` » en utilisant le fichier « `/etc/mailcap.order` » (consultez mailcap.order(5)).

ASTUCE
Le paquet `debianutils` fournit sensible-browser(1), sensible-editor(1), et sensible-pager(1) qui prennent des décisions raisonnables concernant l'éditeur, le visualisateur, le navigateur à appeler respectivement. Je vous recommande de lire ces scripts de l'interpréteur de commandes.

ASTUCE
De façon à faire tourner une application de console telle que `mutt` sous environnement graphique en tant qu'application préférée, vous devriez créer une application graphique comme suit et définir « `/usr/local/bin/mutt-term` » comme étant votre application préférée à lancer comme suit :

```
# cat /usr/local/bin/mutt-term <<EOF
#!/bin/sh
gnome-terminal -e "mutt \$@"
EOF
# chmod 755 /usr/local/bin/mutt-term
```

9.4.12 Tuer un processus

Utilisez kill(1) pour tuer (ou envoyer un signal à) un processus avec son identifiant de processus.

Utilisez killall(1) ou pkill(1) pour faire la même chose avec le nom de commande du processus et d'autres attributs.

9.4.13 Planifier des tâches qui s'exécutent une fois

Exécutez de la manière suivante la commande at(1) pour planifier un travail qui s'exécute une fois :

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

9.4.14 Planifier des tâches qui s'exécutent régulièrement

Utilisez cron(8) pour planifier des tâches qui s'exécutent régulièrement. Consultez crontab(1) et crontab(5).

Vous pouvez planifier le lancement des processus en tant qu'utilisateur normal, par exemple `toto` en créant un fichier crontab(5) file comme « `/var/spool/cron/crontabs/toto` » avec la commande « `crontab -e` ».

Voici un exemple de fichier crontab(5).

```
# use /usr/bin/sh to run commands, no matter what /etc/passwd says
SHELL=/bin/sh
# mail any output to paul, no matter whose crontab this is
MAILTO=paul
# Min Hour DayOfMonth Month DayOfWeek command (Day... are OR'ed)
```

valeur du signal	nom du signal	action	remarque
0	---	aucun signal n'est envoyé (consulter kill(2))	vérifier si le processus est en cours d'exécution
1	SIGHUP	terminer le processus	terminal déconnecté (signal bloqué)
2	SIGINT	terminer le processus	interruption à partir du clavier (`CTRL-C`)
3	SIGQUIT	termine le processus et faire un dump core	quitter depuis le clavier (`CTRL-\`)
9	SIGKILL	terminer le processus	signal kill imblocable
15	SIGTERM	terminer le processus	signal kill blocable

Table 9.11 – Liste des signaux couramment utilisés avec la commande kill

```
# run at 00:05, every day
5  0  *  * *   $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# run at 14:15 on the first of every month -- output mailed to paul
15 14 1  * *   $HOME/bin/monthly
# run at 22:00 on weekdays(1-5), annoy Joe. % for newline, last % for cc:
0 22 *   * 1-5 mail -s "It's 10pm" joe%Joe,%%Where are your kids?%.%%
23 */2 1 2 *   echo "run 23 minutes after 0am, 2am, 4am ..., on Feb 1"
5  4 *   * sun echo "run at 04:05 every Sunday"
# run at 03:40 on the first Monday of each month
40 3 1-7 * *   [ "$(date +%a)" == "Mon" ] && command -args
```

ASTUCE
Sur un système qui ne tourne pas en permanence, installez le paquet `anacron` afin de planifier les tâches périodiques à des intervalles particulier dès que le temps de fonctionnement « uptime » de la machine le permet. Consultez anacron(8) et anacrontab(5).

ASTUCE
Vous pouvez lancer périodiquement les scripts de maintenance planifiée du système, depuis le compte de l'administrateur en les plaçant dans « `/etc/cron.hourly/` », « `/etc/cron.daily/` », « `/etc/cron.weekly/` » ou « `/etc/cron.monthly/` ». L"échéancier d'exécution de ces scripts peut être personnalisé dans « `/etc/crontab` » et « `/etc/anacrontab` ».

Systemd a une capacité de bas niveau pour planifier l'exécution de programmes sans démon `cron`. Par exemple, `/lib/systemd/system/apt-daily.timer` et `/lib/systemd/system/apt-daily.service` configurent des activités quotidiennes de téléchargement d'apt. Consulter systemd.timer(5) .

9.4.15 Planifier des tâches lors d'évènements

Systemd peut planifier des programmes non seulement pour des évènements d'horloge, mais aussi pour des évènements de montage. Consultez Section 10.2.3.3 et Section 10.2.3.2 pour des exemples.

9.4.16 touche Alt-SysRq

Appuyer sur Alt-SysRq (Imp écr) suivi d'une touche fait reprendre magiquement le contrôle du système.

touche suivant Alt-Sys	**description de l'action**
k	**k**ill (tuer) tous les processus sur la **console virtuelle actuelle** (SAK)
s	**s**ynchroniser tous les systèmes de fichiers montés afin d'éviter la corruption des données
u	remonter en lecture seule tous les systèmes de fichiers montés (**u**mount)
r	restaurer le clavier depuis le mode **r**aw (brut) après un plantage de X

Table 9.12 – Listes des touches notables de commande SAK (« Secure attention keys »)

Plus d'informations sur Guide de l'utilisateur et de l'administrateur du noyau Linux -> Linux Magic System Request Key Hacks.

ASTUCE
Depuis un terminal SSH, etc., vous pouvez utiliser la fonctionnalité Alt-SysRq en écrivant vers « `/proc/sysrq-trigger` ». Par exemple;, « `echo s > /proc/sysrq-trigger; echo u > /proc/sysrq-trigger` » depuis l'invite de l'interpréteur de commandes de l'administrateur **s**yncs (synchronise) et **u**mount (démonte) tous les systèmes de fichiers montés.

Le noyau Linux Debian amd64 actuel (2021) a `/proc/sys/kernel/sysrq=438=0b110110110` :

— 2 = 0x2 – contrôle du niveau de journalisation de la console (ON) ;
— 4 = 0x4 – contrôle du clavier (SAK, unraw) (ON) ;
— 8 = 0x8 – vidages de débogage des processus, etc. (OFF) ;
— 16 = 0x10 – activation de la commande sync (ON) ;
— 32 = 0x20 – remontage en lecture seule (ON) ;
— 64 = 0x40 – signalisation des processus (term, kill, oom-kill) (OFF) ;
— 128 = 0x80 – redémarrage/extinction (ON) ;
— 256 = 0x100 – définition de la priorité de toutes les tâches RT (ON).

9.5 Astuces de maintenance du système

9.5.1 Qui se trouve sur le système ?

Vous pouvez rechercher qui se trouve sur le système par les commandes suivantes :

— who(1) affiche qui est connecté ;
— w(1) affiche qui sont connectés et ce qu'ils font ;
— last(1) affiche une liste des derniers utilisateurs connectés ;
— lastb(1) affiche une liste des utilisateurs s'étant mal connectés.

ASTUCE
« `/var/run/utmp` » et « `/var/log/wtmp` » conservent ces informations de l'utilisateur. Consultez login(1) et utmp(5).

9.5.2 Prévenir tout le monde

Vous pouvez envoyer un message à toutes les personnes connectées au système avec wall(1) en faisant ce qui suit :

```
$ echo "We are shutting down in 1 hour" | wall
```

9.5.3 Identification du matériel

Pour les périphériques similaires à PCI (AGP, PCI-Express, CardBus, ExpressCard, etc.), lspci(8) (probablement avec l'option « `-nn` ») est un bon point de départ pour l'identification du matériel.

Vous pouvez aussi identifier le matériel en lisant le contenu de « `/proc/bus/pci/devices` » ou en parcourant l'arborescence de répertoires se trouvant sous « `/sys/bus/pci` » (consultez Section 1.2.12).

paquet	popularité	taille	description
pciutils	V:279, I:994	289	utilitaires PCI de Linux : lspci(8)
usbutils	V:146, I:894	322	utilitaires USB de Linux : lsusb(8)
nvme-cli	V:30, I:39	2222	utilitaires NVMe pour Linux : nvme(1)
pcmciautils	V:3.8, I:5.7	92	utilitaires PCMCIA pour Linux : pccardctl(8)
scsitools	V:0.2, I:2.0	261	collection d'outils pour la gestion des périphériques SCSI : lsscsi(8)
procinfo	V:0.3, I:5.7	149	informations sur le système obtenues dans « /proc » : lsdev(8)
lshw	V:12, I:87	971	informations concernant la configuration matérielle : lshw(1)
discover	V:26, I:527	81	système d'identification du matériel : discover(8)

Table 9.13 – Listes des outils d'identification du matériel

paquet	popularité	taille	description
console-setup	V:87, I:977	422	fonte de la console Linux et utilitaires de table de caractères
keyd	V:1.7, I:1.8	1437	keyboard key remapping daemon using kernel level input primitives (evdev, uinput)
keyd-application-mapper	V:0.02, I:0.07	34	keyboard key remapping daemon - application-specific remapper
x11-xserver-utils	V:376, I:532	559	utilitaires pour le serveur X : xset(1), xmodmap(1)
acpid	V:51, I:74	158	démon servant à gérer les événements délivrés par l'Interface avancée de configuration et de gestion de l'énergie ACPI (« Advanced Configuration and Power Interface »)
acpi	V:6, I:69	49	utilitaire d'affichage des informations des périphériques ACPI
sleepd	V:0.08, I:0.10	84	démon permettant de mettre un ordinateur portable en veille lorsqu'il est inactif
hdparm	V:111, I:192	246	optimisation de l'accès aux disques durs (consultez Section 9.6.9)
smartmontools	V:249, I:274	2455	contrôle et surveillance des systèmes de stockage en utilisant S.M.A.R.T.
setserial	V:3.0, I:4.7	104	collection d'outils pour gérer les ports série
memtest86+	V:1, I:19	12483	collection d'outils pour gérer la mémoire physique
scsitools	V:0.2, I:2.0	261	collection d'outils pour gérer le matériel SCSI
setcd	V:0.06, I:0.59	33	optimisation de l'accès au lecteur de CD
big-cursor	I:0.57	26	curseurs de souris plus grands pour X

Table 9.14 – Liste des outils de configuration du matériel

9.5.4 Configuration matérielle

Bien que l'essentiel de la configuration du matériel puisse être gérée au moyen des outils graphiques qui accompagnent les environnements de bureau graphiques modernes comme GNOME ou KDE, c'est une bonne idée de connaître certaines méthodes de base permettant de le configurer.

For the keyboard and terminal configuration, see Section 8.2 and Section 8.3.

Ici, ACPI est une infrastructure de gestion de l'alimentation électrique du système plus récente qu'APM.

ASTUCE
L'ajustement de la fréquence d'horloge des processeurs modernes est gérée par des modules du noyaux tels que `acpi_cpufreq`.

9.5.5 Heure système et matérielle

Ce qui suit permet de définir l'heure du système et du matériel à MM/DD hh:mm, CCYY :

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

Sur un système Debian, l'heure est normalement affichée en heure locale mais l'heure système et matérielle utilisent habituellement l'heure TUC(GMT).

Si l'heure matérielle est réglée en UTC, modifiez le réglage pour « `UTC=yes` » dans le fichier « `/etc/default/rcS` ».

La commande suivante relance la configuration du fuseau horaire utilisé par le système Debian.

```
# dpkg-reconfigure tzdata
```

Si vous désirez ajuster l'heure de votre système par l'intermédiaire du réseau, vous pouvez envisager l'utilisation du service NTP avec un paquet tel que `systemd-timesyncd`, `ntpsec`, `ntpsec-ntpdate` ou `chrony`.

ASTUCE
Sous systemd, utilisez plutôt `systemd-timesyncd`pour la synchronisation avec l'heure du réseau. Voir systemd-timesyncd(8).

Consultez ce qui suit.

— Comment gérer précisément la date et l'heure
— Projet de services publics NTP
— Le paquet `ntpsec-doc`

ASTUCE
ntptrace(1) du paquet `ntpsec` peut suivre la trace d'une chaîne de serveurs NTP jusqu'à la source primaire.

9.5.6 L'infrastructure de gestion du son

Les pilotes de périphériques des cartes sons pour les versions actuelles de Linux sont fournies par Advanced Linux Sound Architecture (ALSA). ALSA fournit un mode d'émulation du système précédent Open Sound System (OSS) pour des raisons de compatibilité.

Les logiciels d'application peuvent être configurés non seulement pour accéder directement aux périphériques audio, mais aussi pour y accéder à l'aide d'un système de serveur audio standardisé. Actuellement, PulseAudio, JACK et PipeWire sont utilisés comme systèmes de serveur audio. Consulter la page wiki Debian sur Sound pour les dernières avancées.

Il y a habituellement un moteur de son commun pour chacun des environnements de bureau les plus courants. Chaque moteur de son utilisé par l'application peut choisir de se connecter à un serveur de son différent.

ASTUCE
Utilisez « `cat /dev/urandom > /dev/audio` » ou speaker-test(1) pour tester les hauts-parleurs (^C pour arrêter).

ASTUCE
Si vous n'arrivez pas à obtenir de sons, il est possible que votre haut-parleur soit connecté à une sortie muette (« muted »). Les systèmes de son modernes ont de nombreuses sorties. alsamixer(1) du paquet `alsa-utils` est pratique pour configurer les paramètres de volume et de coupure son.

paquet	popularité	taille	description
`alsa-utils`	V:349, I:457	2700	utilitaires de configuration et d'utilisation d'ALSA
`oss-compat`	V:0.5, I:8.7	21	Compatibilité OSS sous ALSA évitant les erreurs « `/dev/dsp not found` »
`pipewire`	V:330, I:370	140	serveur multimédia de moteur de traitement audio et vidéo – métapaquet
`pipewire-bin`	V:336, I:370	2254	serveur multimédia de moteur de traitement audio et vidéo – programmes de serveur audio et en ligne de commande
`pipewire-alsa`	V:175, I:243	197	serveur multimédia de moteur de traitement audio et vidéo – serveur audio pour remplacer ALSA
`pipewire-pulse`	V:304, I:350	56	serveur multimédia de moteur de traitement audio et vidéo – serveur audio pour remplacer PulseAudio
`pulseaudio`	V:144, I:166	6616	serveur PulseAudio
`libpulse0`	V:445, I:578	984	bibliothèque de client pour PulseAudio
`jackd`	V:2, I:13	8	serveur du kit de connexion audio JACK. (JACK) (faible latence)
`libjack0`	V:1.7, I:8.3	320	bibliothèque du kit de connexion JACK. (JACK) (faible latence)
`libphonon4qt5-4`	V:16, I:38	572	Phonon : moteur de son de KDE

Table 9.15 – Liste des paquets son

9.5.7 Désactiver l'économiseur d'écran

Pour désactiver l'écran de veille, utilisez les commandes suivantes :

9.5.8 Désactiver les bips

On peut toujours débrancher le haut-parleur du PC pour désactiver les bips. La suppression du module du noyau `pcspkr` le fait pour vous.

Ce qui suit évite que le programme readline(3) utilisé parbash(1) ne bipe lors de l'apparition d'un caractère d'alerte (ASCII=7).

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

environnement	commande
Console Linux	`setterm -powersave off`
Système X Window (couper l'économiseur d'écran)	`xset s off`
X Window (désactive dpms)	`xset -dpms`
X Window (interface de configuration graphique d'économiseur d'écran)	`xscreensaver-command -prefs`

Table 9.16 – Liste des commandes pour désactiver l'économiseur d'écran

9.5.9 Utilisation de la mémoire

Vous avez à votre disposition deux moyens d'obtenir l'état d'utilisation de la mémoire.

— Le message de démarrage du noyau dans « `/var/log/dmesg` » donne exactement la taille de la mémoire disponible.

— free(1) et top(1) affichent des informations concernant les ressources mémoire sur le système actif.

Voici un exemple.

```
# grep '\] Memory' /var/log/dmesg
[    0.004000] Memory: 990528k/1016784k available (1975k kernel code, 25868k reserved, 931k ↩
    data, 296k init)
$ free -k
             total       used       free     shared    buffers     cached
Mem:        997184     976928      20256          0     129592     171932
-/+ buffers/cache:     675404     321780
Swap:      4545576          4    4545572
```

Vous pourriez vous demander « dmesg me rapporte 990 Mo de mémoire libre et « free -k » me dit que 320 Mo sont libres. Il manque plus de 600 Mo… ».

Ne vous inquiétez pas de la taille importante de « `used` » et de la petite taille de « `free` » sur la ligne « `Mem:` », mais lisez ce qui se trouve sous celle-ci (675404 et 321780 dans l'exemple ci-dessous) et détendez-vous.

Pour mon MacBook avec 1Go=1048576k DRAM (la mémoire vidéo en prend une partie), je peux voir ce qui suit :

affiché	taille
Taille totale dans dmesg	1016784k = 1Go - 31792k
Libre dans dmesg	990528k
Total sous l'interpréteur de commandes	997184k
Libre sous l'interpréteur de commandes	20256k (mais réellement 321780k)

Table 9.17 – Taille mémoire affichée

9.5.10 Vérification de la sécurité et de l'intégrité du système

Une mauvaise maintenance du système peut rendre votre système vulnérable à une attaque externe.

Pour la vérification de la sécurité et de l'intégrité du système, vous pouvez démarrer avec ce qui suit :

— Le paquet `debsums`, consultez debsums(1) et Section 2.5.2.

— Le paquet `chkrootkit`, consultez chkrootkit(1).

— La familles de paquets `clamav`, consultez clamscan(1) et freshclam(1).

— FAQ de sécurité Debian.

— Manuel de sécurisation de Debian.

paquet	popularité	taille	description
logcheck	V:4.2, I:5.2	120	démon pour poster à l'administrateur les anomalies des fichiers journaux du système
debsums	V:4, I:29	108	utilitaire pour vérifier les fichiers des paquets installés d'après leur somme de contrôle MD5
chkrootkit	V:10, I:14	988	détecteur de rootkit
clamav	V:8, I:37	20551	utilitaire anti-virus pour UNIX - interface en ligne de commandes
tiger	V:1.0, I:1.3	7800	signale les vulnérabilités du système pour la sécurité
tripwire	V:1.1, I:1.4	5060	vérificateur d'intégrité des fichiers et répertoires
john	V:1.1, I:7.1	469	outils de casse des mots de passe actifs
aide	V:2.1, I:2.7	328	Environnement avancé de détection d'intrusion (« Advanced Intrusion Detection Environment ») — bibliothèque statique
integrit	V:0.07, I:0.16	2759	programme de vérification de l'intégrité des fichiers
crack	V:0.04, I:0.52	151	programme pour deviner les mots de passe

Table 9.18 – Liste d'outils pour la vérification de la sécurité et de l'intégrité du système

Voici un script simple pour rechercher des fichiers typiques ayant des permissions incorrectes d'écriture pour tout le monde.

```
# find / -perm 777 -a \! -type s -a \! -type l -a \! \( -type d -a -perm 1777 \)
```



Attention
Comme le paquet `debsums` utilise des sommes de contrôle MD5 enregistrées de manière statique, on ne peut pas lui faire entièrement confiance comme outil d'audit de la sécurité envers des attaques malveillantes.

9.6 Astuces relatives au stockage des données

Booting your system as the Linux live system (see Section 3.2.2 and Section 3.2.3) makes it easy for you to reconfigure data storage on the installed system.

Note
Statements on hard disk (HDD) are applicable to other storage devices such as SSD / USB flash drive / Memory card / ... . Replace device names in examples such as `/dev/sda` with applicable device names `/dev/nvme0`, `/dev/mmcblk0`, ...

Il se peut que vous deviez umount(8) certains périphériques manuellement à partir de la ligne de commande avant d'opérer sur eux s'ils sont automatiquement montés par le système de bureau graphique.

9.6.1 Utilisation de l'espace disque

L'utilisation de l'espace disque peut être estimée à l'aide de programmes fournis par les paquets `mount`, `coreutils` et `xdu` :

— mount(8) indique tous les systèmes de fichiers (disques) montés.
— df(1) indique l'espace disque occupé par les systèmes de fichiers.

— du(1) indique l'espace disque occupé par une arborescence de répertoires.

ASTUCE
La sortie de du(1) peut être renvoyée vers xdu(1) pour réaliser une présentation graphique et interactive avec « `du -k . |xdu` », « `sudo du -k -x / |xdu` », etc.

9.6.2 Configuration de la partition du disque

Bien que fdisk(8) ait été considéré comme un standard pour la configuration de la partition du disque dur, parted(8) mérite une certaine attention. « Données de partition du disque », « table de partitions » et « Étiquette de disque » sont tous des synonymes.

Les anciens PC utilisent le schéma classique du MBR (Master Boot Record) (« enregistrement d'amorçage maître ») pour conserver les données de partitionnement du disque sur le premier secteur, c'est-à-dire, le secteur 0 LBA (512 octets).

Les PC récents avec une UEFI (Unified Extensible Firmware Interface) (« Interface micrologicielle extensible unifiée »), incluant les Mac basés sur Intel, utilisent le schéma GPT (GUID Partition Table) (« table de partitionnement GUID ») pour conserver les données de partitionnement du disque ailleurs que sur le premier secteur.

Alors que fdisk(8) a été l'outil standard de partitionnement de disque, parted(8) le remplace maintenant.

paquet	popularité	taille	description
`util-linux`	V:922, I:1000	4826	divers utilitaires systèmes dont fdisk(8) etcfdisk(8)
`parted`	V:443, I:566	126	programme GNU de redimensionnement des partitions Parted
`gparted`	V:12, I:81	2313	éditeur de partitions de GNOME basé sur `libparted`
`gdisk`	V:20, I:240	967	éditeur de partitions pour disque hybride GPT/MBR
`kpartx`	V:16, I:23	78	programme pour créer des mappages de périphériques pour les partitions

Table 9.19 – Listes de paquets de gestion de la partition du disque



Attention
Bien que parted(8) prétend pouvoir créer et redimensionner aussi les systèmes de fichiers, il est plus sûr de toucher à ces choses-là en utilisant des outils spécialisés et bien maintenus tels que mkfs(8) (mkfs.msdos(8), mkfs.ext2(8), mkfs.ext3(8), mkfs.ext4(8), …) et resize2fs(8).

Note
De manière à passer de GPT à MBR, il vous faut d'abord effacer les premiers blocs du disque directement (consultez Section 9.8.6) et utiliser « `parted /dev/sdx mklabel gpt` » ou « `parted /dev/sdx mklabel msdos` » afin de le mettre en place. Vous remarquerez que « `msdos` » est utilisé ici pour MBR.

9.6.3 Accès à une partition en utilisant l'UUID

Bien que la reconfiguration de votre partitionnement ou l'ordre d'activation des médias d'enregistrement amovibles puisse conduire à des noms de partition différents, vous pouvez toujours y accéder. Cela vous aidera aussi si vous avez plusieurs disques et que votre BIOS/UEFI ne leur donne pas toujours le même nom de périphérique.

— mount(8) avec l'option « `-U` » peut monter un périphérique en mode bloc en utilisant l'UUID plutôt que son nom de fichier de périphérique comme « `/dev/sda3` ».

— « `/etc/fstab` » (consultez fstab(5)) peut utiliser l'UUID.
— Les chargeurs initiaux (Section 3.1.2) peuvent aussi utiliser UUID.

ASTUCE
Vous pouvez tester l'UUID d'un périphérique spécial en mode bloc avec blkid(8).
Vous pouvez également l'examiner et obtenir d'autres informations avec « `lsblk -f` ».

9.6.4 LVM2

LVM2 est un gestionnaire de volume logique pour le noyau Linux. Avec LVM2, les partitions peuvent être créées sur des volumes logiques plutôt que sur des disques durs physiques.

LVM requiert ce qui suit :

— la prise en charge de device-mapper dans le noyau Linux (présent par défaut sur les noyaux Debian) ;
— la bibliothèque de prise en charge de device-mapper en espace utilisateur (paquet `libdevmapper*`) ;
— le paquet des outils LVM2 en espace utilisateur (`lvm2`).

Démarrez l'apprentissage de LVM2 par la lecture des pages de manuel suivantes :

— lvm(8) : les bases du mécanisme de LVM2 (liste de toutes les commandes LVM2) ;
— lvm.conf(5) : le fichier de configuration pour LVM2 ;
— lvs(8) : rapport d'informations sur les volumes logiques ;
— vgs(8) : rapport d'informations sur les groupes de volumes ;
— pvs(8) : rapport d'informations sur les volumes physiques.

9.6.5 Configuration de systèmes de fichiers

Pour le système de fichiers ext4, le paquet `e2fsprogs` fournit les éléments suivants :

— mkfs.ext4(8) pour créer un nouveau système de fichiers ext4
— fsck.ext4(8) pour vérifier et réparer un système de fichiers ext4 existant
— tune2fs(8) pour configurer le superbloc d'un système de fichiers ext4
— debugfs(8) pour un débogage interactif du système de fichiers ext4. (Il possède la commande `undel` permettant de récupérer des fichiers effacés.)

Les commandes mkfs(8) et fsck(8) font partie du paquet `e2fsprogs` en tant que frontal à de nombreux programmes dépendant du système de fichiers (`mkfs.fstype` et `fsck.fstype`). Pour le système de fichiers ext4, il y a mkfs.ext4(8) et fsck.ext4(8) (ils sont liés par un lié symboliquement à mke2fs(8) et e2fsck(8)).

Des commandes semblables sont disponibles pour chaque système de fichiers pris en charge par Linux.

ASTUCE
Le système de fichiers ext4 est le système de fichiers par défaut pour les systèmes Linux. Son utilisation est fortement recommandée, sauf cas spécifiques.
L'état de Btrfs peut être trouvé sur le wiki Debian sur btrfs et le wiki kernel.org sur btrfs. Il devrait être le prochain système de fichiers par défaut après le système de fichiers ext4.
Certains outils permettent l'accès au système de fichiers sans prise en charge par le noyau Linux (consultez Section 9.8.2).

paquet	popularité	taille	description
e2fsprogs	V:842, I:998	1560	utilitaires pour les systèmes de fichiers ext2/ext3/ext4
btrfs-progs	V:54, I:78	5291	utilitaire pour le système de fichiers Btrfs
reiserfsprogs	V:12, I:21	473	utilitaire pour le système de fichiers Reiserfs
zfsutils-linux	V:32, I:33	1905	utilitaires pour le système de fichiers OpenZFS
dosfstools	V:274, I:568	310	utilitaire pour le système de fichiers FAT (Microsoft :MS-DOS, Windows)
exfatprogs	V:36, I:466	624	utilitaires pour le système de fichiers exFAT entretenus par Samsung
exfat-fuse	V:2, I:37	73	pilote de système de fichiers en lecture/écriture exFAT (Microsoft) pour FUSE
xfsprogs	V:46, I:82	4651	utilitaire pour le système de fichiers XFS (SGI : IRIX)
ntfs-3g	V:197, I:510	1535	pilote de système de fichiers en lecture/écriture NTFS (Microsoft : Windows NT, …) pour FUSE
jfsutils	V:0.4, I:6.6	1104	utilitaire pour le système de fichiers JFS (IBM : AIX, OS/2)
xorriso	V:16, I:66	378	utilities for the ISO-9660 filesystem and CD/DVD writing from libburnia
wodim	V:7, I:81	898	command line CD/DVD writing tool from cdrkit
genisoimage	V:17, I:150	1567	command line ISO-9660 filesystem tool from cdrkit
hfsprogs	V:0.3, I:3.0	394	utilitaire pour les systèmes de fichiers HFS et HFS Plus (Apple : Mac OS)
zerofree	V:9, I:114	30	programme pour mettre à zéro les blocs libres des systèmes de fichiers ext2, ext3 et ext4

Table 9.20 – Liste des paquets de gestion des systèmes de fichiers

9.6.6 Création et vérification de l'intégrité d'un système de fichiers

La commande mkfs(8) permet de créer un système de fichiers sur un système Linux. La commande fsck(8) permet de vérifier l'intégrité du système de fichiers et de le réparer sur un système Linux.

Maintenant, par défaut, Debian n'utilise pas `fsck` après la création d'un système de fichier.

Attention
En général, il n'est pas sûr de faire tourner `fsck` sur un **système de fichiers monté**.

ASTUCE
Vous pouvez exécuter la commande fsck(8) sans risque sur tous les systèmes de fichiers incluant le système de fichiers racine au redémarrage avec le paramètre "`enable_periodic_fsck`" dans "`/etc/mke2fs.conf`" et le compteur de montage maximum mis à 0 en utilisant "`tune2fs -c0 /dev/`*`partition_name`*". Voir mke2fs.conf(5) et tune2fs(8).
Vous trouverez les résultats de la commande fsck(8) lancée depuis le script de démarrage dans « `/var/log/fsck/` ».

9.6.7 Optimisation du système de fichiers à l'aide des options de montage

La configuration statique de base du système de fichiers est donnée par « `/etc/fstab` ». Par exemple,

```
«file system»                    «mount point» «type» «options»    «dump» «pass»
proc                                    /proc proc   defaults          0 0
UUID=709cbe4c-80c1-56db-8ab1-dbce3146d2f7 /     ext4   errors=remount-ro 0 1
```

```
UUID=817bae6b-45d2-5aca-4d2a-1267ab46ac23 none  swap   sw               0 0
/dev/scd0                         /media/cdrom0  udf,iso9660 user,noauto  0 0
```

ASTUCE
Un UUID (consultez Section 9.6.3) peut être utilisé pour identifier un périphérique bloc au lieu des noms de périphériques blocs normaux comme « `/dev/sda1` », ou « `/dev/sda2` »,…

Depuis Linux 2.6.30, le noyau utilise par défaut le comportement fourni par l'option « `relatime` ».

Consulter fstab(5) et mount(8).

9.6.8 Optimisation du système de fichiers à l'aide du superbloc

Les caractéristiques du système de fichiers peuvent être optimisées par l'intermédiaire de son superbloc en utilisant la commande tune2fs(8).

— L'exécution de « `sudo tune2fs -l /dev/sda1` » affiche le contenu du superbloc du système de fichiers situé sur « `/dev/sda1` ».
— L'exécution de « `sudo tune2fs -c 50 /dev/sda1` » modifie la fréquence des vérifications du système de fichiers (exécution de `fsck` lors du démarrage) à 50 démarrages sur « `/dev/sda1` ».
— L'exécution de la commande « `sudo tune2fs -j /dev/sda1` » ajoute la possibilité de journalisation au système de fichiers, c'est-à-dire la conversion de système de fichiers de ext2 vers ext3 sur « `/dev/sda1` » (effectuez cela sur un système de fichiers non monté).
— L'exécution de « `sudo tune2fs -O extents,uninit_bg,dir_index /dev/sda1 && fsck -pf /dev/sda1` » le convertit de ext3 vers ext4 sur « `/dev/sda1` ». (À effectuer sur un système de fichiers non monté.)

ASTUCE
En dépit de son nom, tune2fs(8) ne fonctionne pas uniquement sur le système de fichiers ext2 mais aussi sur les systèmes de fichiers ext3 et ext4.

9.6.9 Optimisation du disque dur



AVERTISSEMENT
Veuillez vérifier votre matériel et lire la page de manuel hdparm(8) avant de jouer avec la configuration de vos disques durs parce que ce peut être assez dangereux pour l'intégrité des données.

Vous pouvez tester la vitesse de lecture d'un disque dur, par exemple « `/dev/sda` » par « `hdparm -tT /dev/sda` ». Vous pouvez accélérer certains disques (E)IDE avec « `hdparm -q -c3 -d1 -u1 -m16 /dev/sda` » en activant la prise en charge des E/S 32 bits (« (E)IDE 32-bit I/O support »), en positionnant l'indicateur d'utilisation de dma « using_dma flag », en positionnant l'indicateur de démasquage des interruptions (« interrupt-unmask flag ») et en positionnant les E/S multiples sur 15 secteurs (« multiple 16 sector I/O ») (dangereux !).

Vous pouvez tester la fonctionnalité de cache d'un disque dur, par exemple « `/dev/sda` » par « `hdparm -W /dev/sda` ». Vous pouvez désactiver le cache en écriture avec « `hdparm -W 0 /dev/sda` ».

Vous pouvez réussir à lire un CD-ROM vraiment mal pressé sur un lecteur de CD-ROM moderne rapide en le ralentissant avec « `setcd -x 2` ».

9.6.10 Optimisation du SSD

Les disques SSD (Solid State Drive) sont maintenant détectés automatiquement.

Réduisez les accès inutiles aux disques pour éviter l'usure du disque en montant « `tmpfs` » sur le chemin de données volatiles dans `/etc/fstab`.

9.6.11 Utiliser SMART pour prédire les défaillances des disques durs

Vous pouvez surveiller et enregistrer les disques durs conformes à SMART à l'aide du démon smartd(8).

1. Activez la fonctionnalité SMART dans le BIOS.
2. Installez le paquet `smartmontools`.
3. Identifiez vos disques durs en affichant la liste avec df(1).
   — Supposons que le disque dur à surveiller soit « `/dev/sda` ».
4. Contrôlez la sortie de « `smartctl -a /dev/sda` » pour voir si la fonctionnalité SMART est effectivement activée.
   — Si elle ne l'est pas, activez-la avec « `smartctl -s on -a /dev/sda` ».
5. Autorisez le fonctionnement du démon smartd(8) par l'action suivante :
   — décommentez « `start_smartd=yes` » dans le fichier « `/etc/default/smartmontools` » ;
   — relancez le démon smartd(8) avec « `sudo systemctl restart smartmontools` ».

ASTUCE
Le démon smartd(8) peut être personnalisé par l'intermédiaire du fichier `/etc/smartd.conf` y compris pour la manière d'être informé des avertissements.

9.6.12 Indication du répertoire de stockage temporaire à l'aide de $TMPDIR

Les applications créent normalement des fichiers temporaires dans le répertoire de stockage temporaire « `/tmp` ». Si « `/tmp` » ne fournit pas assez d'espace, vous pouvez indiquer un autre répertoire de stockage temporaire à l'aide de la variable `$TMPDIR` pour les programmes qui l'intègrent.

9.6.13 Étendre l'espace de stockage utile à l'aide de LVM

Les partitions créées sur le gestionnaire de volumes logiques (LVM) (« Logical Volume Manager ») (fonctionnalité de Linux) au moment de l'installation peuvent être facilement redimensionnées en y concaténant des extensions (« extents ») ou en tronquant les extensions sur plusieurs périphériques de stockage sans reconfiguration majeure du système.

9.6.14 Extension de l'espace de stockage en montant une autre partition

Si vous avez une partition vide (par exemple « `/dev/sdx` »), vous pouvez la formater avec mkfs.ext4(1) et la monter (« mount(8) ») sur un répertoire où vous avez besoin de davantage d'espace (vous devrez copier les données d'origine).

```
$ sudo mv work-dir old-dir
$ sudo mkfs.ext4 /dev/sdx
$ sudo mount -t ext4 /dev/sdx work-dir
$ sudo cp -a old-dir/* work-dir
$ sudo rm -rf old-dir
```

ASTUCE
Vous pouvez aussi monter un fichier image de disque vide (consultez Section 9.7.5) en tant que périphérique de rebouclage (consultez Section 9.7.3). L'utilisation réelle du disque croît avec les données réellement enregistrées.

9.6.15 Extension de l'espace de stockage en remontant un autre répertoire

Si vous avez un répertoire vide (par exemple, « `/chemin/vers/répertoire-temporaire` ») sur une autre partition avec de l'espace disponible, vous pouvez le remonter avec l'option « `--bind` » vers un répertoire (par exemple « `répertoire-travail` ») où vous avez besoin de place.

```
$ sudo mount --bind /path/to/emp-dir work-dir
```

9.6.16 Extension de l'espace de stockage utilisable en montant en superposition (overlay) un autre répertoire

Si vous avez de l'espace utilisable sur une autre partition (p.ex., « `/chemin/vers/espace_vide` » et « `/chemin/répert` vous pouvez y créer un répertoire et l'ajouter à un répertoire ancien (p.ex., « `/chemin/vers/répertoire_ancien` ») où vous avez besoin de place en utilisant OverlayFS pour les noyaux Linux 3.18 ou plus récents (Debian Stretch 9.0 ou plus récente).

```
$ sudo mount -t overlay overlay \
  -olowerdir=/path/to/old-dir,upperdir=/path/to/empty,workdir=/path/to/work
```

Ici, « `/chemin/vers/espace_vide` » et « `/chemin/vers/répertoire_travail` » doivent être des partitions autorisées en écriture et lecture pour écrire sur « `/chemin/vers/répertoire_ancien` ».

9.6.17 Extension de l'espace utilisable à l'aide de liens symboliques



Attention
Il s'agit d'une méthode obsolète. Certains logiciels peuvent ne pas fonctionner correctement avec un « lien symbolique vers un répertoire ». Utilisez plutôt les approches de « montage » décrites ci-dessus.

Si vous avez un répertoire vide (par exemple, « `/chemin/vers/répertoire-temporaire` ») sur une autre partition avec de l'espace disponible, vous pouvez créer un lien symbolique vers ce répertoire avec ln(8).

```
$ sudo mv work-dir old-dir
$ sudo mkdir -p /path/to/emp-dir
$ sudo ln -sf /path/to/emp-dir work-dir
$ sudo cp -a old-dir/* work-dir
$ sudo rm -rf old-dir
```



AVERTISSEMENT
N'utilisez pas de « liens symboliques vers un répertoire » pour un répertoire géré par le système comme par exemple « `/opt` ». De tels liens symboliques peuvent être écrasés lors de la mise à niveau du système.

9.7 Le fichier image du disque

Nous discutons ici des manipulations sur l'image disque.

9.7.1 Créer le fichier image du disque

The disk image file, "`disk.img`", of an unmounted device, e.g., the second SCSI or serial ATA drive "`/dev/sdb`", can be made by one of the following.

```
# dd if=/dev/sdb of=disk.img; sync
```

```
# cp /dev/sdb disk.img ; sync
```

```
# cat /dev/sdb > disk.img ; sync
```

L'image disque du master boot record (MBR) (secteur principal d'amorçage) (consultez Section 9.6.2) qui se trouve sur le premier secteur du disque primaire IDE peut être faite en utilisant dd(1) comme suit :

```
# dd if=/dev/sda of=mbr.img bs=512 count=1
# dd if=/dev/sda of=mbr-nopart.img bs=446 count=1
# dd if=/dev/sda of=mbr-part.img skip=446 bs=1 count=66
```

— « `mbr.img` » : MBR avec la table des partitions
— « `mbr-nopart.img` » : MBR sans la table des partitions
— « `mbr-part.img` » : table de partition du MBR seul

Si vous réalisez une image d'une partition du disque d'origine, remplacez « `/dev/sda` » par « `/dev/sda1` », etc.

9.7.2 Écrire directement sur le disque

The disk image file, "`disk.img`" can be written to an unmounted device, e.g., the second SCSI drive "`/dev/sdb`" with matching size, by one of the following.

```
# dd if=disk.img of=/dev/sdb ; sync
```

```
# cp disk.img /dev/sdb ; sync
```

```
# cat disk.img >/dev/sdb ; sync
```

De la même manière, le fichier image de la partition du disque, « `partition.img` » peut être écrit sur une partition non montée, par exemple, la première partition du second disque SCSI « `/dev/sdb1` » avec la taille correspondante comme suit :

```
# dd if=partition.img of=/dev/sdb1 ; sync
```

9.7.3 Monter le fichier image du disque

L'image disque « `partition.img` », qui contient une partition image unique, peut être monté et démonté en utilisant le périphérique de rebouclage (loop device) de la manière suivante :

```
# losetup --show -f partition.img
/dev/loop0
# mkdir -p /mnt/loop0
# mount -t auto /dev/loop0 /mnt/loop0
...hack...hack...hack
# umount /dev/loop0
# losetup -d /dev/loop0
```

Cela peut être simplifié de la manière suivante :

```
# mkdir -p /mnt/loop0
# mount -t auto -o loop partition.img /mnt/loop0
...hack...hack...hack
# umount partition.img
```

Chaque partition de l'image disque « `disk.img` » contenant plusieurs partitions peut être montée en utilisant le périphérique de rebouclage (loop device).

```
# losetup --show -f -P disk.img
/dev/loop0
# ls -l /dev/loop0*
brw-rw---- 1 root disk   7,  0 Apr  2 22:51 /dev/loop0
brw-rw---- 1 root disk 259, 12 Apr  2 22:51 /dev/loop0p1
brw-rw---- 1 root disk 259, 13 Apr  2 22:51 /dev/loop0p14
brw-rw---- 1 root disk 259, 14 Apr  2 22:51 /dev/loop0p15
# fdisk -l /dev/loop0
Disk /dev/loop0: 2 GiB, 2147483648 bytes, 4194304 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 6A1D9E28-C48C-2144-91F7-968B3CBC9BD1

Device         Start     End Sectors  Size Type
/dev/loop0p1  262144 4192255 3930112  1.9G Linux root (x86-64)
/dev/loop0p14   2048    8191    6144    3M BIOS boot
/dev/loop0p15   8192  262143  253952  124M EFI System

Partition table entries are not in disk order.
# mkdir -p /mnt/loop0p1
# mkdir -p /mnt/loop0p15
# mount -t auto /dev/loop0p1 /mnt/loop0p1
# mount -t auto /dev/loop0p15 /mnt/loop0p15
# mount |grep loop
/dev/loop0p1 on /mnt/loop0p1 type ext4 (rw,relatime)
/dev/loop0p15 on /mnt/loop0p15 type vfat (rw,relatime,fmask=0002,dmask=0002,allow_utime ↩
    =0020,codepage=437,iocharset=ascii,shortname=mixed,utf8,errors=remount-ro)
...hack...hack...hack
# umount /dev/loop0p1
# umount /dev/loop0p15
# losetup -d /dev/loop0
```

En remplacement, des effets similaires peuvent être obtenus en utilisant les périphériques device mapper créés par kpartx(8) du paquet `kpartx` de la manière suivante :

```
# kpartx -a -v disk.img
add map loop0p1 (253:0): 0 3930112 linear 7:0 262144
```

```
add map loop0p14 (253:1): 0 6144 linear 7:0 2048
add map loop0p15 (253:2): 0 253952 linear 7:0 8192
# fdisk -l /dev/loop0
Disk /dev/loop0: 2 GiB, 2147483648 bytes, 4194304 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 6A1D9E28-C48C-2144-91F7-968B3CBC9BD1

Device         Start     End Sectors  Size Type
/dev/loop0p1  262144 4192255 3930112  1.9G Linux root (x86-64)
/dev/loop0p14   2048    8191    6144    3M BIOS boot
/dev/loop0p15   8192  262143  253952  124M EFI System

Partition table entries are not in disk order.
# ls -l /dev/mapper/
total 0
crw------- 1 root root 10, 236 Apr  2 22:45 control
lrwxrwxrwx 1 root root       7 Apr  2 23:19 loop0p1 -> ../dm-0
lrwxrwxrwx 1 root root       7 Apr  2 23:19 loop0p14 -> ../dm-1
lrwxrwxrwx 1 root root       7 Apr  2 23:19 loop0p15 -> ../dm-2
# mkdir -p /mnt/loop0p1
# mkdir -p /mnt/loop0p15
# mount -t auto /dev/mapper/loop0p1 /mnt/loop0p1
# mount -t auto /dev/mapper/loop0p15 /mnt/loop0p15
# mount |grep loop
/dev/loop0p1 on /mnt/loop0p1 type ext4 (rw,relatime)
/dev/loop0p15 on /mnt/loop0p15 type vfat (rw,relatime,fmask=0002,dmask=0002,allow_utime ↩
    =0020,codepage=437,iocharset=ascii,shortname=mixed,utf8,errors=remount-ro)
...hack...hack...hack
# umount /dev/mapper/loop0p1
# umount /dev/mapper/loop0p15
# kpartx -d disk.img
```

9.7.4 Nettoyage d’un fichier image du disque

Un fichier image disque « `disk.img` » peut être nettoyé de tous les fichiers supprimés pour donner une image propre « `new.img` » de la manière suivante :

```
# mkdir old; mkdir new
# mount -t auto -o loop disk.img old
# dd bs=1 count=0 if=/dev/zero of=new.img seek=5G
# mount -t auto -o loop new.img new
# cd old
# cp -a --sparse=always ./ ../new/
# cd ..
# umount new.img
# umount disk.img
```

Si « `disk.img` » est un système de fichiers ext2, ext3 ou ext4, vous pouvez aussi utiliser zerofree(8) du paquet `zerofree` de la manière suivante :

```
# losetup --show -f disk.img
/dev/loop0
# zerofree /dev/loop0
# cp --sparse=always disk.img new.img
# losetup -d /dev/loop0
```

9.7.5 Réaliser le fichier image d'un disque vide

Le fichier image du disque vide « `disk.img` », qui pourra s'étendre jusqu'à 5Gio peut être fait en utilisant dd(1) comme suit :

```
$ dd bs=1 count=0 if=/dev/zero of=disk.img seek=5G
```

Au lieu d'utiliser dd(1), la fonction fallocate(8) spécialisée peut être utilisée ici.

Vous pouvez créer un système de fichiers ext4 sur cette image disque « `disk.img` » en utilisant le périphérique de rebouclage (loop device) de la manière suivante :

```
# losetup --show -f disk.img
/dev/loop0
# mkfs.ext4 /dev/loop0
...hack...hack...hack
# losetup -d /dev/loop0
$ du  --apparent-size -h disk.img
5.0G  disk.img
$ du -h disk.img
83M disk.img
```

Pour « `disk.img` », sa taille de fichier est de 5.0 Gio et son utilisation disque est uniquement de 83Mio. Cette discordance est possible car ext4 sait maintenir un fichier creux (sparse).

ASTUCE
L'utilisation réelle sur le disque du fichier creux croît au fur et à mesure qu'on y écrit des données .

En utilisant des opérations similaires sur les périphériques créés par loop device ou les périphériques device mapper comme dans Section 9.7.3, vous pouvez partitionner cette image disque « `disk.img` » en utilisant parted(8) ou fdisk(8), et y créer un système de fichiers en utilisant mkfs.ext4(8), mkswap(8), etc.

9.7.6 Créer un fichier image ISO9660

ASTUCE
Both genisoimage(1) provided by cdrkit and xorrisofs(1) provided by Libburnia share the same command syntax except for the command name.

On peut faire le fichier image ISO9660, « `cd.iso` », depuis l'arborescence de répertoire source située à « `répertoire_sou`
en utilisant genisoimage(1) fourni parcdrkit de la manière suivante :

```
#  genisoimage -r -J -T -V volume_id -o cd.iso source_directory
```

De la même manière, on peut créer le fichier image ISO9660 amorçable « `cdboot.iso` » depuis une arborescence comme celle de `debian-installer` située en « `source_directory` », de la manière suivante :

```
#  genisoimage -r -o cdboot.iso -V volume_id \
   -b isolinux/isolinux.bin -c isolinux/boot.cat \
   -no-emul-boot -boot-load-size 4 -boot-info-table source_directory
```

Ici, le chargeur d'amorçage Isolinux (consultez Section 3.1.2) est utilisé pour l'amorçage.

Vous pouvez calculer la valeur de la somme md5 (md5sum) et construire des image ISO9660 directement depuis un lecteur de CD-ROM de la manière suivante :

```
$ isoinfo -d -i /dev/cdrom
CD-ROM is in ISO 9660 format
...
Logical block size is: 2048
Volume size is: 23150592
...
# dd if=/dev/cdrom bs=2048 count=23150592 conv=notrunc,noerror | md5sum
# dd if=/dev/cdrom bs=2048 count=23150592 conv=notrunc,noerror > cd.iso
```



AVERTISSEMENT
Vous devrez prendre garde d'éviter le bogue de lecture anticipée du système de fichiers ISO9660 de Linux comme ci-dessus afin d'obtenir les résultats corrects.

9.7.7 Écriture directe sur CD/DVD-R/RW

ASTUCE
DVD is only a large CD to wodim(1) provided by cdrkit and xorrecord(1) provided by Libburnia. These commands share the same command syntax except for the command name.

Vous pouvez rechercher un périphérique utilisable comme suit :

```
# wodim --devices
```

Le CD-R vierge est alors inséré dans le graveur de CD et le fichier image ISO9660 « `cd.iso` » est écrit vers le périphérique, par exemple, « `/dev/sda` » en utilisant wodim(1) de la manière suivante :

```
# wodim -v -eject dev=/dev/sda cd.iso
```

Si un CD-RW est utilisé à la place d'un CD-R, faites alors ce qui suit :

```
# wodim -v -eject blank=fast dev=/dev/sda cd.iso
```

ASTUCE
Si votre système de bureau monte automatiquement le CD, démontez-le par la commande « `sudo unmount /dev/sda` » depuis un terminal avant d'utiliser wodim(1).

9.7.8 Monter le fichier image ISO9660

Si « `cd.iso` » contient une image ISO9660, ce qui suit permet alors de le monter manuellement sur « `/cdrom` » :

```
# mount -t iso9660 -o ro,loop cd.iso /cdrom
```

ASTUCE
Les systèmes de bureau modernes peuvent monter automatiquement les supports amovibles tels que les CD formatés en ISO9960 (consultez Section 10.1.7).

9.8 Les données binaires

Nous allons ici discuter de la manipulation directe des données binaires sur le support d'enregistrement.

9.8.1 Afficher et éditer des données binaires

La méthode la plus basique pour visualiser des données binaires est d'utiliser la commande « `od -t x1` ».

paquet	popularité	taille	description
`coreutils`	V:914, I:1000	19751	paquet de base utilisant od(1) pour vider des fichiers (HEX, ASCII, OCTAL, …)
`bsdmainutils`	V:4, I:112	17	paquets utilitaires qui utilisent hd(1) pour vider les fichiers (HEX, ASCII, OCTAL, …)
`hexedit`	V:0.7, I:7.4	70	éditeurs et visualisateurs binaires (HEX, ASCII)
`ghex`	V:0.7, I:7.2	2525	éditeur hexadécimal complet (GNOME)
`okteta`	V:1, I:11	1591	éditeur hexadécimal complet (KDE4)
`ncurses-hexedit`	V:0.1, I:1.1	130	éditeur et visualisateur binaire (HEX, ASCII, EBCDIC)

Table 9.21 – Liste des paquets permettant de visualiser et d'éditer des données binaires

ASTUCE
HEX est utilisé comme l'acronyme du format hexadécimal en base 16. OCTAL désigne le format octal en base 8. ASCII est employé pour Code américain standard pour l'échange d'informations (« American Standard Code for Information Interchange ») c'est-à-dire le code pour texte normal en anglais. EBCDIC signifie Code d'échange étendu décimal codé binaire (« Extended Binary Coded Decimal Interchange Code »), il est utilisé avec par les systèmes d'exploitation des mainframe IBM.

9.8.2 Manipulating files without mounting the filesystem

There are tools to read and write files without mounting the filesystems.

paquet	popularité	taille	description
`mtools`	V:7, I:51	404	utilities for MSDOS filesystems without mounting them
`hfsplus`	V:2, I:20	248	utilities for HFS+ filesystems without mounting them

Table 9.22 – List of packages to manipulate files without mounting the filesystems

9.8.3 Redondance des données

Les systèmes s'appuyant sur le RAID logiciel offert par le noyau Linux permettent une redondance des données au niveau du système de fichiers du noyau afin d'obtenir un haut niveau de fiabilité du système de stockage.

Il existe aussi des outils pour ajouter des données de redondance aux fichiers au niveau du programme applicatif permettant d'obtenir de hauts niveaux de fiabilité de stockage.

paquet	popularité	taille	description
par2	V:10, I:115	965	Parity Archive Volume Set, pour vérifier et réparer des fichiers
dvdisaster	V:0.1, I:1.1	1422	protection des supports CD et DVD contre les pertes de données, les rayures et le vieillissement
dvbackup	I:0.04	413	outil de sauvegarde utilisant des caméscopes MiniDV

Table 9.23 – Liste d'outils pour ajouter des données de redondance aux fichiers

paquet	popularité	taille	description
testdisk	V:2, I:25	529	utilitaires pour l'examen de partitions et la récupération de disque
magicrescue	V:0.2, I:1.6	257	utilitaire pour la récupération de fichiers et de recherche des octets magiques
scalpel	V:0.2, I:2.2	89	récupérateur de fichiers sobre de haute performance
myrescue	V:0.2, I:1.9	81	récupérer des données depuis des disques endommagés
extundelete	V:0.4, I:7.1	152	utilitaire pour récupérer des fichiers effacés d'un système de fichiers ext3/4
ext4magic	V:0.3, I:3.5	235	utilitaire pour récupérer des fichiers effacés d'un système de fichiers ext3/4
ext3grep	V:0.1, I:1.7	299	outil pour aider à la récupération de fichiers effacés sur un système de fichiers ext3
scrounge-ntfs	V:0.1, I:1.6	49	programme de récupération de données pour les systèmes de fichiers NTFS
gzrt	V:0.02, I:0.43	34	boîte à outils de récupération gzip
sleuthkit	V:3, I:24	1130	outil pour autopsie (« forensics analysis« ». (Sleuthkit)
autopsy	V:0.1, I:1.1	1026	interface graphique à SleuthKit
foremost	V:0.3, I:3.7	101	application d'autopsie pour la récupération de données
guymager	V:0.09, I:0.70	1049	outil de création d'image d'autopsie basée sur Qt
dcfldd	V:0.2, I:2.4	113	version améliorée de dd pour les autopsies et la sécurité

Table 9.24 – Liste de paquets pour la récupération de données et l'analyse par autopsie

9.8.4 Récupération de fichiers de données et analyse par autopsie

Il y a des outils pour la récupération des données et l'analyse par autopsie.

ASTUCE
Vous pouvez annuler l'effacement de fichiers sur un système de fichiers ext2 en utilisant les commandes `list_deleted_inodes` et `undel` de debugfs(8) dans le paquet `e2fsprogs`.

9.8.5 Éclater un gros fichier en petits fichiers

Lorsque les données ont un volume trop important pour pouvoir être sauvegardée dans un seul fichier, vous pouvez en sauvegarder le contenu après l'avoir éclaté en morceaux de, par exemple, 2000Mio et réassembler ces morceaux par la suite sous la forme du fichier d'origine.

```
$ split -b 2000m large_file
$ cat x* >large_file
```



Attention
Assurez-vous ne pas avoir de nom de fichier commençant par « x » afin d'éviter des plantages de nom.

9.8.6 Effacer le contenu d'un fichier

Pour effacer le contenu d'un fichier comme, par exemple, un fichier journal, n'utilisez pas la commande rm(1) pour supprimer le fichier et recréer ensuite un fichier vide parce qu'on peut encore accéder au fichier dans l'intervalle entre les commandes. Voici la manière sûre d'effacer le contenu d'un fichier :

```
$ :>file_to_be_cleared
```

9.8.7 Fichiers fictifs

Les commandes suivantes créent des fichiers factices ou vides.

```
$ dd if=/dev/zero    of=5kb.file bs=1k count=5
$ dd if=/dev/urandom of=7mb.file bs=1M count=7
$ touch zero.file
$ : > alwayszero.file
```

Vous obtiendrez les fichiers suivants :

— « `5kb.file` » avec 5K de zéros ;
— « `7mb.file` » avec 7Mo de données aléatoires ;
— « `zero.file` » devrait être un fichier de 0 octet. S'il existait, son `mtime` est mis à jour alors que son contenu et sa taille sont conservés ;
— « `alwayszero.file` fait toujours 0 octet. S'il existait son `mtime` est mis à jour et son contenu vidé.

9.8.8 Effacer l'ensemble du disque dur

There are several ways to completely erase data from an entire hard disk like device, e.g., USB flash drive at "`/dev/sda`".



Attention
Check your USB flash drive location with mount(8) first before executing commands here. The device pointed by "`/dev/sda`" may be SCSI hard disk or serial-ATA hard disk where your entire system resides.

Effacer tout le contenu du disque en réinitialisant toutes les données à 0 avec la commande suivante :

```
# dd if=/dev/zero of=/dev/sda
```

Tout effacer en écrasant les données existantes par des données aléatoires par la commande suivante :

```
# dd if=/dev/urandom of=/dev/sda
```

Effacer de manière très efficace toutes les données en les écrasant avec des données aléatoires par la commande suivante :

```
# shred -v -n 1 /dev/sda
```

Vous pouvez également utiliser badblocks(8) avec l'option `-t random`.

Comme dd(1) est disponible depuis l'interpréteur de commandes de nombreux CD amorçables de Linux tels que le CD de l'installateur Debian, vous pouvez effacer complètement votre système installé en lançant la commande d'effacement du disque dur du système, par exemple, « `/dev/sda` », « `/dev/sda` », etc. depuis un tel média.

9.8.9 Effacer l'ensemble du disque dur

Unused area on an hard disk (or USB flash drive), e.g. "`/dev/sdb1`" may still contain erased data themselves since they are only unlinked from the filesystem. These can be cleaned by overwriting them.

```
# mount -t auto /dev/sdb1 /mnt/foo
# cd /mnt/foo
# dd if=/dev/zero of=junk
dd: writing to `junk': No space left on device
...
# sync
# umount /dev/sdb1
```



AVERTISSEMENT
This is usually good enough for your USB flash drive. But this is not perfect. Most parts of erased filenames and their attributes may be hidden and remain in the filesystem.

9.8.10 Récupérer des fichiers supprimés mais encore ouverts

Même si vous avez accidentellement supprimé un fichier, tant que ce fichier est en cours d'utilisation par une application quelconque, (en mode lecture ou écriture), il est possible de récupérer un tel fichier.

Essayez, par exemple, ce qui suit :

```
$ echo foo > bar
$ less bar
$ ps aux | grep ' less[ ]'
bozo    4775  0.0  0.0  92200   884 pts/8    S+   00:18   0:00 less bar
$ rm bar
$ ls -l /proc/4775/fd | grep bar
lr-x------ 1 bozo bozo 64 2008-05-09 00:19 4 -> /home/bozo/bar (deleted)
$ cat /proc/4775/fd/4 >bar
$ ls -l
-rw-r--r-- 1 bozo bozo 4 2008-05-09 00:25 bar
$ cat bar
foo
```

Exécutez sur un autre terminal (lorsque vous avez le paquet `lsof` installé) comme suit :

```
$ ls -li bar
2228329 -rw-r--r-- 1 bozo bozo 4 2008-05-11 11:02 bar
$ lsof |grep bar|grep less
less 4775 bozo 4r REG 8,3 4 2228329 /home/bozo/bar
$ rm bar
$ lsof |grep bar|grep less
less 4775 bozo 4r REG 8,3 4 2228329 /home/bozo/bar (deleted)
$ cat /proc/4775/fd/4 >bar
$ ls -li bar
2228302 -rw-r--r-- 1 bozo bozo 4 2008-05-11 11:05 bar
$ cat bar
foo
```

9.8.11 Rechercher tous les liens physiques

Les fichiers ayant des liens physiques peuvent être identifiés par « `ls -li` ».

```
$ ls -li
total 0
2738405 -rw-r--r-- 1 root root 0 2008-09-15 20:21 bar
2738404 -rw-r--r-- 2 root root 0 2008-09-15 20:21 baz
2738404 -rw-r--r-- 2 root root 0 2008-09-15 20:21 foo
```

« `tutu` » et « `toto` » ont tous les deux un nombre de liens égal à ”« 2 » (>1), ce qui indique qu’ils ont des liens physiques. Leur numéro d’inœud commun est « 2738404 ». Cela signifie qu’ils représentent le même fichier lié par des liens physiques. Si vous n’arrivez pas à trouver de fichiers liés par des liens physiques, vous pouvez les rechercher parinœud, par exemple « 2738404 », de la manière suivante :

```
# find /path/to/mount/point -xdev -inum 2738404
```

9.8.12 Consommation d’espace disque invisible

Tous les fichiers supprimés mais ouverts prennent de l’espace disque même s’ils ne sont pas visibles par la commande du(1) normale. On peut en afficher la liste avec leur taille par la commande suivante :

```
# lsof -s -X / |grep deleted
```

9.9 Astuces de chiffrement des données

Avec un accès physique à votre PC, n'importe qui peut facilement obtenir les privilèges de l'administrateur et accéder à tous les fichiers de votre PC (consultez Section 4.6.4). Cela signifie qu'un système avec un mot de passe de connexion ne permet pas de sécuriser vos données personnelles ou sensibles en cas de vol de votre PC. Vous devez déployer des technologies de chiffrements des données pour assurer cette protection. Bien que GNU privacy guard (consultez Section 10.3) puisse chiffrer des fichiers, il demande quelques efforts de la part de l'utilisateur.

Dm-crypt facilite le chiffrement automatique des données à l'aide des modules natifs du noyau Linux avec un minimum d'efforts de l'utilisateur en utilisant device-mapper.

paquet	popularité	taille	description
`cryptsetup`	V:35, I:77	476	utilitaires pour chiffrer un périphérique en mode bloc (dm-crypt / LUKS)
`cryptmount`	V:1.8, I:2.2	236	utilitaires pour chiffrer un périphérique en mode bloc (dm-crypt / LUKS), l'accent étant mis sur le montage et le démontage par un utilisateur normal
`fscrypt`	V:0.4, I:1.1	6844	utilitaires pour le chiffrement du système de fichiers Linux (fscrypt)
`libpam-fscrypt`	V:0.32, I:0.35	6941	module PAM pour le chiffrement du système de fichiers Linux (fscrypt)

Table 9.25 – Liste d'utilitaires de chiffrement des données



Attention
Le chiffrement des données a un coût en matière de temps processeur, etc. Les données chiffrées deviennent inaccessibles si leur mot de passe est perdu. Veuillez peser les avantages et le coût.

Note
Le système Debian dans son ensemble peut être installé sur un disque chiffré par l'installateur debian (lenny ou plus récent) en utilisant dm-crypt/LUKS et initramfs.

ASTUCE
Consultez Section 10.3 pour un utilitaire de chiffrement de l'espace utilisateur : GNU Privacy Guard.

9.9.1 Chiffrement des disques amovibles à l'aide de dm-crypt/LUKS

You can encrypt contents of removable mass devices, e.g. USB flash drive on "`/dev/sdx`", using dm-crypt/LUKS. You simply format it as the following.

```
# fdisk /dev/sdx
... "n" "p" "1" "return" "return" "w"
# cryptsetup luksFormat /dev/sdx1
...
# cryptsetup open /dev/sdx1 secret
...
# ls -l /dev/mapper/
total 0
crw-rw---- 1 root root  10, 60 2021-10-04 18:44 control
lrwxrwxrwx 1 root root       7 2021-10-04 23:55 secret -> ../dm-0
```

```
# mkfs.vfat /dev/mapper/secret
...
# cryptsetup close secret
```

Ensuite, il peut être monté comme un disque normal sur « `/media/nom_utilisateur/étiquette_disque` », sauf pour la demande de mot de passe (consulter la Section 10.1.7) dans un environnement de bureau moderne en utilisant le paquet `udisks2`. La différence est que toutes les données qui y sont écrites sont chiffrées. La saisie du mot de passe peut être automatisée à l'aide du trousseau de clés (consulter la Section 10.3.6).

Vous pouvez également formater les médias dans différents systèmes de fichiers, par exemple, ext4 avec « `mkfs.ext4 /dev/mapper/sdx1` ». Si btrfs est utilisé à la place, le paquet `udisks2-btrfs` doit être installé. Pour ces systèmes de fichiers, il peut être nécessaire de configurer les propriétaires et les autorisations des fichiers.

9.9.2 Monter des disques amovibles chiffrés à l'aide de dm-crypt/LUKS

Par exemple, une partition de disque chiffrée créée avec dm-crypt/LUKS sur « `/dev/sdc5` » par l'installateur Debian peut être montée sur « `/mnt` » comme suit :

```
$ sudo cryptsetup open /dev/sdc5 ninja --type luks
Enter passphrase for /dev/sdc5: ****
$ sudo lvm
lvm> lvscan
  inactive          '/dev/ninja-vg/root' [13.52 GiB] inherit
  inactive          '/dev/ninja-vg/swap_1' [640.00 MiB] inherit
  ACTIVE            '/dev/goofy/root' [180.00 GiB] inherit
  ACTIVE            '/dev/goofy/swap' [9.70 GiB] inherit
lvm> lvchange -a y /dev/ninja-vg/root
lvm> exit
  Exiting.
$ sudo mount /dev/ninja-vg/root /mnt
```

9.10 Le noyau

Debian distribue des noyaux Linux modulaires sous forme de paquets pour les architectures prises en compte.

Si vous lisez cette documentation, vous n'avez probablement pas besoin de compiler le noyau Linux par vous-même.

9.10.1 Paramètres du noyau

De nombreuses fonctionnalités du noyau peuvent être configurées par l'intermédiaire de paramètres du noyau de la manière suivante :

— Paramètres du noyau initialisés par le gestionnaire d'amorçage (consultez Section 3.1.2)
— Paramètres du noyau modifiés par sysctl(8) lors du fonctionnement du système pour ceux auxquels on a accès par l'intermédiaire de sysfs (consultez Section 1.2.12)
— Paramètres des modules définis par les paramètres de modprobe(8) lors de l'activation d'un module (consultez Section 9.7.3)

Consulter le « Guide de l'utilisateur et de l'administrateur du noyau Linux -> Paramètres de ligne de commande du noyau » pour plus de détails.

9.10.2 En-têtes du noyau

La plupart des **programmes normaux** n'ont pas besoin des en-têtes du noyau et peuvent de fait être cassés si vous les utilisez directement pour la compilation. Ils devront être compilés avec les en-têtes se trouvant dans « `/usr/include/linux` » et « `/usr/include/asm` » qui sont fournis, sur les systèmes Debian, par le paquet `libc6-dev` (créé à partir du paquet source `glibc`).

Note
Pour compiler certains programmes spécifiques au noyau, tels que les modules du noyau, à partir de la source externe et du démon automonteur (`amd`), vous devez inclure le chemin d'accès aux en-têtes de noyau correspondants, par exemple « `-I/usr/src/linux-particular-version/include/` », à votre ligne de commande.

9.10.3 Compiler le noyau et les modules associés

Debian a sa propre manière de compiler le noyau et les modules associés.

paquet	popularité	taille	description
`build-essential`	V:20, I:508	17	paquets essentiels pour la construction de paquets Debian : `make`, `gcc`, …
`bzip2`	V:163, I:973	113	utilitaires de compression et de décompression des fichiers bz2
`libncurses-dev`	V:16, I:148	2380	bibliothèques de développement et documentations pour ncurses
`git`	V:442, I:630	52512	git : système distribué de gestion de versions utilisé par le noyau de Linux
`fakeroot`	V:34, I:508	225	fournit l'environnement fakeroot pour construire le paquet sans être administrateur (« root »)
`initramfs-tools`	V:525, I:989	49	outil pour construire une image mémoire initiale (« initramds ») (spécifique à Debian)
`dkms`	V:81, I:147	235	prise en charge dynamique des modules du noyau (dynamic kernel module support : DKMS) (générique)
`module-assistant`	V:1, I:11	395	outil d'aide pour empaqueter un module (spécifique à Debian)
`devscripts`	V:5, I:30	2790	scripts d'assistance pour un responsable de paquet Debian (spécifique à Debian)

Table 9.26 – Liste des paquets-clés à installer pour la compilation du noyau sur un système Debian

Si vous utilisez un `initrd` dans Section 3.1.2, veuillez lire les informations correspondantes dans initramfs-tools(7), update-initramfs(8), mkinitramfs(8) et initramfs.conf(5).



AVERTISSEMENT
Ne mettez pas de liens symboliques vers le répertoire de l'arborescence des source (par exemple « `/usr/src/linux*` ») depuis « `/usr/include/linux` » et « `/usr/include/asm` » lors de la compilation des sources du noyau de Linux. (Certains documents périmés le suggèrent.)

Note
Lors de la compilation du dernier noyau de Linux sous un système Debian `stable`, l'utilisation des derniers outils rétroportés depuis la distribution Debian `unstable` peuvent être nécessaires.
module-assistant(8) (ou sa forme courte `m-a`) aide les utilisateurs à construire et installer facilement des paquets de module pour un ou plusieurs noyaux personnalisés.
Le gestionnaire de modules dynamique du noyau (« dynamic kernel module support (DKMS) » est une nouvelle architecture indépendante de la distribution conçue pour permettre la mise à jour de modules individuels du noyau sans modifier l'ensemble du noyau. Cela est utilisé pour la maintenance de modules hors arborescence. Cela rend aussi très facile la reconstruction des modules après la mise à niveau des noyaux.

9.10.4 Compiler les sources du noyau : recommandations de l'équipe en charge du noyau Debian

Pour construire des paquets binaires d'un noyau personnalisé à partir des sources du noyau amont, vous devriez utiliser la cible « `deb-pkg` » fournie pour cela.

```
$ sudo apt-get build-dep linux
$ cd /usr/src
$ wget https://mirrors.edge.kernel.org/pub/linux/kernel/v7.x/linux-7.3.5.tar.xz
$ tar --xz -xvf linux-7.3.5.tar.xz
$ cd linux-7.3.5
$ cp /boot/config-7.3.5.config
$ make menuconfig
 ...
$ make deb-pkg
```

ASTUCE
Le paquet linux-source-*version* fournit les sources du noyau Linux avec les correctifs Debian en tant que « `/usr/src/linux-`*`version`*`.tar.bz2` ».

Pour construire des paquets binaires particuliers à partir du paquet source Debian, vous devriez utiliser les cibles « `binary-arch_`*`architecture_jeu_de_fonctionnalités_saveur`* » dans « `debian/rules.gen` ».

```
$ sudo apt-get build-dep linux
$ apt-get source linux
$ cd linux-7.*
$ fakeroot make -f debian/rules.gen binary-arch_i386_none_686
```

Consultez les renseignements complémentaires :

— wiki Debian : KernelFAQ ;
— wiki Debian : DebianKernel ;
— Manuel du noyau Linux pour Debian : https://kernel-handbook.debian.net.

9.10.5 Pilotes de matériel et microprogramme

Le pilote de matériel est le code s'exécutant sur le CPU principal du système cible. La plupart des pilotes de matériel sont maintenant disponibles sont forme de logiciels libres et font partie des paquets normaux de Debian pour le noyau dans la section `main`.

— Pilote de processeur graphique
  — Pilote Intel (`main`)
  — Pilote AMD ou ATI (`main`)

  - Pilote NVIDIA (`main` pour le pilote nouveau et `non-free` pour les pilotes binaires pris en charge par le constructeur)

Le microprogramme est le code ou les données chargées sur le périphérique lié au système cible (par exemple le microcode de processeur, le code de rendu exécuté sur les processeurs graphiques, ou les données de FPGA ou de CPLD, etc.) Certains paquets de microprogramme sont disponibles sous forme de logiciel libre, mais beaucoup de paquets de microprogramme ne le sont pas car ils contiennent des données binaires sans les sources. L'installation de ces données de microprogramme est essentielle pour le bon fonctionnement du périphérique.

- Paquets de données de microprogramme contenant les données chargées dans la mémoire volatile de l'équipement cible :
  - `firmware-linux-free` (`main`)
  - `firmware-linux-nonfree` (`non-free-firmware`)
  - `firmware-linux-*` (`non-free-firmware`)
  - `*-firmware` (`non-free-firmware`)
  - `intel-microcode` (`non-free-firmware`)
  - `amd64-microcode` (`non-free-firmware`)
- Paquets de mise à jour du microprogramme, qui mettent à jour les données sur la mémoire non volatile du périphérique cible :
  - fwupd (`main`) : démon de mise à jour de microprogramme qui télécharge les données du microprogramme à partir de Linux Vendor Firmware Service
  - gnome-firmware (`main`) : interface en GTK pour fwupd
  - plasma-discover-backend-fwupd (`main`) : interface en Qt pour fwupd

Veuillez noter que l'accès aux paquets `non-free-firmware` est fourni par le média d'installation officiel pour offrir une expérience d'installation fonctionnelle à l'utilisateur depuis Debian 12 Bookworm. La section `non-free-firmware` est décrite dans la Section 2.1.5.

Veuillez également noter que les données du microprogramme téléchargées par fwupd à partir de Linux Vendor Firmware Service et chargées dans le noyau Linux en cours d'exécution peuvent être `non libres`.

9.11 Système virtualisé

L'utilisation d'un système virtualisé permet de faire tourner simultanément plusieurs instances du système sur une plateforme unique.

ASTUCE
Consulter le wiki de Debian à propos de la virtualisation du système.

9.11.1 Outils de virtualisation et d'émulation

Il existe plusieurs plateformes d'outils de virtualisation et d'émulation :

- paquets d'émulation matérielle complète, tels que ceux installés par le métapaquet games-emulator ;
- émulation principalement au niveau du processeur avec certaines émulations de périphérique d'E/S, telle que QEMU ;
- virtualisation principalement au niveau du processeur avec certaines émulations de périphérique d'E/S, telle que KVM (machine virtuelle basée sur le noyau) ;
- virtualisation de conteneurs au niveau du système d'exploitation avec prise en charge au niveau du noyau, telle que LXC (Linux Containers), Docker, systemd-nspawn(1), etc. ;
- virtualisation de l'accès au système de fichiers au niveau du système d'exploitation avec le contournement de l'appel système de bibliothèque pour le chemin d'accès au fichier, telle que chroot ;

— virtualisation de l'accès au système de fichiers au niveau du système d'exploitation avec contournement de l'appel système de bibliothèque pour le propriétaire du fichier, tel que fakeroot ;
— émulation d'API du système d'exploitation, telle que Wine ;
— virtualisation au niveau de l'interpréteur avec sa sélection d'exécutables et ses contournements de bibliothèque d'exécution, telle que virtualenv et venv pour Python.

La virtualisation des conteneurs utilise la Section 4.7.5 et c'est la technologie derrière la Section 7.7.

Voici quelques paquets qui vous aideront à configurer le système virtualisé.

Consultez l'article de Wikipedia Comparaison de machines pour plateforme virtuelle pour une comparaison détaillée entre les différentes solutions de plateformes de virtualisation.

9.11.2 Étapes de la virtualisation

Note
Les noyaux par défaut de Debian prennent en charge KVM depuis `lenny`.

La virtualisation met en œuvre plusieurs étapes :
— Créer un système de fichiers vide (une arborescence de fichiers ou une image disque).
  — L'arborescence de fichiers peut être créée par « `mkdir -p /path/to/chroot` ».
  — L'image disque brute peut être créée à l'aide de dd(1) (consultez Section 9.7.1 et Section 9.7.5).
  — qemu-img(1) peut être utilisé pour créer et convertir des fichiers d'image disque pris en charge par QEMU.
  — Les formats de fichier brut et VMDK peuvent être utilisés en tant que formats courants par les outils de virtualisation.
— Monter l'image disque dans le système de fichiers avec mount(8) (optionnel).
  — Pour l'image disque brute, le montage doit être fait avec un périphérique de rebouclage ou des périphériques device mapper (consultez Section 9.7.3).
  — Les images disques prises en charge par QEMU seront montées en tant que périphériques réseau en mode bloc> (consultez Section 9.11.3).
— Peupler le système de fichiers cible avec les données requises.
  — L'utilisation de programmes tels que `debootstrap` et `cdebootstrap` facilite ce processus (consultez Section 9.11.4).
  — Utiliser les installateurs des systèmes d'exploitation sous l'émulation du système complet.
— Lancer un programme dans l'environnement virtualisé.
  — chroot fournit un environnement virtualisé de base, suffisant pour y compiler des programmes, y faire tourner des applications en mode console et des démons.
  — QEMU fournit une émulation de processeur interplateformes.
  — QEMU avec KVM fournit une émulation système complète avec la virtualisation assistée par le matériel.
  — VirtualBox fournit une émulation du système complet sur i386 amd64 avec ou sans la virtualisation assistée par le matériel.

9.11.3 Monter le fichier image du disque virtuel

Pour le fichier image disque brut, consultez Section 8.1.

Pour d'autres fichiers d'images disques virtuels, vous pouvez utiliser qemu-nbd(8) pour les exporter en utilisant le protocole network block device et en les montant à l'aide du module `nbd` du noyau.

qemu-nbd(8) gère les formats de disques pris en compte par QEMU : raw, qcow2, qcow, vmdk, vdi, bochs, cow (mode utilisateur de Linux copy-on-write), parallels, dmg, cloop, vpc, vvfat (VFAT virtuelle) et host_device.

Le network block device peut gérer des partitions de la même manière que le périphérique de rebouclage (« loop device » (consultez Section 9.7.3). Vous pouvez monter la première partition de « `disk.img` » de la manière suivante :

paquet	popularité	taille	description
coreutils	V:914, I:1000	19751	utilitaires centraux GNU contenant chroot(8)
util-linux	V:922, I:1000	4826	miscellaneous system utilities which contain unshare(1)
systemd-container	V:74, I:77	2964	outils conteneur/nspawn de systemd contenant systemd-nspawn(1)
schroot	V:7.1, I:8.9	2241	outil spécialisé pour l'exécution d'un paquet binaire de Debian dans un chroot
sbuild	V:3.8, I:6.8	160	outil pour construire des paquets binaires de Debian depuis les sources Debian
debootstrap	V:4, I:43	332	amorcer un système Debian de base (écrit en sh)
mmdebstrap	V:8, I:12	613	amorcer un système Debian (écrit en Perl)
cdebootstrap	V:0.1, I:1.2	113	amorcer un système Debian (écrit en C)
cloud-image-utils	V:1, I:14	52	utilitaires de gestion d'image d'infonuagique
cloud-guest-utils	V:5, I:25	67	utilitaires de client d'infonuagique
virt-manager	V:12, I:47	2309	gestionnaire de machine virtuelle : application de bureau pour la gestion des machines virtuelles
libvirt-clients	V:49, I:70	1168	programmes pour la bibliothèque libvirt
docker.io	V:51, I:55	80705	docker : environnement d'exécution de conteneur Linux
podman	V:36, I:40	86285	podman : moteur pour exécuter des conteneurs basés sur OCI dans des « Pod »
podman-docker	V:2.9, I:3.4	271	moteur pour exécuter des conteneurs basés sur OCI dans des « Pod » – enveloppe pour docker
incus	V:1.0, I:3.7	24	Incus : gestionnaire de conteneur de système et de machine virtuelle
games-emulator	I:0.18	21	games-emulator : émulateur de jeux de Debian
bochs	V:0.03, I:0.53	8469	Bochs : émulateur PC IA-32
qemu-system	I:19	69	QEMU : binaires pour l'émulation d'un système complet
qemu-user	V:6.3, I:9.8	472221	QEMU : binaires pour l'émulation en mode utilisateur
qemu-utils	V:16, I:117	12651	QEMU : utilitaires
qemu-system-x86	V:55, I:101	69265	KVM : virtualisation complète sur les plateformes x86 ayant une virtualisation assistée par le matériel
virt-manager	V:12, I:47	2309	Desktop application for managing virtual machines that can be KVM virtual machines, Xen, LXC...
gnome-boxes	V:1.1, I:5.8	7070	Boxes : application simple de GNOME d'accès aux systèmes virtuels
xen-tools	V:0.1, I:1.3	719	outils pour gérer le serveur virtuel XEN de Debian
wine	V:13, I:54	209	Wine : implémentation de l'API Windows (suite standard)
dosbox	V:1, I:11	3139	DOSBox : émulateur x86 avec graphisme Tandy/Herc/CGA/EGA/VGA/SVGA, son et DOS
lxc	V:11, I:13	1680	Conteneurs Linux outils de l'espace utilisateur
python3-venv	V:8, I:167	6	venv pour la création d'environnements Python virtuels (bibliothèque système)
python3-virtualenv	V:7, I:35	435	virtualenv pour créer des environnements Python virtuels isolés
pipx	V:8, I:52	3261	pipx pour installer des applications Python dans des environnements isolés

Table 9.27 – Liste des outils de virtualisation

```
# modprobe nbd max_part=16
# qemu-nbd -v -c /dev/nbd0 disk.img
...
# mkdir /mnt/part1
# mount /dev/nbd0p1 /mnt/part1
```

ASTUCE
Vous ne pouvez exporter que la première partition de « `disk.img` » en utilisant l'option « `-P 1` » de qemu-nbd(8).

9.11.4 Système protégé (chroot)

Si vous souhaitez essayer un nouvel environnement Debian à partir d'une console de terminal, je vous recommande d'utiliser chroot. Cela vous permet d'exécuter des applications console de Debian `unstable` et `testing` sans les risques habituels associés et sans redémarrage. chroot(8) est la méthode la plus basique.



Attention
Les exemples ci-dessous supposent que le système parent et le système chroot partagent la même architecture de CPU `amd64`.

Vous pouvez créer manuellement un environnement chroot(8) en utilisant debootstrap(8), cela demande des efforts non négligeables.

Le paquet sbuild pour construire les paquets Debian à partir des sources utilise l'environnement chroot géré par le paquet schroot. Il est livré avec un script d'aide sbuild-createchroot(8). Voyons comment il fonctionne en l'exécutant comme suit :

```
$ sudo mkdir -p /srv/chroot
$ sudo sbuild-createchroot -v --include=eatmydata,ccache unstable /srv/chroot/unstable- ↩
    amd64-sbuild http://deb.debian.org/debian
 ...
```

Vous pouvez voir comment debootstrap(8) installe les données du système pour l'environnement `unstable` sous « `/srv/chroot/unstable-amd64-sbuild` » pour un système de construction minimal.

Vous pouvez vous connecter à cet environnement en utilisant schroot(1) :

```
$ sudo schroot -v -c chroot:unstable-amd64-sbuild
```

Vous voyez comment une interface système fonctionnant sous l'environnement `unstable` est créée.

Note
Le fichier « `/usr/sbin/policy-rc.d` », qui se termine toujours avec 101, évite que des programmes démons ne soient démarrés automatiquement sur le système Debian. Consulter « `/usr/share/doc/init-system-helpers/README.policy-rc.d.gz` ».

Note
Certains programmes sous chroot peuvent nécessiter l'accès à plus de fichiers du système parent pour fonctionner que ce que `sbuild-createchroot` fournit comme ci-dessus. Par exemple, « `/sys` », « `/etc/passwd` », « `/etc/group` », « `/var/run/utmp` », « `/var/log/wtmp` », etc., peuvent avoir besoin d'être montés par lien (« bind ») ou copiés.

ASTUCE
Le paquet `sbuild` aide à construire un système chroot et construit un paquet à l'intérieur du chroot en utilisant `schroot` comme dorsal. C'est un système idéal pour vérifier les dépendances de construction. Plus de détails sur sbuild sur le wiki Debian et l'exemple de configuration de sbuild dans le Guide pour les responsables Debian.

ASTUCE
La commande systemd-nspawn(1) aide à exécuter une commande ou un système d'exploitation dans un conteneur léger de manière similaire à `chroot`. Elle est plus puissante, car elle utilise des espaces de noms pour virtualiser pleinement arbre de processus, IPC, nom d'hôte, nom de domaine et, en option, réseau et bases de données utilisateur. Voir systemd-nspawn.

9.11.5 Systèmes de bureaux multiples

Si vous souhaitez essayer un nouvel environnement de bureau graphique de n'importe quel système d'exploitation, je vous recommande d'utiliser QEMU ou KVM sur un système Debian `stable` pour exécuter plusieurs systèmes de bureau en toute sécurité en utilisant la virtualisation. Cela vous permet d'exécuter n'importe quelle application de bureau, y compris celles de Debian `unstable` et `testing` sans les risques habituels qui y sont associés et sans redémarrage.

Comme un QEMU pur est très lent, il est recommandé de l'accélérer avec KVM lorsque le système hôte le prend en charge.

Virtual Machine Manager, également connu comme `virt-manager`, est un outil graphique pratique pour gérer les machines virtuelles KVM à l'aide de libvirt.

L'image disque virtuelle « `virtdisk.qcow2` » qui contient un système Debian pour QEMU peut être créée en utilisant un CD minimal de debian-installer de la manière suivante :

```
$ wget https://cdimage.debian.org/debian-cd/5.0.3/amd64/iso-cd/debian-503-amd64-netinst.iso
$ qemu-img create -f qcow2 virtdisk.qcow2 5G
$ qemu -hda virtdisk.qcow2 -cdrom debian-503-amd64-netinst.iso -boot d -m 256
...
```

ASTUCE
Faire tourner d'autres distributions de GNU/Linux comme Ubuntu et Fedora sous une virtualisation est une bonne manière d'en étudier les astuces de configuration. D'autres systèmes d'exploitation propriétaires peuvent aussi tourner de manière agréable sous la virtualisation GNU/Linux.

Vous trouverez d'autres conseils sur le wiki Debian : virtualisation du système.

Chapitre 10

Gestion des données

Des outils et astuces pour gérer les données binaires ou textuelles sur le système sont décrits.

10.1 Partager, copier et archiver



AVERTISSEMENT
Il ne faut pas accéder de manière non coordonnée en écriture à des périphériques et des fichiers à haut trafic depuis différents processus pour éviter une compétition d'accès (« race condition ». Pour l'éviter, on peut utiliser les mécanismes de verrouillage de fichier (« File locking » en utilisant flock(1).

La sécurité des données et leur partage contrôlé présentent plusieurs aspects.

— La création d'une archive des données
— L'accès à un stockage distant
— La duplication
— Le suivi de l'historique des modifications
— La facilité de partage des données
— La prévention de l'accès non autorisé aux données
— La détection des modifications de fichier non autorisées

Cela peut être réalisé avec certaines combinaisons d'outils.

— Outils d'archivage et de compression
— Outils de copie et de synchronisation
— Systèmes de fichiers par le réseau
— Supports d'enregistrement amovibles
— L'interpréteur de commandes sécurisé
— Le système d'authentification
— Outils de système de contrôle de version
— Outils de hachage et de chiffrement

10.1.1 Outils d'archivage et de compression

Voici un résumé des outils d'archivage et de compression disponible sur le système Debian :

paquet	popularité	taille	extension	commande	commentaire
tar	V:911, I:1000	3101	.tar	tar(1)	archiveur standard (standard de fait)
cpio	V:388, I:999	1205	.cpio	cpio(1)	archiveur de style UNIX System V, utiliser avec find(1)
binutils	V:201, I:627	1132	.ar	ar(1)	archiveur pour la création de bibliothèques statiques
fastjar	V:0.9, I:9.0	182	.jar	fastjar(1)	archiveur pour Java (semblable à zip)
pax	V:4.9, I:8.5	167	.pax	pax(1)	nouvel archiveur standard POSIX, compromis entre tar et cpio
gzip	V:904, I:1000	262	.gz	gzip(1), zcat(1), ...	utilitaire de compression GNU LZ77 (standard de fait)
bzip2	V:163, I:973	113	.bz2	bzip2(1), bzcat(1), ...	utilitaire de transformée par tri de blocs de Burrows-Wheeler permettant un taux de compression plus élevé que gzip(1) (plus lent que gzip avec une syntaxe similaire)
lzma	V:0.7, I:9.4	349	.lzma	lzma(1)	utilitaire de compression LZMA avec un taux de compression supérieur à gzip(1) (obsolète)
xz-utils	V:323, I:981	1551	.xz	xz(1), xzdec(1), ...	utilitaire de compression XZ avec un plus haut taux de compression que bzip2(1) (plus lent que gzip mais plus rapide que bzip2 ; le remplaçant de LZMA utilitaire de compression)
zstd	V:346, I:850	2308	.zstd	zstd(1), zstdcat(1), ...	utilitaire de compression sans perte rapide Zstandard
7zip	V:86, I:339	7542	.7z	7zr(1), p7zip(1)	archiveur de fichiers 7-Zip avec un haut taux de compression (compression LZMA et autres)
lzop	V:13, I:128	164	.lzo	lzop(1)	utilitaire de compression LZO avec de plus hautes vitesses de compression et de décompression que gzip(1) (plus faible taux de compression que gzip avec une syntaxe similaire)
zip	V:52, I:350	635	.zip	zip(1)	InfoZIP : outil d'archive et de compression DOS
unzip	V:126, I:738	399	.zip	unzip(1)	InfoZIP : outil de désarchivage et de décompression DOS

Table 10.1 – Liste des outils d'archivage et de compression



AVERTISSEMENT
Ne positionnez par la variable « $TAPE » à moins que vous ne sachiez à quoi vous attendre. Elle modifie le comportement de tar(1).

— L'archive tar(1) utilise l'extension de fichier « `.tgz` » ou « `.tar.gz` ».
— L'archive tar(1) avec compression xz utilise l'extension de fichier « `.txz` » ou « `.tar.xz` ».
— La méthode de compression habituelle des outils FOSS tels que tar(1) a été modifiée de la manière suivante : `gzip` → `bzip2` → `xz`
— cp(1), scp(1) et tar(1) peuvent avoir certaines limitations pour des fichiers spéciaux. cpio(1) est plus souple.
— cpio(1) est conçu pour être utilisé avec find(1) et d'autres commandes et est adapté à la création de scripts de sauvegarde car la partie correspondant à la sélection de fichier du script peut être testée indépendamment.
— La structure interne des fichiers de données de Libreoffice est composée de fichiers « `.jar` » qui peut être ouvert également par `unzip`.
— L'outil d'archivage de facto de la plateforme est `zip`. Utilisez-le sous la forme « `zip -rX` » pour obtenir une compatibilité maximale. Utilisez également l'option « `-s` », si la taille maximale du fichier est importante.

10.1.2 Outils de copie et de synchronisation

Voici une liste d'outils simples de copie et de sauvegarde sur le système Debian :

paquet	popularité	taille	outil	fonction
`coreutils`	V:914, I:1000	19751	GNU cp	copier localement des fichiers et répertoires (« -a » pour récursif)
`openssh-client`	V:709, I:997	3611	scp	copier à distance des fichiers et des répertoires (client, « `-r` » pour récursif)
`openssh-server`	V:781, I:824	3615	sshd	copier à distance des fichiers et des répertoires (serveur distant)
`rsync`	V:191, I:538	985		synchronisation et sauvegarde distantes unidirectionnelles
`unison`	V:3, I:12	14		synchronisation et sauvegarde distantes bidirectionnelles

Table 10.2 – Liste des outils de copie et de synchronisation

Copier des fichiers avec rsync(1) offres des fonctionnalités plus riches que les autres méthodes.

— algorithme de transfert delta qui n'envoie que la différence entre les fichiers source et les fichiers existants sur la destination
— algorithme de vérification rapide (par défaut) recherchant les fichiers dont la taille ou l'heure de dernière modification a été modifiée
— Les options « `--exclude` » et « `--exclude-from` » sont semblables à celles de tar(1)
— La syntaxe « un slash en fin de répertoire source » qui évite la création d'un niveau de répertoire supplémentaire à la destination.

ASTUCE
Les outils de système de contrôle de version (VCS) de Tableau 10.14 peuvent fonctionner comme outils de synchronisation et de copie multi-voies.

10.1.3 Idiomes pour les archives

Voici quelques manières d'archiver et de désarchiver le contenu entier du répertoire « `./source` » en utilisant différents outils.

GNU tar(1) :

```
$ tar -cvJf archive.tar.xz ./source
$ tar -xvJf archive.tar.xz
```

ou encore, comme suit :

```
$ find ./source -xdev -print0 | tar -cvJf archive.tar.xz --null -T -
```

cpio(1) :

```
$ find ./source -xdev -print0 | cpio -ov --null > archive.cpio; xz archive.cpio
$ zcat archive.cpio.xz | cpio -i
```

10.1.4 Idiomes pour la copie

Voici quelques manières d'archiver et de désarchiver le contenu entier du répertoire « `./source` » en utilisant différents outils.

— Copie locale du répertoire « `./source` » → répertoire « `/dest` »
— Faire le copie distante du répertoire « `./source` » de la machine locale → répertoire « `/dest` » situé sur la machine « `user@host.dom` »

rsync(8) :

```
# cd ./source; rsync -aHAXSv . /dest
# cd ./source; rsync -aHAXSv . user@host.dom:/dest
```

Vous pouvez, en remplacement, utiliser la syntaxe « un slash en fin du répertoire source ».

```
# rsync -aHAXSv ./source/ /dest
# rsync -aHAXSv ./source/ user@host.dom:/dest
```

ou encore, comme suit :

```
# cd ./source; find . -print0 | rsync -aHAXSv0 --files-from=- . /dest
# cd ./source; find . -print0 | rsync -aHAXSv0 --files-from=- . user@host.dom:/dest
```

GNU cp(1) et openSSH scp(1) :

```
# cd ./source; cp -a . /dest
# cd ./source; scp -pr . user@host.dom:/dest
```

GNU tar(1) :

```
# (cd ./source && tar cf - . ) | (cd /dest && tar xvfp - )
# (cd ./source && tar cf - . ) | ssh user@host.dom '(cd /dest && tar xvfp - )'
```

cpio(1) :

```
# cd ./source; find . -print0 | cpio -pvdm --null --sparse /dest
```

Vous pouvez remplacer « `.` » par « `truc` » dans tous les exemples comportant « `.` » pour copier les fichiers du répertoire « `./source/truc` » vers le répertoire « `/dest/truc` ».

Vous pouvez remplacer « `.` » par le chemin absolu « `/chemin/vers/source/truc` » dans tous les exemples comportant « `.` » pour éviter « `cd ./source;` ». Cela permet de copier les fichiers vers différents emplacements selon les outils utilisés de la manière suivante :

— « `/dest/truc` » : rsync(8), GNU cp(1) et scp(1)
— « `/dest/chemin/vers/source/truc` » : GNU tar(1) et cpio(1)

ASTUCE
rsync(8) et GNU cp(1) possèdent l'option « `-u` » pour sauter les fichiers qui sont plus récents sur la destination.

10.1.5 Idiomes pour la sélection de fichiers

find(1) est utilisé pour la sélection de fichiers pour les commandes d'archive et de copie (consultez Section 10.1.3 et Section 10.1.4) ou pour xargs(1) (consultez Section 9.4.9). Cela peut être amélioré en utilisant ces paramètres de commande.

La syntaxe de base de find(1) peut être résumée comme suit :

— Ses paramètres conditionnels sont évalués de gauche à droite.
— L'évaluation s'arrête lors que son résultat est déterminé.
— Le « **OU** logique » (indiqué par « `-o` » entre les éléments conditionnels) a une plus faible priorité que le « **ET** logique » (indiqué par « `-a` » ou rien entre éléments conditionnels).
— Le « **NON** logique » (indiqué par « `!` » avant un élément conditionnel) a une priorité plus élevée que le « **ET** logique ».
— « `-prune` » retourne toujours un **VRAI** logique et, si c'est un répertoire, la recherche de fichier est arrêtée au-delà de ce point.
— « `-name` » correspond à la base du nom de fichier avec les motifs génériques de l'interpréteur de commandes (voir Section 1.5.6) mais il correspond aussi à son « `.` » de début avec des métacaractères comme « * » et « ? » (nouvelle fonctionnalité POSIX).
— « `-regex` » correspond au chemin complet par défaut dans le style **BRE** Emacs (consultez Section 1.6.2).
— « `-size` » correspond au fichier en se basant sur la taille du fichier (valeur précédée de « + » pour plus grand et précédée de « - » pour plus petit)
— « `-newer` » correspond au fichier plus récent que celui indiqué comme paramètre.
— « `-print0` » retourne toujours la valeur logique **VRAI** et affiche sur la sortie standard le nom de fichier en entier (terminé par le caractère null).

find(1) est souvent utilisé dans un style idiomatique comme ce qui suit :

```
# find /path/to \
    -xdev -regextype posix-extended \
    -type f -regex ".*\.cpio|.*~" -prune -o \
    -type d -regex ".*/\.git" -prune -o \
    -type f -size +99M -prune -o \
    -type f -newer /path/to/timestamp -print0
```

Cela signifie que les actions suivantes doivent être effectuées :

1. rechercher tous les fichiers en partant de « `/chemin/vers` » ;
2. limiter globalement sa recherche à l'intérieur du système de fichiers et utiliser **ERE** (consultez Section 1.6.2) ;
3. exclure les fichiers correspondant à l'expression rationnelle de « `.*\.cpio` » ou « `.*~` » de la recherche en arrêtant le traitement ;
4. exclure les répertoires qui correspondent à l'expression rationnelle de « `.*/\.git` » de la recherche en arrêtant le traitement ;
5. exclure les fichiers plus gros que 99 Megaoctets (unités de 1048576 octets) de la recherche en arrêtant le traitement ;
6. Afficher les noms de fichiers qui satisfont aux conditions de recherche ci-dessus et qui sont plus récents que « `/chemin/vers/horodatage` ».

Remarquez l'utilisation idiomatique de « `-prune -o` » pour exclure les fichiers dans l'exemple ci-dessus.

Note
Pour les systèmes UNIX-like autre que Debian, certaines options peuvent ne pas être prises en compte par find(1). Dans un tel cas, essayez d'adapter la méthode de correspondance et remplacez « `-print0` » par « `-print` ». Vous devrez aussi ajuster les commandes associées.

10.1.6 Support d'archive

Lors du choix d'un support d'enregistrement de données informatiques destiné à l'archivage de données importantes, il faut faire attention à leurs limitations. Pour des petites sauvegardes de données personnelles, j'utilise des CD-R et des DVD-R provenant d'une grande marque et je les range dans un endroit frais, à l'ombre, sec et propre. (Les supports d'archive sur bande semblent être populaires pour les utilisations professionnelles.)

Note
Un coffre-fort anti-feu est destiné aux documents sur papier. La plupart des supports de stockage de données informatiques ont une tolérance en température inférieure à celle du papier. J'utilise en général plusieurs copies chiffrées stockées dans différents endroits sûrs.

Durées de vie optimistes des moyens d'archivage trouvées sur le net (la plupart à partir d'informations des constructeurs).

— 100 ans et plus : papier non acide et encre
— 100 ans : stockage optique (CD/DVD, CD/DVD-R)
— 30 ans : supports magnétiques (bande, disquette)
— 20 ans : disque optique à changement de phase (CD-RW)

Cela ne prend pas en compte les défaillances mécaniques dues aux manipulations, etc.

Nombre de cycles d'écriture optimistes des moyens d'archivage trouvées sur le net (la plupart à partir d'informations des constructeurs).

— plus de 250 000 : disque dur
— plus de 10 000 cycles : mémoires Flash
— 1000 cycles : CD/DVD-RW
— 1 cycle : CD/DVD-R, papier



Attention
Ces chiffres de durée de vie et de nombre de cycles ne devront pas être utilisés pour des décisions concernant l'enregistrement de données critiques. Veuillez consulter les informations spécifiques au produit fournies par le constructeur.

ASTUCE
Comme les CD/DVD-R et le papier n'ont qu'un cycle d'écriture de 1, ils évitent de manière inhérente le risque de perte de données par écrasement. C'est un avantage !

ASTUCE
Si vous devez faire des sauvegardes fréquentes et rapides d'un gros volume de données, un disque dur sur une liaison réseau à haute vitesse peut être la seule option réaliste.

ASTUCE
Si vous utilisez des médias réinscriptibles pour vos sauvegardes, l'utilisation d'un système de fichiers tel que btrfs ou zfs qui prend en charge les instantanés en lecture seule peut être une bonne idée.

10.1.7 Périphériques d'enregistrement amovibles

Les périphériques d'enregistrement amovibles possibles sont les suivants.

— Clé USB
— Disque dur
— Graveur de disque optique
— Appareil photographique numérique
— Lecteur de musique numérique

Ils peuvent être connectés à l'aide de n'importe quel moyen suivant.

— USB
— IEEE 1394 / FireWire

Les environnements de bureau modernes comme GNOME et KDE peuvent monter ces périphériques amovibles automatiquement sans entrée correspondante dans « `/etc/fstab` ».

— Le paquet `udisks2` fournit un démon et les utilitaires associés pour monter et démonter ces périphériques.
— D-bus crée les événements pour initialiser les processus automatiques.
— PolicyKit fournit les droits nécessaires.

ASTUCE
Les périphériques montés automatiquement pourraient avoir l'option de montage « `uhelper=` » qui est utilisée par umount(8).

ASTUCE
Le montage automatique sous les environnements de bureau modernes ne se produit que lorsque ces périphériques amovibles ne se trouvent pas dans « `/etc/fstab` ».

Le nom choisi du point de montage dans les environnements de bureau modernes est « `/media/`*`nom_utilisateur/étiq`* et il peut être personnalisé avec les outils suivants :

— mlabel(1) pour le système de fichiers FAT ;
— genisoimage(1) avec l'option « `-V` » pour le système de fichiers ISO9660 ;
— tune2fs(8) avec l'option « `-L` » pour le système de fichiers ext2, ext3 ou ext4 ;

ASTUCE
Le choix du codage doit être fourni comme option de montage (consultez Section 8.1.3).

ASTUCE
L'utilisation d'une interface graphique pour démonter un système de fichiers peut supprimer son point de montage créé dynamiquement tel que « `/dev/sdc` ». Si vous souhaitez conserver son point de montage, démontez-le à l'aide de la commande umount(8) lancée depuis un interpréteur de commandes.

10.1.8 Choix de système de fichiers pour les données partagées

Lors du partage de données avec d'autres systèmes à l'aide de périphériques de stockage amovibles, vous devez les formatez avec un système de fichiers pris en charge par les deux systèmes. Voici une liste de choix de systèmes de fichiers :

nom du système de fichiers	scénario d'utilisation typique
FAT12	partage de données à l'aide de disquettes entre plateformes (<32Mio)
FAT16	partage de données à l'aide de périphériques semblables à des disques durs de faible capacité (<2Gio) entre plateformes
FAT32	partage de données à l'aide de périphériques semblables à des disques durs de grande capacité (<8Tio, pris en charge par plus récent que MS Windows95 OSR2) entre plateformes
exFAT	partage de données à l'aide de périphériques semblables à des disques durs de grande capacité (<8Tio, pris en charge par WindowsXP, Mac OS X Snow Leopard 10.6.5 et Linux depuis la version 5.4) entre plateformes
NTFS	partage de données sur périphériques semblables à des disques durs de grande capacité (pris en charge de manière native par MS Windows NT et versions plus récentes, et pris en charge par NTFS-3G à l'aide de FUSE sous Linux) entre plateformes
ISO9660	partage de données à l'aide d'un CD-R ou DVD+/-R entre plateformes
UDF	écriture incrémentale de CD-R et de DVD+/-R (nouveau)
MINIX	enregistrement efficace en matière d'espace disque de fichiers de données unix sur disquette
ext2	partage de données sur disque dur avec les anciens systèmes Linux
ext3	partage de données sur disque dur avec les anciens systèmes Linux
ext4	partage de données sur disque dur avec les systèmes Linux actuels
btrfs	partage des données sur le disque dur comme le dispositif avec les systèmes Linux actuels avec des instantanés en lecture seule

Table 10.3 – Liste de choix de systèmes de fichiers pour des périphériques amovibles avec des scénarios typiques d'utilisation

Note
Statements on hard disk (HDD) are applicable to other storage devices such as SSD / USB flash drive / Memory card / ... . Replace device names in examples such as `/dev/sda` with applicable device names `/dev/nvme0`, `/dev/mmcblk0`, ...

ASTUCE
Consultez Section 9.9.1 pour le partage de données entre plateformes en utilisant le chiffrement au niveau du périphérique.

Le système de fichiers FAT est pris en charge par la plupart des systèmes d'exploitation modernes et est assez utile pour l'échange des données par l'intermédiaire de supports du type disque dur amovible.

Pour le formatage de périphériques de type disque dur amovible pour l'échange de données entre plateformes avec un système de fichiers FAT, ce qui suit peut être un choix sûr :

- les partitionner avec fdisk(8), cfdisk(8) ou parted(8) (consultez Section 9.6.2) en une seule partition primaire et la marquer comme suit :
  - type « 6 » pour FAT16 pour les supports faisant moins de 2Go.
  - type « c » pour FAT32 (LBA) pour les supports plus gros.
- formater la partition primaire avec mkfs.vfat(8) comme suit :
  - simplement son nom de périphérique, par exemple « `/dev/sda1` », pour la FAT16
  - l'option explicite et le nom du périphérique, par exemple « `-F 32 /dev/sda1` », pour la FAT32.

Lors de l'utilisation des systèmes de fichiers FAT ou ISO9660 pour le partage de données ce qui suit sera une précaution sûre :

— Archiver d'abord les fichiers dans un fichier d'archive en utilisant tar(1) ou cpio(1) afin de conserver les noms de fichiers longs, les permissions de fichiers d'origine d'UNIX et les informations de propriétaire.
— Découper le fichier d'archive en éléments de moins de 2 Gio à l'aide de la commande split(1) afin de le protéger contre les limitations de taille de fichier.
— Chiffrer le fichier d'archive afin de sécuriser son contenu contre un accès non autorisé.

Note
La taille maximale d'un fichier FAT, par conception, est de `(2^32 - 1) octets = (4GiB - 1 octet)`. Pour certaines applications sur le système 32 bits plus ancien, la taille maximale était même plus faible `(2^31 - 1) octets = (2Gio - 1 octet)`. Debian ne souffre pas de ce dernier problème.

Note
Microsoft lui-même ne recommande pas l'utilisation de FAT pour des disques ou des partitions de plus de 200 Mo. Microsoft met en avant ces limitations comme une utilisation inefficace de l'espace disque dans ses « Informations générales sur les systèmes de fichiers FAT, HPFS et NTFS ». Bien sûr, on peut normalement utiliser le système de fichiers ext4 pour Linux.

ASTUCE
Pour davantage d'informations sur les systèmes de fichiers et les accès aux systèmes de fichiers, veuillez consulter « Filesystems HOWTO ».

10.1.9 Partage de données au travers du réseau

Lors du partage de données avec d'autres systèmes au travers du réseau, vous devrez utiliser un service commun. Voici quelques éléments :

service réseau	description d'un scénario typique d'utilisation
SMB/CIFS système de fichiers monté avec Samba	partage de fichiers par l'intermédiaire de « Microsoft Windows Network », consultez smb.conf(5) et le HOWTO et guide de référence officiel de Samba 3.x.x
système de fichiers monté au travers du réseau NFS avec le noyau Linux	partager des fichiers par « UNIX/Linux Network », consultez exports(5) et Linux NFS-HOWTO
service HTTP	partager des fichiers entre client et serveur web
service HTTPS	partager des fichiers entre le client et le serveur web avec un chiffrement Secure Sockets Layer (SSL) ou Transport Layer Security (TLS)
service FTP	partager des fichiers entre serveur et client FTP

Table 10.4 – Liste des services réseau à choisir avec le scénario typique d'utilisation

Bien que ces systèmes de fichiers montés au travers du réseau et les méthodes de transfert au travers du réseau soient assez pratiques pour partager des données, elles peuvent être non sûres. Leur connexion réseau doit être sécurisée par ce qui suit :

— chiffrez-la avec SSL/TLS
— tunnelez-la par SSH
— tunnelez-la par VPN
— limitez-la derrière un pare-feu sûr

consultez aussi Section 6.5 et Section 6.6.

10.2 Sauvegarde et restauration

Nous savons tous que les ordinateurs sont parfois victimes de pannes ou que des erreurs humaines provoquent des dommages au système et aux données. Les opérations de sauvegarde et de restauration sont les parties essentielles d'une administration système réussie. Vous serez victime, un jour ou l'autre, de tous les modes de défaillance possibles.

ASTUCE
Mettez en place un système de sauvegardes simple et faites une sauvegarde fréquente de votre système. Avoir des données de sauvegarde est plus important que la qualité technique de votre méthode de sauvegarde.

10.2.1 Politique de sauvegarde et de restauration

Il y a 3 facteurs-clé qui permettent de définir une méthode pratique de sauvegarde et de restauration.

1. Ce qu'il faut sauvegarder et restaurer :
   - les fichiers que vous avez directement créés : données de « `~/` » ;
   - les fichiers de données créés par les applications que vous utilisez : données de « `/var/` » (sauf « `/var/cache/` » « `/var/run/` » et « `/var/tmp/` ») ;
   - les fichiers de configuration du système : fichiers de « `/etc/` » ;
   - programmes locaux : données dans « `/usr/local/` » ou « `/opt/` » ;
   - informations concernant l'installation du système : un mémo en texte concernant les étapes-clés ((partition, …) ;
   - un jeu de données testé : confirmé par des opérations de restauration expérimentales réalisées à l'avance :
     - tâches cron en tant que processus utilisateur : fichiers dans le répertoire « `/var/spool/cron/crontabs` » et redémarrage de cron(8). Consulter la Section 9.4.14 pour cron(8) et crontab(1) ;
     - tâches de Systemd liées au temps en tant que processus utilisateur : fichiers dans le répertoire « `~/.config/sy` Consulter systemd.timer(5) et systemd.service(5) ;
     - tâches de démarrage automatique en tant que processus utilisateur : fichiers dans le répertoire « `~/.config/a` Consulter Spécifications pour le démarrage automatique des applications de bureau.
2. Comment sauvegarder et restaurer :
   - entreposer les données de manière sûre : protection des données contre la réécriture et les défaillances du système ;
   - sauvegardes fréquentes ! sauvegardes planifiées ;
   - sauvegardes redondantes : duplication (miroir) des données ;
   - processus indéréglable : sauvegarde facile en une seule commande.
3. Risques et coûts :
   - risque de perte de données :
     - les données devraient être au moins sur des partitions différentes de disque, de préférence sur différents disques et machines pour résister à la corruption du système de fichiers. Il est préférable de stocker les données importantes sur un système de fichiers en lecture seule 1.
   - risque de violation des données :
     - Les données d'identité sensibles telles que « `/etc/ssh/ssh_host_*_key` », « `~/.gnupg/*` », « `~/.ssh/*` « `~/.local/share/keyrings/*` », « `/etc/passwd` », « `/etc/shadow` », « `popularity-contest.con` « `/etc/ppp/pap-secrets`», et « `/etc/exim4/passwd.client` » doivent être sauvegardées en étant chiffrées 2. (Consulter la Section 9.9.)

1. Un média à écriture unique tel qu'un CD/DVD-R peut éviter les accidents d'écrasement. (Consulter la Section 9.8 pour savoir comment écrire sur le média de stockage à partir de la ligne de commande d'interpréteur. L'environnement graphique de bureau GNOME vous donne un accès facile avec le menu : « Places→CD / DVD Creator ».)
2. Certaines de ces données ne peuvent pas être régénérées en saisissant la même chaîne d'entrée dans le système.

  - ne jamais coder en dur le mot de passe de connexion au système ou la phrase de passe de décryptage dans un script, même sur un système de confiance. (Consulter la Section 10.3.6.)
- mode de défaillance avec leur probabilité :
  - le matériel (en particulier le disque dur) va casser ;
  - le système de fichiers peut être corrompu et les données qu'il contient peuvent être perdues ;
  - le système de stockage distant n'est pas fiable à cause des failles de sécurité ;
  - une protection par mot de passe faible peut être facilement compromise ;
  - le système de permissions des fichiers peut être compromis.
- ressources nécessaires pour effectuer les sauvegardes : humaines, matérielles, logicielles, …
  - sauvegarde automatique planifiée avec une tâche cron ou une tâche de minuterie de systemd.

ASTUCE
Vous pouvez récupérer les données de configuration de debconf avec « `debconf-set-selections debconf-selections` » et les données de sélection de dpkg avec « `dpkg --set-selection <dpkg-selections.list` ».

Note
Ne sauvegardez pas le contenu des pseudo systèmes de fichiers se trouvant dans `/proc`, `/sys`, `/tmp`, et `/run` (voir Section 1.2.12 et Section 1.2.13). À moins que vous ne sachiez exactement ce que vous faites, ce ne sont que d'énormes quantités de données inutiles.

Note
Il faudra peut-être arrêter certains démons d'applications comme le MTA (consultez Section 6.2.4) lors de la sauvegarde des données.

10.2.2 Suites d'utilitaires de sauvegarde

Voici une liste d'utilitaires de sauvegarde notables disponibles sur le système Debian :

Les outils de sauvegarde ont chacun des objectifs particuliers.

- Mondo Rescue est un système de sauvegarde qui facilite la restauration rapide d'un système complet depuis de CD/DVD etc. sans passer par le processus normal d'installation d'un système.
- Bacula, Amanda et BackupPC sont des suites de sauvegardes ayant des fonctionnalités avancées qui sont orientées vers les sauvegardes fréquentes au travers du réseau.
- Duplicity et Borg sont des utilitaires très simples de sauvegarde pour les stations de travail typiques.

10.2.3 Astuces de sauvegarde

Pour une station de travail personnelle, les utilitaires de suite de sauvegarde remplis de fonctionnalités et conçus pour des environnements de serveur peuvent ne pas être très adaptés. En outre, les utilitaires de sauvegarde pour les stations de travail peuvent avoir quelques défauts.

Voici quelques astuces pour faciliter la sauvegarde avec un minimum d'efforts pour l'utilisateur. Ces techniques peuvent être utilisées avec n'importe quel utilitaire de sauvegarde.

Dans un but de démonstration, supposons que l'utilisateur principal et le groupe soient `penguin` et créons un script d'instantané et de sauvegarde « `/usr/local/bin/bkss.sh` » :

paquet	popularité	taille	description
bacula-common	V:5.7, I:6.4	2501	Bacula : sauvegarde, restauration et vérification par le réseau - fichiers communs
bacula-client	V:0.2, I:1.7	199	Bacula : sauvegarde, restauration et vérification par le réseau - métapaquet du client
bacula-console	V:0.6, I:2.0	112	Bacula : sauvegarde, restauration et vérification par le réseau - console en mode texte
bacula-server	I:0.56	199	Bacula : sauvegarde, restauration et vérification par le réseau - métapaquet du serveur
amanda-common	V:0.5, I:2.0	9952	Amanda : Advanced Maryland Automatic Network Disk Archiver (Libs). (Archiveur de disque par le réseau de Maryland avancé et automatique
amanda-client	V:0.5, I:2.0	1110	Amanda : Advanced Maryland Automatic Network Disk Archiver (Client)
amanda-server	V:0.11, I:0.22	1096	Amanda : Advanced Maryland Automatic Network Disk Archiver (Serveur)
backuppc	V:1.3, I:1.5	3088	BackupPC est un système de hautes performances pour effectuer la sauvegarde de PC au niveau de l'entreprise (basé sur disques)
duplicity	V:5, I:44	2679	sauvegarde incrémentale (distante)
deja-dup	V:28, I:41	5274	frontal graphique pour duplicity
borgbackup	V:13, I:28	3717	sauvegarde sans doublon (distante)
borgmatic	V:3.4, I:4.4	946	assistant pour borgbackup
rdiff-backup	V:2.1, I:6.0	1204	sauvegarde incrémentale (distante)
restic	V:10, I:17	44216	sauvegarde incrémentale (distante)
backupninja	V:2.0, I:2.4	360	système de sauvegarde **meta-backup** léger et extensible
slbackup	V:0.10, I:0.12	147	sauvegarde incrémentale (distante)
backup-manager	V:0.39, I:0.68	573	outil de sauvegarde en ligne de commandes
backup2l	V:0.34, I:0.46	110	outil de sauvegarde et restauration de faible maintenance pour des supports pouvant être montés (basé sur disque)

Table 10.5 – Liste de suites d'utilitaires de sauvegarde

```
#!/bin/sh -e
SRC="$1" # source data path
DSTFS="$2" # backup destination filesystem path
DSTSV="$3" # backup destination subvolume name
DSTSS="${DSTFS}/${DSTSV}-snapshot" # snapshot destination path
if [ "$(stat -f -c %T "$DSTFS")" != "btrfs" ]; then
  echo "E: $DESTFS needs to be formatted to btrfs" >&2 ; exit 1
fi
MSGID=$(notify-send -p "bkup.sh $DSTSV" "in progress ...")
if [ ! -d "$DSTFS/$DSTSV" ]; then
  btrfs subvolume create "$DSTFS/$DSTSV"
  mkdir -p "$DSTSS"
fi
rsync -aHxS --delete --mkpath "${SRC}/" "${DSTFS}/${DSTSV}"
btrfs subvolume snapshot -r "${DSTFS}/${DSTSV}" ${DSTSS}/$(date -u --iso=min)
notify-send -r "$MSGID" "bkup.sh $DSTSV" "finished!"
```

Ici, seul l'outil basique rsync(8) est utiliser pour faciliter la sauvegarde du système et l'espace de stockage est utilisé efficacement par Btrfs.

ASTUCE
Pour information, l'auteur utilise son propre script similaire d'interpréteur « bss : utilitaire d'instantané de sous-volume » pour sa station de travail.

10.2.3.1 Sauvegarde depuis une interface graphique

Voici un exemple pour configurer une sauvegarde par clic sur une interface graphique.

— Préparez un périphérique de stockage USB pour la sauvegarde :
  — formatez un périphérique de stockage USB avec une seule partition en btrfs avec comme nom « BKUP » et qui peut être chiffrée (consulter la Section 9.9.1) ;
  — enfichez le périphérique dans le système. Le système de bureau le montera automatiquement comme « `/media/pen`
  — exécutez « `sudo chown penguin:penguin /media/penguin/BKUP` » pour le rendre éditable par l'utilisateur.
— Créez « `~/.local/share/applications/BKUP.desktop` » en suivant les techniques décrites dans la Section 9.4.10 contenant :

```
[Desktop Entry]
Name=bkss
Comment=Backup and snapshot of ~/Documents
Exec=/usr/local/bin/bkss.sh /home/penguin/Documents /media/penguin/BKUP Documents
Type=Application
```

Pour chaque clic d'interface graphique, les données sont sauvegardées de « `~/Documents` » vers le périphérique USB et un instantané en lecture seule est créé.

10.2.3.2 Sauvegarde déclenchée par des évènements de montage

Voici un exemple de configuration pour une sauvegarde automatique déclenchée par un évènement de montage.

— Préparez un périphérique de stockage USB à utiliser pour une sauvegarde comme dans la Section 10.2.3.1.
— Créez un fichier d'unité de service systemd « `~/.config/systemd/user/back-BKUP.service` » contenant :

```
[Unit]
Description=USB Disk backup
Requires=media-%u-BKUP.mount
After=media-%u-BKUP.mount

[Service]
ExecStart=/usr/local/bin/bkss.sh %h/Documents /media/%u/BKUP Documents
StandardOutput=append:%h/.cache/systemd-snap.log
StandardError=append:%h/.cache/systemd-snap.log

[Install]
WantedBy=media-%u-BKUP.mount
```

— activer cette configuration d'unité de systemd avec :

```
$ systemctl --user enable bkup-BKUP.service
```

Pour chaque évènement de montage, les données sont sauvegardées de « `~/Documents` » vers le périphérique de stockage USB et un instantané en lecture seule est créé.

Ici, les noms d'unité de montage de systemd que celui-ci a actuellement en mémoire peuvent être demandés au gestionnaire de services de l'utilisateur appelant avec « `systemctl --user list-units --type=mount` ».

10.2.3.3 Sauvegarde déclenchée par des évènements d'horloge

Voici un exemple de configuration de sauvegarde automatique déclenchée par un évènement d'horloge.

— Préparez un périphérique de stockage USB à utiliser pour une sauvegarde comme dans la Section 10.2.3.1.

— Créez un fichier d'unité d'horloge de systemd « `~/.config/systemd/user/snap-Documents.timer` » contenant :

```
[Unit]
Description=Run btrfs subvolume snapshot on timer
Documentation=man:btrfs(1)

[Timer]
OnStartupSec=30
OnUnitInactiveSec=900

[Install]
WantedBy=timers.target
```

— Créez un fichier d'unité de service de systemd « `~/.config/systemd/user/snap-Documents.service` » contenant :

```
[Unit]
Description=Run btrfs subvolume snapshot
Documentation=man:btrfs(1)

[Service]
Type=oneshot
Nice=15
ExecStart=/usr/local/bin/bkss.sh %h/Documents /media/%u/BKUP Documents
IOSchedulingClass=idle
CPUSchedulingPolicy=idle
StandardOutput=append:%h/.cache/systemd-snap.log
StandardError=append:%h/.cache/systemd-snap.log
```

— activer cette configuration d'unité de systemd avec :

```
$ systemctl --user enable snap-Documents.timer
```

Pour chaque évènement d'horloge, les données sont sauvegardées de « `~/Documents` » vers le périphérique de stockage USB et un instantané en lecture seule est créé.

Ici, les noms d'unités d'horloge de systemd que celui-ci a en mémoire peuvent être demandés au gestionnaire de services de l'utilisateur appelant avec « `systemctl --user list-units --type=timer` ».

Pour les systèmes de bureau modernes, cette approche de systemd peut offrir un contrôle plus fin que celles traditionnelles d'Unix utilisant at(1), cron(8) ou anacron(8).

10.3 Infrastructure de sécurité des données

L'infrastructure de sécurité des données est fournie par la combinaison d'un outil de chiffrement des données, d'un outil de condensé de messages et d'un outil de signature.

paquet	popularité	taille	commande	description
gnupg	V:403, I:853	465	gpg(1)	GNU Privacy Guard - outil de signature et de chiffrement OpenPGP
gpgv	V:206, I:911	576	gpgv(1)	GNU Privacy Guard - outil de vérification de signature OpenPGP
sq	V:1, I:37	24191	sq(1)	Sequoia-PGP - OpenPGP encryption and outil de signature et de chiffrement OpenPGP
sqv	V:517, I:595	1919	sqv(1)	Sequoia-PGP - outil de vérification de signature OpenPGP
paperkey	V:1, I:12	58	paperkey(1)	extraire seulement l'information secrète de clés secrètes OpenPGP
cryptsetup	V:35, I:77	476	cryptsetup(8), …	utilitaires pour chiffrement de périphérique en mode bloc dm-crypt prenant en charge le chiffrement LUKS
coreutils	V:914, I:1000	19751	md5sum(1)	calculer et vérifier un condensé MD5 de message
coreutils	V:914, I:1000	19751	sha1sum(1)	calculer et vérifier un condensé SHA1 de message
openssl	V:853, I:996	2564	`openssl`(1ssl)	calculer un condensé de message avec « `openssl dgst` » (OpenSSL)
libsecret-tools	V:2, I:14	44	secret-tool(1)	stocker et récupérer les mots de passe (CLI)
seahorse	V:76, I:258	7999	seahorse(1)	outil de gestion des clés (GNOME)

Table 10.6 – Liste des outils d'une infrastructure de sécurité des données

Consultez la Section 9.9 sur dm-crypt et fscrypt qui mettent en œuvre une infrastructure de chiffrement automatique des données à l'aide des modules du noyau Linux.

10.3.1 Gestion de clés pour GnuPG

Voici les commandes de GNU Privacy Guard pour la gestion de base des clés :

Voici la signification du code de confiance.

Ce qui suit génère ma clé « `9563FC29932C409F1A77F9C1AB8A1522D8234C6A` » .

```
$ gpg --gen-key
gpg (GnuPG) 2.4.7; Copyright (C) 2024 g10 Code GmbH
...
GnuPG needs to construct a user ID to identify your key.
```

commande	description
`gpg --gen-key`	générer une nouvelle clé
`gpg --gen-revoke my_user_ID`	générer une clé de révocation pour ID_de_mon_utilisateur
`gpg --edit-key user_ID`	éditer la clé de manière interactive, « help » pour obtenir de l'aide
`gpg -o file --export`	exporter toutes les clés publiques vers un fichier
`gpg -o file --export-secret-keys`	exporter toutes les clés privées vers un fichier
`gpg --import file`	importer toutes les clés depuis fichier
`gpg --send-keys user_ID`	envoyer la clé de ID_utilisateur vers le serveur de clés
`gpg --recv-keys user_ID`	recevoir la clé de ID_utilisateur du serveur de clés
`gpg --list-keys user_ID`	afficher la liste des clés de ID_utilisateur
`gpg --list-sigs user_ID`	afficher la liste des signatures de ID_utilisateur
`gpg --check-sigs user_ID`	vérifier la signature de ID_utilisateur
`gpg --fingerprint user_ID`	vérifier l'empreinte de ID_utilisateur
`gpg --refresh-keys`	mettre à jour le porte-clé local

Table 10.7 – Liste des commandes de GNU Privacy Guard pour la gestion des clés

code	description de la confiance
`-`	pas de confiance d'utilisateur assignée/pas encore calculée
`e`	échec du calcul de la confiance
`q`	pas assez d'informations pour le calcul
`n`	ne jamais faire confiance à cette clé
`m`	confiance marginale
`f`	confiance complète
`u`	confiance ultime

Table 10.8 – Liste de la signification des codes de confiance

```
Real name: Osamu Aoki
Email address: osamu@debian.org
You selected this USER-ID:
    "Osamu Aoki <osamu@debian.org>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
...
public and secret key created and signed.

pub   ed25519 2026-02-14 [SC] [expires: 2029-02-13]
      9563FC29932C409F1A77F9C1AB8A1522D8234C6A
uid                      Osamu Aoki <osamu@debian.org>
sub   cv25519 2026-02-14 [E] [expires: 2029-02-13]
```

Ce qui suit permet d'envoyer ma clé « 9563FC29932C409F1A77F9C1AB8A1522D8234C6A » vers le serveur de clés populaire « hkp://keys.gnupg.net » :

```
$ gpg --keyserver hkp://keys.gnupg.net --send-keys 9563FC29932C409F1A77F9C1AB8A1522D8234C6A
```

Une bonne configuration de serveur de clés dans « ~/.gnupg/gpg.conf » (ou à l'ancien emplacement « ~/.gnupg/opti contient ce qui suit :

```
keyserver hkp://keys.gnupg.net
```

Ce qui suit obtient les clés inconnues du serveur de clés :

```
$ gpg --list-sigs --with-colons | grep '^sig.*\[User ID not found\]' |\
          cut -d ':' -f 5| sort | uniq | xargs gpg --recv-keys
```

Il y avait un bogue dans OpenPGP Public Key Server (pre version 0.9.6) qui corrompait les clés ayant plus de 2 sous-clés. Le paquet du serveur gnupg (>1.2.1-2) peut gérer ces sous-clés corrompues. Consultez gpg(1) sous l'option « --repair-pks-subkey-bug ».

Use of SHA-1 for hash has been deprecated. If your old RSA based openPGP key uses SHA-1 for hash, fix it using FixKeyWithSha1.

ASTUCE
The sq(1) and sqv(1) commands are an alternative set of openPGP commands. See sq user documentation and A Practical Introduction to using sq, Sequoia PGP's CLI.

10.3.2 Utilisation de GnuPG sur des fichiers

Voici des exemples d'utilisation des commandes de GNU Privacy Guard sur des fichiers :

10.3.3 Utiliser GnuPG avec Mutt

Ajoutez ce qui suit à « ~/.muttrc » afin d'éviter que GnuPG qui est lent ne démarre automatiquement, tout en permettant son utilisation en entrant « S » depuis l'index du menu :

```
macro index S ":toggle pgp_verify_sig\n"
set pgp_verify_sig=no
```

commande	description
`gpg -a -s file`	signer « fichier » dans un fichier ASCII blindé fichier.asc
`gpg --armor --sign file`	,,
`gpg --clearsign file`	signer un fichier en clair
`gpg --clearsign file|mail foo@example.org`	envoyer un message signé en clair à `truc@example.org`
`gpg --clearsign --not-dash-escaped patchfile`	signer en clair un fichier_rustine
`gpg --verify file`	vérifier fichier signé en texte clair
`gpg -o file.sig -b file`	créer une signature détachée
`gpg -o file.sig --detach-sign file`	,,
`gpg --verify file.sig file`	vérifier un fichier avec fichier.sig
`gpg -o crypt_file.gpg -r name -e file`	chiffrement par clé publique destiné au « nom » depuis le « fichier » vers « fichier_chiffré.gpg » binaire
`gpg -o crypt_file.gpg --recipient name --encrypt file`	,,
`gpg -o crypt_file.asc -a -r name -e file`	chiffrement par clé publique destiné au « nom » depuis le « fichier » vers le fichier ASCII blindé « fichier_chiffré.asc »
`gpg -o crypt_file.gpg -c file`	chiffrement symétrique depuis « fichier » vers « fichier_chiffré.gpg »
`gpg -o crypt_file.gpg --symmetric file`	,,
`gpg -o crypt_file.asc -a -c file`	chiffrement symétrique prévu destiné au « nom » depuis le « fichier » vers le fichier ASCII blindé « fichier_chiffré.asc »
`gpg -o file -d crypt_file.gpg -r name`	déchiffrement
`gpg -o file --decrypt crypt_file.gpg`	,,

Table 10.9 – Liste des commandes de GNU Privacy Guard sur des fichiers

10.3.4 Utiliser GnuPG avec Vim

Le greffon `gnupg` vous permet de lancer GnuPG de manière transparente pour les fichiers ayant l'extension « `.gpg` », « `.asc` » et « `.pgp` ». 3

```
$ sudo aptitude install vim-scripts
$ echo "packadd! gnupg" >> ~/.vim/vimrc
```

10.3.5 La somme de contrôle MD5

md5sum(1) fournit un utilitaire permettant de créer un fichier de résumé en utilisant la méthode se trouvant dans rfc1321 et en l'utilisant pour vérifier chaque fichier qu'il contient.

```
$ md5sum foo bar >baz.md5
$ cat baz.md5
d3b07384d113edec49eaa6238ad5ff00  foo
c157a79031e1c40f85931829bc5fc552  bar
$ md5sum -c baz.md5
foo: OK
bar: OK
```

Note
Le calcule de somme de contrôle MD5 consomme moins de ressources processeur que celles utilisées pour le chiffrement des signatures en utilisant GNU Privacy Guard (GnuPG). Habituellement, pour s'assurer de l'intégrité des données, seul le résumé de plus haut niveau est signé par chiffrement.

10.3.6 Trousseau de mots de passe

Sur le système GNOME, l'outil graphique seahorse(1) gère les mots de passe et les stocke de manière sécurisée dans le trousseau `~/.local/share/keyrings/*`.

secret-tool(1) peut stocker le mot de passe du trousseau de clés à partir de la ligne de commande.

Stockons la phrase secrète utilisée pour l'image disque cryptée LUKS/dm-crypt :

```
$ secret-tool store --label='LUKS passphrase for disk.img' LUKS my_disk.img
Password: ********
```

Ce mot de passe stocké peut être récupéré et transmis à d'autres programmes, par exemple, cryptsetup(8) :

```
$ secret-tool lookup LUKS my_disk.img | \
  cryptsetup open disk.img disk_img --type luks --keyring -
$ sudo mount /dev/mapper/disk_img /mnt
```

ASTUCE
Chaque fois que vous devez fournir un mot de passe dans un script, utilisez `secret-tool` et évitez de coder directement en dur la phrase secrète qu'il contient.

10.4 Outils pour fusionner le code source

Il existe de nombreux outils pour fusionner du code source. Les commandes qui suivent ont attiré mon attention :

3. Si vous utilisez « `~/.vimrc` » au lieu de « `~/.vim/vimrc` », veuillez substituer en conséquence.

paquet	popularité	taille	commande	description
patch	V:100, I:729	243	patch(1)	appliquer un fichier de différences (« diff ») à un original
vim	V:73, I:330	4415	vimdiff(1)	comparer deux fichiers côte à côte dans vim
imediff	V:0.02, I:0.21	329	imediff(1)	outil interactif plein écran pour fusionner 2 ou 3 fichiers
meld	V:5, I:23	3996	meld(1)	comparer et fusionner des fichiers (GTK)
wiggle	V:0.01, I:0.12	191	wiggle(1)	appliquer les rustines rejetées
diffutils	V:901, I:998	1768	diff(1)	comparer des fichiers ligne à ligne
diffutils	V:901, I:998	1768	diff3(1)	comparer et fusionner trois fichiers ligne par ligne
quilt	V:2, I:16	880	quilt(1)	gérer une série de rustines
wdiff	V:5, I:37	580	wdiff(1)	afficher les différences de mots entre deux fichiers texte
diffstat	V:8, I:91	86	diffstat(1)	afficher un histogramme des modifications apportées par le fichier de différences
patchutils	V:10, I:91	269	combinediff(1)	créer une rustine (patch) cumulative à partir de deux rustines incrémentales
patchutils	V:10, I:91	269	dehtmldiff(1)	extraire un fichier de différences d'une page HTML
patchutils	V:10, I:91	269	filterdiff(1)	extraire ou exclure des différences d'un fichier de différences
patchutils	V:10, I:91	269	fixcvsdiff(1)	corriger les fichiers de différences créés par CVS que patch(1) interprète mal
patchutils	V:10, I:91	269	flipdiff(1)	échanger l'ordre de deux rustines
patchutils	V:10, I:91	269	grepdiff(1)	afficher quels sont les fichiers modifiés par une rustine correspondant un une expression rationnelle
patchutils	V:10, I:91	269	interdiff(1)	afficher les différences entre deux fichiers de différence unifiés
patchutils	V:10, I:91	269	lsdiff(1)	afficher quels sont les fichiers modifiés par une rustine
patchutils	V:10, I:91	269	recountdiff(1)	recalculer les nombres et les décalages dans un contexte unifié de fichiers de différences
patchutils	V:10, I:91	269	rediff(1)	corriger les décalages et les nombres d'un fichier de différences édité manuellement
patchutils	V:10, I:91	269	splitdiff(1)	séparer les rustines incrémentales
patchutils	V:10, I:91	269	unwrapdiff(1)	réparer les correctifs dont les mots ont été coupés
dirdiff	V:0.1, I:1.3	167	dirdiff(1)	afficher les différences et fusionner les modifications entre deux arbres de répertoires
docdiff	V:0.02, I:0.15	554	docdiff(1)	comparer deux fichiers mot par mot ou caractère par caractère
makepatch	V:0.01, I:0.14	99	makepatch(1)	créer des fichiers de rustines étendus
makepatch	V:0.01, I:0.14	99	applypatch(1)	appliquer des fichiers de rustines étendus

Table 10.10 – Liste d'outils destinés à fusionner du code source

10.4.1 Extraire des différences pour des fichiers sources

La procédure suivante vous permet d'extraire les différences entre deux fichiers sources et créer un fichier de différences unifié « `fichier.patch0` » ou « `fichier.patch1` » selon l'emplacement du fichier.

```
$ diff -u file.old file.new > file.patch0
$ diff -u old/file new/file > file.patch1
```

10.4.2 Fusionner les mises à jour des fichiers source

Le fichier de différences (« diff » (encore appelé fichier « patch » ou rustine) est utilisé pour envoyer une mise à jour de programme. Celui qui reçoit applique cette mise à jour à un autre fichier de la manière suivante :

```
$ patch -p0 file < file.patch0
$ patch -p1 file < file.patch1
```

10.4.3 Fusion interactive

Si vous avez deux versions d'un code source, vous pouvez effectuer une fusion des deux de manière interactive en utilisant imediff(1) de la manière suivante :

```
$ imediff -o file.merged file.old file.new
```

Si vous avez trois versions d'un code source, vous pouvez effectuer une fusion des trois de manière interactive en utilisant imediff(1) de la manière suivante :

```
$ imediff -o file.merged file.yours file.base file.theirs
```

10.5 Git

Git est l'outil de prédilection de nos jours pour un système de gestion de versions (VCS) car Git peut tout faire pour la gestion à la fois du code source local et celui distant.

Debian fournit des services Git gratuits avec le service Salsa de Debian. Sa documentation se trouve sur https://wiki.debian.or
Salsa.

Voici quelques paquets relatifs à Git.

10.5.1 Configuration du client Git

Vous pourrez définir certains éléments de configuration globaux, comme votre nom et votre adresse de courriel utilisée par Git, dans « `~/.gitconfig` » de la manière suivante :

```
$ git config --global user.name "Name Surname"
$ git config --global user.email yourname@example.com
```

Vous pouvez également personnaliser le comportement par défaut de Git de la manière suivante :

```
$ git config --global init.defaultBranch main
$ git config --global pull.rebase true
$ git config --global push.default current
```

paquet	popularité	taille	commande	description
git	V:442, I:630	52512	git(1)	Git, système de contrôle de version rapide, évolutif et distribué
gitk	V:3, I:25	2045	gitk(1)	Interface graphique de navigateur de dépôt Git avec historique
qgit	V:0.2, I:1.8	1431	qgit(1)	Interface graphique de navigateur de dépôt Git avec historique
git-cola	V:0.7, I:4.4	5331	git-cola(1)	Interface graphique de navigateur de dépôt Git avec historique
tig	V:2, I:11	1245	tig(1)	console Git repository browser with history
lazygit	V:1.0, I:4.1	24167	lazygit(1)	console Git repository browser with history
git-gui	V:1, I:15	2549	git-gui(1)	Interface graphique pour Git (pas d'historique)
git-email	V:0.4, I:8.6	1229	git-send-email(1)	envoyer une série de rustines sous forme de courriel à partir de Git
git-buildpackage	V:1.2, I:6.5	2027	git-buildpackage(1)	automatise la mise en paquet Debian avec Git
dgit	V:0.12, I:0.91	758	dgit(1)	interopérabilité de git avec l'archive Debian
imediff	V:0.02, I:0.21	329	git-ime(1)	outil interactif d'aide pour des commits séparés de git
stgit	V:0.07, I:0.45	6005	stg(1)	quilt par-dessus git (Python)
git-doc	I:9.6	15192	N/A	Documentation officielle de git
gitmagic	I:0.43	721	N/A	« Git Magic », le guide le plus facile à comprendre pour Git

Table 10.11 – Liste des paquets et des commandes relatifs à git

Si vous avez l'habitude d'utiliser les commandes de CVS ou de Subversion, vous pourrez définir certains alias de commandes comme suit :

```
$ git config --global alias.ci "commit -a"
$ git config --global alias.co checkout
```

Vous pouvez vérifier votre configuration globale de la manière suivante :

```
$ git config --global --list
```

10.5.2 Commandes de base pour Git

L'opération Git implique plusieurs données.

- L'arborescence de travail qui contient les fichiers concernant l'utilisateur et auxquels vous apportez des modifications.
  - Les changements à enregistrer doivent être explicitement sélectionnés et ajoutés dans un index. Il s'agit des commandes `git add` et `git rm`.
- L'index qui contient les fichiers ajoutés.
  - Les fichiers ajoutés seront validés dans le référentiel local lors de la demande suivante. Il s'agit de la commande `git commit`.
- Le dépôt local qui détient des fichiers validés.
  - Git enregistre l'historique lié aux données validées et organise celles-ci en branches dans le dépôt.
  - Le dépôt local peut envoyer des données au dépôt distant avec la commande `git push`.
  - Le dépôt local peut recevoir des données du dépôt distant avec les commandes `git fetch` et `git pull`.

  - La commande `git pull` effectue `git merge` ou `git rebase` après la commande `git fetch`.
  - Ici, `git merge` combine finalement deux branches distinctes de l'historique à un certain moment. (C'est par défaut `git pull` sans personnalisation et cela peut être valable pour les personnes en amont qui publient la branche vers de nombreuses personnes.)
  - Ici, `git rebase` crée une seule branche de l'historique séquentiel de la branche distante, suivie par la branche locale. (C'est le cas de la personnalisation`pull.rebase true` et cela peut être valable pour nous autres.)
- Le dépôt distant qui contient des fichiers validés.
  - La communication vers le dépôt distant utilise des protocoles de communication sécurisés tels que SSH ou HTTPS.

L'arborescence de travail est constituée de fichiers situés en dehors du répertoire `.git/`. Les fichiers contenus dans le répertoire `.git/` contiennent l'index, les données du dépôt local et certains fichiers texte de configuration git.

Voici un aperçu des principales commandes Git.

commande de Git	**fonction**
`git init`	créer le dépôt (local)
`git clone URL`	cloner le dépôt distant vers un dépôt local avec l'arborescence de travail
`git pull origin main`	mettre à jour la branche locale `main` avec le dépôt distant `origin`
`git add .`	ajouter le(s) fichier(s) de l'arbre de travail à l'index pour les fichiers préexistants dans l'index seulement
`git add -A .`	ajouter le(s) fichier(s) dans l'arbre de travail à l'index pour tous les fichiers, y compris les suppressions
`git rm filename`	supprimer des fichiers de l'arborescence de travail et de l'index
`git commit`	valider les modifications préparées dans l'index dans le dépôt
`git commit -a`	ajouter toutes les modifications de l'arbre de travail à l'index et les valider dans le référentiel local (add + commit)
`git push -u origin branch_name`	mettre à jour le dépôt distant `origine` avec la `branche` locale (invocation initiale)
`git push origin branch_name`	mettre à jour le dépôt distant `origine` avec la `branche` locale (invocation ultérieure)
`git diff treeish1 treeish2`	afficher la différence entre le commit *arbre1* et *arbre2*
`gitk`	affichage graphique de l'arbre des historiques de branche du dépôt VCS

Table 10.12 – Principales commandes de Git

10.5.3 Conseils pour Git

Voici quelques conseils pour Git.



AVERTISSEMENT
Ne pas utiliser d'espaces dans la chaîne de balise même si certains outils comme gitk(1) vous permettent de le faire. Cela peut perturber d'autres commandes de `git`.



Attention
Si une branche locale qui a été intégrée dans un dépôt distant est rebasée ou compressée, pousser cette branche comporte des risques et nécessite l'option `--force`. Cela n'est généralement pas acceptable pour la branche `main` mais peut l'être pour une branche_sujet avant de fusionner avec la branche `main` .

ligne de commande de Git	fonction
`gitk --all`	voir l'historique complet de Git et agir dessus, par exemple en réinitialisant HEAD à un autre commit, en choisissant librement des correctifs, en créant des étiquettes et des branches...
`git stash`	revenir à un arbre de travail propre sans perdre de données
`git remote -v`	vérifier les paramètres pour le dépôt distant
`git branch -vv`	vérifier les paramètres de la branche de dépôt
`git status`	afficher l'état de l'arborescence de travail
`git config -l`	lister les paramètres de git
`git reset --hard HEAD; git clean -x -d -f`	revenir sur toutes les modifications de l'arbre de travail et les nettoyer complètement
`git rm --cached filename`	annuler l'ajout dans l'index intermédiaire modifié par `git add filename`
`git reflog`	obtenir le journal de référence (utile pour récupérer les commits de la branche supprimée)
`git branch new_branch_name HEAD@{6}`	créer une nouvelle branche à partir des informations de reflog
`git remote add new_remote URL`	ajouter un dépôt distant `nouveau_distant` pointé par URL
`git remote rename origin upstream`	renommer le nom du dépôt distant de `origine` à `amont`
`git branch -u upstream/branch_name`	définir le suivi à distance du dépôt distant `amont` et son nom de branche `nom_branche`.
`git remote set-url origin https://foo/bar.git`	changer l'URL de `origine`
`git remote set-url --push upstream DISABLED`	désactiver push vers `amont` (Éditer `.git/config` pour le réactiver)
`git remote update upstream`	récupérer les mises à jour de toutes les branches distantes du dépôt `upstream`
`git fetch upstream foo:upstream-foo`	créer une branche locale (éventuellement orpheline) `upstream-foo` en tant que copie de la branche `foo` du dépôt `upstream`
`git checkout -b topic_branch ; git push -u topic_branch origin`	créer une nouvelle branche `branche_sujet` et l'incorporer dans `origine`
`git branch -m oldname newname`	renommer le nom de la branche locale
`git push -d origin branch_to_be_removed`	supprimer la branche distante (nouvelle méthode)
`git push origin :branch_to_be_removed`	supprimer la branche distante (ancienne méthode)
`git checkout --orphan unconnected`	créer une nouvelle branche `sans lien`
`git rebase -i origin/main`	réorganiser/abandonner/écraser les validations de `origine/main` pour nettoyer l'historique des branches
`git reset HEAD^; git commit --amend`	compresser les deux derniers commits en un seul
`git checkout topic_branch ; git merge --squash topic_branch`	agglomérer l'ensemble de `branche_sujet` dans un commit
`git fetch --unshallow --update-head-ok origin '+refs/heads/*:refs/heads/*'`	convertir un clone superficiel en un clone complet de toutes les branches
`git ime`	diviser le dernier commit en une série de commits plus petits, fichier par fichier, etc. (paquet `imediff` requis)
`git repack -a -d; git prune`	reconditionner le dépôt local en un seul paquet (cela peut limiter les possibilités de récupération de données perdues à partir d'une branche effacée, etc.)

Table 10.13 – Conseils pour Git

Attention
L'appel d'une sous-commande `git` directement avec « `git-xyz` » depuis la ligne de commandes est devenu obsolète depuis début 2006.

ASTUCE
S'il existe un fichier exécutable « `git-toto` » dans le chemin spécifié par « `$PATH` » alors entrer la commande « `git toto` » sans trait d'union dans la ligne de commande appellera ce « `git-toto` ». C'est une fonctionnalité de la commande « `git` ».

10.5.4 Références de Git

Consultez ce qui suit.

— page de manuel : git(1) (`/usr/share/doc/git-doc/git.html`)
— Manuel de l'utilisateur de Git (`/usr/share/doc/git-doc/user-manual.html`)
— Un tutoriel d'introduction à git « A tutorial introduction to git » (`/usr/share/doc/git-doc/gittutorial.html`)
— Un tutoriel d'introduction à git : deuxième partie (`/usr/share/doc/git-doc/gittutorial-2.html`)
— Utilisation de tous les jours de GIT en 20 commandes (`/usr/share/doc/git-doc/giteveryday.html`)
— La magie de Git « Git Magic » (`/usr/share/doc/gitmagic/html/index.html`)

10.5.5 Autres systèmes de gestion de versions

Les systèmes de gestion de versions (VCS) sont parfois appelés système de gestion de révisions (RCS) ou gestion de configuration logicielle (SCM).

Voici un récapitulatif des autres VCS notables pour le système Debian.

paquet	popularité	taille	outil	type du VCS	commentaire
`mercurial`	V:3, I:24	2667	Mercurial	distribué	DVCS en Python avec un peu de C
`darcs`	V:0.2, I:2.9	38856	Darcs	distribué	DVCS avec une algèbre intelligente des rustines (lent)
`tla`	V:0.03, I:0.60	1022	GNU arch	distribué	DVCS principalement par Tom Lord (historique)
`bzr`	V:0.1, I:3.8	28	GNU Bazaar	distribué	DVCS influencé par `tla` écrit en Python
`subversion`	V:9, I:53	5211	Subversion	distant	« CVS en mieux », remplaçant standard du VCS (historique) distant
`cvs`	V:3, I:22	4835	CVS	distant	VCS distant standard précédent (historique)
`tkcvs`	V:0.10, I:0.87	34	CVS, …	distant	interface graphique d'affichage d'une arborescence de dépôt VCS (CVS, Subversion, RCS)
`rcs`	V:1.6, I:8.8	578	RCS	local	« SCCS UNIX en mieux » (historique)
`cssc`	V:0.01, I:0.27	2044	CSSC	local	clone de SCCS UNIX (historique)

Table 10.14 – Liste des autres outils de système de gestion de versions

Chapitre 11

Conversion de données

Description des outils et astuces pour convertir différents formats de données sur un système Debian.

Les outils basés sur des standards sont de très bonne qualité mais la prise en charge des formats propriétaires est limitée.

11.1 Outils de conversion de données textuelles

Mes yeux ont été attirés par les paquets suivants de conversions de données textuelles :

paquet	popularité	taille	mot clé	description
libc6	V:924, I:1000	5483	jeu de caractères	convertisseur de codage de texte entre différents paramètres linguistiques à l'aide d'iconv(1) (fondamental)
recode	V:2, I:12	534	charset+eol	convertisseur de codage de texte entre différents paramètres linguistiques (flexible avec plus d'alias et de fonctionnalités)
konwert	V:2, I:42	137	jeu de caractères	convertisseur de codage de texte entre différents paramètres linguistiques (le luxe)
nkf	V:0.4, I:8.3	360	jeu de caractères	jeux de caractères pour le japonais
tcs	V:0.01, I:0.12	518	jeu de caractères	jeu de caractères du traducteur
unaccent	V:0.03, I:0.26	35	jeu de caractères	remplacer les lettres accentuées par leur équivalent accentué
tofrodos	V:1, I:11	50	eol (fin de ligne)	convertisseur de format de texte entre DOS et UNIX : fromdos(1) et todos(1)
macutils	V:0.03, I:0.38	319	eol (fin de ligne)	convertisseur de format de texte entre Macintosh et UNIX :frommac(1) et tomac(1)

Table 11.1 – Liste des outils de conversion de texte

11.1.1 Convertir un fichier texte avec iconv

ASTUCE
iconv(1) fait partie du paquet libc6 et est toujours disponible sur pratiquement tous les systèmes de type Unix pour permettre la conversion des codages de caractères.

Vous pouvez convertir les codages de caractères d'un fichier texte par iconv(1) en effectuant ce qui suite :

```
$ iconv -f encoding1 -t encoding2 input.txt >output.txt
```

La valeur des codages n'est pas sensible à la casse et ignore « - » et « _ » pour la correspondance. On peut vérifier quels sont les codages pris en charge à l'aide de la commande « `iconv -l` ».

valeur de codage	utilisation
ASCII	American Standard Code for Information Interchange (Code américain standard pour l'échange d'informations), code sur 7 bits sans caractère accentué
UTF-8	norme actuelle multi-langues pour tous les systèmes d'exploitation modernes
ISO-8859-1	ancienne norme pour les langues d'Europe de l'ouest, ASCII + lettres accentuées
ISO-8859-2	ancienne norme pour les langues d'Europe de l'est, ASCII + lettres accentuées
ISO-8859-15	ancienne norme pour les langues d'Europe de l'ouest, ISO-8859-1 avec le signe euro
CP850	page de code 850, caractères de DOS Microsoft avec glyphes pour les langues de l'Europe de l'est, variante de ISO-8859-1
CP932	page de code 932, variante style Microsoft Windows de Shift-JIS pour le japonais
CP936	page de code 936, variante style Microsoft Windows de GB2312, GBK ou GB18030 pour le chinois simplifié
CP949	page de code 949, variante style Microsoft Windows de EUC-KR de « Unified Hangul Code » pour le coréen
CP950	page de code 950, variante style Microsoft Windows de Big5 pour le chinois traditionnel
CP1251	page de code 1251, encodage de style Microsoft Windows pour l'alphabet cyrillique
CP1252	page de code 1252, variante style Microsoft Windows de ISO-8859-15 pour les langues d'Europe de l'ouest
KOI8-R	ancienne norme UNIX de russe pour l'alphabet cyrillique
ISO-2022-JP	codage standard du courrier électronique japonais n'utilisant que des codes à 7 bits
eucJP	ancien code UNIX de japonais sur 8 bits et complètement différent de Shift-JIS
Shift-JIS	norme JIS X 0208 Annexe 1 pour le japonais (consultez CP932)

Table 11.2 – Liste de valeurs de codage et leur utilisation

Note
Certains encodages ne sont pris en charge que pour la conversion de données et ne sont pas utilisés comme valeurs de paramètres régionaux (Section 8.1).

Pour les jeux de caractères qui tiennent dans un seul octet tels que les jeux de caractères ASCII et ISO-8859, le codage des caractères signifie à peu près la même chose que le jeu de caractères.

Pour les jeux de caractères ayant de nombreux caractères tels que JIS X 0213 pour le japonais ou Universal Character Set (UCS, Unicode, ISO-10646-1) (jeu de caractère universel) pour pratiquement toutes les langues, il y a de nombreux schémas de codage pour les insérer dans les séquences d'octets de données.

— EUC et ISO/IEC 2022 (connu aussi en tant que JIS X 0202) pour le japonais
— UTF-8, UTF-16/UCS-2 et UTF-32/UCS-4 pour l'Unicode

Pour ceux-ci, il y a une différence claire entre le jeu de caractères et le codage des caractères.

Page de code est utilisée comme synonyme de table de codage de caractères pour certaines d'entre-elles spécifiques au fournisseur.

Note
Veuillez remarquer que la plupart des systèmes de codage partagent le même code avec ASCII pour les caractères sur 7 bits. Mais il y a quelques exceptions, lors de la conversion de programmes C et des données d'URL anciens en japonais depuis ce qui est parfois appelé format de codage shift-JIS vers le format UTF-8, utilisez « `CP932` » comme nom de codage plutôt que « `shift-JIS` » afin d'obtenir le résultat attendu : `0x5C` → « \ » et `0x7E` → « ~ ». Sinon, ils seront convertis vers les mauvais caractères.

ASTUCE
On peut aussi utiliser recode(1) qui offre plus de fonctionnalités que celles combinées de iconv(1), fromdos(1), todos(1), frommac(1) et tomac(1). Pour plus de détails, consultez « `info recode` ».

11.1.2 Vérifier que les fichiers sont codés en UTF-8 avec iconv

Vous pouvez vérifier si un fichier texte est codé en UTF-8 à l'aide d'iconv(1) en effectuant ce qui suit :

```
$ iconv -f utf8 -t utf8 input.txt >/dev/null || echo "non-UTF-8 found"
```

ASTUCE
Utilisez l'option « `--verbose` » dans les exemples ci-dessus pour trouver le premier caractère non UTF-8.

11.1.3 Convertir les noms de fichiers avec iconv

Voici un exemple de script pour convertir le codage des noms de fichiers, dans un seul répertoire, depuis celui créé par un ancien système d'exploitation vers celui d'un système d'exploitation moderne en UTF-8.

```
#!/bin/sh
ENCDN=iso-8859-1
for x in *;
 do
 mv "$x" "$(echo "$x" | iconv -f $ENCDN -t utf-8)"
done
```

La variable « `$ENCDN` » spécifie l'encodage d'origine des noms de fichier sous d'anciens OS de la même façon que dans Tableau 11.2.

Pour les cas plus compliqués, veuillez monter le système de fichiers (par exemple une partition d'un disque dur) contenant de tels noms de fichiers avec le codage correct comme option de mount(8) (consultez Section 8.1.3) et copier son contenu complet vers un autre système de fichiers monté en UTF-8 avec la commande « `cp -a` ».

11.1.4 Convertir les fins de ligne (EOL)

Le format de fichier texte, particulièrement le code de fin de ligne (EOL), dépend de la plateforme.

The EOL format conversion programs, fromdos(1) or todos(1) (see package `tofrodos`) and frommac(1) or tomac(1) (see package `macutils`), are quite handy. recode(1) is also useful.

plateforme	code pour EOL	contrôle	décimal	hexadécimal
Debian (unix)	LF	^J	10	0A
MSDOS et Windows	CR-LF	^M^J	13 10	0D 0A
Macintosh d'Apple	CR	^M	13	0D

Table 11.3 – Liste des styles d'EOL pour différentes plateformes

Note
La plupart des éditeurs (par exemple `vim`, `emacs`, `gedit`…) peuvent prendre en compte de manière transparente les fichiers ayant une fin de ligne (EOL) de style MSDOS.

ASTUCE
L'utilisation de « `sed -e '/\r$/!s/$/\r/'` » en remplacement de todos(1) est préférable lorsque vous désirez unifier le style de caractère de fin de ligne vers le style MSDOS depuis un style mixte MSDOS et UNIX (par exemple, après avoir fusionné deux fichiers de style MSDOS avec diff3(1)). Cela parce que `todos` ajoute un retour charriot (CR) à toutes les lignes.

11.1.5 Convertir les tabulations (TAB)

Il y a quelques programmes spécialisés dans la conversion des codes de tabulations.

fonction	`bsdmainutils`	`coreutils`
étendre les tabulations en espaces	"`col -x`"	`expand`
convertir les espaces en tabulation	"`col -h`"	`unexpand`

Table 11.4 – Liste des commande de conversion de TAB des paquets `bsdmainutils` et `coreutils`

indent(1) du paquet `indent` reformate complètement les espaces dans un programme en C.

Des programmes d'édition tels que `vim` et `emacs` peuvent aussi être utilisés pour la conversion des tabulations. Par exemple avec `vim`, vous pouvez étendre les tabulation avec la séquence de commandes « `:set expandtab` » et « `:%retab` ». Vous pouvez revenir en arrière de cette conversion par la séquence de commandes « `:set noexpandtab` » et « `:%retab!` ».

11.1.6 Éditeurs avec conversion automatique

Les éditeurs de textes modernes et intelligents comme le programme `vim` sont assez habiles et prennent assez bien en compte les systèmes de codage de caractères et tous les formats de fichiers. Vous devriez utiliser ces éditeurs avec les paramètres linguistiques UTF-8 dans une console compatible avec UTF-8 pour une meilleure compatibilité.

Un ancien fichier texte UNIX d'Europe de l'Ouest, « `u-file.txt` » enregistré dans le codage latin1 (iso-8859-1) peut être édité avec `vim` de la manière suivante :

```
$ vim u-file.txt
```

C'est possible car le mécanisme d'autodétection du codage du fichier dans `vim` suppose d'abord que le codage est UTF-8 et, s'il échoue, suppose qu'il est latin1.

Un ancien fichier texte en polonais, « `pu-file.txt` », enregistré avec le codage latin2 (iso-8859-2) peut être édité avec `vim` de la manière suivante :

```
$ vim '+e ++enc=latin2 pu-file.txt'
```

Un ancien fichier texte UNIX en japonais, « `ju-file.txt` », enregistré avec le codage eucJP peut être édité avec `vim` de la manière suivante :

```
$ vim '+e ++enc=eucJP ju-file.txt'
```

Un ancien fichier MS-Windows en japonais, « `jw-file.txt` », enregistré dans le codage appelé shift-JIS (plus précisément : CP932) peut être édité avec `vim` de la manière suivante :

```
$ vim '+e ++enc=CP932 ++ff=dos jw-file.txt'
```

Lorsqu'un fichier est ouvert avec les options « `++enc` » et « `++ff` », « `:w` » sur la ligne de commandes de Vim l'enregistre dans son format d'origine et écrase le fichier d'origine. Vous pouvez aussi indiquer le format d'enregistrement et le nom du fichier sur la ligne de commandes de Vim, par exemple, « `:w ++enc=utf8 nouveau.txt` ».

Veuillez vous rapporter à mbyte.txt « multi-byte text support » dans l'aide en ligne de `vim` et Tableau 11.2 pour les valeurs de paramètres linguistiques utilisés avec « `++enc` ».

La famille de programmes `emacs` peut effectuer des fonctions équivalentes.

11.1.7 Extraire du texte brut

Ce qui suit permet de lire une page web sous forme de fichier texte. C'est très utile pour copier des informations de configuration depuis le Web ou pour appliquer des outils textuels de base d'UNIX comme grep(1) à la page web .

```
$ w3m -dump https://www.remote-site.com/help-info.html >textfile
```

De la même manière, vous pouvez extraire des données en texte brut vers d'autres formats en utilisant ce qui suit :.

paquet	popularité	taille	mot clé	fonction
`w3m`	V:11, I:125	2853	`html→text`	convertisseur HTML vers texte avec la commande « `w3m -dump` »
`html2text`	V:4, I:66	298	`html→text`	convertisseur avancé HTML vers texte (ISO 8859-1)
`lynx`	V:30, I:454	2031	`html→text`	convertisseur HTML vers texte avec la commande « `lynx -dump` »
`elinks`	V:2, I:14	1839	`html→text`	convertisseur HTML vers texte avec la commande « `elinks -dump` »
`links`	V:2, I:19	2344	`html→text`	convertisseur HTML vers texte avec la commande « `links -dump` »
`links2`	V:0.8, I:9.2	5497	`html→text`	convertisseur HTML vers texte avec la commande « `links2 -dump` »
`catdoc`	V:16, I:169	682	`MSWord→text,TeX`	convertir les fichier MSWord en texte brut ou en TeX
`antiword`	V:0.9, I:6.1	596	`MSWord→text,ps`	convertir des fichiers MSWord en texte brut ou en ps
`unhtml`	V:0.04, I:0.50	40	`html→text`	supprimer les balise d'un fichier HTML
`odt2txt`	V:1, I:17	60	`odt→text`	convertisseur du texte OpenDocument vers texte

Table 11.5 – Liste d'outils pour extraite des données en texte brut

11.1.8 Mettre en évidence et formater des données en texte brut

Vous pouvez mettre en évidence et formater des données en texte brut de la manière suivante :

paquet	popularité	taille	mot clé	description
vim-runtime	V:18, I:347	39473	mise en évidence	MACRO Vim pour convertir du code source en HTML avec « :source $VIMRUNTIME/syntax/html.vim »
cxref	V:0.02, I:0.20	1191	c → html	convertisseur pour les programmes C vers latex et HTML (langage C)
src2tex	V:0.02, I:0.14	1799	mise en évidence	convertit de nombreux codes sources en TeX (langage C)
source-highlight	V:0.4, I:3.0	2131	mise en évidence	convertit de nombreux codes source vers des fichiers HTML, XHTML, LaTeX, Texinfo, séquences d'échappement en couleur ANSI et DocBook files avec mise en évidence (C++)
highlight	V:0.3, I:2.7	1452	mise en évidence	convertit de nombreux codes sources en fichiers HTML, XHTML, RTF, LaTeX, TeX ou XSL-FO avec mise en évidence (C++)
grc	V:0.9, I:5.4	208	text → color	coloriseur générique pour n'importe quoi (Python)
pandoc	V:12, I:48	209559	text → any	Marqueur de conversion global (Haskell)
python3-docutils	V:13, I:51	2012	text → any	formateur de documents ReStructured Text vers XML (Python)
markdown	V:0.4, I:5.5	56	text → html	formateur de document texte Markdown en (X)HTML (Perl)
asciidoctor	V:0.4, I:4.4	101	text → any	formateur de documents texte AsciiDoc vers XML/HTML (Ruby)
python3-sphinx	V:7, I:27	3245	text → any	système de publication de documents basé sur reStructured Text (Python)
hugo	V:0.8, I:5.0	66919	text → html	système de publication de site statique basé sur Markdown (Go)

Table 11.6 – Liste des outils pour mettre en évidence des données de texte brut

11.2 Données XML

Le langage de balisage extensible (« The Extensible Markup Language (XML) » est un langage de balisage de documents dont les informations sont structurées.

Consultez une introduction sur XML.COM.

— « Qu'est-ce qu'XML ? »
— « Qu'est-ce qu'XSLT ? »
— « Qu'est-ce qu'XSL-FO ? »
— « Qu'est-ce qu'XLink ? »

11.2.1 Conseils de base pour XML

Le texte en XML ressemble un peu à HTML. Il vous permet de gérer de nombreux formats de sortie pour un document. Un système XML facile est le paquet docbook-xsl qui est utilisé ici.

Chaque fichier XML commence par la déclaration XML standard suivante :

```
<?xml version="1.0" encoding="UTF-8"?>
```

La syntaxe de base d'un élément XML est balisée de la manière suivante :

```
<name attribute="value">content</name>
```

Un élément XML dont le contenu est vide est balisé de la façon raccourcie suivante :

```
<name attribute="value" />
```

« `attribute="value"` » dans les exemples ci-dessus est optionnel.

L'action commentaire en XML est balisée comme suit :

```
<!-- comment -->
```

En plus d'ajouter des balises, XML demande des conversions mineures de contenu en utilisant des entités prédéfinies pour les caractères suivants :

entité prédéfinie	**caractère devant être converti**
`"`	" : double apostrophe
`'`	' : apostrophe
`<`	< : inférieur à
`>`	> : supérieur à
`&`	& : esperluette

Table 11.7 – Liste des entités XML prédéfinies



Attention
« < » ou « & » ne peuvent pas être utilisés dans des attributs ni des éléments.

Note
Lorsque des entités définies par l'utilisateur de style SGML sont utilisées, par exemple « `&une_étiquette;` », la première définition prend le pas sur les suivantes. La définition de l'entité est exprimée par « `<!ENTITY une_étiquette "valeur entité">` ».

Note
Tant que le balisage XML est fait de manière cohérente avec un jeu particulier de nom de balises (soit certaines données comme valeur de contenu ou d'attribut), la conversion vers un autre XML est une tâche triviale en utilisant Extensible Stylesheet Language Transformations (XSLT).

11.2.2 Traitement XML

Il existe de nombreux outils pour traiter les fichiers XML tels quele langage extensible de feuilles de style (« the Extensible Stylesheet Language XSL «).

En gros, une fois créé un fichier XML correctement formaté, vous pouvez le convertir vers n'importe quel format en utilisant le langage extensible de transformation des feuilles de style (XSLT) (« Extensible Stylesheet Language Transformations »).

Le Langage extensible de feuilles de style pour le formatage des objets (XSL-FO) (« Extensible Stylesheet Language for Formatting Objects » est censé être la solution au formatage. Le paquet `fop` est nouveau dans l'archive `main` de Debian du fait de ses dépendances vers le langage de programmation Java. Le code LaTeX est donc habituellement

paquet	popularité	taille	mot clé	description
docbook-xml	V:15, I:403	2126	xml	Définition de type de document XML (DTD) pour DocBook
docbook-xsl	V:15, I:141	14823	xml/xslt	feuilles de style XSL pour le traitement de DocBook XML vers divers formats de sortie avec XSLT
xsltproc	V:15, I:69	83	xslt	processeur en ligne de commandes XSLT (XML → XML, HTML, texte brut, etc.)
xmlto	V:0.5, I:7.7	124	xml/xslt	convertisseur XML-vers-tout avec XSLT
fop	V:0.6, I:7.6	282	xml/xsl-fo	convertir les fichiers Docbook XML en PDF
dblatex	V:0.9, I:5.3	4635	xml/xslt	convertir les fichiers Docbook en documents DVI, PostScript, PDF avec XSLT
dbtoepub	V:0.04, I:0.52	37	xml/xslt	Convertisseur Docbook XML vers .epub

Table 11.8 – Liste d'outils XML

créé depuis XML en utilisant XSLT, et le système LaTeX est utilisé pour créer des fichiers imprimables comme DVI, PostScript et PDF.

Comme XML est un sous-ensemble du Langage généralisé de balisage (Standard Generalized Markup Language SGML »), il peut être traité par les nombreux outils disponibles pour SGML, comme Document Style Semantics and Specification Language (DSSSL).

paquet	popularité	taille	mot clé	description
openjade	V:1, I:20	1066	dsssl	processeur DSSSL à la norme ISO/IEC 10179:1996 DSSSL (le plus récent)
docbook-dsssl	V:0.5, I:7.2	2594	xml/dsssl	feuilles de style DSSSL pour le traitement des DocBook XML vers divers formats de sortie avec DSSSL
docbook-utils	V:0.3, I:5.1	287	xml/dsssl	utilitaires pour les fichiers DocBook y compris la conversion avec DSSSL vers d'autres formats (HTML, RTF, PS, man, PDF) avec des commandes docbook2*

Table 11.9 – Liste des outils DSSSL

ASTUCE
yelp de GNOME est parfois pratique pour lire les fichiers XML DocBook directement car il effectue un rendu propre sous X.

11.2.3 Extraire des données XML

Vous pouvez extraire des données HTML ou XML depuis d'autres formats en utilisant ce qui suit :

11.2.4 Analyse statique (lint) de données XML

Vous pouvez convertir les fichiers HTML non-XML en XHTML qui est une instance XML correctement formatée. XHTML peut être traité par les outils XML.

La syntaxe des fichiers XML et la qualité des URL qu'ils contiennent peuvent être vérifiées.

Une fois qu'un fichier XML propre est créé, vous pouvez utiliser la technologie XSLT pour extraire des données dans le contexte de balisage, etc.

paquet	popularité	taille	mot clé	description
man2html	V:0.1, I:1.2	142	manpage→html	convertisseur de page de manuel en HTML (prise en charge de CGI)
doclifter	V:0.00, I:0.05	487	troff→xml	convertisseur de troff vers DocBook XML
texi2html	V:0.2, I:2.6	1847	texi→html	convertisseur de Texinfo vers HTML
info2www	V:0.8, I:1.4	76	info→html	convertisseur depuis GNU info vers HTML (prise en charge de CGI)
wv	V:0.2, I:2.2	733	MSWord→any	convertisseur de document depuis Microsoft Word vers HTML, LaTeX, etc.
unrtf	V:0.3, I:2.7	158	rtf→html	convertisseur de documents de RTF vers HTML, etc
wp2x	V:0.00, I:0.07	200	WordPerfect→any	fichiers WordPerfect 5.0 et 5.1 vers TeX, LaTeX, troff, GML et HTML

Table 11.10 – Liste d'outils d'extraction de données XML

paquet	popularité	taille	fonction	description
libxml2-utils	V:64, I:204	214	xml↔html↔xml	outil XML en ligne de commandes xmllint(1) (vérification de la syntaxe, reformatage, décomposition…)
tidy	V:0.8, I:6.7	79	xml↔html↔xml	vérificateur de syntaxe et reformateur HTML
weblint-perl	V:0.05, I:0.85	32	lint	vérificateur de syntaxe et de style minimal pour HTML
linklint	V:0.05, I:0.45	343	vérification lint	vérificateur rapide de liens et outils de maintenance de sites web

Table 11.11 – Liste d'outils d'impression élégante du XML

11.3 Composition

Le programme UNIX troff, développé à l'origine par AT&T, peut être utilisé pour une composition simple. Il est habituellement utilisé pour créer des pages de manuel.

TeX, créé par Donald Knuth, est un outil très puissant de composition et c'est le standard de fait. LaTeX, écrit à l'origine par Leslie Lamport permet un accès de haut niveau à la puissance de TeX.

paquet	popularité	taille	mot clé	description
texlive	V:1, I:26	55	(La)TeX	système TeX pour la composition, l'aperçu et l'impression
lilypond	V:0.7, I:5.1	18010	(La)TeX	composition de partitions musicales
groff	V:2, I:22	16868	troff	système de formatage de texte GNU troff

Table 11.12 – Liste des outils de typographie

11.3.1 Composition roff

Traditionnellement, roff est le système de traitement de texte principal sous UNIX. Consultez roff(7), groff(7), groff(1), grotty(1), troff(1), groff_mdoc(7), groff_man(7), groff_ms(7), groff_me(7), groff_mm(7), et « `info groff` ».

Vous pouvez lire ou imprimer un bon didacticiel et document de référence sur la macro « `-me` » dans « `/usr/share/doc/gr` en installant le paquet `groff`.

ASTUCE
« `groff -Tascii -me -` » produit une sortie en texte brut avec du code d'échappement ANSI. Si vous désirez obtenir une sortie semblable à une page de manuel avec de nombreux « ^H » et « _ », utilisez plutôt « `GROFF_NO_SGR=1 groff -Tascii -me -` ».

ASTUCE
Pour supprimer les « ^H » et les « _ » d'un fichier texte issu de `groff`, filtrez le par « `col -b -x` ».

11.3.2 TeX/LaTeX

La distribution logicielle TeX Live offre un système TeX complet. Le métapaquet `texlive` fournit une bonne sélection de paquets de TeX Live qui devraient suffire aux tâches les plus courantes.

De nombreuses références sont disponibles pour TeX et LaTeX :
— The teTeX HOWTO : le guide local de teTeX sous Linux ;
— tex(1)
— latex(1)
— texdoc(1)
— texdoctk(1)
— « The TeXbook », par Donald E. Knuth, (Addison-Wesley) ;
— « LaTeX - A Document Preparation System », par Leslie Lamport, (Addison-Wesley) ;
— « The LaTeX Companion », par Goossens, Mittelbach, Samarin, (Addison-Wesley).

C'est l'environnement de composition typographique le plus puissant. De nombreux processeurs SGML l'utilisent comme processeur de texte en « backend ». Lyx qu'on trouvera dans le paquet `lyx` et GNU TeXmacs qui provient du paquet `texmacs` offrent un environnement d'édition agréable en mode WYSIWYG pour LaTeX bien que nombreux sont ceux qui choisissent d'utiliser Emacs et Vim pour éditer les sources.

De nombreuses ressources sont disponibles en ligne :

— The TEX Live Guide (”/usr/share/doc/texlive-doc/texlive/texlive-en/texlive-en.html”) (paquet texlive-base)
— Un guide simple pour Latex et Lyx (« A Simple Guide to Latex/Lyx »)
— Traitement de texte avec LaTeX (« Word Processing Using LaTeX »)

Lorsque les documents deviennent plus importants, TeX cause parfois des erreurs. Vous devez augmenter la taille de l'espace dans « /etc/texmf/texmf.cnf » (ou, de manière plus rigoureuse, éditez « /etc/texmf/texmf.d/95NonPat et lancez update-texmf(8)) afin de corriger cela.

Note
Le fichier source au format TeX de « The TeXbook » est disponible sur le site www.ctan.org de l'archive de TeX. Ce fichier contient la plupart des macros nécessaires. J'ai entendu dire que vous pouvez traiter ce document avec tex(1) après avoir commenté les lignes 7 à 10 et ajouté « \input manmac \proofmodefalse ». Je recommande vivement l'achat de ce livre (et de tous les autres livres de Donald E. Knuth) plutôt que d'utiliser la version en ligne, mais la version source est un très bon exemple d'utilisation de TeX !

11.3.3 Imprimer convenablement une page de manuel

Vous pouvez imprimer convenablement une page de manuel en PostScript en utilisant l'une des commandes suivantes :

```
$ man -Tps some_manpage | lpr
```

11.3.4 Créer une page de manuel

Bien que l'écriture d'une page de manuel (manpage) dans le format troff brut soit possible, il existe quelques paquets facilitant cette tâche :

paquet	popularité	taille	mot clé	description
docbook-to-man	V:0.6, I:5.2	189	SGML → manpage	macros de conversion de DocBook SGML vers roff man
help2man	V:0.5, I:6.0	542	text → manpage	générateur automatique d'une pages de manuel depuis --help
info2man	V:0.01, I:0.18	134	info → manpage	convertisseur depuis GNU info vers POD ou page de manuel
txt2man	V:0.05, I:0.59	112	text → manpage	convertir du texte brut ASCII au format d'une page de manuel

Table 11.13 – Liste de paquets facilitant la création de pages de manuel

11.4 Données imprimables

Sur un système Debian, les données imprimables sont définies dans le format PostScript. Common UNIX Printing System (CUPS) utilise Ghostscript en tant que programme de tramage « rasterisation » pour les imprimantes non-PostScript.

Les données imprimables peuvent aussi être exprimées au format PDF sur les systèmes Debian récents.

Les fichiers PDF peuvent être affichés et les entrées de formulaire peuvent être remplies en utilisant des outils d'affichage graphiques tels que Evince ou Okular (consulter la Section 7.4) et les navigateurs modernes tels que Chromium.

Les fichiers PDF peuvent être édités en utilisant quelques outils graphiques tels que LibreOffice, Scribus ou Inkscape (consulter la Section 11.6).

ASTUCE
Les fichiers PDF peuvent être lus avec GIMP et convertis au format PNG en utilisant une résolution supérieure à 300 dpi. Le fichier produit peut être utilisé comme image d'arrière-plan pour LibreOffice pour produire une présentation agréable avec un minimum d'efforts.

11.4.1 Ghostscript

Le cœur de la manipulation des données imprimables est l'interpréteur Ghostscript PostScript (PS) qui génère une image tramée (« raster image »).

paquet	popularité	taille	description
`ghostscript`	V:180, I:554	177	L'interpréteur GPLGhostscript PostScript/PDF
`ghostscript-x`	V:0, I:12	88	interpréteur GPL Ghostscript PostScript/PDF - prise en charge de l'affichage X
`libpoppler162`	V:16, I:24	5321	bibliothèque de rendu PDF dérivée du visualisateur PDF xpdf
`libpoppler-glib8t64`	V:80, I:335	612	bibliothèque de rendu PDF (bibliothèque partagée basée sur GLib)
`poppler-data`	V:170, I:578	13086	CMaps pour PDF la bibliothèque de rendu (pour la prise en charge de CJK : Adobe-*)

Table 11.14 – Liste des interpréteurs Ghostscript PostScript

ASTUCE
« `gs -h` » permet d'afficher la configuration de Ghostscript.

11.4.2 Fusionner deux fichiers PS ou PDF

Vous pouvez fusionner deux fichiers PostScript (PS) ou Portable Document Format (PDF) en utilisant gs(1) de Ghostscript.

```
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pswrite -sOutputFile=bla.ps -f foo1.ps foo2.ps
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile=bla.pdf -f foo1.pdf foo2.pdf
```

Note
Le format PDF qui est un format imprimable multi-plateformes largement utilisé, est essentiellement un format PS compressé avec quelques fonctionnalités et extensions supplémentaires .

ASTUCE
En ligne de commande, psmerge(1) et d'autres commandes du paquet `psutils` sont utiles pour manipuler des documents PostScript. pdftk(1) issu du paquet `pdftk-java` est aussi utile pour la manipulation de documents PDF.

11.4.3 Utilitaires pour les données imprimables

Les paquets suivants fournissant des utilitaires pour les données imprimables ont attiré mon attention :

paquet	popularité	taille	mot clé	description
poppler-utils	V:175, I:475	804	pdf → ps, text, …	utilitaires PDF : pdftops, pdfinfo, pdfimages, pdftotext, pdffonts
psutils	V:4, I:47	34	ps → ps	outils de conversion de document PostScript
poster	V:0.1, I:1.5	57	ps → ps	créer des posters de grande dimension à partir de pages PostScript
enscript	V:1, I:12	2138	text → ps, html, rtf	convertir du text ASCII en PostScript, HTML, RTF ou Pretty-Print
a2ps	V:0.6, I:6.5	4109	text → ps	Convertisseur de « N'importe quoi vers PostScript » et imprimeur élégant
pdftk	V:1, I:18	28	pdf → pdf	outil de conversion de documents PDF : pdftk
html2ps	V:0.1, I:1.7	256	html → ps	convertisseur de HTML vers PostScript
gnuhtml2latex	V:0.04, I:0.55	26	html → latex	convertisseur de html vers latex
latex2rtf	V:0.1, I:1.9	495	latex → rtf	convertir des documents de LaTeX en RTF qui peuvent être lus par MS Word
ps2eps	V:2, I:30	95	ps → eps	convertisseur de PostScript vers EPS (PostScript encapsulé)
e2ps	V:0.01, I:0.10	104	text → ps	convertisseur de texte vers PostScript avec la prise en charge du codage japonais
impose+	V:0.1, I:1.5	118	ps → ps	Utilitaires PostScript
trueprint	I:0.07	148	text → ps	imprime élégamment de nombreux codes source (C, C++, Java, Pascal, Perl, Pike, Sh, et Verilog) vers PostScript. (langage C)
pdf2svg	V:0.2, I:2.8	33	pdf → svg	convertisseur de PDF vers le format Scalable vector graphics (« SVG »)
pdftoipe	V:0.02, I:0.44	73	pdf → ipe	convertisseur de PDF vers le format IPE d'XML

Table 11.15 – Liste des utilitaires pour les données imprimables

11.4.4 Imprimer avec CUPS

Les commandes lp(1) et lpr(1) proposées par le Common UNIX Printing System (CUPS) fournissent toutes deux des options de personnalisation des données imprimables.

Vous pouvez imprimer 3 copies d'un fichier en utilisant une des commandes suivantes :

```
$ lp -n 3 -o Collate=True filename
```

```
$ lpr -#3 -o Collate=True filename
```

Vous pouvez personnaliser davantage le fonctionnement de l'imprimante en utilisant des options d'impression telles que « `-o number-up=2` », « `-o page-set=even` », « `-o page-set=odd` », « `-o scaling=200` », « `-o natural-sc` etc., documentées sur Impression et options en ligne de commande.

11.5 La conversion de données de courrier électronique

Les paquets suivants, destinés à la conversion de données de courrier électronique, ont attiré mon attention :

paquet	popularité	taille	mot clé	description
`sharutils`	V:2, I:26	1436	mail	shar(1), unshar(1), uuencode(1), uudecode(1)
`mpack`	V:0.7, I:7.1	109	`MIME`	encoder et décoder des messages MIME : mpack(1) et munpack(1)
`tnef`	V:0.3, I:3.7	103	`ms-tnef`	dépaqueter des attachements MIME de type « application/ms-tnef » qui est un format propre à Microsoft
`uudeview`	V:0.2, I:1.6	105	`mail`	encodeur et décodeur pour les formats suivants : uuencode, xxencode, BASE64, quoted printable et BinHex

Table 11.16 – Liste de paquets facilitant la conversion de données de courrier électronique

ASTUCE
Le serveur IMAP4 (Internet Message Access Protocol), version 4, peut être utilisé pour récupérer des courriels depuis des systèmes de courriels propriétaires si le logiciel de courriel client peut aussi être configuré pour utiliser IMAP4.

11.5.1 Bases concernant les données de courrier électronique

Les données (SMTP) de courrier électronique doivent être limitées à des séries de données de 7 bits. Les données binaires et les données textuelles sur 8 bits sont codées dans un format sur 7 bits avec Multipurpose Internet Mail Extensions (MIME) et la sélection du jeu de caractères (consultez le Tableau 11.2).

Le format standard d'enregistrement du courrier électronique est « mbox » selon la RFC2822 (RFC822 mise à jour). Consultez mbox(5) (fourni par le paquet `mutt`.

Pour les langues européennes, on utilise habituellement pour le courriel « `Content-Transfer-Encoding: quoted-pri` avec le jeu de caractères ISO-8859-1 car il n'y a pas beaucoup de caractères de 8 bits. Si le texte européen est codé en UTF-8, on préférera utiliser « `Content-Transfer-Encoding: quoted-printable` » car ce sont essentiellement des données sur 7 bits.

Pour le japonais, « `Content-Type: text/plain; charset=ISO-2022-JP` » est habituellement utilisé pour le courriel afin de conserver le texte sur 7 bits. Mais les anciens systèmes Microsoft peuvent envoyer des données de courriel en Shift-JIS sans le déclarer proprement. Si le texte japonais est codé en UTF-8, on utilisera de préférence Base64 car il comporte de nombreuses données sur 8 bits. La situation des autres langues asiatiques est similaire.

Note
Pour les données de courriel non-UNIX accessibles par un logiciel client ne venant pas de Debian qui est capable de dialoguer avec le serveur IMAP4, vous pourrez peut-être les récupérer en utilisant votre propre serveur IMAP4).

Note
Si vous utilisez d'autres formats d'enregistrement de courriel, les mettre dans le format mbox est une bonne première étape. Un programme client souple comme mutt(1) peut être pratique pour le faire.

Vous pouvez éclater le contenu d'une boîte à lettre en messages séparés en utilisant procmail(1) et formail(1).

Chaque message de courrier électronique peut être dépaqueté en utilisant munpack(1) qui provient du paquet `mpack` (ou d'autres outils spécialisés) afin d'en obtenir le contenu codé en MIME.

11.6 Outils de données graphiques

Bien que des programmes ayant une interface graphique comme gimp(1) soient très puissants, des outils en ligne de commandes comme imagemagick(1) sont assez utiles pour la manipulation automatique d'images au moyen de scripts.

Le format standard de fait pour les images d'appareils photo numériques est Exchangeable Image File Format (EXIF) qui est composé d'une image au format JPEG à laquelle sont ajoutées des balises de métadonnées. Il peut contenir des informations telles que la date, l'heure ou les paramètres de l'appareil photo.

Le brevet de compression de données sans perte Lempel-Ziv-Welch (LZW) est arrivé en fin de validité. Les utilitaires du format Graphics Interchange Format (GIF) qui utilise la méthode de compression LZW peuvent être maintenant librement disponibles sur un système Debian.

ASTUCE
Tous les appareils photo numériques ou les scanners ayant un support d'enregistrement amovible fonctionnent sous Linux avec des lecteurs « USB storage » s'ils sont conformes à la Design rule for Camera Filesystem et utilisent un système de fichiers FAT. Consultez Section 10.1.7.

11.6.1 Outils de données graphiques (métapaquet)

Les métapaquets suivants sont des bons points de départ pour rechercher des outils pour des données graphiques en utilisant aptitude(8). « Packages overview for Debian PhotoTools Maintainers » peut être un autre point de départ.

paquet	popularité	taille	mot clé	description
`education-graphics`	I:0.30	26	`svg,jpeg,…`	métapaquet pour l'apprentissage des arts graphiques et picturaux
`open-font-design-toolkit`	I:0.04	9	`ttf,ps,…`	métapaquet pour la conception de fontes libres

Table 11.17 – Liste d'outils pour les données graphiques (métapaquet)

ASTUCE
Recherchez d'autres outils pour les images en utilisant l'expression rationnelle « `~Gworks-with::image` » dans aptitude(8) (consultez Section 2.2.6).

11.6.2 Outils de données graphiques (interface graphique)

Les paquets suivants contenant des outils graphiques pour la conversion, l'édition et l'organisation de données graphiques ont attiré mon attention.

paquet	popularité	taille	mot clé	description
gimp	V:40, I:206	33386	image(bitmap)	programme GNU de manipulation d'images (« GNU Image Manipulation Program »)
xsane	V:9, I:127	1512	image(bitmap)	interface graphique X11 de SANE (Scanner Access Now Easy) basée sur GTK
scribus	V:1, I:12	32942	ps/pdf/SVG/...	éditeur DTP Scribus
libreoffice-draw	V:78, I:412	11424	image(vector)	suite de bureautique LibreOffice - dessin
inkscape	V:12, I:73	113357	image(vector)	éditeur SVG (Scalable Vector Graphics)
dia	V:1, I:16	3846	image(vector)	éditeur de diagrammes (Gtk)
xfig	V:0.5, I:8.4	7951	image(vector)	outil de génération interactive de figures sous X11
gocr	V:0.4, I:3.4	549	image→text	logiciel OCR libre
eog	V:20, I:121	10542	image(Exif)	visionneur d'images « Eye of Gnome » (l'œil de Gnome)
gthumb	V:3, I:11	5131	image(Exif)	visionneur et gestionnaire de photos (GNOME)
geeqie	V:3, I:10	3854	image(Exif)	visionneur d'images utilisant GTK
shotwell	V:15, I:244	6334	image(Exif)	gestionnaire de photos numériques (GNOME)
gwenview	V:40, I:113	6051	image(Exif)	visionneur d'images (KDE)
kamera	I:112	992	image(Exif)	gestion des appareils photo numériques dans les applications KDE
digikam	V:1.5, I:8.2	333	image(Exif)	application de gestion de photos numériques pour KDE
darktable	V:3, I:10	36092	image(Exif)	table lumineuse et chambre noire virtuelles pour photographes
hugin	V:0.5, I:5.2	6885	image(Exif)	assembleur de photos pour panorama
librecad	V:1, I:13	12361	DXF,…	éditeur de données 2D pour la CAD
freecad	V:1, I:20	112	DXF,…	éditeur de données 3D pour la CAD
blender	V:3, I:20	110904	blend,TIFF,VRML,…	éditeur de contenu 3D pour l'animation, etc.
mm3d	V:0.03, I:0.23	4123	ms3d,obj,dxf,…	éditeur de modèles 3D basé sur OpenGL
fontforge	V:0.5, I:5.2	4736	ttf,ps,…	éditeur de fontes pour les fontes PS, TrueType et OpenType
xgridfit	V:0.02, I:0.07	878	ttf	programme pour l'ajustement à la grille et l'optimisation de rendu des fontes TrueType

Table 11.18 – Liste d'outils pour les données graphiques (interfaces graphiques)

11.6.3 Outils de données graphiques (ligne de commande)

Les paquets suivants contenant des outils en ligne de commande pour la conversion, l'édition et l'organisation de données graphiques ont attiré mon attention.

11.7 Diverses conversions de données

Il y a de nombreux programmes pour convertir les données. Les paquets suivants ont attiré mon attention en utilisant l'expression rationnelle « ~Guse::converting » avec aptitude(8) (consultez Section 2.2.6).

Vous pouvez aussi extraire des données depuis le format RPM avec ce qui suit :

paquet	popularité	taille	mot clé	description
imagemagick	V:7, I:266	82	image(bitmap)	programmes de manipulation d'images
graphicsmagick	V:1.0, I:7.8	5988	image(bitmap)	programmes de manipulation d'images (dérivés d'imagemagick)
netpbm	V:27, I:275	8470	image(bitmap)	outils de conversion graphique
libheif-examples	V:0.4, I:3.7	534	heif→jpeg(bitmap)	conversion de High Efficiency Image File Format (HEIF) aux formats JPEG, PNG ou Y4M avec la commande heif-convert(1)
icoutils	V:3, I:31	221	png↔ico(bitmap)	conversion des icônes et curseurs de MS Windows de et vers des formats PNG (favicon.ico)
pstoedit	V:2, I:36	1075	ps/pdf→image(vector)	convertisseur de fichiers PostScript et PDF en graphiques vectoriels éditables (SVG)
libwmf-bin	V:5, I:77	149	Windows/image (vectorielle)	outils de conversion de metafile de Windows (données graphiques vectorielles)
fig2sxd	V:0.02, I:0.14	158	fig→sxd(vector)	conversion de fichiers XFig dans le format OpenOffice.org Draw
unpaper	V:2, I:16	417	image→image	outil de post-traitement pour des pages numérisées pour OCR
tesseract-ocr	V:9, I:35	2411	image→text	logiciel OCR libre basé sur le moteur commercial OCR de HP
tesseract-ocr-eng	V:9, I:35	4032	image→text	moteur de données OCR : fichier de langue de tesseract-ocr pour le texte en anglais
ocrad	V:0.3, I:2.2	608	image→text	logiciel OCR libre
exif	V:2, I:50	334	image(Exif)	utilitaire en ligne de commandes pour afficher les informations EXIF contenues dans les fichiers JPEG
exiv2	V:1, I:18	436	image(Exif)	outil pour manipuler les méta-données EXIF/IPTC
exiftran	V:1, I:11	81	image(Exif)	transformer les images JPEG des appareils photo numériques
exiftags	V:0.2, I:2.5	309	image(Exif)	utilitaire pour lire les balises Exif depuis un fichier JPEG d'appareil photo numérique
exifprobe	V:0.2, I:2.0	506	image(Exif)	lire les métadonnées des images numériques
dcraw	V:0.7, I:7.0	428	image(Raw)→ppm	décoder les images brutes (« raw ») des appareils photo numériques
findimagedupes	V:0.10, I:0.94	76	image→fingerprint	rechercher des images visuellement similaires ou dupliquées
ale	V:0.01, I:0.14	850	image→image	assembler des images pour en améliorer la fidélité ou créer des mosaïques
imageindex	V:0.1, I:1.1	143	image(Exif)→html	créer des galeries HTML statiques depuis des images
outguess	V:0.1, I:1.0	230	jpeg,png	outil universel Stéganographique
jpegoptim	V:0.5, I:5.6	59	jpeg	optimisation de fichiers JPEG
optipng	V:2, I:38	187	png	optimisation de fichiers PNG, compression sans perte
pngquant	V:1, I:11	62	png	optimisation de fichiers PNG, compression avec pertes

Table 11.19 – Liste d'outils pour les données graphiques (ligne de commande)

paquet	popularité	taille	mot clé	description
alien	V:1, I:12	150	rpm/tgz → deb	convertisseur de paquets étrangers vers des paquets Debian
freepwing	I:0.01	447	EB → EPWING	convertisseur de « Electric Book » (Livre électronique - populaire au Japon) en un simple format JIS X 4081 (un sous-ensemble de EPWING V1)
calibre	V:7, I:24	76289	any → EPUB	convertisseur de livre numérique et gestion de bibliothèque

Table 11.20 – Liste d'outils divers de conversion de données

```
$ rpm2cpio file.src.rpm | cpio --extract
```

Chapitre 12

Programmation

Je donne quelques indications pour apprendre à programmer sous le système Debian, suffisantes pour suivre le code source mis en paquets. Voici les paquets importants correspondant aux paquets de documentation pour la programmation .

Une référence en ligne est accessible en entrant « `man name` » après l'installation des paquets `manpages` et `manpages-dev`. Les références en ligne des outils GNU tools sont disponibles en entrant « `info nom_programme` » après l'installation des paquets de documentation pertinents. Vous devrez peut-être inclure les archives `contrib` et `non-free` en plus de l'archive `main` car certaines documentations GFDL ne sont pas considérées comme conformes à DFSG.

Veuillez envisager d'utiliser des outils ds système de gestion de versions. Consulter la Section 10.5.



AVERTISSEMENT
N'utilisez pas « `test` » comme nom d'un fichier exécutable. « `test` » fait partie de l'interpréteur de commandes.



Attention
Vous devrez installer les programmes directement compilés à partir des sources dans « `/usr/local` » ou « `/opt` » afin d'éviter des collisions avec les programmes du système.

12.1 Les scripts de l'interpréteur de commande

Le script de l'interpréteur de commandes (« shell script » est un fichier texte dont le bit d'exécution est positionné et qui contient des commandes dans le format suivant :

```
#!/bin/sh
 ... command lines
```

La première ligne indique l'interpréteur qui sera utilisé pour lire et exécuter le contenu de ce fichier.

La lecture des scripts de l'interpréteur de commandes est la **meilleure** manière de comprendre comment fonctionne un système de type UNIX. Je donne ici quelques indications et rappels de la programmation avec l'interpréteur de commandes. Consultez « Erreurs en shell » (https://www.greenend.org.uk/rjk/2001/04/shell.html) pour apprendre à partir d'erreurs.

Contrairement à l'interpréteur de commandes en mode interactif (consultez Section 1.5 et Section 1.6), les scripts de l'interpréteur de commandes utilisent souvent des paramètres, des conditions et des boucles.

12.1.1 Compatibilité de l'interpréteur de commandes avec POSIX

De nombreux scripts système peuvent être interprétés par l'un des interpréteurs de commandes POSIX (consulter le Tableau 1.13).

— L'interpréteur de commandes POSIX par défaut non interactif « `/usr/bin/sh` » est un lien symbolique pointant vers `/usr/bin/dash` et est utilisé par de nombreux programmes du système.
— L'interpréteur POSIX interactif par défaut est `/usr/bin/bash`.

Évitez d'écrire des scripts de l'interpréteur de commandes avec des **bashismes** ou des **zshismes** afin de les rendre portables entre tous les interpréteurs POSIX. Vous pouvez le vérifier en utilisant checkbashisms(1).

Bon : POSIX	**À éviter : bashisme**
`if [ "$foo" = "$bar" ] ; then …`	`if [ "$foo" == "$bar" ] ; then …`
`diff -u file.c.orig file.c`	`diff -u file.c{.orig,}`
`mkdir /foobar /foobaz`	`mkdir /foo{bar,baz}`
`funcname() { … }`	`function funcname() { … }`
format octal : « `\377` »	format hexadécimal : « `\xff` »

Table 12.1 – Liste de bashismes typiques

La commande « `echo` » doit être utilisée avec les précautions suivantes car son implémentation diffère selon que l'on utilise les commandes internes ou externes de l'interpréteur de commandes :

— Éviter d'utiliser toutes les options de commandes sauf « `-n` ».
— Éviter d'utiliser les séquences d'échappement dans les chaînes de caractères car leur prise en compte varie.

Note
Bien que l'option « `-n` » ne soit **pas** vraiment de la syntaxe POSIX, elle est généralement acceptée.

ASTUCE
Utilisez la commande « `printf` » plutôt que la commande « `echo` » si vous avez besoin d'intégrer des séquences d'échappement dans la chaîne de sortie.

12.1.2 Paramètres de l'interpréteur de commandes

Des paramètres spéciaux de l'interpréteur de commandes sont souvent utilisés dans les scripts de l'interpréteur de commandes.

paramètre de l'interpréteur de commandes	**valeur**
`$0`	nom de l'interpréteur ou du script de l'interpréteur
`$1`	premier (1er) paramètre de l'interpréteur
`$9`	neuvième (9ème) paramètre de l'interpréteur
`$#`	nombres de paramètres positionnels
`"$*"`	`"$1 $2 $3 $4 … "`
`"$@"`	`"$1" "$2" "$3" "$4" …`
`$?`	valeur de retour de la commande la plus récente
`$$`	PID de ce script de l'interpréteur de commandes
`$!`	PID de la tâche de fond la plus récemment lancée

Table 12.2 – Liste des paramètres de l'interpréteur de commandes

forme de l'expression du paramètre	valeur si `var` est positionnée	valeur si `var` n'est pas positionnée
`${var:-string}`	"`$var`"	"`string`"
`${var:+string}`	"`string`"	"`null`"
`${var:=string}`	"`$var`"	« `chaîne` » (et lancer « `var=chaîne` »)
`${var:?string}`	"`$var`"	echo « `chaîne` » vers **stderr** (et quitter avec une erreur)

Table 12.3 – Liste des expansions de paramètre de l'interpréteur

Les **expansions de paramètre** les plus courantes à retenir sont mentionnées ci-dessous :

Ici, les deux points « `:` » dans tous ces opérateurs sont en fait optionnels.
— **avec** « `:` » = opérateur de test pour **existe** et **différent de null**
— **sans** « `:` » = opérateur de test pour **existe** uniquement

forme de substitution de paramètre	résultat
`${var%suffix}`	supprimer le motif le plus petit de la partie finale
`${var%%suffix}`	supprimer le motif le plus grand de la partie finale
`${var#prefix}`	supprimer le motif le plus petit de la partie initiale
`${var##prefix}`	supprimer le motif le plus grand de la partie initiale

Table 12.4 – Liste des substitutions-clés de paramètres de l'interpréteur

12.1.3 Opérateurs conditionnels de l'interpréteur

Chaque commande retourne un **état de sortie** qui peut être utilisé pour des expressions conditionnelles.
— Succès : 0 (« Vrai »)
— Erreur : différent de 0 (« Faux »)

Note
« 0 » dans le contexte conditionnel de l'interpréteur signifie « Vrai » alors que « 0 » dans le contexte conditionnel de C signifie « Faux ».

Note
« `[` » est l'équivalent de la commande `test`, qui évalue, comme expression conditionnelle, les paramètres jusqu'à « `]` ».

Les **idiomes conditionnels** de base à retenir sont les suivants :
— « *`commande`* `&&` *`si_succès_lancer_aussi_cette_commande`* `||` `true` »
— « *`commande`* `||` *`en_cas_de_non_succès_lancer_aussi_cette_commande`* `||` `true` »
— Un morceau de script sur plusieurs lignes comme le suivant :

```
if [ conditional_expression ]; then
 if_success_run_this_command
else
 if_not_success_run_this_command
fi
```

Ici, le « `|| true` » était nécessaire pour s'assurer que ce script de l'interpréteur ne se termine pas accidentellement à cette ligne lorsque l'interpréteur est appelé avec l'indicateur « `-e` ».

Les opérateurs de comparaison **arithmétique** entière dans les expressions conditionnelles sont « `-eq` », « `-ne` », « `-lt` », « `-le` », « `-gt` » et « `-ge` ».

équation	condition pour retourner une valeur logique vraie
`-e` *file*	*fichier* existe
`-d` *file*	*fichier* existe et est un répertoire
`-f` *file*	*fichier* existe et est un fichier normal (« régulier »)
`-w` *file*	*fichier* existe et peut être écrit
`-x` *file*	*fichier* existe et est exécutable
file1 `-nt` *file2*	*fichier1* est plus récent que *fichier2* (modification)
file1 `-ot` *file2*	*fichier1* est plus ancien que *fichier2* (modification)
file1 `-ef` *file2*	*fichier1* et *fichier2* sont sur le même périphérique et le même numéro d'inœud

Table 12.5 – Liste des opérateurs de comparaison dans les expressions conditionnelles

équation	condition pour retourner une valeur logique vraie
`-z` *str*	la longueur de *str* est nulle
`-n` *str*	la longueur de *str* est non nulle
str1 `=` *str2*	*str1* et *str2* sont égales
str1 `!=` *str2*	*str1* et *str2* ne sont pas égales
str1 `<` *str2*	*str1* est trié avant *str2* (dépendant des paramètres linguistiques)
str1 `>` *str2*	*str1* est trié après *str2* (dépendant des paramètres linguistiques)

Table 12.6 – Liste des opérateurs de comparaison de chaîne de caractères dans les expressions conditionnelles

12.1.4 Boucles de l'interpréteur de commandes

Il existe un certains nombre d'idiomes de boucles qu'on peut utiliser avec un interpréteur de commandes POSIX.

— « `for x in toto1 toto2 ... ; do commande ; done` » boucle en assignant les éléments de la liste « `toto1 toto2 ...` » à la variable « `x` » et en exécutant la « `commande` ».
— « "`while condition ; do commande ; done` » répète la « `commande` » tant que la « `condition` » est vraie.
— « `until condition ; do commande ; done` » répète la « `commande` » tant que « « `condition` » n'est pas vraie.
— « `break` » permet de quitter la boucle.
— « `continue` » permet de reprendre l'itération suivante de la boucle.

ASTUCE
Les itérations numériques semblables à celles du langage C peuvent être réalisées en utilisant seq(1) comme générateur de « `toto1 toto2 …` ».

ASTUCE
Consultez Section 9.4.9.

12.1.5 Variables d'environnement de l'interpréteur de commandes

Certaines variables d'environnement populaires pour l'invite de commande normale de l'interpréteur peuvent ne pas être disponibles dans l'environnement d'exécution de votre script.

— Pour « `$USER` », utilisez « `$(id -un)` »
— Pour « `$UID` », utilisez « `$(id -u)` »
— Pour « `$HOME` », utilisez « `$(getent passwd "$(id -u)"|cut -d ":" -f 6)` » (cela fonctionne aussi dans la Section 4.5.2)

12.1.6 Séquence de traitement de la ligne de commandes de l'interpréteur

En gros, l'interpréteur de commandes traite un script de la manière suivante :

— l'interpréteur de commandes lit une ligne :
— l'interpréteur de commandes regroupe une partie de la ligne sous forme d'**un élément** (« token » si elle se trouve entre "…" ou '…' :
— l'interpréteur de commandes découpe les autres parties de la ligne en **éléments** comme suit :
  — Espaces : *espace tabulation saut-de-ligne*
  — métacaractères : `< > | ; & ( )`
— l'interpréteur de commandes vérifie les **mots réservés** pour chacun des éléments et ajuste son comportement s'il ne se trouve pas entre "…" ou '…'.
  — **mot réservé** : `if then elif else fi for in while unless do done case esac`
— L'interpréteur de commandes étend les **alias** s'ils ne se trouvent pas entre "…" ou '…'.
— l'interpréteur de commandes étend les **tilde** s'ils ne se trouvent pas entre "…" ou '…'.
  — « `~` » → répertoire personnel de l'utilisateur actuel
  — « *~utilisateur* » → répertoire personnel de l'*utilisateur*
— l'interpréteur de commandes étend les **paramètres** en leur valeur s'ils ne sont pas entre '…'.
  — **paramètre** : « $PARAMETRE » ou « ${PARAMETRE} »
— l'interpréteur de commandes étend la **substitution de commande** si elle n'est pas entre '…'.
  — « `$( commande )` » → sortie de la « `commande` »
  — « `` ` commande ` `` » → sortie de la « `commande` »
— l'interpréteur de commandes étend les **motifs génériques du chemin** aux fichiers correspondants s'ils ne sont pas entre "…" ou '…'.
  — * → n'importe quel caractère
  — ? → un caractère
  — […] → un caractère quelconque parmi « … »
— l'interpréteur de commandes recherche la **commande** dans ce qui suit et l'exécute.
  — définition de **fonction**
  — commande **interne** (« builtin »)
  — **fichier exécutable** dans « $PATH »
— l'interpréteur de commandes passe à la ligne suivante et recommence ce traitement depuis le début de la séquence.

Des guillemets simples dans des guillemets doubles n'ont pas d'effet.

Exécuter « `set -x` » dans le script de l'interpréteur ou l'appel du script avec l'option « `-x` » fait imprimer par l'interpréteur de commandes toutes les commandes exécutées. C'est assez pratique pour le débogage.

12.1.7 Programmes utilitaires pour les scripts de l'interpréteur de commandes

De façon à rendre vos programmes de l'interpréteur de commandes aussi portables que possible dans tous les systèmes Debian, c'est une bonne idée de limiter les programmes utilitaires à ceux fournis par les paquets **essentiels**.

— « `aptitude search ~E` » affiche la liste des **essentiels**.
— « `dpkg -L` *nom_paquet* `|grep '/man/man.*/'` » affiche la liste des pages de manuel pour les commandes que fournit le paquet *nom_paquet*.

ASTUCE
Bien que `moreutils` puisse ne pas exister en dehors de Debian, il propose d'intéressants petits programmes. Le plus remarquable est sponge(8) qui est bien pratique pour écraser le fichier d'origine.

Consulter la Section 1.6 pour des exemples.

paquet	popularité	taille	description
dash	V:933, I:999	211	petit et rapide interpréteur de commandes compatible POSIX pour sh
coreutils	V:914, I:1000	19751	utilitaires du cœur de GNU
grep	V:800, I:1000	1297	GNU grep, egrep et fgrep
sed	V:789, I:1000	987	GNU sed
mawk	V:517, I:998	299	petit et rapide awk
debianutils	V:940, I:997	228	divers utilitaires spécifiques à Debian
bsdutils	V:445, I:1000	338	utilitaires de base provenant de 4.4BSD-Lite
bsdextrautils	V:771, I:877	382	utilitaires supplémentaires de 4.4BSD-Lite
moreutils	V:18, I:38	232	utilitaires supplémentaires d'UNIX

Table 12.7 – Lites des paquets comportant des petits programmes utilitaires pour les scripts de l'interpréteur de commandes

12.2 Scriptage avec des langages interprétés

paquet	popularité	taille	documentation
dash	V:933, I:999	211	sh : petit et rapide interpréteur compatible POSIX pour sh
bash	V:900, I:1000	7309	sh : « info bash » fourni par bash-doc
mawk	V:517, I:998	299	AWK : petit et rapide awk
gawk	V:246, I:288	3289	AWK : « info gawk » fourni par gawk-doc
perl	V:672, I:991	840	Perl : perl(1) et pages HTML fournies par perl-doc et perl-doc-html
libterm-readline-gnu-perl	V:2, I:25	435	Extension Perl pour la bibliothèque GNU ReadLine/History : perlsh(1)
libreply-perl	I:0.08	171	REPL pour Perl : reply(1)
libdevel-repl-perl	V:0.02, I:0.47	237	REPL pour Perl : re.pl(1)
python3	V:748, I:977	81	Python : python3(1) et pages HTML fournies par python3-doc
tcl	V:24, I:171	20	Tcl : tcl(3) et les pages de manuel détaillées fournies par tcl-doc
tk	V:18, I:165	20	Tk : tk(3) et pages de manuel détaillées fournies par tk-doc
ruby	V:61, I:150	32	Ruby : ruby(1), erb(1), irb(1), rdoc(1), ri(1)

Table 12.8 – Liste des paquets relatifs aux interpréteurs de commandes

Lorsque vous souhaitez automatiser une tâche sur Debian, vous devez d'abord la scripter avec un langage interprété. La ligne directrice pour le choix de la langue interprétée est la suivante :

— utilisez dash, si c'est une tâche simple qui combine des programmes en ligne de commandes avec un interpréteur ;
— utilisez python3, si la tâche n'est pas simple et que vous l'écrivez à partir de zéro ;
— utilisez perl, tcl, ruby..., s'il existe un code existant utilisant l'un de ces langages sur Debian et qui doit être retouché pour effectuer cette tâche.

Si le code résultant est trop lent, vous pouvez réécrire uniquement la partie critique pour la vitesse d'exécution dans un langage compilé et l'appeler depuis le langage interprété.

12.2.1 Débogage du code d'un langage interprété

La plupart des interpréteurs offrent des fonctionnalités de base de vérification de la syntaxe et de traçage du code :
— « **dash -n** *script.sh* » – vérification de la syntaxe d'un script d'interpréteur ;
— « **dash -x** *script.sh* » – traçage d'un script d'interpréteur ;
— « **python3 -m py_compile** *script.py* » – contrôle syntaxique d'un script Python ;
— « **python3 -mtrace --trace** *script.py* » – traçage d'un script Python ;
— « **perl -I .. /libpath -c** *script.pl* » – vérification de la syntaxe d'un script Perl ;
— « **perl -d:Trace** *script.pl* » – traçage d'un script Perl.
Pour tester le code de `dash`, essayez la Section 9.1.4 qui s'adapte à un environnement interactif de type `bash`.

Pour tester le code de `perl`, essayez l'environnement REPL pour Perl qui s'adapte au REPL (=READ + EVAL + PRINT + LOOP) de type Python pour Perl.

12.2.2 Programmes graphiques avec des scripts de d'interpréteur de commandes

Le script d'interpréteur peut être amélioré pour créer un programme graphique attrayant. L'astuce consiste à utiliser l'un des programmes dits de dialogue au lieu d'une interaction insipide utilisant les commandes `echo` et `read`.

paquet	popularité	taille	description
`x11-utils`	V:233, I:553	651	xmessage(1) : afficher un message ou une question dans une fenêtre (X)
`whiptail`	V:344, I:997	61	afficher des boîtes de dialogues conviviales depuis des scripts de l'interpréteur de commandes (newt)
`dialog`	V:8, I:75	535	afficher des boîtes de dialogues conviviales depuis des scripts de l'interpréteur de commandes (ncurses)
`zenity`	V:69, I:337	193	afficher des boîtes de dialogue graphiques à partir de scripts d'interpréteur (GTK)
`ssft`	V:0.01, I:0.14	80	Outil frontal de scripts de l'interpréteur de commandes (enrobeur pour zenity, kdialog et dialog avec gettext)
`gettext`	V:49, I:206	20549	« `/usr/bin/gettext.sh` » : traduire des messages

Table 12.9 – Liste des programmes de dialogue

Voici un exemple de programme graphique pour démontrer à quel point c'est facile avec un script d'interpréteur.

Ce script utilise `zenity` pour sélectionner un fichier (par défaut `/etc/motd`) et l'afficher.

Le lanceur graphique pour ce script peut être créé en suivant la Section 9.4.10.

```
#!/bin/sh -e
# Copyright (C) 2021 Osamu Aoki <osamu@debian.org>, Public Domain
# vim:set sw=2 sts=2 et:
DATA_FILE=$(zenity --file-selection --filename="/etc/motd" --title="Select a file to check ↩
    ") || \
  ( echo "E: File selection error" >&2 ; exit 1 )
# Check size of archive
if ( file -ib "$DATA_FILE" | grep -qe '^text/' ) ; then
  zenity --info --title="Check file: $DATA_FILE" --width 640  --height 400 \
    --text="$(head -n 20 "$DATA_FILE")"
else
  zenity --info --title="Check file: $DATA_FILE" --width 640  --height 400 \
    --text="The data is MIME=$(file -ib "$DATA_FILE")"
fi
```

Ce type d'approche du programme graphique avec un script d'interpréteur n'est utile que dans le cas de choix simple. Si vous devez écrire un programme avec des complexités, veuillez envisager de l'écrire à partir d'une plateforme plus adaptée.

12.2.3 Actions personnalisées pour le gestionnaire de fichiers graphique

Les programmes graphiques de gestionnaire de fichiers peuvent être étendus pour effectuer certaines actions fréquentes sur les fichiers sélectionnés à l'aide de paquets d'extension supplémentaires. Ils peuvent également effectuer des actions personnalisées très spécifiques en ajoutant vos scripts personnels.

— Pour GNOME, consulter NautilusScriptsHowto.
— Pour KDE, consulter Création de menus de service pour Dolphin.
— Pour Xfce, consulter Thunar – Actions personnalisées et https://help.ubuntu.com/community/ThunarCustomActions.
— Pour LXDE, consulter Actions personnalisées.

12.2.4 Extravagances des scripts courts en Perl

Pour traiter les données, `sh` doit générer un sous-processus exécutant `cut`, `grep`, `sed`, etc., et est lent. D'autre part, `perl` a des capacités internes pour traiter les données et est rapide. Un certain nombre de scripts d'entretien du système Debian utilisent `perl`.

Considérons un bout de code uniligne en AWK et ses équivalents en Perl :

```
awk '($2=="1957") { print $3 }' |
```

Il est équivalent à l'une quelconque des lignes suivantes :

```
perl -ne '@f=split; if ($f[1] eq "1957") { print "$f[2]\n"}' |
```

```
perl -ne 'if ((@f=split)[1] eq "1957") { print "$f[2]\n"}' |
```

```
perl -ne '@f=split; print $f[2] if ( $f[1]==1957 )' |
```

```
perl -lane 'print $F[2] if $F[1] eq "1957"' |
```

```
perl -lane 'print$F[2]if$F[1]eq+1957' |
```

La dernière est une devinette. Elle tire parti des fonctionnalités suivantes de Perl :

— L'espace est optionnel.
— Il existe une conversion automatique des nombres en chaîne de caractères.
— Astuces d'exécution de Perl à travers des options de la ligne de commande : perlrun(1)
— Variables spéciales Perl : perlvar(1)

Cette flexibilité est la force de Perl. En même temps, cela nous permet de créer des codes énigmatiques et fouillis. Soyez donc prudent.

12.3 Codage dans les langages compilés

Ici, la Section 12.3.3 et la Section 12.3.4 sont incluses pour indiquer comment un programme de type compilateur peut être écrit en langage C en compilant une description de niveau supérieur en langage C.

paquet	popularité	taille	description
gcc	V:167, I:557	36	compilateur GNU C
libc6-dev	V:259, I:577	12863	bibliothèque GNU C : bibliothèques de développement et fichiers d'en-tête
g++	V:64, I:527	13	compilateur GNU C++
libstdc++-16-dev	V:4, I:21	24416	bibliothèque GNU C++ standard v3 – fichiers de développement
cpp	V:346, I:717	18	préprocesseur GNU C
gettext	V:49, I:206	20549	utilitaires d'internationalisation GNU
glade	V:0.2, I:2.5	1613	constructeur d'interface utilisateur en GTK
valac	V:0.2, I:2.9	533	langage de type C# pour le système GObject
flex	V:7, I:66	1247	générateur d'analyse lexicale rapide compatible avec LEX
bison	V:7, I:70	3122	générateur d'analyseurs syntaxiques compatible avec YACC
susv2	I:0.02	16	aller chercher « The Single UNIX Specifications v2 »
susv3	I:0.04	16	aller chercher « The Single UNIX Specifications v3 »
susv4	I:0.03	16	aller chercher « The Single UNIX Specifications v4 »
golang	I:21	11	compilateur du langage de programmation Go
rustc	V:6, I:20	12745	langage Rust de programmation système
gfortran	V:4, I:47	15	compilateur GNU Fortran 95
fpc	I:2.3	101	Free Pascal

Table 12.10 – Liste des paquets relatifs à un compilateur

12.3.1 C

Vous pouvez définir un environnement propre pour compiler des programmes écrits dans le langage de programmation C par ce qui suit :

```
# apt-get install glibc-doc manpages-dev libc6-dev gcc build-essential
```

Le paquet `libc6-dev`, c'est-à-dire la bibliothèque GNU C, fournit la bibliothèque C standard qui est une collection de fichiers d'en-têtes et de routines de bibliothèque utilisée par le langage de programmation C.

Consultez les références pour C comme suit :

— « `info libc` » (références des fonctions de la bibliothèque C)
— gcc(1) et « `info gcc` »
— each_C_library_function_name(3)
— Kernighan & Ritchie, « Le langage de programmation C », 2ème édition (Prentice Hall)

12.3.2 Programme simple en C (gcc)

Un exemple simple « `example.c` » peut être compilé avec la bibliothèque « `libm` » pour donner l'exécutable « `run_exampl` par ce qui suit :

```
$ cat > example.c << EOF
#include <stdio.h>
#include <math.h>
#include <string.h>

int main(int argc, char **argv, char **envp){
        double x;
        char y[11];
        x=sqrt(argc+7.5);
        strncpy(y, argv[0], 10); /* prevent buffer overflow */
        y[10] = '\0'; /* fill to make sure string ends with '\0' */
        printf("%5i, %5.3f, %10s, %10s\n", argc, x, y, argv[1]);
```

```
        return 0;
}
EOF
$ gcc -Wall -g -o run_example example.c -lm
$ ./run_example
        1, 2.915, ./run_exam,     (null)
$ ./run_example 1234567890qwerty
        2, 3.082, ./run_exam, 1234567890qwerty
```

Ici, « `-lm` » est nécessaire pour lier la bibliothèque « `/usr/lib/libm.so` » depuis le paquet `libc6` pour sqrt(3). La bibliothèque réelle se trouve dans « `/lib/` » avec le nom de fichier « `libm.so.6` » avec un lien symbolique vers « `libm-2.7.so` ».

Regardez le dernier paramètre du texte en sortie. Il y a plus de 10 caractères bien que « `%10s` » soit indiqué.

L'utilisation de fonctions effectuant des opérations sur des pointeurs en mémoire sans vérification des limites, telles que sprintf(3) et strcpy(3) a été rendue obsolète afin d'éviter les exploits de débordements de tampons qui utilisent les effets des débordements ci-dessus. Utilisez snprintf(3) et strncpy(3) en remplacement..

12.3.3 Flex -- un meilleur Lex

Flex est un générateur d'analyse lexicale rapide compatible avec Lex.

On trouve un didacticiel de flex(1) dans « `info flex` ».

Plusieurs exemples simples sont disponibles dans « `/usr/share/doc/flex/examples/` ».[1]

12.3.4 Bison -- un meilleur Yacc

Un certain nombre de paquets fournissent un analyseur LR à lecture anticipée (« lookahead ») compatible avec Yacc ou un générateur d'analyseur LALR sous Debian.

paquet	popularité	taille	description
`bison`	V:7, I:70	3122	générateur d'analyseur GNU LALR
`byacc`	V:0.1, I:2.6	263	générateur d'analyseur Berkeley LALR
`btyacc`	V:0.01, I:0.04	247	générateur d'analyseur avec retour arrière basé sur `byacc`

Table 12.11 – Liste de générateurs d'analyseur LALR compatible avec Yacc

On trouve un didacticiel de bison(1) dans « `info bison` ».

Vous devez fournir vos propre « `main()` » et « `yyerror()` ». « `main()` » appelle « `yyparse()` » qui appelle « `yylex()` » habituellement créé avec Flex.

Voici un exemple de création de programme de calcul dans un terminal.

Créons `example.y` :

```
/* calculator source for bison */
%{
#include <stdio.h>
extern int yylex(void);
extern int yyerror(char *);
%}

/* declare tokens */
```

1. Quelques ajustements peuvent être nécessaires pour qu'ils fonctionnent sur le système actuel.

```
%token NUMBER
%token OP_ADD OP_SUB OP_MUL OP_RGT OP_LFT OP_EQU

%%
calc:
 | calc exp OP_EQU    { printf("Y: RESULT = %d\n", $2); }
 ;

exp: factor
 | exp OP_ADD factor  { $$ = $1 + $3; }
 | exp OP_SUB factor  { $$ = $1 - $3; }
 ;

factor: term
 | factor OP_MUL term { $$ = $1 * $3; }
 ;

term: NUMBER
 | OP_LFT exp OP_RGT  { $$ = $2; }
  ;
%%

int main(int argc, char **argv)
{
  yyparse();
}

int yyerror(char *s)
{
  fprintf(stderr, "error: '%s'\n", s);
}
```

Créons `example.l` :

```
/* calculator source for flex */
%{
#include "example.tab.h"
%}

%%
[0-9]+ { printf("L: NUMBER = %s\n", yytext); yylval = atoi(yytext); return NUMBER; }
"+"    { printf("L: OP_ADD\n"); return OP_ADD; }
"-"    { printf("L: OP_SUB\n"); return OP_SUB; }
"*"    { printf("L: OP_MUL\n"); return OP_MUL; }
"("    { printf("L: OP_LFT\n"); return OP_LFT; }
")"    { printf("L: OP_RGT\n"); return OP_RGT; }
"="    { printf("L: OP_EQU\n"); return OP_EQU; }
"exit" { printf("L: exit\n");   return YYEOF; } /* YYEOF = 0 */
.      { /* ignore all other */ }
%%
```

Puis exécutons comme suit à partir de l'invite de l'interpréteur pour tester cet exemple :

```
$ bison -d example.y
$ flex example.l
$ gcc -lfl example.tab.c lex.yy.c -o example
$ ./example
1 + 2 * ( 3 + 1 ) =
L: NUMBER = 1
L: OP_ADD
L: NUMBER = 2
```

```
L: OP_MUL
L: OP_LFT
L: NUMBER = 3
L: OP_ADD
L: NUMBER = 1
L: OP_RGT
L: OP_EQU
Y: RESULT = 9

exit
L: exit
```

12.4 Outils d'analyse du code statique

Des outils de type lint pour aider à l'analyse du code statique automatique.

Des outils de type Indent peuvent aider pour des révisions humaines du code en reformatant les codes sources de manière cohérente.

Des outils de type Ctags peuvent aider pour les révisions humaines de code en générant un fichier d'index (ou de balise) des noms trouvés dans les codes source.

ASTUCE
Configurer votre éditeur préféré (`emacs` ou `vim`) pour utiliser des extensions de moteur de vérification syntaxique asynchrone facilite l'écriture de votre code. Ces extensions deviennent très puissantes en tirant parti du Language Server Protocol. Comme elles évoluent rapidement, utiliser leur code amont au lieu du paquet Debian peut être une bonne option.

12.5 Déboguer

Le débogage est une partie de l'activité de programmation. Savoir comment déboguer des programmes fera de vous un bon utilisateur de Debian qui pourra produire des rapports de bogues documentés.

12.5.1 Exécution de base de gdb

Le debogueur primaire sous Debian est gdb(1), il vous permet d'inspecter un programme alors qu'il tourne.

Installons `gdb` et les programmes associés par ce qui suit :

```
# apt-get install gdb gdb-doc build-essential devscripts
```

Des bons tutoriels sur `gdb` sont disponibles :

— « `info gdb` » ;
— « Déboguer avec GDB » dans `/usr/share/doc/gdb-doc/html/gdb/index.html` ;
— « tutoriel sur le web ».

Voici un exemple simple d'utilisation de gdb(1) sur un « `programme` » compilé avec l'option « `-g` » qui produit les informations de débogage :

paquet	popularité	taille	description
vim-ale	I:0.72	2833	moteur de vérification syntaxique asynchrone pour Vim 8 et NeoVim
vim-syntastic	I:1.8	1379	bidouilles de vérification de la syntaxe pour vim
elpa-flycheck	V:0.1, I:1.5	1130	vérification moderne de la syntaxe à la volée pour Emacs
elpa-relint	I:0.05	150	recherche d'erreurs d'expressions rationnelles dans Emacs Lisp
cppcheck-gui	V:0.09, I:0.93	7973	outil graphique d'analyse statique du code C/C++
shellcheck	V:4, I:17	22860	outil d'analyse syntaxique pour les scripts d'interpréteur
pyflakes3	V:2, I:14	20	vérificateur passif des programmes en Python 3
pylint	V:4, I:19	2109	vérificateur de code statique Python
perl	V:672, I:991	840	interpréteur ayant un vérificateur de code statique interne : `B::Lint`(3perl)
rubocop	V:0.08, I:1.00	4708	analyseur de code statique en Ruby
clang-tidy	V:2, I:11	20	outil d'analyse de C++ basé sur clang
splint	V:0.05, I:0.75	2325	outil pour vérifier de manière statique les bogues d'un programme en C
flawfinder	V:0.05, I:0.41	187	outil pour examiner le code source en C/C++ et rechercher des faiblesses du point de vue de la sécurité
black	V:3, I:15	11130	formateur de code Python intransigeant
perltidy	V:0.4, I:2.9	3086	indenteur et formateur de script Perl
indent	V:0.3, I:4.7	438	programme de formatage de code source en langage C
astyle	V:0.2, I:2.4	770	indenteur de code source pour C, C++, Objective-C, C# et Java
bcpp	V:0.02, I:0.21	114	embellisseur pour C(++)
xmlindent	V:0.06, I:0.70	52	reformateur de flux XML
global	V:0.1, I:1.4	1923	outils de recherche et de navigation dans le code source
exuberant-ctags	V:1, I:12	341	constructeur des index de fichiers de balises des définitions du code source
universal-ctags	V:1, I:12	4238	constructeur des index de fichiers de balises des définitions du code source

Table 12.12 – Liste des outils d'analyse du code statique :

paquet	popularité	taille	documentation
gdb	V:94, I:163	13472	« `info gdb` » fourni par `gdb-doc`
ddd	V:0.2, I:4.8	4519	« `info ddd` » fourni par `ddd-doc`

Table 12.13 – Liste des paquets de débogage

```
$ gdb program
(gdb) b 1               # set break point at line 1
(gdb) run args          # run program with args
(gdb) next              # next line
...
(gdb) step              # step forward
...
(gdb) p parm            # print parm
...
(gdb) p parm=12         # set value to 12
...
(gdb) quit
```

ASTUCE
De nombreuses commandes de gdb(1) possèdent une abréviation. L'expansion à l'aide de la touche de tabulation fonctionne comme avec l'interpréteur de commandes.

12.5.2 Déboguer un paquet Debian

Étant donné que par défaut tous les binaires installés doivent être réduits à l'essentiel sur le système Debian, la plupart des symboles de débogage sont supprimés dans les paquets normaux. Pour déboguer les paquets Debian avec gdb(1), les paquets `*-dbgsym` doivent être installés (par exemple, `coreutils-dbgsym` dans le cas de `coreutils`). Les paquets source génèrent automatiquement les paquets `*-dbgsym` avec les paquets binaires normaux et ces paquets de débogage sont placés séparément dans l'archive debian-debug. Veuillez vous référer aux articles sur le wiki de Debian pour plus d'informations.

Si un paquet à déboguer ne fournit pas son paquet `*-dbgsym`, vous devrez l'installer après l'avoir reconstruit comme suit :

```
$ mkdir /path/new ; cd /path/new
$ sudo apt-get update
$ sudo apt-get dist-upgrade
$ sudo apt-get install fakeroot devscripts build-essential
$ apt-get source package_name
$ cd package_name*
$ sudo apt-get build-dep ./
```

Corriger les bogues si nécessaire.

Modifier la version du paquet pour ne pas entrer en collision avec les versions officielles de Debian, par exemple, en ajoutant « `+debug1` » pour la compilation d'une version de paquet existante, ou « `~pre1` » pour la compilation d'une version de paquet qui n'est pas encore diffusée de la manière suivante :

```
$ dch -i
```

Compiler et installer les paquets avec les symboles de débogage comme suit :

```
$ export DEB_BUILD_OPTIONS="nostrip noopt"
$ debuild
$ cd ..
$ sudo debi package_name*.changes
```

Vous devrez vérifier les scripts de construction du paquet et vous assurer que les options « `CFLAGS=-g -Wall` » sont positionnées pour la compilation des binaires.

12.5.3 Obtenir une trace

Si vous rencontrez un plantage de programme, signaler le bogue avec un copier-coller des informations de trace est une bonne idée.

La trace d'appels peut être obtenue avec `gdb` en utilisant une des manières suivantes :

- Approche « plantage dans GDB » :
  - exécutez le programme à partir de GDB ;
  - plantez le programme ;
  - tapez « `bt` » à l'invite de GDB.
- Approche « plantage d'abord » :
  - mettez à jour le fichier « **/etc/security/limits.conf** » pour inclure l'élément suivant :

```
* soft core unlimited
```

  - tapez « `ulimit -c unlimited` » à l'invite de l'interpréteur ;
  - exécutez le programme à partir de cette invite de l'interpréteur ;
  - faites planter le programme pour produire un fichier core dump ;
  - chargez le fichier core dump dans GDB en tant que « `gdb gdb ./program_binary core` » ;
  - tapez « `bt` » à l'invite de GDB.

Pour une situation de boucle infinie ou de clavier gelé, vous pouvez forcer le plantage du programme en appuyant sur `Ctrl-\` ou `Ctrl-C` ou en exécutant « `kill -ABRT` *`PID`* » (consulter la Section 9.4.12).

ASTUCE
Souvent, vous voyez une trace où une ou plusieurs des lignes de départ se trouvent dans « `malloc()` » ou « `g_malloc()` ». Lorsque cela arrive, il y a des chances pour que votre trace ne soit pas très utile. La meilleure façon de trouver des informations utiles est de définir la variable d'environnement « `$MALLOC_CHECK_` » à la valeur 2 (malloc(3)). Vous pouvez le faire en lançant `gdb` de la manière suivante :

```
$ MALLOC_CHECK_=2 gdb hello
```

12.5.4 Commandes avancées de gdb

commande	description des objectifs des commandes
`(gdb) thread apply all bt`	obtenir une trace de tous les processus d'un programme multi-processus (multi-threaded)
`(gdb) bt full`	obtenir les paramètres qui se trouvent sur la pile d'appel des fonctions
`(gdb) thread apply all bt full`	obtenir une trace et les paramètres en combinant les options précédentes
`(gdb) thread apply all bt full 10`	obtenir une trace et les paramètres des 10 premiers appels en supprimant ce qui n'est pas significatif
`(gdb) set logging on`	écrire le journal de sortie de `gdb` dans un fichier (le fichier par défaut est « `gdb.txt` »)

Table 12.14 – Liste des commandes avancées de gdb

12.5.5 Vérifier les dépendances avec les bibliothèques

Utilisez ldd(1) pour trouver les dépendances d'un programme avec des bibliothèques :

```
$ ldd /usr/bin/ls
        librt.so.1 => /lib/librt.so.1 (0x4001e000)
        libc.so.6 => /lib/libc.so.6 (0x40030000)
        libpthread.so.0 => /lib/libpthread.so.0 (0x40153000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Pour que ls(1) fonctionne dans un environnement `chroot`é, les bibliothèques ci-dessus doivent être disponibles dans votre environnement `chroot`é.

Consultez Section 9.4.6.

12.5.6 Outils de traçage dynamique des appels

Il existe plusieurs outils de traçage d'appels dynamique disponibles dans Debian. Consulter la Section 9.4.

12.5.7 Déboguer les erreurs de X

Si un programme `apercu1` de GNOME a reçu une erreur X, vous devriez obtenir un message comme suit :

```
The program 'preview1' received an X Window System error.
```

Dans ce cas, vous pouvez essayer de faire tourner le programme avec « `--sync` » et arrêter sur la fonction « `gdk_x_error` » de manière à obtenir une trace.

12.5.8 Outils de détection des fuites de mémoire

Il y a plusieurs outils de détection des fuites de mémoire disponibles sous Debian.

paquet	popularité	taille	description
`libc6-dev`	V:259, I:577	12863	mtrace(1) : fonctionnalité de débogage de malloc dans glibc
`valgrind`	V:4, I:31	89318	débogueur mémoire et optimiseur
`electric-fence`	V:0.2, I:2.2	69	débogueur malloc(3)
`libdmalloc5`	I:0.54	373	bibliothèque de débogage de l'allocation mémoire
`duma`	I:0.06	295	bibliothèque permettant de détecter les dépassements et les sous-alimentations de tampon dans les programmes C et C++

Table 12.15 – Liste des outils de détection des fuites de mémoire

12.5.9 Désassembler un binaire

Vous pouvez désassembler du code binaire avec objdump(1) en faisant ce qui suit :

```
$  objdump -m i386 -b binary -D /usr/lib/grub/x86_64-pc/stage1
```

Note
gdb(1) peut être utilisé pour désassembler du code de manière interactive.

12.6 Outils de construction

paquet	popularité	taille	documentation
make	V:155, I:558	1762	« info make » fourni par make-doc
autoconf	V:28, I:189	2244	« info autoconf » fourni par autoconf-doc
automake	V:27, I:188	1959	« info automake » fourni par automake
libtool	V:23, I:172	1276	« info libtool » fourni par libtool-doc
cmake	V:23, I:124	63999	cmake(1), système make multiplateforme et au code source ouvert
ninja-build	V:9, I:55	456	ninja(1), petit système de construction le plus proche de l'esprit de Make
meson	V:6, I:29	4402	meson(1), système de construction à haute productivité sur au-dessus de ninja
xutils-dev	V:0.6, I:6.6	1495	imake(1), xmkmf(1), etc.

Table 12.16 – Liste des paquets d'outil de construction

12.6.1 Make

Make est un utilitaire destiné à la maintenance d'un groupe de programmes. Lors de l'exécution de make(1), make lit le fichier de règles, « Makefile » et met à jour une cible si elle dépend de fichiers qui ont été modifiés depuis que la cible a été modifiée pour la dernière fois ou si la cible n'existe pas. L'exécution de ces mises à jour peut être faite simultanément.

La syntaxe du fichier de règles est la suivante :

```
target: [ prerequisites ... ]
 [TAB]  command1
 [TAB]  -command2 # ignore errors
 [TAB]  @command3 # suppress echoing
```

Ici, « [TAB] » est un code de tabulation. Chaque ligne est interprétée par l'interpréteur de commandes après que make ait effectué la substitution des variables. Utilisez « \ » à la fin d'une ligne pour poursuivre le script. Utilisez « $$ » pour entrer un « $ » pour les valeurs des variables d'environnement d'un script de l'interpréteur de commandes.

On peut écrire des règles implicites pour la cible et les prérequis, par exemple, de la manière suivante :

```
%.o: %.c header.h
```

Ici, la cible contient le caractère « % » (exactement 1 caractère). Le caractère « % » peut correspondre à n'importe quelle sous-chaîne non vide des noms de fichiers de la cible actuelle. De même pour les prérequis, utilisez « % » pour afficher la manière dont leur nom est en relation avec le nom de la cible actuelle.

variables automatiques	valeur
$@	cible
$<	première exigence
$?	toutes les exigences plus récentes
$^	toutes les exigences
$*	« % » correspond au radical dans le motif cible

Table 12.17 – Liste des variables automatiques de make

Exécutez « make -p -f/dev/null » afin de voir les règles automatiques internes.

expansion de la variable	description
`foo1 := bar`	expansion à la volée
`foo2 = bar`	expression récursive
`foo3 += bar`	ajouter

Table 12.18 – Liste de l'expansion des variables de make

12.6.2 Autotools

Autotools est une suite d'outils de programmation conçus pour aider à rendre les paquets de code source portables sur de nombreux systèmes de type Unix .

— Autoconf est un outil permettant de produire un script d'interpréteur « `configure` » à partir de « `configure.ac` ».

— « `configure` » est utilisé ultérieurement pour produire un « `Makefile` » à partir du modèle « `Makefile.in` ».

— Automake est un outil permettant de produire un « `Makefile.in` » à partir de « `Makefile.am` ».

— Libtool est un script d'interpréteur permettant de résoudre le problème de portabilité des logiciels lors de la compilation de bibliothèques partagées à partir du code source.

12.6.2.1 Compiler et installer un programme



AVERTISSEMENT
Ne pas écraser les fichiers du système avec les programmes que vous avez compilés en les installant.

Debian ne touche pas aux fichiers se trouvant dans « `/usr/local/` » ou « `/opt` ». Donc, si vous compilez un programme depuis ses sources, installez-le dans « `/usr/local/` » de manière à ce qu'il n'interfère pas avec Debian.

```
$ cd src
$ ./configure --prefix=/usr/local
$ make # this compiles program
$ sudo make install # this installs the files in the system
```

12.6.2.2 Désinstaller un programme

Si vous avez les sources d'origine et s'ils utilisent autoconf(1) et automake(1) et si vous-vous souvenez comment vous l'avez configuré, exécutez-le comme suit pour désinstaller le programme :

```
$ ./configure all-of-the-options-you-gave-it
$ sudo make uninstall
```

Sinon, si vous êtes absolument certain que le processus d'installation n'a mis des fichiers que sous « `/usr/local/` » et qu'il n'y a là rien d'important, vous pouvez supprimer tout son contenu avec :

```
# find /usr/local -type f -print0 | xargs -0 rm -f
```

Si vous n'êtes pas certain de l'emplacement où les fichiers ont été installés, vous devriez envisager d'utiliser checkinstall(8) du paquet `checkinstall`, qui fournit un chemin propre pour la désinstallation. Il prend maintenant en charge la création d'un paquet Debian à l'aide de l'option « `-D` ».

12.6.3 Meson

Le système de construction des logiciels a évolué :

— Autotools au-dessus de Make a été la norme de facto pour l'infrastructure de construction portable depuis les années 1990. Ce système est extrêmement lent ;
— CMake, initialement publié en 2000, a considérablement amélioré la vitesse, mais il était à l'origine construit sur Make, intrinsèquement lent (aujourd'hui, Ninja peut être son dorsal) ;
— Ninja initialement publié en 2012 est destiné à remplacer Make pour une meilleure vitesse de construction et est conçu pour que ses fichiers d'entrée soient générés par un système de construction de plus haut niveau ;
— Meson, initialement publié en 2013, est le nouveau système de construction de haut niveau, populaire et rapide, qui utilise Ninja comme dorsal.

Consultez les documents disponibles dans « The Meson Build system » et « The Ninja build system ».

12.7 Web

Des pages web dynamiques et interactives simples peuvent être faites de la manière suivante :

— Les requêtes sont présentées au navigateur de l'utilisateur en utilisant des formulaires HTML.
— Remplir et cliquer sur les entrées de formulaires envoie une des chaînes d'URL suivantes avec des paramètres codés depuis le navigateur vers le serveur web.
  — "`https://www.foo.dom/cgi-bin/program.pl?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3`"
  — "`https://www.foo.dom/cgi-bin/program.py?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3`"
  — "`https://www.foo.dom/program.php?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3`"
— « %nn » dans l'URL est remplacé par le caractère dont la valeur hexadécimale est `nn`.
— La variable d'environnement est définie à : « `QUERY_STRING="VAR1=VAL1 VAR2=VAL2 VAR3=VAL3"` ».
— Le programme CGI (l'un quelconque des « `programme.*` ») sur le serveur web s'exécute lui-même avec la variable d'environnement « `$QUERY_STRING` ».
— La `sortie standard (stdout)` du programme CGI est envoyée au navigateur web et présentée sous forme d'une page web dynamique interactive.

Pour des raisons de sécurité, il est préférable de ne pas réaliser soi-même de nouvelles bidouilles pour analyser les paramètres CGI. Il existe des modules bien établis pour cela, en Perl et Python. PHP est fourni avec ces fonctionnalités. Lorsqu'il est nécessaire d'enregistrer des données du client, on utilise des cookies HTTP. Lorsqu'un traitement de données est nécessaire côté client, on utilise fréquemment Javascript.

Pour davantage d'informations, consultez Common Gateway Interface, The Apache Software Foundation et JavaScript.

Rechercher « CGI tutorial » sur Google en entrant l'URL encodée https://www.google.com/search?hl=en&ie=UTF-8&q=CGI+tutorial directement dans la barre d'adresse du navigateur est une bonne méthode pour voir un script CGI en action sur le serveur Google.

12.8 La conversion du code source

Il existe des programmes pour convertir les codes sources.

12.9 Créer un paquet Debian

Si vous désirez créer un paquet Debian, lisez ce qui suit :

— Chapitre 2 pour comprendre les bases du système de paquets

paquet	popularité	taille	mot clé	description
f2c	V:0.1, I:1.9	443	FORTRAN → C	convertir le code source de FORTRAN 77 vers C/C++ : f2c(1)
intel2gas	V:0.01, I:0.17	178	NASM → GAS	convertisseur depuis NASM (format Intel) vers l'Assembleur GNU (GAS)

Table 12.19 – Liste des outils de conversion de code source

— Section 2.7.13 pour comprendre les bases du processus de portage
— Section 9.11.4 pour comprendre les techniques de base d'un environnement isolé (« chroot »)
— debuild(1) et sbuild(1)
— Section 12.5.2 pour recompiler avec les informations de débogage
— Guide pour les responsables Debian (le paquet `debmake-doc`)
— Référence du développeur Debian (paquet `developers-reference`)
— Charte Debian (paquet `debian-policy`)

Il existe des paquets tels que `debmake`, `dh-make`, `dh-make-perl`, etc., qui facilitent la réalisation des paquets.

Annexe A

Annexe

Voici les fondements de ce document.

A.1 Le labyrinthe de Debian

Le système Linux est une plateforme informatique très performante pour un ordinateur connecté au réseau. Cependant, apprendre à utiliser toutes ses possibilités n'est pas si facile. Configurer LPR avec une imprimante qui ne soit pas PostScript en était un bon exemple. (Il n'y a plus de problème maintenant car les nouvelles installations utilisent CUPS.)

Il existe une carte très détaillée appelée « CODE SOURCE ». Elle est très précise mais très difficile à comprendre. Il existe aussi des références appelées HOWTO et mini-HOWTO. Elles sont plus faciles à comprendre mais ont tendance à trop se concentrer sur des détails et perdent de vue les aspects généraux. J'ai de temps en temps des problèmes à trouver la bonne section dans un long HOWTO quand j'ai besoin d'exécuter certaines commandes.

J'espère que cette « Référence Debian (version 2.153) ») (2026-11-29 05:38:00 UTC) fournit un bon guide pour les personnes s'engageant dans le dédale de Debian.

A.2 Historique du Copyright

La Référence Debian fut lancée par moi-même, Osamu Aoki <osamu at debian dot org> en tant qu'aide-mémoire personnel pour l'administration système. De nombreuses parties proviennent des connaissances que j'ai acquises sur la liste de diffusion debian-user et d'autres ressources Debian.

En suivant les suggestions de Josip Rodin, qui a été très actif dans le Projet de documentation Debian (DDP), la « Référence Debian (version 1, 2001-2007) » a été créée en tant que partie des documents du DDP.

Au bout de 6 années, j'ai réalisé que la « Référence Debian (version 1) » était dépassée et ai commencé à en réécrire de nombreux passages. La nouvelle « Référence Debian (version 2) » a été diffusée en 2008.

J'ai mis à jour la « Référence Debian (version 2) » pour traiter les nouveaux sujets (Systemd, Wayland, IMAP, PipeWire, noyau Linux 5.10) et supprimé les sujets obsolètes (SysV init, CVS, Subversion, protocole 1 SSH, noyaux Linux avant 2.5). Les références aux contextes des publications Jessie 8 (2015-2020) ou plus ancienne sont pour la plupart supprimées.

Cette « Référence Debian (version 2.153) » (2026-11-29 05:38:00 UTC) couvre principalement les publications Trixie (`=stable`) et Forky (`=testing`) de Debian.

Le contenu de ce tutoriel tire son origine et son inspiration dans ce qui suit :

— « Linux User's Guide » de Larry Greenfield (décembre 1996)

  - rendu obsolète par « Debian Tutorial »
- « Debian Tutorial » de Havoc Pennington. (11 décembre 1998)
  - écrit partiellement par Oliver Elphick, Ole Tetlie, James Treacy, Craig Sawyer et Ivan E. Moore II
  - rendu obsolète par « Debian GNU/Linux : Guide to Installation and Usage »
- « Debian GNU/Linux : Guide to Installation and Usage » de John Goerzen et Ossama Othman (1999)
  - rendue obsolète par la « Référence Debian (version 1) »

Les descriptions des paquets et des archives peuvent trouver une partie de leur origine et de leur inspiration dans ce qui suit :

- « FAQ Debian » (version de mars 2002, alors qu'elle était maintenue par Josip Rodin)

Le reste du contenu peut trouver son origine et son inspiration dans ce qui suit :

- « Référence Debian (version 1) » de Osamu Aoki (2001–2007)
  - rendue obsolète par la nouvelle « Debian Reference (version 2) » en 2008.

La version précédente « Référence Debian (version 1) » avait été créée par de nombreux contributeurs :

- la contribution principale pour les sujets relatifs à la configuration du réseau par Thomas Hood ;
- une importante contribution au contenu sur les sujets relatifs à X et VCS par Brian Nelson ;
- l'aide pour les scripts de construction et de nombreuses corrections de contenu par Jens Seidel ;
- une relecture intensive de David Sewell ;
- de nombreuses contributions par les traducteurs, les contributeurs et ceux qui ont signalés des bogues.

De nombreuses pages de manuel, de pages info du système Debian, de pages web de l'amont ainsi que Wikipédia ont servi de références pour écrire ce document. Dans la mesure où Osamu Aoki considérait que cela faisait partie d'un usage loyal, de nombreuses parties d'entre-elles, particulièrement les définitions des commandes, ont été utilisées comme morceaux de phrase après un effort éditorial soigneux afin de les insérer dans le style et les objectifs de ce document.

La description du débogueur gdb a été augmentée en utilisant le contenu du wiki Debian sur les « backtrace » avec le consentement d'Ari Pollak, Loïc Minier et Dafydd Harries.

Le contenu de « Référence Debian (version 2.153) » (2026-11-29 05:38:00 UTC) est principalement un travail personnel à l'exception de ce qui est mentionné ci-dessus. Il a aussi été mis à jour par les contributeurs.

L'auteur, Osamu Aoki, remercie tous ceux qui ont aidé à rendre possible ce document.

A.3 Format du document

Le source du document originel en anglais est actuellement écrite dans des fichiers XML en DocBook. Cette source XML Docbook est convertie en HTML, texte brut, PostScript et PDF. (Certains formats peuvent être ignorés pour la distribution.)