Guide de référence pour Debian

Osamu Aoki `<osamu\#at\#debian.org>`
Traduction en Français : Guillaume Erbs `<gerbs\#at\#free.fr>`
'Auteurs' page 241

CVS, jeu 18 jan 2007 11 :52 :23 UTC

Résumé

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

Copyright

Table des matières

Chapitre 1

Préface

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

J'ai pris la décision de **ne pas** expliquer tout en détail si cela peut être trouvé dans une **page de manuel**, une **page info**, ou un **HOWTO**. Plutôt que de tout expliquer, j'essaie de donner des informations pratiques plus directement en fournissant des **séquences de commandes exactes** ou des **scripts d'exemple** dans `http://www.debian.org/doc/manuals/debian-reference/examples/` comme référence. Vous devez comprendre le contenu des exemples avant de lancer les commandes. Votre système peut nécessiter des commandes légèrement différentes.

La plupart des informations données consistent en rappels ou en pointeurs vers les références officielles listées dans 'Références' page 233. Ceci partiellement parce que ce document a été démarré en tant que « **Référence Rapide** ».

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

Pour une aide en vue de la maintenance urgente d'un système, rendez-vous à 'Commandes de survie Debian' page 70 immédiatement.

1.1 Document officiel

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

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

1.2 Conventions du document

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

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

Voir 'Bash – le shell intéractif standard de **GNU**' page 211 pour plus d'information sur Bash.

Références :

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

Les abréviations suivantes sont utilisées :

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

Dans ce document, seules des URL sont fournies pour les documents du LDP, mais ils peuvent aussi être obtenus dans des paquets et installés dans `/usr/share/doc/HOWTO/`. Voir 'Références' page 233.

1.3 Scripts d'exemple

Des scripts d'exemple sont fournis dans le répertoire des exemples (`http://www.debian.org/doc/manuals/debian-reference/examples/`); pour les fichiers cachés, le préfixe « . » est remplacé par un « _ ». Un suffixe a été ajouté lorsqu'il y a plusieurs scripts d'exemple pour un même sujet.

1.4 Configuration de base

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

```
# aptitude install mc less ssh vim kernel-package \
                  manpages-dev doc-debian doc-linux-text \
                  debian-policy developers-reference maint-guide \
                  apt-howto harden-doc debian-reference \
                  libpam-doc glibc-doc samba-doc exim4-doc-html
```

1.5 Fondements de la distribution Debian

Debian se présente sous la forme simultanée de 3 éditions :

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

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

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

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

  - 'Mise à jour d'une distribution vers `testing`' page 61 (principalement pour des stations de travail)
  - 'Porter un paquet vers le système `stable`' page 78 (principalement pour des serveurs)

Suivre la version `testing` de Debian a l'effet de bord de recevoir très lentement les correctifs de sécurité. Vous êtes prévenus.

Si vous mélangez les versions de Debian, `testing` dans `stable` ou `unstable` dans `stable`, vous aurez certainement à récupérer beaucoup de paquets importants de `testing` ou `unstable` qui peuvent être bogués. Vous êtes prévenus.

Utiliser les versions `testing` ou `unstable` de Debian impliquent l'augmentation du risque de tomber sur des bogues sérieux. Ce risque peut-être géré en utilisant un schéma de multiboot avec une version plus stable de Debian ou en utilisant l'astuce de `chroot` avec la version stabe décrite dans '`chroot`' page 116. Cette dernière permet d'utiliser différentes versions de Debian simultanément sur différentes consoles.

Après avoir expliqué quelques éléments fondamentaux de la distribution Debian dans 'Notions fondamentales sur Debian' page 5, je vais présenter quelques informations de base pour vous aider à vivre tranquillement avec les dernières applications, en tirant parti de la distribution `testing` et de la distribution `unstable` de Debian. Les impatients devront aller immédiatement à 'Commandes de survie Debian' page 70. Bonne et heureuse mise à jour !

Chapitre 2

Notions fondamentales sur Debian

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

- Charte Debian
- Référence du développeur Debian
- Guide des nouveaux responsables Debian

listés dans 'Références' page 233.

Si l'on recherche des explications basées sur des solutions et sans les détails, se référer directement à 'Gestion des paquets Debian' page 65 ou aux chapitres appropriés.

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

2.1 Les archives Debian

2.1.1 Structure de répertoires

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

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

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

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

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

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

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

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

- **`project/experimental/` :** Ce répertoire contient des paquets et des outils qui sont en développement, et sont encore en état de test alpha. Les utilisateurs ne devraient pas utiliser des paquets de ce répertoire parce qu'ils peuvent être dangereux même pour des utilisateurs expérimentés.
- **`project/orphaned/` :** Paquets qui ont été abandonnés par leur ancien responsable et ont été retirés de la distribution.

2.1.2 Distributions Debian

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

2.1.3 La distribution `stable`

Les paquets de la distribution `stable`, Debian Sarge (3.1r0), sont enregistrés dans le répertoire `stable` (lien symbolique vers `Sarge`) :

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

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

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

2.1.4 La distribution `testing`

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

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

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

2.1.5 La distribution `unstable`

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

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

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

2.1.6 La distribution `frozen`

Lorsque la distribution `testing` est mûre, elle est gelée (NdT : *frozen* en Anglais), c'est-à-dire que l'on n'accepte plus de nouveau code, seulement des corrections de bogues, si nécessaire. De plus, un nouvel arbre testing est créé dans le répertoire `dists`, avec un nouveau nom de code. La distribution frozen subit quelques mois de test, avec par intermittence des mises à jour et des gelées complètes, ce qu'on appelle des 'cycles de test'.

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

2.1.7 Les noms de code de la distribution Debian

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

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

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

2.1.8 Noms de code utilisés par le passé

Les noms de code qui ont déjà été utilisés sont : `buzz` pour la version 1.1, `rex` pour la version 1.2, `bo` pour les versions 1.3.x, `hamm` pour la version 2.0, `slink` pour la version 2.1, `potato` pour la version 2.2, `woody` pour la version 3.0 et `sarge` pour la version 3.1.

2.1.9 Source d'inspiration pour les noms de code

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

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

2.1.10 Le répertoire `pool`

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

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

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

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

Normalement, vous n'avez pas à vous occuper de cela, puisque le nouvel `apt` et probalement l'ancien `dpkg-ftp` (voir 'Méthodes de mise à jour d'un système Debian' page 18) vont gérer cela de façon transparente. Si vous souhaitez plus d'information, consultez RFC : implementation of package pools (`http://lists.debian.org/debian-devel-announce/2000/debian-devel-announce-200010/msg00007.html`) (en Anglais).

2.1.11 Notes historiques sur `sid`

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

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

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

2.1.12 Paquets téléchargés dans `incoming`

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

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

2.1.13 Récupérer un paquet ancien

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

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

2.1.14 Sections architectures

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

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

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

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

2.1.15 Le code source

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

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

Certains paquets, notamment `pine`, sont seulement disponibles sous forme de paquet source, à cause de limitations de leur licence. (Récemment, le paquet `pine-tracker` a été fourni pour faciliter l'installation de Pine.) Les procédures décrites dans 'Porter un paquet vers le système `stable`' page 78 et 'Paquetage' page 226 permettent de construire un paquet manuellement.

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

2.2 Système de gestion des paquets Debian

2.2.1 Vue générale des paquets Debian

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

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

L'installation de logiciels par le système de paquets utilise des « dépendances » qui sont soigneusement conçues par les responsables du paquet. Ces dépendances sont documentées dans le fichier `control` associé à chaque paquet. Par exemple, le paquet contenant le compilateur GNU C (`gcc`) « dépend » du paquet `binutils` qui inclut l'éditeur de liens et l'assembleur. Si un utilisateur tente d'installer `gcc` sans avoir d'abord installé `binutils`, le système de gestion de paquets (dpkg) renverra un message d'erreur disant qu'il a besoin de `binutils`, et cessera l'installation de `gcc`. (Cependant, un utilisateur insistant pourra passer outre ; voir `dpkg(8)`.) Pour plus de détails, voir 'Dépendances des paquets' page 15 ci-dessous.

Les outils de paquetage de Debian peuvent être utilisés pour :

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

2.2.2 Format des paquets Debian

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

Les données internes de ce format de paquets binaires Debian sont décrites dans la page de manuel `deb(5)`. Parce que ce format interne est sujet à des changements (entre les sorties majeures de Debian), utilisez toujours `dpkg-deb(1)` pour manipuler des fichiers `.deb`.

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

2.2.3 Conventions de nommage pour les fichiers de paquets Debian

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

```
foo_VersionNumber-DebianRevisionNumber.deb
```

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

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

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

La composante `RRR` est le numéro de révision Debian spécifié par le développeur Debian (ou un utilisateur s'il choisit de construire le paquet lui-même). Ce numéro correspond au niveau de révision du paquet Debian ; ainsi, un nouveau niveau de révision correspond habituellement à un changement dans le Makefile Debian (`debian/rules`), le fichier de contrôle Debian

(`debian/control`), les scripts d'installation ou de suppression (`debian/p*`), ou les fichiers de configuration utilisés avec le paquet.

2.2.4 Préservation de la configuration locale

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

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

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

```
dpkg --status package
```

et regardez la ligne « Conffiles : ».

Les détails du contenu d'un fichier Debian `conffiles` sont fournis dans la Charte Debian, section 11.7 (voir 'Références' page 233).

2.2.5 Scripts de maintenance Debian

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

Les fichiers individuels sont :

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

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

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

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

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

2.2.6 Priorité des paquets

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

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

2.2.7 Paquets virtuels

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

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

Debian fournit un mécanisme pour que, si plus d'un paquet qui fournit le même paquet virtuel est installé sur un système, l'administrateur puisse configurer l'un des deux comme paquet préféré. La commande utilisée est `update-alternatives`, et est décrite dans 'Commandes de rechange' page 82.

2.2.8 Dépendances des paquets

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

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

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

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

2.2.9 Signification de « pre-depends »

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

Cependant, pour certains paquets, `dpkg` refusera même de les dépaqueter tant que certaines dépendances ne seront pas satisfaites. On dit que ces paquets « pré-dépendent » de la présence d'autres paquets. Le projet Debian fournissait ce mécanisme pour supporter la mise à jour sûre des systèmes du format `a.out` au format ELF, pendant laquelle **l'ordre** dans lequel les paquets étaient dépaquetés était critique. Il y a d'autres situations de mise à jour pour lesquelles cette méthode est utile, par exemple pour les paquets avec la priorité « required » et leur dépendance à la libc.

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

2.2.10 Etat d'un paquet

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

Leur signification est :

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

2.2.11 Garder des paquets lors d'une mise à jour

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

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

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

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

```
libc6                         install
```

en :

```
libc6                         hold
```

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

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

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

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

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

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

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

2.2.12 Paquets sources

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

```
apt-get source foo
```

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

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

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

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

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

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

Lancez la commande suivante pour compiler le paquet binaire :

```
$ cd foo-version
$ su -c "apt-get update ; apt-get install fakeroot"
$ dpkg-buildpackage -rfakeroot -us -uc
```

Puis,

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

pour installer le paquet nouvellement construit. Voir 'Porter un paquet vers le système `stable`' page 78.

2.2.14 Créer de nouveaux paquets Debian

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

2.3 Mettre à jour un système Debian

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

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

Un guide pratique pour les mises à jour est fourni dans la 'Gestion des paquets Debian' page 65. Cette section décrit les détails fondamentaux.

2.3.1 Méthodes de mise à jour d'un système Debian

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

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

2.3.2 Vue générale des outils de gestion de paquets

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

2.3.3 `dpkg`

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

`dpkg` vient avec plusieurs programmes primitifs supplémentaires.

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

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

2.3.4 APT

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

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

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

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

2.3.5 `dselect`

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

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

2.3.6 Mise à jour d'un système en marche

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

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

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

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

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

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

2.3.8 Garder une trace des mises à jour

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

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

2.4 Le processus de démarrage de Debian

2.4.1 Le programme `init`

Comme tous les Unices, Debian démarre en exécutant le programme `init`. Le fichier de configuration de `init` (qui est `/etc/inittab`) spécifie que le premier script à exécuter doit être `/etc/init.d/rcS`. Ce script lance tous les scripts de `/etc/rcS.d/` en incluant le source ou en forkant un sous-processus, selon leur extension, pour exécuter des initialisations, comme la vérification et le montage des systèmes de fichiers, le chargement des modules, le démarrage des services réseau, le réglage de l'horloge, et l'exécution d'autres initialisations. Ensuite, pour compatibilité, il lance aussi les fichiers (sauf ceux ayant un « . » dans leur nom) de `/etc/rc.boot/`. Les scripts de ce dernier répertoire sont habituellement réservés à l'administrateur système, et leur utilisation dans des paquets est obsolète. Voir 'Initialisation du système' page 121 pour plus d'information.

2.4.2 Niveaux de fonctionnement

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

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

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

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

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

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

```
# /etc/init.d/exim4 reload
```

envoie au daemon exim4 un signal pour qu'il relise son fichier de configuration.

2.4.3 Personnaliser les niveaux de fonctionnement

Personnaliser les niveaux de fonctionnement est une tâche d'administration avancée. Les conseils suivants fonctionnent pour la plupart des services.

Pour activer le service *service* dans le niveau de fonctionnement *R*, créez le lien symbolique `/etc/rcR.d/Sxyservice` avec comme cible `../init.d/service`. Le numéro *xy* doit être le numéro assigné au service lors de l'installation du paquet.

Pour désactiver le service, renommez le lien symbolique en le faisant commencer par `K` à la place de `S` et en lui donnant le numéro 100 moins *xy*.

Il est plus facile d'utiliser un éditeur de niveaux de fonctionnement, comme `sysv-rc-conf` ou `ksysv` pour effectuer ces modifications.

Il est possible de supprimer le lien symbolique `S` d'un service dans le répertoire d'un niveau de fonctionnement au lieu de le renommer. Cela ne désactive pas le service, mais le laisse dans un état « flottant » du point de vue du système d'initialisation `sysv-rc` : lors d'un changement de niveau de fonctionnement, le service ne sera ni démarré ni arrêté mais sera laissé tel quel, qu'il soit en fonctionnement ou pas. Notez cependant qu'un service laissé dans un tel état sera démarré si son paquet est mis à jour, qu'il soit en fonctionnement ou pas lors de la mise à jour. C'est un défaut du système Debian actuel. Notez aussi que vous devriez laisser le lien symbolique `K` d'un service dans les niveaux de fonctionnement 0 et 6. Si vous supprimez tous les liens symboliques d'un service, le paquet les restaurera lors d'une mise à jour.

Il n'est **pas** conseillé de faire des changements sur les liens symboliques de `/etc/rcS.d/`.

2.5 Support de la diversité

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

- `dpkg-divert`, voir 'La commande `dpkg-divert`' page 81.
- `equivs`, voir 'Le paquet `equivs`' page 81.
- `update-alternative`, voir 'Commandes de rechange' page 82.
- `make-kpkg` peut s'accomoder de beaucoup de chargeurs . Voir `make-kpkg(1)`.

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

2.6 Internationalisation

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

Si votre système installé ne supporte pas toutes les possibilités de la langue dont vous avez besoin, si vous avez besoin de changer de langue ou d'installer un clavier différent pour supporter votre langue, voyez 'Localisation (l10n)' page 154.

2.7 Debian et le noyau

Voir 'Le noyau Linux et Debian' page 85.

2.7.1 Compiler un noyau avec des sources non Debian

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

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

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

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

2.7.2 Outils pour compiler un noyau personnalisé

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

```
# make-kpkg kernel_image
```

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

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

et dans la page de manuel `make-kpkg(1)` et 'Le noyau Linux et Debian' page 85.

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

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

2.7.3 Dispositions spéciales pour manipuler les modules

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

Comme les (nouveaux) fichiers Configure.help qui sont maintenant disponibles pour supporter la compilation de noyaux personnalisés, le paquet `modconf` est livré avec une série de fichiers d'aide (dans `/usr/share/modconf/`) qui fournissent des informations détaillées sur les arguments possibles pour chacun des modules. Voir 'Le noyau 2.4 modulaire' page 87 pour des exemples.

2.7.4 Désinstaller le paquet d'un vieux noyau

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

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

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

Chapitre 3

Installation du système Debian

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

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

Bien que « Guide de référence pour Debian » ait été commencé à l'époque de l'installateur de Potato, la majeure partie de son contenu a été mis à jour pour l'installateur de Woody qui est similaire. Comme Sarge utilisera un installateur complètement différent, utilisez ceci comme une référence de départ pour l'installateur de Sarge.

3.1 Astuces générales sur l'installation du système Linux

N'oubliez pas de consulter `http://www.debian.org/CD/netinst/` si vous recherchez une image CD de petite taille de l'installateur Debian.

Exécuter les versions `testing` ou `unstable` de Debian accroît le risque de tomber sur un bogue sérieux. Ce risque peut être géré en utilisant un schéma de multi-démarrage avec une version plus stable de Debian, ou en utilisant l'astuce fournie par `chroot` à l'intérieur de la version stable, comme décrite dans '`chroot`' page 116. Ce dernier permet d'exécuter plusieurs versions de Debian simultanément sur différentes consoles.

3.1.1 Compatibilité matérielle

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

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

Ma recommandation pour le matériel type PC de bureau est « soyez conservateur » :
- SCSI plutôt qu'IDE pour le travail, disque dur IDE/ATAPI pour une utilisation personnelle.
- CD-ROM (ou CD-RW) IDE/ATAPI.
- PCI plutôt qu'ISA, surtout pour la carte réseau.
- Utilisez une carte réseau de base. Des cartes Tulip sur bus PCI, NE2000 sur bus ISA conviennent bien.
- Évitez PCMCIA (portables) lors de la première tentative d'installation de Linux.
- Pas de clavier ou de souris USB,. . . à moins de vouloir faire face à un défi.

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

3.1.2 Déterminer le matériel et les puces du PC

Durant l'installation, on sera interrogé sur le matériel ou les puces. Parfois, ces informations ne sont pas toujours faciles à trouver. Voici une méthode :
1. Ouvrez le PC et inspectez l'intérieur.
2. Notez les codes produit qui sont sur les grandes puces de la carte graphique, de la carte réseau, sur la puce à côté des ports série et la puce à côté des ports IDE.
3. Notez les noms des cartes imprimés au dos des cartes PCI et ISA.

3.1.3 Trouver le matériel du PC avec Debian

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

```
$ lspci -v |less
$ pager /proc/pci
$ pager /proc/interrupts
$ pager /proc/ioports
$ pager /proc/bus/usb/devices
```

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

Utilisations classiques des interruptions :
- IRQ0 : sortie du timer (8254)
- IRQ1 : contrôleur clavier
- IRQ2 : cascade vers IRQ8–IRQ15 sur PC-AT
- IRQ3 : port série secondaire (io-port=0x2F8) (`/dev/ttyS1`)
- IRQ4 : port série primaire (io-port=0x3F8) (`/dev/ttyS0`)

- IRQ5 : libre (anciennement interruption disque dur sur XT)
- IRQ6 : contrôleur disquette (io-port=0x3F0) (`/dev/fd0`, `/dev/fd1`)
- IRQ7 : port parallèle (io-port=0x378) (`/dev/lp0`)
- IRQ8 : RTC
- IRQ9 : interruption logicielle (int 0x0A), redirige vers IRQ2
- IRQ10 : libre
- IRQ11 : libre
- IRQ12 : souris PS/2
- IRQ13 : libre (anciennement coprocesseur mathématique 80287)
- IRQ14 : contrôleur IDE primaire (`/dev/hda`, `/dev/hdb`)
- IRQ15 : contrôleur IDE secondaire (`/dev/hdc`, `/dev/hdd`)

Pour les périphériques USB, les classes de périphériques sont listées dans `/proc/bus/usb/devices` par `Cls=`*`nn`* :

- Cls=00 : Inutilisé
- Cls=01 : Audio (haut-parleur, ...)
- Cls=02 : Communication (modem, carte réseau, ...)
- Cls=03 : HID (Human Interface Device : clavier, souris, joystick)
- Cls=07 : Imprimante
- Cls=08 : Stockage de masse (Lecteur de disquette, de CD/DVD, disque dur, Flash, ...)
- Cls=09 : Hub (hub USB)
- Cls=255 : Spécifique au fabricant

Si la classe d'un périphérique n'est pas 255, alors Linux supporte ce périphérique.

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

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

Installez une autre distribution Linux commerciale. La détection du matériel tend à être meilleure que celle de Debian, pour l'instant. (Cette situation devrait changer lorsque `debian-installer` sera introduit dans Sarge).

Installez Windows. La configuration matérielle peut être obtenue en faisant un clic droit sur « Poste de travail » pour aller sur Propriétés / Gestionnaire de périphériques. Enregistrez toutes les ressources d'information comme IRQ, I/O port address, DMA. Quelques vieilles cartes ISA doivent être configurées sous DOS et utilisées en conséquence.

3.1.5 Le Mythe Lilo

« Lilo est limité à 1024 cylindres. » FAUX !

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

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

3.1.6 GRUB

Le nouveau gestionnaire de démarrage `grub` du projet GNU Hurd peut être installé sur un système Debian Woody :

```
# apt-get update
# apt-get install grub-doc
# mc /usr/share/doc/grub-doc/html/
... lisez le contenu
# apt-get install grub
# pager /usr/share/doc/grub/README.Debian.gz
... à lire :)
```

Pour modifier le menu de GRUB, éditez `/boot/grub/menu.lst`. Voir 'Comment configurer les paramètres de démarrage de GRUB' page 96 pour la configuration des paramètres de démarrage car la syntaxe est différente de celle de `lilo`.

3.1.7 Choix des disquettes de boot

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

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

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

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

3.1.8 Installation

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

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

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

Voici ce que je choisis lors du processus d'installation.
- MD5 passwords « yes »
- shadow passwords « yes »
- Installation « advanced » (dselect **) et sélection de
  - exclure emacs (si sélectionné), nvi, tex, telnet, talk(d) ;
  - inclure mc, vim, nano-tiny ou elvis-tiny. Voir '`dselect`' page 68. Même si vous êtes un fan d'Emacs, laissez-le pour le moment et utilisez nano pendant l'installation. De même, évitez d'installer de gros paquets comme TeX (Potato le faisait) à ce point-là. Voir 'Editeurs de sauvetage' page 195 pour la raison de l'installation de nano-tiny ou elvis-tiny ici.
- Pour toutes les questions de configuration pendant l'installation de chaque paquet : « y » (remplace version courante).
- `exim` : sélectionner 2 pour envoyer le courrier électronique par le serveur SMTP du FAI.

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

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

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

```
Internet
   |
   +--- FAI externe fournit le service POP (accédé par fetchmail)
   |
  le FAI fournit les services DHCP et relai SMTP
   |                    :
  Modem Cable        (Téléphone)
   |                    :
Port externe de la passerelle : eth0 (IP donnée par le DHCP du FAI)
 utilise un vieux PC portable (IBM Thinkpad, 486 DX2 50 MHz, 20 MB RAM)
 tourne avec le noyau Linux 2.4 et le système de fichiers ext3
 tourne avec le paquet « ipmasq » (avec fortification, NAT et pare-feu)
 tourne avec le paquet « dhcp-client » configuré pour eth0
         (remplace les paramètres DNS)
 tourne avec le paquet « dhcp » configuré pour eth1
 tourne avec « exim » comme smarthost (mode 2)
 tourne avec « fetchmail » avec un long intervalle (fallback)
 tourne avec « bind » comme serveur de nom en cache pour l'Internet
                              depuis le réseau local
```

```
                      comme serveur de nom officiel pour le domaine
        du réseau local
  tourne avec « ssh » sur les ports 22 et 8080
          (connexions depuis n'importe où)
  tourne avec « squid » comme serveur de cache pour
          l'archive Debian (pour APT)
 Port interne de la passerelle : eth1 (IP = 192.168.1.1, fixe)
                          |
          +--- LAN Switch (100 base T) ---+
          |                               |
 Quelques clients avec IP fixe    Quelques clients DHCP
 (IP = 192.168.1.2-127, fixe)     (IP = 192.168.1.128-200, dynamique)
```

Voir 'Configuration réseau' page 167 pour les détails de la configuration du réseau. Voir 'Configurer une passerelle' page 190 pour les détails de la configuration d'une passerelle pour le réseau local.

3.1.10 Comptes utilisateurs

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

Je crée toujours un premier compte utilisateur avec un nom comme "admin" (uid=1000). Tous les messages de root sont redirigés dessus. Ce compte est membre du groupe `adm` (voir 'Pourquoi `GNU su` ne supporte pas le groupe `wheel`' page 124), à qui l'on peut donner beaucoup des privilèges de root au travers des programmes `su` en utilisant PAM ou `sudo`. Voir 'Ajouter un compte utilisateur' page 47 pour les détails.

3.1.11 Création des systèmes de fichiers

Partition du disque dur

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

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

La taille du répertoire `/usr` dépend beaucoup des applications X Window et de la documentation. `/usr` peut être de 300Mo si l'on n'utilise que le terminal, tandis que 2Go–3Go n'est pas inhabituel si l'on installe un certain nombre d'applications Gnome. Lorsque `/usr` devient trop gros, déplacer `/usr/share/` vers une autre partition est le meilleur remède. Avec un noyau 2.4 récent, / peut avoir besoin de plus de 200Mo.

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

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

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

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

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

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

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

Monter les systèmes de fichiers

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

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

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

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

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

Pour une machine sous Windows connectée avec Samba (smbfs), `rw,auto,soft,intr` peut être une bonne idée. Voir 'Configuration Samba' page 38.

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

Montage autofs

Points clés pour le montage automatique :

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

```
# modprobe vfat # avant d'essayer d'accéder à la disquette
 ... ou pour automatiser cela,
# echo "vfat" >> /etc/modules
```

```
        ... et redémarrer le système
```

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

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

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

Montage NFS

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

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

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

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

3.1.12 Lignes directrices pour la mémoire DRAM

Ce qui suit sont des indications grossières pour la DRAM.

```
   4 Mo :  Minimum suffisant pour faire fonctionner le noyau Linux.
  16 Mo :  Minimum pour un usage du système en mode console.
  32 Mo :  Minimum pour un système X simple.
  64 Mo :  Minimum pour un système X avec GNOME/KDE.
 128 Mo :  Confortable pour le système X avec GNOME/KDE.
 256+Mo :  Pourquoi pas si vous le pouvez. La DRAM est bon marché.
```

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

3.1.13 Espace de Swap

J'utilise la ligne directrice suivante :

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

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

3.2 Configuration de Bash

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

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

Voir détails dans mon exemple. (`http://www.debian.org/doc/manuals/debian-reference/examples/`) J'aime les systèmes transparents, j'ai donc paramétré umask à 002 ou 022.

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

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

3.3 Configuration de la souris

3.3.1 Souris PS/2

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

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

Ici, un lien symbolique `/dev/mouse` est créé et pointe vers `/dev/gpmdata` pour accomoder certains utilitaires de configuration et faciliter la reconfiguration. (Par exemple, si vous décidez de ne pas utiliser `gpm`, pointez `/dev/mouse` vers `/dev/psaux` après avoir supprimé le démon `gpm`.)

Cette succession de signaux permet au clavier ou à la souris d'être déconnecté et réinitialisé en redémarrant `gpm` après reconnexion. X restera activé !

Le protocole du signal entre la sortie de `gpm` et l'entrée de X peut être implémenté de deux façons, soit "ms3" (protocole des souris séries Microsoft à 3 boutons) soit "brut" (protocole de la souris connectée), et ce choix impose le choix du protocol utilisé pour configurer X.

Voici des exemples de configuration pour les souris Logitech 3 boutons PS/2,(souris Unix traditionnelle).

Si vous faites partie de ceux dont la carte vidéo n'est pas supportée par le nouveau X4 et devez utiliser X3 (cartes ATI 64 bit par exemple), configurez `/etc/X11/XF86Config` au lieu de `/etc/X11/XF86Config-4` dans les exemples suivants.

Le protocole ms3

```
/etc/gpm.conf             | /etc/X11/XF86Config-4
==========================+=======================================
device=/dev/psaux         | Section "InputDevice"
responsiveness=           |  Identifier "Configured Mouse"
repeat_type=ms3           |  Driver     "mouse"
type=autops2              |  Option     "CorePointer"
append=""                 |  Option     "Device"   "/dev/mouse"
sample_rate=              |  Option     "Protocol" "IntelliMouse"
                          | EndSection
```

Si cette approche est utilisée, l'ajustement du type de souris est fait simplement en éditant le fichier `gpm.conf` et la configuration de X reste la même. Voir mes scripts d'exemple (`http://www.debian.org/doc/manuals/debian-reference/examples/`).

Le protocole brut

```
/etc/gpm.conf             | /etc/X11/XF86Config-4
==========================+=======================================
device=/dev/psaux         | Section "InputDevice"
responsiveness=           |  Identifier "Configured Mouse"
repeat_type=raw           |  Driver     "mouse"
type=autops2              |  Option     "CorePointer"
append=""                 |  Option     "Device"   "/dev/mouse"
```

```
sample_rate=               |  Option     "Protocol" "MouseManPlusPS/2"
                           | EndSection
```

Si vous utilisez cette approche, l'ajustement du type de souris est fait en éditant le fichier `gpm.conf` et en ajustant la configuration de X.

Comment adapter différentes souris

Le type de périphérique `autops2` est supposé détecter la plupart des souris PS/2 du marché. Malheureusement, cela ne marche pas toujours et il n'est pas disponibles dans les versions antérieures à Woody. Essayez d'utiliser `ps2`, ou `imps2`, dans le fichier `gpm.conf` à la place de `autops2` dans ces cas-là. Pour trouver les types de souris disponibles, tapez `gpm -t help`. Voir `gpm(8)`.

Si une souris PS/2 à 2 boutons est utilisée, activez `Emulate3Buttons` dans la configuration du protocole X. La différence de protocole entre les souris 2 boutons et 3 boutons est détectée automatiquement et ajustée par `gpm` après avoir cliqué sur le bouton du milieu.

Pour le protocole X avec 'Le protocole brut' page précédente ou sans `gpm`, utilisez :
- `IntelliMouse` : souris sur port série (répéteur `gpm` avec « ms3 »)
- `PS/2` : souris sur port PS/2 (à tester en premier)
- `IMPS/2` : n'importe quelle souris PS/2 (2, 3, ou à roulette)
- `MouseManPlusPS/2` : souris PS/2 Logitech
- ...

Pour plus de détails, consultez Mouse Support in XFree86 (`http://www.xfree86.org/current/mouse.html`).

Pour une souris à roulette Microsoft typique, la meilleure configuration est :

```
/etc/gpm.conf              | /etc/X11/XF86Config-4
===========================+=======================================
device=/dev/psaux          | Section "InputDevice"
responsiveness=            |  Identifier "Configured Mouse"
repeat_type=raw            |  Driver     "mouse"
type=autops2               |  Option     "CorePointer"
append=""                  |  Option     "Device"   "/dev/mouse"
sample_rate=               |  Option     "Protocol" "IMPS/2"
                           |  Option     "Buttons" "5"
                           |  Option  "ZAxisMapping" "4 5"
                           | EndSection
```

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

3.3.2 Souris USB

Assurez vous d'avoir toutes les fonctions du noyaux activées à la compilation :
- Sous « Input core support » :
  - « Input Core Support » (CONFIG_INPUT, `input.o`),
  - « Mouse Support » (CONFIG_INPUT_MOUSEDEV, `mousedev.o`),
- Sous « : USB support » :
  - « Support for USB » (CONFIG_USB, `usbcore.o`),
  - « Preliminary USB device filesystem » (CONFIG_USB_DEVICEFS),
  - « UHCI » ou « OHCI » (CONFIG_USB_UHCI || CONFIG_USB_UHCI_ALT || CONFIG_USB_OHCI, `usb-uhci.o` || `uhci.o` || `usb-ohci.o`),
  - « USB Human Interface Device (full HID) support » (CONFIG_USB_HID, `hid.o`), et
  - « HID input layer support » (CONFIG_USB_HIDINPUT)

Les noms de fichiers sont les noms des modules.

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

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

Pour les souris **USB** à roulette typiques, la configuration doit être :

```
/etc/gpm.conf             | /etc/X11/XF86Config-4
==========================+=======================================
device=/dev/input/mice    | Section "InputDevice"
responsiveness=           |  Identifier "Generic Mouse"
repeat_type=raw           |  Driver     "mouse"
type=autops2              |  Option     "SendCoreEvents" "true"
append=""                 |  Option     "Device"   "/dev/input/mice"
sample_rate=              |  Option     "Protocol" "IMPS/2"
                          |  Option     "Buttons" "5"
                          |  Option  "ZAxisMapping" "4 5"
                          | EndSection
```

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

3.3.3 Touchpad

Bien que le touchpad d'un portable émule une souris PS/2 à 2 boutons par défaut, le paquet `tpconfig` permet le contrôle complet du périphérique. Par exemple, en mettant `OPTIONS="--tapmode=0"` dans `/etc/default/tpconfig` on peut désactiver le « clic en tapant ». Configurez `/etc/gpm.conf` comme suit pour utiliser à la fois le touchpad et une souris USB externe sur la console :

```
device=/dev/psaux
responsiveness=
repeat_type=ms3
type=autops2
append="-M -m /dev/input/mice -t autops2"
sample_rate=
```

3.4 Configuration NFS

Configurer NFS dans `/etc/exports`.

```
# apt-get install nfs-kernel-server
# echo "/ *.domainname-for-lan-hosts(rw,no_root_squash,nohide)" \
        >> /etc/exports
```

Voir détails dans mon exemple. (`http://www.debian.org/doc/manuals/debian-reference/examples/`).

3.5 Configuration Samba

Références :
- `http://www.samba.org/`
- paquet `samba-doc`

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

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

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

Voir détails dans mon script d'exemple. (`http://www.debian.org/doc/manuals/debian-reference/examples/`)

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

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

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

La signification du rang de l'OS est donnée ci-dessous. Plus il est grand, plus la priorité du serveur est haute.

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

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

3.6 Configuration de l'imprimante

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

3.6.1 `lpr/lpd`

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

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

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

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

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

- `lpr` et `magicfilter`
- `lprng` et `lprngtool`
- `lprng` et `apsfilter`
- `lprng` et `magicfilter`

Pour exécuter des outils de configuration graphiques comme `printtool`, voir 'Devenir root sous X' page 142 pour obtenir les privilèges de root. Les queues d'imprimante créées avec `printtool` utilisent `gs` et agissent comme des imprimantes PostScript. Donc, lorsque vous y accédez, utilisez des pilotes PostScript. Du côté de Windows, « Apple LaserWriter » est le standard.

3.6.2 CUPS™

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

```
# apt-get install cupsys cupsys-bsd cupsys-client cupsys-driver-gimpprint
# apt-get install foomatic-db-engine foomatic-db-hpijs
# apt-get install foomatic-filters-ppds foomatic-gui
```

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

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

Par exemple, pour ajouter votre imprimante sur un port de la liste des imprimantes accessibles :

- cliquez sur « Printers » sur la page principale, puis « Add Printer »,
- entrez « root » comme nom d'utilisateur et son mot de passe,
- ajoutez l'imprimante en suivant les instructions,
- retournez sur la page « Printers » et cliquez sur « Configure Printers », et
- configurez la taille du papier, la résolution et les autres paramètres.

Pour plus d'informations, consultez `http://localhost:631/documentation.html` et `http://www.cups.org/cups-help.html`.

3.7 Autres conseils de configuration de l'hôte

3.7.1 Installer quelques paquets supplémentaires après l'installation

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

- Lancez `tasksel`. Voir 'Installation de tâches' page 67.
  Vous pouvez choisir ces options si vous en avez besoin :

  - Utilisateur – X window system
  - Développement – C et C++
  - Développement – Python
  - Développement – Tcl/Tk
  - Divers – environnement TeX/LaTeX
  - Pour les autres, je préfère utiliser `tasksel` comme guide en regardant les composants listés dans « Task Info » et en les installant manuellement avec `dselect`.

- Lancez `dselect`.

  Là, la première chose à faire est de sélectionner votre éditeur de texte favori, et tout programme dont vous avez besoin. Vous pouvez installer plusieurs variantes d'Emacs en même temps. Voir '`dselect`' page 68 et 'Editeurs populaires' page 195.

  De plus, vous pouvez remplacer des paquets par défaut avec d'autres ayant plus d'options.

  - ...

- ...

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

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

3.7.2 Modules

Les modules pour les pilotes de périphériques sont configurés lors de l'installation initiale. `modconf` permet de configurer les modules ensuite au travers d'une interface utilisant des menus. Ce programme est utile lorsque des modules ont été oubliés lors de l'installation ou lorsqu'un nouveau noyau est installé.

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

De plus, assurez-vous d'ajouter quelques lignes dans `/etc/modules` pour gérer l'IP masquerading (FTP, etc.) sur les noyaux 2.4. Voir 'Le noyau 2.4 modulaire' page 87, et notamment 'Fonctions réseau' page 88.

3.7.3 Configuration de base d'un graveur de CD

Éditer les fichiers suivants :

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

Voir 'Graveurs de CD' page 127 pour les détails.

3.7.4 Grande capacité mémoire et arrêt automatique

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

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

Lancer `lilo` pour installer cette configuration. `apm=power-off` est requis pour un noyau multiprocesseur (SMP) et `noapic` est nécessaire pour éviter les problèmes sur mon matériel bogué. La même chose peut être faite directement à l'invite de démarrage. Voir 'Autres astuces avec l'invite de démarrage' page 95.

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

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

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

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

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

```
ospm_busmgr
ospm_system
```

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

3.7.5 Étranges problèmes d'accès à certains sites web

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

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

Pour le désactiver, utilisez

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

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

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

3.7.6 Configuration d'une connexion RTC avec PPP

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

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

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

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

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

D'une autre façon, le paquet `wvdial` peut être utilisé pour configurer l'accès PPP. Veuillez noter qu'il y a un bogue bien connu `http://bugs.debian.org/82095`, qui empêche parfois l'appel pour les utilisateurs autres que root.

Tous les programmes de connexion appellent le démon `pppd`, qui exécute les scripts de `/etc/ppp/ip-up.d/` ou `/etc/ppp/ip-down.d/` après la connexion ou la déconnexion. On peut l'utiliser pour récupérer et envoyer des courriels.

3.7.7 Autre configuration à vérifier dans `/etc/`

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

Chapitre 4

Didacticiels Debian

Cette section propose une orientation basique dans le monde Debian pour les débutants. Si vous utilisez un système de type Unix depuis un certain temps déjà, vous connaissez probablement déjà tout ce qui est expliqué ici. Utilisez-la alors en cas de doute.

4.1 Débuter

Après l'installation du système Debian sur votre PC, vous devez apprendre quelques trucs pour le rendre utile. Voici un apprentissage express.

4.1.1 Se connecter à un shell en tant que root

Après redémarrage du système, il vous présentera soit l'écran de connexion graphique, soit l'écran de connexion en mode texte, selon votre sélection initiale de paquets. Pour simplifier, si c'est l'écran de connexion graphique qui vous est présenté, appuyez sur Ctrl-Alt-F1 [1] pour avoir l'écran de connexion en mode texte.

Supposons que le nom de votre machine est *`foo`*, alors l'invite de connexion ressemble à cela :

```
foo login:
```

Tapez `root`, appuyez sur la touche Entrée et tapez le mot de passe que vous avez choisi lors de l'installation. Sur un système Debian, selon la tradition Unix, le mot de passe est sensible à la casse. Ensuite, le système donne un message d'accueil, vous présente l'invite de commande root et attend vos commandes. [2]

```
foo login: root
```

---

[1] Les touches Ctrl gauche, Alt gauche et F1 sont appuyées simultanément.

[2] Notez que si vous avez édité le message d'accueil dans `/etc/motd`, cela sera différent.

```
Password:
Last login: Sun Oct 26 19:04:09 2003 on tty3
Linux foo 2.4.22-1-686 #6 Sat Oct 4 14:09:08 EST 2003 i686
GNU/Linux

Most of the programs included with the Debian GNU/Linux system are
freely redistributable; the exact distribution terms for each program
are described in the individual files in /usr/share/doc/*/copyright

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

root@foo:root#
```

Vous êtes prêt à administrer le système depuis cette invite de commande root. Le compte root est aussi appelé compte super-utilisateur ou utilisateur privilégié. Depuis ce compte, vous pouvez tout faire :

- lire, écrire et effacer n'importe quel fichier du système quelles que soient ses permissions
- configurer le propriétaire et les permissions de n'importe quel fichier du système
- configurer le mot de passe de n'importe quel utilisateur non-privilégié du système
- vous connecter à n'importe quel compte sans mot de passe

Partager l'accès au compte root en donnant son mot de passe est une très mauvaise idée. Utiliser des programmes comme `sudo(8)` est le bon moyen de partager des privilèges administratifs.

Notez qu'il est considéré comme une bonne habitude Unix de se connecter d'abord avec un compte non-privilégié même lorsque vous comptez administrer le système. Utilisez des commandes comme `sudo`, `super`, ou `su -c` pour obtenir un accès limité aux privilèges dont vous avez besoin. Voir 'Travailler de façon sûre – `sudo`' page 125. [3]

4.1.2 Configurer un système minimal pour débutant

Je crois qu'apprendre à utiliser un système informatique est comme apprendre une langue étrangère. Bien que les didacticiels soient utiles, il faut pratiquer avec des outils d'aide. Dans ce contexte, je crois qu'il est bon d'installer quelques paquets comme `mc`, `vim`, `lynx`, `doc-linux-text` et `debian-policy`. [4]

```
# apt-get update
 ...
# apt-get install mc vim lynx doc-linux-text debian-policy
 ...
```

---

[3] J'admets que j'ai utilisé le compte root plus de fois que nécessaire juste parce que c'est facile et que je suis négligeant.

[4] Il peut aussi être bon d'installer `gpm`, `emacs21` et `doc-linux-html`. Voir 'Configuration de la souris' page 34 et 'Editeurs' page 195.

Si ces paquets sont déjà installés, rien ne sera installé.

4.1.3 Ajouter un compte utilisateur

Pendant l'installation, vous avez normalement créé un compte utilisateur non-privilégié qui reçoit les e-mails envoyés au compte root. [5] Puisque vous ne voulez pas utiliser ce compte utilisateur pour les activités d'apprentissage suivantes, vous devriez créer un autre compte utilisateur.

Supposons que ce nouvel utilisateur est *penguin*, tapez alors :

```
root@foo:root# adduser penguin
... répondez à toutes les questions
```

le créera. [6] Avant d'aller plus loin, apprenons quelques trucs d'abord.

4.1.4 Changer de console virtuelle

Dans le système Debian par défaut, il y a six pseudo-terminaux indépendants disponibles, c'est-à-dire que vous pouvez utiliser l'écran du PC comme six terminaux VT-100. Changez de l'un à l'autre en appuyant sur la touche Alt gauche et l'une des touches F1–F6 simultanément. Chaque pseudo-terminal permet de se connecter indépendamment sur un compte. L'environnement multi-utilisateur est l'une des grandes caractéristiques d'Unix et l'on en prend l'habitude.

Si vous appuyez accidentellement sur Alt-F7 sur un système qui exécute X Window System et que s'affiche un écran graphique, la console en mode texte peut être obtenue en appuyant sur Ctrl-Alt-F1. Changez de console et revenez à la première pour vous habituer à ces commandes.

4.1.5 Comment arrêter le système

Comme tout autre système d'exploitation moderne où les fichiers sont mis en cache en mémoire, Debian a besoin d'être arrêté proprement avant que le courant ne soit coupé sans danger pour les fichiers. Voici la commande pour arrêter le système depuis l'invite de commande root :

```
# shutdown -h now
```

---

[5] J'appelle généralement ce compte `admin`, mais cela peut être n'importe quel nom.

[6] Vous voudrez peut-être ajouter aussi cet utilisateur *penguin* aux groupe `adm` pour lui permettre de lire les fichiers du journal dans `/var/log/`. Voir `passwd(5)`, `group(5)`, `shadow(5)`, `group(5)`, `vipw(8)` et `vigr(8)`. Pour la signification officielle des utilisateurs et groupes, consultez une version récente du document Users and Groups (`/usr/share/doc/base-passwd/users-and-groups.html`).

Ceci est pour le mode multi-utilisateur normal. Si vous êtes en mode mono-utilisateur, utilisez la commande suivant depuis l'invite de commande root :

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

Autrement, vous pouvez taper Ctrl-Alt-Suppr pour arrêter le système. [7]

Attendez jusqu'à ce que le système vous affiche « System halted » (NdT : système arrêté), puis coupez le courant. Si APM ou ACPI est activé dans le BIOS et sous Linux, le système s'éteindra puis coupera l'alimentation par lui même. Voir 'Grande capacité mémoire et arrêt automatique' page 42 pour plus de détails.

4.1.6 À vous de jouer

Maintenant vous êtes prêt à jouer avec le système Debian sans risque tant que vous utilisez ce compte utilisateur non-privilégié `penguin`. [8] Connectons-nous au compte `penguin`. Si vous êtes à l'invite de commande root, tapez Ctrl-D [9] pour fermer le compte root et retourner à l'invite de connexion. Entrez le nouveau nom d'utilisateur `penguin` et son mot de passe. [10] L'invite de commande suivante sera affichée :

```
penguin@foo:penguin$
```

À partir de maintenant, les exemples seront donnés avec une invite de commande simplifiée. J'utiliserai :

- `#` : invite de commande root
- `$` : invite de commande d'utilisateur non-privilégié

Nous allons démarrer l'apprentissage du système Debian de la façon facile 'Midnight Commander (MC)' page suivante puis plus tard de la façon normale 'Environnement de travail de type Unix' page 52.

---

[7] Les touches Ctrl gauche, Alt gauche et Suppr sont appuyées simultanément depuis la console. Dans le système par défaut, cela fera redémarrer le système. Vous devez modifier `/etc/inittab` pour avoir la commande `shutdown` avec l'option `-h` comme décrit dans 'Installer quelques paquets supplémentaires après l'installation' page 40.

[8] C'est parce que le système Debian, même juste après l'installation par défaut, est configuré avec les bonnes permissions sur les fichiers, ce qui empêche un utilisateur non-privilégié d'altérer le système. Bien sûr, il peut encore y avoir quelques trous qui peuvent êtres exploités, mais ceux qui s'inquiètent de cela ne devraient pas lire cette section mais Securing Debian Manual (`http://www.debian.org/doc/manuals/securing-debian-howto/`) à la place.

[9] Les touches Ctrl gauche et D sont appuyées simultanément. Pas besoin de presser la touche Shift bien que ces caractères de contrôle soient donnés avec une majuscule.

[10] Si vous tapez `root` et son mot de passe à la place de `penguin`, vous accéderez au compte root. Cette procédure sera nécessaire pour réaccéder au compte root.

4.2 Midnight Commander (MC)

Midnight Commander (MC) est le « couteau suisse » GNU pour la console Linux et autres environnements de terminaux. Il donne aux débutants une expérience de la console avec des menus qui sont plus facile à apprendre que les commandes Unix standard.

Utilisez cette commande pour explorer le système Debian. C'est le meilleur moyen d'apprendre. Explorez ces quelques endroits clé en utilisant les touches du curseur et Entrée :

- `/etc` et ses sous-répertoires.
- `/var/log` et ses sous-répertoires.
- `/usr/share/doc` et ses sous-répertoires.
- `/sbin` et `/bin`

4.2.1 Améliorer MC

Afin que MC change de répertoire de travail lorsqu'on le quitte, il faut éditer `~/.bashrc` (ou `/etc/bash.bashrc`, appelé depuis `.bashrc`), comme détaillé dans sa page de manuel, `mc(1)`, dans l'option `-P`. [11]

4.2.2 Démarrer MC

```
$ mc
```

Toutes les opérations sur fichiers peuvent être effectuées depuis les menus de MC, moyennant un effort minimal de la part de l'utilisateur. Appuyez sur F1 pour obtenir l'écran d'aide. Vous pouvez jouer avec MC en utilisant simplement les touches du curseur et les touches de fonction. [12]

4.2.3 Gestionnaire de fichiers de MC

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

- F1 : Menu d'aide
- F3 : Afficheur de fichier interne
- F4 : Éditeur interne
- F9 : Active le menu déroulant
- F10 : Quitte Midnight Commander

---

[11] Si vous ne comprenez pas ce que je dis ici, vous pouvez le faire plus tard.

[12] Si l'on utilise un terminal comme `kon` et `kterm` pour le Japonais qui a des problèmes avec certains caractères graphiques, on peut ajouter `-a` à la ligne de commande de MC pour ne pas avoir de problème.

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

4.2.4 Astuces en ligne de commande avec MC

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

4.2.5 Éditeur de MC

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

Cet éditeur peut charger un fichier au démarrage :

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

Il ne s'agit pas d'un éditeur multi-fenêtres mais on peut faire usage de plusieurs consoles Linux pour atteindre le même effet. Pour copier d'une fenêtre sur l'autre, utilisez les touches Alt-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 d'un fichier dans un autre fichier.

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

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

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

Je recommande vraiment de positionner ces variables à `vim` si c'est possible. Utiliser fréquemment les commandes de `vim` est une bonne chose à faire, puisqu'elles sont toujours présentes dans le monde Linux/Unix. [13]

4.2.6 Visionneur de MC

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

```
$ mc -v fichier _à_voir
```

4.2.7 Possibilités de démarrage automatique de MC

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

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

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

4.2.8 Système de fichiers FTP virtuel de MC

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

Essayez `http.us.debian.org/debian` comme URL et parcourez l'archive de Debian. Voir 'Les archives Debian' page 5 pour son organisation.

---

[13] En fait, `vi` ou `nvi` sont les programmes que vous trouverez partout. J'ai choisi `vim` à la place pour les débutants parce qu'il offre de l'aide avec la touche F1, est similaire, et plus puissant. Si vous n'êtes pas à l'aise avec `vim`, vous pouvez continuer à utiliser `mcedit` pour la plupart des tâches de maintenance du système. Comme `mcedit` ne s'occupe pas du codage des textes, il a parfois des avantages lorsque l'on édite des fichiers dont le codage n'est pas connu. `mcedit` n'affiche pas les fichiers UTF-8 correctement.

4.3 Environnement de travail de type Unix

Bien que MC vous permette de faire presque tout, il est très important d'apprendre comment utiliser les outils en ligne de commande depuis l'invite du shell et de devenir familier avec l'environnement de travail de type Unix. [14]

4.3.1 Combinaisons de touches spéciales

Dans un environnement Unix, quelques combinaisons de touches ont une signification particulière. [15]

- Ctrl-U : Efface la ligne précédant le curseur.
- Ctrl-H : Efface le caractère précédant le curseur.
- Ctrl-D : Termine l'entrée. (ferme le shell si vous l'utilisez)
- Ctrl-C : Termine un programme en cours.
- Ctrl-Z : Arrête temporairement un programme. (le met en tâche de fond, voir '`command &`' page 58)
- Ctrl-S : Arrête l'affichage à l'écran. [16]
- Ctrl-Q : Réactive l'affichage à l'écran.

Le shell par défaut, `bash`, permet d'éditer l'historique des commandes et de compléter les commandes en appuyant sur la touche Tab pour aider son utilisation interactive.

- flèche haut : Commence une recherche dans l'historique.
- Ctrl-R : Commence une recherche dans l'historique incrémentale.
- TAB : Complète l'entrée du nom de fichier sur la ligne de commande.
- Ctrl-V TAB : Utilisation de la touche TAB sans complétion.

D'autres combinaisons de touches à se rappeler :

- Ctrl-Alt-Del : Redémarre/arrête le système, voir 'Installer quelques paquets supplémentaires après l'installation' page 40.
- Clic gauche et mouvement de la souris : Sélectionne et copie dans le presse-papier.
- Clic milieu : Copie le contenu du presse-papier.
- La touche Meta (terminologie Emacs) est traditionnellement associée à la touche Alt gauche. Certains systèmes peuvent être configurés pour utiliser la touche Windows.

Ici, afin d'utiliser une souris avec la console en mode texte, vous devez avoir `gpm` en démon. [17] Voir 'Configuration de la souris' page 34.

---

[14] Dans ce didacticiel, le shell est `bash`. Pour plus de détails sur les différents shells, voir 'Shell' page 211.

[15] Sur une console Unix normale, seuls les touches Ctrl et Alt de gauche ont le comportement attendu.

[16] Vous pouvez désactiver cette possibilité du terminal en utilisant `stty(1)`.

[17] Avec X Window System, les fonctions de la souris s'utilisent de la même façon avec `xterm`.

4.3.2 Commandes Unix de base

Apprenons les commandes Unix de base. [18] Essayez toutes les commandes suivantes depuis le compte non-privilégié `penguin` :

- `pwd`
  - Affiche le nom du répertoire courant.
- `whoami`
  - Affiche le nom de l'utilisateur courant.
- `file foo`
  - Affiche le type du fichier *foo*.
- `type -p commandname`
  - Affiche l'emplacement de la commande `commandname`.
  - `which commandname` fait la même chose. [19]
- `type commandname`
  - Affiche des informations sur la commande `commandname`.
- `apropos key-word`
  - Trouve les commandes ayant rapport à `key-word`.
  - `man -k key-word` fait la même chose.
- `whatis commandname`
  - Affiche une explication en une ligne sur la commande `commandname`.
- `man -a commandname`
  - Affiche une explication sur la commande `commandname`. (à la Unix)
- `info commandname`
  - Affiche une explication assez longue sur la commande `commandname`. (à la GNU)
- `ls`
  - Affiche le contenu du répertoire. (fichiers ne commençant pas par '.' et répertoires) [20]
- `ls -a`
  - Affiche le contenu du répertoire. (tous les fichiers et répertoires)
- `ls -A`
  - Affiche le contenu du répertoire. (presque tous les fichiers et répertoires, sauf `..` et `.`)
- `ls -la`
  - Affiche tout le contenu du répertoire de façon détaillée. Voir 'Concepts du système de fichiers de Debian' page 58.
- `ls -d`
  - Affiche tous les sous-répertoires du répertoire courant.
- `lsof foo`
  - Affiche l'état d'ouverture du fichier `foo`.
- `mkdir foo`

---

[18] Ici, j'utilise « Unix » dans son sens générique. Tous les clones d'Unix offrent des commandes équivalentes. Le système Debian ne fait pas exception. Ne vous inquiétez pas si certaines commandes ne fonctionnent pas comme vous le voudriez. Ces exemples ne sont pas fait pour être exécutés dans cet ordre.

[19] Si un `alias` est utilisé dans le shell, leurs sorties sont différentes.

[20] Unix cache par tradition les fichiers dont le nom commence par '.'. Ce sont habituellement des fichiers qui contiennent la configuration ou les préférences d'un utilisateur.

  - Crée un nouveau répertoire *`foo`* dans le répertoire courant.
- `rmdir` *`foo`*
  - Efface un répertoire *`foo`* du répertoire courant.
- `cd` *`foo`*
  - Change de répertoire et va dans le répertoire *`foo`* du répertoire courant ou dans le répertoire listé dans la variable `CDPATH`. Voir la commande `cd` dans `builtins(7)`.
- `cd /`
  - Va dans le répertoire racine.
- `cd`
  - Va dans le répertoire personnel de l'utilisateur courant.
- `cd /`*`foo`*
  - Va dans le répertoire `/`*`foo`*.
- `cd ..`
  - Va dans le répertoire parent.
- `cd ~`*`foo`*
  - Va dans le répertoire personnel de l'utilisateur *`foo`*.
- `cd -`
  - Va dans le répertoire précédent.
- `</etc/motd pager`
  - Affiche le contenu de `/etc/motd` avec l'afficheur par défaut. Voir '`command <` *`foo`*' page 58. [21]
- `touch` *`junkfile`*
  - Crée un fichier vide *`junkfile`*.
- `cp` *`foo bar`*
  - Copie un fichier existant *`foo`* vers un nouveau fichier *`bar`*.
- `rm` *`junkfile`*
  - Efface le fichier *`junkfile`*.
- `mv` *`foo bar`*
  - Renomme le fichier *`foo`* en *`bar`*.
- `mv` *`foo bar/baz`*
  - Déplace le fichier *`foo`* vers un nouvel emplacement avec le nouveau nom *`bar/baz`*. Le répertoire *`bar`* doit exister.
- `chmod 600` *`foo`*
  - Rend le fichier *`foo`* non lisible et non modifiable pour les autres utilisateurs. (non exécutable pour tous)
- `chmod 644` *`foo`*
  - Rend le fichier *`foo`* lisible mais non modifiable pour les autres utilisateurs. (non exécutable pour tous)
- `chmod 755` *`foo`*
  - Rend le fichier *`foo`* lisible mais non modifiable pour les autres utilisateurs. (exécutable pour tous)
- `top`
  - Affiche de l'information sur les processus en plein écran. Appuyez sur 'q' pour quit-

---

[21] L'afficheur par défaut sur un système Debian est `more` qui ne peut pas retourner en arrière. En installant le paquet `less` avec la commande `apt-get install less`, `less` devient l'afficheur par défaut et vous pouvez retourner en arrière avec les touches du curseur.

ter.

- `ps aux | pager`
  - Affiche les processus en cours en utilisant un affichage de type BSD. Voir '`command1 | command2`' page 58.
- `ps -ef | pager`
  - Affiche les processus en cours en utilisant un affichage de type Unix System-V.
- `ps aux | grep -e "[e]xim4*"`
  - Affiche les processus `exim` ou `exim4`. Apprenez les expressions rationnelles avec la page de manuel `grep(1)` en tapant `man grep`.[22]
- `ps axf | pager`
  - Affiche les processus en cours avec une sortie en ASCII art.
- `kill` *`1234`*
  - Tue un processus identifié par l'ID *1234*. Voir 'Tuer un processus' page 101.
- `grep -e "`*`pattern`*`" *.html`
  - Cherche *pattern* dans tous les fichiers finissant par `.html` du répertoire courant et les affiche tous.
- `gzip` *`foo`*
  - Compresse *`foo`* pour créer *`foo`*`.gz` avec le codage Lempel-Ziv (LZ77).
- `gunzip` *`foo`*`.gz`
  - Décompresse *`foo`*`.gz` pour créer *`foo`*.
- `bzip2` *`foo`*
  - Compresse *`foo`* pour créer *`foo`*`.bz2` avec l'algorithme de compression de texte Burrows-Wheeler, et le codage Huffman. (Meilleure compression que `gzip`)
- `bunzip2` *`foo`*`.bz2`
  - Décompresse *`foo`*`.bz2` pour créer *`foo`*.
- `tar -xvvf` *`foo.tar`*
  - Extrait les fichiers de l'archive *`foo`*`.tar`.
- `tar -xvvzf` *`foo`*`.tar.gz`
  - Extrait les fichiers de l'archive gzip *`foo`*`.tar.gz`.
- `tar -xvvf --bzip2` *`foo.tar.bz2`*
  - Extrait les fichiers de l'archive *`foo`*`.tar.bz2`.[23]
- `tar -cvvf` *`foo`*`.tar` *`bar/`*
  - Archive le contenu du répertoire *`bar`*`/` dans l'archive *`foo`*`.tar`.
- `tar -cvvzf` *`foo`*`.tar.gz` *`bar/`*
  - Archive le contenu du répertoire *`bar`*`/` dans l'archive compressée *`foo`*`.tar.gz`.
- `tar -cvvf --bzip2` *`foo`*`.tar.bz2` *`bar/`*
  - Archive le contenu du répertoire *`bar`*`/` dans l'archive *`foo`*`.tar.bz2`.[24]
- `zcat README.gz | pager`
  - Affiche le contenu du fichier compressé `README.gz` avec l'afficheur par défaut.
- `zcat README.gz > foo`

---

[22] `[` et `]` dans l'expression rationnelle permet à `grep` de ne pas se trouver. `4*` dans l'expression rationnelle signifie une répétition de 0 ou plusieurs caractères '4' et permet à `grep` de trouver à la fois `exim` et `exim4`. La signification de `*` dans le shell et dans les expressions rationnelles est différente.

[23] `--bzip2` est utilisé ici à la place de la nouvelle option courte `-j` pour assurer que cela fonctionne avec la vieille version de `tar` dans Potato.

[24] `--bzip2` est utilisé ici aussi pour des raisons de compatibilité.

  - Crée un fichier `foo` avec le contenu décompressé de `README.gz`.
- `zcat README.gz >> foo`
  - Ajoute le contenu décompressé de `README.gz` à la fin du fichier `foo`. (S'il n'existe pas, le crée d'abord.)
- `find . -name` *`pattern`*
  - Cherche les noms de fichiers contenant *`pattern`* avec le shell (plus lent).
- `locate -d .` *`pattern`*
  - Cherche les noms de fichiers contenant *`pattern`* (plus rapide, utilise une base de donnée générée régulièrement).

Parcourez les répertoires et jetez un coup d'oeil au système avec les commandes ci-dessus. Si vous avez des questions sur ces commandes de la console, consultez la page de manuel. Par exemple, ces commandes permettent de démarrer :

```
$ man man
$ man bash
$ man ls
```

C'est aussi le bon moment pour apprendre à utiliser `vim` en appuyant sur la touche F1. Vous devriez au moins lire les 35 premières lignes. Ensuite, faites le didacticiel en déplaçant le curseur sur `|tutor|` et en appuyant sur Ctrl-]. Voir 'Editeurs' page 195 pour en savoir plus sur les éditeurs.

Notez que beaucoup de commandes Unix, y compris celles de GNU et BSD, affichent une aide brève si vous les exécutez de l'une des façons suivantes (ou parfois sans argument) :

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

Essayez aussi les exemples de 'Astuces Debian' page 93 pour apprendre.

4.3.3 L'exécution de commandes

4.3.4 Commande simple

4.3.5 Exécution de commande et variables d'environnement

4.3.6 Chemin de recherche des commandes

4.3.7 Options de ligne de commande

4.3.8 Expressions de remplacement du shell

4.3.9 Valeur de retour d'une commande

4.3.10 Suites de commandes typiques

`command &`

`command1 | command2`

`command1 ; command2`

`command1 && command2`

`command1 || command2`

`command >` *`foo`*

`command >>` *`foo`*

`command >` *`foo`* `2>&1`

`command <` *`foo`*

4.3.11 Alias de commande

4.4 Traitement de texte à la Unix

4.4.1 Expressions rationnelles

4.4.2 Expressions de remplacement

4.5 Système de fichiers Unix

4.5.1 Bases sur les fichiers Unix

4.5.2 Concepts du système de fichiers de Debian

4.5.3 Permissions d'accès des fichiers et répertoires

4.5.4 Horodatage

4.5.5 Liens

4.6.1 Démarrer le système X Window

Le système X Window peut être démarré automatiquement avec un démon de connnexion comme `xdm` ou depuis la console en tapant :

```
$ exec startx
```

4.6.2 Menu du système X Window

Comme l'environnement X peut être utilisé avec différents gestionnaires de fenêtres, leurs interfaces utilisateur peuvent sensiblement varier. Rappelez-vous que cliquer avec le bouton droit sur la fenêtre principale permet d'afficher un menu. Celui-ci est toujours est disponible.

- Pour obtenir une invite de commande shell, exécutez Xterm depuis le menu :
  - "XShells" –> "XTerm".
- Pour la navigation graphique sur le web, exécutez Mozilla depuis le menu :
  - "Apps" –> "Net" –> "Mozilla Navigator".
- Pour l'affichage graphique de fichiers PDF, exécutez Xpdf depuis le menu :
  - "Apps" –> "Viewers" –> "Xpdf".

Si vous ne trouvez pas une entrée du menu, installez le paquet correspondant. Voir 'Bases de la gestion de paquets Debian' page 66.

4.6.3 Suites de commandes au clavier pour le système X Window

Les commandes au clavier suivantes sont importantes à se rappeler lorsque l'on est dans l'environnement X Window :

- Ctrl-Alt-F1 à F6 : Change vers un autre pseudo-terminal (depuis X Window, DOSEMU, etc.)
- Alt-F7 : Retourne sous W Window
- Ctrl-Alt-moins : Change la résolution de l'écran (la touche '-' est celle du clavier numérique)
- Ctrl-Alt-plus : Change la résolution de l'écran dans le sens opposé (la touche '+' est celle du clavier numérique)
- Ctrl-Alt-Backspace : Stoppe le serveur X
- Alt-X, Alt-C, Alt-V : Opérations de couper/copier/coller usuelles de Windows/MacOS où Ctrl est remplacé par Alt dans certaines applications comme Netscape Composer.

4.7 Étude approfondie

Je vous recommande de lire les guides-clé du The Linux Documentation Project : Guides (`http://www.tldp.org/guides.html`) :

- "The Linux System Administrators' Guide",

  - Il couvre tous les aspects permettant de garder un système en marche, gérer les comptes utilisateurs, les sauvegardes, la configuration du système.
  - paquet : `sysadmin-guide`
  - fichier : `/usr/share/doc/sysadmin-guide/html/index.html`
  - web : `http://www.tldp.org/LDP/sag/index.html`
- "The Linux Network Administrator's Guide, Second Edition",
  - C'est une référence unique pour l'administration réseau dans un environnement Linux.
  - paquet : `(not available)`
  - fichier : `(notapplicable)`
  - web : `http://www.tldp.org/LDP/nag2/index.html`
- "Linux : Rute User's Tutorial and Exposition"
  - Un bon livre avec versions en ligne et papier couvrant l'administration système GNU/Linux.
  - Par Paul Sheer
  - Publié by Prentice Hall
  - Paquet : `rutebook` (from `non-free`)
  - Fichier : `/usr/share/doc/rutebook/`

Voir 'Support Debian' page 233 pour plus de ressources.

Chapitre 5

Mise à jour d'une distribution vers `testing`

Ce chapitre est basé sur une version plus ancienne de texte original anglais. Veuillez vérifier la version anglaise aussi.

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

Le processus de mise à jour d'un système vers la version `testing/unstable` est le suivant :

- mettre à jour le système APT vers la version de Woody si vore système est Potato, car l'APT de Potato ne possédait pas les capacités décrites dans `apt_preferences(5)` de Woody.
- modifier vos fichiers `/etc/apt/sources.list` et `/etc/apt/preferences`, pour ajouter des références vers la section « testing » des dépôts que vous utilisez. Si vous le souhaitez, vous pouvez aussi ajouter des références vers la section « unstable ».
- mettre à jour vos listes de paquets, et installer les paquets mis à jour.

5.1 Transition de APT vers la version de Woody

Une mise à jour par le réseau du système APT et de certains paquets principaux vers la version de Woody peut être effectuée comme suit après avoir inclus une source `stable` dans `/etc/apt/sources.list` si vous utilisez toujours Potato.

```
# apt-get update
# apt-get install libc6 perl libdb2 debconf
# apt-get install apt apt-utils dselect dpkg
```

5.2 Préparation de la transition (« stable » vers « testing »)

Suivre la version `testing` de Debian a l'effet de bord de recevoir très lentement les correctifs de sécurité. Vous êtes prévenus.

Une mise à jour via le réseau vers « testing » peut être effectuée comme suit (lancer le script suivant go-woody (`http://www.debian.org/doc/manuals/debian-reference/examples/`) pour le faire en une seule commande) :

Videz le fichier `/etc/apt/sources.list` existant

```
# cd /etc/apt
# cp -f sources.list sources.old
# :>sources.list
```

Récupérez une liste propre de dépôts, pour « stable »

```
# cd /
# apt-setup noprobe
 ... sélectionner http ou ftp
```

Ajoutez la section « testing » à cette nouvelle liste. Les lignes `deb-src` sont commentées.

```
# cd /etc/apt
# grep -e "^deb " sources.list             >sources.deb
# grep -e "^deb-" sources.list             >sources.src
# sed -e "s/stable/testing/"  sources.deb \
     >>sources.list
# sed -e "s/stable/testing/" sources.src | \
     sed -e "s/^deb-/#deb-/"  >>sources.list

# apt-get update
# apt-get install apt apt-utils
# cat >preferences <<EOF
 Package: *
 Pin: release a=testing
 Pin-Priority: 600

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

 EOF
```

Eventuellement, ajoutez la section « unstable » des archives.

```
# sed -e "s/stable/unstable/" sources.deb \
  >>sources.list
# sed -e "s/stable/unstable/" sources.src | \
 sed -e "s/^deb-/#deb-/"  >>sources.list
```

Voir 'Bases de la gestion de paquets Debian' page 66 pour configurer `/etc/apt/sources.list` et `/etc/apt/preferences`.

Vous pouvez maintenant mettre à jour, en utilisant une des méthodes décrites dans la section suivante.

5.3 Mise à jour du système Debian

Après avoir configuré correctement `/etc/apt/sources.list` et `/etc/apt/preferences`, le système peut être mis à jour vers `testing`. Voir 'Gestion des paquets Debian' page 65 pour les bases, et 'Résolution de problèmes de mise à jour avec APT' page 71 si vous rencontrez des problèmes.

5.3.1 Meilleure méthode de mise à jour avec `dselect`

Si le système comprend beaucoup de paquets `-dev`, etc., la méthode suivante avec `dselect` est recommandée pour un contrôle fin des paquets.

```
# dselect update # à faire avant chaque mise à jour
# dselect select # sélectionne des paquets additionnels
```

Tous vos paquets actuels seront sélectionnés lorsque `dselect` démarre. `dselect` peut vous proposer d'autres paquets selon les priorités `depends`, `suggests` et `recommends`. Si vous ne voulez pas ajouter de paquet, tapez simplement `Q` pour sortir de `dselect`.

```
# dselect install
```

Vous devrez répondre à quelques questions de configuration des paquets pendant cette partie du processus, aussi ayez vos notes à portée de main et comptez un peu de temps pour cette partie. Voir '`dselect`' page 68.

Utilisez `dselect`. **Il fonctionne toujours :)**

5.3.2 Méthode obsolète de mise à jour avec `apt-get`

L'utilisation de `apt-get` décrite ci-dessous est répandue mais n'est *pas* recommandée pour des mises à jour du système. Si vous devez mettre à jour sans `dselect` après Woody, considérez l'utilisation d'`aptitude` ou d'autres options.

Si un système n'a pas beaucoup de paquets installés ou si l'archive Debian n'a pas de changements majeurs, ce qui suit peut être suffisant (parfois).

```
# apt-get update # à faire avant la mise à jour
 ... pour mettre à jour l'ensemble du système avec les sélections « depends »
# apt-get -u dist-upgrade # à faire avant la mise à jour
 ... pour mettre à jour le système entier avec les sélections « depends »
# apt-get -u dist-upgrade
 ... ou pour mettre à jour en gardant les sélections de dselect
     (meilleure solution)
# apt-get -u dselect-upgrade # utilise les sélections de dselect
```

Comme cette méthode de mise à jour utilise `apt-get`, la gestion des paquets *recommends* et *suggests* est limitée. Voir 'Dépendances des paquets' page 15.

Chapitre 6

Gestion des paquets Debian

`aptitude` est désormais le frontal en mode texte pour APT à utiliser de préférence. Il se rappelle des paquets que vous avez installés volontairement et quels paquets ont été ajoutés par dépendance ; ces derniers sont automatiquement désinstallés par `aptitude` lorsqu'ils ne sont plus nécessaires à un paquet installé volontairement. Il possède des fonctions avancées de filtrage des paquets mais elles peuvent être difficiles à configurer.

`synaptic` est désormais le frontal graphique GTK pour APT à utiliser de préférence. Ses fonctions de filtrage des paquets sont plus faciles à utiliser que celles d'`aptitude`. Il possède aussi un support expérimental pour les Debian Package Tags (`http://debtags.alioth.debian.org/`) (balises de paquets Debian).

Pour réduire la charge réseau des dépôts Debian et accélérer vos téléchargements, vous devriez récupérer vos paquets depuis les miroirs Debian.

Si vous devez installer le même paquet sur plusieurs machines de votre réseau local, vous pouvez configurer un proxy HTTP local en utilisant `squid` pour les paquets téléchargés par APT. Si nécessaire, positionnez la variable d'environnement `http_proxy` ou la variable `http` dans `/etc/apt/apt.conf`.

Bien que la fonction de *pinning* d'APT décrite dans `apt_preferences(5)` soit puissante, ses effets peuvent être difficiles à comprendre et à gérer. Vous devriez la considérer comme une Fonction Avancée.

L'utilisation de la méthode décrite dans '`chroot`' page 116 est utile pour simultanément assurer à la fois la stabilité du système et l'accès aux dernières versions des programmes.

Ce chapitre est basé sur un système d'après Woody. Certaines fonctions peuvent nécessiter un système Sarge, ou plus récent.

6.1 Introduction

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

6.1.1 Outils de gestion des paquets principaux

```
dpkg      - installation de paquets Debian
apt-get   - frontal pour APT en ligne de commande
aptitude  - frontal avancé pour APT en mode texte et ligne de commande
synaptic  - frontal pour APT en mode graphique GTK
dselect   - gestion des paquets à l'aide de menus
tasksel   - installation de tâches
```

Ces outils ne sont pas tous des alternatives. Par exemple `dselect` utilise à la fois APT et `dpkg`.

APT utilise `/var/lib/apt/lists/*` pour suivre les paquets disponibles tandis que `dpkg` utilise `/var/lib/dpkg/available`. Si vous avez installé des paquets directement en utilisant `aptitude` ou un autre frontal pour APT et que vous voulez utiliser `dselect` pour installer des paquets, assurez-vous de mettre à jour le fichier `/var/lib/dpkg/available` en sélectionnant `[M]ise à jour` dans le menu de `dselect` (ou en exécutant `dselect update`).

`apt-get` récupère automatiquement les paquets dont un paquet demandé dépend. Il n'installe pas les paquets recommandés ou suggérés par le paquet demandé.

`aptitude` au contraire peut être configuré pour installé les paquets recommandés ou suggérés.

`dselect` présente à l'utilisateur une liste de paquets qu'un paquet sélectionné recommande ou suggère et permet de les sélectionner ou pas. Voir 'Dépendances des paquets' page 15.

6.1.2 Outils pratiques

```
dpkg-reconfigure  - reconfigure un paquet déjà installé
                    (s'il utilise debconf)
dpkg-source       - gère les paquets sources
dpkg-buildpackage - automatise la création d'un paquet
apt-cache         - recherche un paquet dans le cache local
```

6.2 Bases de la gestion de paquets Debian

6.2.1 Configurer APT

Configurer `sources.list` comme décrit dans 'Préparation de la transition (« stable » vers « testing »)' page 62. [1] Référez-vous aussi à 'Installation du système Debian' page 25, 'Mise à jour d'une distribution vers `testing`' page 61 and 'Editeurs de sauvetage' page 195.

---

[1] Si vous suivez `testing` ou `unstable` vous pouvez enlever les références à `stable` de `sources.list` et `/etc/apt/preferences` parce que `testing` démarre à partir d'une copie de `stable`.

6.2.2 Installation de tâches

Vous pouvez installer des ensembles de paquets habituellement nécessaires pour certaines utilisations d'un système Debian. Ces ensembles de paquets sont appelés « tâches ».

Le moyen le plus simple d'installer des tâches lors de l'installation initiale est d'utiliser `tasksel`. Notez que vous devez exécuter

```
dselect update
```

avant de l'utiliser.

`aptitude` permet aussi d'installer des tâches et c'est l'outil recommandé pour cela. Il permet de désélectionner des paquets dans les tâches avant de procéder à l'installation.

6.2.3 `aptitude`

`aptitude` est un programme d'installation de paquets par sélection dans des menus, similaire à `dselect` mais développé de zéro au-dessus d'APT. Il peut être utilisé comme alternative à `apt-get` pour la plupart des commandes. Voir `aptitude(1)` et `/usr/share/doc/aptitude/README`.

`aptitude` en plein écran accepte des commandes à une touche, généralement en minuscule.

Touche	Action
F10	Menu
?	Aide (listing complet)
u	Mise à jour des informations de l'archive de paquets
+	Marque un paquet pour mise-à-jour ou installation
-	Marque un paquet pour suppression (garde la configuration)
_	Marque un paquet pour purge (supprime la configuration)
=	Place un paquet en maintien
U	Marque tous les paquets qui peuvent être mis à jour
g	Téléchargement et installation des paquets sélectionnés
q	Sortie de l'écran courant et sauvegarde des changements
x	Sortie de l'écran courant sans sauvegarde
Enter	Visualisation d'information sur un paquet
C	Visualisation des changements d'un paquet
|	Change la limite des paquets affichés
/	Recherche la première occurence
\	Répète la dernière recherche

Comme `apt-get`, `aptitude` installe les dépendances d'un paquet demandé. `aptitude` offre aussi une option pour récupérer les paquets qui sont recommandés ou suggérés par un paquet à installer. Vous pouvez changer ce comportement en choisissant `F10 -> Options -> Dependency handling` dans le menu.

Autres avantages d'`aptitude` :

- `aptitude` offre accès à toutes les versions d'un paquet.
- `aptitude` enregistre toutes ses actions dans `/var/log/aptitude`.
- `aptitude` rend facile le suivi des logiciels obsolètes en les listant dans « *Obsolete and Locally Created Packages* ».
- `aptitude` inclut un système de recherche puissant pour trouver des paquets particuliers ou limité l'affichage des paquets. Les utilisateurs familiers avec `mutt` seront rapidement à l'aise, puisque mutt a inspiré la syntaxe des expressions. Voir « *SEARCHING, LIMITING, AND EXPRESSIONS* » dans `/usr/share/doc/aptitude/README`.
- `aptitude` en plein écran intègre `su` et peut être utilisé par un utilisateur normal jusqu'à ce qu'il y ait réellement besoin des privilèges de l'administrateur.

6.2.4 `dselect`

Dans les versions stables jusqu'à Potato inclus, `dselect` était l'outil de gestion des paquets principal. Pour Sarge, il est recommandé d'utiliser `aptitude` à la place.

Quand `dselect` est lancé, il sélectionne automatiquement tous les paquets « Required », « Important » et « Standard ».

`dselect` a une interface utilisateur déroutante. La plupart des utilisateurs s'y font cependant. Il a quatre commandes (en majuscules !) :

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

Avec `D` et `Q`, vous pouvez sélectionner des paquets en conflit à vos risques et périls. Faites attention en utilisant ces commandes.

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

Si votre machine exécute `dselect` lentement, vous pouvez lancer `dselect` sur une machine plus rapide pour trouver les paquets et utiliser `apt-get install` sur la machine lente pour les installer.

6.2.5 Suivre une distribution avec APT

Pour suivre la distribution `testing` au fur et à mesure de ses changements, votre fichier `/etc/apt/preferences` doit ressembler à cela :

```
Package: *
```

```
Pin: release a=testing
Pin-Priority: 800

Package: *
Pin: release a=stable
Pin-Priority: 600
```

Attention : suivre la version `testing` de Debian a l'effet de bord de recevoir très lentement les correctifs de sécurité. Ces paquets sont envoyés dans `unstable` et migrent vers `testing` seulement après un certain délai.

Voir `apt_preferences(5)` pour des exemples plus compliqués qui vous permettront, par exemple, de suivre `testing` tout en installant certains paquets de `unstable`.

Des exemples de `/etc/apt/preferences` qui bloquent certains paquets importants dans leur version la plus stable en suivant la version moins stable d'autres paquets non essentiels sont disponibles dans le sous-répertoire d'exemples (`http://www.debian.org/doc/manuals/debian-reference/examples/`) dans les fichiers `preferences.testing` et `preferences.unstable`.

Si vous mélangez les versions de Debian, `testing` dans `stable` ou `unstable` dans `stable`, vous pourriez récupérer sans le vouloir des paquets importants, comme `libc6`, de `testing` ou `unstable` et il n'y a aucune garantie qu'ils ne soient pas bogués. Vous êtes prévenus.

Aussi, le fichier `preferences.stable` force tous les paquets à faire une mise à niveau inférieur vers `stable`.

La mise à niveau inférieur depuis une version plus récente d'un **paquet** n'est pas supportée officiellement dans Debian. Cependant, vous pouvez avoir à mettre à niveau inférieur un paquet spécifique pour réinstaller une version d'un paquet qui marche lorsque la nouvelle version ne fonctionne pas. Vous pouvez trouver ces paquets précédents localement dans `/var/cache/apt/archives/` ou à distance sur `http://snapshot.debian.net/`. Voir aussi 'Sauvetage avec `dpkg`' page 72.

La mise à niveau inférieur depuis une version plus récente d'une **distribution** n'est pas non plus supportée officiellement et risque de poser des problèmes. Cependant, cela peut valoir le coup d'essayer en dernier recours si vous êtes désespérés.

6.2.6 Commandes `aptitude`, `apt-get` et `apt-cache`

Pour fonctionner avec `testing` comme décrit dans l'exemple précédent, le système peut être géré avec les commandes suivantes :

- `aptitude upgrade` (ou `apt-get upgrade` ou `aptitude dist-upgrade` ou `apt-get dist-upgrade`)
  Cela suit la distribution `testing` et met à jour tous les paquets du système en installant leurs dépendances de `testing`. [2]

---

[2] La différence entre `upgrade` et `dist-upgrade` n'apparaît que lorsque de nouvelles versions de paquets ont

- `apt-get dselect-upgrade`
  Cela suit la distribution `testing` et met à jour tous les paquets du système avec la sélection de `dselect`.
- `aptitude install` *`package`*`/unstable`
  Cela installe *package* de `unstable` en prenant les dépendances dans `testing`.
- `aptitude install -t unstable` *`package`*
  Cela installe *package* de `unstable` en prenant les dépendances aussi dans `unstable` en mettant `Pin-Priority` de `unstable` à 990.
- `apt-cache policy` *`foo bar ...`*
  Cela affiche l'état des paquets *foo bar ...*.
- `aptitude show` *`foo bar ...`* `| less` (ou `apt-cache show` *`foo bar ...`* `| less`)
  Cela affiche l'information sur les paquets *foo bar ...*.
- `aptitude install` *`foo=2.2.4-1`*
  Cela installe la version *2.2.4-1* du paquet *foo*.
- `aptitude install` *`foo bar-`*
  Cela installe la paquet *foo* et supprime le paquet *bar*.
- `aptitude remove` *`bar`*
  Cela supprime le paquet *bar* mais garde ses fichiers de configuration.
- `aptitude purge` *`bar`*
  Cela supprime le paquet *bar* et tous ses fichiers de configuration.

Dans les exemples précédents, l'utilisation de l'option `-u` permet qu'`apt-get` affiche la liste des paquets qui seront mis à jour et demande à l'utilisateur de confirmer avant de continuer. Ce qui suit permet de configurer l'option `-u` par défaut :

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

Utilisez l'option `--no-act` pour simuler des actions sans les effectuer.

6.3 Commandes de survie Debian

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

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

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

---

des dépendances différentes des anciennes versions. Voir `apt-get(8)` pour plus de détails. `aptitude upgrade` et `aptitude dist-upgrade` lancent `aptitude` en ligne de commande. Vous pouvez changer pour le mode plein écran en appuyant sur `e`.

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

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

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

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

et tapez

```
$ cd packagename
$ pager README.Debian # s'il existe
$ mc
```

Plus de ressources sont listées dans 'Support Debian' page 233.

6.3.2 Résolution de problèmes de mise à jour avec APT

Des problèmes de dépendances peuvent arriver lors de la mise à jour avec `unstable/testing`, comme décrit dans 'Mise à jour du système Debian' page 63. La plupart du temps, c'est parce qu'un paquet qui doit être mis à jour possède une dépendance qui n'est pas trouvée. Ces problèmes sont résolus en utilisant

```
# aptitude dist-upgrade
```

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

```
# aptitude -f upgrade         # continue « upgrade » malgré les erreurs
... ou
# aptitude -f dist-upgrade    # continue « dist-upgrade » malgré les erreurs
```

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

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

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

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

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

Une autre méthode pour résoudre ces problèmes est d'utiliser le paquet `equivs`. Voir 'Le paquet `equivs`' page 81. Voir `/usr/share/doc/equivs/README.Debian` et 'Le paquet `equivs`' page 81.

6.3.3 Sauvetage avec `dpkg`

Si vous n'arrivez plus à utiliser APT vous pouvez télécharger les paquets sur un miroir Debian et les installer en utilisant `dpkg`. Si vous n'avez pas accès au réseau, vous pouvez chercher des copies des paquets dans `/var/cache/apt/archives/`.

```
# dpkg -i fetchmail_6.2.5-4_i386.deb
```

Si l'installation d'un paquet de cette façon échoue à cause d'une dépendance non résolue, et que vous avez vraiment besoin d'installer le paquet, vous pouvez ignorer les dépendances en utilisant `--ignore-depends`, `--force-depends` et d'autres options avec `dpkg`. Voir See `dpkg(8)` pour plus de détails.

6.3.4 Récupérer les données de sélection des paquets

Si `/var/lib/dpkg/status` est corrompu, le système Debian perd les données de sélection des paquets et en est grandement affecté. Cherchez le vieux fichier `/var/lib/dpkg/status` dans `/var/lib/dpkg/status-old` ou `/var/backups/dpkg.status.*`.

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

Si le vieux fichier `/var/lib/dpkg/status` n'est pas disponible, vous pouvez quand même récupérer les données depuis des répertoires de `/usr/share/doc/`.

```
# ls /usr/share/doc | \
  grep -v [A-Z] | \
  grep -v '^texmf$' | \
  grep -v '^debian$' | \
  awk '{print $1 " install"}' | \
  dpkg --set-selections
# dselect --expert # réinstallez le système, utilisez dselect si besoin
```

6.3.5 Sauver un système après avoir perdu `/var`

Comme le répertoire `/var` contient des données mises à jour régulièrement, comme le mail, il est plus susceptible de corruption que, par exemple `/usr/`. Le placer dans une partition indépendante limite les risques. Si le pire arrive, vous pouvez avoir à reconstruire le répertoire `/var` pour sauver le système Debian.

Obtenez la structure du répertoire `/var` à partir d'un système Debian qui fonctionne, basé sur la même ou une plus ancienne version de Debian, par exemple `/var`, et placez la à la racine du système cassé. Ensuite

```
# cd /
# mv var var-old      # s'il reste des données utiles
# tar xvzf var.tar.gz # utilisez la structure de Woody
# aptitude            # ou dselect
```

Cela doit fournir un système qui fonctionne. Vous pouvez récupérer les sélections de paquets en utilisant la technique décrite à 'Récupérer les données de sélection des paquets' page précédente. ([FIXME] : Cette procédure nécessite plus de tests.)

6.3.6 Installer un paquet sur un système qui ne démarre plus

Démarrez Linux en utilisant une disquette/CD de sauvetage Debian ou une autre partition sur un système Linux multi-boot. Voir 'Démarrer le système' page 93. Montez le système qui ne démarre plus sur `/target` et utilisez le mode d'installation chroot de `dpkg`.

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

Configurez ensuite et résolvez les problèmes.

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

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

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

6.3.7 Que faire si `dpkg` est cassé

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

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

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

6.4 Commandes « nirvana » de Debian

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

6.4.1 Informations sur un fichier

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

```
$ dpkg {-S|--search} pattern # cherche « pattern »
                             # parmi les paquets installés
$ wget http://ftp.us.debian.org/debian/dists/sarge/Contents-i386.gz
$ zgrep -e pattern Contents-i386.gz
             # cherche « pattern » parmi les fichiers de l'archive Debian
```

Sinon, utilisez les commandes spécialisées pour les paquets :

```
# aptitude install dlocate
$ dlocate filename
        # solution de rechange rapide pour dpkg -L et dpkg -S
...
# aptitude install auto-apt # outil d'installation de paquets à la demande
# auto-apt update          # crée le fichier db pour auto-apt
$ auto-apt search pattern
        # recherche exhaustive dans l'archive de paquets, installés ou non
```

6.4.2 Informations sur un paquet

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

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

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

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

```
/var/lib/apt/lists/*
/var/lib/dpkg/available
```

La comparaison des fichiers suivants fournit des informations sur ce qu'il s'est exactement passé lors des dernières installations.

```
/var/lib/dpkg/status
/var/backups/dpkg.status*
```

6.4.3 Installation automatique avec APT

Pour une installation non interactive, ajoutez la ligne suivante dans `/etc/apt/apt.conf` :

```
Dpkg::Options {"--force-confold";}
```

C'est l'équivalent de lancer `aptitude -y install` *packagename* ou `apt-get -q -y install` *packagename*. Parce que cela répond automatiquement « oui » à toutes les questions, cela peut causer des problèmes, aussi faites attention en utilisant cette astuce. Voir `apt.conf(5)` et `dpkg(1)`.

Vous pouvez configurer les paquets par la suite en suivant 'Reconfigurer les paquets installés' page suivante.

6.4.4 Reconfigurer les paquets installés

Utilisez les commandes suivantes pour reconfigurer un paquet installé.

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

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

Certains logiciels sont livrés avec des scripts de configuration. [3]

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

6.4.5 Enlever et purger des paquets

Enlever un paquet en gardant la configuration :

```
# aptitude remove package ...
# dpkg  --remove package ...
```

Enlever un paquet et sa configuration :

```
# aptitude purge package ...
# dpkg    --purge        package ...
```

---

[3] Certains scripts de configuration disparaissent dans les dernières versions de Sarge et les fonctions de configuration des paquets sont déplacées dans le système `debconf`.

6.4.6 Garder de vieux paquets

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

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

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

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

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

La commande suivante liste les paquets à garder :

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

6.4.7 Système mixte `stable/testing/unstable`

`apt-show-versions` peut lister les versions des paquets disponibles dans chaque distribution.

```
$ apt-show-versions | fgrep /testing | wc
... nombre de paquets de testing
$ apt-show-versions -u
... liste de paquets qu'il est possible de mettre à jour
$ aptitude install `apt-show-versions -u -b | fgrep /unstable`
... met à jour tous les paquets de unstable à leur dernière version
```

6.4.8 Supprimer les paquets du cache

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

```
# aptitude autoclean # efface les fichiers de paquets inutiles
# aptitude clean     # efface tous les fichiers de paquets du cache
```

6.4.9 Enregistrer/copier la configuration du système

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

```
# debconf-get-selections > debconfsel.txt
# dpkg --get-selections "*" >myselections   # ou utiliser \*
```

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

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

```
# dselect update
# debconf-set-selections < debconfsel.txt
# dpkg --set-selections <myselections
# apt-get -u dselect-upgrade    # ou dselect install
```

6.4.10 Porter un paquet vers le système `stable`

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

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

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

Ensuite, récupérez les sources et construisez un paquet :

```
$ apt-get update  # met à jour la liste des paquets source
$ apt-get source package
$ dpkg-source -x package.dsc
$ cd package-version
  ... cherche parmi les paquets requis (Build-Depends dans le fichier .dsc)
      et les installe. Il faut le paquet « fakeroot » aussi.

$ dpkg-buildpackage -rfakeroot

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

```
   ...ensuite pour installer
$ su -c "dpkg -i packagefile.deb"
```

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

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

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

6.4.11 Archive locale de paquets

Afin de créer une archive locale de paquets qui soit compatible avec les systèmes APT et dselect, un fichier `Packages` doit être créé et les fichiers de paquets ont besoin d'être archivés dans un arbre de répertoires particulier.

Un dépôt local similaire à une archive officielle Debian peut être créé avec :

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

Autrement, un dépôt local peut être créé de façon rapide et sale :

```
# aptitude install dpkg-dev
# mkdir /usr/local/debian
# mv /some/where/package.deb /usr/local/debian
# dpkg-scanpackages /usr/local/debian /dev/null | \
  gzip - > /usr/local/debian/Packages.gz
#  echo "deb file:/usr/local/debian ./" >> /etc/apt/sources.list
```

Ces archives peuvent être accédées à distance en fournissant ces répertoires aux méthodes HTTP ou FTP des entrées du fichier `/etc/apt/sources.list`.

6.4.12 Convertir ou installer un paquet non Debian

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

6.4.13 Installation automatique

`auto-apt` est une commande d'installation de paquets à la demande.

```
$ sudo auto-apt update
 ... met à jour la base de données
$ auto-apt -x -y run
Entering auto-apt mode: /bin/bash
Exit the command to leave auto-apt mode.
$ less /usr/share/doc/med-bio/copyright # accède un fichier non existant
 ... installe le paquet qui fournit ce fichier
 ... installe aussi les dépendances
```

6.4.14 Vérifier les fichiers de paquets installés

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

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

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

6.4.15 Améliorer le fichier `sources.list`

En bref, mes efforts pour optimiser `sources.list` n'ont pas réellement apporté de gain de performance pour moi qui vit aux Etats-Unis. J'ai choisi manuellement un site proche avec `apt-setup`.

`apt-spy` crée un fichier `sources.list` automatiquement en testant chaque site sur son temps de latence et sa bande passante. `netselect-apt` crée un fichier `sources.list` plus complet, mais utilise une méthode moins bonne pour choisir le meilleur miroir (comparaison des temps de ping).

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

6.5 Autres particularités de Debian

6.5.1 La commande `dpkg-divert`

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

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

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

6.5.2 Le paquet `equivs`

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

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

6.5.3 Commandes de rechange

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

```
# update-alternatives --display vi
...
# update-alternatives --config vi
  Selection    Command
-----------------------------------------------
      1        /usr/bin/elvis-tiny
      2        /usr/bin/vim
*+    3        /usr/bin/nvi

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

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

Pour avoir votre environnement de fenêtres X favori, appliquez `update-alternatives` à `/usr/bin/x-session-manager` et `/usr/bin/x-window-manager`. Pour plus de détails, voir 'Sessions X personnalisées' page 137.

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

6.5.4 Utilisation des niveaux de fonctionnement

Lorsqu'ils sont installés, la plupart des paquets Debian configurent leurs services pour être lancés dans les niveaux de fonctionnement 2 à 5. Ainsi, il n'y a pas de différence entre les niveaux 2, 3, 4 et 5 sur un système Debian qui n'a pas été personnalisé ; Debian laisse à l'administrateur local le soin de configurer les niveaux de fonctionnement comme décrit dans 'Personnaliser les niveaux de fonctionnement' page 22. Cela diffère de la façon dont les niveaux de fonctionnement sont gérés dans d'autres distributions GNU/Linux populaires. Un changement que vous pouvez vouloire faire est de désactiver `xdm` ou `gdm` dans le niveau de fonctionnement 2 pour que X ne soit pas démarré à la fin de la séquence de démarrage ; vous pouvez alors le démarrer en passant au niveau 3.

Pour plus d'information sur les niveaux de fonctionnement, voir 'Niveaux de fonctionnement' page 21.

6.5.5 Services désactivés

Les développeurs Debian prennent très au sérieux la sécurité du système. Beaucoup de services sont installés avec le minimum de fonctions activées.

Exécutez `ps aux` ou consultez le contenu de `/etc/init.d/*` et `/etc/inetd.conf`, si vous avez des doutes (à propos d'Exim, DHCP, ...). De plus, consultez `/etc/hosts.deny` de la même façon qu'à la section 'Contrôle des connexions avec PAM' page 123. La commande `pidof` est aussi utile (voir `pidof(8)`).

X11 ne permet pas les connexions TCP/IP (distantes) par défaut dans les versions récentes de Debian. Voir 'Utiliser X sur TCP/IP' page 139. La redirection X est aussi désactivée dans SSH. Voir 'Connexion à un serveur X distante – `ssh`' page 140.

Chapitre 7

Le noyau Linux et Debian

Debian a une manière de recompiler le noyau Linux et les modules associés qui lui est propre. Voir aussi 'Debian et le noyau' page 23.

7.1 Recompilation du noyau

L'utilisation de `gcc`, `binutils` et `modutils` de Debian `unstable` peut aider lors de la compilation du dernier noyau Linux. Voir `/usr/share/doc/kernel-package/README.gz`, et spécialement la fin, pour l'information officielle.

Comme c'est en évolution constante, la compilation d'un noyau est difficile même pour les meilleurs développeurs :

Manoj Srivastava a écrit : `--initrd` nécessite un patch cramfs de Debian.

Herbert Xu a écrit : Non, il n'y en a pas besoin, tout ce qu'il faut faire pour utiliser un système de fichier autre que CRAMFS est de configurer MKIMAGE dans `/etc/mkinitrd/mkinitrd.conf`.

Attention et reposez-vous toujours sur `/usr/share/doc/kernel-package/README.gz` de Manoj et Kent. Assurez-vous d'obtenir la dernière version de `kernel-package` de unstable si vous voulez compiler la dernière version du noyau.

`initrd` n'est pas nécessaire pour un noyau compilé pour une seule machine. Je l'utilise parce que je veux que mon noyau soit presque identique à celui fournit par kernel-image. Si vous utilisez `initrd`, lisez bien `mkinitrd(8)` et `mkinitrd.conf(5)`. Voir aussi `http://bugs.debian.org/149236`.

7.1.1 Méthode Debian standard

Soyez attentifs aux rapports de bogues relatifs à `kernel-package` `gcc`, `binutils` et `modutils`. Utilisez une nouvelle version de ceux-ci si nécessaire.

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

```
# apt-get install debhelper modutils kernel-package libncurses5-dev
# apt-get install kernel-source-2.4.18   # utilisez la dernière version
# apt-get install fakeroot
# vi /etc/kernel-pkg.conf     # entrez votre nom et adresse électronique
$ cd /usr/src                 # dossier où va se faire la compilation
$ tar --bzip2 -xvf kernel-source-2.4.18.tar.bz2
$ cd kernel-source-2.4.18     # si c'est votre source du noyau
$ cp /boot/config-2.4.18-386 .config     # récupérez la configuration
                                           actuelle comme defaut
$ make menuconfig             # personnalisez votre noyau
$ make-kpkg clean             # obligatoire  (voir : man make-kpkg)
$ fakeroot make-kpkg --append_to_version -486 --initrd \
        --revision=rev.01 kernel_image \
        modules_image # modules_image pour pcmcia-cs* etc.
$ cd ..
# dpkg -i kernel-image*.deb pcmcia-cs*.deb # installation
```

En réalité, `make-kpkg kernel_image` lance `make oldconfig` et `make dep`. N'utilisez pas `--initrd` si initrd n'est pas utilisé.

Si vous voulez les modules de pcmcia-cs, ou pas de support pcmcia, sélectionnez « General setup —> » dans « PCMCIA/CardBus support —> » dans `make menuconfig` et en mettant la configuration à « < > PCMCIA/CardBus support » (càd décocher la case).

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

7.1.2 Méthode classique

Récupérez les sources officielles depuis :

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

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

```
# cd /usr/src
# tar xfvz linux-whatever.tar.gz
# rm -rf linux
# ln -s linux-whatever linux
# tar xfvz pcmcia-cs-whatever.tar.gz
# ln -s pcmcia-cs-whatever pcmcia
# cd linux
```

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

7.1.3 En-têtes du noyau

La plupart des programmes « normaux » n'ont pas besoin des en-têtes du noyau et peuvent même planter si vous les utilisez directement ; à la place, ils doivent être compilés avec les en-têtes avec lesquels la **`glibc` a été compilée**, qui sont les versions de `/usr/include/linux` et `/usr/include/asm` du système Debian.

Donc, ne créez pas de liens vers `/usr/src/linux` depuis `/usr/include/linux` et `/usr/include/asm`, comme suggéré par certains documents trop vieux.

Si vous **avez besoin** d'en-têtes du noyau particuliers pour une application spécifique à un noyau, changez le `Makefile` pour inclure les chemins vers *`rep-des-entêtes-particulières`*`/include/linux` et *`rep-des-entêtes-particulières`*`/include/asm`.

7.2 Le noyau 2.4 modulaire

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

Bien qu'il y ait de nombreux exemples pour `/etc/modules` dans la section suivante, on me dit qu'un moyen correct de corriger les problèmes relatifs aux modules est de fournir un alias au périphérique dans un fichier de `/etc/modutils/` puisqu'assez d'alias sont disponibles avec les noyaux actuels.

Voir 'Dispositions spéciales pour manipuler les modules' page 24 et `Documentation/*.txt` dans les sources de Linux pour l'information précise.

7.2.1 PCMCIA

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

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

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

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

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

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

—Mike »

7.2.2 SCSI

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

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

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

7.2.3 Fonctions réseau

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

```
# net/ipv-4
ip_gre
ipip

# net/ipv-4/netfilter
# iptable (dans l'ordre)
ip_tables
ip_conntrack
ip_conntrack_ftp
iptable_nat
iptable_filter
iptable_mangle
#
ip_nat_ftp
ip_queue
#
ipt_LOG
ipt_MARK
ipt_MASQUERADE
ipt_MIRROR
ipt_REDIRECT
ipt_REJECT
ipt_TCPMSS
ipt_TOS
ipt_limit
ipt_mac
ipt_mark
ipt_multiport
ipt_owner
ipt_state
ipt_tcpmss
ipt_tos
ipt_unclean
#
#ipchains
#ipfwadm
```

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

7.2.4 Système de fichiers EXT3 (> 2.4.17)

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

```
# cd /etc; mv fstab fstab.old
```

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

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

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

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

La journalisation EXT3 du système de fichier est maintenant activée.

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

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

Sur l'écran de démarrage (`dmesg`), « cramfs : wrong magic » apparaît mais cette erreur est connue pour être sans conséquence. Ce problème a été résolu dans Sarge

(10/2002). Voir `http://bugs.debian.org/135537` et le EXT3 File System mini-HOWTO (`http://www.zip.com.au/~akpm/linux/ext3/ext3-usage.html`) ou `/usr/share/doc/HOWTO/en-txt/mini/extra/ext3-mini-HOWTO.gz` pour plus d'information.

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

7.2.5 Support Realtek RTL-8139 dans le noyau 2.4

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

7.2.6 Support du port parallèle

Avec `kernel-image-2.4.*`, le support du port parallèle est fournit en module. Activez-le avec :

```
# modprobe lp
# echo lp >> /etc/modules
```

Voir `Documentation/parport.txt` dans le source de Linux.

7.2.7 Trop de fichiers ouverts

Le noyau Linux peut se plaindre : « Too many open files » (trop de fichiers ouverts). Cela est dû à la petite valeur (8096) de `file-max`. Pour corriger ce problème, lancez les commandes suivantes en tant que root (ou mettez-les dans un script de démarrage de `/etc/rcS.d/*`.

```
# echo "65536"  >/proc/sys/fs/file-max  # pour les noyaux 2.2 et 2.4
# echo "131072" >/proc/sys/fs/inode-max # pour le noyau 2.2 seulement
```

Pour plus d'information sur le changement des paramètres du noyau dans le système de fichiers `/proc`, lisez `Documentation/sysctl/*` dans les sources de Linux.

D'autres exemples de manipulation des paramètres du noyau sont dans `/etc/init.d/networking` et 'Étranges problèmes d'accès à certains sites web' page 42.

Chapitre 8

Astuces Debian

8.1 Démarrer le système

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

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

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

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

A l'écran d'invite de `lilo`, dès que `boot:` apparaît (il faut appuyer sur la touche MAJ sur certains systèmes pour éviter le démarrage automatique et lorsque `lilo` utilise le framebuffer il faut appuyer sur TAB pour voir les options que vous tapez), entrez :

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

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

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

```
init-2.03# vi passwd
init-2.03# vi shadow
```

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

C’est une bonne idée d’avoir un éditeur minimum dans `/bin` au cas où `/usr` ne serait pas accessible (voir ‘Editeurs de sauvetage’ page 195).

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

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

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

8.1.2 « J’ai oublié le mot de passe de root ! » (2)

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

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

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

8.1.3 Je ne peux pas lancer le système

Pas de problème, personne ne prend la peine de faire une disquette de démarrage lors de l’installation. Si `lilo` est cassé, récupérez la disquette de démarrage de l’installation de Debian (rescue). Démarrez depuis cette disquette. A l’invite de démarrage, en supposant que la partition racine de Linux est *`/dev/hda12`* et que vous voulez le niveau d’exécution (runlevel) 3, entrez :

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

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

Voyez aussi 'Installer un paquet sur un système qui ne démarre plus' page 73 si votre système est planté.

Si vous avez besoin d'une disquette de démarrage personnalisée, suivez les instructions de `readme.txt` sur la disquette de secours.

8.1.4 « Je veux désactiver X au démarrage ! »

Suivre `unstable/sid` est sympa, mais un bogue de `xdm`, `gdm`, `kdm`, ou `wdm` lancé pendant le processus de démarrage peut faire mal.

D'abord, obtenez un shell avec root en tapant ce qui suit à l'invite de démarrage :

```
boot: Linux vga=normal s
```

Là, *Linux* est le nom de l'image du noyau que vous démarrez, « vga=normal » s'assure que `lilo` s'exécute en mode VGA, et « s » (ou « S ») est le paramètre passé à `init` pour lancer le mode mono-utilisateur. Entrez le mot de passe de root à l'invite.

Il existe quelques moyens de désactiver tous les démons X qui démarrent :

- lancez `update-rc.d ?dm stop 99 1 2 3 4 5 6 .`
- insérez « exit 0 » au début de tous les fichiers `/etc/init.d/ ?dm`
- renommez tous les fichiers `/etc/rc2.d/S99 ?dm` en `/etc/rc2.d/K99 ?dm`
- supprimez tous les fichiers `/etc/rc2.d/S99 ?dm`
- exécutez `:>/etc/X11/default-display-manager`

Ici, `rc2.d` doit correspondre au niveau de fonctionnement spécifié dans `/etc/inittab`. De plus, `?dm` signifie tous les fichiers parmi `xdm`, `gdm`, `kdm`, et `wdm`.

Seul le premier est le « véritable moyen unique » de Debian. Le dernier est facile mais marche seulement sur Debian et nécessite que vous reconfiguriez le gestionnaire d'affichage par la suite avec `dpkg-reconfigure`. Les autres sont des méthodes génériques pour désactiver des démons.

Vous pouvez toujours lancer X avec la commande `startx` depuis une console.

8.1.5 Autres astuces avec l'invite de démarrage

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

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

```
boot: Linux 4
```

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

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

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

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

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

8.1.6 Comment configurer les paramètres de démarrage de GRUB

GRUB est un nouveau gestionnaire de démarrage issu du projet Hurd et est beaucoup plus flexible que Lilo mais a une manière différente de gérer les paramètres de démarrage.

```
grub> find /vmlinuz
grub> root (hd0,0)
grub> kernel /vmlinuz root=/dev/hda1
grub> initrd /initrd
grub> boot
```

Là, vous devez connaître les noms de périphériques de Hurd :

```
the Hurd/GRUB          Linux              MSDOS/Windows
 (fd0)              /dev/fd0          A:
 (hd0,0)            /dev/hda1         C: (habituellement)
 (hd0,3)            /dev/hda4         F: (habituellement)
 (hd1,3)            /dev/hdb4         ?
```

Voir `/usr/share/doc/grub/README.Debian.gz` et `/usr/share/doc/grub-doc/html/` pour les détails.

8.2 Enregistrer les activités

8.2.1 Enregistrer les activités du shell

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

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

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

Sous le shell : utilisez la commande `screen` avec « ^A H » comme décrit à la section 'Changement de console avec `screen`' page 113 ou la commande `script`.

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

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

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

8.2.2 Enregistrer les activités sous X

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

8.3 Copier et archiver un sous-répertoire entier

8.3.1 Commandes de base pour copier un sous-répertoire entier

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

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

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

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

8.3.2 **`cp`**

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

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

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

8.3.3 **`tar`**

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

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

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

8.3.4 **`pax`**

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

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

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

8.3.5 **`cpio`**

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

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

8.3.6 **`afio`**

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

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

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

8.4 Sauvegarde différentielle et synchronisation de données

La sauvegarde différentielle et la synchronisation de données peuvent être implémentés par plusieurs méthodes :

- `rcs` : sauvegarde et historique, texte seulement
- `rdiff-backup` : sauvegarde et historique. liens symboliques OK.
- `pdumpfs` : sauvegarde et historique à l'intérieur d'un système de fichier. liens symboliques OK.

- `rsync` : synchronisation dans un seul sens
- `unison` : synchronisation dans deux sens
- `cvs` : synchronisation dans plusieurs sens avec sauvegarde et historique sur un serveur, texte seulement, mûr. Voir 'Concurrent Versions System (CVS)' page 203.
- `arch` : synchronisation dans plusieurs sens avec sauvegarde et historique sur un serveur, pas de « répertoire de travail ».
- `subversion` : synchronisation dans plusieurs sens avec sauvegarde et historique sur un serveur, Apache.

La combinaison d'une des méthodes décrites dans la section 'Copier et archiver un sous-répertoire entier' page 97 et le processus régulier automatisé décrit dans la section 'Programmer des activités (`cron`, `at`)' page 112 permet de faire un bon système de sauvegarde.

Je vais décrire trois outils faciles à utiliser.

8.4.1 Sauvegarde différentielle avec rdiff

`rdiff-backup` offre une sauvegarde bonne et simple avec historique différentiel pour n'importe quel type de fichiers y compris des liens symboliques. Pour sauvegarder la plus grande partie de `~/` vers `/mnt/backup` :

```
$ rdiff-backup --include ~/tmp/keep --exclude ~/tmp  ~/ /mnt/backup
```

Pour restaurer les données d'il y a trois jours de cette archive vers `~/old` :

```
$ rdiff-backup -r 3D /mnt/backup ~/old
```

Voir `rdiff-backup(1)`.

8.4.2 Sauvegarde quotidienne avec `pdumpfs`

`pdumpfs` est un système simple de sauvegarde quotidienne similaire à `dumpfs` du système Plan9, et qui garde chaque image quotidienne. Vous pouvez accéder aux images antérieures n'importe quand pour récupérer un fichier d'un certain jour. Sauvegardons votre répertoire personnel avec `pdumpfs` et `cron` !

`pdumpfs` construit une image `AAAA/MM/JJ` dans le répertoire destination. Tous les fichiers source sont copiés vers le répertoire de l'image la première fois que `pdumpfs` est exécuté. A partir de la deuxième fois, `pdumpfs` copie seulement les ficheirs modifiés ou créés et stocke les fichiers inchangés avec des liens durs vers les fichiers du jour précédent, pour économiser de l'espace disque.

```
$ pdumpfs src-dir dest-dir [dest-basename]
```

Voir `pdumpfs(8)`.

8.4.3 Sauvegarde différentielle régulière avec RCS

`changetrack` enregistre les changements des fichiers texte de configuration dans des archives RCS régulièrement. Voir `changetrack(1)`.

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

8.5 Récupération d'un système bloqué

8.5.1 Tuer un processus

Exécutez `top` pour voir quel processus agit bizarrement. Appuyez sur « P » pour trier par temps CPU utilisé, « M » pour trier par mémoire utilisée, et « k » pour tuer un processus. Autrement, `ps aux | less` (style BSD) ou `ps -efH | less` (style System V) peut être utilisé. La syntaxe System V affiche le numéro de processus parent `PPID` qui peut être utilisé pour tuer des processus zombie (defunct).

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

```
 1 : HUP,  redémarre un démon
15 : TERM, termine un process normalement
 9 : KILL, tue un process
```

8.5.2 Alt-SysRq

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

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

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

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

8.6 Petites commandes utiles à se rappeler

8.6.1 Pager

`less` est le *pager* (visualisateur de contenu des fichiers). Appuyez sur 'h' pour de l'aide. Il peut faire beaucoup plus que `more`. `less` peut être amélioré en exécutant `eval $(lesspipe)` ou `eval $(lessfile)` dans les scripts de démarrage du shell. Plus d'information dans `/usr/share/doc/lessf/LESSOPEN`. L'option `-R` permet l'affichage de caractères bruts et active les séquences d'échappement couleur ANSI. Voir `less(1)`.

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

8.6.2 Mémoire libre

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

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

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

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

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

8.6.3 Régler l'heure (BIOS)

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

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

Si l'heure matérielle (BIOS) est réglée en GMT, changez le réglage pour `UTC=yes` dans le fichier `/etc/default/rcS`.

8.6.4 Régler l'heure (NTP)

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

Régler l'heure avec une connexion Internet permanente

Régler l'horloge système automatiquement via un serveur distant :

```
# ntpdate server
```

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

Régler l'heure avec une connexion Internet intermitente

Utilisez le paquet `chrony`.

8.6.5 Comment contrôler des possibilités de la console comme l'écran de veille

Pour désactiver l'écran de veille, utilisez les commandes suivantes.

En mode console :

```
# setterm -powersave off
```

Démarrez la console kon2 (kanji) avec :

```
# kon -SaveTime 0
```

Sous X :

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

Lisez les pages de manuel correspondantes pour contrôler d'autres possibilités de la console. Voir aussi `stty(1)` pour changer et afficher la configuration du terminal.

8.6.6 Chercher dans la base de données administrative

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

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

8.6.7 Désactiver le son (beep)

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

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

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

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

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

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

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

8.6.9 Régler la console

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

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

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

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

Si l'entrée terminfo pour `xterm` ne marche pas avec un `xterm` non Debian, changez le type de terminal de « xterm » à l'une des versions limitée en options comme « xterm-r6 » lorsque vous vous connectez à un système Debian à distance. Voir `/usr/share/doc/libncurses5/FAQ` pour plus d'information. « dumb » est le plus petit dénominateur commun pour terminfo.

8.6.10 Remettre la console dans un état sain

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

```
$ reset
```

8.6.11 Convertir des fichiers texte DOS vers Unix

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

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

8.6.12 Convertir un fichier texte avec `recode`

Ce qui suit convertit les fichiers entre les caractères de fin de ligne DOS, Mac et Unix :

```
$ recode /cl../cr <dos.txt >mac.txt
$ recode /cr.. <mac.txt >unix.txt
$ recode ../cl <unix.txt >dos.txt
```

`recode` convertir aussi les fichiers entre les différents jeux de caractères et surfaces :

```
$ recode charset1/surface1..charset2/surface2 \
 <input.txt >output.txt
```

Les jeux de caractères les plus courants sont (voir aussi 'Introduction aux locales' page 156) [1] :

- `us` — ASCII (7 bits)
- `l1` — ISO Latin-1 (ISO-8859-1, Europe de l'Ouest, 8 bits)

---

[1] `recode` permet des alias plus pratiques que `iconv`.

  - `EUCJP` — EUC-JP pour le Japonais (Unix)
  - `SJIS` — Shift-JIS pour le Japonais (Microsoft)
  - `ISO2022JP` — codage mail pour le Japonais (7 bits)
  - `u2` — UCS-2 (Universal Character Set, 2 octets)
  - `u8` — UTF-8 (Universal Transformation Format, 8 bits)

Les surfaces les plus courantes sont[2] :

  - `/cr` — Retour chariot comme fin de ligne (texte Mac)
  - `/cl` — Retour chariot - remplissage de ligne comme fin de ligne (texte DOS)
  - `/` — Remplissage de ligne comme fin de ligne (texte Unix)
  - `/d1` — sortie décimale lisible par un humain
  - `/x1` — sortie hexadécimale lisible par un humain
  - `/64` — texte codé Base64
  - `/QP` — texte codé Quoted-Printable

Pour plus de détails, voir la description de `info recode`.

Il existe aussi des outils de conversion plus spécialisés :

- conversion de jeux de caractères :
  - `iconv` — conversions de codages de locales
  - `konwert` — conversions de codages variés
- conversion de fichiers binaires :
  - `uuencode` et `uudecode` — pour Unix
  - `mimencode` — pour le courriel

8.6.13 Substitutions à l'aide des expressions rationnelles

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

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

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

8.6.14 Editer un fichier avec un script

Le script suivant supprime les lignes 5 à 10 et 16 à 20.

```
#!/bin/bash
```

---

[2] Fins de lignes :
- Retour chariot signifie ASCII 13, ASCII 0xD, ^M, ou \r.
- Remplissage de ligne signifie ASCII 10, ASCII 0xA, ^J, or \n.

```
ed $1 <<EOF
16,20d
5,10d
w
q
EOF
```

Les commandes de `ed` sont les mêmes que celles de `vi` en mode commande. L'édition de fichiers de cette façon rend facile la création de scripts.

8.6.15 Extraire des différences et introduire des mises à jour dans un fichier source

Suivez l'une de ces procédures pour extraire les différences par rapport au fichier source et créer des fichiers diff unifiés *file.patch0* ou *file.patch1* selon l'emplacement du fichier :

```
$ diff -u file.old file.new1 > file.patch0
$ diff -u old/file new1/file > file.patch1
```

Le fichier diff (aussi appelé rustine ou patch) est utilisé pour envoyer des mises à jour du programme. Celui qui le reçoit peut appliquer cette mise à jour à un autre fichier *file* avec :

```
$ patch -p0 file < file.patch0
$ patch -p1 file < file.patch1
```

Si vous avez trois versions d'un code source, vous pouvez les mélanger plus efficacement avec `diff3` :

```
$ diff3 -m file.mine file.old file.yours > file
```

8.6.16 Convertir un gros fichier en plusieurs petits

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

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

Considérons un fichier texte `DPL` dans lequel tous les responsables précédents du projet Debian (DPL, Debian Project Leader) et leur date d'investiture sont listés séparés par des espaces dans un tableau.

```
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
```

AWK est fréquemment utilisé pour extraire des données à partir de ce type de tableaux.

```
$ awk '{ print $3 }' <DPL                   # mois d'investiture
August
April
January
January
April
April
March
$ awk '($1=="Ian") { print }' <DPL          # DPL appelé Ian
Ian     Murdock   August  1993
Ian     Jackson   January 1998
$ awk '($2=="Perens") { print $3,$4 }' <DPL # quand Perens a été investi
April 1996
```

Des shells comme Bash peuvent aussi être utilisés pour ce genre de fichiers :

```
$ while read first last month year; do
    echo $month
  done <DPL
... même sortie que le premier exemple Awk
```

Ici, la commande de Bash `read` utilise les caractères de $IFS (internal field separators, séparateurs de champs internes) pour séparer les lignes en mots.

Si vous changez $IFS à “ :”, vous pouvez traiter le fichier `/etc/passwd` avec le shell :

```
$ oldIFS="$IFS"   # garde l'ancienne valeur
$ IFS=":"
$ while read user password uid gid rest_of_line; do
    if [ "$user" = "osamu" ]; then
      echo "$user's ID is $uid"
    fi
  done < /etc/passwd
osamu's ID is 1001
$ IFS="$oldIFS"   # remet l'ancienne valeur
```

(Si Awk est utilisé pour faire cela, utilisez `FS=":"` pour régler le séparateur de champs.)

IFS est aussi utilisé par le shell pour séparer les résultats de l'expansion de paramètres, de la substitution de commande, et l'expansion arithmétique. Celles-ci ne sont pas effectuées à l'intérieur de guillemets simples ou doubles. La valeur par défaut de IFS est <space>, <tab> et <newline>.

Faites attention en utilisant IFS. Des choses étranges peuvent survenir, lorsque le shell interprète certaines parties du script comme **entrée**.

```
$ IFS=":,"                          # utilise ":" et "," come IFS
$ echo IFS=$IFS,   IFS="$IFS"       # echo est une commande Bash
IFS=  , IFS=:,
$ date -R                           # une simple sortie de commande
Sat, 23 Aug 2003 08:30:15 +0200
$ echo $(date -R)                   # sous-shell --> entrée vers
                                      le shell principal
Sat  23 Aug 2003 08 30 36 +0200
$ unset IFS                         # restaure la valeur par défaut
$ echo $(date -R)
Sat, 23 Aug 2003 08:30:50 +0200
```

8.6.18 Bouts de scripts pour les tubes

Les scripts suivants font des choses sympa avec les tubes.

```
find /usr | egrep -v "/usr/var|/usr/tmp|/usr/local"
                     # trouve tous les fichiers de /usr sauf certains
xargs -n 1 command   # exécute command avec tous les arguments de stdin
xargs -n 1 echo |    # place chaque argument séparé par des espaces
                       sur une ligne
xargs echo      |    # fusionne toutes les lignes en une seule
grep -e pattern|     # extrait les lignes contenant pattern
cut -d: -f3 -|
        # extrait le 3e champ séparé par : (fichier passwd, etc.)
awk '{ print $3 }' | # extrait le 3e champ séparé par des espaces
awk -F'\t' '{ print $3 }' |
        # extrait le 3e champ séparé par des tabulations
col -bx |            # supprime les retour-arrières et transforme
                       les tabulations en espaces
expand -|            # transforme les tabulations
sort -u|             # trie et supprime les doublons

tr '\n' ' '|         # concatène les lignes en une seule
tr '\r' ''|          # supprime CR
```

```
tr 'A-Z' 'a-z'|        # convertit les majuscules en minuscules
sed 's/^/# /'|         # change chaque ligne en commentaire
sed 's/\.ext//g'|      # supprime .ext
sed  -n -e 2p|         # affiche la 2e ligne
head -n 2 -|           # affiche les 2 premières lignes
tail -n 2 -|           # affiche les 2 dernières lignes
```

8.6.19 Bouts de scripts pour boucler sur chaque fichier

Les façons suivantes de boucler sur chaque fichier correspondant à `*.ext` assure une gestion correcte des noms de fichiers atypiques comme ceux ayant des espaces et effectuent la même chose :

- Boucle shell (ceci est une entrée multi-ligne avec `PS2="> "`)[3] :
  ```
  for x in *.ext; do
    if test -f "$x"; then
      command "$x"
    fi
  done
  ```
- `find` et `xargs` :
  ```
  find . -type f -maxdepth 1 -name '*.ext' -print0 | \
   xargs -0 -n 1 command
  ```
- `find` avec l'option `-exec` et une commande :
  ```
  find . -type f -maxdepth 1 -name '*.ext' \
   -exec command '{}' \;
  ```
- `find` avec l'option `-exec` et un script shell :
  ```
  find . -type f -maxdepth 1 -name '*.ext' \
   -exec sh -c "command '{}' && echo 'successful'" \;
  ```

8.6.20 Série de courts scripts en Perl

Tous les scripts Awk peuvent être réécrits avec Perl. Par exemple :

```
awk '($2=="1957") { print $3 }'|
```

peut être remplacé par l'une des lignes suivantes au choix :

```
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"' |
```

---

[3] Si vous tapez ceci sur une seule ligne, vous devez ajouter des points-virgule pour marquer la fin des commandes shell.

On peut enlever tous les espaces dans les arguments de `perl` dans la ligne précédente et prendre avantage de la conversion automatique entre les nombres et les chaînes de caractères en Perl :

```
perl -lane 'print$F[2]if$F[1]eq+1957' |
```

Voir `perlrun(1)` pour les options de la ligne de commande. Pour des scripts Perl plus fous, le site `http://perlgolf.sourceforge.net` peut être intéressant.

8.6.21 Récupérer du texte ou une archive de liste de diffusion à partir d'une page web

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

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

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

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

8.6.22 Imprimer joliment une page web

Les commandes suivantes vont lire une page web et l'imprimer sur une imprimante/dans un fichier PostScript.

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

Voir '`lpr/lpd`' page 39. Regardez aussi `a2ps` et `mpage` pour créer des fichiers PostScript.

8.6.23 Imprimer joliment une page de manuel

Les commandes suivantes impriment une page de manuel sur une imprimante/un fichier PostScript.

```
$ man -Tps some-man-page | lpr
$ man -Tps some-man-page | mpage -2 | lpr
```

8.6.24 Joindre deux fichiers PostScript ou PDF

Vous pouvez joindre deux fichiers PostScript ou PDF.

```
$ 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
```

8.6.25 Mesurer la durée d'une commande

Affiche le temps pris par un processus.

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

8.6.26 Commande `nice`

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

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

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

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

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

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

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

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

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

8.6.28 Changement de console avec `screen`

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

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

Scénario d'accès distant

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

1 Supposez que vous êtes connectés via une connexion téléphonique, et que vous exécutez une session `screen` complexe avec des éditeurs et des programmes ouverts dans plusieurs fenêtres.

2. Vous avez alors besoin de quitter votre terminal, mais vous ne voulez pas perdre votre travail en raccrochant.
3. Tapez simplement `^A d` pour détacher la session, puis déconnectez-vous. (Ou encore plus rapide, tapez `^A DD` pour que `screen` détache la session et déconnecte lui-même.)
4. Quand vous vous connectez de nouveau, entrez la commande `screen -R`, et `screen` va automatiquement réattacher toutes les fenêtres que vous aviez ouvertes.

Commandes typiques de `screen`

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

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

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

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

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

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

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

Programme équivalent de `screen` pour X

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

8.6.29 Bases pour tester un réseau

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

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

8.6.30 Vider les courriers électroniques de la file locale

Pour vider la file locale des courriers électroniques présents :

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

`-qff` est peut-être une meilleure option pour le script `/etc/ppp/ip-up.d/exim`. Pour Sarge, remplacez `exim` par `exim4`.

8.6.31 Supprimer les courriers électroniques gelés de la file locale

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

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

Pour Sarge, remplacez `exim` par `exim4`.

8.6.32 Redélivrer le contenu de `mbox`

Vous devez redélivrer manuellement les courriels dans vos boîtes triées de votre répertoire personnel depuis le fichier `/var/mail/username` si votre répertoire personnel est devenu plein ou si `procmail` a planté. Après avoir fait de l'espace dans le répertoire personnel, lancez :

```
# /etc/init.d/exim stop
# formail -s procmail </var/mail/username
# /etc/init.d/exim start
```

Pour Sarge, remplacez `exim` par `exim4`.

8.6.33 Supprimer le contenu d'un fichier

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

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

8.6.34 Fichiers fantômes

Les commandes suivantes créent des fichiers fantômes ou vides.

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

Par exemple, les commandes suivantes exécutées depuis le shell de la disquette de dméarrage Debian efface tout le contenu du disque dur `/dev/hda`.

```
# dd if=/dev/urandom of=/dev/hda ; dd if=/dev/zero of=/dev/hda
```

8.6.35 `chroot`

Le programme `chroot`, `chroot(8)`, nous permet de lancer plusieurs environnements GNU/Linux simultanément sur un seul système sans redémarrer.

On peut aussi lancer un programme demandant des ressources comme `apt-get` ou `dselect` dans le chroot d'une machine rapide. On monte alors par NFS le système de fichiers d'une machine auxiliaire lente sur la machine rapide, en lecture/écriture, le point de montage étant le répertoire chroot.

Exécuter plusieurs versions de Debian avec `chroot`

Un environnment chroot Debian peut aisément être créé par la commande `debootstrap` de Woody. Par exemple, pour créer un chroot Sid sur */sid-root* avec une connexion Internet rapide :

```
main # cd / ; mkdir /sid-root
main # debootstrap sid /sid-root http://ftp.debian.org/debian/
... regardez le système se télécharger
main # echo "proc-sid /sid-root/proc proc none 0 0" >> /etc/fstab
main # mount proc-sid /sid-root/proc -t proc
main # cp /etc/hosts /sid-root/etc/hosts
main # chroot /sid-root /bin/bash
chroot # cd /dev; /sbin/MAKEDEV generic ; cd -
chroot # apt-setup # set-up /etc/apt/sources.list
chroot # vi /etc/apt/sources.list # mettre la source sur unstable
chroot # dselect  # ou aptitude, installez mc et vim :-)
```

A ce point, vous devriez avoir un systÃ¨me Debian complètement fonctionnel, avec lequel vous pouvez jouer sans avoir peur de toucher votre installation Debian principale.

Cette astuce avec `debootstrap` peut aussi être utilisée pour installer Debian sur un système dans utiliser le disque d'installation Debian mais celui d'une autre distribution GNU/Linux. Voir `http://www.debian.org/releases/stable/i386/apcs04`.

Configurer login pour `chroot`

Taper `chroot /sid-root /bin/bash` est facile, mais cela garde toutes les variables d'environnement, ce que vous pouvez ne pas vouloir, et apporte d'autres problèmes. Une meilleure approche est d'exécuter un autre processus login sur un terminal virtuel différent où vous pouvez vous connecter au chroot directement.

Puisque, sur les systèmes Debian par défaut, les consoles Linux sont de `tty1` à `tty6` et X Window System sur `tty7`, configurons une console chroot sur `tty8`. Après avoir créé un système chroot comme décrit dans 'Exécuter plusieurs versions de Debian avec `chroot`' de la présente page, exécutez ce qui suit en root sur le système principal :

```
main # echo "8:23:respawn:/usr/sbin/chroot /sid-root "\
       "/sbin/getty 38400 tty8"  >> /etc/inittab
main # init q    # reload init
```

Configurer X pour `chroot`

Vous voulez exécuter les dernières version de X et GNOME en toute sécurité dans votre chroot ? C'est possible ! L'exemple suivant permet d'exécuter GDM sur le terminal virtuel `vt9`.

D'abord, installez un système chroot comme décrit dans 'Exécuter plusieurs versions de Debian avec `chroot`' page précédente. Avec le root du système principal, copiez les fichiers de configration importants vers le système chroot.

```
main # cp /etc/X11/XF86Config-4 /sid-root/etc/X11/XF86Config-4
main # chroot /sid-root # ou utilisez la console chroot
chroot # cd /dev; /sbin/MAKEDEV generic ; cd -
chroot # apt-get install gdm gnome x-window-system
chroot # vi /etc/gdm/gdm.conf # faites s/vt7/vt9 dans la section [servers]
chroot # /etc/init.d/gdm start
```

Ici, `/etc/gdm/gdm.conf` a été édité pour que la console virtuelle de lancement soit `vt9` au lieu de `vt7`.

Maintenant vous pouvez facilement passer de l'environnement X du chroot à celui du système principal simplement en changeant de terminal virtuel, par exemple en utilisant Ctrl-Alt-F7 et Ctrl-Alt-F9. Amusez-vous bien !

[FIXME] Ajouter un commentaire et un lien vers le script `gdm` du chroot.

Exécuter d'autres distributions avec `chroot`

Un environnement chroot avec une autre distribution peut facilement être créé. Vous installez un système sur une partition séparée en utilisant son installateur. Si sa partition racine est *`/dev/hda9`* :

```
main # cd / ; mkdir /other-dist
main # mount -t ext3 /dev/hda9 /other-dist
main # chroot /other-dist /bin/bash
```

La suite est similaire à 'Exécuter plusieurs versions de Debian avec `chroot`' page précédente, 'Configurer login pour `chroot`' page précédente, et 'Configurer X pour `chroot`' page précédente.

Compiler un paquet avec `chroot`

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

8.6.36 Comment vérifier les liens durs

Vous pouvez vérifier si deux fichiers sont le même avec deux liens durs avec :

```
$ ls -li file1 file2
```

8.6.37 Monter une image de disque dur

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

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

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

8.6.38 Samba

Bases pour récupérer des fichiers depuis Windows :

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

Les voisins de Samba peuvent être trouvés depuis Linux :

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

8.6.39 Utilitaires pour les systèmes de fichiers étrangers

Un grand nombre de systèmes de fichiers est supporté par le noyau Linux, et peuvent être utilisés simplement en montant les périphériques les contenant. Pour certains systèmes de fichiers, il existe aussi des outils spécifiques permettant de les utiliser sans les monter. Ces opérations sont accomplies par des programmes en espace utilisateur, et donc le support du système de fichiers par le noyau n'est pas nécessaire.

- `mtools` : pour le système de fichiers MSDOS (MS-DOS, Windows)
- `cpmtools` : pour le système de fichiers CP/M
- `hfsutils` : pour le système de fichiers HFS (anciens Macintosh)
- `hfsplus` : pour le système de fichiers HFS+ (Macintosh modernes)

Pour créer et vérifier un système de fichiers FAT MSDOS, le paquet `dosfstools` est utile.

8.7 Fautes courantes à éviter

Voici quelques exemples d'actions dangereuses. Les aspects négatifs seront amplifiés si vous utilisez le compte `root`.

8.7.1 `rm -rf .*`

Dans `rm -rf .*`, "`.*`" est remplacé et contient "`.`" and "`..`", ce qui, si vous en avez les privilèges, supprime aussi tous les répertoires **à côté** du répertoire courant.

- "`rm -rf .`" : supprime tout sous le répertoire courant, le répertoire y compris.
- "`rm -rf *`" : supprime tous les fichiers et répertoires ne commençant pas par un point sous le répertoire courant.
- "`rm -rf .[^.]*`" : supprime tous les fichiers et répertoires commençant par un point sous le répertoire courant.
- "`rm -rf .*`" : supprime tout sous le répertoire parent et le répertoire parent lui-même.

8.7.2 `rm /etc/passwd`

La perte de fichiers importants comme `/etc/passwd` par stupidité est dur. Le système Debian en effectue des sauvegardes régulières dans `/var/backups/`. Lorsque vous restaurez ces fichiers, vous devez manuellement configurer les permissions.

```
# cp /var/backups/passwd /etc/passwd
# chmod 644 /etc/passwd
```

Voir aussi 'Récupérer les données de sélection des paquets' page 72.

Chapitre 9

Configurer un système Debian

Ce chapitre décrit seulement les bases de la configuration du système. Avant de lire ce chapitre, vous devriez lire 'Installation du système Debian' page 25.

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

9.1 Initialisation du système

Debian utilise le système de scripts d'initialisation System V. Voir 'Le programme `init`' page 21 pour une introduction.

9.1.1 Personnaliser les scripts d'initialisation

Le moyen le plus facile de contrôler le comportement d'un script d'initialisation est de changer les variables d'environnement dans le fichier situé dans le répertoire `/etc/default` ayant le même nom que le script d'initialisation. [1] Par exemple, `/etc/default/hotplug` peut-être utilisé pour contrôler comment `/etc/init.d/hotplug` fonctionne. Le fichier `/etc/default/rcS` peut être utilisé pour personnaliser les défauts au démarrage pour `motd`, `sulogin`, etc.

Si vous ne pouvez pas obtenir le comportement que vous souhaitez en changeant ces variables, alors vous pouvez modifier les scripts eux-mêmes : ce sont tous des fichiers de configuration.

---

[1] Les fichiers de `/etc/default` contiennent des définitions de variables d'environnement **seulement**. Chaque fichier est ajouté au script d'initialisation de telle façon que ces définitions remplacent des valeurs de variables par défaut définies dans le script d'initialisation lui-même. Le choix du nom du répertoire est particulier (`http://lists.debian.org/debian-devel/2003/debian-devel-200308/msg02114.html`) à Debian. C'est à peu près l'équivalent du répertoire `/etc/sysconfig` de RedHat ou d'autres distributions.

9.1.2 Personnaliser le journal du système

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

9.1.3 Optimisation de l'accès au matériel

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

- `hdparm`
  - Optimisation du disque dur. Très efficace.
  - Dangereux. Il faut lire `hdparm(8)` d'abord.
  - `hdparm -tT` *`/dev/hda`* pour tester la vitesse du disque.
  - `hdparm -q -c3 -d1 -u1 -m16` *`/dev/hda`* pour accélérer un système IDE moderne. (Cela peut être dangereux.)

- `setcd`
  - Optimisation de l'accès au lecteur de CD.
  - `setcd -x` *`2`* pour réduire à la vitesse 2x.
  - Voir `setcd(1)`.

- `setserial`
  - Collection d'outils pour gérer les ports série.

- `scsitools`
  - Collection d'outils pour gérer le matériel SCSI.

- `memtest86`
  - Collection d'outils pour gérer la mémoire.

- `hwtools`
  - Collection d'outils pour la gestion bas niveau du matériel.
    - `irqtune` : change la priorité d'IRQ des périphériques pour autoriser ceux qui ont besoin d'une priorité haute et d'un service rapide (par exemple ports série, modems) à l'avoir. Une accélération de 3 fois des débits du port série/modem est possible.
    - `scanport` : scanne l'espace E/S de 0x100 à 0x3ff pour trouver les périphériques ISA installés.
    - `inb` : un petit programme rapide qui lit un port E/S et retourne la valeur en hexadécimal et en binaire.

- `schedutils`
  - Utilitaires pour l'ordonnanceur Linux.
  - `taskset`, `irqset`, `lsrt` et `rt` sont inclus.
  - Avec `nice` et `renice` (non inclus), ils permettent un contrôle complet des paramètres de l'ordonnancement des processus.

Monter un système de fichiers avec l'option `noatime` permet aussi d'accélérer les accès en lecture au fichier. Voir `fstab(5)` et `mount(8)`.

Certains matériels peuvent être optimisés directement dans le noyau Linux lui-même à travers le système de fichiers proc. Voir 'Trop de fichiers ouverts' page 91.

Il existe beaucoup d'utilitaires de configuration pour du matériel spécifique dans Debian. Beaucoup d'entre eux répondent à des besoins spécifiques des portables. Voici quelques paquets intéressants disponibles dans Debian.

- `tpconfig` - Utilitaires pour Advanced Power Management (APM)
- `acpi` - Affiche des informations sur les périphériques ACPI
- `acpid` - Utilitaires pour utiliser ACPI
- `lphdisk` - Prépare une partition d'hibernation pour le NoteBIOS Phoenix
- `sleepd` - Met un portable en veille en cas d'inactivité
- `noflushd` - Permet aux disques durs de s'arrêter
- `big-cursor` - Curseurs de souris plus grands pour X
- `acme` - Active les boutons multimedia des portables
- `tpctl` - Ouils de configuration pour les IBM ThinkPad
- `mwavem` - Support pour les modems Mwave/ACP
- `toshset` - Accède à la plupart de l'interface matériel des portables Toshiba
- `toshutils` - Utilitaires pour portables Toshiba
- `sjog` - Un programme pour utiliser le Jog Dial sur les portables Sony Vaio
- `spicctrl` - Programme pour régler la luminosité de l'écran LCD sur Sony Vaio

ACPI est un modèle de gestion d'énergie plus récent qu'APM.

Certains de ces paquets ont besoin de modules du noyau spéciaux, qui sont inclus dans les dernières sources du noyau dans la plupart des cas. En cas de problème, vous aurez peut-être besoin de patcher le noyau vous-même.

9.2 Contrôle d'accès

9.2.1 Contrôle des connexions avec PAM

PAM (*Pluggable Authentication Modules*, Modules d'Authentification Greffables) permet de contrôler les connexions des utilisateurs.

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

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

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

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

Au contraire, installez `cracklib2` et réglez `/etc/pam.d/passwd` comme suit, si vous voulez forcez une bonne politique de mots de passe.

```
password required       pam_cracklib.so retry=3 minlen=6 difok=3
```

Pour l'activation d'un compte, un mot de passe temporaire peut aussi être utile. Pour cela utilisez la commande `passwd` avec l'option `-e`. Voir `passwd(1)`.

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

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

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

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

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

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

9.2.3 Objet des groupes par défaut

Quelques groupes intéressants :
- Le groupe `root` est le groupe `wheel` par défaut pour `su` si `pam_wheel.so` est utilisé sans l'argument `group=`.
- Le groupe `adm` peut lire les fichiers du journal.

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

Pour une liste complète, voir la section « FAQ » dans Securing Debian Manual (`http://www.debian.org/doc/manuals/securing-debian-howto/`), qui se trouve aussi dans le paquet `harden-doc` dans Woody. De plus, le nouveau paquet `base-passwd` (>3.4.6) contient une liste officielle :`/usr/share/doc/base-passwd/users-and-groups.html`.

9.2.4 Travailler de façon sûre – `sudo`

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

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

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

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

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

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

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

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

Un autre programme permet à des utilisateurs ordinaires de lancer des commandes avec les privilèges de root, voir le paquet `super`.

9.2.5 Contrôle d'accès aux services

Le *super-serveur* internet, `inetd`, est exécuté lors du démarrage par `/etc/rc2.d/S20inetd` (pour le niveau d'exécution 2) qui est un lien vers `/etc/init.d/inetd`. `inetd` permet de n'exécuter qu'un démon qui en invoque plusieurs autres, réduisant ainsi la charge du système.

Lorsqu'une requête pour un service arrive, le protocole et le service sont identifiés en regardant dans la base de données stockée dans `/etc/protocols` et `/etc/services`. Pour un service internet normal, `inetd` utilise `/etc/inetd.conf`. Pour un service basé sur Sun-RPC, `inetd` utilise `/etc/rpc.conf`.

Pour la sécurité du système, assurez-vous de désactiver les services non utilisés dans `/etc/inetd.conf`. Les services Sun-RPC doivent être actifs pour NFS et d'autres programmes basés sur RPC.

Parfois, `inetd` n'exécute pas le serveur demandé directement, mais exécute le programme d'enveloppe de démon TCP/IP `tcpd` avec le serveur demandé comme argument dans `/etc/inetd.conf`. Dans ce cas, `tcpd` lance le serveur approprié après avoir enregistré dans le journal la requête et avoir fait quelques autres vérifications en utilisant les fichiers `/etc/hosts.deny` et `/etc/hosts.allow`.

Si vous avez un problème avec l'accès à distance sur un système Debian récent, commentez la ligne "ALL : PARANOID" dans `/etc/hosts.deny` si elle existe.

Pour plus de détails, voir `inetd(8)`, `inetd.conf(5)`, `protocols(5)`, `services(5)`, `tcpd(8)`,`hosts_access(5)`,et `hosts_options(5)`.

Pour plus d'information sur Sun-RPC, voir `rpcinfo(8)`,`portmap(8)`,et `/usr/share/doc/portmap/portmapper.txt.gz`.

9.2.6 Centraliser l'identification – LDAP

Utilisez *Lightweight Directory Access Protocol* (LDAP). Références :

- OpenLDAP (`http://www.openldap.org/`)
- OpenLDAP Admin Guide dans le paquet `openldap-guide`
- LDP : LDAP Linux HOWTO (`http://www.tldp.org/HOWTO/LDAP-HOWTO/index.html`)
- LDP : LDAP Implementation HOWTO (`http://www.tldp.org/HOWTO/LDAP-Implementation-HOWTO/index.html`)
- OpenLDAP, extensive use reports (`http://portal.aphroland.org/~aphro/ldap-docs/ldap.html`)
- Open LDAP with Courier IMAP and Postfix (`http://alinux.washcoll.edu/docs/plc/postfix-courier-howto.html`)

9.3 Graveurs de CD

Les graveurs de CD sur interface ATAPI/IDE sont devenus des périphériques courants. Le CD un bon média pour sauvegarder et archiver un système pour un utilisateur individuel ayant besoin d'une capacité < 640 Mo. Pour des informations plus officielles, consultez le CD-Writing-HOWTO (`http://www.tldp.org/HOWTO/CD-Writing-HOWTO.html`) du LDP.

9.3.1 Introduction

D'abord, sachez qu'une interruption de l'envoi des données vers le graveur entraînera des erreurs irrécupérables sur le CD. Prenez donc un graveur avec un cache aussi gros que possible. Si le prix n'a pas d'importance, optez pour une version SCSI plutôt qu'ATAPI/IDE. Si vous avez le choix de l'interface IDE sur laquelle connecter le graveur, préférez celle sur bus PCI (une de celles de la carte mère) plutôt qu'une sur bus ISA (carte SB16, etc.).

Un graveur connecté sur interface IDE doit être piloté par le pilote IDE-SCSI et non par le pilote CD IDE habituel pour les noyaux Linux 2.2 et 2.4. Le pilote SCSI générique doit aussi être activé. Pour cela, deux approches existent pour les noyaux distribués avec des distributions modernes (à la date de mars 2001).

Pour le noyau Linux 2.6, vous devriez utiliser le pilote IDE et accéder au graveur de CD directement avec le nom de périphérique `/dev/hdx`. Vous pouvez utiliser le DMA de cette façon.

9.3.2 Première approche : modules + `lilo`

Ajoutez la ligne suivante à `/etc/lilo.conf` si vous utilisez le noyau distribué avec Debian. Si vous utilisez plusieurs options, listez-les en les séparant par des espaces.

```
append="hdx=ide-scsi ignore=hdx"
```

Ici, le graveur, accédé par le pilote ide-scsi, est identifié par `hdx`, où *x* représente :

```
hdb      esclave sur le premier port IDE
hdc      maître sur le second port IDE
hdd      esclave sur le second port IDE
hde ... hdh  sur un port IDE externe
```

Lancez les commandes suivantes, en tant que root, pour activer les options après avoir terminé la configuration :

```
# lilo
# shutdown -h now
```

9.3.3 Seconde approche : recompiler le noyau

Debian utilise `make-kpkg` pour créer le noyau. Utilisez la nouvelle option `--append_to_version` avec `make-kpkg` pour créer plusieurs images du noyau. Voir 'Le noyau Linux et Debian' page 85.

Utilisez la configuration suivante avec `make menuconfig` :

- bzImage
- Excluez le pilote CD IDE (non obligatoire mais c'est plus simple comme ça)
- Ajoutez ide-scsi et sg, ou compilez-les en tant que modules

9.3.4 Étapes post-configuration

Le support pour le graveur peut être activé dans le noyau lors du démarrage avec les commandes suivantes :

```
# echo ide-scsi >>/etc/modules
# echo sg       >>/etc/modules
# cd /dev; ln -sf scd0 cdrom
```

Une activation manuelle peut aussi être effectuée avec :

```
# modprobe ide-scsi
# modprobe sg
```

Après un redémarrage, vous pouvez vérifier l'installation avec :

```
$ dmesg|less
# apt-get install cdrecord
# cdrecord -scanbus
```

[Par Warren Dodge] Quelquefois, il peut y avoir des conflits entre `ide-scsi` et `ide-cd` s'il y a à la fois un lecteur de CDROM et un graveur CD-R/RW sur le système. Essayez d'ajouter ce qui suit dans votre fichier `/etc/modutils/aliases`, exécutez `update-modules` et redémarrez.

```
pre-install      ide-scsi      modprobe ide-cd
```

Cela force le pilote IDE à être chargé avant `ide-scsi`. Le pilote `ide-cd` prend le contrôle du lecteur de CDROM ATAPI—tout ce qu'on ne lui a pas dit d'**ignorer**. Cela laisse simplement les périphériques ignorés au contrôle de `ide-scsi`.

9.3.5 Image de CD (bootable)

Pour créer un CDROM contenant des fichiers contenus dans le répertoire `target-directory/` en tant qu'image nommée `cd-image.raw` (CD bootable, format Joliet TRANS.TBL. Pour un CD non bootable, enlevez les options `-b` et `-c`), insérez une disquette de démarrage dans le premier lecteur de disquette et faites :

```
# dd if=/dev/fd0 target-directory/boot.img
# mkisofs -r -V volume_id -b boot.img -c bootcatalog -J -T \
        -o cd-image.raw target_directory/
```

Une utilisation amusante est de créer un CDROM DOS bootable. Si *`boot.img`* contient l'image d'une disquette de démarrage DOS, le CDROM démarrera comme si une disquette DOS était dans le premier lecteur de disquette (A :). Faire cela avec freeDOS est encore plus intéressant.

L'image créée peut être vérifiée en la montant en utilisant le périphérique de boucle (loop device).

```
# mount -t iso9660 -o ro,loop cd-image.raw /cdrom
# cd /cdrom
# mc
# umount /cdrom
```

9.3.6 Graver un CD (R, RW)

Premier test avec les commandes suivantes (cas d'un graveur double vitesse)

```
# nice --10 cdrecord -dummy speed=2 dev=0,0 disk.img
```

Si le test est réussi, gravez un CD-R avec

```
# nice --10 cdrecord -v -eject speed=2 dev=0,0 disk.img
```

Ou gravez un CD-RW avec

```
# nice --10 cdrecord -v -eject blank=fast speed=2 dev=0,0 disk.img
```

Certains graveurs de CD marchent mieux avec

```
# nice --10 cdrecord -v blank=all speed=2 dev=0,0 disk.img
```

suivi de

```
# nice --10 cdrecord -v -eject speed=2 dev=0,0 disk.img
```

Deux étapes sont nécessaires pour empêcher que des fins d'attente SCSI lors du formatage n'interfèrent avec l'étape de gravure. Les arguments passés à `nice` peuvent avoir besoin d'un ajustement.

9.3.7 Créer l'image d'un CD

Certains CD-R et CD commerciaux ont des secteurs parasites à la fin qui les rendent incopiables par `dd` (le CD de Windows 98 est l'un d'eux). Le paquet `cdrecord` comprend la commande `readcd`. Utilisez-la pour copier le contenu de n'importe quel CD dans un fichier image. S'il s'agit d'un CD de données, montez-le et exécutez `mount` pour connaître sa taille. Divisez le nombre affiché (en blocs de 1024 octets) par 2 pour connaître le nombre de secteurs du CD (2048 octets). Lancez `readcd` avec les bonnes options et utilisez l'image du CD pour graver un CD-R/RW.

```
# readcd target,lun,scsibusno # choisissez la fonction 11
```

Ici, positionnez tous les paramètres de ligne de commande à 0 dans la plupart des cas. Parfois, le nombre de secteurs donné par `readcd` est trop important ! Utilisez le nombre donné par mount pour de meilleurs résultats.

Il faut noter que l'utilisation de `dd` pose des problèmes avec un CD-ROM. La première exécution de `dd` peut causer un message d'erreur et entraîner une image disque plus petite avec la perte de la fin. La deuxième exécution de `dd` peut entraîner une image disque plus grande avec des données à jeter à la fin sur certains systèmes si la taille des données n'est pas spécifiée. Seule la seconde exécution de `dd` avec la taille des données correcte spécifiée, et sans éjecter le CD après un message d'erreur, semble permettre d'éviter ces problèmes. Si par exemple la taille de l'image affichée par `df` est `46301184` blocs, utilisez la commande suivante pour obtenir la bonne image disque (information empirique) :

```
# dd if=/dev/cdrom of=cd.img bs=2048 count=$((46301184/2))
```

9.3.8 Images de CD Debian

Pour obtenir les dernières informations, consultez le site Debian CD. (`http://cvs.debian.org/debian-cd/`)

Si vous avez une connexion Internet rapide, installez par le réseau en utilisant :

- quelques images de disquettes (`http://www.debian.org/distrib/floppyinst`).
- une image de CD minimale (`http://www.debian.org/CD/netinst/`).

Si vous n'avez pas de connexion Internet rapide, achetez des CD depuis un vendeur de CD (`http://www.debian.org/CD/vendors/`).

Merci de ne pas gâcher la bande passante en téléchargeant des images de CD standard à moins que vous ne soyez un testeur d'images de CD (même avec la nouvelle méthode jigdo).

Une image de CD à noter est celle de KNOPPIX - Système de Fichier Linux Fonctionnel sur CD (`http://www.knopper.net/knoppix/index-en.html`). Ce CD démarre avec un système Debian fonctionnel sans s'installer sur le disque dur.

9.3.9 Système de sauvegarde sur CD-R

Pour copier les fichiers de configuration et de données importants sur CD-R, utilisez le script de sauvegarde `backup` (`http://www.debian.org/doc/manuals/debian-reference/examples/`). Voir aussi 'Copier et archiver un sous-répertoire entier' page 97 et 'Sauvegarde différentielle et synchronisation de données' page 99.

9.3.10 Copier un CD audio sur un CD-R

Je n'ai pas testé cela personnellement :

```
# apt-get get install cdrecord cdparanoia
# cdparanoia -s -B
# cdrecord dev=0,0,0 speed=2 -v -dao -eject defpregap=1 -audio *.wav
```

ou,

```
# apt-get install cdrdao #disk at once
# cdrdao read-cd --device /dev/cdrom --paranoia-mode 3 my_cd # lit le CD
# cdrdao write --device /dev/cdrom --speed 8 my_cd    # grave un nouveau CD
```

`cdrdao` fait une vraie copie (pas de trous, etc.).

9.3.11 Graver des DVD-R, DVD-RW et DVD+RW

Pour graver des DVD, il existe deux approches :

- Utiliser `growisofs` avec `mkisofs`.
- Recompiler `cdrecord` avec l'option dvd pour créer un paquet local en suivant `/usr/share/doc/cdrecord/README.DVD.Debian`.

9.4 X

L'environnement X Window System est fourni par Xfree86 (`http://www.xfree86.org/`). Il existe deux versions majeures du serveur X dans Debian : XFree86 Version 3.3 (XF3) et la série des XFree86 Version 4.x (XF4) toutes les deux basées sur les spécifications X11R6 de X.Org (`http://www.x.org/`).

Pour les bases sur X, référez-vous à `X(7)`, et les XWindow-User-HOWTO (`http://www.tldp.org/HOWTO/XWindow-User-HOWTO.html`) et Remote X Apps mini-HOWTO (`http://www.tldp.org/HOWTO/mini/Remote-X-Apps.html`) du LDP. Pour un guide d'utilisation spécifique à Debian, lisez `/usr/share/doc/xfree86-common/FAQ.gz` fournie dans le paquet `xfree86-common`. Elle contient une revue intéressante et exhaustive des problèmes d'assignation des touches par Branden Robinson.

'Le serveur X' page ci-contre un programme sur la machine locale qui affiche des fenêtres X et/ou un bureau sur le moniteur (CRT, LCD) d'un utilisateur et accepte des entrées au clavier et à la souris.

'Clients X' page 135 un programme sur une machine (locale ou distante) qui exécute un logiciel d'application compatible avec X-Window.

Cela inverse l'utilisation habituelle de « serveur » et « client » dans d'autres contextes.

Il y a plusieurs façons de faire accepter les connexions distantes d'un client X (côté application) à un serveur X (côté affichage) :

- Méthode `xhost`
  - le mécanisme de liste d'hôtes (non sûr).
  - protocole non chiffré (sujet à des attaques d'écoute de ligne).
  - à ne pas utiliser, si possible.
  - voir 'Connexion à un serveur X distant – `xhost`' page 140 et `xhost(1x)`.
- Méthode **`xauth`**
  - le mécanisme des cookies magiques du MIT (non sûr mais mieux que `xhost`).
  - protocole non chiffré (sujet à des attaques d'écoute de ligne).
  - à utiliser seulement pour des connexions locales car il est moins gourmand en temps processeur que `ssh -X`.
  - voir 'Devenir root sous X' page 142 et `xauth(1x)`.
- Méthodes `xdm`, `wdm`, `gdm`, `kdm`,…
  - mécanisme des cookies magiques du MIT (aussi non sûr que `xauth`).
  - voir `xdm(1x)` et `Xsecurity(7)` pour les bases du contrôle d'accès à un affichage X.
  - voir `wdm(1x)`, `gdm(8)`, et `kdm.options(5)` pour plus d'information, si ces programmes sont installés.
  - voir 'Personnaliser les niveaux de fonctionnement' page 22 pour savoir comment désactiver `xdm` pour avoir accès à la console Linux après le démarrage sans purger le paquet `xdm`.
- Méthode **`ssh -X`**
  - mécanisme de redirection de port à travers SSH (**sûr**).
  - protocole chiffré (gâchis de ressources en utilisation locale).
  - à utiliser pour des connexions distantes.
  - voir 'Connexion à un serveur X distante – `ssh`' page 140.

Toutes les méthodes de connexion distante, sauf `ssh`, ont besoin de l'activation des connexions TCP/IP au serveur X. Voir 'Utiliser X sur TCP/IP' page 139.

9.4.1 Paquets X

Il existe quelques paquets globaux pour installer X sur un système Woody.

`x-window-system-core` Ce paquet global fournit les composants essentiels pour une station de travail exécutant X Window System. Il fournit les bibliothèques X, un serveur X `xserver-xfree86`, un ensemble de polices, et un ensemble de clients et utilitaires X de base.

`x-window-system` Ce paquet global fournit à peu près tous les composants de X Window System tels que développés par XFree86 Project, ainsi qu'un ensemble d'accessoires historiquement populaires. (Notamment, il dépend de `x-window-system-core`, `twm`, et `xdm`, càd pas besoin d'installer `x-window-system-core` si vous installez celui-ci.)

`xserver-common-v3` Fichiers et utilitaires communs aux serveurs X XFree86 3.x (XF3)

`xserver-*` Paquets supplémentaires pour le serveur XF3 afin de supporter du matériel non supporté par le nouveau serveur XF4 (`xserver-xfree86`), qu'elle qu'en soit la raison. Certaines cartes ATI mach64 anciennes ne sont pas supportées par XF4, certaines cartes vidéo plantent avec la version de XF4 de Woody, etc. (Pour connaître les paquets disponibles, utilisez `apt-cache search xserver-|less`. Tous les serveurs XF3 dépendent de `xserver-common-v3`.)

Dans la plupart des cas, `x-window-system` est le paquet à installer. (Si vous voulez vous connecter par la console, assurez-vous de désactiver `xdm` tel que décrit dans '« Je veux désactiver X au démarrage ! »' page 95.)

9.4.2 Détection de matériel pour X

Pour activer la détection du matériel lors de la configuration de X, installez les paquets suivants avant d'installer le système X :

- `discover` – système d'identification du matériel
- `mdetect` – outil d'autodétection de la souris
- `read-edid` – outil pour récupérer des informations matérielles pour les moniteurs VESA PnP.

9.4.3 Le serveur X

Consultez `XFree86(1)` pour des informations sur le serveur X.

Exécution du serveur X depuis une console locale :

```
$ startx -- :<display> vtXX
par exemple :
$ startx -- :1 vt8 -bpp 16
... exécution sur un terminal vt8 connecté à localhost:1 en mode 16 bpp
```

Les arguments donnés après `--` sont pour le serveur X.

Notez que lorsque vous utilisez le script `~/.xserverrc` pour personnaliser le démarrage du serveur X, il faut utiliser `exec` avec le serveur X réel. Sinon, cela peut rendre le serveur X lent à démarrer ou à quitter. Par exemple :

```
#!/bin/sh
exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

Configurer le serveur X (version 4)

Pour (re-)configurer un serveur XF4,

```
# dpkg-reconfigure --priority=low xserver-common
# dpkg-reconfigure --priority=low xserver-xfree86
```

générera le fichier `/etc/X11/XF86Config-4` et configurera X avec le script `dexconf`.

Configurer le serveur X (version 3)

Pour (re-)configurer un serveur XF3, par exemple pour une carte ATI mach64,

```
# dpkg-reconfigure --priority=low xserver-common-v3
# dpkg-reconfigure --priority=low xserver-mach64
```

générera le fichier `/etc/X11/XF86Config` et configurera X avec le script `xf86config-v3`.

Configurer manuellement le serveur X

Avec Woody, pour ajouter des personnalisations à `/etc/X11/XF86Config-4`, **n'éditez pas le fichier de configuration entre le texte** :

```
### BEGIN DEBCONF SECTION
[snip]
### END DEBCONF SECTION
```

Au lieu de cela, **ajoutez les personnalisations avant le texte**. Par exemple, pour utiliser une carte video personnalisée, ajoutez quelquechose comme ce qui suit au début du fichier :

```
Section "Device"
  Identifier        "Custom Device"
  Driver            "ati"
  Option            "NoAccel"
EndSection

Section "Screen"
  Identifier  "Custom Screen"
  Device      "Custom Device"
  Monitor     "Generic Monitor"
  DefaultDepth 24
  Subsection "Display"
```

```
    Depth       8
    Modes       "1280x960" "1152x864" "1024x768" "800x600" "640x480"
  EndSubsection
  Subsection "Display"
    Depth       16
    Modes       "1280x960" "1152x864" "1024x768" "800x600" "640x480"
  EndSubsection
  Subsection "Display"
    Depth       24
    Modes       "1280x960" "1152x864" "1024x768" "800x600" "640x480"
  EndSubsection
EndSection

Section "ServerLayout"
    Identifier       "Custom"
    Screen           "Custom Screen"
    InputDevice      "Generic Keyboard" "CoreKeyboard"
    InputDevice      "Configured Mouse" "CorePointer"
EndSection
```

Pour Sarge (`testing` au moment de l'écriture), si vous souhaitez garder les personnalisations de `/etc/X11/XF86Config` à travers la mise-à-jour, lancez les commandes suivantes en tant que root :

```
# cp /etc/X11/XF86Config-4 /etc/X11/XF86Config-4.custom
# md5sum /etc/X11/XF86Config-4 > /var/lib/xfree86/XF86Config-4.md5sum
# dpkg-reconfigure xserver-xfree86
```

Afin d'obtenir de **belles polices de caractères**, vous devez éditer `/etc/X11/XF86Config-4` comme décrit dans 'Polices TrueType dans X' page 144.

Vérifiez aussi les autres parties de votre configuration X. De mauvais paramètres pour le moniteur peuvent donner un mal de tête plus important que de mauvaises polices, aussi assurez-vous que le taux de rafraichissement est aussi élevé que peut le supporter votre moniteur (85Hz est super, 75Hz est OK, 60Hz fait mal aux yeux).

9.4.4 Clients X

La plupart des clients X peuvent être démarrés comme ceci :

```
client $ xterm -geometry 80x24+30+200 -fn 6x10 -display hostname:0 &
```

Ici, les arguments de ligne de commande optionnels signifient :

– `-geometry` *WIDTH*x*HEIGHT*+*XOFF*+*YOFF* : taille initiale et emplacement de la fenêtre.

- `-fn` *`FONTNAME`* : police utilisée pour l'affichage du texte. *`FONTNAME`* peut être :
  - a14 : Taille normale
  - a24 : Grande taille
  - … (consultez les polices disponibles avec `xlsfont`.)
- `-display` *`displayname`* : le nom du serveur X à utiliser. *`displayname`* peut être :
  - *`hostname`*`:`*`D.S`* signifie écran *S* sur affichage *D* de l'hôte `hostname` ; le serveur X de cet affichage écoute le port TCP 6000+D.
  - *`host`*`/unix:`*`D.S`* signifie écran *S* sur affichage *D* de l'hôte `host` ; le serveur X de cet affichage écoute la socket UNIX `/tmp/.X11-unix/XD` (et n'est donc utilisable depuis *`host`*).
  - *`:D.S`* est équivalent à *`host`*`/unix:`*`D.S`*, où `host` est le nom de l'hôte local

Le *`displayname`* (nom d'affichage) par défaut pour un programme client X (côté application) peut être défini par la variable d'environnement DISPLAY. Par exemple, avant de lancer un programme client X, on peut exécuter l'une des commandes suivantes pour cela :

```
$ export DISPLAY=:0
        # Valeur par défaut, machine locale avec premier écran X
$ export DISPLAY=hostname.fulldomain.name:0.2
$ export DISPLAY=localhost:0
```

Le démarrage peut être personnalisé avec le fichier `~/.xinitrc`. Par exemple :

```
xrdb -load $HOME/.Xresources
xsetroot -solid gray &
xclock -g 50x50-0+0 -bw 0 &
xload -g 50x50-50+0 -bw 0 &
xterm -g 80x24+0+0 &
xterm -g 80x24+0-0 &
twm
```

Comme décrit dans 'Sessions X personnalisées' page suivante, cela outrepasse tout ce qu'une exécution normale de Xsession fait lorsqu'elle est démarrée depuis `startx`. Utilisez plutôt `~/.xsession` et utilisez cette approche seulement en dernier recours. Voir `xsetroot(1x)`, `xset(1x)` and 'Ressources X' page 141.

9.4.5 Sessions X

Une session X (serveur X + client X) peut être lancée par :

- `startx` : script encapsulant `xinit` pour démarrer le serveur et le client X depuis la console de Linux. Si `~/.xinitrc` n'existe pas, `/etc/X11/Xsession` est exécuté au travers de `/etc/X11/xinit/xinitrc`.
- `xdm`, `gdm`, `kdm`, ou `wdm` : démon de gestion de l'affichage X pour démarrer le serveur et le client X, et contrôler les connexions depuis un écran graphique. `/etc/X11/Xsession` est exécuté directement.

La console peut être rappelée par '« Je veux désactiver X au démarrage ! »' page 95.

Sessions X personnalisées

Le script de démarrage par défaut `/etc/X11/Xsession` est une combinaison de `/etc/X11/Xsession.d/50xfree86-common_determine-startup` et `/etc/X11/Xsession.d/99xfree86-common_start`.

L'exécution de `/etc/X11/Xsession` est affectée par `/etc/X11/Xsession.options` et consiste essentiellement en l'exécution d'un programme avec la commande `exec` trouvé en premier, en cherchant dans l'ordre suivant :

1 `~/.xsession` ou `~/.Xsession`, s'il existe.
2 `/usr/bin/x-session-manager`, s'il existe.
3 `/usr/bin/x-window-manager`, s'il existe.
4 `/usr/bin/x-terminal-emulator`, s'il existe.

La signification exacte de ces commandes est déterminée par le système de solutions de rechange de Debian décrit dans 'Commandes de rechange' page 82. Par exemple :

```
# update-alternatives --config x-session-manager
... ou
# update-alternatives --config x-window-manager
```

Pour que n'importe quel gestionnaire de fenêtre soit le défaut, tout en gardant les gestionnaires de session de GNOME et KDE installés, remplacez le fichier `/etc/X11/Xsession.d/50xfree86-common_determine-startup` par celui attaché au deuxième rapport de bogue à `http://bugs.debian.org/168347` (j'espère que cela sera inclus rapidement) et éditez `/etc/X11/Xsession.options` comme suit pour désactiver le gestionnaire de session X :

```
# /etc/X11/Xsession.options
#
# configuration options for /etc/X11/Xsession
# See Xsession.options(5) for an explanation of the available options.
# Default enabled
allow-failsafe
allow-user-resources
allow-user-xsession
use-ssh-agent
# Default disabled (enable them by uncommenting)
do-not-use-x-session-manager
#do-not-use-x-window-manager
```

Sans la modification ci-dessus, `gnome-session` et `kdebase` sont les paquets contenant ces gestionnaires de session X. Les enlever permet le gestionnaire de fenêtre d'être le défaut. (Beurk, une autre idée ?)

Sur un système où `/etc/X11/Xsession.options` contient une ligne `allow-user-xsession` sans caractère la précédent, n'importe quel utilisateur peut personnaliser l'action de `/etc/X11/Xsession` en créant `~/.xsession` ou `~/.Xsession`.

La dernière commande dans le fichier `~/.xsession` devrait être de la forme `exec` *`some-window/session-manager`* pour démarrer votre gestionnaire de fenêtres/session favori.

Un bon exemple de script `~/.xsession` est donné par `/usr/share/doc/xfree86-common/examples/xsession.gz`.

J'utilise ceci pour définir le gestionnaire de fenêtres, l'accès à l'écran et le support de la langue pour chaque compte utilisateur. Voir 'Démarrer une session X pour un utilisateur' de la présente page, 'Devenir root sous X' page 142, et 'Exemple de système X bilingue' page 159.

Si vous souhaitez avoir plusieurs clients X lancés automatiquement, voir les exemples de 'Clients X' page 135 et appelez-les depuis `~/.xsession` plutôt que `~/.xinitrc`.

D'autres ressources X spécifiques à chaque utilisateur peuvent être stockées dans `~/.Xresources`. Voir 'Ressources X' page 141.

Des cartes clavier personnalisées et des actions spécifiques associées aux boutons de la souris peuvent être spécifiés dans le script de démarrage de l'utilisateur. Voir 'Clavier et souris sous X' page 141.

Démarrer une session X pour un utilisateur

Suivant les principes décrits à 'Sessions X personnalisées' page précédente, un gestionnaire de session/fenêtres X spécifique à chaque utilisateur peut être activé en installant le paquet indiqué et en configurant le fichier `~/.xsession` comme suit. (J'aime bien `blackbox/fluxbox` pour son style simple et sa rapidité.) :

- gestionnaire de session X par défaut.
  - Voir 'Commandes de rechange' page 82.
  - `exec /usr/bin/x-session-manager`
- gestionnaire de fenêtres X par défaut.
  - Voir 'Commandes de rechange' page 82.
  - `exec /usr/bin/x-window-manager`
- gestionnaire de session GNOME (lourd)
  - Installer le paquet : `gnome-session`
  - `exec /usr/bin/gnome-session`
- gestionnaire de session KDE (lourd)
  - Installer le paquet : `kdebase` (ou `kdebase3` pour KDE3)
  - `exec /usr/bin/kde2`
- gestionnaire de fenêtres Blackbox (léger)
  - Installer le paquet : `blackbox`
  - `exec /usr/bin/blackbox`

- gestionnaire de fenêtres Fluxbox (léger, nouveau Blackbox)
  - Installer le paquet : `fluxbox`
  - `exec /usr/bin/fluxbox`
- gestionnaire de fenêtres Xfce (Ressemble à Mac OS-X, SUN CDE)
  - Installer le paquet : `xfce`
  - `exec /usr/bin/xfwm`
- gestionnaire de fenêtres IceWM (léger, alternative à GNOME)
  - Installer le paquet : `icewm`
  - `exec /usr/bin/X11/icewm`
- gestionnaire de fenêtres virtuel FVWM2 (léger, ressemble à Windows 95)
  - Installer le paquet : `fvwm`
  - `exec /usr/bin/fvwm2`
- gestionnaire de fenêtres Windowmaker (ressemble à NeXT)
  - Installer le paquet : `wmaker`
  - `exec /usr/bin/wmaker`
- gestionnaire de fenêtres Enlightenment (lourd)
  - Installer le paquet : `enlightenment`
  - `exec /usr/bin/enlightenment`

Voir Window Managers for X (`http://www.xwinman.org`).

Configurer KDE et GNOME

Afin de configurer un environnement KDE ou GNOME complet, les meta-paquets suivants sont utiles :

- KDE : installer le paquet `kde`
- GNOME : installer le paquet `gnome`

En installant ces paquets avec des outils qui gèrent la priorité « `recommends` », comme `dselect` et `aptitude`, vous aurez un choix de logiciel plus riche qu'en les installant avec `apt-get`.

Si vous voulez une connexion à la console, assurez-vous de désactiver les gestionnaires d'affichage X, comme `kdm`, `gdm`, et `wdm`, qui peuvent être installés par le système de dépendances, en suivant les instructions décrites à '« Je veux désactiver X au démarrage ! »' page 95.

Si vous voulez avoir GNOME par défaut plutôt que KDE, configurez `x-session-manager` comme décrit à 'Commandes de rechange' page 82.

9.4.6 Utiliser X sur TCP/IP

Comme les connexions TCP/IP distantes sans chiffrage peuvent être sujettes à des attaques d'écoute de ligne, la configuration par défaut de X de Debian désactive les sockets TCP/IP. Pensez à utiliser `ssh` pour des connexions X distantes (Voir 'Connexion à un serveur X distante – `ssh`' page suivante).

La méthode décrite ici n'est pas encouragée à moins que l'on soit dans un environnement très sûr derrière un bon pare-feu et en présence d'utilisateurs de confiance uniquement. La configuration suivante sur le serveur X restaure les connexions TCP/IP :

```
# find /etc/X11 -type f -print0 | xargs -0 grep nolisten
/etc/X11/xinit/xserverrc:exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

Supprimez `-nolisten` pour permettre les connexions TCP/IP à X.

9.4.7 Connexion à un serveur X distant – `xhost`

`xhost` autorise l'accès en se basant sur les noms d'hôtes, ce qui n'est pas sûr du tout. Les commandes suivantes permettent de désactiver la vérification de l'hôte et autorisent des connexions de n'importe où si les connexions TCP/IP sont autorisées (voir 'Utiliser X sur TCP/IP' page précédente) :

```
$ xhost +
```

Vous pouvez réactiver la vérification de l'hôte avec :

```
$ xhost -
```

`xhost` ne fait pas la différence entre les utilisateurs sur l'hôte distant. De plus, les noms d'hôtes (les adresses en fait) peuvent être falsifiés.

Cette méthode doit être évitée même avec des critères sur les hôtes plus restrictifs si vous êtes sur un réseau dans lequel vous ne pouvez avoir confiance (par exemple avec une connexion PPP sur Internet). Voir `xhost(1x)`.

9.4.8 Connexion à un serveur X distante – `ssh`

L'utilisation de `ssh` active une connexion sûre d'un serveur X local vers un serveur d'applications distant.

- Réglez `X11Forwarding` et `AllowTcpForwarding` à `yes` dans `/etc/ssh/sshd_config` sur l'hôte distant.
- Démarrez le serveur X sur la station locale.
- Démarrez un `xterm` sur l'hôte local.
- Lancez `ssh` pour établir une connexion vers le site distant.

```
localname @ localhost $ ssh -q -X -l loginname remotehost.domain
Password:
.....
```

- Lancez des applications X sur le site distant.

```
loginname @ remotehost $ gimp &
```

Cette méthode permet l'affichage du client X distant comme s'il était connecté par une socket UNIX locale.

9.4.9 L'émulateur de terminal X – `xterm`

Apprenez tout sur `xterm` à l'adresse `http://dickey.his.com/xterm/xterm.faq.html`.

9.4.10 Ressources X

Beaucoup d'anciens programmes X, comme `xterm`, utilisent la base de données de ressources X pour configurer leur apparence. Le fichier `~/.Xresources` est utilisé pour spécifier les ressources de l'utilisateur. Ce fichier est automatiquement ajouté aux ressources X par défaut au moment de la connexion. Les ressources X par défaut du système sont stockées dans `/etc/X11/Xresources/*` et les ressources par défaut des applications dans `/etc/X11/app-defaults/*`. Utilisez ces paramètres comme point de départ.

Voici quelques paramètres intéressants à ajouter à `~/.Xresources` :

```
! Règle la police à 9x15, plus lisible
XTerm*font: 9x15

! Affiche une barre de défilement
XTerm*scrollBar: true

! Règle la taille du tampon à 1000 lignes
XTerm*saveLines: 1000

! Large kterm screen
KTerm*VT100*fontList: -*-fixed-medium-r-normal--24-*,\
 -*-gothic-medium-r-normal--24-*,\
 -*-mincho-medium-r-normal--24-*
```

Pour que ces paramètres prennent effet immédiatement, ajoutez-les à la base de données avec la commande :

```
xrdb -merge ~/.Xresources
```

Voir `xrdb(1x)`.

9.4.11 Clavier et souris sous X

Le programme `xmodmap` est utilisé pour éditer et afficher la carte des modificateurs de clavier et la table des touches du clavier qui est utilisée par les applications clientes pour converter les codes touches des événements en symboles de touches sous X.

```
$ xmodmap -pm
 ... affiche la carte des modificateurs courante
$ xmodmap -pk | pager
 ... affiche la table clavier courante
$ xmodmap -e "pointer = 3 2 1" # configure la souris pour les gauchers
$ xmodmap ~/.xmodmaprc # configure le clavier comme dans ~/.xmodmaprc
```

Il est généralement exécuté depuis le script de démarrage de session d'un utilisateur, `~/.xsession`.

Pour obtenir le code touche, lancez `xev` sous X et appuyez sur des touches. Pour obtenir la signification d'un symbole de touche, regardez les définitions de MACRO dans `/usr/include/X11/keysymdef.h`. Tous les `#define` de ce fichier ont le préfixe `XK_` suivi du nom du symbole de touche.

Voir `xmodmap(1x)`.

9.4.12 Devenir root sous X

Si une application graphique a besoin d'être lancée avec les privilèges de root, utilisez les procédures suivantes pour le faire à partir du serveur X d'un utilisateur. N'essayez jamais de démarrer un serveur X depuis le compte root, pour vous garder d'éventuels risques de sécurité.

Démarrez le serveur X avec un utilisateur normal et ouvrez un `xterm`. Ensuite :

```
$ XAUTHORITY=$HOME/.Xauthority
$ export XAUTHORITY
$ su root
Password:*****
# printtool &
```

Lorsque vous utilisez cette astuce pour effectuer un `su` vers un utilisateur non root, faites attention à ce que le fichier `~/.Xauthority` soit lisible par le groupe de cet utilisateur.

Cette séquence de commandes peut être automatisée en créant un fichier `~/.xsession` sur le compte de l'utilisateur, avec les lignes suivantes :

```
# Cela permet que X marche quand je fais un su pour devenir root
if [ -z "$XAUTHORITY" ]; then
    XAUTHORITY=$HOME/.Xauthority
    export XAUTHORITY
fi
unset XSTARTUP
# Si vous désirez un gestionnaire de fenêtres/session particulier,
# décommentez les lignes suivantes et éditez-les
```

```
#XSTARTUP=/usr/bin/blackbox
# Démarre le programme de gestion de fenêtres/session
if [ -z "$XSTARTUP" ]; then
  if [ -x /usr/bin/x-session-manager ]; then
    XSTARTUP=x-session-manager
  elif [ -x /usr/bin/x-window-manager ]; then
    XSTARTUP=x-window-manager
  elif [ -x /usr/bin/x-terminal-emulator ]; then
    XSTARTUP=x-terminal-emulator
  fi
fi
# exécute le gestionnaire de fenêtres/session sélectionné
exec $XSTARTUP
```

Ensuite, exécutez `su` (pas `su -`) dans une fenêtre `xterm` de l'utilisateur. Maintenant vous pouvez exécuter des applications graphiques avec les privilèges de root sur l'écran X d'un utilisateur normal. Cette astuce fonctionne tant que le Xsession par défaut est utilisé. Si un utilisateur personnalise `~/.xinit` ou `~/.xsession`, la variable d'environnement `XAUTHORITY` mentionnée ci-dessus doit être positionnée de la même façon dans ces scripts.

Une autre méthode est d'utiliser le paquet `sudo` :

```
$ sudo xterm
... ou
$ sudo -H -s
```

Ici le fichier `/root/.bashrc` doit contenir :

```
if [ $SUDO_USER ]; then
    sudo -H -u $SUDO_USER xauth extract - $DISPLAY | xauth merge -
fi
```

Cela marche bien même si le répertoire home de l'utilisateur est sur une partition NFS, puisque root ne lit pas le fichier `.Xauthority`.

Il existe aussi des paquets spécialisés dans ce but : `kdesu`, `gksu`, `gksudo`, `gnome-sudo`, et `xsu`. Enfin, trois autres méthodes peuvent être utilisées pour atteindre des résultats similaires : créer un lien symbolique depuis `/root/.Xauthority` vers celui de l'utilisateur ; utiliser le script sux (`http://fgouget.free.fr/sux/sux-readme.shtml`); ou mettre `xauth merge ~`*`USER_RUNNING_X`*`/.Xauthority` dans le script d'initialisation de root.

Plus d'informations sur la liste de diffusion debian-devel (`http://lists.debian.org/debian-devel/2002/debian-devel-200207/msg00259.html`).

9.4.13 Polices TrueType dans X

Le `xfs` standard de XFree86-4 fonctionne bien avec les polices TrueType. Il faut installer un autre serveur de polices comme `xfs-xtt` si vous utilisez XFree86-3.

Vous avez simplement besoin de vous assurer que les applications que vous voulez utiliser avec les polices TrueType sont liées avec les bibliothèques libXft ou libfreetype (vous n'avez probablement pas à vous en soucier si vous utilisez des .deb précompilés).

D'abord, configurez l'infrastructure de support des polices.

- Installez les paquets `x-ttcidfont-conf` et `defoma`. Cela automatise la génération des fichiers `fonts.scale` et `fonts.dir`.

  ```
  # apt-get install x-ttcidfont-conf
  ```

- Éditez `/etc/X11/XF86Config-4` dans la section `Section "Files"` comme suit :

  ```
  Section "Files"
      FontPath  "/var/lib/defoma/x-ttcidfont-conf.d/dirs/TrueType"
      FontPath  "/usr/share/fonts/truetype"
      FontPath  "/usr/lib/X11/fonts/CID"
      FontPath  "/usr/lib/X11/fonts/Speedo"
      FontPath  "/usr/lib/X11/fonts/misc"
      FontPath  "/usr/lib/X11/fonts/cyrillic"
      FontPath  "/usr/lib/X11/fonts/100dpi:unscaled"
      FontPath  "/usr/lib/X11/fonts/75dpi:unscaled"
      FontPath  "/usr/lib/X11/fonts/Type1"
  EndSection
  ```

  La première ligne configure XFree86 pour utiliser toutes les polices TrueType que vous installez à partir de paquets Debian. L'entrée pour les polices Type1 est déplacée en bas car XFree86 ne rend pas bien les polices Type1. Les options `:unscaled` pour les polices bitmap ne sont en principe pas nécessaires, mais je les inclus ici au cas où. Afin de préserver les changements manuels de `/etc/X11/XF86Config-4`, suivez les instructions de 'Configurer manuellement le serveur X' page 134.

Ensuite, installez des paquets de polices suivant les DFSG :

- Polices TrueType occidentales :
  - `ttf-bitstream-vera` : ensemble de polices TrueType de grande qualité créées par Bitstream, Inc. [2]
  - `ttf-freefont` : ensemble de polices TrueType gratuites de grande qualité pour le jeu de caractères UCS.
  - `ttf-thryomanes` : police Unicode TrueType pour les jeux de caractères latin, grec, cyrillique et IPA.
- Polices asiatiques :
  - `tfm-arphic-bsmi00lp` : police TrueType et métriques de police TeX chinoise Arphic "AR PL Mingti2L Big5"
  - `tfm-arphic-bkai00mp` : police TrueType et métriques de police TeX chinoise Arphic "AR PL KaitiM Big5"

---

[2] Bien que cela ne soit pas disponible dans Woody, vous pouvez les installer depuis Sarge.

- `tfm-arphic-gbsn00lp` : police TrueType et métriques de police TeX chinoise Arphic “AR PL SungtiL GB”
- `tfm-arphic-gkai00mp` : police TrueType et métriques de police TeX chinoise Arphic “AR PL KaitiM GB”
- `ttf-baekmuk` : série de polices TrueType coréennes Baekmuk
- `hbf-jfs56` : police bitmap chinoise Jianti Fangsong 56x56 (GB2312) pour CJK
- `hbf-cns40-b5` : police bitmap chinoise Fanti Song 40x40 (Big5) pour CJK
- `hbf-kanji48` : police bitmap japonaise Kanji 48x48 (JIS X-0208)

Puisque les polices **libres** sont parfois limitées, l’installation ou le partage de certaines polices TrueType commerciales est une option pour les utilisateurs Debian. Afin de rendre ce processus facile pour l’utilisateur, il existe certains paquets :

- `ttf-commercial`
- `msttcorefonts (>1.1.0)` [3]

Vous aurez ainsi une très bonne sélection de polices TrueType au prix de la contamination de votre système **libre** par des polices non-libres.

Tous ces paquets de polices dans Debian devraient marcher sans effort et être disponibles pour tous les programmes X qui utilisent le système de polices standard. Cela inclut par exemple Xterm, Emacs et la plupart des autres applications non-KDE et non-GNOME.

Maintenant, lancez `xfontsel` et sélectionnez une police TrueType dans le menu « fndry ». Vous devriez voir un grand nombre d’entrées non grisées dans le menu « fmly ».

Pour KDE2.2 et GNOME1.4 (avec libgdkxft0, qui est un hack pour que GTK1.2 fasse de l’anticrénelage sur les polices), vous devez configurer Xft1 aussi. Xft1 est obsolète, et n’est utilisé pratiquement que par GNOME1.4 et KDE2.2. Éditez le fichier `/etc/X11/XftConfig` et ajoutez une ligne

```
dir "/var/lib/defoma/x-ttcidfont-conf.d/dirs/TrueType"
```

avant les autres lignes `dir`. [4]

Pour GNOME2 et KDE3 (après la sortie de Sarge), vous devez configurer `fontconfig` que Xft2 utilise pour trouver les polices. [5] Vous ne devriez pas à avoir installer autre chose pour cela, puisque tous les paquets utilisant `fontconfig` auront une dépendance sur lui.

D’abord, regardez le fichier `/etc/fonts/fonts.conf`. Il devrait y avoir une ligne comme celle ci-dessous. Sinon, éditez `/etc/fonts/local.conf` et ajoutez :

```
<dir>/var/lib/defoma/x-ttcidfont-conf.d/dirs/TrueType</dir>
```

---

[3] Le paquet de Woody ne marche plus depuis 8/2002 à cause d’un changement sur le site web de Microsoft. Utilisez la version de Sarge à la place, même dans Woody.

[4] Je n’ai plus Xft1 sur ma machine, donc je ne suis pas sûr s’il faut redémarrer X ou pas avant que les changements prennent effet. Il me semble que `xftcache` met à jour le cache Xft1, mais j’aimerais que quelqu’un me le confirme.

[5] `fontconfig` n’existe pas dans Woody.

juste après la ligne <fontconfig>

`fontconfig` devrait prendre cela en compte immédiatement et `fc-list` devrait lister les nouvelles polices. Une autre possibilité intéressante de `fontconfig` est que vous pouvez mettredes polices dans `~/.fonts/` et tous les programmes utilisant `fontconfig` y auront accès immédiatement.

Si vous installez manuellement un nouvel ensemble de polices TrueType dans X sans un paquet Debian, lancez

```
# xset fp rehash
```

pour que XFree86 les prenne en compte.

9.4.14 Navigateurs Web pour X

Il existe quelques paquets de navigateurs web disponibles avec la sortie de Woody :

- `mozilla` Le navigateur Mozilla (nouveau)
- `galeon` Navigateur basé sur Mozilla avec une interface Gnome (nouveau)
- `konqueror` Navigateur KDE
- `dillo` Navigateur GTK
- `amaya-gtk` Navigateur de référence du W3C
- `amaya-lesstif` Navigateur de référence du W3C
- `netscape-...` (plusieurs, vieux)
- `communicator-...` (plusieurs, vieux)
- ...

La version de `mozilla` doit correspondre avec la version que `galeon` requiert. A part l'interface graphique, ces deux logiciels partagent le moteur de rendu HTML Gecko.

Des ajouts pour les navigateurs comme `mozilla` et `galeon` peuvent être activés en installant des fichiers `*.so` directement dans le répertoire `plugin` et en relançant les navigateurs.

Ressources :

- Java : installer le binaire « J2SE » depuis `http://java.sun.com`.
- Flash : installer le binaire « Macromedia Flash Player 5 » depuis `http://www.macromedia.com/software/flashplayer/`.
- `freewrl` : navigateur VRML et plugin Netscape
- ...

9.5 SSH

SSH (Secure SHell) est le moyen sécurisé de se connecter à d'autres machines au travers d'Internet. Une version libre de SSH appelée OpenSSH est disponible dans le paquet `ssh` de Debian.

9.5.1 Bases de SSH

Installez d'abord le serveur et le client OpenSSH.

```
# apt-get update && apt-get install ssh
```

Le fichier `/etc/ssh/sshd_not_to_be_run` ne doit pas être présent si l'on veut exécuter le serveur OpenSSH.

SSh possède deux protocoles d'identification :

- Protocole SSH version 1 :
  - la version de Potato supporte seulement ce protocole.
  - méthodes d'identification disponibles :
    - RSAAuthentication : identification basée sur une clé RSA
    - RhostsAuthentication : identification basée sur .rhosts (non sûr, désactivé)
    - RhostsRSAAuthentication : identification .rhosts combinée avec clé RSA (désactivé)
    - ChallengeResponseAuthentication : identification par challenge-response RSA
    - PasswordAuthentication : identification basée sur un mot de passe
- Protocole SSH version 2 :
  - les versions d'après Woody utilisent ce protocole par défaut.
  - méthodes d'identification disponibles :
    - PubkeyAuthentication : identification de l'utilisateur basée sur une clé publique
    - HostbasedAuthentication : identification par `.rhosts` ou `/etc/hosts.equiv` combinée avec identification de l'hôte par clé publique (désactivé)
    - ChallengeResponseAuthentication : identification par challenge-response RSA
    - PasswordAuthentication : identification basée sur un mot de passe

Faites attention avec ces différences si vous migrez vers Woody ou si vous utilisez un système non Debian.

Voir `/usr/share/doc/ssh/README.Debian.gz`, `ssh(1)`, `sshd(8)`, `ssh-agent(1)`, et `ssh-keygen(1)` pour plus de détails.

Les fichiers suivants sont les fichiers de configuration importants.

- `/etc/ssh/ssh_config` : valeurs par défaut pour le client SSH. Voir `ssh(1)`. Entrées importantes :
  - `Host` : Restreint les entrées suivantes (jusqu'au prochain Host) aux hôtes qui correspondent au motif donné après le mot clé.
  - `Protocol` : Spécifie la version du protocole SSH. Par défaut, « 2,1 ».
  - `PreferredAuthentications` : Spécifie la méthode d'identification pour le client SSH2. Par défaut, « hostbased,publickey,keyboard-interactive,password ».
  - `PasswordAuthentication` : Si vous voulez vous connecter avec un mot de passe, vous devez vous assurer que cela n'est pas mis à `no`.
  - `ForwardX11` : Désactivé par défaut. Peut être outrepassé par l'option `-X` en ligne de commande.

- `/etc/ssh/sshd_config` : valeurs par défaut pour le serveur SSH. Voir `sshd(8)`. Entrées à noter :
  - `ListenAddress` : Spécifie les adresses locales que `sshd` doit écouter. Plusieurs options sont permises.
  - `AllowTcpForwarding` : Désactivé par défaut.
  - `X11Forwarding` : Désactivé par défaut.
- `$HOME/.ssh/authorized_keys` : liste de clés publiques par défaut utilisées pour se connecter à ce compte sur cette machine. Voir `ssh-keygen(1)`.
- `$HOME/.ssh/identity` : Voir `ssh-add(1)` et `ssh-agent(1)`.

Les commandes suivantes permettent de démarrer une connexion `ssh`.

```
$ ssh username@hostname.domain.ext
$ ssh -1 username@hostname.domain.ext # Force SSH version 1
$ ssh -1 -o RSAAuthentication=no -l username foo.host
            # force un mot de passe avec SSH1
$ ssh -o PreferredAuthentications=password -l username foo.host
            # force un mot de passe avec SSH2
```

Pour un utilisateur, `ssh` est plus intelligent que `telnet` (il ne crashe pas avec ^]).

9.5.2 Redirection de port pour un tunnel SMTP/POP3

Pour établir un tunnel pour se connecter au port 25 de *remote-server* depuis le port 4025 de la machine locale, et au port 110 de *remote-server* depuis le port 4110 de la machine locale par `ssh`, exécutez les commandes suivantes sur la machine locale :

```
# ssh -q -L 4025:remote-server:25 4110:remote-server:110 \
           username@remote-server
```

C'est un moyen sécurisé d'effectuer une connexion vers un serveur SMTP/POP3 au travers d'Internet. Positionnez l'entrée `AllowTcpForwarding` à `yes` dans `/etc/ssh/sshd_config` sur la machine distante.

9.5.3 Se connecter avec moins de mots de passe – RSA

On peut éviter de se rappeler le mot de passe pour chaque système distant en utilisant RSAAuthentication (protocole SSH1) ou PubkeyAuthentication (protocole SSH2).

Sur le système distant, le contenu de `/etc/ssh/sshd_config` doit contenir « RSAAuthentication yes » ou « PubkeyAuthentication yes ».

Générez ensuite les clés d'identification localement et installez la clé publique sur le système distant :

```
$ ssh-keygen          # RSAAuthentication : clé RSA1 pour SSH1
$ cat .ssh/identity.pub | ssh user1@remote \
        "cat - >>.ssh/authorized_keys"
...
$ ssh-keygen -t rsa   # PubkeyAuthentication : clé RSA pour SSH2
$ cat .ssh/id_rsa.pub | ssh user1@remote \
        "cat - >>.ssh/authorized_keys"
...
$ ssh-keygen -t dsa   # PubkeyAuthentication : clé DSA pour SSH2
$ cat .ssh/id_dsa.pub | ssh user1@remote \
        "cat - >>.ssh/authorized_keys"
```

On peut changer la phrase de passe par la suite avec `ssh-keygen -p`. Assurez-vous de vérifier la configuration en testant la connexion. En cas de problème, utilisez `ssh -v`.

Vous pouvez ajouter des options dans les entrées de `authorized_keys` pour limiter les hôtes et exécuter des commandes spécifiques. Consultez `sshd(8)` pour plus de détails.

Notez que SSH2 fait une identification de type `HostbasedAuthentication`. Pour que cela fonctionne, il faut régler la configuration de `HostbasedAuthentication` à `yes` dans `/etc/ssh/sshd_config` sur le serveur et `/etc/ssh/ssh_config` ou `$HOME/.ssh/config` sur le client.

9.5.4 Clients SSH étrangers

Il existe quelques clients SSH gratuits disponibles pour des systèmes non Unix.

Windows puTTY (`http://www.chiark.greenend.org.uk/~sgtatham/putty/`) (GPL)

Windows (cygwin) SSH in cygwin (`http://www.cygwin.com/`) (GPL)

Macintosh Classic macSSH (`http://www.macssh.com/`) (GPL) [Notez que Mac OS X inclut OpenSSH ; utilisez `ssh` depuis l'application Terminal]

Voir SourceForge.net, documentation du site (`http://www.sourceforge.net/docman/?group_id=1`), « 6. CVS Instructions ».

9.5.5 Configurer `ssh-agent`

Il est plus sûr de protéger votre clé d'identification SSH avec une phrase de passe. Si elle n'a pas été configurée, exécutez `ssh-keygen -p`.

Mettez votre clé publique (par exemple `~/.ssh/id_rsa.pub`) dans `~/.ssh/authorized_keys` sur un hôte distant comme décrit dans 'Se connecter avec moins de mots de passe – RSA' page précédente.

```
$ ssh-agent bash # ou lancez zsh/tcsh/pdksh à la place
$ ssh-add ~/.ssh/id_rsa
Enter passphrase for /home/osamu/.ssh/id_rsa:
Identity added: /home/osamu/.ssh/id_rsa (/home/osamu/.ssh/id_rsa)
$ scp foo user@remote.host:foo
 ... pas besoin de phrase de passe à partir de maintenant :-)
$^D
 ... fin de la session ssh-agent
```

Pour le serveur X, les scripts de démarrage de Debian exécutent `ssh-agent` comme processus parent. Donc, vous n'avez pas besoin d'exécuter `ssh-add` qu'une seule fois.

Pour plus d'information, consultez `ssh-agent(1)` et `ssh-add(1)`.

9.5.6 Résoudre les problèmes avec SSH

Si vous rencontrez des problèmes, vérifiez les permissions du fichier de configuration et lancez `ssh` avec l'option `-v`.

Utilisez `-P` si vous êtes root et avez des problèmes avec un pare-feu ; cela évite d'utiliser des ports serveur 1–1023.

Si les connexions `ssh` vers un site distant arrêtent subitement de fonctionner, cela peut être suite à des bidouilles de l'administrateur, sûrement un changement de `host_key` pendant une maintenance du système. Après s'être assuré que c'est bien le cas et que personne n'essaie de se faire passer pour la machine distante par une bidouille, on peut se reconnecter en enlevant l'entrée `host_key` de `~/.ssh/known_hosts` sur la machine locale.

9.6 Logiciels de courrier électronique

La configuration du courrier électronique se divise en trois parties :

- Agent de transfert de courriel (Mail Transfer Agent, MTA) : `exim4`, `exim`, `postfix`, `sendmail`, `qmail`, `ssmtp`, `nullmailer`, ...
- Utilitaires : `procmail`, `fetchmail`, `mailx`, `crm114`, ...
- MUA : `mutt`, `emacs+gnus`, ...

9.6.1 Agent de transport de courrier électronique

Pour un MTA (*Mail Transport Agent*) complet, utilisez `exim` dans Woody et `exim4` dans Sarge. [6] Références :

- paquets `exim-doc` et `exim-doc-html` pour `exim`
- paquets `exim4-doc-info` et `exim4-doc-html` pour `exim4`

---

[6] Les sections suivantes utilisent `exim` dans les exemples. pour Sarge, remplacez par `exim4`.

- `http://www.exim.org/`

La seule alternative raisonnable est `postfix` si vous vous souciez de la sécurité. `sendmail` et `qmail` sont disponibles dans Debian mais ne sont pas recommandés.

Si vous n'avez pas besoin de la capacité de relai d'un MTA comme dans le cas d'un système satellite comme un portable, vous pouvez utiliser l'un de ces paquets légers :
- `ssmtp` : a besoin d'une connexion SMTP et est capable de gérer des alias, ou
- `nullmailer` : gère une file d'attente mais n'est pas capable de gérer des alias.

A ce jour, je trouve qu'`exim` est le plus approprié même pour ma station de travail personnelle, qui est un portable.

Vous pouvez avoir besoin de supprimer `exim` pour installer l'un de ces paquets entrant en conflit avec :

```
# dpkg -P --force-depends exim
# apt-get install nullmailer         # ou ssmtp
```

Smarthost

Si vous utilisez `exim4` ou `exim` sur une machine qui est connectée par un service grand public, assurez-vous d'envoyer le courriel sortant à travers le serveur offert par votre FAI ou un autre. [7] Voici quelques bonnes raisons pour le faire :
- pour assurer les ré-essais SMTP puisque le serveur de votre FAI a une connexion de meilleure qualité ;
- pour éviter d'envoyer le courriel directement d'une **adresse IP dynamique** qui risque d'être bloquée par les listes anti-spam ;
- pour sauver de la bande passante localement lors d'envois de courriels à plusieurs destinataires.

Les seules exceptions éventuelles sont :
- pallier à une défaillance du serveur de votre FAI ;
- une expérience à but éducatif ;
- votre machine est un serveur professionnel.

Configuration de base d'Exim

Pour configurer `exim4` ou `exim` en tant que MTA, configurez ce qui suit :

```
/etc/exim.conf          "eximconfig" pour créer et éditer (exim)
/etc/exim4/*            "dpkg-reconfigure exim4-config" pour créer et éditer (ex
/etc/inetd.conf         mettez smtp en commentaire pour exécuter exim en
                        tant que daemon
/etc/email-addresses    ajoutez les listes d'adresses sources spoofées
```

---

[7] Vous devez suivre cette règle pour toute machine en accès RTC, DSL, câble ou LAN par un routeur haut débit. Même si votre machine a une adresse IP fixe, c'est toujours une bonne idée de suivre cette règle. La plupart des stations de travail et serveurs à la maison tombent dans cette catégorie.

Vérifiez les filtres avec `exim4` ou `exim` avec les options `-brw, -bf, -bF, -bV, ...` etc.

Configurer Exim pour récupérer les adresses non existantes

Dans `/etc/exim/exim.conf` (Woody ou plus), dans la partie DIRECTORS, à la fin (après localuser : director) ajoutez une directive "catch-all" qui correspond à toutes les adresses que la directive précédente ne pouvait résoudre (par Miquel van Smoorenburg) :

```
catchall:
  driver = smartuser
    new_address = webmaster@mydomain.com
```

Si vous souhaitez une recette plus détaillée pour chaque domaine virtuel, ajoutez ce qui suit à la fin de `exim.conf` (par moi, pas bien testé) :

```
*@yourdomain.com ${lookup{$1}lsearch*{/etc/email-addresses} \
        {$value}fail} T
```

Et ajoutez une entrée « * » dans `/etc/email-addresses`.

Configurer Exim pour réécrire les adresses sélectionnées sur les courriels sortants

La réécriture des adresses pour le courrier sortant afin d'avoir des en-têtes « From : » corrects peut être réalisée avec `exim` avec la configuration suivante près de la fin de `/etc/exim/exim.conf` :

```
*@host1.something.dyndns.org \
  "${if eq {${lookup{$1}lsearch{/etc/passwd}{1}{0}}} {1}  \
   {$0}{$1@something.dyndns.org}}"  frFs
```

Cela réécrit toutes les adresses de type `*@host1.something.dyndns.org`.

1. Cela cherche dans le fichier `/etc/password` pour voir si la partie locale ($1) est un utilisateur local ou pas.
2. Si c'est un utilisateur local, cela réécrit l'adresse telle qu'elle était en premier lieu ($0).
3. Si ce n'est pas un utilisateur local, cela réécrit le nom de domaine.

Configurer Exim pour l'authentification SMTP

Certains services SMTP comme yahoo.com demandent une authentification SMTP. Configurez `/etc/exim/exim.conf` comme suit :

```
remote_smtp:
  driver = smtp
  authenticate_hosts = smtp.mail.yahoo.com
...

smarthost:
  driver = domainlist
  transport = remote_smtp
  route_list = "* smtp.mail.yahoo.com bydns_a"
...

plain:
  driver = plaintext
  public_name = PLAIN
  client_send = "^cmatheson3^this_is_my_password"
```

N'oubliez pas les double guillemets sur la dernière ligne.

9.6.2 Récupérer le courrier électronique – Fetchmail

`fetchmail` est exécuté en mode démon et récupère le courrier électronique du compte POP3 du FAI sur le système de mail local. Configurez :

```
/etc/init.d/fetchmail   voir ci-dessous pour le script
/etc/rc?.d/???fetchmail exécutez update-rc.d fetchmail defaults 30
/etc/fetchmailrc        fichier de configuration (chown 600)
```

Les informations pour démarrer `fetchmail` en mode démon depuis un script `init.d` dans Potato ne sont pas claires (Woody corrige cela). Voir les exemples `/etc/init.d/fetchmail` et `/etc/fetchmailrc` dans exemples de scripts (`http://www.debian.org/doc/manuals/debian-reference/examples/`).

Si vos en-têtes de courriers électroniques sont contaminées par des ^M dûs au logiciel de courrier électronique de votre FAI, ajoutez « stripcr » aux options dans `$HOME/.fetchmailrc` :

```
options fetchall no keep stripcr
```

9.6.3 Traiter le courrier électronique – Procmail

`procmail` est un filtre pour le courrier électronique délivré. Il faut créer un `$HOME/.procmailrc` pour chaque compte qui l'utilise. Exemple : _procmailrc (`http://www.debian.org/doc/manuals/debian-reference/examples/`).

9.6.4 Filtrer le spam avec `crm114`

Le paquet `crm114` fournit le script `/usr/share/crm114/mailfilter.crm` écrit en CRM114. Ce script fournit un filtre anti-spam très efficace qui peut être entraîné en lui fournissant du spam et des mails valides.

CRM114 est un petit langage conçu pour écrire des filtres ; c'est un espèce de grep avec des super pouvoirs. Voir `crm(1)`.

9.6.5 Lire le courrier électronique – Mutt

Utilisez `mutt` comme agent pour utilisateur de courrier électronique (Mail User Agent, MUA) associé à `vim`. Personnalisez-le dans `~/.muttrc` :

```
# utiliser le mode visuel et "gq" pour reformater les citations
set editor="vim -c 'set tw=72 et ft=mail'"
#
# sélection des en-têtes prise dans le manuel (« Sven's Draconian header
# weeding »)
#
ignore *
unignore from: date subject to cc
unignore user-agent x-mailer
hdr_order from subject to cc date user-agent x-mailer
auto_view application/msword
....
```

Ajoutez ce qui suit dans `/etc/mailcap` ou `$HOME/.mailcap` pour afficher les courriers électroniques en HTML et les attachements MS Word dans le message :

```
text/html; lynx -force_html %s; needsterminal;
application/msword; /usr/bin/antiword '%s'; copiousoutput;
description="Microsoft Word Text"; nametemplate=%s.doc
```

9.7 Localisation (l10n)

Debian est internationalisé et supporte un nombre toujours plus élevé de langues et de conventions locales. La sous-section suivante liste quelques formes de la diversité que Debian supporte actuellement, et la sous-section suivante parle de la **localisation**, processus de personnalisation de votre environnment de travail pour permettre l'entrée et l'affichage de votre langue et conventions pour les dates, les formats numériques et monétaires, et d'autres aspects d'un système qui diffèrent selon votre région.

9.7.1 Bases de la localisation

Il y a plusieurs aspects à la personnalisation de la localisation et du support de la langue nationale.

Localiser le clavier

Debian est distribuée avec le support pour près de deux douzaines de claviers. Dans Woody, reconfigurez le clavier avec :

- `dpkg-reconfigure --priority=low console-data # console`
- `dpkg-reconfigure --priority=low xserver-xfree86 # XF4`
- `dpkg-reconfigure --priority=low xserver-common-v3 # XF3`

Localiser les fichiers de données

La grande majorité des paquets de logiciels Debian supporte l'entrée de caractères non US-ASCII avec la variable d'environnement LC_TYPE offerte par la technologie des **locales** de la glibc.

- Propres en 8 bits : presque tous les programmes
- Autres langues latines (par exemple ISO-8859-1 ou ISO-8859-2) : majorité des programmes
- Langues multi-octets comme le Japonais ou le Coréen : beaucoup de nouvelles applications

Localiser l'affichage

X peut afficher n'importe quel codage y compris UTF-8 et supporte toutes les fontes. La liste inclut non seulement les polices 8 bits mais aussi des polices 16 bits comme pour le Chinois, le Japonais et le Coréen. L'entrée de caractères multi-octets est supportée par le mécanisme 'Méthodes d'entrée alternatives pour X' page 162. Voir 'Exemple de système X bilingue' page 159 et 'Support UTF-8 pour l'émulateur de terminal X' page 163.

L'affichage du codage japonais EUC est aussi possible sur la console graphique (S)VGA avec le paquet `kon2`. Il existe aussi un nouvel affichage japonais `jfbterm` qui utilise la console FB. Dans ces environnements de console, la méthode d'entrée du Japonais doit être fournie par l'application. Utilisez le paquet `egg` pour Emacs et utilisez le paquet `jvim` pour Vim.

L'installation de polices non Unicode dans X aide à afficher des documents ayant n'importe quel codage. Aussi ne vous inquiétez pas trop à propos du codage des polices.

Localiser les messages et la documentation

Des traductions existent pour beaucoup de messages texte et documents qui sont affichés par le système Debian, comme les messages d'erreur, l'affichage standard des programmes, les menus et les pages de manuel. Actuellement, le support pour les langues Allemand, Espagnol, Finlandais, Français, Hongrois, Italien, Japonais, Koréen et Polonais dans les pages de manuel est fourni par les paquets `manpages-`*`LANG`* (où *LANG* est le code ISO à deux lettres du pays. Utilisez `apt-cache search manpages-|less` pour avoir une liste des pages de manuel disponibles).

Pour accéder à une page de manuel NLS, l'utilisateur doit configurer la variable du shell LC_MESSAGES de la façon appropriée. Par exemple, dans le cas des pages de manuel en Italien, LC_MESSAGES doit être positionnée à `it`. Le programme `man` cherchera alors les pages de manuel en Italien sous `/usr/share/man/it/`.

9.7.2 Locales

Debian supporte la technologie **locale**. C'est un mécanisme qui permet aux programmes de fournir un affichage et des fonctionnalités adaptés aux conventions locales comme le jeu de caractères, le format de la date et de l'heure, le symbole de la monnaie, et ainsi de suite. Il utilise des variables d'environnement pour déterminer le comportement approprié. Par exemple, en supposant que vous avez les locales Anglais américain et Allemand installées sur votre système, les messages d'erreur de beaucoup de programmes peuvent être multilingues :

```
$ LANG="en_US" cat foo
cat: foo: No such file or directory
$ LANG="de_DE" cat foo
cat: foo: Datei oder Verzeichnis nicht gefunden
```

La glibc offre le support pour cette fonctionnalité dans la bibliothèque. Voir `locale(7)`.

9.7.3 Introduction aux locales

La description complète d'une locale comprend trois parties : `xx_YY.ZZZZ`.
- **`xx`** : code de la langue ISO 639 (minuscule)
- **`YY`** : code du pays ISO 3166 (majuscule)
- **`ZZZZ`** : table du code, c-à-d table des caractères ou identificateur de codage.

Pour les codes de langue et de pays, voir l'information de `info gettext`.

Veuillez noter que la partie table du code peut être normalisée de façon interne pour obtenir une compatibilité inter-plateformes en enlevant tous les – et en convertissant tous les caractères en minuscules. Voici les tables de code typiques :
- **UTF-8** : Unicode pour toutes les régions, principalement en 1-3 octets (nouveau standard de fait)

- **ISO-8859-1** : Europe de l'ouest (ancien standard de fait)
- **ISO-8859-2** : Europe de l'est (Bosniaque, Croate, Tchèque, Hongrois, Polonais, Roumain, Serbe, Slovaque, Slovène)
- **ISO-8859-3** : Malte
- **ISO-8859-5** : Macédonien, Serbe
- **ISO-8859-6** : Arabe
- **ISO-8859-7** : Grec
- **ISO-8859-8** : Hébreux
- **ISO-8859-9** : Turc
- **ISO-8859-11** : Thai (=TIS-620)
- **ISO-8859-13** : Léton, Lithuanien, Maori
- **ISO-8859-14** : Gallois
- **ISO-8859-15** : Europe de l'ouest avec l'euro
- **KOI8-R** : Russe
- **KOI8-U** : Ukrainien
- **CP1250** : Tchèque, Hongrois, Polonais (sur MS Windows à l'origine)
- **CP1251** : bulgare, Biélorusse (sur MS Windows à l'origine)
- **eucJP** : Japonais façon Unix (=ujis)
- **eucKR** : Coréen façon Unix
- **GB2312** : Chinois Simplifié façon Unix (=GB, =eucCN) pour zh_CN
- **Big5** : Chinois Traditionel pour zh_TW
- **sjis** : Japonais façon Microsoft (Shift-JIS)

Et voici la signification du jargon des systèmes de codage :

- **ASCII** : 7 bits (0-0x7f)
- **ISO-8859- ?** : 8 bits (0-0xff)
- **ISO-10646-1** : Universal Character Set (UCS) (31 bits, 0-0x7fffffff)
- **UCS-2** : 16 premiers bits de UCS comme 2 octets simples (Unicode : 0-0xffff)
- **UCS-4** : UCS comme 4 octets simples (UCS : 0-0x7fffffff)
- **UTF-8** : UCS codé sur 1-6 octets (généralement 3 octets)
- **ISO-2022** : 7 bits (0-0xff) avec la séquence d'échappement ; c'est le codage japonais le plus populaire pour le courrier électronique
- **EUC** : 8 bits + 16 bits (0-0xff), façon Unix
- **Shift-JIS** : 8 bits + 16 bits (0-0xff), façon Microsoft

ISO-8859- ?, EUC, ISO-10646-1, UCS-2, UCS-4 et UTF-8 partagent le même code qu'ASCII pour les caractères de 7 bits. EUC ou Shift-JIS utilisent des caractères de bit haut (0x80-0xff) pour indiquer qu'une partie du codage est sur 16 bits. UTF-8 utilise aussi des caractères de bit haut (0x80-0xff) pour indiquer les séquences de caractères sur plus de 7 bits. C'est le système de codage le plus intelligent pour gérer les caractères non-ASCII.

Veuillez noter la différence d'ordre des octets des implémentations d'Unicode :

- **Standard UCS-2, UCS-4** : big endian
- **Microsoft UCS-2, UCS-4** : little endian pour ix86 (dépend de la machine)

Voir 'Convertir un fichier texte avec `recode`' page 105 pour convertir entre les jeux de caractères. Pour plus d'information, consultez Introduction to i18n (`http://www.debian.org/doc/manuals/intro-i18n/`) (en Anglais).

9.7.4 Activer le support des locales

Debian n'est **pas** livrée avec toutes les locales disponibles précompilées. Consultez le fichier `/usr/lib/locale` pour voir quelles locales (à part le défaut, « C ») sont compilées pour votre système. Si celle dont vous avez besoin n'est pas présente, vous avez deux possibilités :

- Editer `/etc/locale.gen` pour ajouter la locale souhaitée, et lancer `locale-gen` en tant que root pour la compiler. Voir `locale-gen(8)` et les pages de manuel listées dans la section « Voir aussi ».
- Lancer `dpkg-reconfigure locales` pour reconfigurer le paquet `locales`. Ou s'il n'est pas déjà installé, l'installation de `locales` invoquera l'interface debconf pour choisir les locales désirées et les compiler.

9.7.5 Activer une locale particulière

Les variables d'environnement suivantes sont évaluées dans cet ordre pour fournir les valeurs de locale aux programmes :

1 LANGUAGE : cette variable d'environnement consiste en une liste de locales séparées par deux points verticaux, dans l'ordre de priorité. Utilisé seulement si la locale POSIX est positionnée à une valeur autre que "C" [dans Woody ; la version de Potato a toujours priorité sur la locale POSIX]. (GNU extension)

2 LC_ALL : Si cette variable est non nulle, sa valeur est utilisée pour toutes les catégories de locales. (POSIX.1) Habituellement "" (null).

3 LC_* : Si cette variable est non nulle, sa valeur est utilisée pour la catégorie correspondante. (POSIX.1) Habituellement "C".

  Les variables LC_* sont :
  - LC_CTYPE : Classification des caractères et conversion de casse.
  - LC_COLLATE : Ordre de collation.
  - LC_NUMERIC : Formats des nombres non monétaires.
  - LC_MONETARY : Formats monétaires.
  - LC_MESSAGES : Formats des messages d'information et de diagnostic et des réponses intéractives.
  - LC_PAPER : Taille du papier.
  - LC_NAME : Formats des noms.
  - LC_ADDRESS : Formats des adresses et information sur les lieux.
  - LC_TELEPHONE : Formats des numéros de téléphone.
  - LC_MEASUREMENT : Unités de mesure (Metric ou Other).
  - LC_IDENTIFICATION : Donnée générale sur l'information de la locale.

4 LANG : Si cette variable est non nulle, et que LC_ALL n'est pas définie, sa valeur est utilisée pour toutes les catégories de locales LC_* sans valeur définie. (POSIX.1) Habituellement, "C".

Notez que certaines applications (par exemple Netscape 4) ignorent les valeurs LC_*.

Le programme `locale` peut afficher la locale active et les locales disponibles; voir `locale(1)`. (NOTE : `locale -a` liste toutes les locales du système; cela ne signifie pas qu'elles sont toutes compilées ! Voir 'Activer le support des locales' page ci-contre.)

9.7.6 Format de date local ISO 8601

Le support de locale pour le standard international de la date `yyyy-mm-dd` (format de date ISO 8601) est fournit par la locale appelée `en_DK`, « Anglais au Danemark » qui est une espèce de blague :-) Cela semble ne marcher que sur la console pour `ls`.

9.7.7 Exemple pour les États-Unis (ISO-8859-1)

Ajoutez les lignes suivantes dans `~/.bash_profile` :

```
LC_CTYPE=en_US.ISO-8859-1
export LC_CTYPE
```

9.7.8 Exemple pour la France avec le caractère Euro (ISO-8859-15)

Ajoutez les lignes suivantes dans `~/.bash_profile` :

```
LANG=fr_FR@euro
export LANG
LC_CTYPE=fr_FR@euro
export LC_CTYPE
```

Configurez le clavier français « AZERTY » comme décrit dans la section 'Localiser le clavier' page 155 et ajoutez les pages de manuel en Français en installant `manpages-fr`. La touche Alt-Droite s'appelle Alt-Gr en Europe. En appuyant dessus en même temps que d'autres touches on obtient de nombreux caractères accentués et spéciaux. Par exemple, Alt-Gr+E donne le signe Euro.

La plupart des langues d'Europe de l'Ouest peuvent être configurées de la même façon.

Voir Debian Euro HOWTO (`http://www.debian.org/doc/manuals/debian-euro-support/`) pour ajouter le support de la nouvelle monnaie Euro et Utiliser et configurer Debian pour le Français (`http://www.debian.org/doc/manuals/fr/debian-fr-howto/`) pour plus de détails en Français.

9.7.9 Exemple de système X bilingue

Configurons un système X bilingue qui supporte simultanément le Japonais, l'Anglais, l'Allemand et le Français avec les codages EUC, UTF-8 et ISO-8859-1 dans différentes consoles.

Je vais vous montrer une personnalisation en utilisant le système de menu Debian. Pour les détails du système de menu Debian consultez `/usr/share/doc/menu/html/index.html`. Je crée aussi un raccourci pour le navigateur web `mozilla` dans l'exemple. [8]

- ajoutez le support de la locale pour le Japonais ja_JP.eucJP et les autres locales requises avec la méthode décrite à la section 'Localisation (l10n)' page 154 (pour toutes).
- installez un système de conversion et un dictionnaire Kana vers Kanji (pour le Japonais) :
  - `canna` – serveur local (licence gratuite), ou
  - `freewnn-jserver` – serveur extensible par le réseau (domaine public)
- installez le système d'entrée pour le Japonais :
  - `kinput2-canna` – pour X, ou
  - `kinput2-canna-wnn` – pour X, et
  - `egg` – marche directement avec Emacsen même en console (optionnel)
- installez des terminaux compatibles (pour toutes) :
  - `xterm` – X (pour ISO-8859-1 and UTF-8),
  - `kterm` – X (pour Japonais EUC), et
  - `mlterm` – X (multilingue).
- ajoutez tous les paquets de polices nécessaires (pour toutes)
- créez `~/.xsession` qui définit l'environnement X spécifique à l'utilisateur comme décrit dans 'Sessions X personnalisées' page 137 (pour toutes) :

```
#!/bin/sh
# Cela fait marcher su avec X
if [ -z "$XAUTHORITY" ]; then
        XAUTHORITY=$HOME/.Xauthority
        export XAUTHORITY
fi

# règle l'environnement spécifique par le système de menu debian
# remet à zéro la locale
unset LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES
unset LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
unset LC_IDENTIFICATION LC_ALL LANG LANGUAGE PAGER
# règle la locale par défaut dans X
LANG=C
# exporte la locale
export LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES
export LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
export LC_IDENTIFICATION LC_ALL LANG LANGUAGE PAGER
###

# active la méthode d'entrée pour le Japonais avec kinput2
kinput2 &
```

---

[8] Dans cet exemple, deux bogues de `blackbox` sont contournés. J'utilise `sh -c` comme shell. De plus, `~/.menu/*` n'est pas utilisé mais `/etc/menu/*`, qui requiert les privilèges root.

```
XMODIFIERS="@im=kinput2"
export XMODIFIERS
# gestionnaire de fenêtres blackbox (léger)
exec /usr/bin/blackbox
#exec xfwm
#exec wmaker
```

- réglez la locale dans `~/.bash_profile` pour les consoles Linux (pour toutes)
- supprimez la configuration de la locale de `~/.bashrc`, si elle existait (pour toutes)
- créez quelques fichiers dans `/etc/menu` (pour toutes) :
  - `/etc/menu/xterm-local` : (ajoute de nouvelles entrées dans le menu) [9]

```
?package(xterm):\
 needs=x11\
 section=XShells\
 longtitle="XTerm: terminal emulator (en_US.ISO-8859-1)"\
 title="XTerm (en_US.ISO-8859-1)"\
 command="sh -c ’LC_ALL=en_US.ISO-8859-1 xterm’"
?package(xterm):\
 needs=x11\
 section=XShells\
 longtitle="XTerm: terminal emulator (de_DE.ISO-8859-1)"\
 title="XTerm (de_DE.ISO-8859-1)"\
 command="sh -c ’LC_ALL=de_DE.ISO-8859-1 xterm -T xterm-de’"
?package(xterm):\
 needs=x11\
 section=XShells\
 longtitle="XTerm: terminal emulator for X with Unicode support
 (Japanese)"\
 title="UXTerm (ja_JP.UTF-8)"\
 command="sh -c ’LC_ALL=ja_JP.UTF-8 uxterm’"
```

  - `/etc/menu/kterm` : (remplace les entrées par défaut) [10]

```
?package(kterm):\
 needs="x11"\
 section="XShells"\
 command="sh -c ’LC_ALL=ja_JP.eucJP PAGER=w3m /usr/X11R6/bin/kterm -xi
 title="Kanji Terminal"
?package(kterm):\
 needs="x11"\
 section="XShells"\
 command="sh -c ’LANG=ja_JP.eucJP \
          LC_MESSAGES=en_US.ISO-8859-1  PAGER=w3m /usr/X11R6/bin/kterm
 title="Kanji Terminal (bilingal)"
```

---

[9] Utilisez un nom de fichier qui ne soit pas celui d’un nom de paquet.

[10] Utilisez un nom qui remplace le nom du paquet.

  - `/etc/menu/mozilla-local` : (ajoute un nouveau raccourci) [11]

```
       ?package(mozilla-browser):needs="x11" section="/" \
               title=" Mozilla Navigator" command="mozilla-1.5"
        hints="Web browsers"\
               icon=/usr/share/pixmaps/mozilla.xpm
```

  - exécutez `update-menus` depuis le compte root.
- ajoutez les lignes suivantes à `~/.muttrc` (pour le Japonais) :

```
# le support UTF-8 n'est pas populaire dans l'environnement japonais
# EMACS
# codage 7 bits pour iso-2022-jp est plus facile pour tout le monde
# ordre de codage par défaut = us-ascii > iso-8859-1 > utf-8 > iso-2022-jp
set send_charset="us-ascii:iso-8859-1:iso-2022-jp"
set allow_8bit=no
```

- activez XIM `kinput2` pour les applications X (pour le Japonais) :
  - ajoutez `*inputMethod: kinput2` et `KTerm*VT100*OpenIm: true` à votre fichier de ressources X `~/.Xresources` (on dirait que Debian le fait automatiquement).
  - Quelques applications (comme `mlterm`) permettent aussi de configurer `*inputMethod:` et d'autres paramètres dynamiquement au démarrage (appuyez sur **Ctrl-Bouton-3** dans `mlterm`).
- lancez X avec `startx` ou à partir d'un des gestionnaires d'affichage (`xdm`, `gdm`, `kdm`, `wdm`, ...) (pour toutes)
- lancez une application compatible avec le Japonais comme Vim 6, (x)emacs21, mc-4.5, mutt-1.4, ... dans `kterm` (pour le Japonais). (Emacs semble être populaire, bien que je ne l'utilise pas.)
- appuyez sur **Shift+Space** pour activer/désactiver l'entrée de caractères japonais (pour le Japonais).
- lisez les pages de manuel localisées en tapant la commande dans la console localisée (pour toutes).

Pour d'autre support sur CJK, consultez les sections suivantes et SuSE pages for CJK (`http://www.suse.de/~mfabian/suse-cjk/suse-cjk.html`).

9.7.10 Méthodes d'entrée alternatives pour X

Il existe beaucoup de paquets pour le support de méthodes d'entée alternatives pour X :

Language	LC_CTYPE	XIM server	XMODIFIERS	Démarrage par
Japanese	ja_JP*	kinput2	"@im=kinput2"	Shift-Espace
Korean	ko_KR*	ami	"@im=Ami"	Shift-Espace

---

[11] Le slash de `section="/"` permet d'ajouter l'entrée dans le menu initial, et l'espace au début de `title=" Mozilla Navigator"` permet d'avoir cette entrée en haut de la liste.

```
Chinese(T) zh_TW.Big5   xcin       "@im=xcin-zh_TW.big5"   Ctrl-Espace
Chinese(S) zh_CN.GB2312 xcin       "@im=xcin-zh_CN.GB2312" Ctrl-Espace
```

La méthode d'entrée japonaise `kinput2` est offerte par les paquets tels que `kinput2-canna-wnn`, `kinput2-canna` et `kinput2-wnn`. Le Japonais a besoin d'un serveur de dictionnaire comme `canna` et `freewnn-jserver` pour être utilisable.

9.7.11 Émulateurs de terminal X

Il existe beaucoup de consoles X qui supportent les codages sur 8 bits simples lorsque les bons paquets de polices sont installés :

- `xterm` – l'émulateur de terminal X
- `gnome-terminal` – `xterm` pour Gnome
- `konsole` – `xterm` pour KDE
- `rxvt` – terminal VT102 (plus léger)
- `aterm` – VT102 pour Afterstep
- `eterm` – VT102 pour Enlightment
- `wterm` – VT102 pour WindowMaker

Le support des codages multi-octets pour la console X est fourni par `xterm` par le codage UTF-8 ('Support UTF-8 pour l'émulateur de terminal X' de la présente page). Le support d'autres codages traditionnels est en progrès (en 2003). Les paquets suivants offrent le support de codages traditionnels :

- `aterm-ml` – multilingue
- `kterm` – multilinge (Japonais, . . .)
- `rxvt-ml` – multilingue
- `wterm-ml` – multilingue
- `cxterm-big5` – Chinois (Trad., Big5)
- `cxterm-gb` – Chinois (Simp., GB)
- `cxterm-ks` – Chinois (KS)
- `cxterm-jis` – Japonais
- `hanterm-classic` – Coréen (Hangul)
- `hanterm-xf` – Coréen (Hangul)
- `hztty` – Chinois (GB, Big5, zW/HZ)

Pour `kterm` (et peut-être d'autres), vous pouvez activer XIM par le menu après un Ctrl-click-milieu.

9.7.12 Support UTF-8 pour l'émulateur de terminal X

Le support UTF-8 pour l'émulateur de terminal X est founi par le programme `uxterm` du paquet `xterm` de XFree86 4.x. Cela permet le support pour toutes les langues. C'est une encapsulation de `xterm(1)` qui l'invoque en paramétrant la classe de ressource « UXTerm ».

Par exemple, pour activer l'affichage de l'Anglais, du Russe, du Japonais, du Chinois et du Coréen, ajoutez ce qui suit à votre `~/.Xresources` après avoir installé les polices nécessaires :

```
! set large font
UXTerm*font: -misc-fixed-medium-r-normal-*-18-120-100-100-c-90-iso10646-1
! Use XIM for Japanese
*inputMethod: kinput2
```

Ensuite exécutez `xrdb -merge ~/.Xresources` pour mettre à jour les ressources X, comme décrit dans 'Ressources X' page 141.

La plupart des programmes console les plus populaires, comme `vim`, `mutt` et `emacs`, aient été rendus compatibles avec UTF-8 récemment -Woody-Sarge). Des programmes comme `mc` ne sont toujours pas compatibles UTF-8, mais gèrent les codages 8 bits. Si vous éditez une partie en ASCII 7 bits d'un fichier inconnu ou avec un codage mixte, il est plus prudent d'utiliser un éditeur gérant les codages 8 bits.

Voir The Unicode HOWTO (`http://www.tldp.org/HOWTO/Unicode-HOWTO.html`).

9.7.13 Exemple pour UTF-8 avec la console framebuffer

Le support UTF-8 pour la console FB est fournit par `bterm` utilisé dans `debian-installer`.

9.7.14 Après locale

Lorsque vous configurez un système pour la première fois pour un **environnement de langue nationale**, considérez l'utilisation de `tasksel` ou `aptitude` pour connaître les paquets sélectionnés en choisissant la tâche d'environnement de langue correspondante. Le choix de paquets est utile même pour la configuration multilingue. Si vous rencontrez des problèmes de dépendances pendant l'installation de votre système configuré, évitez d'installer les logiciels qui créent les conflits. Vous aurez peut-être à utiliser `update-alternative` pour retrouver l'état originel de certaines commandes puisque celles nouvellement installées peuvent avoir des priorités plus élevées que celles existantes.

Les programmes importants les plus récents qui utilisent la glibc 2.2 sont pour la plupart internationalisés. Les versions localisées spéciales comme `jvim` pour VIM peuvent ne pas être nécessaires et ses fonctionnalités sont offertes dans `vim` version 6.0 pour X. En réalité, il est encore difficile à utiliser. Puisque `jvim` possède une version avec le support de l'entrée des caractères japonais (`canna`) compilé directement même avec la console et résoud beaucoup d'autres problèmes spécifiques au Japonais, vous pouvez toujours en avoir besoin :-)

Certains programmes peuvent nécessiter une configuration en plus de la `locale` pour obtenir un environnement de travail confortable. Le paquet `language-env` avec sa commande `set-language-env` est un script qui facilite la configuration de l'environnement de langue nationale sur un système Debian.

Voir aussi le document sur l'internationalisation, Introduction to i18n (`http://www.debian.org/doc/manuals/intro-i18n/`). Il est destiné aux développeurs mais est aussi utile aux administrateurs système.

9.8 Multilingualisation (m17n)

La 'Localisation (l10n)' page 154, activée par les paquets `language-env` et d'autres, a pour objectif de permettre une localisation monolingue. Ces paquets utilisent aussi les codages traditionnels pour le codage du texte. Vous ne pouvez pas mélanger des textes en Français et en Japonais dans de tels environnements car ils utilisent les codages incompatibles entre eux ISO-8859-1 et EUC-JP.

Vous pouvez obtenir un bureau multilingue utilisant le codage UTF-8 en utilisant Gnome ou KDE avec l'une des locales UTF-8 disponibles. (Sarge) Dans de tels environnements, vous pouvez mélanger des caractères Anglais, Chinois, Russe et Japonais dans les logiciels compatibles avec UTF-8. [12]

Dans un tel environnement, la nouvelle méthode d'entrée (IM, *Input Method*) multilingue du paquet `scim` est préférée. La méthode d'entrée de `scim` est activée et désactivée en tapant `Ctrl-Espace`. Le moteur de conversion de l'entrée peut être changé dans la petite fenêtre de SCIM.

`vim` offers the multilingualized environment and can handle both UTF-8 and conventionally encoded files (EUC-JP, ISO-8859-1, . . .) when it is run under the UTF-8 console such as `gnome-teminal`. See vim help message with pressing [Esc] and typing `:help mbyte.txt`.

---

[12] `language-env` n'est pas très utile dans un environnement multilingue.

Chapitre 10

Configuration réseau

Ce chapitre présente l'administration réseau avec Debian. Pour une introduction générale au réseau sous GNU/Linux, lisez le Net-HOWTO (`http://www.tldp.org/HOWTO/Net-HOWTO/index.html`).

Afin qu'une machine Debian accède à l'Internet, ses interfaces réseau doivent être supportées par le noyau et configurées.

Tout d'abord, il faut que le noyau supporte les périphériques réseau, comme les cartes Ethernet, les cartes Wi-Fi et les modems. Pour obtenir ce support, vous devrez peut-être recompiler le noyau ou ajouter des modules comme décrit dans 'Le noyau Linux et Debian' page 85.

La configuration des périphériques réseau est expliquée ci-dessous. L'information contenue dans ce chapitre a été mise à jour pour Sarge. La plus grande partie s'applique aussi pour les versions antérieures.

10.1 Bases des réseaux IP

Une machine Debian peut avoir plusieurs interfaces ayant chacune une adresse *Internet Protocol* (IP) différente. Les interfaces peuvent être de plusieurs types, dont :

- Loopback : `lo`
- Ethernet : `eth0`, `eth1`, ...
- Wi-Fi : `wifi0`, `wlan1`, ...[1]
- Token Ring : `tr0`, `tr1`, ...
- PPP : `ppp0`, `ppp1`, ...

Il existe un grand nombre d'autres périphériques réseau disponibles, comme SLIP, PLIP (IP sur ligne série et parallèle), périphériques *shaper* pour contrôler le trafic sur certaines interfaces, Frame Relay, AX.25, X.25, ARCnet et LocalTalk.

---

[1] Notez qu'une interface Wi-Fi est en fait un alias pour une interface Ethernet qui donne accès aux paramètres de configuration spécifiques à Wi-Fi. Ces paramètres sont contrôlés par le programme `iwconfig`.

Chaque interface réseau connectée directement à l'Internet (ou à un réseau IP) est identifiée par une adresse IP unique sur 32 bits. [2] L'adresse IP est divisée en deux parties : l'une identifie le réseau et l'autre la machine. Si vous prenez une adresse IP, mettez à 1 les bits de l'adresse réseau et à 0 les bits de l'adresse de la machine, vous obtenez le masque de réseau.

Traditionnellement, les réseaux IP ont été groupés en classes dont les adresses de réseau font 8, 16 ou 24 bits de longueur. [3]

```
          adresse IP                      masque réseau   taille
Class A   1.0.0.0     - 126.255.255.255  255.0.0.0     =  /8
Class B   128.0.0.0   - 191.255.255.255  255.255.0.0   = /16
Class C   192.0.0.0   - 223.255.255.255  255.255.255.0 = /24
```

Les adresses IP n'étant pas dans ces plages sont utilisées dans des buts précis.

Il y a dans chaque classe des plages d'adresses réservées pour l'utilisation dans des réseaux locaux. Il est garanti que ces adresses ne peuvent pas entrer en conflit avec les adreses utilisées sur l'Internet (et par conséquent, si une de ces adresses est assignée à une machine alors cette machine ne peut pas accéder à l'Internet directement et doit y accéder par une passerelle qui agit comme *proxy* pour des services spécifiques ou fait de la translation d'adresse réseau – NAT). Ces plages d'adresses sont données dans la table suivante avec le nombre de plages dans chaque classe.

```
          addresse réseau               taille  nombre
Class A   10.x.x.x                      /8      1
Class B   172.16.x.x -  172.31.x.x      /16     16
Class C   192.168.0.x - 192.168.255.x   /24     256
```

La première adresse dans un réseau IP est l'adresse du réseau lui-même. La dernière adresse est l'adresse de diffusion du réseau. [4] Toutes les autres adresses peuvent être allouées à des machines sur le réseau. Parmi celles-ci, la première ou la dernière adresse est généralement allouée à la passerelle Internet du réseau.

La table de routage contient l'information sur la façon d'envoyer les paquets IP vers leur destination. Voici un exemple de table de routage prise sur une machine Debian d'un réseau local avec une adresse IP 192.168.50.x/24. La machine 192.168.50.1 (aussi sur le réseau local) est un routeur pour le réseau d'entreprise 172.20.x.x/16 et la machine 192.168.50.254 (aussi sur le réseau local) est un routeur vers l'Internet.

```
# route
Kernel IP routing table
```

---

[2] Cela est vrai si IP version 4 est utilisé. Dans IPv6, les adresses font 128 bits. Voir `http://www.ipv6.org/`.

[3] Ce systèmes n'est pas souple et a gâché beaucoup d'adresses IP, aussi aujourd'hui les réseaux IPv4 sont alloués avec une adresse réseau de taille variable.

[4] L'adresse du réseau peut être obtenue en effectuant un ET bit à bit sur une adresse du réseau et le masque de réseau.

```
Destination   Gateway        Genmask       Flags Metric Ref Use Iface
127.0.0.0     *              255.0.0.0     U     0      0     2 lo
192.168.50.0  *              255.255.255.0 U     0      0   137 eth0
172.20.0.0    192.168.50.1   255.255.0.0   UG    1      0     7 eth0
default       192.168.50.254 0.0.0.0       UG    1      0    36 eth0
```

- La première ligne après l'en-tête dit que le trafic destiné au réseau `127.x.x.x` sera routé par `lo`, l'interface boucle.
- La deuxième ligne dit que le trafic destiné aux machines du réseau local sera routé par `eth0`.
- La troisième ligne dit que le trafic destiné au réseau d'entreprise sera routé par la passerelle `192.168.50.1` par `eth0`.
- La quatrième ligne dit que le trafic destiné à l'Internet sera routé par la passerelle `192.168.50.254` par `eth0`.

Les adresses IP de la table peuvent aussi apparaître comme noms obtenus en cherchant les adresses dans le fichier `/etc/networks` ou en utilisant le résolveur de la bibliothèque C.

En plus du routage, le noyau peut effectuer de la translation d'adresse, de la modulation de trafic et du filtrage.

Voir Net-HOWTO (`http://www.tldp.org/HOWTO/Net-HOWTO/index.html`) et other networking HOWTOs (`http://www.tldp.org/HOWTO/Networking-Overview-HOWTO.html`) pour plus d'information.

10.2 Configuration réseau bas niveau

Les outils de configuration réseau bas niveau traditionnels des systèmes GNU/Linux sont les programmes `ifconfig` et `route` qui viennent dans le paquet `net-tools`. Ces outils ont officiellement été rendus obsolètes par `ip` qui est dans le paquet `iproute`. Le programme `ip` fonctionne avec Linux 2.2 et supérieurs et a plus de possibilités que les anciens outils. Cependant, les anciens outils fonctionnent toujours et sont plus familiers à de nombreux utilisateurs.

10.2.1 Configuration réseau bas niveau – `ifconfig` et `route`

Voici une illustration de comment changer l'adresse IP de l'interface `eth0` de `192.168.0.3` en `192.168.0.111` et faire de `eth0` la route vers le réseau `10.0.0.0` via `192.168.0.1`. On commence par exécuter `ifconfig` et `route` sans arguments pour afficher l'état courant de toutes les interfaces réseau et du routage.

```
# ifconfig
eth0 Link encap:Ethernet  HWaddr 08:00:46:7A:02:B0
     inet addr:192.168.0.3  Bcast:192.168.0.255  Mask:255.255.255.0
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     RX packets:23363 errors:0 dropped:0 overruns:0 frame:0
```

```
          TX packets:21798 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:13479541 (12.8 MiB)  TX bytes:20262643 (19.3 MiB)
          Interrupt:9

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:230172 errors:0 dropped:0 overruns:0 frame:0
          TX packets:230172 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:22685256 (21.6 MiB)  TX bytes:22685256 (21.6 MiB)
# route
Kernel IP routing table
Destination  Gateway      Genmask          Flags Metric Ref Use Iface
192.168.0.0  *            255.255.0.0      U     0      0     0 eth0
default      192.168.0.1  255.255.255.255  UG    0      0     0 eth0
```

D’abord, on désactive l’interface.

```
# ifconfig eth0 inet down
# ifconfig
lo   Link encap:Local Loopback
  ... (plus d’entrée eth0)
# route
  ... (plus d’entrée dans la table de routage)
```

Ensuite, on l’active avec la nouvelle adresse IP et le nouveau routage.

```
# ifconfig eth0 inet up 192.168.0.111 \
          netmask 255.255.0.0 broadcast 192.168.255.255
# route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.0.1 dev eth0
```

Le résultat :

```
# ifconfig
eth0 Link encap:Ethernet  HWaddr 08:00:46:7A:02:B0
     inet addr:192.168.0.111  Bcast:192.168.255.255  Mask:255.255.0.0
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     ...

lo   Link encap:Local Loopback
     inet addr:127.0.0.1  Mask:255.0.0.0
     ...
```

```
# route
Kernel IP routing table
Destination  Gateway      Genmask      Flags Metric Ref Use Iface
192.168.0.0  *            255.255.0.0  U     0      0     0 eth0
10.0.0.0     192.168.0.1  255.0.0.0    UG    0      0     0 eth0
```

Pour plus d'information voir `ifconfig(8)` et `route(8)`.

10.2.2 Configuration réseau bas niveau – `ip`

Les commandes `ip` équivalentes à `ifconfig` et `route`, décrits ci-dessus, sont :

- `ip link show`
- `ip route list`
- `ip link set eth0 down`
- `ip addr del dev eth0 local 192.168.0.3`
- `ip addr add dev eth0 local 192.168.0.111/16 broadcast 192.168.255.255`
- `ip link set eth0 up`
- `ip route add dev eth0 to 10.0.0.0/8 src 192.168.0.111 via 192.168.0.1`

Le programme `ip` affiche la syntaxe de ses commandes lorsqu'il est exécuté avec l'argument `help`. Par exemple, `ip link help` affiche :

```
Usage: ip link set DEVICE { up | down | arp { on | off } |
                     dynamic { on | off } |
                     multicast { on | off } | txqueuelen PACKETS |
                     name NEWNAME |
                     address LLADDR | broadcast LLADDR |
                     mtu MTU }
       ip link show [ DEVICE ]
```

Voir aussi `ip(8)`.

10.2.3 Configurer une interface Wi-Fi

Pour les interfaces Wi-Fi, le programme `iwconfig` qui vient dans le paquet `wireless-tools` est utilisé en plus de `ifconfig` ou `ip`.

Voir `iwconfig(8)`.

10.2.4 Configurer une interface PPP

Si vous accédez à l'Internet par un modem RTC alors la connexion est négociée en utilisant le protocole Point-to-Point Protocol (PPP). Ces connexions sont accédées par les interfaces `ppp0`, `ppp1`, etc.

Une interface PPP est gérée par le démon `pppd` qui vient dans le paquet `ppp`. Ainsi, pour l'utilisateur, configurer une interface PPP signifie configurer `pppd`.

Configurer `pppd` à la main

Pour qu'un lien réseau soit établi, un port de communication (habituellement un port série) doit être ouvert, des commandes doivent être envoyées au périphérique de communication (habituellement un modem), un numéro de téléphone doit être appelé, le démon PPP distant doit identifier l'utilisateur, une interface PPP doit être créée et les tables de routage doivent être modifiées pour que le trafic soit envoyé sur ce lien. `pppd` peut faire tout cela et possède donc une longue liste d'options. Ces options sont décrites dans `pppd(8)`.

Sur un système Debian, les options globales sont définies dans `/etc/ppp/options`. Les options spécifiques à l'utilisateur sont définies dans `~/.ppprc`. Les options qui dépendent du port de communication utilisé sont définies dans `/etc/ppp/options.`*`portname`*. Par exemple, supposons que vous avez deux modems—un modem intégré Lucent LT accédé par `/dev/LT-modem` et un modem externe accédé par `/dev/ttyS0`. Créez les deux fichiers d'options suivants.

```
# cat > /etc/ppp/options.LT-modem <<EOF
115200
init "/usr/sbin/chat -f /etc/chatscripts/setup-LT-modem"
EOF
# cat > /etc/ppp/options.ttyS0 <<EOF
115200
init "/usr/sbin/chat -f /etc/chatscripts/setup-ttyS0"
EOF
```

Ils réfèrent aux script `chat` suivants. D'abord `/etc/chatscripts/setup-LT-modem`.

```
ABORT ERROR
'' ATZ
OK 'ATW2X2 S7=70 S11=55'
OK AT
```

Ensuite, `/etc/chatscripts/setup-ttyS0`.

```
ABORT ERROR
'' ATZ
OK 'ATL1M1Q0V1W2X4&C1&D2 S6=4 S7=70 S11=55 S95=63 S109=1 +FCLASS=0'
OK AT
```

Le contenu de ces fichiers doit dépendre de votre matériel, bien sûr.

Les options peuvent aussi être données à `pppd` comme arguments.

Dans Debian `pppd` est habituellement démarré par la commande `pon`. Lorsque `pon` est utilisé, son premier argument définit un fichier d'options dans `/etc/ppp/peers/`, qui est aussi lu par `pppd`. [5] C'est ici que vous définissez des options spécifiques à un hôte distant particulier—par exemple, un Fournisseur d'Accès à Internet (FAI).

Supposons par exemple que vous vous déplacez entre Amsterdam et La Haye. Dans chaque ville vous avez accçs à deux FAI—Planet et KPN. D'abord, créez un fichier d'options de base pour chaque FAI.

```
# cat > /etc/ppp/peers/KPN <<EOF
remotename KPN
noauth
user kpn
noipdefault
ipparam KPN
EOF
# cat > /etc/ppp/peers/Planet <<EOF
remotename Planet
auth
user user3579@planet.nl
noipdefault
mru 1000
mtu 1000
ipparam Planet
EOF
```

Ces fichiers définissent les options qui diffèrent entre les deux FAI. Les options communes peuvent être placées dans `/etc/ppp/options` ou dans l'un des fichiers spécifiques à une interface si besoin.

Maintenant créez les fichiers d'options pour chaque FAI dans chaque ville. Dans notre exemple la seule différence entre se connecter à un FAI dans une ville et se connecter à l'autre est le script `chat` qui est requis. (Le script `chat` est différent parce que le numéro de téléphone local est différent.)

```
# cat > /etc/ppp/peers/KPN-Amsterdam <<EOF
connect "/usr/sbin/chat -v -f /etc/chatscripts/KPN-Amsterdam"
file /etc/ppp/peers/KPN
EOF
# cat > /etc/ppp/peers/KPN-DenHaag <<EOF
connect "/usr/sbin/chat -v -f /etc/chatscripts/KPN-DenHaag"
file /etc/ppp/peers/KPN
EOF
```

---

[5] Ce fichier d'options est inclus en utilisant l'option `call`.

```
# cat > /etc/ppp/peers/Planet-Amsterdam <<EOF
connect "/usr/sbin/chat -v -f /etc/chatscripts/Planet-Amsterdam"
file /etc/ppp/peers/Planet
EOF
# cat > /etc/ppp/peers/Planet-DenHaag <<EOF
connect "/usr/sbin/chat -v -f /etc/chatscripts/Planet-DenHaag"
file /etc/ppp/peers/Planet
EOF
```

Les directives `file` incluent chacune un des fichiers d'options défini ci-dessus. La directive `connect` spécifie la commande que `pppd` utilise pour effectuer la connexion. Normalement, on utilise le programme `chat` pour cela, en adaptant le script `chat` à chaque FAI. Voici les scripts `chat` pour La Haye ; les scripts `chat` pour Amsterdam peuvent être similaires excepté les numéros de téléphone ou ils peuvent être différents si le FAI offre un service à travers une autre compagnie.

```
# cat > /etc/chatscripts/KPN-DenHaag <<EOF
ABORT BUSY
ABORT 'NO CARRIER'
ABORT VOICE
ABORT 'NO DIALTONE'
ABORT 'NO DIAL TONE'
ABORT 'NO ANSWER'
ABORT ERROR
OK-AT-OK ATDT 0676012321
CONNECT \d\c
EOF
# cat > /etc/chatscripts/Planet-DenHaag <<EOF
ABORT BUSY
ABORT 'NO CARRIER'
ABORT VOICE
ABORT 'NO DIALTONE'
ABORT 'NO DIAL TONE'
ABORT 'NO ANSWER'
ABORT ERROR
OK-AT-OK ATDT 0676002505
CONNECT \d\c
EOF
```

Pour se connecter à ces ISP, vous devez avoir un nom de client et un mot de passe que `pppd` peut fournir à l'hôte distant à la demande. Cette information est définie soit dans `/etc/ppp/pap-secrets` (si le protocole PAP est utilisé) soit dans `/etc/ppp/chap-secrets` (si le protocole CHAP est utilisé). Bien que CHAP soit plus sûr, PAP est toujours beaucoup plus utilisé. Parce que ces fichiers contiennent des informations secrètes, seul root doit avoir la permission de les lire et écrire. Le format de ces fichiers est expliqué dans `pppd(8)`. Un « secret » (troisième champ) est cherché dans le fichier par recherche du nom de client (premier champ) et/ou du nom du serveur (deuxième champ). Lorsque vous vous connectez à un FAI, vous ne connaissez généralement pas le nom du serveur, donc il faut fournir un nom de client à la place ; cela est fait sur la ligne `user` de `peers/KPN` et `peers/Planet` ci-dessus.

```
# client name        server name  secret
kpn                  *            kpn
user3579@planet.nl   *            myfavoritepet
```

Configurer `pppd` avec `pppconfig`

Un moyen rapide de configurer `pppd` est d'utiliser le programme `pppconfig` qui vient dans le paquet du même nom. `pppconfig` configure les fichiers comme ceux ci-dessus en posant des questions à l'utilisateur.

Si vous choisissez d'utiliser `resolvconf` alors assurez-vous de sélectionner « *None* » à l'écran « *Configure Nameservers* ».

Configurer une interface PPP avec `wvdial`

Une approche différente pour utiliser `pppd` est de l'exécuter avec `wvdial` qui vient dans le paquet du même nom. Au lieu de faire exécuter `chat` à `pppd` pour numéroter et négocier une connexion, `wvdial` effectue la numérotation et la négociation initiale et ensuite démarre `pppd` pour faire le reste. Avec seulement un numéro de téléphone, un nom d'utilisateur et un mot de passe, `wvdial` réussit à effectuer la connexion dans la plupart des cas.

10.3 Nommer l'ordinateur

10.3.1 Nom de machine

Un système Debian a parfois besoin de s'identifier par un nom. Pour cela, un **nom de machine** est maintenu par le noyau.

Le script d'initialisation `/etc/init.d/hostname.sh` défini le nom de machine au démarrage (en utilisant la commande `hostname`) au nom donné dans `/etc/hostname`. Ce fichier doit contenir **seulement** le nom de machine, pas un nom de domaine complet.

Pour afficher le nom de machine actuel, exécutez `hostname` sans argument.

10.3.2 Nom de courriel

Le **nom de courriel** d'une machine est le nom que les programmes de courrier électronique utilisent pour l'identifier. Le fichier `/etc/mailname` contient ce nom suivi d'un retour à la ligne. Le nom de courriel est habituellement l'un des noms de domaine complets de la machine. Voir `mailname(5)`.

Ce que les destinataires de courriers électroniques voient dans le champ `From:` des courriels envoyés depuis votre machine Debian dépend de la façon dont les agents pour utilisateur de courriel (Mail User Agent, MUA) et agents de transport de courriel (Mail Transport Agent, MTA) sont configurés. Supposons qu'un utilisateur local *foo* envoie un courriel depuis une machine dont le nom de courriel est *myhost.dom*. Le champ `From:` du courriel sortant sera :

- "`From:` *foo@myhost.dom*" si le MUA n'a pas de champ `From:` configuré ;
- "`From:` *bar@myhost.dom*" si le MUA est configuré avec `From:` *bar* ;
- "`From:` *bar@bogus.dom*" si le MUA est configuré avec `From:` *bar@bogus.dom*.

Même lorsque le MUA a un champ `From:` configuré, le MTA peut ajouter un champ `Sender:` *foo@herman.dom* pour indiquer son origine réelle.

Bien sûr, lorsque le MTA réécrit l'adresse comme décrit dans 'Configurer Exim pour récupérer les adresses non existantes' page 152 et 'Configurer Exim pour réécrire les adresses sélectionnées sur les courriels sortants' page 152, l'adresse e-mail vue par le destinataire peut être changée en n'importe quoi.

10.4 Domain Name Service (DNS)

Les machines sont identifiées par un nom de domaine comme par une adresse IP. DNS est un système client-serveur dans lequel des résolveurs de noms consultent des serveurs de noms afin d'associer les noms de domaine avec les adresses IP et d'autres propriétés des machines. La bibliothèque GNU C `resolver(3)` peut aussi chercher des adresses IP dans des fichiers ou consulter Network Information Services (NIS).

Pour voir quel nom de domaine est associé avec la machine locale, utilisez la commande `hostname --fqdn`. Cela affiche le premier nom de domaine complet que le résolveur trouve pour la machine locale. [6]

10.4.1 Le résolveur

Trouver quelle adresse IP est associée avec un domaine particulier est le travail du résolveur. Le résolveur le plus communément utilisé est l'ensemble de fonctions qui ont ce nom (`resolver(3)`) dans la bibliothèque GNU C. Un autre est le résolveur FireDNS qui vient dans le paquet `libfiredns`.

La façon dont le résolveur de la bibliothèque C résout les noms est définie par la ligne `hosts` du fichier de configuration `/etc/nsswitch.conf`. Cette ligne liste les services qui doivent être utilisés pour résoudre un nom : par exemple, `dns`, `files`, `nis`, `nisplus`. [7] Voir

---

[6] En fait, c'est le FQDN (Fully Qualified Domain Name, nom de domaine complet valide) retourné par `gethostbyname(2)` pour le nom de machine retourné par `gethostname(2)`.

[7] La façon dont le résolveur résout les noms est aussi théoriquement définie par le fichier de configuration `/etc /host.conf` ? La ligne `order` de ce fichier liste les méthodes qui doivent être utilisées pour résoudre un nom :

`nsswitch.conf(5)`. Si le service `files`, le comportement est aussi défini par le fichier de configuration `/etc/hosts`. Voir `hosts(5)`.

Tous les fichiers ci-dessus sont statiques et peuvent être édités avec votre éditeur favori.

Si le service `dns` est utilisé, le comportement du résolveur est aussi défini par le fichier de configuration `/etc/resolv.conf`. Voir `resolv.conf(5)`. Une des fonctions importantes de `/etc/resolv.conf` est de lister les adresses IP des serveurs de noms qui seront contactés pour résoudre le nom. Cette liste dépend souvent de l'environnement réseau, qui peut changer de temps en temps pendant que votre machine tourne. Des programmes comme `pppd` et `dhclient` sont capables de manipuler `resolv.conf` pour y ajouter ou enlever des lignes, mais ces possibilités ne marchent pas toujours correctement et peuvent créer des conflits mutuels. Le paquet `resolvconf` résout le problème en fournissant un cadre standard pour mettre à jour ce fichier. Voir 'Gérer l'information sur les serveurs de noms – `resolvconf`' de la présente page.

10.4.2 Gérer l'information sur les serveurs de noms – `resolvconf`

Le paquet `resolvconf` fournit un cadre pour la gestion dynamique des informations sur les serveurs de noms disponibles. Il résout le vieux problème du maintien de listes dynamiques de serveurs de noms pour le résolveur et le cache DNS. `resolvconf` se place comme intermédiaire entre les programmes qui contrôlent les interfaces réseau et fournissent l'information sur les serveurs de noms, et les applications qui ont besoin de l'information sur les serveurs de noms.

`resolvconf` est conçu pour fonctionner sans aucune configuration manuelle préalable. Cependant, le paquet est relativement nouveau et peut nécessiter des interventions manuelles pour qu'il fonctionne correctement. C'est certainement vrai si vous avez personnalisé des paquets pour qu'ils mettent à jour `/etc/resolv.conf` : il faudra désactiver ces modifications.

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

10.4.3 Garder en mémoire les noms résolus – `dnsmasq`

Sauf si votre serveur de nom est le serveur officiel d'un domaine, il vaut mieux utiliser un serveur de noms cache local comme `dnsmasq`. Il fonctionne bien avec `resolvconf`.

10.4.4 Fournir le Domain Name Service – `bind`

Si vous devez fournir un service de noms officiel pour un domaine alors il vous faut un serveur de noms complet comme `named` qui vient dans l'un des paquets `bind9` ou `bind`. `bind9` est recommandé pour une nouvelle installation.

---

par exemple `bind`, `hosts`, `nis`. Voir `host.conf(5)`. Je pense que cette ligne a été remplacée par la ligne `hosts` de `nsswitch.conf` mais je n'en suis pas sûr.

Pour installer `bind9`, installez ces paquets de base : `bind9`; `dnsutils`. Vous voudrez peut-être aussi installer ces utilitaires : `bind9-host`; `dns-browse`; `dnscvsutil`; `nslint`. Vous voudrez peut-être aussi installer ce paquet de documentation : `bind9-doc`. Vous voudrez peut-être aussi installer ces paquets de développement : `libbind-dev`; `libnet-dns-perl`. Si vous configurez des interfaces par DHCP alors vous trouverez peut-être ce paquet utile : `dhcp-dns`.

Installez `bind9` ou `dpkg-reconfigure` pour faire la configuration de base. La configuration consiste à éditer `named.conf`. Dans Debian, ce fichier se trouve dans `/etc/bind/` et est principalement utilisé pour définir les zones DNS de base ; il inclut deux autres fichiers : `named.conf.local`, pour définir les zones locales, et `named.conf.options`, pour définir les options. (Ce dernier est traité par `resolvconf` pour produire `/var/run/bind/named.options` qui est le même que l'original sauf que l'entrée `forwarders` est une liste des serveurs de noms non locaux disponibles. Pour utiliser cela, changez la ligne `include` de `named.conf` pour qu'elle inclue `/var/run/bind/named.options`. Voir 'Gérer l'information sur les serveurs de noms – `resolvconf`' page précédente.)

Des fichiers de base de données cités dans `named.conf*` sans un chemin complet seront stockés dans `/var/cache/bind/`. C'est le bon endroit pour stocker les fichiers générés par `named` : par exemple, les fichiers de données pour les zones dont le démon est secondaire. Les fichiers de données statiques de `/etc/bind/` sont et doivent être cités dans `named.conf` par leur chemin complet. Voir `/usr/share/doc/bind9/README.Debian.gz` pour plus de détails.

10.5 Configurer les interfaces réseau par DHCP

La configuration bas niveau des interfaces réseau peut être automatisée par le moyen de Dynamic Host Configuration Protocol (DHCP). Votre pare-feu ou votre routeur ou votre FAI haut-débit fournit peut-être l'adresse IP et d'autres paramètres de cette façon.

Pour que cela fonctionne, vous devez installer un des paquets suivants :
- `dhcp3-client` (version 3, Internet Software Consortium)
- `dhcpcd` (Yoichi Hariguchi et Sergei Viznyuk)
- `pump` (Redhat)

`pump` est simple et courament utilisé. `dhcp3-client` est complexe mais plus configurable. [8]

10.6 Configuration réseau haut niveau avec Debian

Afin de permettre la configuration réseau plus facile Debian fournit un outil de configuration haut niveau standard qui comprend les commandes `ifup` et `ifdown` et le fichier `/etc`

---

[8] À la date d'octobre 2003, il y a aussi un paquet `dhcp-client`. Il contient la version 2 du client DHCP d'ISC, qui a été remplacée par la version 3 qui vient dans le paquet `dhcp3-client`. Les responsables ont prévu de renommer `dhcp3-client` en `dhcp-client` un jour mais il semble que cela n'arrivera pas pour Sarge.

`/network/interfaces`.[9] Si vous choisissez d'utiliser `ifupdown` pour configurer votre réseau alors vous ne devriez pas utiliser les commandes bas niveau aussi.[10] `Ifupdown` est écrit avec l'hypothèse qu'il sera seul utilisé pour configurer et déconfigurer les interfaces réseau.

Pour mettre à jour la configuration des interfaces, faites ceci :

```
# ifdown eth0
# editor /etc/network/interfaces  # modifiez à votre convenance
# ifup eth0
```

Pour plus d'information voir `interfaces(5)`, `/usr/share/doc/ifupdown/examples/network-interfaces.gz` et `ifup(8)`.

10.6.1 Configurer une interface avec une adresse IP fixe

Supposons que vous voulez configurer une interface Ethernet de façon à avoir l'adresse IP fixe `192.168.0.123`. Cette adresse commence par `192.168.0` donc elle doit être sur un réseau local. Supposons aussi que `192.168.0.1` est l'adresse de la passerelle du réseau local vers l'Internet. Éditez `/etc/network/interfaces` et ajoutez une partie comme suit :

```
iface eth0 inet static
        address 192.168.0.123
        netmask 255.255.255.0
        gateway 192.168.0.1
```

Si vous avez le paquet `resolvconf` installé alors vous pouvez ajouter ces lignes pour spécifier l'information DNS. Par exemple :

```
iface eth0 inet static
        address 192.168.0.123
        netmask 255.255.255.0
        gateway 192.168.0.1
        dns-search nicedomain.org
        dns-nameservers 195.238.2.21 195.238.2.22
```

Après que l'interface ait été activée, les arguments des options `dns-search` et `dns-nameservers` sont disponibles à `resolvconf` pour inclusion dans `resolv.conf`. L'argument `nicedomain.org` de l'option `dns-search` correspond à l'argument d'une option `search` de `resolv.conf(5)`. Les arguments `195.238.2.21` et `195.238.2.22` de l'option `dns-nameservers` correspondent aux arguments des options `nameserver` de `resolv.conf(5)`. D'autres options sont `dns-domain` et `dns-sortlist`. Voir 'Gérer l'information sur les serveurs de noms – `resolvconf`' page 177.

---

[9] Le format du fichier `/etc/network/interfaces` de la version actuelle de `ifupdown` est légèrement incompatible avec le format de fichier des versions du paquet livrées avec Potato. Le script de post-installation de `ifupdown` devrait mettre à jour le fichier automatiquement si nécessaire.

[10] Cela signifie aussi que vous ne devriez pas utiliser d'autre outil de configuration haut niveau comme `whereami` qui appelle les outils de configuration bas niveau.

10.6.2 Configurer une interface par DHCP

Pour configurer une interface par DHCP, éditez le fichier `/etc/network/interfaces` afin d'y inclure :

```
iface eth0 inet dhcp
```

Afin que cela fonctionne vous devez avoir installé l'un des clients DHCP mentionnés dans 'Configurer les interfaces réseau par DHCP' page 178.

10.6.3 Configurer une interface Wi-Fi

Le paquet `wireless-tools` inclut un script `/etc/network/if-pre-up.d/wireless-tools` qui permet de configurer du matériel Wi-Fi (802.11a/b/g) avant qu'une interface soit activée. La configuration se fait à l'aide du programme `iwconfig`; voir `iwconfig(8)`. Pour chaque paramètre de la commande `iwconfig` vous pouvez inclure une option dans `/etc/network/interfaces` avec le nom du paramètre préfixé par « wireless- ». Par exemple, pour configurer l'ESSID de `eth0` en `myessid` et la clé de chiffrage à `123456789e` avant d'activer `eth0` par DHCP, éditez `/etc/network/interfaces` comme suit :

```
iface eth0 inet dhcp
        wireless-essid myessid
        wireless-key 123456789e
```

10.6.4 Configurer une interface PPP

Les programmes `ifup` et `ifdown` utilisent `pon` et `poff` pour activer ou désactiver des interfaces PPP ; lisez-donc d'abord 'Configurer une interface PPP' page 171.

Supposons que vous avez fait la configuration pour utiliser le fournisseur `myisp`. Éditez `/etc/network/interfaces` pour y inclure :

```
iface ppp0 inet ppp
provider myisp
```

Avec ces lignes, `ifup ppp0` exécute

```
pon myisp
```

Malheureusement, il n'est pour le moment pas possible de fournir d'autres options à `pppd` dans une section `ppp` de `/etc/network/interfaces`.[11]

---

[11] Voir le bogue #196877 (`http://bugs.debian.org/196877`).

Il n'est pour le moment pas possible d'utiliser `ifupdown` pour effectuer la configuration post-connexion des interfaces PPP. Comme `pon` se termine avant que `pppd` ait terminé de créer la connexion, `ifup` exécute les scripts `up` avant que l'interface soit activée. Jusqu'à ce que ce bogue [12] soit corrigé il est nécessaire de faire la configuration post-connexion dans `/etc/ppp/ip-up` ou `/etc/ppp/ip-up.d/`.

10.6.5 Configurer une interface PPPoE

Beaucoup de Fournisseurs d'Accès Internet (FAIs) utilisent PPP pour négocier les connexions bien que leurs clients soient connectés par des réseaux Ethernet et/ou ATM. Cela se fait à l'aide de PPP sur Ethernet (PPPoE) qui est une technique d'encapsulation des flux PPP dans des trames Ethernet. Supposons que votre FAI soit *`myisp`*. D'abord, configurez PPP et PPPoE pour le fournisseur *`myisp`*. Le moyen le plus facile de le faire est d'installer le paquet `pppoeconf` et de lancer `pppoeconf` depuis une console. Ensuite éditez `/etc/network/interfaces` pour y inclure :

```
iface eth0 inet ppp
        provider myisp
```

Il y a quelques fois des problèmes d'Unité Maximum de Transmission (Maximum Transmit Unit – MTU) avec PPPoE sur des lignes DSL (Digital Subscriber Lines). Voir DSL-HOWTO (`http://www.tldp.org/HOWTO/DSL-HOWTO/`) pour plus de détails.

Notez que si votre modem haut-débit contient un routeur alors celui-ci prend en charge la connexion PPPoE lui-même et apparaît sur le LAN comme une simple passerelle Ethernet vers l'Internet.

10.6.6 Configurer plusieurs interfaces Ethernet pour une passerelle

Supposons que `eth0` soit connectée à l'Internet avec une adresse IP obtenue par DHCP et `eth1` soit connectée au LAN avec l'adresse IP fixe `192.168.1.1`. Éditez `/etc/network/interfaces` pour y inclure :

```
iface eth0 inet dhcp

iface eth1 inet static
        address 192.168.1.1
        netmask 255.255.255.0
```

Si vous activez NAT sur cette machine comme décrit dans 'Configurer une passerelle' page 190 alors vous pourrez partager la connexion Internet avec toute les machines du LAN.

---

[12] Voir le bogue #127786 (`http://bugs.debian.org/127786`).

10.6.7 Configurer des interfaces virtuelles

Avec des interfaces virtuelles, vous pouvez configurer une seule carte Ethernet pour qu'elle soit connectée à plusieurs sous-réseaux IP. Par exemple, supposons que votre machine est sur le réseau local 192.168.0.x/24. Vous voulez vous connecter à l'Internet en utilisant une adresse IP publique obtenue par DHCP en utilisant cette même carte Ethernet. Éditez `/etc/network/interfaces` pour y inclure :

```
iface eth0 inet static
        address 192.168.0.1
        netmask 255.255.255.0
        network 192.168.0.0
        broadcast 192.168.0.255

iface eth0:0 inet dhcp
```

L'interface `eth0:0` est virtuelle. Lorsqu'elle est activée, son parent `eth0` l'est aussi.

10.7 Reconfiguration du réseau

Dans ce qui suit, il est important de comprendre la différence entre une interface **physique** et une interface **logique**. [13] Une interface physique est ce que l'on a appelé précédemment « l'interface », qui s'appelle `eth0`, `ppp1`, etc. Une interface logique est un ensemble de valeurs qui peuvent être assignées aux paramètres variables d'une interface physique. Si vous trouvez cela confus, remplacez l'expression « configuré comme interface logique `X` » par « configuré avec le profil d'interface `X` » en lisant.

Les définitions `iface` de `/etc/network/interfaces` sont en fait des définitions d'interfaces logiques, pas physiques. [14] Si vous ne voulez jamais reconfigurer vos interfaces alors vous pouvez ignorer cela puisque l'interface physique `X` sera par défaut configurée comme interface logique `X`.

Cependant, supposons que votre ordinateur est un portable que vous transportez entre la maison et le travail. Lorsque vous connectez votre ordinateur au réseau de l'entreprise ou votre réseau personnel, vous devez configurer `eth0` en conséquence.

Définissez d'abord deux interfaces logiques `home` et `work` (au lieu de `eth0` comme on l'a fait auparavant) qui décrit comment l'interface doit être configurée pour le réseau personnel et le réseau de l'entreprise, respectivement.

```
iface home inet static
        address 192.168.0.123
```

---

[13] Cette terminologie est utilisée dans la documentation de `ifupdown`.

[14] Notez que les interfaces des lignes `auto` sont physiques, pas logiques.

```
        netmask 255.255.255.0
        gateway 192.168.0.1

iface work inet static
        address 81.201.3.123
        netmask 255.255.0.0
        gateway 81.201.1.1
```

Ensuite l'interface physique `eth0` peut être activée pour le réseau personnel avec la configuration appropriée en le spécifiant sur la ligne de commande :

```
# ifup eth0=home
```

Pour reconfigurer `eth0` pour le réseau de l'entreprise, exécutez les commandes :

```
# ifdown eth0
# ifup eth0=work
```

Notez qu'avec le fichier `interfaces` ci-dessus il n'est plus possible d'activer `eth0` par `ifup eth0` tout seul. La raison est que `ifup` utilise le nom de l'interface physique comme défaut pour le nom de l'interface logique et dans notre exemple aucune interface logique `eth0` n'est définie.

10.8 Reconfiguration magique du réseau

Les noms d'interfaces logiques peuvent être transformés en autres noms d'interfaces logiques lorsque `ifup` est exécuté. La façon dont les noms sont transformés dépend des circonstances. Ainsi, `ifup` peut être configuré pour activer une interface physique avec la bonne interface logique parmi un ensemble d'alternatives prédéterminées.

La transformation de noms d'interfaces logiques apparaît dans les cas suivants :

– Si aucun nom d'interface logique n'est donné sur la ligne de commande de `ifup` alors le nom de l'interface physique est utilisé comme nom d'interface logique.
– Si le nom d'interface logique correspond au motif d'une section `mapping` alors la transformation est appliquée pour générer un nouveau nom d'interface logique. Cela est fait récursivement jusqu'à ce qu'il n'y ait plus de correspondance.
– Si le nom de la dernière interface logique correspond à la définition d'une interface logique dans `/etc/network/interfaces` alors l'interface physique est activée comme cette interface logique. Sinon `ifup` affiche un message « Ignoring unknown interface » et s'arrête.

La syntaxe de la section `mapping` est :

```
mapping glob-pattern
        script script-name
        [map script input]
```

Le script appelé dans la section `mapping` est exécuté avec le nom de l'interface physique comme argument et avec le contenu de toutes les lignes `map` suivantes de la section. Le script affiche les résultats de la transformation sur la sortie standard avant de s'arrêter.

Par exemple, la section `mapping` suivante fera que `ifup` activera l'interface `eth0` comme interface logique `home`.

```
mapping eth0
        script /usr/local/sbin/echo-home
```

where `/usr/local/sbin/echo-home` is :

```
#!/bin/sh
echo home
```

Parce que la transformation est faite par un script, il est possible de sélectionner l'interface logique par un test. Commençons par un exemple. Supposons que vous avez différentes cartes réseau pour le travail et pour la maison. Le répertoire `/usr/share/doc/ifupdown /examples/` contient un script de transformation qui peut être utilisé pour sélectionner une interface logique en se basant sur l'adresse MAC (Media Access Controller). Installez d'abord le script dans un répertoire approprié.

```
# install -m770 /usr/share/doc/ifupdown/examples/get-mac-address.sh \
   /usr/local/sbin/
```

Ensuite ajoutez une section comme celle qui suit à `/etc/network/interfaces` :

```
mapping eth0
        script /usr/local/sbin/get-mac-address.sh
        map 02:23:45:3C:45:3C home
        map 00:A3:03:63:26:93 work
```

Des programmes plus sophistiqués de transformation sont décrits ci-dessous dans 'guessnet' de la présente page, et 'laptop-net' page suivante.

10.8.1 guessnet

Installez `guessnet` et ajoutez une section comme suit à `/etc/network/interfaces` :

```
mapping eth0
        script guessnet-ifupdown
        map home
        map work
```

Maintenant lorsque vous exécutez `ifup eth0`, `guessnet` vérifiera s'il faut activer `home` ou `work`. Pour faire cela, il utilise l'information des définitions d'interfaces logiques.

10.8.2 laptop-net

Le paquet `laptop-net` a une approche différente de la reconfiguration du réseau *automagique*. Laptop-net n'utilise pas les interfaces logiques de `ifupdown` mais ses propres schémas de configuration et profils système. Laptop-net utilise néanmoins `ifupdown` pour configurer les interfaces physiques. Pour plus d'informations, consultez sa documentation `laptop-net-doc`.

10.9 Activer la reconfiguration du réseau

Nous avons vu comment les interfaces peuvent être reconfigurées. La reconfiguration doit être effectuée au moment approprié.

Traditionnellement, le réseau était configuré pendant la séquence de démarrage par le fichier `/etc/rcS.d/S40networking` et était rarement reconfiguré. Les services qui dépendaient du réseau étaient démarrés plus tard dans la séquence de démarrage. À l'arrêt ou au redémarrage, les scripts de démarrage étaient exécutés dans l'ordre inverse.

Actuellement, cependant, il y a une tendance dans GNU et Linux à supporter du matériel et des environnements qui changent dynamiquement. D'abord, le support pour l'insertion de cartes PCMCIA à chaud a été ajouté ; plus récemment, le mécanisme `hotplug` a été ajouté afin que beaucoup plus de périphériques puissent être ajoutés pendant que l'ordinateur tourne. Cela inclut du matériel réseau. Notez que les services qui dépendent sur du matériel qui est installé à chaud doivent seulement être démarrés après que le matériel ait été ajouté et doivent être arrêtés lorsque le matériel est enlevé. Cela signifie que ces services doievnt être enlevés du contrôle du démarrage System V et mis sous contrôle du paquet `ifupdown` à la place.

Par exemple, supposons que le service `foo` contrôlé par le script de démarrage `/etc/init.d/foo` dépend de l'interface réseau `eth0` reconfigurée dynamiquement.

- D'abord supprimez `foo` du contrôle du système de démarrage :

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

- Ensuite mettez `foo` sous contrôle de `ifupdown` en ajoutant les options `up` et `down` avec des appels au script de démarrage de `foo` à la section `eth0` de `/etc/network/interfaces`

  ```
  iface eth0 inet dhcp
          up /etc/init.d/foo start
          down /etc/init.d/foo stop
  ```

10.9.1 Activer la configuration réseau au démarrage

Au démarrage le script `/etc/rcS.d/S40networking` exécute la commande `ifup -a`. Cela active toutes les interfaces physiques listées dans les sections `auto` de `/etc/network/interfaces`.

Aujourd'hui, il est souvent mieux de gérer la configuration réseau par des méthodes dynamiques. Une fois que les mécanismes pour supporter du matériel qui change dynamiquement sont en place, il devient plus simple de traiter le matériel fixe comme s'il était dynamique aussi. Le démarrage peut être traité comme un simple événement *hotplug*. (Voir 'Activer la configuration réseau – `hotplug`' de la présente page.)

Cependant, dans presque tous les cas, on veut au moins l'interface `lo` au démarrage. Ainsi, assurez-vous que `/etc/network/interfaces` inclut les lignes suivantes.

```
auto lo

iface lo inet loopback
```

Vous pouvez ajouter d'autres interfaces physiques dans des sections `auto` si vous voulez qu'elles soient activées au démarrage aussi. N'ajoutez **jamais** d'interface PCMCIA dans des sections `auto`. `cardmgr` est démarré après `/etc/rcS.d/S40networking` dans la séquence de démarrage.

10.9.2 Activer la configuration réseau – `hotplug`

Pour le support de *hotplug* installez le paquet `hotplug`.

Le matériel réseau peut être installé à chaud soit au démarrage soit après qu'une carte (par exemple PCMCIA) a été insérée dans la machine, ou après qu'un utilitaire comme `discover` est exécuté et charge les modules nécessaires.

Lorsque le noyau détecte un nouveau matériel il initialise le pilote pour ce matériel et exécute ensuite le programme `hotplug` pour le configurer. Ensuite si le matériel est enlevé alors le noyau exécute `hotplug` de nouveau avec des paramètres différents. Dans Debian, lorsque `hotplug` est appelé il exécute les scripts de `/etc/hotplug/` et `/etc/hotplug.d/`. Voir `hotplug(8)` pour plus de détails.

Le matériel réseau qui vient d'être inséré est configuré par le script `/etc/hotplug/net.agent`. [15] Supposons que votre carte réseau PCMCIA ait été insérée, ce qui rend l'interface `eth0` disponible pour utilisation. `/etc/hotplug/net.agent` exécute les actions suivantes :

```
ifup eth0=hotplug
```

À moins d'avoir ajouté une interface logique appelée `hotplug` à `/etc/network/interfaces`, cette commande ne fera rien. Pour que cette commande configure `eth0`, ajoutez la section suivante dans `/etc/network/interfaces` :

```
mapping hotplug
        script echo
```

---

[15] Il peut aussi être configuré par un script installé dans `/etc/hotplug.d/net/`.

Comme expliqué dans ‘**Re**configuration du réseau’ page 182 cela va transformer la commande ci-dessus pour être équivalente à :

```
ifup eth0=eth0
```

(**N’**incluez **pas** de section de ce type si vous utilisez des instances de `ifplugd` démarrées par `hotplug` pour contrôler l’interface, comme décrit dans ‘Activer la configuration réseau – `ifplugd`’ de la présente page.)

Si vous voulez que seulement `eth0` soit activée par *hotplug*, et pas d’autres interfaces, alors utilisez `grep` au lieu de `echo` :

```
mapping hotplug
        script grep
        map eth0
```

Voir ‘Reconfiguration magique du réseau’ page 183 et `/usr/share/doc/hotplug/README.Debian` pour plus d’astuces.

10.9.3 Activer la configuration réseau – `ifplugd`

`ifplugd` active ou désactive une interface selon que le matériel sous-jacent est ou n’est pas connecté à un réseau. Le programme peut détecter un câble connecté à une interface Ethernet ou un point d’accès Wi-Fi. Lorsque `ifplugd` voit que l’état d’un lien a changé il lance un script qui par défaut exécute `ifup` ou `ifdown` pour l’interface.

`ifplugd` fonctionne bien avec `hotplug`. Lorsqu’une carte est insérée, ce qui rend une interface disponible, `/etc/hotplug.d/net/ifplugd.hotplug` exécute une instance de `ifplugd` pour cette interface. Lorsque `ifplugd` détecte que la carte est connectée à un réseau, il exécute `ifup` pour cette interface.

10.9.4 Activer la configuration réseau – `waproamd`

Afin de s’associer à un point d’accès, une carte Wi-Fi peut devoir être programmée avec une clé de chiffrement WEP. Si vous utilisez `ifplugd` pour contrôler `ifup` comme décrit dans ‘Activer la configuration réseau – `ifplugd`’ de la présente page alors vous ne pouvez évidemment pas configurer la clé de chiffrement avec `ifup` parce que `ifup` n’est appelé qu’après que la carte ait été associée au point d’accès. Une solution possible est de programmer toutes les clés requises dans la mémoire non volatile de la carte Wi-Fi. Si vous vous déplacez sur de nombreux réseaux, votre carte Wi-Fi peut ne pas pouvoir stocker assez de clés.

Une autre solution est d’utiliser `waproamd` qui configure la clé de chiffrement WEP selon les points d’accès disponibles révélés par une recherche.

`waproamd` fonctionne bien avec `hotplug`. Lorsqu'une carte est insérée, ce qui rend une interface disponible, `/etc/hotplug.d/net/waproamd.hotplug` exécute une instance de `waproamd` pour cette interface. `waproamd` fonctionne bien avec `ifplugd`.

Voir le README du paquet `waproamd` pour plus d'information.

10.9.5 Configuration réseau et PCMCIA

La configuration des cartes réseau PCMCIA peut se faire de différentes façons.

- Cartes PCMCIA PCI 32 bits (CardBus) :
  - `ifupdown` sous le contrôle de `hotplug`
- Cartes réseau PCMCIA ISA 16 bits :
  - `ifupdown` sous le contrôle de `hotplug` avec `pcmcia-cs` désactivé en ajoutant `exit 0` au début du ficher `/etc/pcmcia/network` (recommandé)
  - `ifupdown` sous le contrôle de `pcmcia-cs` avec `/etc/pcmcia/network` par défaut (obsolète)
  - outils de configuration réseau bas niveau sous le contrôle de `pcmcia-cs` modifié en activant des possibilités de `/etc/pcmcia/network` (obsolète)

L'approche recommandée pour les cartes 16 bits prend avantage du fait que le sous-système *hotplug* de Linux 2.4 supporte PCMCIA. Suivez simplement les instructions de 'Activer la configuration réseau – `hotplug`' page 186. [16] Afin que le paquet `pcmcia-cs` n'interfère pas avec le paquet `hotplug`, vous devez le modifier en ajoutant la ligne :

```
exit 0
```

au début de `/etc/pcmcia/network` pour le désactiver.

Notez qu'il n'y a rien de mauvais à exécuter `cardmgr`. On en veut simplement pas qu'il appelle les programmes de configuration réseau.

Afin que `cardmgr` fonctionne correctement vous devrez peut-être éditer `/etc/pcmcia/config.opts` afin de configurer les ressources assignées aux cartes PCMCIA 16 bits. Voir 'PCMCIA' page 88 et le Linux PCMCIA HOWTO (`http://www.tldp.org/HOWTO/PCMCIA-HOWTO.html`) pour plus d'information.

---

[16] Dans de plus anciennes versions de Debian le moyen standard de configurer les cartes réseau PCMCIA utilisait `cardmgr` avec les scripts `/etc/pcmcia/network` et `/etc/pcmcia/network.opts`. Ces scripts ont été développés avant que Linux n'ait *hotplug*. Certains utilisent toujours ces scripts tels quels en appelant simplement `ifup` après que l'interface ait été ajoutée et `ifdown` lorsqu'elle est enlevée. Comme indiqué ci-dessus, il est maintenant recommandé d'utiliser `hotplug` pour faire cela. D'autres appellent directement des commandes de configuration réseau bas-niveau activées lorsque certaines variables de `/etc/pcmcia/network.opts` sont mises à `y`. Ce système a plusieurs inconvénients. Il est affecté par des conditions de concurrence ; il ne fonctionne qu'avec les cartes PCMCIA 16 bits ; il duplique ce que `ifupdown` fait déjà. Il est donc obsolète.

10.10 Configuration des services réseau

La configuration des services réseau sur une machine de bureau ou un serveur personnel inclut :

- Le *super-serveur* Internet et l'enveloppe de démons TCP/IP, voir 'Contrôle d'accès aux services' page 126.
  - `/etc/inetd.conf`
- `ssh` : shell sécurisé OpenSSH, voir 'SSH' page 146.
  - `/etc/ssh/ssh_config`
  - `/etc/ssh/sshd_config`
- `exim` : agent de transport de courriel, voir 'Nom de courriel' page 175 et 'Agent de transport de courrier électronique' page 150.
  - `/etc/exim/exim.conf`
  - `/etc/mailname`
  - `/etc/aliases`
  - `/etc/email-addresses`
- `fetchmail` : démon pour récupérer le courriel depuis un compte POP3, voir 'Récupérer le courrier électronique – Fetchmail' page 153.
  - `/etc/fetchmailrc`
- `procmail` : livraison locale de courriel et filtre, voir 'Traiter le courrier électronique – Procmail' page 154.
  - `~/.procmailrc`
- Nom de machine et DNS (proxy, cache, ...), voir 'Nom de machine' page 175 et 'Domain Name Service (DNS)' page 176.
  - `/etc/host.conf`
  - `/etc/hostname`
  - `/etc/hosts`
  - `/etc/hosts.allow`
  - `/etc/hosts.deny`
  - `/etc/resolv.conf`
  - `/etc/bind/named.conf` (à éditer)
  - `/etc/bind/db.lan` (à ajouter pour les machines du réseau local)
  - `/etc/bind/db.`*`192.168.0`* (à ajouter pour le DNS inversé du réseau local)
- DHCP, voir 'Configurer les interfaces réseau par DHCP' page 178.
  - `/etc/dhcp3/dhclient.conf` (sur le client DHCP)
  - `/etc/default/dhcp3-server` (sur le serveur DHCP)
  - `/etc/dhcp3/dhcpd.conf` (sur le serveur DHCP)
- `cvs` : Concurrent Version System, voir 'Concurrent Versions System (CVS)' page 203.
  - `/etc/cvs-cron.conf`
  - `/etc/cvs-pserver.conf`
- `nfs-kernel-server` : Network File System, voir 'Configuration NFS' page 38. (pour systèmes Unix)
  - `/etc/exports`
- `samba` : partage de fichiers et d'imprimantes surr le réseau pour Windows, voir 'Configuration Samba' page 38 et 'Samba' page 119.

  - `/etc/samba/smb.conf`
- Démon pour l'impression, voir 'Configuration de l'imprimante' page 39.
  - `/etc/printcap` (pour lpr)
- `apache` et `apache2` : serveur web.
  - `/etc/apache/*`
  - `/etc/apache2/*`
- `squid` : cache web.
  - `/etc/squid/*`

10.11 Résolution des problèmes réseau

Si vous rencontrez des problèmes, alors consultez la sortie des programmes suivants en premier :

```
# ifconfig
# cat /proc/pci
# cat /proc/interrupts
# dmesg | more
```

Consultez aussi les sections suivant 'Bases pour tester un réseau' page 115.

Si vous avez des problèmes avec certains sites web, consultez 'Étranges problèmes d'accès à certains sites web' page 42.

10.12 Configurer une passerelle

Une machine Debian peut être une passerelle multi-possibilités, qui fait Translation d'Adresse Réseau (NAT, aussi connu sous le nom de masquerading), transfert de courriel, DHCP, cache DNS, cache HTTP, serveur CVS, serveur NFS, serveur Samba. Voir 'Nom d'hôte et IP à utiliser pour le réseau local' page 29 pour un exemple de configuration.

10.12.1 Configuration de Netfilter

Le projet netfilter/iptables est un sous-système de pare-feu pour Linux 2.4 et supérieur. Voir Netfilter (`http://www.netfilter.org/`), où de nombreux problèmes de configuration réseau sont expliqués.

Bases de Netfilter

Le traitement des paquets par Netfilter utilise cinq chaînes par défaut : PREROUTING, INPUT, FORWARD, OUTPUT, et POSTROUTING.

```
                routing
                decision
IN ------> PRE ---> ------> FORWARD -----> ----> POST -----> OUT
interface  ROUTING  \       filter       /       ROUTING     interface
           DNAT     |       tracking     ^       SNAT
           REDIRECT |                    |      MASQUERADE
                    v                    |
                  INPUT                OUTPUT
                    | filter             ^ filter,DNAT
                    v                    |
                    \--> Local Process --/
                            user-space programs
```

Table Netfilter

Les paquets sont traités pour chaque chaîne par défaut selon la table suivante.

- filtre (filtre de paquets, défaut)
  - INPUT (pour les paquets venant vers la machine)
  - FORWARD (pour les paquets routés par la machine)
  - OUTPUT (pour les paquets générés localement)
- nat (network address translation, translation d'adresse réseau)
  - PREROUTING (pour modifier les paquets dès leur entrée)
  - OUTPUT (pour modifier les paquets générés localement avant le routage)
  - POSTROUTING (pour modifier les paquets à leur sortie)
- troncage (troncage d'adresse réseau, seulement après le noyau 2.4.18)
  - les cinq chaînes par défaut

Cibles Netfilter

Les règles de pare-feu ont plusieurs cibles :

- quatre cibles de base :
  - ACCEPT signifie qu'on laisse le paquet passer.
  - DROP signifie qu'on laisse tomber le paquet.
  - QUEUE signifie qu'on passe le paquet en espace utilisateur (si cela est supporté par le noyau).
  - RETURN signifie qu'on arrête le passage dans la chaîne et qu'on continue à la règle suivante de la chaîne (appelante) précédente.
- cibles étendues :
  - LOG active l'enregistrement dans le journal.
  - REJECT renvoie un paquet d'erreur et laisse tomber le paquet.
  - SNAT modifie l'adresse source du paquet et est utilisé seulement dans la chaîne POSTROUTING. (table nat seulement)

    ```
    --to-source  ipaddr[-ipaddr][:port-port]
    ```
  - MASQUERADE est la même chose que SNAT mais pour des adresses IP assignées dynamiquement (connexion téléphonique). (table nat seulement)

```
--to-ports port[-port]
```

- DNAT modifie l'adresse de destination du paquet et est utilisé dans les chaînes PREROUTING et OUTPUT, et les chaînes définies par l'utilisateur qui sont appelées seulement depuis ces chaînes. (table nat seulement)

```
--to-destination ipaddr[-ipaddr][:port-port]
```

- REDIRECT modifie l'adresse IP de destination pour envoyer le paquet vers la machine elle-même.

```
--to-ports port[-port]
```

Commandes Netfilter

Les commandes de base de `iptables` sont :

```
iptables -N chaine                    # crée une chaine

iptables -A chaine \                  # ajoute une règle à  chaine
         -t table \                   # utilise table
                                        (filtre, nat, troncage)
         -p protocol \                # tcp, udp, icmp, ou tous
         -s source-address[/mask] \
         --sport port[:port] \        # port source si -p est tcp ou udp
         -d destination-address[/mask] \
         --dport port[:port] \        # port destination si
                                        -p est tcp ou udp
         -j target \                  # que faire si cela correspond
         -i in-interface-name \       # pour INPUT,  FORWARD, PREROUTING
         -o out-interface-name        # pour FORWARD, OUTPUT, POSTROUTING
```

Translation d'adresse réseau (NAT)

Les machines du LAN peuvent accéder à l'Internet à travers une passerelle qui traduit les adresses IP du LAN en des adresses IP utilisables sur l'Internet.

```
# apt-get install ipmasq
```

Appliquez les règles de l'exemple pour fortifier la protection de `ipmasq`. Consultez `/usr/share/doc/ipmasq/examples/stronger/README`. Pour l'image du noyau Debian kernel-image-2.4 de Woody, assurez-vous de charger les modules appropriés. La version de Sarge corrige ceci. Consultez 'Fonctions réseau' page 88 pour les instructions de configuration.

Pour l'image du noyau Debian 2.2, éditez `Z92timeouts.rul` dans `/etc/masq/rules` comme suit pour assurer une connexion plus longue aux sites distants (bon pour de gros emails, etc.) :

```
# tcp, tcp-fin, udp
# 2hr, 10 sec, 160 sec - default
# 1 day, 10 min, 10 min - longer example
$IPCHAINS -M -S 86400 600 600
```

De plus, si le réseau est accédé par une carte PCMCIA, `ipmasq` a besoin d'être démarré soit depuis `/etc/pcmcia/network.opts` (lire `/usr/share/doc/ipmasq/ipmasq.txt.gz`) soit depuis `/etc/network/interfaces` (lire 'Configuration réseau et PCMCIA' page 188 et 'Activer la reconfiguration du réseau' page 185).

Rediriger une connexion SMTP (2.4)

Supposons que vous avez un PC portable qui est configuré pour utiliser un autre environnement réseau et que vous voulez utiliser votre logiciel de courrier électronique sans le reconfigurer.

En ajoutant les règles suivantes avec la commande `iptables` sur la passerelle, la connexion SMTP sera redirigée vers la passerelle.

```
# iptables -t nat -A PREROUTING -s 192.168.1.0/24 -j REDIRECT \
           -p tcp --dport smtp --to-port 25 # smtp=25,
                                              INPUT est ouvert
```

Pour une règle de redirection plus complexe, installez le paquet `ipmasq` et ajoutez M30redirect.def (`http://www.debian.org/doc/manuals/debian-reference/examples/`) au répertoire `/etc/ipmasq/rules/`.

10.12.2 Gérer plusieurs connexions à l'Internet

[FIXME] Politique de routage (par Phil Brutsche pbrutsch@tux.creighton.edu) : Voir le Manuel d'iproute (`http://ds9a.nl/2.4Routing/`) pour les détails. Le contrôle de traffic peut aussi être intéressant.

Environnement :

```
eth0: 192.168.1.2/24; gateway 192.168.1.1
eth1: 10.0.0.2/24; gateway 10.0.0.1
Pas de masquerading sur cette machine.
```

Un peu de magie :

1 ip rule add from 192.168.1.2 lookup 1
2 ip rule add from 10.0.0.2 lookup 2
3 ip route add to default via 10.0.0.1 metric 0
4 ip route add to default via 192.168.1.1 metric 1

```
ip route add table 1 to 192.168.1.0/24 via eth0
ip route add table 1 to 10.0.0.2/24 via eth1
ip route add table 1 to default via 192.168.1.1
ip route add table 2 to 192.168.1.0/24 via eth0
ip route add table 2 to 10.0.0.2/24 via eth1
ip route add table 2 to default via 10.0.0.2
```

[FIXME] Je n’ai jamais fait cela. Comment configurer une connexion RTC en cas de défaillance d’une connexion rapide ? Envoyez-moi une rustine pour cela s’il-vous-plaît :)

Chapitre 11

Editeurs

11.1 Editeurs populaires

Linux offre plusieurs solutions comme éditeurs en mode texte. Parmi eux :
- `vim` : Éditeur puissant et léger hérité de BSD. VI iMproved (VI aMélioré).
- `emacs` : L'éditeur ultime et lourd hérité de GNU. L'original de RMS (Richard M. Stallman).
- `xemacs` : Emacs : The Next Generation (la nouvelle génération), à l'origine par Lucid.
- `mcedit` : Éditeur GNU pour débutant. Identique à l'éditeur interne de `mc`. Voir 'Éditeur de MC' page 50.
- `ae` : Petit éditeur par défaut (Potato). À éviter.
- `nano` : Petit éditeur GNU par défaut (Woody). Emule `pico`.
- `joe` : Pour les vieux habitués de WordStar ou de TurboPascal.
- `jed` : Éditeur rapide et complet avec menus et raccourcis Emacs.
- `jove` : Très petit éditeur avec raccourcis Emacs.
- `nvi` : Nouveau vi. Compatible bogue pour bogue avec le vi original.

Utilisez `update-alternatives --config editor` pour changer d'éditeur par défaut. De plus, beaucoup de programmes utilisent les variables d'environnement `EDITOR` ou `VISUAL` pour décider quel éditeur utiliser. Voir 'Éditeur de MC' page 50.

Il existe aussi quelques éditeurs de texte pour X :
- `gvim` : Vim avec une interface graphique (paquets `vim` et `vim-gtk`)
- `emacs` : Le vrai Emacs (détecte X automatiquement)
- `xemacs` : L'Emacs nouvelle génération (détecte X automatiquement)

Ces commandes xclient prennent des options standard comme `-fn` *`a24`* qui rendent les choses faciles pour les plus vieux comme moi :-) Voir 'Clients X' page 135.

11.2 Editeurs de sauvetage

Il y a quelques éditeurs dans `/bin`. L'un d'eux doit être installé pour faciliter l'édition de fichiers quand `/usr` n'est pas accessible.

- `elvis-tiny` : Editeur minimum type vi (`vi` pour le démarrer)
- `nano-tiny` : Editeur minimum non vi (`nano-tiny` pour le démarrer)
- `ed` : Editeur minimum (toujours là mais difficile à utiliser)

11.3 Emacs et Vim

11.3.1 Conseils pour Vim

Lisez la page « VIM - main help file » en tapant <F1> dans le programme.

```
<F1>     Aide
<esc>    Retour au mode normal
V              Mode visuel
i              Mode insertion
:              Commandes en ligne
:set tw=72     Paramètre la largeur du texte à 72
<F11>    Mode Insertion (collage)
:r! date -R    Insertion de la date selon RFC-822
qa  Enregistre la frappe dans le registre a
q              Arrête l'enregistrement de la frappe
@a  Exécute la frappe enregistrée dans le registre a
:edit foo.txt Edite un autre fichier en chargeant
foo.txt
:wnext         Enregistre le fichier courant et édite le fichier suivant
```

`q` et `@` peuvent être utilisés pour enregistrer et exécuter de simples macros. Par exemple, pour créer une macro qui insère des balises HTML pour l'italique autour du mot sur lequel se trouve le curseur, vous pourriez taper `qii<i>^[ea</i>^[q` (où `^[` est la touche ESC). Taper `@i` au début d'un mot ajoute les balises <i> et </i>.

Voir aussi 'Utiliser GnuPG avec Vim' page 231.

11.3.2 Conseils pour Emacs

```
<F1>          Aide
<F10>         Menu
C-u M-! date -R     Insertion de la date selon RFC-822
```

11.3.3 Démarrer l'éditeur

```
démarrer l'éditeur :            emacs nom_de_fichier    vim nom_de_fichier
démarrage en compatibilité vi :                         vim -C
démarrage en non-compatible vi :                        vim -N
démarrage en mode par défaut :  emacs -q                vim -N -u NONE
```

11.3.4 Résumé des commandes de l'éditeur (Emacs, Vim)

```
Sortie :                    C-x C-c         :qa /:wq /:xa /:q!
Retour/mode commande :    C-g             <esc>
Retour(gauche) :          C-b             h
Après(droite) :           C-f             l
Suivant(en bas) :         C-n             j
Précédent(en haut) :      C-p             k
Début de ligne(^) :       C-a             0
Fin de ligne($) :         C-e             $
Commandes mUltiples :     C-u nnn cmd     :count cmd
Commandes Multiples :     M-digitkey cmd
Sauvegarder le fichier :  C-x C-s         :w file
Début du tampon :         M-<          1G
Fin du tampon :           M->           G
Écran suivant :           C-v             ^F
1/2 écran suivant :                       ^D
Ligne suivante :                          ^E
Revenir d'1 écran :       M-v             ^B
Revenir d'1/2 écran :                     ^U
Ligne précédente :                        ^Y
Avance l'autre fenêtre :  M-C-v
Effacer sous le curseur : C-d             x
Effacer à partir du curseur
jusqu'à la fin de ligne : C-k             D
iRecherche en avant :     C-s
iRecherche en arrière :   C-r
Chercher en avant :       C-s enter       /
Chercher en arrière :     C-r enter       ?
iRecherche avec
expression rationnelle :  M-C-s
iRecherche en arrière avec
expression rationnelle :  M-C-r
Chercher avec
expression rationnelle :  M-C-s enter     /
Chercher en arrière avec
expression rationnelle :  M-C-r enter     ?
Aide :                    C-h C-h         :help
Aide Apropos :            C-h a
Aide sur les liens :      C-h b           :help key
Aide Info :               C-h i
Help mode Majeur :        C-h m
Aide tutorial :           C-h t           :help howto
Annulation :              C-_             u
Refaire :                 C-f             ^R
Marquer la position
```

```
du curseur :              C-@             m{a-zA-Z}
Échange Marque
et position :             C-x C-x
Aller à la marque dans
le fichier actuel :                       ’{a-z}
Aller à la marque dans
n’importe quel fichier :                  ’{A-Z}
Copier région :           M-w             {visual}y
Tuer région :             C-w             {visual}d
Copier et garder tampon : C-y
Copier depuis le tampon : M-y             p
Conversion d’une zone
en majuscules :           C-x C-u
Conversion d’une zone
en minuscules :           C-x C-l
Insère un caractère
spécial :                 C-q octalnum/keystroke  ^V decimal/keystroke
Remplace :                M-x replace-string      :%s/aaa/bbb/g
Remplace regexp :         M-x replace-regexp      :%s/aaa/bbb/g
Recherche et remplace :   M-%                     :%s/aaa/bbb/gc
Recherche et remplace :   M-x query-replace
Recherche et remplace avec
expression rationnelle :  M-x query-replace-regexp
Ouvre un fichier :        C-x C-f         :r file
Sauvegarde le fichier :   C-x C-s         :w
Sauve tous les tampons :  C-x s           :wa
Sauvegarde sous :         C-x C-w file    :w file
Demande un tampon :       C-x b
Liste le tampon :         C-x C-b         :buffers
Commute en lecture seule :C-x C-q         :set ro
Demande et tue le tampon :C-x k
Scinde verticalement :    C-x 2           :split
Scinde horizontalement :  C-x 3           :vsplit (ver. 6)
Va à une autre fenêtre :  C-x o           ^Wp
Efface cette fenêtre :    C-x 0           :q
Efface autre fenêtre :    C-x 1
Lance un shell
en arrière-plan :         M-x compile
Tue un shell lancé
en arrière-plan :         M-x kill-compilation
Lance make :                              :make Makefile
Inspecte les messages d’erreur : C-x`
Lance un shell et enregistre :  M-x shell       :!script -a tmp
...clean BS, ...                                :!col -b <tmp >record
sauve/rappelle le shell/enregistre : C-x C-w record  :r record
Lance shell :                    M-! sh          :sh
```

```
Lance une commande :             M-! cmd          :!cmd
Lance une commande et insère :   C-u M-! cmd      :r!cmd
Lance un filtre :                M-| file         {visual}:w file
Lance un filtre et insère :      C-u M-| filter   {visual}:!filter
Montre une option                                :se[t] {option}?
relance l'option                                 :se[t] {option}&
relance une option booléenne                     :se[t] no{option}
commute une option booléenne                     :se[t] inv{option}
met le texte sur 72 colonnes                     :se tw=72
pas d'organisation du texte                      :se tw=0
autoindentation                                  :se ai
étend la tabulation                              :se et
spécifie un commentaire (courrier)               :se comments=n:>,n:\|

Lance GDB                       M-x gdb
décrit le mode GDB              C-h m
saute une ligne                 M-s
va à la ligne suivante          M-n
saute une instruction (stepi)   M-i
Finit le cadre de la pile courante      C-c C-f
continue                        M-c
up arg frames                   M-u
down arg frames                 M-d
copie le numéro à partir du point, insère à la fin
                                C-x &
positionne un point d'arrêt     C-x SPC
```

11.3.5 Configuration de Vim

Afin de mettre en surbrillance les spécificités et la syntaxe avec Vim, ajoutez les lignes suivantes dans `~/.vimrc` ou `/etc/vimrc` :

```
set nocompatible
set nopaste
set pastetoggle=<f11>
syn on
```

Le mode paste permet d'éviter que l'autoindentation n'interfère avec le couper/coller en mode console/terminal. C'est mieux que faire un simple " :set noai".

Voir 'Utiliser GnuPG avec Vim' page 231 pour l'intégration de GnuPG.

11.3.6 Ctags

`apt-get install exuberant-ctags` et lancez `ctags` dans les fichiers source. Taper `:tag` *`function_name`* dans vim pour aller à la ligne qui commence *function_name*. Les balises marchent pour C, C++, Java, Python, beaucoup d'autres langages.

Emacs a les mêmes possibilités avec les ctags.

11.3.7 Convertir un écran avec syntaxe en surbrillance en un source HTML

`so \$VIMRUNTIME/syntax/2html.vim` avec Vim en mode commande convertira le texte en texte html. Sauvegarder avec `:w file.html` et `:q`. Très utile pour du code C, etc.

11.3.8 Ecran scindé avec `vim`

`vim` peut éditer plusieurs fichiers dans un environnement multi-fenêtré en scindant l'écran. Tapez `:help usr_08.txt` pour plus de détails.

Pour scinder l'écran pour afficher différents fichiers, tapez à l'invite de commande de vi :

```
:split another-file
:vsplit another-file
```

Ou à l'invite du shell :

```
$ vi -o file1.txt file2.txt   # Séparation horizontale
$ vi -O file1.txt file2.txt   # Séparation verticale
```

lancera vi en mode multi-fenêtré.

```
$ vimdiff file.txt~ file.txt          # donne les changements récents
                                        de file.txt
$ vimdiff file.en.sgml file.fr.sgml  # donne les changements dans la
                                        traduction
$ gvimdiff file.txt~ file.txt         # sous X
```

fournit une vue sympa des différences entre un original et un fichier de sauvegarde. En SGML, cela fait la correspondance des balises, donc comparer les traductions dans ce mode marche très bien.

Mouvements spéciaux du curseur avec CTRL-W :

```
CTRL-W +      augmente la taille d'une fenêtre
CTRL-W -      réduit la taille d'une fenêtre
CTRL-W h      se déplace vers la fenêtre de gauche
CTRL-W j      se déplace vers la fenêtre en dessous
CTRL-W k      se déplace vers la fenêtre au dessus
CTRL-W l      se déplace vers la fenêtre de droite
...
```

Les commandes suivantes permettent de contrôler le défilement de l'écran :

```
:set scrollbind
:set noscrollbind
```

Chapitre 12

Systèmes de contrôle de version

12.1 Concurrent Versions System (CVS)

Consultez `/usr/share/doc/cvs/html-cvsclient`, `/usr/share/doc/cvs/html-info`, `/usr/share/doc/cvsbook` avec `lynx` ou lancez `info cvs` et `man cvs` pour de l'information détaillée.

12.1.1 Installation d'un serveur CVS

La configuration suivante autorise un enregistrement sur le dépôt CVS seulement par un membre du groupe « src », et l'administration de cvs seulement par un membre du groupe « staff », ceci afin de réduire les risques de conflits.

```
# cd /var/lib; umask 002; mkdir cvs # [Woody] FSH
# apt-get install cvs cvs-doc cvsbook
# export CVSROOT=/var/lib/cvs
# cd $CVSROOT
# chown root:src .  # utiliser « staff » pour restreindre plus
                      pour un projet débutant
# chmod 3775 .      # si la ligne précédente utilise « staff »,
                      alors régler sur 2775
# cvs -d /var/lib/cvs init # il est plus prudent de spécifier
                                          -d explicitement ici !
# cd CVSROOT
# chown -R root:staff .
# chmod 2775 .
# touch val-tags
# chmod 664 history val-tags
# chown root:src history val-tags
```

12.1.2 Exemples de sessions CVS

Ce qui suit configure les environnements de shell pour l'accès au dépôt CVS.

CVS anonyme (téléchargement seulement)

Accès distant en lecture seule :

```
$ export CVSROOT=:pserver:anonymous@cvs.sf.net:/cvsroot/qref
$ cvs login
$ cvs -z3 co qref
```

Utilisation d'un serveur CVS local

Accès local depuuis un shell sur la même machine :

```
$ export CVSROOT=/var/lib/cvs
```

Utilisation d'un pserver CVS distant

Accès distant sans SSH (utilise le protocole RSH de `cvs`) :

```
$ export CVSROOT=:pserver:account@cvs.foobar.com:/var/lib/cvs
$ cvs login
```

Cela peut être sujet à des attaques d'écoute de ligne.

Utilisation d'un CVS distant avec `ssh`

Accès distant avec SSH :

```
$ export CVSROOT=:ext:account@cvs.foobar.com:/var/lib/cvs
```

ou pour SourceForge :

```
$ export CVSROOT=:ext:account@cvs.sf.net:/cvsroot/qref
```

Vous pouvez aussi utiliser l'authentification RSA ('Se connecter avec moins de mots de passe – RSA' page 148), qui élimine la demande du mot de passe.

Créer une nouvelle archive CVS

Pour,

OBJET	VALEUR	SIGNIFICATION
Arbre source :	~/*project-x*	Tous le code source
Nom du Projet :	*project-x*	Nom pour ce projet
Balise vendeur :	*Main-branch*	Balise pour la branche entière
Balise de version :	*Release-original*	Balise pour une version spécifique

Ensuite,

```
$ cd ~/project-x                 # va dans le répertoire du source
 ... crée un arbre source ...
$ cvs import -m "Start project-x" project-x Main-branch Release-initial
$ cd ..; rm -R ~/project-x
```

Travailler avec CVS

Pour travailler avec *project-x* en utilisant le dépôt CVS local :

```
$ cd                            # va à la zone de travail.
$ cvs co project-x              # importe les sources du CVS en local
$ cdproject-x
 ... effectuez des changements au contenu ...
$ cvs diff -u                   # similaire à diff -u repository/ local/
$ cvs up -C modified_file       # undo changes to a file
$ cvs ci -m "Décrire les changements"   # sauvegarde les sources locales
                                          sur CVS
$ vi newfile_added
$ cvs add newfile_added
$ cvs ci -m "Ajout de newfile_added"
$ cvs up                        # récupère la dernière version du CVS
 ... pour créer les sous-répertoires récemment ajoutés, utilisez
 ... "cvs up -d -P" à la place
 ... attention aux lignes commençant par "C filename"
 ... le code non modifié est déplacé vers `.#filename.version'
 ... rechercher "<<<<<<<" et ">>>>>>>" dans filename
$ cvs tag Release-1             # ajoute une balise de version
... modifiez un peu plus ...
$ cvs tag -d Release-1          # enlève une balise de version
$ cvs ci -m "commentaires supplémentaires"
$ cvs tag Release-1             # rajoute une balise de version
$ cd                            # retourne à la zone de travail.
```

```
$ cvs co -r Release-initial -d old project-x
... récupère la version originale dans le répertoire old
$ cd old
$ cvs tag -b Release-initial-bugfixes # création d'une branche (-b)
 ... maintenant vous pouvez travailler sur l'ancienne version (Tag=sticky)
$ cvs update
 ... l'arbre des sources a maintenant la balise permanente
     "Release-initial-bugfixes"
 ... travaillez sur cette branche
$ cvs up # synchronise avec les autres développeurs de la branche
$ cvs ci -m "ajout à la branche"
$ cvs update -kk -A
 ... enlève la balise collante et oublie le contenu
 ... met à jour depuis le tronc principal sans le remplacement des mots-clé
$ cvs update -kk -j Release-initial-bugfixes
 ... fusionne la branche Release-initial-bugfixes avec le tronc principal
 ... sans le remplacement des mots-clé.
     Réparez les conflits avec un éditeur
$ cvs ci -m "merge Release-initial-bugfixes"
$ cd
$ tar -cvzf old-project-x.tar.gz old     # crée une archive, -j pour bz2
$ cvs release -d old     # supprimez les sources locales (optionnel)
```

Options utiles à se rappeler (premier argument de `cvs`) :

```
-n      lancement sec, sans effet
-t      message des étapes de l'activité cvs
```

Exportation de fichiers à partir de CVS

Pour avoir la dernière version par CVS, utiliser « tomorrow » :

```
$ cvs ex -D tomorrow module_name
```

Administrer CVS

Ajouter un alias à un projet (serveur local) :

```
$ su - admin           # un membre de l'équipe
$ export CVSROOT=/var/lib/cvs
$ cvs co CVSROOT/modules
$ cd CVSROOT
$ echo "px -a project-x" >>modules
```

```
$ cvs ci -m "Maintenant px est un alias pour project-x"
$ cvs release -d .
$ exit                 # Ctrl-D pour quitter su
$ cvs co -d project px
 ... récupère project-x (alias:px) de CVS vers le répertoire du projet
$ cd project
 ... faites les changements de contenu ...
```

12.1.3 Résoudre les problèmes de CVS

Permissions de fichiers dans le dépôt

CVS ne va pas écraser le fichier du dépôt courant mais va le remplacer par un autre. Ainsi, *la permission en écriture sur le répertoire de dépôt* est critique. Pour chaque nouvelle création d'un dépôt, lancez ce qui suit pour vous assurer que cette condition et remplie.

```
# cd /var/lib/cvs
# chown -R root:src repository
# chmod -R ug+rwX   repository
# chmod    2775     repository # si besoin est, ça et les sous-répertoires
```

Bit d'exécution

Le bit d'exécution de fichier est conservé lorsqu'on quitte la session. Si on observe n'importe quel problème de permission d'exécution dans les fichiers de contrôle, on peut changer la permission sur ce fichier dans le dépôt CVS avec la commande suivante.

```
# chmod ugo-x filename
```

12.1.4 Commandes CVS

Voici les commandes CVS avec leurs raccourcis.

```
{add|ad|new} [-k kflag] [-m 'message'] files...
{admin|adm|rcs} [rcs-options] files...
{annotate|ann} [options] [files...]
{checkout|co|get} [options] modules...
{commit|ci|com}   [-lnR]  [-m  'log_message'  |  -f  file] \
        [-r revision] [files...]
{diff|di|dif} [-kl] [rcsdiff_options] [[-r rev1 | -D date1] \
        [-r rev2 |  -D date2]] [files...]
{export|ex|exp} [-flNn] -r rev|-D date [-d dir] [-k kflag] module...
```

```
{history|hi|his} [-report] [-flags] [-options args] [files...]
{import|im|imp} [-options] repository vendortag releasetag...
{login|logon|lgn}
{log|lo|rlog} [-l] rlog-options [files...]
{rdiff|patch|pa} [-flags] [-V vn] [-r t|-D d [-r t2|-D d2]] modules...
{release|re|rel} [-d] directories...
{remove|rm|delete} [-lR] [files...]
{rtag|rt|rfreeze} [-falnR]  [-b]  [-d]  [-r  tag  |  -D  date] \
         symbolic_tag modules...
{status|st|stat} [-lR] [-v] [files...]
{tag|ta|freeze} [-lR] [-F] [-b] [-d] [-r tag | -D date]  [-f] \
         symbolic_tag [files...]
{update|up|upd} [-AdflPpR] [-d] [-r tag|-D date] files...
```

12.2 Subversion

Subversion est un système de contrôle de version de nouvelle génération qui a pour but de remplacer CVS. Les développeurs le considèrent actuellement en étape « alpha », mais il est probablement assez stable pour la plupart des utilisations. Aujourd'hui, Subversion est seulement disponible dans Debian `unstable`.

12.2.1 Installation d'un serveur Subversion

Le meta-paquet `subversion-server` dépend des paquets dont on a besoin (`libapache2-dav-svn` et `subversion-tools`) pour configurer un serveur.

Configurer un dépôt

Actuellement, le paquet `subversion` ne configure pas un dépôt, donc il faut le configurer à la main. Un emplacement possible pour un dépôt est `/var/local/repos`.

Créer le répertoire :

```
# mkdir -p /var/local/repos
```

Créer la base de données du dépôt :

```
# svnadmin create /var/local/repos
```

Mettre le dépôt en écriture par le serveur web :

```
# chown -R www-data:www-data /var/local/repos
```

Configuration d'Apache2

Pour permettre l'accès au répertoire par authentification des utilisateurs, ajoutez (ou décommentez) ce qui suit à `/etc/apache2/httpd.conf` :

```
<Location /repos>
  DAV svn
  SVNPath /var/local/repos
  AuthType Basic
  AuthName "Subversion repository"
  AuthUserFile /etc/subversion/passwd
  <LimitExcept GET PROPFIND OPTIONS REPORT>
  Require valid-user
  </LimitExcept>
</Location>
```

Ensuite, créer un fichier d'authentification des utilisateurs avec la commande :

```
htpasswd2 -c /etc/subversion/passwd some-username
```

Relancez Apache2, et votre nouveau dépôt Subversion sera accessible par l'URL http ://*hostname*/repos.

12.2.2 Déplacer un dépôt CVS vers Subversion

12.2.3 Exemples d'utilisation de Subversion

Les sections suivantes vous enseignent comment utiliser les différentes commandes de Subversion.

Créer une nouvelle archive Subversion

Pour créer une nouvelle archive Subversion, tapez ce qui suit :

```
$ cd ~/your-project         # go to your source directory
$ svn import http://localhost/repos your-project \
  project-name -m "initial project import"
```

Cela crée un répertoire nommé *project-name* dans votre dépôt Subversion qui contient vos fichiers de projet. Regardez sur http ://localhost/repos/ s'il est là.

Travailler avec Subversion

Travail sur *project-y* en utilisant Subversion :

```
$ cd                            # aller dans le répertoire de travail
$ svn co http://localhost/repos/project-y  # récupérer les sources
$ cd project-y
 ... travailler ...
$ svn diff                      # similaire à diff -u repository/ local/
$ svn revert modified_file      # annuler les changements sur un
fichier
$ svn ci -m "Describe changes"  # mettre vos changements sur le
dépôt
$ vi newfile_added
$ svn add newfile_added
$ svn add new_dir               # ajoute récursivement tous les
fichiers de new_dir
$ svn add -N new_dir2           # ajoute le répertoire
non-récursivement
$ svn ci -m "Added newfile_added, new_dir,
new_dir2"
$ svn up                        # fusionne la dernière version du dépôt
$ svn log                       # affiche tous les changements
$ svn copy http://localhost/repos/project-y \
      http://localhost/repos/project-y-branch \
      -m "creating my branch of project-y"  # ajouter une branche
project-y
$ svn copy http://localhost/repos/project-y \
      http://localhost/repos/proj-y_release1.0 \
      -m "project-y 1.0 release"    # ajoute une borne de version
 ... notez qu'ajouter une branche ou une borne est pareil.
 ... la seule différence est que l'on peut effectuer des
 ... changements sur une branche et pas sur une borne.

 ... faire des changements sur la branche ...

$ # fusionne la copie de la branche vers la copie principale
$ svn merge http://localhost/repos/project-y \
   http://localhost/repos/project-y-branch
$ svn co -r 4 http://localhost/repos/project-y # récupère la
                                                 revision 4
```

Chapitre 13

Programmation

N'utilisez pas « test » comme nom d'exécutable pour un fichier de test. `test` est une commande du shell.

13.1 Où commencer

Références :
- Documents et exemples sous `/usr/share/doc/`*`packages`*
- Unix / Programming Information (`http://arioch.unomaha.edu/~jclark/#info`)
- *Linux Programming Bible* (John Goerzen/IDG books)

Beaucoup de longs documents informatifs peuvent être obtenus sur papier par GNU (`http://www.gnu.org/`).

Les quatres sections suivantes contiennent quelques scripts d'exemple dans différents langages pour créer un fichier texte d'information sur les comptes utilisateur qui peut être ajouté à `/etc/passwd`, avec par exemple le programme `newusers`. Chaque script prend en entrée un fichier contenant des lignes de la forme `prénom nom mot_de_passe`. (Les répertoires « home » des utilisateurs ne seront pas créés avec ces scripts.)

13.2 Shell

Lire des scripts shell est le **meilleur** moyen de comprendre comment un système Unix fonctionne. Ici, je donne quelques pointeurs et rappels sur la programmation du shell.

13.2.1 Bash – le shell intéractif standard de GNU

Références pour Bash :

- `bash(1)`
- `info bash`
- BASH Programming - Introduction HOWTO (`http://www.tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html`) du LDP comme information pour commencer
- `mc /usr/share/doc/bash/examples/ /usr/share/doc/bash/` (Installez le paquet `bash-doc` pour avoir les exemples.)
- *Learning the bash Shell*, 2nd edition (O'Reilly)

Court exemple de programme (crée des entrées de compte pour `newusers` à partir de l'entrée standard) :

```
#!/bin/bash
# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
pid=1000;
while read n1 n2 n3 ; do
if [ ${n1:0:1} != "#" ]; then
let pid=$pid+1
echo ${n1}_${n2}:password:${pid}:${pid}:,,,/home/${n1}_${n2}:/bin/bash
fi
done
```

13.2.2 Shells POSIX

Plusieurs paquets fournissent un shell POSIX dans Debian :

- `dash` (Sarge)
  - Priorité : optionnel
  - Taille installé : 176
  - Le plus petit et le plus rapide — le meilleur pour le démarrage initial
- `ash` (Woody)
  - Priorité : optionnel
  - Taille installé : 180
  - Plus petit et plus rapide — bon pour le démarrage initial
- `bash`
  - Essentiel : yes
  - Priorité : requis
  - Taille installé : 580
  - Plus gros et plus complet — beaucoup d'extensions sont implémentées
- `pdksh`
  - Priorité : optionnel
  - Taille installé : 408
  - Clone complet du ksh d'AT&T

Si vous écrivez des scripts shell portables, il vaut mieux les écrire comme scripts shell POSIX. Utilisez `/bin/sh` lié vers `ash` ou (`dash`) pour tester sa conformité POSIX. Évitez d'écrire des scripts avec des **bashismes**.

13.2.3 Paramètres du shell

Plusieurs **paramètres spéciaux** à se rappeler :

```
$0       = nom du shell ou du script shell
$1       = premier argument
 ...
$9       = neuvième argument
$#       = nombre de paramètres
"$*"     = "$1 $2 $3 $4 ... $n"
"$@"     = "$1" "$2" "$3" "$4" ... "$n"
$?       = état de sortie de la commande la plus récente
$$       = PID du script shell
$!       = PID du processus en arrière-plan le plus récent
```

Expansions des paramètres de base à se rappeler :

Forme	Si *var* est positionné(*)	Si *var* n'est pas positionné(*)
${*var*:-*string*}	$*var*	*string*
${*var*:+*string*}	*string*	null
${*var*:=*string*}	$*var*	*string* (et lance *var*=*string*)
${*var*:?*string*}	$*var*	(affiche *string* et sort)

Ici, les deux points « : » dans tous ces opérateurs est optionnel.

- Avec « : » = l'opérateur teste l'existence et la non nullité.
- Sans « : » = l'opérateur teste l'existence seulement.

Substitutions de paramètres de base à se rappeler :

Forme	Résultat
${*var*%*suffix*}	Supprime le plus petit motif *suffix*
${*var*%%*suffix*}	Supprime le plus grand motif *suffix*
${*var*#*prefix*}	Supprime le plus petit motif *prefix*
${*var*##*prefix*}	Supprime le plus grand motif *prefix*

13.2.4 Redirection du shell

Redirection de base à se rappeler (ici le [n] est un nombre optionnel) :

[n]> *file*	Redirige la sortie standard (ou n) vers *file*.
[n]>> *file*	Ajoute la sortie standard (ou n) à *file*.
[n]< *file*	Redirige l'entrée standard (ou n) depuis *file*.
[n1]>&n2	Redirige la sortie standard (ou n1) vers n2
> *file* >&2	Redirige l'entrée et l'erreur standards vers *file*.
| *command*	Crée un tuyau (pipe) de l'entrée standard (ou n) vers *command*.
>&2 | *command*	Crée un tuyau (pipe) de l'entrée et l'erreur standards vers *command*.

13.2.5 Opérateurs conditionnels du shell

Toutes les commandes retournent un **état de sortie** qui peut être utilisé par les expressions conditionnelles :
- Succès : 0 (Vrai)
- Erreur : 1 - 255 (Faux)

Notez que l'utilisation de la valeur 0 signifie « vrai » contrairement à d'autres domaines de l'informatique. De même, '[' est l'équivalent de la commande `test`, qui évalue ses arguments jusqu'à ']' comme expression conditionnelle.

Expressions conditionnelles de base à se rappeler :

```
command && si_succès_lance_cette_commande_aussi
command || si_échec_lance_cette_commande_aussi

if [ expression_conditionnelle ]; then
 si_succès_lance_cette_commande
else
 si_échec_lance_cette_commande
fi
```

Les opérateurs de comparaison de **fichiers** dans les expressions conditionnelles sont :

```
-e file         Vrai si file existe.
-d file         Vrai si file existe et est un répertoire.
-f file         Vrai si file existe et est un fichier normal.
-w file         Vrai si file existe et est en écriture.
-x file         Vrai si file existe et est exécutable.
file1 -nt file2 Vrai si file1 est plus récent que file2. (modification)
file1 -ot file2 Vrai si file1 est plus vieux que file2. (modification)
file1 -ef file2 Vrai si ce sont les numéros de périphérique et d'inode.
```

Les opérateurs de comparaison de **chaînes** dans les expressions conditionnelles sont :

```
     -z str    Vrai si la longueur de str est zéro.
     -n str    Vrai si la longueur de str est différente de zéro.
str1 == str2   Vrai si les chaînes sont égales.
str1 = str2    Vrai si les chaînes sont égales.
               ( = peut être utilisé à la place de == )
str1 != str2   Vrai si les chaînes ne sont pas égales.
str1 <  str2   Vrai si str1 est classée avant str2 (dépend de la locale).
str1 >  str2   Vrai si str1 est classée après str2 (dépend de la locale).
```

Les opérateurs de comparaison **arithmétique** entière dans les expressions conditionnelles sont `-eq`, `-ne`, `-lt`, `-le`, `-gt`, ou `-ge`.

13.2.6 Traitement de la ligne de commande

Le shell traite un script de la façon suivante :
- séparation en **motifs** par les caractères : ESPACE, TAB, NEWLINE, ;, (,), <, >, |, &
- recherche de **mots clé** si pas entre “…” ou ’…’ (boucle)
- expansion des **alias** si pas entre “…” ou ’…’ (boucle)
- expansion des **accolades**, A{b|c} -> Ab Ac, si pas entre “…” ou ’…’
- expansion des **tilde**, ~*user* -> $HOME/$USER, si pas entre “…” ou ’…’
- expansion des **paramètres**, $*PARAMETER*, si pas entre ’…’
- expansion des **substitutions de commandes**, $(*command*), si pas entre ’…’
- séparation en **mots** avec $IFS si pas entre “…” ou ’…’
- expansion des **noms de chemins** * ?[] si pas entre “…” ou ’…’
- recherche de la **commande**
  - fonction
  - commande interne
  - fichier dans $PATH
- boucle

Des guillemets simples dans des guillemets doubles n’ont pas d’effet.

13.3 Awk

Références pour Awk :
- *Effective awk Programming*, 3rd edition (O’Reilly)
- *Sed & awk*, 2nd edition (O’Reilly)
- `mawk(1)` et `gawk(1)`
- `info gawk`

Court exemple de programme (crée des entrées de compte pour `newusers`) :

```
#!/usr/bin/awk -f
# Script pour créer un fichier utilisable avec la commande 'newusers'
# à partir d'un fichier contenant des IDs utilisateur et des mots de
# passe sous la forme :
# Prénom Nom Mot_de_passe
# Copyright (c) KMSelf Sat Aug 25 20:47:38 PDT 2001
# Distributed under GNU GPL v 2, or at your option, any later version.
# This program is distributed WITHOUT ANY WARRANTY.

BEGIN {
    # Assign starting UID, GID
    if ( ARGC > 2 ) {
        startuid = ARGV[1]
        delete ARGV[1]
    }
```

```
        else {
            printf( "Usage:  newusers startUID file\n" \
              "...where startUID is the starting userid " \
              "to add, and file is \n" \
              "an input file in form firstname last name password\n" \
            )
            exit
        }

        infile = ARGV[1]
        printf( "Starting UID: %s\n\n", startuid )
    }

    /^#/ { next }

    {
        ++record
        first = $1
        last = $2
        passwd = $3
        user= substr( tolower( first ), 1, 1 ) tolower( last )
        uid = startuid + record - 1
        gid = uid
        printf( "%s:%s:%d:%d:%s %s,,/home/%s:/bin/bash\n",  \
            user, passwd, uid, gid, first, last, user \
            )
    }
```

Deux paquets fournissent un `awk` POSIX dans Debian :

- `mawk`
  - Priorité : requis
  - Taille installé : 228
  - Plus petit et beaucoup plus rapide — bon pour l'installation par défaut
  - Des limites définies à la compilation existent
    - NF = 32767
    - sprintf buffer = 1020
- `gawk`
  - Priorité : Optionnel
  - Taille installé : 1708
  - Plus gros et plus complet — beaucoup d'extensions sont implémentées
    - Version d'UNIX System V Release 4
    - awk de Bell Labs
    - spécifiques à GNU

13.4 Perl

C'est l'interpréteur sur un système Unix.

Références pour Perl :
- `perl(1)`
- *Programming Perl*, 3rd edition (O'Reilly)

Court exemple de programme (crée une entrée `newusers`) :

```
#!/usr/bin/perl
# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
$pid=1000;
while (<STDIN>) {
        if (/^#/) { next;}
        chop;
        $pid++;
        ($n1, $n2, $n3) = split / /;
        print $n1,"_",$n2,":", $n3, ":",$pid,
                  ":",$pid,",,,/home/",$n1,"_",$n2,":/bin/bash\n"
}
```

Installer le module Perl *modulename* :

```
# perl -MCPAN -e 'install modulename'
```

13.5 Python

C'est un interpréteur orienté objet sympa.

Références pour Python :
- `python(1)`
- *Learning Python* (O'Reilly).

Court exemple de programme (crée une entrée `newusers`) :

```
#! /usr/bin/env python
import sys, string

# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
# Ported from awk script by KMSelf Sat Aug 25 20:47:38 PDT 2001
# This program is distributed WITHOUT ANY WARRANTY.

def usages():
    print \
```

```
"Usage:  ", sys.argv[0], " start_UID [filename]\n" \
"\tstartUID is the starting userid to add.\n" \
"\tfilename is input file name. If not specified, standard input.\n\n" \
"Input file format:\n"\
"\tfirstname lastname password\n"
                return 1

def parsefile(startuid):
    #
    # main filtering
    #
    uid = startuid
    while 1:
        line = infile.readline()
        if not line:
            break
        if line[0] == '#':
            continue
        (first, last, passwd) = string.split(string.lower(line))
        # crashe si le nombre de paramètres est mauvais :-)
        user = first[0] + last
        gid = uid
        lineout = "%s:%s:%d:%d:%s %s,,/home/%s:/bin/bash\n" %  \
            (user, passwd, uid, gid, first, last, user)
        sys.stdout.write(lineout)
        +uid

if __name__ == '__main__':
    if len(sys.argv) == 1:
        usages()
    else:
        uid = int(sys.argv[1])
        #print "# UID start from: %d\n" % uid
        if len(sys.argv) > 1:
            infilename   = string.join(sys.argv[2:])
            infile = open(infilename, 'r')
            #print "# Read file from: %s\n\n" % infilename
        else:
            infile = sys.stdin
        parsefile(uid)
```

13.6 Make

Références pour Make :

- `info make`
- `make(1)`
- *Managing Projects with make,* 2nd edition (O'Reilly)

Variables automatiques simples :

Syntaxe des règles :

```
Target: [ Prerequisite ... ]
 [TAB]  command1
 [TAB]  -command2 # ignore les erreurs
 [TAB]  @command3 # supprime l'écho
```

Ici, `[TAB]` est un code TAB. Chaque ligne est interprétée par le shell après la substitution des variables par make. Utiliser `\` à la fin d'une ligne pour continuer le script. Utiliser `$$` pour entrer `$` pour les variables d'environnement du script shell.

Equivalents des règles implicites :

```
.c:   header.h == %  : %.c header.h
.o.c: header.h == %.c: %.o header.h
```

Variables automatiques pour les règles ci-dessus :

```
foo.o: new1.c new2.c.c old1.c new3.c
$@ == foo.o                        (cible)
$< == new1.c                       (premier)
$? == new1.c new2.c new3.c         (plus récent)
$^ == new1.c new2.c.c old1.c new3.c (tout)
$* == `%' motif correspondant dans le motif cible.
```

Référence des variables :

```
foo1 := bar    # expansion unique
foo2  = bar    # expansion récursive
foo3 += bar    # ajoute
SRCS := $(wildcard *.c)
OBJS := $(foo:c=o)
OBJS := $(foo:%.c=%.o)
OBJS := $(patsubst %.c,%.o,$(foo))
DIRS  = $(dir directory/filename.ext) # Extrait "directory"
$(notdir NAMES...), $(basename NAMES...), $(suffix NAMES...) ...
```

Exécuter `make -p -f/dev/null` pour voir les règles automatiques internes.

13.7 C

Préparation :

```
# apt-get install glibc6-doc manpages-dev
```

Références pour le C :

- `info libc` (C library function reference)
- `gcc(1)`
- `<var>nom_de_la_fonction_de_la_bibliothèque_C</var>(3)`
- Kernighan & Ritchie, *The C Programming Language*, 2nd edition (Prentice Hall).

13.7.1 Programme simple en C (gcc)

Un exemple simple pour compiler `example.c` avec la bibliothèque `libm` dans l'exécutable `run_example` :

```
$ cat > 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); /* empêche un dépassement de tampon */
        y[10] = '\0'; /* remplissage pour être sûr que la chaîne finit
                  par '\0' */
        printf("%5i, %5.3f, %10s, %10s\n", argc, x, y, argv[1]);
        return 0;
}
EOF
$ gcc -Wall -g -o run_example exmple.c -lm
$ ./run_example
        1, 2.915, ./run_exam,     (null)
$ ./run_example 1234567890qwerty
        2, 3.082, ./run_exam, 1234567890qwerty
```

Ici, `-l`**m** est nécessaire pour lier la bibliothèque `lib`**m**, pour utiliser sqrt(). La bibliothèque est en fait dans `/lib` sous le nom `libm.so.6` qui est un lien symbolique vers `libm-2.1.3.so`.

Regardez le dernier paramètre dans le texte affiché. Il y a plus de 10 caractères affichés alors que `%10s` est spécifié.

L'utilisation de fonctions effectuant des opérations sur des pointeurs sans vérifier la limite, comme `sprintf` et `strcpy`, est dépréciée pour prévenir les exploits de dépassement de tampon, annulant les effets ci-dessus. Utilisez `snprintf` et `strncpy` à la place.

13.7.2 Déboguer

Deboguer avec `gdb`

Préparation :

```
# apt-get install gdb
```

Références pour `gdb` :
- `info gdb` (didacticiel)
- `gdb(1)`
- `http://www.unknownroad.com/rtfm/gdbtut/gdbtoc.html`

Utilisez `gdb` pour déboguer des programmes compilés avec l'option `-g`. Beaucoup de commandes peuvent être abbréviées. L'expansion par tab marche comme avec le shell.

```
$ gdb program
(gdb) b 1                # positionne un point d'arrêt à la ligne 1
(gdb) run arg1 arg2 arg3 # exécute le programme
(gdb) next               # ligne suivante
...
(gdb) step               # pas en avant
...
(gdb) p parm             # affiche parm
...
(gdb) p parm=12          # met la valeur de parm à 12
```

Pour déboguer depuis Emacs, référez-vous à 'Résumé des commandes de l'éditeur (Emacs, Vim)' page 197.

Vérifier les dépendances sur des bibliothèques

Utilisez `ldd` pour trouver les dépendances d'un programme sur des bibliothèques :

```
$ ldd /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` marche dans un environnement `chrooté`, les bibliothèques suivantes doivent être disponibles dans votre environnement `chrooté`.

Les commandes suivantes sont aussi utiles :

- `strace` : trace les appels et signaux système
- `ltrace` : trace les appels aux bibliothèques

Déboguer avec des outils de détection des fuites de mémoire

Il y a plusieurs outils de détection des fuites de mémoire disponibles dans Debian.

- `njamd`
- `valgrind`
- `dmalloc`
- `electric-fence`
- `memprof`
- `memwatch`
- `mpatrol`
- `leaktracer`
- `libgc6`
- `Insure++` de Parasoft (`http://www.parasoft.com`). (non-libre, commercial gratuit)

Consultez aussi Debugging Tools for Dynamic Storage Allocation and Memory Management (`http://www.cs.colorado.edu/homes/zorn/public_html/MallocDebug.html`).

13.7.3 Flex – un meilleur Lex

`flex` est un générateur rapide d'analyseur lexical.

Références pour `flex` :

- `info flex` (didacticiel)
- `flex(1)`

Vous devez fournir vos propres `main()` et `yywrap()`, ou votre `program.l` devrait ressembler à ce qui suit pour compiler sans bibliothèque. (`yywrap` est une macro; `%option main` active `%option noyywrap` implicitement) :

```
%option main
%%
.|\n    ECHO ;
%%
```

Autrement, vous pouvez compiler avec l'option de l'éditeur de liens -`lfl` à la fin de la ligne de commande de `cc` (comme ATT-Lex avec -`ll`). `%option` n'est pas nécessaire dans ce cas.

13.7.4 Bison – un meilleur Yacc

Quelques paquets fournissent un générateur d'analyseur LALR compatible avec Yacc dans Debian :

- `bison` : générateur d'analyseur LALR GNU
- `btyacc` : générateur d'analyseur avec retour arrière basé sur byacc

Références pour `bison` :

- `info bison` (didacticiel)
- `bison(1)`

Vous devez fournir vos propres `main()` et `yyerror()`. `main()` appelle `yyparse()` qui appelle `yylex()`, habituellement créé avec Flex.

```
%%

%%
```

13.7.5 Autoconf

`autoconf` est un otil pour produire des scripts shell qui configurent automatiquement les paquets de code source pour s'adapter aux nombreux systèmes Unix qui utilisent le système de compilation GNU entier.

`autoconf` produit le script de configuration `configure`. `configure` crée automatiquement un `Makefile` personnalisé et `Makefile.am`.

Compiler et installer un programme

Debian ne touche pas aux fichiers de `/usr/local` (voir 'Support de la diversité' page 22). Donc si vous compilez un programme depuis le source, installez le dans `/usr/local` pour qu'il n'interfère pas avec Debian.

```
$ cd src
$ ./configure --prefix=/usr/loca
$ make
$ make install # cela met les fichiers dans le système
```

Désinstaller un programme

SI vous avez toujours les sources et si elles utilisent `autoconf/automake` et si vous vous souvenez comment vous avez configuré la compilation :

```
$ ./configure les-options-de-configuration
# make uninstall
```

Autrement, si vous êtes absolument sûr que le processus d'installation n'a mis des fichiers que sous `/usr/local` et qu'il n'y a rien d'important là, vous pouvez supprimer tout son contenu avec :

```
# find /usr/local -type f -print0 | xargs -0 rm -f
```

Si vous n'êtes pas sûr de l'emplacement des fichiers installés, vous devriez considérer l'utilisation de `checkinstall` qui fournit un moyen de désinstallation propre.

13.8 Préparation de documents

13.8.1 Traitement de texte **`roff`**

Traditionnellement, `roff` est le système de traitement de texte principal d'Unix.

Voir `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` ».

Un bon didacticiel sur les macros `-me` existe. Si vous avez groff (1.18 ou plus récent), cherchez le fichier `/usr/share/doc/groff/meintro.me.gz` et faites ce qui suit :

```
$ zcat /usr/share/doc/groff/meintro.me.gz | \
     groff -Tascii -me - | less -R
```

Ce qui suit construira un fichier texte :

```
$ zcat /usr/share/doc/groff/meintro.me.gz | \
    GROFF_NO_SGR=1 groff -Tascii -me - | col -b -x > meintro.txt
```

Pour imprimer, utilisez la sortie PostScript.

```
$ groff -Tps meintro.txt | lpr
$ groff -Tps meintro.txt | mpage -2 | lpr
```

13.8.2 SGML

Préparation :

```
# apt-get install debiandoc-sgml debiandoc-sgml-doc
```

Références pour `debiandoc-sgml` :
- `/usr/share/doc/debiandoc-sgml-doc`

- `debiandoc-sgml(1)`
- *DocBook : The Definitive Guide*, par Walsh et Muellner (O'Reilly)

SGML permet de gérer plusieurs formats pour un document. Un système SGML facile est Debiandoc, qui est utilisé ici. Cela requiert des conversions mineurs dans le fichier texte original, pour les caractères suivants :

```
<   &lt;
>   &gt;
" " "&nbsp;"  (espace non séquable)
&   &amp;
%   &percnt;
©   &copy;
-   &ndash;
--   &mdash;
```

Pour marquer une section en tant que commentaire non imprimable, faites :

```
<!-- Indiquez le problème ici ... -->
```

Pour marquer une section en tant que commentaire ou pas à l'impression, faites :

```
<![ %FIXME; [ Indiquez le problème ici ... ]]>
```

En SGML, la *première définition* d'une entité l'emporte. Par exemple :

```
<!entity % qref "INCLUDE">
<![ %qref; [ <!entity param "Data 1"> ]]>
<!entity param "Data 2">
&param;
```

Cela termine en tant que « Data 1 ». Si la première ligne inclut « IGNORE » au lieu de « INCLUDE », cela termine en tant que « Data 2 » (la deuxième ligne est une expression conditionnelle). De plus, les phrases souvent répétées peuvent être définies à l'avance séparément du contexte.

```
<!entity whoisthis "my">
Hello &whoisthis; friend.
This is &whoisthis; book.
```

Le résultat est :

```
Hello my friend.
This is my book.
```

Voir le court exemple de SGML `sample.sgml` dans les exemples (`http://www.debian.org/doc/manuals/debian-reference/examples/`).

Lorsque des documents SGML deviennent plus gros, TeX cause parfois des erreurs. Il faut augmenter la taille du bassin dans `/etc/texmf/texmf.cnf` (ou mieux éditez `/etc/texmf/texmf.d/95NonPath` et lancez `update-texmf`) pour réparer cela.

13.9 Paquetage

Préparation :

```
# apt-get install debian-policy developers-reference \
        maint-guide dh-make debhelper
# apt-get install packaging-manual # dans le cas de Potato
```

Références pour le paquetage :
- 'Système de gestion des paquets Debian' page 11 (bases)
- Guide des nouveaux responsables Debian (didacticiel)
- `dh-make(1)`
- Référence du développeur Debian (meilleure utilisation)
- Charte Debian (officielle)
- Manuel de paquetage (Potato)

13.9.1 Paqueter un binaire seul

Méthode rapide et sale pour paqueter un binaire seul par Joey Hess.

```
# mkdir -p mypkg/usr/bin mypkg/DEBIAN
# cp binary mypkg/usr/bin
# cat > mypkg/DEBIAN/control
Package: mypackage
Version: 1
Architecture: i386
Maintainer: Joey Hess <joeyh@debian.org>
Description: my little package
Don't expect much.
^D
# dpkg-deb -b mypkg
```

13.9.2 Paqueter avec des outils

Utilisez `dh_make` du paquet `dh-make` pour créer l'architecture du paquet. Ensuite, suivez les instructions de `dh-make(1)`, qui utilise `debhelper` dans `debian/rules`.

Une approche plus ancienne est d'utiliser `deb-make` du paquet `debmake`. Cela n'utilise aucun script `debhelper` et dépend seulement du shell.

Pour plusieurs paquets source, voir « mc » (`dpkg-source -x mc_4.5.54.dsc`) qui utilise « sys-build.mk » par Adam Heath (`<doogie@debian.org>`) et « glibc » (`dpkg-source -x glibc_2.2.4-1.dsc`) qui utilise un autre système par feu Joel Klecker (`<espy@debian.org>`).

Chapitre 14

GnuPG

Références :
- `gpg(1)`.
- `/usr/share/doc/gnupg/README.gz`
- *Manuel de GNU Privacy* dans `/usr/share/doc/gnupg-doc/GNU_Privacy_Handbook/` (installez le paquet `gnupg-doc`)

14.1 Installer GnuPG

```
# gpg --gen-key                     # génère une nouvelle clé
# gpg --gen-revoke my_user_ID       # génère une clé de révocation
                                      pour my_user_ID
# host -l pgp.net | grep www|less # cherche les serveurs de clés pgp
```

Aujourd'hui, les bons serveurs de clés sont :

```
keyserver wwwkeys.eu.pgp.net
keyserver wwwkeys.pgp.net
```

Il faut faire attention de **ne pas créer plus de deux sous-clés**. Si vous le faites, les serveurs de clés de pgp.net vont **corrompre** votre clé. Utilisez un paquet `gnupg` récent (>1.2.1-2) pour gérer ces sous-clés corrompues. Voir `http://fortytwo.ch/gpg/subkeys`.

De plus, seul un serveur de clé peut être spécifié dans `$HOME/.gnupg/options`.

Malheureusement, les serveurs suivants ne fonctionnent plus :

```
keyserver search.keyserver.net
keyserver pgp.ai.mit.edu
```

14.2 Utiliser GnuPG

Gestion des fichiers :

```
$ gpg  [options]  command  [args]
$ gpg {--armor|-a} {--sign|-s} file # signe file dans le fichier texte
                                      file.asc
$ gpg --clearsign file              # signe file sans le chiffrer
$ gpg --clearsign --not-dash-escaped patchfile  # signe sans le chiffrer
                                      patchfile
$ gpg --verify file         # vérifie un fichier signé sans chiffrement
$ gpg -o file.sig {-b|--detach-sign} file # crée une signature détachée
$ gpg --verify file.sig file        # vérifie file à l'aide de file.sig
$ gpg -o crypt_file {--recipient|-r} name {--encrypt|-e} file
        # chiffrement à clé publique pour le destinataire name
$ gpg -o crypt_file {--symmetric|-c} file # chiffrement symétrique
$ gpg -o file --decrypt crypt_file  # déchiffrement
```

14.3 Gérer GnuPG

Gestion des clés :

```
$ gpg --edit-key user_ID               # « help » pour l'aide, intéractive
$ gpg -o file --exports                # exporte toutes les clés
                                         vers le fichier file
$ gpg --imports file                   # importe toutes les clés
                                         depuis le fichier file
$ gpg --send-keys user_ID              # envoie la clé de user_ID
                                         vers un serveur de clés
$ gpg --recv-keys user_ID              # reçoit la clé de user_ID
                                         depuis le serveur de clés
$ gpg --list-keys user_ID              # liste les clés de user_ID
$ gpg --list-sigs user_ID              # liste les  signatures de user_ID
$ gpg --check-sigs user_ID             # vérifie la signature de user_ID
$ gpg --fingerprint user_ID            # vérifie l'empreinte de user_ID
$ gpg --list-sigs | grep '^sig' | grep '[User id not found]' \
  | awk '{print $2}' | sort -u | xargs gpg --recv-keys
  # récupère les clés inconnues
  # met à jour les clés pour toutes les signatures inconnues
```

Code de confiance :

```
-         Pas de confiance assignée au possesseur / pas encore calculée
```

```
e         Calcul de la confiance échoué
q         Pas assez d'informations pour le calcul
n         Ne jamais faire confiance à cette clé
m         Confiance marginale
f         Confiance complète
u         Confiance ultime
```

Les commandes suivantes téléchargent ma clé « *A8061F32* » vers plusieurs serveurs de clés :

```
$ for xx in us es cz de dk uk ch net.uk earth.net.uk; \
$ do gpg --keyserver wwwkeys.$xx.pgp.net  --send-keys A8061F32; done
```

14.4 Utiliser GnuPG avec des applications

14.4.1 Utiliser GnuPG avec Mutt

Ajoutez les instructions suivantes à `~/.muttrc` pour éviter que GnuPG, qui est lent, s'exécute automatiquement, mais l'autoriser à être utilisé en tapant '`S`' au menu principal.

```
macro index S ":toggle pgp_verify_sig\n"
set pgp_verify_sig=no
```

14.4.2 Utiliser GnuPG avec Vim

Ajoutez le contenu de `_vimrc` obtenu depuis le sous-répertoire des exemples (`http://www.debian.org/doc/manuals/debian-reference/examples/`) dans `~/.vimrc` pour lancer GnuPG de façon transparente.

Chapitre 15

Support Debian

Les références suivantes fournissent de l'aide et des conseils pour Debian. Utilisez au mieux ces documents pour vous aider vous-même avant de venir crier sur les listes de diffusion. :)

Notez que vous avez accès à beaucoup de documentation sur votre système en utilisant un navigateur web, en utilisant les commandes `dwww` ou `dhelp`, que vous trouverez dans les paquets du même nom.

15.1 Références

Les références suivantes sont disponibles pour Debian et Linux en général. Si leur contenu est en conflit avec l'un l'autre, comptez toujours davantage sur les sources primaires d'informations que sur les sources secondaires telles que ce document.

- Manuel d'installation (primaire)
  - À lire avant d'installer et de mettre à jour.
  - Web : `http://www.debian.org/releases/stable/installmanual`
  - Web : `http://www.debian.org/releases/testing/installmanual` (en cours d'écriture)
  - Paquet : `install-doc`
  - Fichier : `/usr/share/doc/install-doc/index.html`
- Les notes de version (primaire)
  - À lire absolument avant l'installation et la mise à jour même si avez de bonnes connaissances.
  - Web : `http://www.debian.org/releases/stable/releasenotes`
  - Web : `http://www.debian.org/releases/testing/releasenotes` (en cours d'criture)
  - Paquet : `install-doc`
- FAQ (secondaire)
  - Foire aux questions
  - Web : `http://www.debian.org/doc/manuals/debian-faq/`

  - Paquet : `doc-debian`
  - Fichier : `/usr/share/doc/debian/FAQ/index.html`

- Guide de référence Debian (secondaire)
  - Manuel utilisateur le plus complet
  - Web : `http://www.debian.org/doc/manuals/debian-reference/`
  - Paquet : `debian-reference-fr`
  - Fichier : `/usr/share/doc/Debian/reference/`

- APT HOWTO (secondaire)
  - Guide détaillé pour l'utilisateur de la gestion de paquets Debian (woody)
  - Web : `http://www.debian.org/doc/manuals/apt-howto/`
  - Paquet : `apt-howto`
  - Fichier : `/usr/share/doc/Debian/apt-howto/`

- Manuel de sécurisation Debian (secondaire)
  - Guide de l'utilisateur détaillé pour sécuriser et renforcer l'installation par défaut de Debian. (woody)
  - Web : `http://www.debian.org/doc/manuals/securing-debian-howto/`
  - Paquet : `harden-doc`
  - Fichier : `/usr/share/doc/harden-doc/html/securing-debian-howto/`

- Documentation `dselect` pour débutants (secondaire)
  - Didacticiel pour `dselect`
  - Web : `http://www.debian.org/releases/woody/i386/dselect-beginner`
  - Paquet : `Not available in install-doc: Bug#155374`
  - Fichier : `DebianCDunder/doc/`

- La charte Debian (primaire)
  - Charte Technique de Debian.
  - Web : `http://www.debian.org/doc/debian-policy/`
  - Paquet : `debian-policy`
  - Fichier `/usr/share/doc/debian-policy/`

- Référence du développeur Debian (primaire)
  - Savoir de base pour les développeurs.
  - Chacun devrait aussi le consulter au moins une fois.
  - Web : `http://www.debian.org/doc/manuals/developers-reference/`
  - Paquet : `developers-reference`
  - Fichier : `/usr/share/doc/developers-reference/`

- Guide des nouveaux responsables Debian (primaire)
  - Guide pratique pour les développeurs.
  - Didacticiel d'empaquetage.
  - Web : `http://www.debian.org/doc/manuals/maint-guide/`
  - Paquet : `maint-guide`
  - Fichier : `/usr/share/doc/maint-guide/`

- Manuel d'empaquetage (potato)

  - Paquet `packaging-manual` dans Potato. (Déplacé dans l'annexe de *Référence du développeur*)
- Pages de manuel Unix (primaire)
  - `man` *`package-name`*
- Pages info GNU (primaire)
  - `info` *`package-name`*
- Documents spécifiques à chaque paquet (primaire)
  - Retrouvez-les sous `/usr/share/doc/`*`package-name`*
- LDP : Linux Documentation Project (secondaire)
  - HOWTOs et mini-HOWTOS généraux pour Linux
  - Web : `http://www.tldp.org/`
  - Paquet : `doc-linux-text`
  - Fichier : `/usr/share/doc/HOWTO/`
- Projet de Documentation Debian (secondaire)
  - Manuels spécifiques pour Debian
  - Web : `http://www.debian.org/doc/`
- Le coin du développeur Debian (secondaire)
  - Informations clef pour les responsables de paquets Debian
  - Permet à l'utilisateur final de mieux comprendre la Debian
  - Web : `http://www.debian.org/devel/`
- Code source (absolument primaire)
  - Personne ne peut donner d'arguments contre ça :-)
  - Téléchargez le code source en suivant les instructions de 'Le code source' page 11

Les références suivantes sont disponibles pour Unix en général. Notez qu'il y a de petites différences entre les différents systèmes Unix. Les noms de périphériques et les méthodes de démarrage nécessitent une attention particulière.

- *The UNIX Programming Environment*
  - Le livre à lire pour comprendre comment Unix fonctionne.
  - Par B. W. Kernighan et R. Pike,
  - Publié par Princeton Hall Software Series
- *The C Programming Language* (second edition)
  - Le livre à lire pour apprendre le C ANSI.
  - Par B. W. Kernighan et D. M. Ritchie
  - Publié par Princeton Hall Software Series
- *UNIX Power Tools*
  - Le livre à lire pour apprendre des astuces Unix.
  - Par Jerry Peek, Tim O'Reilly et Mike Loukides
  - Publié par O'Reilly and Associates
- *Essential System Administration* (second edition)
  - Le livre à lire pour apprendre l'administration système Unix pour plusieurs saveurs d'Unix.

  - Par Aeleen Frisch
  - Publié par O'Reilly and Associates
- Bell Labs : Computing Sciences Research
  - Riche archive d'histoire d'Unix
  - Page prrincipale : `http://cm.bell-labs.com/cm/cs/`
  - Sélection de rapports techniques : `http://cm.bell-labs.com/cm/cs/cstr.html`
  - Quelques publications : `http://cm.bell-labs.com/cm/cs/papers.html`
- Support général pour Linux en ligne
  - Debian Planet (`http://www.debianplanet.org/`)
  - debianHELP (`http://www.debianhelp.org/`)
  - Linux.com (`http://linux.com/`)
  - The Linux Home Page at Linux Online (`http://www.linux.org/`)
  - Red Hat (Vendeur commercial de Linux) (`http://www.redhat.com/`) (RPM, Sys-V init)
  - SuSE, Inc. (Vendeur commercial de Linux) (`http://www.suse.de/`) (RPM, Sys-V init)
  - Slackware (`http://www.slackware.com/`) (TGZ, BSD-style init)
- Guides généraux pour Unix en ligne
  - A UNIX Introductory Course from Ohio State University (`http://www-wks.acs.ohio-state.edu/unix_course/unix.html`)
  - UNIXhelp from The University of Edinburgh (`http://unixhelp.ed.ac.uk/`)
  - Unix / Programming Information (`http://arioch.unomaha.edu/~jclark/#info`)
  - comp.unix.questions FAQ (`http://www.faqs.org/faqs/unix-faq/faq/`)
  - comp.unix.user-friendly FAQ (`http://www.camelcity.com/~noel/usenet/cuuf-FAQ.htm`)
  - FreeBSD Documentation (`http://www.freebsd.org/docs.html`)
  - The FreeBSD Handbook (`http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/index.html`)
  - UNIX GUIDE (`http://ieee.uow.edu.au/documents/`)
- Pages principales de projets libres
  - GNU Project (`http://www.gnu.org/`)
  - The Linux Documentation Project (`http://www.tldp.org/`)
  - The Linux Kernel Archives (`http://www.linux.org/`)
  - The XFree86 Project, Inc (`http://www.xfree86.org/`)
  - GNOME (`http://www.gnome.org/`)
  - K Desktop Environment (`http://www.kde.org/`)
  - GNU software at Red Hat (`http://sources.redhat.com/`)
  - Mozilla (`http://www.mozilla.org`)
  - FreeBSD (`http://www.freebsd.org/`)
  - OpenBSD (`http://www.openbsd.org/`)
  - NetBSD (`http://www.netbsd.org/`)

15.2 Trouver le sens d'un mot

Beaucoup de mots utilisés dans Debian sont des mots issus du jargon technique ou des acronymes. L'instruction suivante répondra à la plupart des questions de vocabulaire :

```
$ dict mettez un mot bizarre ici;
```

15.3 Trouver la popularité d'un paquet Debian

Il y a beaucoup de paquets dans Debian et il est parfois difficile de savoir lequel essayer en premier. Voir le site Debian Popularity Contest Results (`http://popcon.debian.org/`) pour savoir ce que les autres utilisent. Installez aussi le paquet `popularity-contest` pour y contribuer.

15.4 Système de suivi des bogues Debian

La distribution Debian a un système de suivi de bogues bug tracking system (BTS) (`http://bugs.debian.org/`) dans lequel les détails des bogues rapportés par les utilisateurs et les développeurs sont enregistrés. Chaque bogue reçoit un numéro, et est répertorié dans un fichier jusqu'à ce qu'il soit marqué comme ayant été traité.

Vous devriez vérifier que votre bogue n'a pas déjà été signalé par quelqu'un d'autre avant de le soumettre. Les listes des bogues non encore résolus sont disponibles sur la Toile (`http://bugs.debian.org/`) et ailleurs (`http://www.debian.org/Bugs/Access`). Voyez aussi 'Rechercher les bogues de Debian et demander de l'aide' page 70.

La méthode pour rapporter un bogue est décrite à l'adresse `http://www.debian.org/Bugs/Reporting`.

15.5 Listes de diffusion

Lisez au moins « debian-devel-announce » (Anglais, lecture seule et trafic bas) pour être au courant des nouvelles sur Debian.

Les listes de diffusions les plus utiles aux utilisateurs de Debian sont « debian-user » (Anglais, ouverte et trafic important) et autres « debian-user-*langue* » (pour les autres langues).

Pour plus d'informations sur ces listes et la manière de s'y inscrire voyez `http://lists.debian.org/`. Veuillez rechercher dans les archives si vos questions n'ont pas déja trouvé de réponses avant de poster, et respecter le code de conduite des listes.

15.6 Internet Relay Chat (IRC)

IRC (Internet Relay Chat, Dialogue par Internet) est un moyen de discuter avec des gens de partout dans le monde en temps réel. Des canaux IRC dédiés à Debian peuvent être trouvés sur le réseau IRC freenode (`http://www.freenode.info/`). Pour s'y connecter, vous avez besoin d'un client IRC. Les clients les plus populaires sont XChat, BitchX, ircII, irssi, epic4 et KSirc, qui ont tous été empaquetés pour Debian. Une fois le client installé, vous devez lui dire de se connecter au serveur. Avec la plupart des clients, vous pouvez le faire en tapant :

```
/server irc.debian.org
```

Une fois connecté, rejoignez le canal #debian en tapant :

```
/join #debian
```

Note : les clients comme XChat ont souvent une interface graphique différente pour rejoindre les serveurs/canaux.

15.7 Moteurs de recherche

De nombreux moteurs de recherche offrent des documentations liées à Debian :

- Page de recherche du site Debian (`http://search.debian.org/`).
- Google (`http://www.google.com/`) : inclure « site :debian.org » comme terme de recherche.
- Google Groups (`http://groups.google.com/`) : un moteur de recherche pour groupes de nouvelles. Inclure « linux.debian.user » comme terme de recherche.
- AltaVista (`http://www.altavista.com/`)

Par exemple, chercher le terme « cgi-perl » donne une description de ce paquet plus détaillée que le texte bref dans le champs description dans son fichier de contrôle. Voir 'Rechercher les bogues de Debian et demander de l'aide' page 70 pour plus de détails à ce sujet.

15.8 Sites internet

Les sites suivants sont des sites que j'ai référencés pour des sujets spécifiques.

- Paquets d'Adrian Bunk pour faire tourner un noyau 2.4.x sur potato (`http://www.fs.tum.de/~bunk/`)
- Linux sur portables (`http://www.linux-laptop.net/`)
- Xterm FAQ (`http://dickey.his.com/xterm/xterm.faq.html`)
- EXT3 File System mini-HOWTO (`http://www.zip.com.au/~akpm/linux/ext3/ext3-usage.html`)

- Support de Grands Fichiers par Linux (`http://www.suse.de/~aj/linux_lfs.html`)
- Window Managers for X (`http://www.xwinman.org`)
- Linux USB Project (`http://www.linux-usb.org/`)
- Pages SuSE pour CJK (`http://www.suse.de/~mfabian/suse-cjk/suse-cjk.html`)
- LNX-BBC (Projet de boot CD au format carte de visite) (`http://www.lnx-bbc.org/`)
- Linux info par Karsten Self (Partitionnement, sauvegardes, navigateurs…) (`http://kmself.home.netcom.com/Linux/index.html`)
- Backup info HOWTO par Alvin Oga (`http://www.Linux-Backup.net/`)
- Security info HOWTO par Alvin Oga (`http://www.Linux-Sec.net/`)
- Various UNOFFICIAL sources for APT (`http://www.apt-get.org/`)
- Configuration Ethernet d'un portable (`http://www.orthogony.com/gjw/lap/lap-ether-intro.html`)

Annexe A

Annexe

A.1 Auteurs

Guide de référence pour Debian a été démarré par Osamu Aoki <debian@aokiconsulting.com> en tant que mémo d'installation personnel et finalement appelé « Quick Reference … ». Une grande partie du contenu venait des archives de la liste de diffusion « debian-user ». Le « Manuel d'installation Debian » et les « Notes de version Debian » ont aussi été utilisés.

Suite à une suggestion de Josip Rodin, qui est très actif dans le Projet de Documentation Debian (`http://www.debian.org/doc/ddp`) (DDP) et est le responsable actuel de la « FAQ Debian », ce document a été renommé en « Guide de référence pour Debian » et fusionné avec des chapitres de la « FAQ Debian » ayant un contenu de référence. Par la suite, l'extrait « Guide de référence rapide pour Debian » a été créé.

Ce document a été modifié, traduit et amélioré par les membres de l'équipe QREF suivants :

- Relecture et réécriture en Anglais du « Quick Reference… » original
  - Osamu Aoki <`osamu\#at\#debian.org`> (leader : tout le contenu)
- Relecture et réécriture en Anglais
  - David Sewell <`dsewell\#at\#virginia.edu`> (leader : style en)
  - Thomas Hood <`jdthood\#at\#yahoo.co.uk`> (network related)
  - Brian Nelson <`nelson\#at\#bignachos.com`> (especially X related)
  - Jan Michael C Alonzo <`jmalonzo\#at\#spaceants.net`>
  - Daniel Webb <`webb\#at\#robust.colorado.edu`>
- Traduction en Français
  - Guillaume Erbs <`gerbs\#at\#free.fr`> (leader : fr)
  - Rénald Casagraude <`rcasagraude\#at\#interfaces.fr`>
  - Jean-Pierre Delange <`adeimantos\#at\#free.fr`>
  - Daniel Desages <`daniel\#at\#desages.com`>
- Traduction en Italien
  - Davide Di Lazzaro <`mc0315\#at\#mclink.it`> (leader : it)
- Traduction en Portugais (Brésil)
  - Paulo Rogério Ormenese <`pormenese\#at\#uol.com.br`> (leader : pt-br)
  - Andre Luis Lopes <`andrelop\#at\#ig.com.br`>

  - Marcio Roberto Teixeira `<marciotex\#at\#pop.com.br>`
  - Rildo Taveira de Oliveira `<to_rei\#at\#yahoo.com>`
  - Raphael Bittencourt Simoes Costa `<raphael-bsc\#at\#bol.com.br>`
  - Gustavo Noronha Silva `<kov\#at\#debian.org>` (coordinator)
- Traduction en Espagnol (à peu près terminée)
  - Walter Echarri `<wecharri\#at\#infovia.com.ar>` (leader : es)
  - José Carreiro `<ffx\#at\#urbanet.ch>`
- Traduction en Allemand
  - Jens Seidel `<tux-master\#at\#web.de>` (leader : de)
  - Willi Dyck `<wdyck\#at\#gmx.net>`
  - Stefan Schröder `<stefan\#at\#fkp.uni-hannover.de>`
  - Agon S. Buchholz `<asb\#at\#kefk.net>`
- Traduction en Polonais—les membres de PDDP (`http://debian.linux.org.pl`) suivants :
  - Marcin Andruszkiewicz
  - Mariusz Centka `<mariusz.centka\#at\#debian.linux.org.pl>`
  - Bartosz Feński `<fenio\#at\#debian.linux.org.pl>` (leader : pl)
  - Radosław Grzanka `<radekg\#at\#debian.linux.org.pl>`
  - Bartosz 'Xebord' Janowski
  - Jacek Lachowicz
  - Rafał Michaluk
  - Leonard Milcin, Jr.
  - Tomasz Z. Napierała `<zen\#at\#debian.linux.org.pl>`
  - Oskar Ostafin `<cx\#at\#debian.linux.org.pl>`
  - Tomasz Piękoś
  - Jacek Politowski
  - Mateusz Prichacz `<mateusz\#at\#debian.linux.org.pl>`
  - Marcin Rogowski
  - Paweł Różański
  - Mariusz Strzelecki
  - Krzysztof Ścierski
  - Przemysław Adam Śmiejek `<tristan\#at\#debian.linux.org.pl>`
  - Krzysztof Szynter
  - Mateusz Tryka `<uszek\#at\#debian.linux.org.pl>`
  - Cezary Uchto
  - Krzysztof Witkowski `<tjup\#at\#debian.linux.org.pl>`
  - Bartosz Zapałowski `<zapal\#at\#debian.linux.org.pl>`
- Traduction en Chinois (simplifié)
  - Hao "Lyoo" LIU `<iamlyoo\#at\#163.net>` (leader : zh-cn)
  - Ming Hua `<minghua\#at\#rice.edu>`
  - Xiao Sheng Wen `<atzlinux\#at\#163.com>` (leader : zh-cn)
  - Haifeng Chen `<optical.dlz\#at\#gmail.com>`
  - Xie Yanbo `<xieyanbo\#at\#gmail.com>`
  - easthero `<easthero\#at\#gmail.com>`
- Traduction en Chinois (traditionnel)
  - Asho Yeh `<asho\#at\#debian.org.tw>` (leader : zh-tw)

    - Tang Wei Ching <`wctang\#at\#csie.nctu.edu.tw`> (ex-leader : zh-tw)
- Traduction en Japonais
    - Shinichi Tsunoda <`tsuno\#at\#ngy.1st.ne.jp`> (leader : ja)
    - Osamu Aoki <`osamu\#at\#debian.org`>

QREF est un raccourci pour le document original « Quick Reference… » et aussi le nom du projet sur qref.sourceforge.net.

La plus grande partie du contenu de 'Notions fondamentales sur Debian' page 5 vient de la « FAQ Debian » (mars 2002)
- 5. Les archives FTP Debian `ftparchives.sgml` (chapitre entier)
- 6. Bases du système de gestion des paquets Debian `pkg_basics.sgml` (chapitre entier)
- 7. Outils de gestion des paquets Debian `pkgtools.sgml` (chapitre entier)
- 8. Maintenir votre système à jour `uptodate.sgml` (chapitre entier)
- 9. Debian et le noyau `kernel.sgml` (chapitre entier)
- 10. Personnaliser votre installation de Debian GNU/Linux `customizing.sgml` (part of chapter)

Ces sections de la « FAQ Debian » ont été incluses dans ce document après des réorganisations majeures reflétant des changements récents dans le système Debian. Les deux documents sont mis à jour parallèlement maintenant.

La « FAQ Debian » originale a été écrite et maintenue par J.H.M. Dassen (Ray) et Chuck Stickelman. Les auteurs de la « FAQ Debian » réécrite sont Susan G. Kleinmann et Sven Rudolph. Après eux, la « FAQ Debian » a été maintenue par Santiago Vila. Le responsable actuel est Josip Rodin.

Une partie de l'information utilisée pour la « FAQ Debian » venait de
- L'annonce de sortie de Debian-1.1, par Bruce Perens (`http://www.perens.com/`).
- La FAQ Linux, par Ian Jackson (`http://www.chiark.greenend.org.uk/~ijackson/`).
- Archives de la liste de diffusion Debian (`http://lists.debian.org/`),
- Le manuel des développeurs dpkg et la Charte Debian (voir 'Références' page 233)
- beaucoup de développeurs, volontaires, beta testeurs, et
- la mémoire embrumée de ses auteurs. :-)

Les auteurs souhaitent remercier tous ceux qui ont aidé à la création de ce document.

A.2 Garanties

Comme je ne suis pas un expert, je ne prétends pas tout connaître sur Debian ou Linux en général. Les considérations de sécurité que j'utilise peuvent être utilisées seulement dans le cas d'une utilisation personnelle.

Ce document ne remplace aucun guide officiel.

Toute garantie est désavouée. Toutes les marques de commerce sont la propriété de leurs propriétaires respectifs.

A.3 Retour

Les commentaires et additions à ce document sont toujours les bienvenus. Merci d'envoyer un courriel au système Debian BTS (`http://bugs.debian.org/`) sous le paquet `debian-reference` ou le paquet correspondant à la traduction. L'utilisation de `reportbug` rend facile de remplir un rapport de bogue complet. Vous pouvez toujours envoyer un courriel à Osamu Aoki (`http://people.debian.org/~osamu/`) `<osamu\#at\#debian.org>` en Anglais ou à chacun des traducteurs dans leur langue respective.

Bien que je vive aux États-Unis, l'Anglais n'est pas ma langue maternelle. Toute correction grammaticale est la bienvenue.

Le meilleur retour est un patch pour le SGML, mais un diff de la version texte est aussi bienvenu. Voir 'Document officiel' page 1 pour le site du document officiel.

Les fichiers SGML originaux utilisés pour créer ce document sont aussi disponibles par CVS à `:pserver:anonymous@cvs.sf.net/cvsroot/qref` ou `http://qref.sourceforge.net/Debian/qref.tar.gz`.

A.4 Format du document

Ce document a été écrit en utilisant la DTD SGML DebianDoc (réécrite à partir de LinuxDoc SGML). Le système SGML DebianDoc permet d'obtenir des fichiers dans de nombreux formats à partir d'une seule source, par exemple ce document peut être visualisé en HTML, texte, TeX DVI, PostScript, PDF, ou GNU info.

Les utilitaires de conversion pour DebianDoc SGML sont disponibles dans le paquet Debian `debiandoc-sgml`.

A.5 Le labyrinthe de Debian

Le système Linux est une plateforme informatique très performante lorsqu'elle est utilisée en réseau. Cependant, connaître ses possibilités et l'utiliser n'est pas si facile. Configurer une imprimante en est un bon exemple.

Il existe une carte très détaillée appelée « CODE SOURCE ». Elle est très précise mais très dure à comprendre. Il existe aussi des références appelées HOWTO et mini-HOWTO. Elles sont plus faciles à comprendre mais ont tendance à trop se concentrer sur des détails et perdent de vue les aspects généraux. J'ai de temps en temps des problèmes à trouver la bonne section dans un long HOWTO quand j'ai besoin de quelques commandes à exécuter.

Afin de m'y retrouver dans ce labyrinthe d'information sur la configuration d'un système Linux, je m'étais fait des mémos dans un fichier texte pour une référence rapide. Ces mémos ont grossi et dans le même temps j'avais appris debiandoc. Le résultat est ce *Guide de référence pour Debian*.

A.6 Les citations Debian

Voici quelques citations intéressantes venant de la liste de diffusion Debian.

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