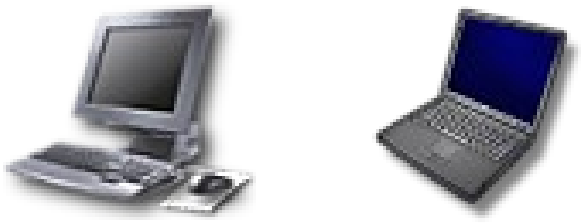
Linux et le système Debian

Tutoriel



Pascal MIETLICKI

Table des matières

Partitionnement debian

Le partitionnement assisté

"tout dans une partition" ne fera que deux partitions : une pour la racine (/) contenant tout le système et une autre pour le swap.
« Partition séparée pour répertoires personnels » : trois partitions : une racine (/), une pour les fichiers utilisateurs (/home) et une autre pour le swap.
« système multi-utilisateurs » : configuration typique pour les serveurs avec de nombreuses partitions : une pour la racine (/), les comptes utilisateurs (/home), une pour les applications (/usr), pour les données des logiciels serveurs (/var) et pour les fichiers temporaires (/tmp) sans oublier le fichier d' échange (swap). Ce type de division a plusieurs avantages. Les utilisateurs ne pourront pas bloquer le serveur en consommant tout l'espace disque disponible (ils ne pourront remplir que /tmp et /home). Les données des démons (et notamment les logs) ne pourront pas non plus bloquer le reste du système.

```
Exemple : partitionnement d'un disque de 1 Go
n°1 primaire 70  MB ext3 /
n°5 logique 500 MB ext3 /usr
n°6 logique  70  MB ext3 /var
n°7 logique  20  MB ext3 /tmp
n°8 logique 300 MB ext3 /home
n°9 logique  64  MB swap swap
```

Application d'un correctif (ou patch)
L'utilitaire créant le patch s'appelle diff et s'utilise comme suit:
diif -u file.old file.new > file.patch
le fichier file.patch contient alors les instructions permettant de transformer le contenu file.old en celui de file.new, on peut le transmettre pour patcher file.old :
patch -p0 file.old <file.patch

Aptitude
Chaque paquet peut être marqué à installer (touche « + ») ou à supprimer (touche « - »). Toutes ces opérations seront effectuées simultanément après confirmation par appui sur la touche « g » (comme go ou allez!). Avant aptitude, le programme standard pour sélectionner les paquets à installer était dselect, ancienne interface graphique associée à dpkg. Difficile d'emploi pour les débutants, il est donc déconseillé.

Dpkg, apt et tar

dpkg est le programme qui permet de manipuler les fichiers .deb, notamment de les extraires, analyser, décompacter, etc.
APT est un ensemble logiciel pour effectuer des modification globales sur le système : installation ou suppression d'un paquet gérant les dépendances, mise à jour du système, consultation des paquets disponibles, etc.
Ar permet de manipuler les archives.
ar t archive donne la liste des fichiers contenus dans l'archive, ar x archive extrait les fichiers de l'archive dans le répertoire courant.
ar d archive fichier supprime un fichier de l'archive.
Ar est un outil très rudimentaire qu'un administrateur n'emploiera qu'à de rares occasion. A la place, on se sert régulièrement de **tar**, programme de gestion d'archives et de fichiers plus évolué.

```
Exemple : suppression de dpkg par erreur, on ne peut plus installer de
paquets Debian!
On télécharge le fichier .deb du paquet dpkg et on l'installe
manuellement
ar x dpkg_1.10.18_i386.deb
tar -C / -p -zxf data.tar.gz

Examinons le contenu d'un fichier .deb :
ar t dpkg_1.10.18_i386.deb
debian-binary
control.tar.gz
data.tar.gz
ar x dpkg_1.10.18_i386.deb
ls
control.tar.gz data.tar.gz debian-binary dpkg_1.10.18_i386.deb
tar ztf data.tar.gz | head
./
./usr/
./usr/share
./usr/share/doc/
./usr/share/doc/dpkg/
./usr/share/doc/dpkg/THANKS.gz
./usr/share/doc/dpkg/TODO.gz
./usr/share/doc/dpkg/changelog.gz
./usr/share/doc/dpkg/dpkg.cfg
./usr/share/doc/dpkg/pseudo-tags.gz
tar ztf control.tar.gz
./
./conffiles
./preinst
./prerm
./postinst
./postrm
./control
cat debian-binary
2.0
```

L'archive ar d'un paquet Debian est constituée de trois fichiers :

- debian-binary. Il s'agit d'un fichier texte ne renfermant que le numéro de version du format .deb

employé

- control.tar.gz. Ce fichier d'archive rassemble les divers méta-informations disponibles. Les outils de gestion des paquets y trouvent, entre autres, le nom et la version de l'ensemble abrité, les dépendances, les conflits, etc. Certaines de ces méta-info leur permettent de déterminer s'il est ou non possible de l'installer ou le désinstaller, par exemple en fonction de la liste des paquets déjà présents sur la machine.
- data.tar.gz. Cette archive contient tous les fichiers à extraire du paquet, c'est là que sont stockés les exécutables, la documentation, etc.

Méta Informations

fichier control (similaire à un entête de mail)

```
apt-cache show apt
Package: apt
Priority: important
Section: base
Installed-Size: 3996
Maintainer: APT Development Team <deity@lists.debian.org>
Architecture: amd64
Version: 0.6.40.1ubuntu9
Replaces: libapt-pkg-doc (<< 0.3.7), libapt-pkg-dev (<< 0.3.7)
Provides: libapt-pkg-libc6.3-6-3.10
Depends: libc6 (>= 2.3.4-1), libgcc1 (>= 1:4.0.1), libstdc++6 (>= 4.0.1)
Suggests: aptitude | synaptic | gnome-apt | wajig, dpkg-dev, apt-doc,
bzip2, gnupg
Filename: pool/main/a/apt/apt_0.6.40.1ubuntu9_amd64.deb
Size: 1310040
MD5sum: 8685579f4dde68a91996fc6506e7e4ca
Description: Advanced front-end for dpkg
 This is Debian's next generation front-end for the dpkg package manager.
 It provides the apt-get utility and APT dselect method that provides a
 simpler, safer way to install and upgrade packages.
 .
 APT features complete installation ordering, multiple source capability
 and several other unique features, see the Users Guide in apt-doc.
Bugs: mailto:ubuntu-users@lists.ubuntu.com
Origin: Ubuntu
```

Champ Dépends : dépendances du paquets. Liste de conditions à remplir pour que le paquet fonctionne correctement, informations utilisés par des outils comme apt. La virgule sert de séparateur, son sens est un « et » logique. Le « ou » logique est |. Ainsi (A ou B) et C s'ecrit A | B, C. En revanche, A ou (B et C) s'écrit A | B, A | C puisque ce champ ne dispose pas de parenthèses pour changer les priorités de traitement.
Champ Recommends sont des dépendances « recommandées » mais pas nécessairement obligatoire et champ suggests sont des dépendances « suggérées ».
Champ conflicts : paquet rentrant en conflit avec un autre programme.
Champ Provides : paquet « virtuel » auquel on associe un service ou qui en remplace complètement un autre. Il s'agit juste d'un moyen d'identifier des paquets réels sur la base d'un critère logique commun.

Exemple : postfix ou sendmail déclarent « fournir » le paquet virtuel mail-transport-agent. Ainsi tout paquet qui a besoin de ce service pour fonctionner (comme un gestionnaire de liste de diffusion tel que major-domo, smartlist, sympa) se contentera dans ses dépendances de déclarer mail-transport-agent au lieu d'y préciser une grande liste de choix toujours incomplète (postfix | sendmail | exim...).

Champ Replaces indique que le paquet contient des fichiers également présents dans un autre paquet mais qu'il a le droit de les remplacer. (On peut aussi forcer la main de dpkg avec --force-overwrite).

Scripts de configuration

En plus du fichier control, l'archive control.tar.gz de chaque paquet debian peut contenir un certain nombre de scripts appelés par dpkg à différentes étapes du traitement du paquet.
Ces scripts concernent notamment l'installation, la mise à jour ou la suppression du paquet.
Purge complète du paquet : **dpkg --purge** ou **dpkg -P** ou **apt-get remove --purge**, les fichiers de configuration sont supprimées ainsi qu'un certain nombre de copies et de fichiers temporaires.
- **control.tar.gz** contient aussi une somme de contrôle md5sums représentant la liste des empreintes numériques de tous les fichiers du paquet (qu'on peut vérifier grâce à debsums).
- **conffiles** liste les fichiers du paquet à gérer comme des fichiers de configuration. Si les fichiers de configuration ont évolué entre les deux versions, dpkg va demander lequel choisir. Si on conserve l'ancienne, une sauvegarde .dpkg-new est crée ou .dpkg-old sinon.

Eviter les questions sur les fichiers de configuration

pour une mise à jour non interactive avec **dpkg** :
--force-confold conserve l'ancienne version
--force-confnew utilise la nouvelle version
--force-confdef fait le choix automatique quand c'est possible (lorsque le fichier original n'a pas été modifié)
Options à passer à **dpkg** avec **apt-get** :

```
apt-get -o DPkg::Options::=''--force-confdef'' -o DPkg::Options::=''--
force-confold'' dist-upgrade
```

On peut placer ces options directement dans la configuration du programme apt plutôt que de les lui spécifier à chaque fois en ligne de commande. Il suffit d'écrire la ligne suivante dans **/etc/apt/apt.conf.d/local** :

```
DPkg::Options {''--force-confdef''; ''--force-confold''; }
```

Intégrer cette option permettra d'en profiter dans tous les programmes graphiques gérant apt (synaptic, aptitude...)

Outil debsums et ses limites

Les fichiers md5sums sont stockés sur le disque dur, un intrus consciencieux aura pris le soin de modifier ces fichiers pour leur faire refléter les nouvelles sommes de contrôle. On peut contourner cet inconvénient en utilisant directement un paquet .deb mais il faut au préalable le télécharger :

```
apt-get --reinstall -d install `debsums -l`
debsums -p /var/cache/apt/archives -g
```

Si l'on veut effectuer la vérification à partir d'un mirroir disposant d'un .deb intégre :

```
apt-get --reinstall -d install `grep-status -e 'Status : install ok
installed' -n -s Package`
debsums -p /var/cache/apt/archives --generate=all
```

debsums n'est pas le seul détecteur de modifications. Le programme AIDE, par exemple, détecte de manière fiable toute modification inhabituelle par rapport à une image du système préalablement enregistrée et validée.

N.B : il arrive que le système soit endommagé suite à la suppression ou à la modification de fichiers appartenant à un paquet. Le moyen le plus simple de le récupérer est de le réinstaller. Malheureusement, apt considère qu'il est déjà installé et refuse de s'éxecuter; l'option --reinstall de la commande apt-get permet de contourner cela.

```
Exemple :  réinstaller postfix
apt-get --reinstall install postfix
```

Structure d'un paquet source

Généralement constitué de **3 paquets** : **.dsc, .orig.tar.gz** et un **diff.gz.**
.dsc (debian source control ou contrôle des sources de debian) est un court fichier texte contenant un en-tête RFC 822 (tout comme le fichier control) qui décrit le paquet source et indique quels autres fichiers en font partie.
Le paquet source compte lui aussi des dépendances (Build-Depends) totalement distinctes de celles des paquets binaires puisqu'il s'agit des outils nécessaires pour le compiler. Il n'y a pas forcément correspondance entre le nom du paquet source et le nom du ou des paquets binaires qu'il génére puisque chaque paquet source peut générer plusieurs paquets binaires. Le fichier .dsc dispose des champs Source et Binary pour nommer explicitement le paquet source et stocker la liste des paquets binaires qu'il génére.
orig.tar.gz est une archive contenant les codes sources du programme tel que fournit par l'auteur. Il ne faut pas modifier cette archive afin de vérifier facilement la provenance et l'intégrité du fichier (par simple comparaison du md5sum) et par respect pour la volonté de certains auteurs.
diff.gz contient l'ensemble des modifications apportées par le mainteneur Debian, notamment l'ajout d'un répertoire debian contenant les instructions à exécuter pour construire le paquet.

Décompresser un paquet source

```
dpkg-source -x paquet_0.7-1.dsc
```

Télécharger un paquet source (disposer de la ligne deb-src adéquate dans sources.list)

```
apt-get source paquet
```

Lorsqu'une nouvelle version d'un paquet arrive, c'est le paquet source le plus important, c'est lui qui est utilisé par tout un réseau de machines pour compilation sur différentes architecture (processus automatique exécuté par un ensemble de serveurs).

Manipuler des paquets

dpkg ou apt-get?
dpkg est un outil système (de backend), il n'a aucune connaissance de tous les paquets disponible mais est utile pour installer ou analyser les fichiers .deb. **Apt-get** est un outil plus proche de l'utilisateur qui permet de dépasser les limitations du précédent. Ces deux outils marchent de concert, chacun a ses spécificités et convient mieux à certaines taches.

Installer un paquet déjà téléchargé

```
dpkg -i ou --install paquet
```

Dépaquetage

```
dpkg --unpack paquet.deb
```

Configuration

```
dpkg --configure paquet
```

Une erreur fréquente est la collision de fichiers. Lorsqu'un paquet contient un fichier déjà installé, dpkg refusera de l'installer:

```
dpkg : erreur de traitement de /var/cache/apt/archives/kdepim-libs_4.1.1-
0woody1_i386.deb (--unpack) :
tentative de remplacement de « /usr/lib/libkcal1.so.2.0.0 » qui
appartient aussi au paquet libkcal2
```

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

Suppression de paquet

dpkg -r ou --remove paquet
Cette suppression n'est pas complète, tous les fichiers de configuration, scripts, logs et données utilisateurs subsistent. Ceci dans l'intérêt d'avoir la possibilité de remettre en service le programme rapidement si besoin est. Pour tout supprimer pour de bon :
dpkg -P ou --purge paquet
Exemple : suppression puis purge du paquet debian-cd
dpkg -r debian-cd
dpkg -P debian-cd
Autres fonctionnalités de dpkg
Un certain nombre d'options permettent d'interroger sa base de données interne afin d'obtenir des informations :
--listfiles (ou -L) paquet qui affiche la liste des fichiers installés par le paquet
--search (ou -S) paquet qui retrouve le paquet d'où provient ce fichier
--status (ou -s) paquet qui affiche les en-têtes d'un paquet installé
--list (ou -l) qui affiche la liste des paquets connus du système ainsi que leur état d'installation
--contents (ou -c) fichier.deb qui affiche la liste des fichiers contenus dans le paquet Debian spécifié
--info (ou -I) fichier.deb qui affiche les en-têtes de ce paquet Debian

Exemple:
dpkg -L base-passwd
```
/.
/usr
/usr/sbin
/usr/sbin/update-passwd
/usr/share
/usr/share/man
/usr/share/man/man8
/usr/share/man/man8/update-passwd.8.gz
/usr/share/man/pl
/usr/share/man/pl/man8
/usr/share/man/pl/man8/update-passwd.8.gz
/usr/share/base-passwd
/usr/share/base-passwd/passwd.master
/usr/share/base-passwd/group.master
/usr/share/doc
/usr/share/doc/base-passwd
/usr/share/doc/base-passwd/README
/usr/share/doc/base-passwd/changelog.gz
/usr/share/doc/base-passwd/copyright
/usr/share/doc/base-passwd/users-and-groups.html
/usr/share/doc/base-passwd/users-and-groups.txt.gz
```

dpkg -S /bin/date
```
coreutils: /bin/date
```

dpkg -s coreutils
```
Package: coreutils
Essential: yes
Status: install ok installed
Priority: required
Section: base
Installed-Size: 8036
Maintainer: Michael Stone <mstone@debian.org>
Architecture: amd64
```

```
Version: 5.2.1-2ubuntu2
Replaces: textutils, shellutils, fileutils, stat, debianutils (<= 2.3.1)
Provides: textutils, shellutils, fileutils
Pre-Depends: libacl1 (>= 2.2.11-1), libc6 (>= 2.3.4-1)
Conflicts: stat
Description: The GNU core utilities
 This package contains the essential basic system utilities.
 .
 Specifically, this package includes:
 basename cat chgrp chmod chown chroot cksum comm cp csplit cut date dd
df dir dircolors dirname du echo env expand expr factor false fmt fold
groups head hostid id install join link ln logname ls md5sum mkdir mkfifo
mknod mv nice nl nohup od paste pathchk pinky pr printenv printf ptx pwd
readlink rm rmdir sha1sum seq shred sleep sort split stat stty sum sync
tac tail tee test touch tr true tsort tty uname unexpand uniq unlink
users vdir wc who whoami yes
```

dpkg -l 'b*' | head

```
Souhait=inconnU/Installé/suppRimé/Purgé/H=à garder
| État=Non/Installé/fichier-Config/dépaqUeté/échec-conFig/H=semi-installé
|/ Err?=(aucune)/H=à garder/besoin Réinstallation/X=les deux (État,Err:
majuscule=mauvais)
||/ Nom                  Version            Description
+++-====================-==================-
=============================================
un  baekmuk-ttf          <néant>            (aucune description n'est
disponible)
un  base                 <néant>            (aucune description n'est
disponible)
ii  base-config          2.67ubuntu19       Debian base system
configurator
ii  base-files           3.1.5ubuntu4       Debian base system
miscellaneous files
ii  base-passwd          3.5.10             Debian base system master
password and group
```

dpkg -c /var/cache/apt/archives/cvs_1.12.9-13ubuntu1_amd64.deb | head

```
drwxr-xr-x root/root         0 2005-09-30 07:13:39 ./
drwxr-xr-x root/root         0 2005-09-30 07:13:38 ./etc/
drwxr-xr-x root/root         0 2005-09-30 07:13:38 ./etc/pam.d/
-rw-r--r-- root/root       267 2005-09-30 07:13:38 ./etc/pam.d/cvs
drwxr-xr-x root/root         0 2005-09-30 07:13:38 ./etc/cron.weekly/
-rwxr-xr-x root/root      1370 2005-09-30 07:13:38 ./etc/cron.weekly/cvs
drwxr-xr-x root/root         0 2005-09-30 07:13:38 ./usr/
drwxr-xr-x root/root         0 2005-09-30 07:13:40 ./usr/bin/
-rwxr-xr-x root/root    594248 2005-09-30 07:13:40 ./usr/bin/cvs
-rwxr-xr-x root/root     19679 2005-09-30 07:13:38 ./usr/bin/rcs2log
drwxr-xr-x root/root         0 2005-09-30 07:13:38 ./usr/sbin/
-rwxr-xr-x root/root       695 2005-09-30 07:12:45 ./usr/sbin/cvs-pserver
drwxr-xr-x root/root         0 2005-09-30 07:13:38 ./usr/share/
drwxr-xr-x root/root         0 2005-09-30 07:13:38 ./usr/share/doc/
drwxr-xr-x root/root         0 2005-09-30 07:13:40 ./usr/share/doc/cvs/
drwxr-xr-x root/root         0 2005-09-30 07:13:38
./usr/share/doc/cvs/html-info/
-rw-r--r-- root/root      8124 2005-09-30 07:13:37
./usr/share/doc/cvs/html-info/cvs.html
```

```
dpkg -I  /var/cache/apt/archives/cvs_1.12.9-13ubuntu1_amd64.deb
 nouveau paquet Debian, version 2.0.
 taille 1467302 octets : archive de contrôle= 27478 octets.
      36 octets,     2 lignes      conffiles
    8961 octets,   368 lignes   *  config               #!/bin/sh
    1063 octets,    24 lignes      control
    8098 octets,   106 lignes      md5sums
    5566 octets,   165 lignes   *  postinst             #!/bin/sh
     484 octets,    18 lignes   *  postrm               #!/bin/sh
     103 octets,     9 lignes   *  preinst              #!/bin/sh
     782 octets,    24 lignes   *  prerm                #!/bin/sh
   54276 octets,   975 lignes      templates
 Package: cvs
 Version: 1:1.12.9-13ubuntu1
 Section: devel
 Priority: optional
 Architecture: amd64
 Depends: libc6 (>= 2.3.4-1), libpam0g (>= 0.76), zlib1g (>= 1:1.2.1),
debconf (>= 0.5.00), libpam-runtime (>= 0.76-14)
 Recommends: netbase (>= 2.08-1), info | info-browser
 Conflicts: cvs-doc, cvs2cl (<< 2.55-1)
 Replaces: cvs-doc (<< 1.11-2)
 Provides: cvs-doc
 Installed-Size: 3056
 Maintainer: Steve McIntyre <93sam@debian.org>
 Description: Concurrent Versions System
  CVS is a version control system, which allows you to keep old versions
  of files (usually source code), keep a log of who, when, and why
  changes occurred, etc., like RCS or SCCS.  Unlike the simpler systems,
  CVS does not just operate on one file at a time or one directory at
  a time, but operates on hierarchical collections of directories
  consisting of version controlled files.
  .
  CVS helps to manage releases and to control the concurrent editing of
  source files among multiple authors.  CVS allows triggers to
  enable/log/control various operations and works well over a wide area
  network.
```

Comparaison de versions

Option **--compare-versions** qui requiert trois paramètres : un numéro de version, un opérateur de comparaison (lt (<), le(<=), eq(=), ne(!=), ge(>=), gt(>)) et un deuxième numéro de version.
Si la comparaison est vraie, code de retour 0 (succés) sinon valeur non nulle (échec)

```
Exemple :
dpkg --compare-versions 1.2-3 gt 1.1-4
echo $?
```

Cohabitation avec d'autres systèmes de paquetages

on peut installer un paquet rpm sur un système debian grâce à la commande alien. L'utilitaire alien permet de convertir des paquets RPM en paquets debian et vice versa.
fakeroot alien --to-deb phpMyAdmin-2.0.5-1.noarch.rpm génére un fichier **phpmyadmin_2.0.5-2_all.deb**

liste de sources de paquets dans **/etc/apt/sources.list** indiquant les différents emplacements disponibles des paquets Debian. APT devra ensuite rapatrier la liste ainsi que l'en-tête de ces sources. Il réalise cette opération en téléchargeant les fichiers Packages.gz ou Packages.bz2 (cas d'une source de paquets binaires) et Sources.gz ou Sources.bz2 (Cas d'une source de paquets sources) et en analysant leur contenu.
Le fichier **sources.list** contient sur chaque ligne active une description de source qui se décompose en 3 parties séparées par des blancs.
Le premier champ indique le type de la source:
« deb » pour des paquets binaires
« deb-src » pour des paquets sources
Le deuxième champ indique l'URL de base de la source. Il s'agit d'un mirroir debian ou de toute autre archive de paquet mise en place par des tierces personnes. L'URL peut débuter par **file://** pour indiquer une source locale située dans l'arborescence de fichiers, **http://** pour indiquer une source accessible depuis un serveur web, ou encore par **ftp://** pour une source disponible sur un serveur FTP.
Le dernier champ a une syntaxe variable selon que la source correspond à un miroir Debian ou non. Dans le cas d'un mirroir Debian, on nomme la distribution choisie (stable contenant des paquets éprouvés, sûrs pour les serveurs, testing contenant des paquets éprouvés d'une dizaine de jours de la version unstable : c'est un bon compromis entre un système à jour et sécurisé, unstable : pour tester des paquets en développement : si vous voulez vous investir dans le projet Debian en indiquant les différents bogue, utilisez cette distribution) puis les différentes sections souhaitées (main, contrib, non-free). Dans les autres cas, on indique simplement le sous-répertoire de la source désirée (souvent un simple ./ dénotant l'absence de répertoire).

```
Exemple : un sources.list standard pour distribution stable (à remplacer
par testing ou unstable selon la distribution voulue)
#Mises à jour de sécurité
deb http://security.debian.org/ stable/updates main contrib non-free
#Mirroir Debian
deb http://ftp.fr.debian.org/debian stable main contrib non-free
deb-src http://ftp.fr.debian.org/debian stable main contrib non-free
A l'université, on passe par un proxy apt sur chouchen
#Mises à jour de sécurité
deb http://chouchen.upf.pf:9995/security stable/updates main contrib non-
free
#Mirroir Debian
deb http://chouchen.upf.pf:9995/debian stable main contrib non-free
deb-src http://chouchen.upf.pf:9995/debian stable main contrib non-free
```

Pour utiliser des sources situées sur un **CD-ROM**, on ajoutera les entrées à l'aide du programme **apt-cdrom add**. Ce dernier va alors parcourir le cdrom dans le lecteur à la recherche de fichiers Packages (qu'il ajoutera à sa base de données). Dès lors, apt-get pourra vous demander d'insérer le CD-ROM en question si besoin est.

Les archives main, contrib et non-free

Main (archive principale) rassemble tous les paquets répondant pleinement aux principes du logiciel libre selon Debian
Non-free (non libre), spéciale, contient des logiciels ne répondant pas (totalement) à ces principes mais néanmoins distribuables librement.

Contrib (contributions) est un stock de logiciels libres ne fonctionnant pas sans certains éléments non libres. Il peut s'agir de programmes dépendant de logiciels de la section non-free.

La distribution parralèle : experimental

En amont de **unstable**, elle est réservée aux utilisateurs expérimentés capables de gérer les soucis importants. Elle peut permettre ponctuellement de récupérer un paquet que l'on tient à essayer ou utiliser. La ligne qu'il convient d'ajouter dans **sources.list** est la suivante :

```
deb http://ftp.fr.debian.org/debian ../project/ experimental main contrib non-free
```

Les sources non officielles : **www.apt-get.org** et **mentors.debian.net**

Commande apt-get

Un préalable à tout travail APT est la mise à jour de la liste des paquets disponibles avec un **apt-get update** qui télécharge un certain nombre de fichiers (Packages.gz|bz2, Sources.gz|bz2) devenu assez volumineux au fil de la croissance de Debian (3Mo pour le plus Packages.gz correspondant à la section main).

Installation et suppression

APT permet d'ajouter ou de supprimer des paquets avec **apt-get install paquet** et **apt-get remove paquet**, l'option **--purge** demande une désinstallation complète.

Installer la même sélection de paquets plusieurs fois

On veut parfois installer systématiquement la même liste de paquets sur plusieurs ordinateurs.
Récupérons d'abord cette liste sur l'ordinateur « modèle »

```
dpkg --get-selections > liste-pkg
```

Il faut alors transférer liste-pkg sur les autres ordinateurs puis :

```
dpkg --set-selections < liste-pkg
apt-get dselect-upgrade
```

La première commande enregistre les voeux de paquets à installer que l'invocation d'apt-get les exauce.

Supprimer et installer en même temps

Pour apt-get install, ajoutez un « - » aux paquets à supprimer

```
apt-get install paquet1 paquet2-
```

Pour apt-get remove, ajoutez un « + » aux paquets que vous souhaitez installer

```
apt-get remove paquet1+ paquet2
```

Installation d'une version différente d'un paquet

Si le sources.list mentionne plusieurs distributions, il est possible de préciser la version du paquet à installer. On peut demander un numéro de version précise avec :

```
apt-get install paquet=version
```

Ou indiquer la distribution d'origine du paquet (stable, testing ou unstable) avec :

```
apt-get install paquet/distribution
```

Exemple : installation de la version unstable de SpamAssassin

```
apt-get install spamassassin/unstable
```

ou

```
apt-get -t unstable install spamassassin
```

Gérer le cache .deb

apt-get conserve dans **/var/cache/apt/archives/** une copie de chaque fichier .deb téléchargé. Ce répertoire peut rapidement occupé beaucoup d'espace disque. Il convient donc d'y faire régulièrement le tri :
apt-get clean vide le répertoire
apt-get autoclean ne supprime que les paquets qui, n'étant plus téléchargeable (disparu du mirroir Debian) sont clairement inutiles. Le paramètre **APT::Clean-Installed** permet d'empêcher la suppression de fichiers .deb encore installés.

Mise à jour

Recommandé pour les derniers correctifs de sécurité. On utilise **apt-get upgrade** (précéde de apt-get update evidemment). Cette commande cherche à mettre à jour les paquets selon le numéro de version. Si vous avez mentionné testing ou unstable dans votre sources.list, apt-get upgrade migrera tout votre système stable vers testing ou unstable, ce qui n'est peut être pas l'effet recherché!
Pour indiquer quelle distribution utilisé, il faut l'indiquer à apt-get avec l'option **-t ou --target-release** (version cible) ou **--default-release** (version par défaut), suivie du nom de distribution en question (**apt-get -t stable upgrade**). Pour éviter de la spécifier à chaque invocation, on peut ajouter dans **/etc/apt/apt.conf.d/local : APT::Default-Release « stable »**.
Pour les mises à jour plus importantes, comme lors du basculement d'une version majeure de Debian à la suivante ou dans le cadre de la distribution unstable, il faut utiliser **apt-get dist-upgrade**. Cela effectue la mise à jour même s'il y a des paquets obsolètes à supprimer et de nouvelles dépendances à installer.

Options de configuration

Plusieurs options sont possibles dans **/etc/apt/apt.conf.d/local**, comme ignorer les erreurs de collision de fichiers :

```
DPkg::Options { ''--force-overwrite''; }
```

serveur proxy :

```
http : Acquire::http::proxy ''http://monproxy:3128''.
```

Gérer les priorités associées aux paquets

Chaque version de **paquet déjà installé** a une priorité de **100**, une version **non installée** a une priorité de **500 sauf** si elle fait partie de la **distribution cible** (Target Release), elle a alors une priorité de **990**. On peut **modifier ces priorités** en intervenant sur le fichier **/etc/apt/preferences** pour y ajouter des entrées décrivant le nom du ou des paquets, leur version, leur origine et leur nouvelle priorité. APT **refusera** toujours **d'installer** une **version antérieure** d'un paquet sauf si priorité supérieur à **1000**. APT installera toujours la version de priorité la plus élevée. Si deux versions ont la même priorité, APT installe la plus récente (de numéro de version le plus grand). Si deux paquets de **même version** ont la **même priorité** mais **diffère** dans leur contenu, celui qui n'est **pas installé** sera **privilégié** (cas d'une mise à jour).
Si l'on souhaite utiliser exclusivement des versions stable de Debian sans installer ceux des autres versions sauf demande explicite.
Dans le fichier **/etc/apt/preferences** :

```
Package: *
Pin: release a=stable
Pin-Priority: 900

Package: *
Pin: release o=Debian
```

```
Pin-Priority: -10
```

Supposons que nous diposions d'un serveur ayant installé de nombreux programmes spécifiques à **Perl 5.8** et que l'on veuille qu'**aucune mise à jour** n'en installe une autre version. On peut utiliser :

```
Package: perl
Pin: version 5.8*
Pin-Priority: 1001
```

Commentaire dans /etc/apt/preferences

```
Explanation : Le paquet xfree86-xserver contenu dans
Explanation: experimental peut être utilisé
Package: xfree86-xserver
Pin: release a=experimental
Pin-Priority: 500
```

Travailler avec plusieurs distributions

Supposons que votre **sources.list** contiennent les trois types de distributions, pour installer un paquet **testing**, il suffit de faire:

apt-get install paquet/testing si l'installation échoue parce que certaines dépendances ne sont pas satisfaites, autorisez le à les satisfaires pour testing en ajoutant le paramètre **-t testing**.

Dans cette situation, les mises à jour (« upgrade » et « dist-upgrade ») ont lieu dans le cadre de stable sauf pour les paquets mis à jour depuis une autre distribution : ces derniers suivront les dernières évolutions dans celles-là.

Pour vérifier les **priorités de traitement** des paquets, utilisez **apt-cache policy**.

```
Exemple : paquet dont la version 2 a été installé depuis testing, la
version 1 est disponible dans stable et la 3 dans unstable. La version 1
(priorité 990 - apt refuse d'installer une version antérieure d'un paquet
si priorité inférieure à 1000) est ignorée car plus petite que la version
installée. Il reste alors les versions 2 et 3, toutes deux de priorité
500. Face à ce choix, APT choisit la plus récente, la version unstable!
Pour éviter qu'un paquet testing puisse migrer vers unstable, il faut
affecter une priorité inférieure à 500 pour unstable dans
/etc/apt/preferences :
Package: *
Pin: release a=unstable
Pin-Priority: 490
```

Commande apt-cache

Le programme apt-cache permet d'effectuer des **reqêtes** sur la liste des **programmes disponibles**. Elle permet notamment de rechercher des paquets en tapant **apt-cache search mot-clé**. Pour consulter les **en-têtes** : **apt-cache show paquet** (description, dépendances, mainteneur..).

apt-cache policy pour les **priorités affectés** aux paquets, **apt-cache dumpavail** qui affiche les **en-têtes** de **toutes les versions** diponibles de **tous les paquets**, **apt-cache pkgnames** affiche une liste de tous les paquets existant dans le cache. Pour la vérification des signatures des paquets, il existe l'outil **apt-check-sigs** à faire après un apt-get update.

Mise à jour automatique

Configuration de **dpkg** pour le remplacement du fichier de configuration (avec **--force-confdef ou --force--confold**). Il reste les **interactions d'apt**, de **debconf** et celle en ligne de commande.

Pour **apt**, il suffit de préciser l'option **--assume-yes** ou **-y** qui répondra **« oui » automatiquement** à

toutes questions.
Pour debconf, on peut choisir grâce à **dpkg-reconfigure debconf**, l'interface **non-interactive** qui désactive toute interaction. Toutefois, si une note d'information importante doit être communiqué, elle sera transmise par mail.
Pour les lignes de commande, c'est plus complexe, on peut supprimer l'entrée standard avec commande **</dev/null**. Cette méthode n'est **pas fiable à 100%**, elle repose sur le fait que la plupart des scripts interprétent l'absence de réponse comme une validation de la valeur par défaut.

Script pour mise à jour non interactive

```
export DEBIAN_FRONTEND=noninteractive
yes ' ' | apt-get -y -o Dpkg::Options::=«--force-confdef» -o
Dpkg::Options::=«--force-confold» dist-upgrade
```

Mise à jour semi-automatique

```
Dans le cron : apt-get update; apt-get -y -d dist-upgrade; apt-get
autoclean;
-d : pour download only, il suffira alors d'executer la commande apt-get
dist-upgrade qui s'effectuera alors immédiatement.
```

Les sources de documentation

La commande man

Pour accéder aux pages du **manuel** d'un programme, **man nom-programme.**
Pages de manuel en **français** accessible grâce au paquet **manpages-fr** (avec **LC_ALL=fr_FR@euro** dans **/etc/environment**).
Les pages du manuel sont classées par **section**:

1. commandes exécutables depuis l'interpréteur
2. appels système (fonctions fournies par le noyau)
3. fonctions de la bibliothèques (fournies par les bibliothèques système)
4. périphériques (sous Unix, ce sont des fichiers spéciaux, habituellement placés sous /dev/)
5. fichiers de configuration (formats et conventions)
6. jeux
7. ensemble de macros et de standards
8. commandes d'administration système
9. routines du noyau

On peut donc **préciser** la **section** à sonder, pour l'**appel système read** par exemple, on tapera **man 2 read**.
Si on souhaite seulement avoir une **description courte** de la commande, on peut utiliser **whatis**.
Pour connaître quels sont les **commandes associés** à une fonction, il existe la commande **man -k** ou **apropos**. Elle renvoie une liste des pages de manuel qui mentionnent le mot clé associé.

```
Exemple : Retrouver cp avec apropos
apropos « copy file »
cp (1)               - copy files and directories
cpio (1)             - copy files to and from archives
install (1)          - copy files and set attributes
```

On peut placer les **pages** du **manuel** en **html** sur un serveur accessible à tous avec **man2html** qui sera alors accessible sur : **http://serveur/cgi-bin/man/man2html** (konqueror dispose nativement de la doc en ligne).

Format info

La commande **info** fournit **quelques avantages** mais est **plus complexe** à utiliser. Elle a une **structure hiérarchique**, appelée sans paramètre elle affiche la liste des noeuds disponibles au premier niveau (on peut avoir accés à l'aide en tapant h).

La documentation spécifique

Tout logiciel contient en général une **documentation supplémentaire**, accessible dans le répertoire **/usr/share/doc/<paquet>**. Il contient également d'**autres ressources** :
README.Debian signale toutes les **adaptations effectuées** pour être en conformité avec la charte Debian
changelog.Debian.gz qui permet de suivre **les modifications apportées** au paquet au fil du temps
NEWS.Debian.gz documentant les **changements majeurs** du programme
copyright qui contient la **licence** mais aussi généralement des **ressources externes** comme le site web officiel du programme

Ressources Web

Faites une **recherche du programme** en indiquant le **mot clé debian** dans un moteur de recherche ou essayer un **annuaire** de logiciels libres comme **http://freshmeat.net ou www.framasoft.net**
Si une erreur survient, saisissez la dans un moteur de recherche entre apostrophes doubles.

Les HOWTO (comment faire?)

Ce sont des documentations décrivant, **étape par étape**, comment atteindre un **but prédéfini**. Ces buts sont relativement variés mais souvent techniques : **mettre en place l'IP masquerading** (IP masqué), **configurer le RAID logiciel, installer un serveur Samba, etc**. Ces HOWTO sont gérés par Linux Documentation Project sur **http://www.tldp.org**
Pour les installer localement, installer les paquets **doc-linux-html** et **doc-linux-fr-html**. Les versions HTML seront alors disponibles dans **/usr/share/doc/HOWTO**. (apt-howto-fr, quick-reference-fr et debian-reference-fr sont les versions française de la doc apt et des guides de référence debian).

Commande et informations utiles

dpkg -L paquet donne la **liste** des **fichiers inclus dans le paquet**, on pourra alors rapidement identifier la documentation disponible avec ce paquet ainsi que les fichiers de configuration.
dpkg -s paquet donne les **entêtes du paquetages** et indique les paquets recommandés.
Les fichiers de configuration sont souvent commentés à tel point qu'il suffit parfois de choisir la ligne à activer parmi celles proposées.
Dans la plupart des cas, des exemples de configuration sont fournis dans le répertoire **/usr/share/doc/<paquet>/examples/**.

Surveiller l'activité des démons

Un **démon n'interagit pas** directement avec l'administrateur. Pour vérifier son fonctionnement, il est nécessaire de le **tester**. Chaque démon garde généralement des traces **logs dans /var/log**.

Le démon syslogd

syslogd est **particulier**, il collecte les **traces** qui lui sont **envoyées** par les **autres programmes**. Les **priorités de traitement** peuvent être configuré dans **/etc/syslog.conf**.
On pourra faire appel à un **utilitaire spécifique** pour analyser leurs contenus, par exemple, pour les **serveurs web : analog, awstats, webalyze**... Ils existent des utilitaires modulaires permettant d'analyser **plusieurs fichiers de logs** comme **lire**, **modlogan** ou encore **logcheck**.

Aide de la liste de diffusion

Si vous rencontrez un **problème** pour lequel vous ne trouvez pas de solutions, on peut utiliser la liste de diffusion **debian-user-french@lists.debian.org**.
On peut également rechercher dans les **archives récentes de la liste** :
http://wiki.debian.net/?DebianFrench
http://lists.debian.org/debian-user-french/
ou sur le BTS :
http://www.debian.org/Bugs/index.fr.html
Si c'est un bogue du programme, retrouver le **paquet concerné** avec **dpkg -S fichier_en_cause**, regarder le suivi de bogues (http://bugs.debian.org/paquet) et le cas échéant, signalez le à l'aide de la commande **reportbug**.

Configuration de base : réseau, comptes, impressions...

Francisation du sytème

la commande **locale** permet d'afficher un résumé de la configuration courante (format des dates, des nombres, etc)

Définir la langue par défaut

Utilisez la commande **dpkg-reconfigure locales** (sélectionnez celles débutant par **fr_FR** semble être un choix raisonnable). N'hésitez pas à sélectionner d'**autres locales** si la machine héberge des utilisateurs étrangers. Cette liste est stockée dans le fichier **/etc/locale.gen**. On peut **intervenir** sur ce fichier à la main en pensant à **exéctuer** la commande **locale-gen après** chaque **modification**.
Il faut ensuite sélectionner le jeu par défaut, choisissez **fr_FR@euro** qui a pour effet de modifier le fichier **/etc/environment** pour la variable **LANG**.

Jeux de caractères

L'encodage ISO-8859-1 (ou Latin 1) avait cours en France. Pour des raisons historiques, il n'incluait pas certains caractères (e dans l'o, y tréma majuscule, symbole euro...), c'est pourquoi **ISO-8859-15** a vu le jour.
Quel programme utilise **/etc/environment**?
Les programmes **login**, **gdm** ou encore **ssh** pour créer leur variable d'environnement. Ces applications effectuent cela via le **module PAM** (pam_env.so). **PAM (Pluggable Authentication Module ou module d'authentification centrale)** est une bibliothèque modulaire centralisant les mécanismes d'authentification, d'initialisation de sessions et de gestions des mots de passe.

Le clavier

Configurer le clavier en mode console

Le paquet **console-date** contient les différents dispositions de clavier accessibles au paquet **console-tools** pour configurer la console. La commande **dpkg-reconfigure console-data** permet de revenir à tout moment sur la disposition du clavier pour ce mode. A la première question, optez pour « Choisir un codage clavier pour votre architecture » puis répondez « azerty » puis « french » puis « With Euro (Latin 9) ».

Configurer le clavier en mode graphique

Inclut dans la configuration du serveur XFree86, répondez « xfree86 » puis « pc105 ».
Quelques options peuvent vous faciliter la vie comme **altwin:left_meta_win**, **compose rwin** qui transforme la touche windows de gauche en modificateur Meta, ce qui permet de l'employer pour des raccourcis claviers.

Configuration du réseau

Lors de l'installation, le fichier **/etc/network/interfaces** contient déjà une configuration valide. Une ligne débutant par **auto** donne la liste des interfaces à configurer automatiquement.

```
Exemple : configuration par DHCP
auto eth0
```

```
iface eth0 inet dhcp
hostname adelscott

configuration statique
auto eth0
iface eth0 inet static
address 10.1.1.228
netmask 255.255.0.0
broadcast 10.1.255.255
network 10.1.1.0
gateway 10.1.1.254
```

Connexion PPP

Une connexion par **modem** peut être configuré grâce à l'outil **pppconfig** du paquet Debian eponyme. En cas de doute sur le protocole d'authentification, choisissez **PAP**.
Après configuration, il est possible de se connecter par la commande **pon** et se déconnecter avec **poff**.
Il est possible d'effectuer des connexions à la demande grâce à l'utilitaire **diald**.

Connexion par modem ADSL

Les connexions **PPP** par **ADSL** sont par définition intermittentes. Comme elles ne sont pas facturées à la durée, la tentation est grande les garder toujours ouvertes : un moyen simple est de les faire démarrer par le processus **init**. On ajoute pour cela dans **/etc/inittab** :
`adsl:2345:respawn:/usr/sbin/pppd call dsl-provider`
La plupart des connexions ADSL subissent une déconnexion quotidienne, cette méthode permet de réduire la durée de la coupure.

Modem fonctionnant avec PPPOE

Pour la configuration, **ppoeconf** du paquet éponyme. Il modifiera **/etc/ppp/peers/dsl-provider** avec les paramètres fournis et enregistrera les informations d'**authentification** dans **/etc/ppp/pap-secrets** et **/etc/ppp/chap-secrets**.
Une fois mis en place, on **démarre la connexion** avec la commande **pon dsl-provider** et on la **stoppe** avec **poff dsl-provider**.

Modem fonctionnant avec DHCP

Ne fait pas du tout intervenir PPP. Il suffit de configurer une **connexion réseau par dhcp**. Le **modem** s'inscrit **automatiquement** comme **passerelle** par défaut et prend en charge le travail de **routage** (cad qu'il gère le trafic réseau entre l'ordinateur et l'Internet).

`/proc et /sys, systèmes de fichiers virtuels`
`Il s'agit en fait d'un` **`moyen pratique`** `de récupérer des` **`infos`** `du noyau ou de lui en communiquer.`
`/sys est prévu pour donner accès à des objets internes du noyau.`

Attribution et résolution des noms

Une adresse **IP identifie** une **interface** réseau, chaque **machine** peut en compter **plusieurs** et recevoir **plusieurs noms** dans le **DNS**.
Chaque **machine** est cependant identifié par un **nom principal** dans **/etc/hostname** et communiqué au noyau linux à travers la commande **hostname**. On peut en prendre connaissance dans le fichier virtuel **/proc/sys/kernel/hostname**.
Le **nom** de **domaine** n'est pas géré de la même manière, mais **provient** du **nom complet** de la machine,

obtenu par une résolution de noms dans le fichier **/etc/hosts**. Il suffit d'y **placer** un **nom complet** au début de la liste des noms associés :

```
127.0.0.1 localhost
10.1.1.228 adelscott.upf.pf adelscott
```

Résolution de noms

Le mécanisme de résolution sous linux est **modulaire**. Il peut s'appuyer sur différentes sources d'informations déclarées dans le fichier **/etc/nsswitch.conf**. L'entrée concernant la résolution des noms d'hôtes est **hosts**, par défaut, elle contient **files dns** ce qui signifie en **priorité /etc/hosts** puis les **serveurs DNS.** Des **serveurs NIS/NIS+** ou **LDAP** forment d'**autres sources** possibles.

Configuration des serveurs DNS

Pour accéder aux **informations DNS**, il faut disposer d'un **serveur DNS relayant** les **requêtes**. Les serveurs **DNS** à employer sont dans le fichier **/etc/resolv.conf** à raison d'un par ligne :

```
nameserver 10.1.1.253
nameserver 10.1.1.23
```

Fichier /etc/hosts

En raison de **l'absence** d'un **serveur** de **noms** local, il est possible d'**établir** une **table des correspondances** entre **IP** et **nom de machines**. La **syntaxe** est **simple**, chaque **ligne** précise une **IP suivie** de la **liste** de tous les **noms associés** (le premier étant FQDN).
Ce fichier est disponible même en cas de panne réseau mais cette **méthode** est **fastidieuse** car il faut **modifier** chacun des **fichiers** des **machines** du réseau local. Ce **fichier suffira** pour un **réseau minimal** mais **à partir de cinq machines**, il vaut mieux installer un **serveur DNS.**

Court-circuiter le DNS

/etc/hosts est **consulté avant** le **DNS**. Cela permet en cas de **changement** pas encore propagé du **DNS** de **tester** l'accés à un site web avec le nom prévu. Autre **emploi original**, il est possible de **rediriger** le **trafic destiné** à un **hôte donné** vers la **machine locale**. Les noms de serveurs dédiés à l'envoi de bannières publicitaires pourraient faire l'objet d'une telle mesure, ce qui rendrait la navigation plus fluide.

Base de données des utilisateurs et des groupes

la **liste utilisateurs** est stockée dans **/etc/passwd** et **/etc/shadow** stocke les **mots de passe chiffrés**. Pour **éditer** ces **fichiers** (en garantissant qu'il ne soit **pas modifié** par **plusieurs personnes** à la fois), on peut employer **vipw** (pour **/etc/passwd**) et **vigr** (pour **/etc/group**).L'option **-s** permet d'**éditer** le fichier **shadow** correspondant.

Liste des champs de /etc/passwd

identifiant ou **login** par exemple **miepas**
mot de passe : **chiffré** à sens unique avec **md5** ou **crypt**. La valeur **« x »** indique que le **mot de passe** est **stocké** dans **/etc/shadow**
uid: numéro unique identifiant l'utilisateur
gid: numéro unique du groupe principal (Debian crée par défaut un groupe spécifique à chacun)
GECOS: **champ** de **renseignements** (nom complet, numéro de bureau...)
répertoire de connexion, attribué à l'utilisateur ($HOME)
programme à exécuter **après connexion**. Généralement l'interpréteur de commande (donnant libre cours à l'utilisateur). Si on précise **/bin/false**, l'utilisateur ne pourra **pas se connecter**.

Le fichier des mots de passe chiffrés et cachés : /etc/shadow

Contient
identifiant(ou login)
mot de passe chiffré
plusieurs **champs de gestion** de l'**expiration** du mot de passerelle
Il est **inaccessible** en lecture **aux utilisateurs** contrairement à /etc/passwd. C'est pourquoi il convient d'utiliser /etc/shadow afin d'éviter qu'un utilisateur mal intentionné tente de casser par la méthode de force brute ou du dictionnaire le mot de passe.

Modifier un compte ou mot de passe existant

passwd change le **mot de passe**. **chfn** intervient sur le champ **GECOS**. **chsh** permet de **changer** le **shell** de login qui est limité à la liste donnée dans **/etc/shells** pour les utilisateurs.
chage donne à l'administrateur la possibilité de **modifier** les **conditions** d'**expiration** du **mot de passe** (**-l utilisateur** donnant la **configuration actuelle**). On peut **forcer** l'**expiration** du mot de passe grâce à la commande **passwd -e utilisateur** qui oblige l'utilisateur à changer de mot de passe à la prochaine connexion.

Bloquer un compte

Pour désactiver un compte, on utilise la commande **passwd -l utilisateur** (pour **lock**). La remise en **service** s'effectue avec l'option **-u**. (pour **unlock**)

Liste des groupes : /etc/group

Liste des **champs** :
identifiant (le nom du groupe)
mot de passe (facultatif) : ne sert qu'à intégrer un groupe dont on n'est pas habituellement membre (avec la commande newgrp ou sg)
gid : numéro unique identifiant le groupe
liste des membres : liste des identifiants d'utilisateurs membres du groupe, séparées par des virgules

Pour **changer** temporairement de **groupe principal** : **newgrp** qui démarre un **nouveau shell** ou **sg** qui se contente d'**exécuter** une **commande**.
la commande **id** permet de **vérifier** à tout instant son **identifiant** et ses **groupes**.
groupadd et **groupdel** permet de **créer** et de **supprimer** un **groupe**, **groupmod modifie** les informations d'un groupe (**gid**).
passwd -g groupe **modifie** le **mot de passe** d'un **groupe**, **passwd -r -g** groupe **supprime** le **mot de passe.**

Création de comptes

adduser pour créer un **nouveau compte**. Le fichier **/etc/adduser.conf** offre quelques **paramétrages intéressants**. On peut ainsi **prévoir** automatiquement un **quota** en dupliquant celui d'un **utilisateur modèle**, **modifier** l'**emplacement** du **compte** ou choisir un autre **interpréteur** de **commande** par défaut.
Le répertoire modèle est **/etc/skel** dont le **système recopie** le **contenu** afin de **fournir** quelques **fichiers** standards.
Dans certains cas, il est utile d'ajouter un utilisateur dans un groupe. Par exemple, le **groupe audio** permet à l'utilisateur d'**accéder** aux **périphériques son** :
`adduser utilisateur groupe`

`Exemple : ajouter un utilisateur au groupe principal groupe`
`adduser -g <groupe> -d /home/<classement>/<login> -c « <Prenom NOM - Fonction Département - DateArrivéé - DateDépart » -s /bin/false`
`<`**`classement`**`> représente la place dans l'arborescence, par exemple, Administration/CRI`
`DateArrivée` `et` **`DateDépart`** `au format` **`AAMMJJ`**
`passwd <login>`
`smbpasswd -a <login> # pour samba`
`vi /etc/postfix/aliases`
`on y rajoute l'entrée :` **`prénom.nom:<login>`**
`Voir les informations de` **`/etc/passwd : getent`** `consulte les` **`entrées`** `des` **`bases de données :`**
`getent passwd utilisateur`

Droits d'accès à un périphérique

Chaque périphérique est représenté par un **fichier spécial** dans **/dev/**. Il en existe **2 types** selon le périphérique : mode **caractère** et mode **bloc**.
Le mode **caractère limite** les **interactions** aux opérations de **lecture** et **d'écriture**.
Le mode **bloc** permet aussi de se **déplacer** dans le **flux de données**.
Les **droits d'accès à /dev/mixer** n'est accessible en **lecture écriture** qu'à **root** et aux membres du groupe **audio**. Seuls ces utilisateurs peuvent utiliser les périphériques audio.

Les interpréteurs de commande

Un shell de connexion est invoqué lors d'une connexion sur la console via telnet ou ssh ou à travers bash --login.
Un shell interactif est celui qui prend place dans un terminal de type xterm, un shell non interactif permet d'exécuter un script.
bash emploie les scripts d'**initialisation /etc/bash.bashrc** (interactifs) et **/etc/profile** (connexion).
bash gère la **complétion automatique** qui permet de compléter un nom de commande ou d'argument à activer dans **/etc/bash.bashrc**. En plus des fichiers communs, chaque utilisateur peut se créer un

fichier **~/.bashrc** et **~/.bash_profile** pour **personnaliser** son shell. Les ajouts les plus courants sont la mise en place d'**alias** ce qui **accélère** la **saisie** (dans .bashrc) :

```
    export VNC_OPTIONS='-fullscreen -bgr233'
    alias vncviewer='vncviewer $VNC_OPTIONS'
    alias ssh_soler='ssh -L 5900:localhost:5900 -p 5901 -l soler
soler.upf.pf'
    alias ssh_mamao='ssh -L 5900:localhost:5900 -p 5901 -l administrateur
mamao-upf.dyndns.org'
    alias ssh_maison='ssh -p 1222 -l miepas vaiopoo.dyndns.org'
    alias reservation='cd Reservation.app/; sh Reservation.sh'
```

Les **variables** d'**environnement** permettent de stocker des **paramètres globaux**, on peut par exemple définir : **HTTP_PROXY=''http://proxy.upf.pf:8080''**

Configuration de l'impression

On utilise désormais **cups**. Ce logiciel est réparti en **plusieurs fichiers** Debian : **cupsys** est le **serveur** central, **cupsys-bsd** est une **couche** de **compatibilité** offrant les commandes du système d'impression traditionnel (lpd, lpr, lpg, etc.), **cupsys-client** renferme un ensemble de composant pour interagir avec le serveur (bloquer, débloquer une imprimante...). Enfin **cupsys-driver-gimpprint** contient une collection supplémentaire de pilotes d'imprimantes.
Cupsys s'administre très facilement grâce à son interface web : **http://localhot:631**. On peut aussi administrer cupsys avec l'interface graphique **gnome-cups-manager**.
Par la suite, **/etc/printcap** est obsolète, il faut donc le supprimer et en faire un lien symbolique vers **/var/run/cups/printcap**.

Configuration du chargeur d'amorçage

Au cas où celui-ci disparaîtrait du **Master Boot Record**. Cela peut se produire suite à l'installation d'un autre système d'exploitation tel que **Windows**.

Identifier ses disques

Pour identifier les disques, il existe un système spécial stockés dans le répertoire **/dev/**. Ainsi, **/dev/hda** est le disque maître du **premier contrôleur IDE** et **/dev/hdb** son **disque esclave**. **/dev/hdc** et **/dev/hdd** sont **respectivement** les **disques maîtres** et **esclaves** du **deuxième contrôleur IDE**. **/dev/sda** correspond au **premier disque dur SCSI**, **/dev/sdb** au **deuxième**, etc.
Chaque **partition** est ensuite **représenté par** un **numéro /dev/hda1** est la **première partition** du **disque** dur **maître** du **premier contrôleur** et **/dev/sdb1** la **première** du **disque SCSI**.
L'**architecture PC** (ou i386) est **limitée** à **quatre partitions** primaires par disque. Pour **outrepasser** cette **limite**, il faut en créer une **étendue** qui contiendra des **partitions secondaires**. Ces **dernières** portent toujours un **numéro >= 5**. La **première** pourra donc être **/dev/hda5** suivie de /dev/hda6, etc.
On peut afficher les messages du noyau au démarrages afin de voir les **périphériques détectés** à l'aide de la commande **dmesg**.

udev et /dev

La structure statique de **/dev** ne **permet pas** de **définir** des **périphériques dynamiques** nécessaires à **hotplug**. En employant **udev** (Userspace), un **système** de **fichiers** stockés en **RAM** et **géré par udevd** dissimule le contenu de /dev. Il **collabore avec hotplug pour** être informé de **l'apparition** (à chaud) des **périphériques** puis **crée dynamiquement** les **fichiers spéciaux** correspondant dans **/dev.**
On peut ainsi garder le même nom pour un périphérique donné quelque soit le connecteur employé. /dev ne contient plus que les fichiers utiles sur le moment. Auparavant, certains modules se chargeaient automatiquement lorsqu'on tentait d'accéder au périphérique. Désormais, le fichier spécial n'existe plus avant d'avoir chargé le module, ce qui n'est pas grave puisque la **plupart** des **modules** sont **chargés** au **démarrage** grâce à la **détection automatique**. Mais pour des **périphériques non détectables**, comme le **lecteur disquette** ou **la souris PS/2**, cela ne fonctionne pas. **Pensez** donc à **ajouter** les **modules floppy**, **psmouse** et **mousedev** dans **/etc/modules** afin de forcer leur chargement au démarrage.

Configuration de LILO

C'est le plus **ancien chargeur d'amorçage**. Il code **en dur dans** le **MBR** l'adresse du noyau à démarrer, c'est pourquoi chaque **mise à jour** de **LILO** doit être **suivie** de la **commande lilo**. Il a pour fichier de configuration **/etc/lilo.conf**

```
Exemple : Fichier de configuration de LILO
# le disque sur lequel LILO doit s'installer.
# En indiquant le disque et non une partition,
# on ordonne à LILO de s'installer sur le MBR.
boot=/dev/hda
# la partition qui contient Debian
root=/dev/hda2
# l'élément à charger par défaut
default=Linux
# Noyau le plus récent
image=/vmlinuz
label=Linux
```

```
initrd=/initrd.img
read-only
# Ancien noyau (si le nouveau ne démarre pas)
image=/vmlinuz.old
label=LinuxOLD
initrd=/initrd.img.old
read-only
optional
# Seulement pour un double amorçage Linux/Windows
other=/dev/hda1
label=Windows
```

Configuration de GRUB

Il est **plus récent**, ne **nécessite pas de l'invoquer après** chaque **mise à jour** car il **sait lire** les **systèmes** de **fichiers** et retrouve la position du noyau sur le disque. Pour l'**installer dans** le **MBR** du **premier disque IDE**, il suffit de saisir **grub-install /dev/hda.**
Il a pour fichier de configuration **/boot/grub/menu.lst.**

```
Exemple :  Fichier de configuration de GRUB
# Démarre automatiquement après 30 secondes
timeout 30
# Démarre la première entrée par défaut
default 0
# Si celle-ci échoue alors essaie la seconde
fallback 1
# Dernier noyau installé
title GNU/Linux
root (hd0,1)
kernel /vmlinuz root=/dev/hda2
initrd /initrd.img
# Ancien noyau (si le récent bug)
title GNU/Linux OLD
root (hd0,1)
kernel /vmlinuz.old root=/dev/hda2
initrd /initrd.img.old
# Double amorçage Linux/Windows
title Microsoft Windows
rootnoverify (hd0,0)
makeactive
chainloader +1
```

Grub fait appel au **BIOS** pour **identifier** les **disques**. **hd0** correspond au **premier disque** detecté, **hd1** le **second**. Dans la **majorité** des **cas**, cela **correspond** à l'**ordre habituel** des **disques sous Linux**. Mais parfois les **disques SCSI** peuvent **poser** problème. **Grub stocke** les **correspondances** dans **/boot/grub/device.map**. **Si** vous y trouvez des **erreurs**, **corrigez** les **manuellement** et **exécuter grub-install**. **Première partition** du **premier disque** est **notée (hd0,0)**, la **deuxième (hd0,1)**, etc.

Synchronisation, logs, partages...

le **fuseau horaire** peut être modifié par l'**outil base-config**. Sa **configuration** est **stockée** dans le fichier **/etc/timezone**; par ailleurs le **lien symbolique /etc/localtime** est **mis à jour** pour **pointer** vers **/usr/share/zoneinfo** qui contient notamment les dates des changements d'heure pour les pays.
Pour **changer temporairement** les fuseaux horaires on peut utiliser la **variable TZ**.

```
date
sam aoû 28 15:49:09 CEST 2004
TZ=''Pacific/Honolulu'' date
sam aoû 28 03:50:07 HST 2005
```

Pour les **liens symboliques**, utilisez la commande **ln -s cible nom-lien** qui **crée** un **lien nom-lien** pointant **sur cible**.

Rotation des fichiers de logs

Les fichiers de **logs** prennent **rapidement** du **volume**. On emploie en général une **archive tournante** : les **fichiers** sont régulièrement **archivés** et seules ses **X dernières archives** sont **conservées**. **logrotate**, le programme **chargé** de ces **rotations**, suit les **directives** du fichier **/etc/logrotate.conf et** de tous **ceux** du répertoire **/etc/logrotate.d**. Il peut être intéressant d'**augmenter** le **nombre** de **fichiers archivés ou** de les **déplacer dans** un **répertoire spécifique**. Ce **programme** est **exécuté quotidiennement par** l'ordonnanceur **cron**.

Synchronisation horaire

Il faut **synchroniser** les **stations** de travail **par NTP** (Network Time Protocol) au **démarrage**. Il faut pour cela **installer ntpdate**. On pourra **changer** au **besoin** le **serveur NTP** employé en **modifiant** le fichier **/etc/default/ntpdate.**

Pour les serveurs

Pour conserver une **heure correcte** en **permanence**, il faut un **serveur NTP** local **grâce** au paquet **ntp-simple** qui **utilisera** par **défaut pool.ntp.org**. On peut le **configurer** à travers **/etc/ntp.conf**.

Module GPS

Si la **synchronisation** est **primordial**, on peut **utiliser** un **module GPS** (par satellite) **ou DCF-77** (qui capte sur l'horloge atomique de Francfort). Il faut **utiliser** le paquet **ntp-refclock** capable de **piloter** ces **modules**.

Partage des droits d'administration

Bien souvent, **plusieurs administrateurs s'occupent** du **réseau**. Et **partager** le **mot de passe de** l'utilisateur **root** ouvre la porte à des **abus** du fait de l'**anonymat** de ce **compte partagé**. La **solution** est le **programme sudo** qui permet d'**exécuter** certaines **commandes avec** des **droits particuliers**. Dans son emploi le plus courant, **sudo permet** à un **utilisateur de confiance** d'**exécuter** n'importe quelle **commande en** tant que **root**.
Pour **déléguer** les **droits**, il faut faire appel à la **commande visudo** qui modifie le fichier **/etc/sudoers** (avec vi ou tout editeur mentionné dans la variable EDITOR).
L'ajout d'une ligne utilisateur ALL=(ALL) ALL permettra à l'utilisateur d'exécuter n'importe quelle commande en tant que root.

```
Exemple : fichier sudo basique
# Host alias specification
```

```
# User alias specification

# Cmnd alias specification
Cmnd_Alias STOPPC = /sbin/shutdown, /sbin/reboot, /sbin/halt,
/usr/sbin/xfsm-shutdown-helper
Cmnd_Alias SOUND = /usr/bin/xmms
Cmnd_Alias SYNAPTIC = /usr/sbin/synaptic
Cmnd_Alias SYNAPTIC32 = /usr/local/bin/synaptic32
Cmnd_Alias KONQ = /usr/bin/konqueror
# Defaults
Defaults   !lecture,tty_tickets,!fqdn
# User privilege specification
root  ALL=(ALL) ALL
# miepas localhost = NOPASSWD: SYNAPTIC, KONQ
ALL localhost = NOPASSWD: STOPPC
# Members of the admin group may gain root privileges
%admin     ALL=(ALL) ALL
```

Liste des points de montage

Le fichier **/etc/fstab donne** la **liste** de tous les **montages** possibles (**effectués automatiquement** au **démarrage** ou à **exécuter** manuellement pour les **périphériques amovibles**). Chaque **point** y est **détaillé** sur une ligne par **plusieurs champs** séparés par des blancs :

- **périphérique à monter** : partition locale (DD, cédérom) ou système de fichiers distant (NFS)
- **point de montage** : c'est l'endroit où ce système de fichiers sera rendu accessible
- **type** : ce champ défini le système de fichiers employé (ext3, vfat, ntfs, reiserfx, xfs, vfat...)
  La valeur spéciale swap sert au fichier d'échange. La valeur spéciale auto demande à mount de le détecter automatiquement.
- **options** : nombreuses et documentées dans la page de manuel de mount. En voici quelques unes :

– **rw ou ro** (lecture/écriture ou lecture seule)
– **noauto** désactive le montage automatique au démarrage
– **user** autorise tous les utilisateurs à monter ce système
– **defaults** (options **rw,suid,dev,exec,auto,nouser et async**)

- **sauvegarde** : ce champ est presque **toujours** à **0**. Lorsqu'il vaut **1**, il **indique** à **dump** que la partition contient des **données** à **sauvegarder**
- **ordre de vérification** : ce dernier champ indique si l'intégrité du système doit être vérifié au démarrage et dans quel ordre. 0, aucune vérification. Le système racine doit être à 1 et les autres permanents recevront la valeur 2.

```
Exemple : /etc/fstab
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
/dev/sda3       /               ext3    defaults,errors=remount-ro 0
1
/dev/sda5       /home           ext3    defaults        0       2
/dev/sda1       /media/sda1     ntfs    defaults,uid=1000,gid=1000
0
0
/dev/sda2       /media/sda2     vfat
defaults,rw,users,uid=1000,gid=1000       0       2
/dev/sda6       none            swap    sw              0       0
/dev/hda        /media/cdrom0   udf,iso9660 user,noauto     0       0
/dev/fd0        /media/floppy0  auto    rw,user,noauto  0       0
//draco/miepas  /media/miepas   smbfs
username=miepas,password=password,users,hard,rw,uid=1000,gid=1000     0 0
//draco/Infos-UPF  /media/Infos-UPF   smbfs
username=miepas,password=password,users,hard,rw,uid=1000,gid=1000     0 0
//draco/Micros  /media/Micros   smbfs
username=miepas,password=password,users,hard,rw,uid=1000,gid=1000     0 0
//draco/Applis  /media/Applis   smbfs
username=miepas,password=password,users,hard,rw,uid=1000,gid=1000     0 0
guiness:/partage /partage       nfs     defaults                      0
0
```

AutoMonteur

Le paquet **am-utils** fournit l'**auto-monteur amd**, capable de **monter** les **périphériques amovibles** à la demande lorsqu'un utilisateur tentera d'accéder à leur point de montage habituel. D'autres **auto**

monteurs existe comme **automount** du paquet **autofs**.

Locate et updatedb

la commande **locate retrouve** l'**emplacement** d'un **fichier** dont on **connait** une partie du **nom**. Le **résultat** est **quasi-instantanée** car elle **consulte** une **base de données** particulière **mise à jour quotidiennement** grâce à **updatedb** (à travers **/etc/cron.daily/find**). Il est possible dans **/etc/updatedb.conf** d'**indiquer** des **répertoires** à ne **pas prendre en compte** (**dans** la **variable PRUNED-PATHS**). Le paquet **slocate va** plus **loin** en utilisant une **version sécurisée** n'**indiquant** que les **noms** de **fichiers** auquel l'**utilisateur** a **accés**.

Compilation d'un noyau

Certains utilisateurs préfèrent **recompiler** le **noyau** en incluant que le strict nécessaire afin d'**optimiser** la **consommation** de **mémoire** et les **performances** ainsi que **limiter** les **failles de sécurité**. La **recompilation** est aussi **nécessaire si** l'on souhaite **employer** certaines **fonctionnalités** non intégrées **dans** la version **standard** mais **disponible sous** forme de **correctifs** ou **patches**.

Introduction

Debian gère le **noyau** sous forme de paquet. Des **outils** ont été **développé** permettant de **générer** facilement un **paquet Debian** à partir des **sources** du **noyau** ainsi que d'**éventuels patches**. Pour **compiler** un **noyau** à la **Debian**, il faut **employer** les **outils présents** dans le **paquet kernel-package**. Cette **compilation nécessite** aussi le **paquet libncurses5-dev**. Et **fakeroot** permettra de **créer** le **paquet Debian sans** utiliser les **droits administrateurs**.

Récupérer les sources

Installer un paquet **kernel-source-version**. Une **requête apt-cache search ^kernel-source** permet d'**obtenir** la **liste** des différentes **versions**. Les **dernières versions** en date sont vraisemblablement **disponibles** dans **unstable**. On peut les **récupérer** sans grand risque. Il est à noter que les sources ne sont pas celle direct de Linus Torvald, en effet, comme toute distribution, Debian y intégre des correctifs ainsi que quelques fonctionnalités (comme cramfs). Ainsi, le paquet **kernel-source-2.6.10** a été installé. Il contient le fichier /usr/src/kernel-source-2.6.10.tar.bz2. Il faut le **décompacter dans** un nouveau répertoire (**~/kernel/**)

```
mkdir ~/kernel; cd~/kernel
tar jxf /usr/src/kernel-source-2.6.10.tar.bz2
```

Configuration du noyau

Dépend du mode opératoire, si l'on veut juste **compiler** une **version** plus **récente** du noyau, on va **rester** le plus **proche** possible de la **configuration standard**. Il est alors bon de **copier** le fichier **/boot/config-<version> (<version> étant égale à uname-r) en .config** dans le **répertoire** des **sources** du noyau.

```
cp /boot/config-2.6.8.2-686 ~/kernel/kernel-source-2.6.10/.config
```

Si vous avez décidé de **tout reconfigurer**, il existe **différentes interfaces** qu'on invoque depuis le répertoire source par la **commmande make** :

- **make menuconfig** compile et exécute une interface évoluée en mode texte, (c'est ici que libncurses5-dev est requis) qui propose de naviguer dans une structure hiérarchique présentant les différentes options. Espace change la valeur et Entrée valide le bouton sélectionné en bas de l'écran. Select permet de rentrer dans le sous menu, Exit remonte d'un cran et Help produit des informations détaillées. Pour quitter, sélectionner Exit dans le menu principal puis sauvegarder.
- **make xconfig** emploie la boite à outils Qt et make gconfig recourt à GTK+. La première a besoin de libqt3-mt-dev et la seconde libglib2.0-dev, libglade2-dev et libgtk2.0-dev.
- **make-kpkg** exécutera automatiquement make oldconfig. Cette méthode se contente de réutiliser les choix mémorisés dans le fichier .config. En l'absence de ce dernier, elle exécute make config.

Vous pouvez indiquer à **make-kpkg** d'employer une **autre méthode** que **oldconfig** grâce à :

```
make-kpkg --config (menuconfig,xconfig ou gconfig)
```

Compilation et génération du paquetages

Si vous souhaitez recommencer depuis des **sources vierges**, il faut exécuter **fakeroot make-kpkg clean**. Cela vous permettra aussi de **générer** un **paquet** avec un **nouveau nom** (**--append-to-version**).

make-kpkg permet de compiler le noyau puis de générer le paquet Debian. Tout comme make, elle prend en paramètre le nom d'une **cible** à exécuter:

- **kernel-image** génére un paquet du noyau compilé,
- **kernel-doc** un paquet contenant la documentation incluse avec le noyau,
- **kernel-headers** un paquet des fichiers d'en-tête, utilise /etc/kernel-pkg.conf (mettre des informations correctes si vous souhaitez diffuser le paquet)
- **kernel-source** un paquet contenant les sources

make-kpkg accepte des **paramètres** : **--append-to-version** suffixe ajoute **suffixe** à la fin du **nom** du **noyau** et du **paquet généré**. **--revision** définit le **numéro** de **version** du paquet. (Debian emploie certains **suffixe** pour **identifier** les **processeurs** (-386, -686, -686-smp, -k6, -k7, -k7-smp, il ne faut pas les utiliser afin de **distinguer** les **paquets officiels** des autres).
make-kpkg tentera d'**exécuter avec** les **droits roots**, c'est pourquoi on utilise **fakeroot** :

```
fakeroot make-kpkg --append-to-version -upf --revision 1 kernel-image
ls ../*.deb
./kernel-image-2.6.10-upf_1_i386.deb
```

Compilation de modules externes

Debian fournit les **sources** d'un certain nombre de **modules externes** : **pcmcia-source** (gestion PCMCIA récent), **alsa-source** (pour certaines cartes sons), **qc-usb-source** (pour certaines webcam usb), etc. Les **deux premiers paquets** sont **intégrés** dans la **version 2.6**
La commande **apt-cache search source$** permet d'identifier les sources disponibles.
Prenons **qc-usb-source** :

```
cd ~/kernel/
tar zxf /usr/src/qc-usb-modules.tar.gz
ls modules/
qc-usb-source
```

Pour **compiler** ces **modules** et **créer** un **paquet Debian**, il faut appeler **make-kpkg** avec le **paramètre modules-image en** lui **indiquant** via **MODULE_LOC où trouver** les **modules** (si absente, emploie **/usr/src/modules**). L'option **--added-modules** permet d'**indiquer explicitement** les **modules externes** à **compiler** :

```
export MODULE_LOC=~/kernel/modules
cd ~/kernel/kernel-source-2.6.10
fakeroot make-kpkg --append-to-version -upf modules-image
ls ../*.deb
../kernel-image-2.6.10-upf_1_i386.deb
../qc-usb-modules-2.6.10-upf_0.6.2-2+1_i386.deb
```

Emploi d'un patch sur le noyau

Debian les diffuse par le biais de **kernel-patch-*** comme **kernel-patch-debianlogo** qui **remplace Tux par** le **logo Debian**. Ils sont dans **/usr/src/kernel-patches/**.
Il faut employer l'option **--added-patches** :

```
cd ~/kernel/kernel-source-2.6.10
fakeroot make-kpkg clean
fakeroot make-kpkg --append-to-version -mppe --revision 1 --added-patches
mppe kernel-image
```

Installation d'un noyau

Un paquet Debian de noyau installe l'image du noyau (**vmlinuz-<version>**), sa configuration (**config-**

<version>) et sa **table** de **symboles** (**System.map-<version>**) dans **/boot/**.
Divers **paramètres** existe dans **/etc/kernel-img.conf**.

Installation avec dpkg

```
dpkg -i kernel-image-2.6.10-upf_1_i386.deb
```

Services Unix

Démarrage du système

En premier lieu, le **BIOS** prend le **contrôle**, **détecte** les **disques**, **charge** le **MBR et** l'**exécute**.
Le **chargeur** d'**amorçage** prend le relais, il **trouve** le **noyau**, le **charge et** l'**exécute**. **Puis** le **noyau s'initialise**, **monte** la **partition racine et** peut enfin **démarrer** le **premier programme** : **init**.
Celui-ci **exécute** un **ensemble de processus** en suivant **/etc/inittab**. Le **premier programme exécuté** est **/etc/init.d/rcS,** script **qui exécute** tous les **programmes** du **répertoire /etc/rcS.d.**
Parmi ceux-ci, on trouve :

- la configuration du clavier de la console
- le chargement des pilots : modules noyaux spécifiés dans /etc/modules, puis les modules indiqués par discover en fonction du matériel qu'il détecte
- la vérification de l'intégrité des systèmes de fichiers
- le montage des partitions locales
- la configuration du réseau
- le montage des systèmes de fichiers distants (NFS)

Modules du noyau et options

Le noyau **2.4** utilise le **/etc/modules.conf généré** par **update-modules** à partir de **/etc/modutils/**
Le noyau **2.6** préfère **/etc/modprobe.conf incluant /lib/modules/modprobe.conf** (généré par **update-modules** à partir de **/etc/modprobe.d/**). Pour celui-ci **modprobe** provient du **paquet module-init-tools et** non **plus** de **modutils**.
Après cette phase, **init reprend la main** et **démarre** les **programmes** associés au **niveau d'exécution normal**, par défaut le **niveau 2**. **/etc/init.d/rc 2** qui **démarre** les **services** du répertoire **/etc/rc2.d/ débutant par** la lettre **S** (pour **Start** et **K** pour **Kill**). D'une manière générale, les **services de base** comme **syslogd** ou **portmap suivis** par les **services standards** puis le **module graphique gdm**.
Init distingue **plusieurs niveau,** on peut passer de l'un à l'autre avec **telinit nouveau-niveau.**
Tous les **scripts** contenus dans **/etc/rc<X>.d**, ne sont que des **liens symboliques** créés à l'installation concerné par le **programme update-rc.d** et **menant** vers les **scripts réels** sous **/etc/init.d**. Pour **adapter** les **services** à **démarrer**, on peut exécuter **update-rc.d.**

Rédémarrage des services

/etc/init.d/service restart ne **prend pas en compte** le **niveau d'exécution** qui **suppose** à tort que le **service** est **actuellement employé** et **peut donc** le **redémarrer** à mauvais escient. Debian a donc introduit **invoke-rc.d.**
Enfin, **init démarre** les **programmes** de **contrôle** des différentes **consoles virtuelles** (**getty**). Ils **affichent** une **invite**, **attendent** un **nom d'utilisateur** puis **exécutent login utilisateur** pour démarrer la session.

Connexion à distance

telnet, le **pire protocole** au point de vue de la **sécurité**. **Mots de passe en clair.** Désinstallez le:

```
apt-get remove telnetd
```

Il existe une version **SSL** : **telnetd-ssl** et **telnet-ssl** (**serveur** et **client**)
SSH (OpenSSH) est conçu dans une **optique** de **sécurité**. Le partenaire est **authentifié** et tous les **échanges** sont **chiffrés**. Il offre **deux services** de transfert :

`scp : scp fichier machine:/home/miepas/`
`sftp similaire à ftp`
Pour l'installer, **apt-get install ssh** et **accepter** l'exécution de **sshd**

Utiliser des applications X11 à distance

Pour activer le **X11 forwarding** avec **SSH**, indiquez **X11Forwarding yes** dans **/etc/ssh/sshd_config** (désactivé par défaut pour des raisons de sécurité). On peut ensuite en profiter en spécifiant l'**option -X** de **ssh**.
Créer des **tunnels chiffrés** avec le **port forwarding**
-R et **-L** permettent à **ssh** de **créer** des **tunnels chiffrés** déportant de manière **sécurisé** un **port TCP local sur** une **machine distante** ou vice versa.
ssh -L 8000:serveur:25 intermediaire crée une **socket locale** sur le **port 8000**. Toute **connexion établie** sur ce port fera **débuter** par ssh une **connexion** de **l'ordinateur intermediare** vers le **port 24 du serveur** à laquelle elle la reliera.
ssh -R 8000:serveur:25 intermediaire crée une **socket écoutant** le **port 8000 sur** la **machine intermediaire**. Toute **connexion sur ce port** fera **débuter** par ssh une **connexion** depuis la **machine locale vers** le **port 25 du serveur**, à laquelle elle la reliera.

Accéder à distance à des bureaux graphiques

VNC (Virtual Network Computing ou informatique en réseau virtuel) permet d'**accéder** à distance à des **bureaux graphiques**. Il suffit d'**exécuter x11vnc** du paquet éponyme puis à l'**administrateur** d'exécuter **xvncviewer**. Il existe aussi **vncserver** (suivre les indications de **/usr/share/doc/vncserver/README.inetd**) qui **crée** un **serveur X destiné** exclusivement à une **connexion** à distance, **x11vnc exploite** quant à lui un **serveur X existant.**

Gestion des droits

Chaque **fichier** ou répertoire **dispose** de **permissions spécifiques** pour **3 catégories** d'utilisateurs :
- son **propriétaire** (symbolisé par u comme user)
- son **groupe propriétaire** (symbolisé par g comme group) représentant tous les utilisateurs membres
- les **autres** (symbolisé par o comme other)

Trois types de droits peuvent s'y combiner :
- lecture (r comme read)
- écriture (ou modification, w comme write)
- exécution (x comme eXecute)

Pour un **fichier**, c'est simple, **lecture** pour **consulter** le contenu, **écriture** pour le **modifier** et **exécution** pour tenter de l'**exécuter** (ce qui ne **fonctionne que si** c'est un **programme**)
Pour un **répertoire**, la **lecture** donne le droit de **consulter** la **liste** de ses **entrées**, l'**écriture** permet d'y **créer** ou **supprimer** des **fichiers** et l'**exécution** permet de le **traverser** (**cd rep**). Pouvoir **exécuter** un répertoire **sans le lire** permet d'**accéder à ses entrées** dont on **connaît** le **nom** mais **pas** de les **retrouver** le cas échéant.

Exécutables setuid et setgid

2 droits particuliers concernent les fichiers exécutables : le bit **setuid** et le bit **setgid** (symbolisé par la lettre s). Ils permettent à **n'importe quel utilisateur** d'**exécuter** le **programme** en question **avec** respectivement les **droits** de son **propriétaire ou** de son **groupe propriétaire**. Ce mécanisme **donne accès** à des **droits plus élevés** que ceux habituels.

Répertoire setgid et sticky bit

Le bit **setgid s'applique également** aux **répertoires**. Toutes les **entrées crées** recevront alors pour **groupe propriétaire celui** du **répertoire au lieu de prendre** comme d'habitude **le groupe principal** du créateur. Le bit **sticky** (symbolisé par la lettre « t ») est **réservé** aux **répertoires**. Il est employé **notamment** pour **/tmp** puisqu'il **limite** la **suppression** d'un **fichier** à son **propriétaire et** à **celui** de son répertoire **parent**. En son **absence**, **tout le monde** pourrait **supprimer** les **fichiers** d'**autrui** sous **/tmp** puisqu'**ouvert** en **écriture** à **tous**.

Manipuler les permissions

chown utilisateur fichier affecte un nouveau **propriétaire** à un **fichier**
chgrp groupe fichier opère sur son **groupe propriétaire**
chmod droits fichier intervient **sur** ses **droits**
On peut **utiliser** l'**option -R** pour **effectuer** l'**opération récursivement**
Pour les **droits**, on peut **utiliser** les **symbols**, pour **chaque catégorie** d'utilisateur **(u/g/o)**, on peut **définir** les **droits (=),** en **ajouter (+)** ou en **retrancher (-)**. Ainsi la **formule u=rwx,g+rw,o-r donne** au **propriétaire tous** les **droits**, **ajoute** au **groupe** les **droits** de **lecture et** d'**exécution** et **supprime** les **droits** de **lecture** aux **autres**.
La **distinction** entre **répertoire** et **fichier pose** parfois **problème** lors des **opérations récursives**. C'est la **raison** de la **lettre X**. Elle **représente** un **droit** d'**exécution** qui ne **concerne** que les **répertoires**.
Ainsi **chmod -R a+X** répertoire n'**ajoutera** les **droits** d'**exécution** pour **toutes** les **catégories** d'**utilisateurs** qu'**a tous** les **sous répertoires**.
La **représentation** numérique **octale** associe **chaque droit à une valeur**: **4** pour **lecture**, **2** pour **écriture**, **1** pour **exécution**.
chmod 765 fichier mettra donc **tout** les **droits** au **propriétaire** (**7=4+2+1**); **lecture** et **écriture** au

groupe (**6=4+2**); **lecture** et **exécution** aux **autres** (**5=4+1**).
Pour **représenter** le cas échéant les **droits spéciaux**, on pourra **préfixer** à **ce nombre** un **quatrième chiffre** sachant que **setuid**, **setgid** et **sticky valent** respectivement **4,2 et 1**. **chmod 4765 associera** donc le **bit setuid** aux **droits décrits** précédemment.
Losqu'une **application crée** un **fichier**, elle lui **donne** des **permissions indicatives** sachant que le **système retire** automatiquement **certains droits donnés par umask**. Par **défaut**, à **0022 (droit en ecriture pour le groupe et les autres utilisateurs sont retirés).**
Si on lui **passe** une **nouvelle valeur**, on peut donc **positionné** les **droits par défaut**.

Administrer sur interface web : webmin

Une des interfaces les plus abouties. Il dispose de **nombreux modules** couvrant un **large panel.**
Changez le mot de passe root qui est par défaut celui du système. Pour voir la **liste** des **modules**, **apt-cache search webmin-**

- webmin-bind : configuration du serveur DNS
- webmin-postfix : configuration du serveur SMTP
- webmin-inetd : configuration du super-serveur
- webmin-quota : gestion des quotas utilisateurs
- webmin-dhcpd : configuration du serveur DHCP
- webmin-proftpd : configuration du serveur FTP
- webmin-samba : configuration du serveur Samba
- webmin-software : installation ou suppression de logiciels à partir des paquets Debian et mise à jour du système

L'interface est **accessible** depuis l'adresse **https://localhost:10000**. Autre **alternatives**, **linuxconf** (plutôt orienté rpm) et **gnome-system-tools** (encore instable).
Pour les **mises à jour**, le programme **ucf** permet de **configurer** la manière dont **debconf** interagit avec l'utilisateur.

Les événements sytème de syslog

2 démons (**syslogd** et **klogd**) ont pour charge de **collecter** les **messages** provenant des **applis** et du **noyau** puis de les **répartir** dans des **fichiers** de **logs (/var/log)**. Ils **obéissent** au fichier de **configuration /etc/sylog.conf**.
Chaque **message** est **associé** à un **sous système applicatif** (**facility**) :

- auth et authpriv concernent l'authentification
- cron provient du serveur cron ou atd
- daemon concerne un démon sans classification particulière (serveur DNS, ntp, etc)
- ftp concerne le serveur FTP
- kern message provenant du noyau
- lpr provient du système d'impression
- mail provient de la messagerie electronique
- news : message du sous système Usenet (notamment de NNTP)
- syslog message de syslog lui-même
- user message utilisateur générique
- uucp messages du sous-sytème UUCP (vieux protocole utilisé pour faire circuler entre autres des messages electroniques)
- local0 à local7 réservés pour les utilisations locales

A chaque **message** une **priorité** :

- emerg : au secours, système probablement inutilisable
- alert : péril, des actions doivent être entreprises immédiatement
- crit : conditions critiques
- err: erreur,
- warn : avertissement (erreur potentielle)
- notice : condition normale mais message significatif
- info : message informatif
- debug : message de déboguage

Le fichier de configuration

Le **principe** est d'écrire des **paires sélecteur** et **action**. Le **sélecteur** défini l'**ensemble des messages** concernés et l'**action** décrit **comment les traiter**. Le **sélecteur** est une **liste** (séparé par un ;) de **couples « facilité.priorité »** (ex: **auth.notice; mail.info, *** représente tout ex: ***.alert ou mail.***).
On peut **regrouper plusieurs facilités** en les **séparant** par une **virgule** (ex: **auth,mail.info**). La **priorité** indiqué **recouvre** aussi les **messages** de **priorité supérieure ou égale** : **auth.alert désigne** les **messages** du **sous-système auth** de **priorité alert** ou **emerg**. **Préfixée** par un **point d'exclamation**, elle **désigne** les **priorités strictement inférieures** (ex: **auth.!notice** désigne donc les **messages** issus **de auth** et **de priorité info ou debug**). **Préfixé** par un **signe égal**, elle correspond **exactement** à la **priorité indiquée** (**auth.=notice** ne concerne que les **messages auth** de **priorité notice**). Au sein du **sélecteur**, chaque **élément** de la **liste surcharge** les **éléments précédents**. Il est donc possible de **restreindre** un **ensemble** ou d'en **exclure certains éléments**. Ex: **kern.info;kern.!err** définit les **messages** du **noyau** de **priorité comprise** entre **info** et **warn**. La **priorité none** désigne l'**ensemble vide** et peut **servir pour exclure** une **facilité** d'un **ensemble de messages.** Ainsi, ***.crit;kern.none désigne** tous les **messages** de priorité **supérieure ou égale à crit** ne **provenant pas** du **noyau**.

Syntaxe des actions

Les différentes actions possibles sont:

- ajouter le message à un fichier (/var/log/mesages)

- envoyer le message à un serveur syslog distant (exemple: @log.upf.pf)
- envoyer le message dans un tube nommé préexiste (exemple : /dev/xconsole, pour en créer un utiliser mkfifo)
- envoyer le message à un ou plusieurs utilisateurs s'ils sont connectés (ex: root,miepas)
- envoyer le message à tous les utilisateurs connectés (exemple: *)
- écrire le message sur une console text (ex: /dev/tty8)

Déporter les logs

C'est une bonne idée que d'enregistrer les **logs** les **plus importants** sur une **machine distante**, cela **empêchera** un éventuel **intrus** de **supprimer** les **traces** de son passage. Par ailleurs, en cas de **problème majeur** (plantage noyau), on pourra voir les **logs sur** l'**autre machine**.

Le super-serveur inetd

Inetd est un **serveur de serveur**, employé pour **invoquer** à la demande des **serveurs**. Le fichier **/etc/inetd.conf** en **donne** la **liste** ainsi que les **ports habituels qu'inetd écoute** tous, **dès** qu'il détecte une **connexion** il **exécute** le **serveur correspondant**.
Après avoir **enregistrer** un **service** dans **inetd.conf** : la **commande update-inetd** demande à inetd de prendre en compte un nouveau serveur. Chaque **ligne** de **inetd.conf décrit** un **service** par **7 champs** (séparé par des blancs) :

- Le **nom du service** (qui définit implicitement le numéro de **port TCP ou UDP** par **correspondance** de **/etc/services)**
- Le **type de socket** : **stream** pour une connexion **TCP**, **dgram** pour des datagrammes **UDP**
- Le **protocole** : tcp ou udp
- Les **options** : 2 valeurs sont possibles : **wait** ou **nowait**. Ce dernier convient aux connexions TCP, facilement multiplexables. Pour les programmes UDP, il ne faut retenir nowait que si le serveur est capable de gérer plusieurs connexions parralèles. On peut suffixer ce champ d'un point suivi du maximum de connexions autorisées. (limite par défaut : 40)
- **L'identifiant** utilisateur exécutant le serveur
- Le **chemin** complet du **programme** serveur à exécuter
- Les **arguments** (liste complète des arguments du programme, y compris son propre nom argv[0])
-

```
Exemple : extrait de /etc/inetd.conf
talk dgram udp wait nobody.tty /usr/sbin/in.talkd in.talkd
finger stream tcp nowait nobody /usr/sbin/tcpd /usr/sbin/in.fingerd
ident stream tcp nowait nobody /usr/sbin/identd identd –i
```

Le programme **tcpd** est souvent employé, il permet de **restreindre** les **connexions entrantes** en **appliquant** des **règles** de contrôle **(man hosts_access)** et se **configurent** dans **/etc/hosts.allow** et **/etc/hosts.deny**.

Planification synchrone : cron et atd

cron est le **démon** en charge d'**exécuter** des **commandes planifiées** et **récurrentes** (chaque jour, chaque semaine, etc.); **atd** est celui qui s'occupe des **commandes** à **exécuter une seule fois**.
Sous linux, de **nombreuses tâches** sont **planifiées** :

- la rotation des logs
- la mise à jour de la base de données de locate;
- les sauvegardes;
- des scripts d'entretien (nettoyage fichiers temporaires)

Tous les **utilisateurs** peuvent **planifier** l'exécution de **tâche**. On **modifie** le **crontab** avec **crontab -e**

(stockées **dans /var/spool/cron/crontabs/<utilisateur>**)
On peut **restreindre** l'**accés grâce** au fichier d'autorisation **/etc/cron.allow** où l'on **indique les** seuls **utilisateurs autorisés**. Pour n'en **priver que quelques uns**, **/etc/cron.deny**. De même pour **atd** (**/etc/atd.allow, /etc/atd.deny**)
Root dispose de son crontab mais **peut employer /etc/crontab** ou **déposer** des **crontab supplémentaires** dans **/etc/cron.d**. Ces deux dernières solutions ont l'avantage de préciser l'utilisateur sous l'idendité duquel exécuter la commande.
cron propose **par défaut** des **commandes planifiées** qui exécutent :
- 1 fois/heure les programmes de /etc/cron.hourly;
- 1 fois/jour les programmes /etc/cron.daily;
- 1 fois/semaine les programmes de /etc/cron.weekly;
- 1 fois/mois les programmes de /etc/cron.monthly;

Raccourcis textules pour cron : **@yearly (1 fois/an)**, **@monthly (1 fois/mois)**, **@weekly (1 fois/semaine)**, **@daily (1 fois/jour) et @hourly (1 fois/heure)**

Format d'un fichier crontab

chaque **ligne** décrit une **commande planifiée** grâce à 6 champs :
- la condition sur les **minutes** (0 à 59)
- la condition sur les **heures** (0 à 23)
- la condition sur le **jour du mois** (1 à 31)
- la condition sur le **mois** (1 à 12)
- la condition sur le **jour de la semaine** (0 à 7, 7 correspondant au dimanche, on peut aussi employer les 3 premières lettre du jour en anglais comme Sun,Mon,etc.)
- la **commande à exécuter**

Chaque **condition** peut s'**exprimer** sous la **forme** d'un **énumération** de **valeurs possibles** (séparées par des virgules). La **syntaxe a-b** décrit l'**intervalle** de **toutes** les **valeurs** comprises **entre a et b**.
a-b/c décrit un **intervalle avec** un **incrément** de **c** (ex; 0-10/2 correspond à 0,2,4,6,8,10). Le joker * représente toutes les valeurs possibles

```
Exemple :
# Format
# min heu jou moi jsem commande
# Télécharge les données tous les soirs à 19:25
25 19 * * * $HOME/bin/get.pl
# Le matin à 8h00, en semaine (lundi à vendredi)
00 08 * * 1-5 $HOME/bin/fait_quelquechose
# Redémarre le proxy IRC après chaque reboot
@reboot /usr/bin/dircproxy
```

@reboot : exécute une commande une seule fois après le démarrage de l'ordinateur, comme tous les @machin, elle remplace les 5 premiers champs.

Emploi de la commande at

at prévoit l'**exécution** à un moment **ultérieur**. Elle prend en **paramètre** l'**horaire** et la **date** prévue puis la **commande à exécuter**. Pour l'horaire, 16:12 représente 16h12. La date peut être précisée au format JJ.MM.AA (27.07.04, le 27 juillet 2004). En son absence, dès que l'horloge atteint l'heure signalée (le même jour ou le lendemain). On peut aussi saisir **today** ou **tomorraw**.

```
Exemple :
cat <<FIN | at 16:12 30.12.06
echo ''Joyeux anniversaire'' | mail miepas@upf.pf
FIN
```

Une **autre syntaxe** permet d'exprimer une **durée d'attente** : **at now + nombre période**. **Période** peut valoir **minutes**, **hours**, **days** ou **weeks**. **nombre** indique simplement le **nombre** de ces **unités**.
On peut **annuler** une **commande** planifiée avec **atrm numéro-de-tache**. Le **numéro** est indiquée lors de la planification mais on peut le **retrouver** avec la commande **atq**.

Planification asynchrone : anacron

anacron complète cron pour les **ordinateurs non allumés** en **permanence**. **Anacron** tente d'**exécuter** les **commandes** en prenant **en compte** les **périodes** où l'**ordinateur ne fonctionne pas**.
Les tâches du fichier **/etc/anacrontab** sont **démarrées sous** la commande **nice** (pour **limiter** leur **priorité**)
l'installation d'**anacron désactive** l'exécution par **cron** des scripts **/etc/cron.hourly,** etc. Mais **cron** reste **actif** et se chargera d'**exécuter** les **autres commandes planifiées** (notamment par les utilisateurs)

Les quotas

Ce système permet de **limiter** l'**espace disque** alloué à un **utilisateur** ou **groupe d'utilisateurs**. Les **logiciels** de **gestion** de **quotas** se trouvent dans le **paquet** Debian **quota**.
Pour les **activer sur** un **système** de **fichier**, il faut **indiquer** dans **/etc/fstab usrquota** et **grpquota**. **Redémarrer** l'**ordinateur** permet ensuite de **mettre à jour** les **quotas** en l'absence d'activité disque.
La commande **edquota utilisateur ou edquota -g groupe** permet de **charger** les **limites** tout en consultant la consommation actuelle.
setquota peut être **employé** dans un **script** pour **modifier automatiquement** de nombreux **quotas**.
Le système de **quota** permet de **définir quatre limites** :

- **deux limites (soft et hard)** concernent le **nombre** de **blocs consommés**. Un **bloc contient** jusqu'à **1024 octets** du même fichier. Les **blocs non saturés induisent** donc des **pertes** d'espace disque. Un **quota de 100 bloc permet** théoriquement de **stocker 102400 octets**, **sera** pourtant **saturé par 100 fichiers de 500 octets** ne représentant que 50000 octets au total.
- **deux limites (soft et hard)** concernent le **nombre d'inodes** employés. **Chaque fichier consomme** au moins **un inode** pour **stocker** les **informations** le concernant (droits, propriétaires, date de dernier accès,etc.). Il s'agit donc d'une **limite sur** le **nombre** de **fichiers utilisateurs**.

Une **limite soft** peut être **franchie**, l'**utilisateur** sera simplement **averti par** le **programme warnquota** invoqué par **cron**. Une **limite hard** ne peut **jamais être franchie.**
La commande **edquota -t** définit une **période** de « grâce » **maximale** autorisé **pour** un **dépassement soft**. Ce **délai écoulé**, **soft** se comportera **comme hard** (impossible d'écrire).
Pour instaurer un **quota systématique** chez les **nouveaux utilisateurs**, il faut le **configurer sur** un **utilisateur modèle** (avec **edquota** ou **setquota**) et **indiquer** son **nom** dans la **variable QUOTAUSER** du **fichier /etc/adduser.conf**. Il **sera** automatiquement **repris** pour **chaque nouvel utilisateur crée** avec **adduser**.

Supervison

Surveillance des logs avec logcheck

Ce programme **scrute** les **fichiers** de **logs toutes** les **heures** et **envoie** par **courrier électronique** à « **root** » les **plus inhabituels** pour **aider** à **détecter** tout **nouveau problème**.
La liste des **fichiers scrutés se trouve** dans **/etc/logcheck/logcheck.logfiles**. Les **choix par défaut conviendront** si **/etc/syslog.conf** n'a **pas été entièrement remodelé**.
logcheck fonctionne en **3 modes** : **paranoid**, **server** et **workstation**. Le **premier**, le plus **verbeux**, est pour les **serveurs spécialisés** (comme les pare-feu);
Selon votre machine, il faudra probablement **paramétrer logcheck** pour **exclure** des **messages**. Le **mécanisme** est assez **complexe** (**lire /usr/share/doc/logcheck-database/README.logcheck-database.gz).**

Logs en fond d'écran

commande **root-tail** (éponyme), le programme **xconsole** (xbase-clients) les fera **défiler** dans un petite fenêtre.

Surveillance de l'activité

En temps réel

top affiche la **liste** des **processus en** cours d'**exécution**. Son **critère** de **tri** est l'**utilisation actuelle** du **processeur** (**touche p**) mais on peut **opter** pour la **mémoire occupée** (**touche M**), le **temps processeur** consommé (**touche T**) ou le **numéro** de **procesus** ou PID (**touche N**). La **touche k** permet d'indiquer un **numéro de processus à tuer.**
gnome-system-monitor et qps sont des **outils graphiques similaires** à top.

Historiques

Charge processeur, trafic réseau ou espace disque disponible varient en permanence. Il est intéressant de **garder** une **trace** de leur **évolution**. Un **outil** comme **cacti** (multi-système) est un **programme complet** de **suivi d'informations système** utilisant **SNMP** pour la surveillance. Après **installation** du **paquet**, on peut le configurer à travers l'**interface web** **http://localhost/cacti**.
La **documentation** : **/usr/share/doc/cacti/html/index.html**
mrtg est un **outil ancien** et **rustique** capable d'**agréger** des **données historiques** et d'en faire des **graphiques**. Il dispose d'un certain nombre de **scripts** (charge, trafic réseau, impacts web,etc.)
Les paquets **mrtg-contrib** et **mrtgutils** contiennent des **scripts** d'**exemples.**
Outil **plus récent et simple zabbix** et un **outil complexe mais** très **professionnel** : **nagios**.

Sauvegarde

Il existe des **outils puissants** mais **difficile à maîtriser**, comme **amanda**, un système client/serveur. Des **dizaines d'autres** sont consacrés à cette tâche : **apt-cache search backup.**
La commande **rsync** permet aussi de **faire** une **synchronisation** sur des **serveurs distants**. Information sur **http://www.mikerubel.org/computers/rsync_snapshots/**
Elle peut être **précédée** d'une **duplication** du **contenu** de la **dernière sauvegarde** par des **liens durs** (**évitant** de **consommer** trop d'espace disque). Un **lien dur** revient en fait à **affecter** un **deuxième nom** au **fichier cible**, contrairement à une copie, il ne **consomme pas** d'**espace disque supplémentaire**, commande : **ln cible lien.** Les **ordinateurs de bureau** peuvent être **régénérer** à partir de cédéroms fabriqués par le programme **mondo**. Amorçable, ils permettent de réinstaller complètement le système de la machine. Le programme **systemimager** permet aussi de **restaurer rapidement** des **ordinateurs complets** à partir, par exemple, d'une **image stockée** sur un serveur.
Pour les **sauvegardes SQL**, **LDAP**, il faut faire appel à une **procédure d'export** des données dont on **sauvegarde** le **dump**. Pour **réduire l'espace** de stockage, on ne **stockera qu'un fichier complet par semaine** et un **diff par jour**. Le programme **xdelta** produira les **différences incrémentales**.

Branchements « à chaud » : hotplug

Il **collabore** avec le **noyau** pour **charger** les **pilotes** de **périphérique à chaud** : USB, PCMCI, IEE 1394 et même SCSI ou PCI. La **base de données associe** à chaque **identifiant** le **pilote requis**. Pour les **cartes réseaux**, **hotplung** tente de les **configurer** avec **ifup** (avec le fichier **/etc/network/interfaces**).
hotplug peut être **désactivé** avec **dpkg-reconfigure hotplug.**

Gestion de l'énergie : APM

Présent dans tous les **noyaux Debian**, **désactivé par défaut**. Pour l'**activer,** on **passe** l'**option apm=on** à la **ligne démarrant** le **noyau**. Avec **LILO**, on ajoutera la **directive append=''apm=on''**. Pour **grub**, on

ajoute **apm=on** à la **ligne débutant** par **kernel** (dans **/boot/grub/menu.lst**).

Économie d'énergie : ACPI

Interface avancée, plus **difficile** à **mettre en oeuvre**, le paquet **acpid** est le **pendant d'apmd pour** le monde **ACPI**. Si votre **BIOS gère correctement ACPI**, utilisez le. Il faut veiller à **supprimer apmd** car les deux sont **incompatibles.**

Cartes pour portables : PCMCIA

requiert **deux paquets**:

- **pcmcia-cs** qui hébergent les démons réagissant à l'insertion de nouvelles cartes.
- **kernel-pcmcia-modules-version-noyau** fournit les pilotes eux mêmes

Liste des **cartes** reconnues et fonctionnelles :
http://pcmcia-cs.sourceforge.net/ftp/SUPPORTED.CARDS
Le paquet **wireless-tools** est **nécessaire** à la bonne **prise en charge** des cartes **Wifi**.
A chaque **connexion** ou **déconnexion**, le **démon exécute** un **script** de **/etc/pcmcia** qui **trouve** ses **paramètres** dans **/etc/pcmcia/*.opts.**

Le réseau

Passerelle : **relie plusieurs réseaux** entre eux. **Désigne** souvent la **porte de sortie** d'un **réseau local** afin d'**atteindre** les **adresses IP externes**. Elle est **connectée** à **chacun des réseaux** qu'elle relie et **agit** en tant que **routeur** pour **rediriger** les **paquets** entre ses **différentes interfaces**.
Plages d'**adresses privées prévues** pour les **réseaux locaux** :

- 10.0.0.0/8 de classe A (2^24 adresses),
- 172.16.0.0/12 (16 plages de classe B : 172.16.0.0/16 à 172.31.0.0/16 pouvant contenir 2^16 adresses chacune),
- 192.168.0.0/16 de classe B (regroupant 256 plages de classe C 192.168.0.0/24 à 192.168.255.0/24 de 256 adresses chacune)

http://www.faqs.org/rfcs/rfc1918.html
Lorsqu'un **réseau** utilise une **plage privée** (**non routable** sur Internet), la **passerelle** doit **effectuer** du **masquerading** (**masquage** d'**adresses IP**) pour que ses **machines** puissent **communiquer** avec l'**extérieur**. Ça consiste à **remplacer** chaque **connexion sortante par** une **connexion provenant** de la **passerelle** elle-même (**disposant** elle d'une **adresse valable**) **puis** à **faire suivre** les **données reçues** en réponse à la machine ayant **initié** la **connexion**. La **passerelle** dispose donc d'une **plage de ports TCP dédiés** au **masquerading** (numéro généralement **supérieurs à 60000**). Chaque **nouvelle connexion apparaîtra** à l'extérieur comme **provenant** de l'un de ces **ports**. Lorsque la **passerelle reçoit** une **réponse**, elle sait à qui les **faire suivre**.
Elle **peut également effectuer** une **traduction d'adresses réseau** (**NAT** ou Network Address Translation). Il en existe **deux types** : Le Destination NAT (**DNAT**) est une technique pour altérer l'adresse IP (et/ou le port TCP ou UDP) destinataire d'une nouvelle connexion (généralement entrante). Le mécanisme de « suivi des connexions » (connection tracking) altérera aussi les autres paquets de la même connexion pour assurer la continuité de la communication. Son pendant, le Source NAT (**SNAT**) et dont le masquerading est un cas particulier altère l'adresse IP (et/ou le port TCP ou UPD) source d'une nouvelle connexion (généralement sortante). Comme pour le DNAT, le suivi des connexions gère de manière adéquate les paquets suivants.

Place à la pratique, il est très facile de **transformer** un **système** Debian en **passerelle** :

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Pour activer cette option à chaque démarrage, dans /etc/network/options :
ip_forward=yes
spoofprotect=yes
syncookies=no

Activer le masquerading nécessite la **configuration** du **pare-feu netfilter**. La paquet **ipmasq** s'en charge : il le **configure** automatiquement à **chaque démarrage** de l'ordinateur. Une option permet aussi de ne **l'activer** qu'à **chaque ouverture** d'une **connexion PPP.**
L'emploi du **NAT nécessite** lui aussi de **configurer netfilter**. L'**absence de configuration** standard explique qu'il n'y ait **pas de solution prête à l'emploi**. Des **outils** permettent de **simplifier** la **configuration** en visualisant graphiquement les règles définies **comme fwbuilder.**

Pare-feu ou filtre de paquets

Une **pare-feu** est une **passerelle filtrante** : il applique des **règles de filtrage** aux paquets qui le traversent (**c'est pourquoi** il **n'est utile** qu'en tant que **point de passage obligé**). C'est un **ensemble matériel** ou **logiciel** qui **trie** les **paquets qui circulent** par son intermédiaire **en provenance ou vers le réseau local** et ne laisse passer que ceux qui vérifient certaines conditions.
Les noyaux **2.4 ou 2.6** intègre **netfilter** que l'**outil iptables permet** de **configurer**.

Fonctionnement de netfilter

netfilter dispose de **trois tables distinctes** donnant les **règles régissant trois types de paquets** :
- **filter** pour les règles de filtrage (accepter, refuser, ignorer un paquet)
- **nat** pour modifier les adresses IP et les ports sources ou destinataires des paquets
- **mangle** pour modifier d'autres paramètres des paquets IP (notamment le champ ToS - Type of Service - et les options)

Chaque **table contient** des **listes** de **règles appelées « chaînes »**; les chaînes standards **servent** au pare-feu **pour traiter les paquets** dans différentes circonstances prédéfinies.
La table filter compte trois chaînes standards :
- **INPUT** : concerne les paquets destinés au pare-feu
- **OUTPUT** : concerne les paquets émis par le pare-feu
- **FORWARD** : appliquée aux paquets transitant via le pare-feu (et dont il n'est donc ni la source ni le destinataire)

La table **nat** dispose également de **trois chaînes** standards :
- **PREROUTING** : modifie les paquets dès qu'ils arrivent
- **POSTROUTING** : modifie les paquets alors qu'ils sont prêts à partir
- **OUTPUT** : modifie les paquets générés par le pare-feu lui-même

Chaque **chaînes** est une **liste** de **règles prévoyant** une **action** à **exécuter**. Le **pare-feu parcourt séquentiellement** la **chaîne exécutant l'action indiquée dès** qu'une **règle** est **satisfaite**. Les **actions possibles** sont les suivantes:
- **ACCEPT** : autoriser le paquet à suivre son parcours
- **REJECT** : rejeter le paquet (ICMP signale une erreur, l'option --reject-with type d'iptables permet de choisir le type d'erreur renvoyée)
- **DROP** : supprimer, ignorer le paquet
- **LOG** : enregistrer un message de log via syslogd
- **ULOG** : enregistrer un message de log via ulogd, plus adapté et plus efficace que syslogd pour gérer de grandes quantités de messages
- **RETURN** : stopper l'évaluation de la chaîne courante et revenir sur la chaîne appelante
- **SNAT** : effectuer du Source NAT (des options précises les modifications à effectuer)
- **DNAT** : effectuer du Destination NAT
- **MASQUERADE** : effectuer du masquerading (SNAT particulier)

- **REDIRECT** : rediriger un paquet vers un port particulier du pare-feu lui-même, action notamment utile pour mettre en place un proxy web transparent

D'autres actions concernent davantage la **table mangle**.

Syntaxe d'iptables

iptables permet de manipuler les tables, les chaînes et les règles. L'**option -t table** permet **indique** la **table sur** laquelle **opérer** (**par defaut filter**)

Les commandes

Option -N chaîne crée une nouvelle **chaîne**, **-X chaîne supprime** une **chaîne vide** et **inutilisée**. **-A chaîne règle ajoute** une **règle à la fin** de la **chaîne indiquée**. **-I chaîne numrègle règle insère** une **règle avant** la **règle numérotée numrègle**. **-D chaîne numrègle ou -D chaîne règle supprime** une **règle dans** la **chaîne**. **-F chaîne supprime toutes** les **règles** de la **chaîne**. **-L chaîne affiche** le **contenu** de la **chaîne**. **-P chaîne action définit** l'**action par défaut pour** la **chaîne donnée** (pour chaînes standards)

Les règles

Chaque **règle** s'exprime **sous** la **forme conditions -j action options_de_l'action. En** les **écrivant bout à bout**, on en produit la **conjonction** (**liées par des et logique**) **donc** une **condition plus restrictive**.
La condition **-p protocole sélectionne selon** le **champ protocole** du **paquet IP** (tcp, udp, icmp...)
Préfixer le protocole **par un point d'exclamation inverse** la **condition** (soit tous les paquets n'ayant pas le protocole indiqué)
La condition **-s adresse ou -s réseau/masque vérifie l'adresse source** du **paquet**; **-d adresse ou -d réseau/masque** en **est le pendant pour l'adresse** de **destination**.
La condition **-i interface sélectionne** les **paquets** provenant **de l'interface réseau indiquée**; **-o interface sélectionne** les **paquets en fonction** de leur **interface réseau d'émission**.
La condition **-p tcp** peut être **accompagnée** de **conditions sur** les **ports TCP** avec **--source-port port et --destination-port port.**
L'option **--state état indique** le **statut** du **paquet dans** une **connexion** (le module ipt_conntrack, qui implémente le suivi des connexions, lui est nécessaire). L'état **NEW désigne** un **paquet** qui **débute** une **nouvelle connexion**. L'état **ESTABLISHED concerne** les **paquets** d'une **connexion existante** et l'état **RELATED** les **paquets** d'une **nouvelle connexion liée** à une **connexion existante** (cas des connexions **ftp-data d'une session ftp**).
L'action **LOG** dispose de **plusieurs options** visant à:

- **indiquer la priorité du message** à syslog (**--log-priority par défaut warnin**g)
- **préciser un préfixe textuel** pour différencier les messages (**--log-prefix**)
- **indiquer** les **données à intégrer dans** le **message** (**--log-tcp-sequence pour le numéro de séquence TCP, --log-tcp-options pour les options TCP et --log-ip-options pour les options IP**)

L'action **DNAT** dispose de l'**option --to-destination adresse:port** pour **indiquer** la **nouvelle adresse IP et/ou** le **nouveau port** de **destination**. De la même manière, l'action **SNAT dispose** de l'**option --to-ports ports pour indiquer** le **port ou** l'**intervalle de ports vers lesquels rediriger les paquets.**

Créer les règles

Il faut **invoquer iptables une fois par règle à créer**; c'est pourquoi **on consigne habituellement tous** les **appels à cette commande dans** un **fichier** de **script pour mettre en place la même configuration à chaque redémarrage de la machine**. On peut **écrire** ce **script à la main** mais il est souvent intéressant de le **préparer** à l'aide d'un outil de plus haut niveau tel que **fwbuilder**.
Son **principe** est **simple**. Tout d'abord il faut **décrire** tous les **éléments susceptibles d'intervenir dans** les différentes **règles** :

- le **pare-feu** et ses **interfaces réseaux**

- les **réseaux** (et **plages d'IP associées**)
- les **serveurs**
- les **ports** correspondant aux services hébergés sur les différents serveurs

On **crée** ensuite les **règles** par **glisser/déposer** des **différents objets**, quelques **menus contextuels** permettent de **modifier** la **condition** (l'inverser par exemple). Il ne reste qu'à **saisir l'action souhaitée et** à la **paramétrer**.
fwbuilder peut alors **générer** un **script** de **configuration** du **pare-feu selon** les **règles** saisies. Son **architecture modulaire** lui **permet** de **générer** des **scripts pour** les **différents pare-feu** existants (iptables, ipchains, ipf et pf) :
apt-get install fwbuilder fwbuilder-linux

Installer les règles à chaque démarrage

Si le **pare-feu** doit **protéger** une **connexion** réseau **intermittente** par PPP, le plus simple est de charger le nom du script de configuration du pare-feu et de **l'installer** sous **/etc/ppp/ip-up.d/0iptables** (le nom du script ne doit pas contenir de point). **Ainsi** il sera **rechargé à chaque démarrage** d'une connexion PPP.
Dans les **autres cas**, le plus simple est d'**inscrire** le **script** de **configuration** du **pare-feu dans** une **directive up** du **fichier /etc/network/interfaces.**

```
Exemple : Fichier interfaces avec script pare-feu
auto eth0
iface eth0 inet static
      address 192.168.0.1
      network 192.168.0.0
      netmask 255.255.255.0
      broadcast 192.168.0.255
      up /usr/local/etc/adelscott.fw
```

Réseau privé virtuel

Un **VPN** est un **moyen** de **relier par l'Internet deux réseaux locaux distants via** un **tunnel** (généralement **chiffré pour** des **raisons** de **confidentialité**). Souvent cette **technique sert** simplement **à intégrer** une **machine distante au sein** du **réseau local de l'entreprise.**
Il y a **plusieurs moyen** d'obtenir ce résultat. L'une est d'**employer SSH pour** le **tunnel chiffré** et **PPP** pour **effectuer** le **lien entre** les deux **réseaux**. Une **autre méthode** repose **sur** le **protocole IPsec**, qui **permet** de **chiffrer** les **communications IP** de **manière transparente entre deux hôtes**. Il est encore **possible** de **faire appel** au **protocole PPTP** de Microsoft.

SSH et PPP

Cette **méthode** fonctionnelle est **très simple** à mettre en oeuvre mais **pas adapté** aux **gros débits**.
Pour la mettre en place : **http://www.tldp.org/HOWTO/ppp-ssh/**
Cette méthode emploie **deux fois** le **protocole TCP** (au niveau de PPP et de SSH). Ce **double emploi pose problème à cause** de la **capacité** du **TCP à s'adapter** aux **conditions** du **réseau en variant** les **délais** de **timeout** :
http://sites.inka.de/sites/bigred/devel/tcp-tcp.html

IPsec

Avec **freeswan**, **openswan** et **racoon.Openswan** (**supplante freeswan** qui n'est plus maintenu) **apporte** seulement les **outils nécessaires** à **IPSec notamment** le **démon IKE** (IPsec Key Exchange qui **permet** l'**échange** de **clés cryptographiques entre deux hôtes avec IPsec**)

L'installation du **paquet simplifie** le **travail en créant** un **certificat X.509 ou** un **couple** de **clés RSA**.
Reste ensuite à **configurer** le **fichier /etc/ipsec.conf pour** y **paramétrer** les « **tunnels IPsec** » (security association, lire **/usr/share/doc/openswan/doc/config.html**)
Le fonctionnement d'**IPsec induit** des **échanges** de **données sur** le **port UDP 500** (et **aussi** sur le **port UDP 4500 si NAT-T,** qui encapsule le paquet IPsec dans un paquet UDP car IPsec est incompatible avec NAT à cause des signatures, est employé). De plus, les paquets IPsec utilisent deux **protocoles** IP **dédiés** que le **pare-feu doit** aussi **laisser passer** : les **protocoles 50(ESP) et 51(AH).**

PPTP

Le Protocole de Tunnel Point à Point emploie **deux canaux de communication** pour **échanger** respectivement des **informations** de **contrôle et** des **données** (protocole GRE - Generic Routing Encapsulation)

Configuration du client

avec le paquet pptp-linux, lire **http://pptpclient.sourceforge.net/howto-debian.phtml**

```
Exemple : /etc/ppp/options.pptp
# Options PPP employées pour une connexion PPTP
lock
noauth
nobsdcomp
nodeflate

Exemple : /etc/ppp/peers/adelscott
# vpn.falcot.com est le serveur PPTP
pty ''pptp vpn.falcot.com --nolaunchppd''
# la connexion s'identifiera comme utilisateur « vpn »
user vpn
remotename pptp
# la prise en charge du chiffrement est nécessaire
require-mppe-128
file /etc/ppp/options.pptp
ipparam adelscott

Exemple : /etc/ppp/ip-up.d/adelscott
# Créer la route vers le réseau local
if [ ''$6'' = ''adelscott'' ]; then
   # 192.168.0.0/24 est le réseau distant
   route add -net 192.168.0.0 netmask 255.255.0.0 dev $1
fi
Exemple : fichier /etc/ppp/ip-down.d/adelscott
# Supprimer la route vers le réseau local
if [ ''$6'' = ''adelscott'' ]; then
   # 192.168.0.0/24 est le réseau distant
   route del -net 192.168.0.0 netmask 255.255.255.0 dev $1
fi
```

La **sécurisation** de PPTP recourt à MPPE, malheureusement **pas pris en charge** par les **noyaux Debian** standards. Il faut donc compiler un **noyau spécifique** avec le **patch kernel-patch-mppe** :
http://pptpclient.sourceforge.net/howto-debian-build.phtml
ou utiliser un **noyau non officiel** à vos risques et périls :
http://pptpclient.sourceforge.net/howto-debian-prepackaged.phtml
Les **pare-feu intermédiaire** doivent **autoriser** le **protocole GRE (47) et le port 1723 du serveur PPTP** doit être ouvert.

Configuration du serveur

pptpd, il faut renseigner **localip** (adresse IP locale) et **remoteip** (adresse IP distante dans **/etc/pptpd.conf** :

```
# TAG: speed
#
#    Specifies the speed for the PPP daemon to talk at
#
speed 115200

# TAG: option
#
#    Specifies the location of the PPP options file.
#    By default PPP looks in '/etc/ppp/options'
#
option /etc/ppp/pptpd-options

# TAG: debug
#
#    Turns on (more) debugging to syslog
#
# debug
# TAG: localip
# TAG: remoteip
#
#    Specifies the local and remote IP address ranges.
#    You can specify single IP addresses separated by commas or you can
specify ranges or both. For example :
#    192.168.0.234, 192.168.0.245-249, 192.168.0.254
#
#    IMPORTANT RESTRICTIONS
#    1. No spaces are permitted between commas or within addresses.
#    2. If you give more IP addresses than MAX_CONNECTIONS, it will
start at the beginning of the list and go until it gets MAX_CONNECTIONS
IPs. Others will be ignored
#    3. No shortcuts in ranges! ie. 234-8 does not mean 234 to 238, you
must type 234-238 if you mean this
#    4. If you give a single localIP, that's ok - all local IPs will be
set to the given one. You must still give at least one remote IP for each
simultaneous client.
localip 10.1.1.199
remoteip 192.168.0.200-250
```

Il faut aussi **modifier** la **configuration PPP employée** par le serveur dans **/etc/pp/pptpd-options** comme le nom du serveur et les adresses IP des serveurs DNS et Wins :

```
# turn pppd syslog debugging on
# debug
## change 'servername' to whatever you specify as your server name in
chap-secrets
name pptp
## change the domainname to your local domain
domain upf.pf
## these are reasonable defaults for WinXXXX clients
## for the security related settings
# The Debian pppd pachage now supports both MSCHAP and MPPE, so enable
them here. Please note that the kernel support for MPPE must also be
```

```
present!
auth
require-chap
require-mschap
require-mschap-v2
require-mppe-128
## Fill in your addresses
ms-dns 192.168.0.1
ms-wins 192.168.0.1
## Fill in your netmask
netmask 255.255.255.0
## some defaults
nodefaultroute
proxyarp
lock
```

La **dernière étape** est d'**enregistrer** l'**utilisateur vpn et** le **mot de passe** associe dans **/etc/ppp/chap-secrets** :

```
# Secrets for authentication using CHAP
# client server secret IP addresses
vpn       pptp    f@Lc3au *
ADELSCOTT\\vpn pptp f@Lc3au *
```

Le **protocole version 1 présentait** un **certain nombre** de **failles** pour la **plupart corrigée** dans la dernière version, laissez absolument les options require-mppe-128 et require-mschap-v2.

Qualité de service

HOWTO du routage avancé : **http://www.linux-france.org/prj/inetdoc/guides/lartc/lartc.html**

Principe

Désigne l'ensemble des techniques permettant de **garantir** ou d'**améliorer** sensiblement la **qualité de service apportée par** des **applications**. La **plus populaire** consiste à **traiter différemment chaque type de trafic** réseau **comme** le **shaping**. Cela permet de **limiter** les **débits attribués** à certains **services et/ou** à certaines **machines**, notamment **pour** ne **pas saturer** la **bande passante**. Elle s'**adapte bien** aux **flux TCP grâce** au **débit disponible**.
On peut aussi **modifier** les **priorités** du **trafic** en permettant généralement de **traiter d'abord** les **services interactifs** (ssh, telnet) **ou ceux échangeant** des **petits blocs** de **données**.
Les **noyaux Debian intègre** le **QoS** et toute la panoplie des modules associés.

Configuration et mise en oeuvre

Le **QoS** se **paramètre avec** le logiciel **tc**, du paquet **iproute**. Son **interface** est **extrêmement complexe**.

Wondershaper

Il **minimise** les **temps** de **latence** en **limitant** le **trafic total** en **deçà** de la **valeur** de **saturation** de la ligne. Il est possible de le mettre en place par la **commande wondershaper interface debit_descendant debit_montant**. **L'interface** étant **eth0** ou **ppp0** et les **débits** s'exprimant **en kbits/s**.
La **commande wondershaper remove interface désactive** le **contrôle**.
Pour Ethernet, dans **/etc/network/interfaces** :

```
iface eth0 inet dhcp
      up /sbin/wondershaper eth0 500 100
```

```
down /sbin/wondershaper remove eth0
```

Lire **/usr/share/doc/wondershaper/README.Debian.gz pour** un **paramétrage optimal**.

Définir des priorités et des limites : shaper

Il agit **par classe correspondant** à un **type** de **trafic identifié par divers critères tel que** la **source**, **destination**, **l'horaire**, etc. Elles sont **décrites** dans **/etc/shaper** dont le **format** est **détaillé** dans **/usr/share/doc/shaper/README.shapper.gz.** Le paquet **installe** un **script** de **démarrage automatique**.

Configuration standard

En l'absence de configuration particulière, le **noyau emploie pfifo_fast**. **Pour établir** les **priorités**, elle **utilise ToS** qu'il **suffit** de **modifier pour** en **bénéficier**. Il y a **5 valeurs** :

- **Normal-Service** (0) (service normal)
- **Minimize-Cost** (2) (minimiser le coût)
- **Maximize-Reliability** (4) (maximiser la fiabilité)
- **Maximize-Troughput** (8) (maximiser le débit)
- **Minimize-Delay** (16) (minimiser le délai)

```
Exemple : amélioration du service SSH d'un serveur
iptables -t mangle -A PREROUTING -p tcp --sport ssh -j TOS --set-tos
Minimize-Delay
iptables -t mangle -A PREROUTING -p tcp --dport ssh -j TOS --set-tos
Minimize-Delay
```

Routage dynamique

Adresse routeur RENATER : 194.214.253.254
On **se connecte** sur **RENATER** puis **tracert 132.227.60.30** (pour voir la route définie)

Permet aux **routeurs** d'**ajuster** en **temps réel** les **chemins employés** des **paquets IP**. Chaque **protocole** à sa **propre définition** des **routes** (**calcul** du **chemin** le **plus court**, **récupération** des **routes annoncées**...)
Pour le **noyau Linux**, une **route associe** un **périphérique réseau à** un **ensemble de machines** qu'il peut atteindre. La commande **route permet de** les **définir et de** les **consulter**.
Le logiciel **quagga** est la **référence** en **matière** de **routage dynamique**. C'est un **ensemble de démons** qui **coopèrent** pour **définir** les **tables** de **routage**. Le démon **zebra centralise** les **informations reçues** des autres démons (bgpd, ospfd, ospf6d, ripd et ripngd) et **gère** les **tables** de **routage statiques**.
On **active** un **démon** en **modifiant** le fichier **/etc/quagga/daemons** et en **créant dans** le **répertoire /etc/quagga** son **fichier** de **configuration** qui **doit porter son nom suivi** de **.conf** et **appartenir** à l'**utilisateur quagga et** au **groupe quaggavty**. **lire** le manuel **info quagga-doc** ou **http://www.quagga.net/docs/docs-info.php** ainsi que **http://perso.wanadoo.fr/pmassol/main.html**

IPv6

Sucesseur d'**IPv4**, il **doit** en **corriger** les **défauts et** notamment le **nombre** trop **faible** d'**adresses IP** existantes. Les **outils de base** comme **ping** et **traceroute** ont **pour équivalent ping6** et **traceroute6**, **disponible dans** les paquets Debian **iputils-ping** et **iputils-tracepath**.
On peut **configurer IPv6 comme** un réseau **IPv4 à travers /etc/network/interfaces**. Pour **ne pas se contenter** d'un **réseau IPv6 privé**, il **faut** cependant un **routeur capable de relayer** le **trafic sur** le **6bone**, **réseau public expérimental** employant **IPv6**.

```
Exemple :
iface th0 inet6 static
address 3ffe:ffff:1234:5::1:1
```

```
netmask 64
# Pour désactiver l'auto-configuration
# up echo 0 > /proc/sys/net/ipv6/conf/all/autocon
# le routeur est auto-configuré et n'a pas d'adresse fixe
# (/proc/sys/net/ipv6/conf/all/accept_ra). Sinon pour le forcer :
# gateway 3ffe:ffff:1234:5::1
```

En l'absence de **point** de **connexion** au **6bone**, on **peut** toujours **s'y connecter via** un **tunnel IPv4**. **Freenet6** est un **fournisseur gratuit de** tels **tunnels**:
http://www.freenet6.net
Il faut **installer** alors le paquet Debian **freenet6** et **configurer** ce **tunnel** dans **/etc/freenet6/tspc.conf** (**userid** et **password**). Pour **proposer** une **connectivité** IPv6 à **tout** le **réseau local** (connecté à eth0) :

```
host_type=router
prefix_len=48
if_prefix=eth0
```

La **machine** est alors le **routeur d'accès à** un **sous réseau dont** le **préfixe** fait **38 bits**. **Sur** le **réseau local**, il faut alors **installer** le **démon radvd**, **démon** de **configuration IPv6 jouant** le **même rôle** que **dhcpd** pour IPv4.
Il faut ensuite **créer** son **fichier** de **configuration /etc/radvd.conf** (voir **/usr/share/doc/radvd/examples/simple-radvd.conf**). En l'occurrence **changer** le **préfixe par celui** de **Freenet6** (que l'on retrouve avec **ifconfig** dans le bloc relatif à sit1).
Pour **rendre** le **réseau fonctionnel**, **/etc/init.d/freenet6 restart** et **/etc/init.d/radvd restart**.
Programmes compilés avec **IPv6** : **http://debian.fabbione.net**
IPv6 et **pare-feu** : adaptation de **netfilter** pour l'**IPv6** qui se **configure** avec **ip6tables**.

Serveur de noms (DNS)

Principe

Le **service** de **gestion** des **noms** permet d'**associer** des **noms à** des **adresses IP** (et vice versa).
exemple : www.yahoo.fr au lieu de 217.12.3.11

Les **informations DNS** sont **regroupées par zones**, **correspondant** chacune **à un domaine ou** à une **plage d'adresses IP**. Un **serveur primaire fait autorité sur** le **contenu** d'une **zone**; un **serveur secondaire**, normalement **hébergé sur** une **autre machine**, se contente de **proposer** une **copie** de la **zone primaire**, qu'il **met à jour régulièrement**.
Chaque zone peut **contenir différents types d'enregistrements** (Resource Records):
. **A** : attribution d'une adresse IPv4
. **CNAME** : définition d'un alias
. **MX** : définition d'un serveur de courrier électronique, employé par le serveur de messagerie pour retrouver le serveur correspondant à l'adresse de destination d'un courrier électronique. Chaque serveur MX a une priorité associée. Le serveur de plus haute priorité (portant le nombre le plus petit) recevra les connexions SMTP. S'il ne répond pas, le deuxième sera contacté, etc.
. **PTR** : correspondance adresse IP vers nom. Elle est stockée dans une zone dédiée à la résolution inverse nommée en fonction de la plage d'adresses IP : par exemple, 1.168.192.in-addr.arpa pour toutes les adresses 192.168.1.0/24
. **AAAA** : correspondance nom vers adresse IPv6
. **NS** : correspondance nom vers serveur de noms
Chaque domaine doit en **compter** au **moins un**. Tous ces **enregistrements pointent** sur un **serveur DNS** capable de **répondre** aux **requêtes** portant sur ce **domaine**; ils **signaleront** les **serveurs primaires** et

secondaires du domaine concerné. Permet aussi de **mettre en place** une **délégation DNS**. On peut aussi **indiquer** que le **domaine interne.upf.pf** est **géré par** un autre **serveur** de **noms** et **délégué** ainsi une **partie du service**. Évidemment, le **serveur concerné** devra **déclarer** une **zone interne** interne.upf.pf.
Le **logiciel** de **référence Bind** est **proposé** dans les paquets Debian **bind** (version 8) et **bind9** (version 9). Cette **dernière** permet d'**employer** le **serveur** sous une **identité utilisateur non privilégiée**. D'autre part, elle **prend en charge DNSSEC**, qui **permet** de **signer et** d'**authentifier** les **enregistrements DNS** (voir **http://www.afnic.fr/afnic/r_d/dnssec)**

Configuration

On peut utiliser le module **webmin-bind**. Pour **tester** le **serveur** DNS, utilisez la **commande host**. La **commande host machine.upf.pf localhost contrôle** la **réponse** du **serveur DNS local pour** la **requête machine.upf.pf**.
La **zone inverse** couvrant le **réseau 192.168.0.0/16 s'appellera 168.192.in-addr.arpa**
La **syntaxe désignant** les **noms** de **machine** est **particulière**. **machine sous entend** ainsi **machine.domaine.** Il **convient d'écrire machine.** (en suffixant ce nom d'un point). Pour **indiquer** un **nom** DNS **extérieur**, on **écrira** donc **machine.autredomaine.com. (avec un point final).**

```
Exemple : /etc/bind/named.conf.local
zone ''test.com'' {
      type master;
      file ''/etc/bind/db.test.com'';
      allow-query { any; };
      allow-transfer {
            195.20.195.149/32 ; // ns0.xname.org
            193.23.158.13/32 ; // ns1.xname.org
      };
};

zone ''interne.test.com'' {
      type master;
      file ''/etc/bind/db.interne.test.com'';
      allow-query { 192.168.0.0/16; };
};

zone ''168.192.in-addr.arpa'' {
      type master;
      file ''/etc/bind/db.192.168'';
      allow-query { 192?168?0?0/16; };
};

zone ''168.192.in-addr.arpa'' {
      type master;
      file ''/etc/bind/db.192.168'';
      allow-query { 192.168.0.0/16; };
};


Exemple: /etc/bind/db.test.com
; zone test.com
; admin.test.com => contact pour la zone: admin@test.com
$TTL 604800
@     IN    SOA   test.com   admin.test.com. (
                  20040121   ; Serial
                     604800  ; Refresh
                       86400 ; Retry
```

```
                  2419200  ; Expire
                   604800  ; Negative Cache TTL
;
; Le @ fait référence au nome de la zone (« test.com » en l'occurence)
; ou à $ORIGIN si cette directive a été employée
;
@     IN    NS    ns
@     IN    NS    ns0.xname.org

interne IN NS    192.168.0.2

@     IN    A     212.94.201.10
@     IN    MX    5 mail
@     IN    MX    10 mail2
ns    IN    A     212.94.201.10
mail  IN    A     212.94.201.10
mail2 IN    A     212.94.201.11
www   IN    A     212.94.201.11
dns   IN    CNAME ns

Exemple: /etc/bind/db.192.168
; Zone inverse pour 192.168.0.0/16
; admin.test.com => contact pour la zone: admin@test.com
$TTL  604800
@     IN    SOA   ns.interne.test.com  admin.test.com. (
                  20040121        ; Serial
                     604800       ; Refresh
                       86400      ; Retry
                    2419200       ; Expire
                     604800  )          ; Negative Cache TTL
      IN    NS    ns.interne.test.com
; 192.168.0.1 -> arrakis
1.0   IN    PTR   arrakis.interne.test.com.
; 192.168.0.2 -> neptune
2.0   IN    PTR   neptune.interne.test.com.
; 192.168.3.1 -> pau
1.3   IN    PTR   pau.interne.test.com.
```

DHCP

Présentation

C'est un **moyen** de **rapatrier automatiquement** la **configuration** de la **carte réseau d'une machine**. On peut **centraliser** la **gestion** des **configurations réseau pour que toutes** les **machines reçoivent** des **réglages identiques**. Paquet Debian : **dhcp3-server**.

Configuration

Dans **/etc/dhcp3/dhcpd.conf** avec le **nom de domaine et** les **serveurs DNS**. Il faut aussi **activer** l'**option authoritative** si ce **serveur** est le **seul** sur le **réseau local**. On créera aussi une **section subnet décrivant** le **réseau local et** les **informations** de **configuration**. L'**exemple ci-dessous convient pour** le **réseau** local **192.168.0.0/24** qui **dispose** d'un **routeur** (**192.168.0.1**) **faisant** office de **passerelle externe**. Les **adresses IP disponibles sont** comprises **entre 192.168.0.128 et 192.168.0.254**.

Exemple :

```
# Sample configuration file for ISC dhcpd for Debian
```

```
# The ddns-updates-style parameter controls whether or not the server
will attempt to do a DNS
# update when a lease is confirmed. We default to the behavior of the
version 2 packages ('none',
# since DHCP v2 didn't have support for DDNS.)
ddns-update-style interim;
# option definitions common to all supported networkds...
option domain-name ''interne.test.com'';
option domain-name-servers ns.interne.test.com;
default-lease-time 600;
max-lease-time 7200;
# If this is the official DHCP server must be set
authoritative;
# Use this to send dhcp log messages to a different log file
log-facility local7;
# My subnet
subnet 192.168.0.0 netmask 255.255.255.0 {
      option routers 192.168.0.1;
      option broadcast-address 192.168.0.255;
      range 192.168.0.128 192.168.0.254;
      ddns-domainname ''interne.test.com'';
}
```

DHCP et DNS

Option utile: l'**enregistrement automatique** des **clients** DHCP **dans** la **zone DNS** de sorte que **chaque machine** est un **nom significatif** (et pas automatique comme machine-192-168-0-131.interne.test.com). Pour exploiter cette possibilité, il faut **autoriser** le **serveur DHCP** à **mettre à jour** la **zone DNS interne.test.com** et configurer celui-ci pour qu'il s'en charge.
Dans le cas de **bind**, on ajoutera la **directive allow-update** aux **deux zones** que le serveur DHCP devra modifier (celle du domaine **interne.test.com** et celle de la **résolution inverse**). Cette **directive** donne la **liste** des **adresses autorisées à effectuer la mise à jour**; on y consignera donc les adresses possibles du serveur DHCP (adresses IP locales et publiques le cas échéant).

```
allow-update { 127.0.0.1 192.168.0.1 212.94.201.10 !any };
```

L'exemple de configuration de serveur DHCP de la section précédente comporte déjà les **directives nécessaires** à l'**activation** de la **mise à jour du DNS**: **ddns-update-style interim et ddns-domain-name ''interne.test.com'' dans le bloc décrivant le réseau.**

Détection d'intrusion

snort est un outil de **détection d'intrusions** : il **écoute** en permanence le **réseau** afin de **repérer** les **tentatives** d'infiltration et/ou les **actes malveillants** (notamment les dénis de service). Il faut **préciser** la **plage d'adresses couvertes** afin d'indiquer toutes les cibles potentielles d'attaques. Il faut aussi **préciser l'interface réseau à surveiller** (eth0...). Dans **/etc/snort/snort.conf,** il est possible d'**indiquer** les **machines hébergeant chaque service** pour limiter le nombre d'incidents rapportés par snort (un déni de service sur une machine bureautique n'est pas aussi dramatique que sur un serveur DNS). On peut encore y **renseigner** les **correspondances** entre **adresses IP et MAC** (pour détecter les **ARP-spoofing**). Branché sur un commutateur, il ne surveillera que les attaques ciblant la machine l'hébergeant. Pensez donc à **relier** la **machine** au **port « miroir »** (qui **permet** de **chaîner** les **commutateurs**) sur **lequel** tout le **trafic** est **dupliqué**.
Un **autre NIDS** employant Snort : **Prelude** qui permet d'employer un **HIDS performant** sur chaque serveur ainsi que de **remonter** les **alertes Snort** dont il fera lui-même le **filtrage par niveau critique.**

Serveur de messagerie électronique

Le serveur **Exim** est **installé par défaut** sous Debian, on **préfère Postfix**.
Postfix, lire **http://x.guimard.free.fr/postfix/.** Le paquet **Postfix contient** le démon **SMTP** principal. Divers **modules** (comme **postfix-ldap ou postfix-pgsql**) existent. Au cours de l'installation, Debconf va générer **/etc/postfix/main.cf** en fonction de vos réponses. La première porte sur le type d'installation, deux réponses sont pertinents dans le cadre d'un serveur connecté à l'Internet: **« Site Internet » ou « Internet par un FAI »**. Le premier est adapté à un serveur qui envoie et reçoit du courrier directement à ses destinataires. Le deuxième correspond à un serveur qui reçoit directement le courrier mais envoie par l'intermédiaire d'un serveur SMTP intermédiaire désigné par le terme smarthost. C'est surtout utile pour les particuliers disposant d'une adresse IP dynamique parce que certains serveurs de messagerie refusent tout message provenant directement d'une telle adresse IP. Le smarthost ici sera le serveur SMTP du FAI qui est toujours configuré pour transmettre le courrier provenant de ses clients.
La deuxième question porte sur le **nom complet de la machine employé** pour générer une adresse de courrier électronique depuis un nom d'utilisateur local (c'est la partie suivant l'arobase '@', upf.pf). Ensuite il faut indiquer les **noms de domaine associés** à cette machine, il faut ajouter upf.pf manuellement. D'une manière générale, il faut indiquer tous les noms de domaine pour lesquels cette machine fait office de **serveur MX (tous ceux pour lesquels le DNS indique qu'elle est apte à accepter du courrier**). Ces informations sont ensuite stockées dans la variable **mydestination** du fichier **/etc/postfix/main.cf**.
Par défaut, **Postfix** est **configuré** pour **accepter** les **courriers électroniques** issus de la **machine elle-même**, il faut donc **indiquer** le **réseau local (ex: 192.168.0.0/16)** dans la variable **mynetworks**.
Procmail peut être proposé. Cet **outil** permet d'**indiquer** de **règles de tri** dans ~/.procmail dans le dossier des utilisateurs.

```
Exemple: /etc/postfix/main.cf initial
# version
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff=no
# appending .domain is the MUA's job
append_dot_mydomain = no
# Uncomment the next line to generate ''delayed mail'' warnings
#delay_warning_time = 4h

myhostname = localhost
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = mail.upf.pf, upf.pf, localhost, localhost.localdomain
relayhost =
mynetworks = 127.0.0.0/8 192.168.0.0/16
mailbox_command = procmail -a ''$EXTENSION''
mailbox_size_limit = 0
recipient_delimiter = +
```

Configuration de domaines virtuels

Le serveur de **messagerie principal** peut **recevoir** le **courrier** pour d'**autres domaines** que le principal.
Aucun domaine virtuel ne doit être indiqué dans **mydestination**. Celle ci contient uniquement les **domaines canoniques** directement **associés** à la **machine** et ses **utilisateurs locaux**.

Domaine virtuel d'alias

Ne contient que des **alias**, cad **des adresses électroniques renvoyant** le **courrier** vers d'**autres**

adresses électroniques. Pour l'activer, il faut **préciser** son **nom** dans **virtual_alias_domains** et **indiquer** le **fichier stockant** les correspondances dans **virtual_alias_maps**.

```
Exemple :
virtual_alias_domains = exemple.tm.fr
virtual_alias_maps = hash:/etc/postfix/virtual
```

Le fichier **virtual** est simple. Chaque **ligne** contient **deux champs** séparés par des blancs dont le **premier** est le **nom de l'alias** et le **second** une **liste d'adresses électroniques** vers lesquelles il pointe. La syntaxe ''@domaine.fr'' englobe tous les alias restants d'un domaine.

```
Exemple :
webmaster@exemple.tm.fr jean@upf.pf
contact@exemple.tm.fr laure@upf.pf, sophie@upf.pf
# Cet alias est générique, il englobe toutes les adresses electroniques
du domaine exemple.tm.fr.
# Ces adresses sont renvoyés au nom d'utilisateur mais pour le domaine
correspondant (upf.pf)
@exemple.tm.fr @upf.pf
```

Domaine virtuel de boîtes aux lettres

On **ne peut indiquer** le **même domaine dans** les **variables virtual_alias_domains et virtual_mailbox_domains**. Par contre, **tout domaine virtual_mailbox_domains** est **implicitement compris** dans **virtual_alias_domains**. Il est donc **possible** de **mélanger alias** et **boîtes aux lettres** au sein d'un domaine virtuel.

Les **courriers destinés** à un **domaine virtuel** sont **stockés** dans des **boîtes aux lettres** qui ne sont **pas associées** à un **utilisateur local** du système.

Pour **l'activer**, il faut l'écrire dans **virtual_mailbox_domains** et **préciser** le **fichier** donnant les **boîtes aux lettres avec virtual_mailbox_maps**. Le paramètre **virtual_mailbox_base** indique le **répertoire sous lequel** les **différentes boîtes** sont **stockées**.

Les **paramètres virtual_uid_maps** et **virtual_gid_maps définissent** des **tables de correspondances** entre l'**adresse électronique**, l'**utilisateur** et le **groupe Unix propriétaire** de la boite aux lettres. Pour indiquer **systématiquement** le **même propriétaire**, la syntaxe **static:5000** dénote un **UID/GID fixe**.

```
Exemple :
virtual_mailbox_domains = upf.org
virtual_mailbox_maps = hash:/etc/postfix/vmailbox
virtual_mailbox_base = /var/mail/vhosts
```

Format de **vmailbox** : **adresse électronique** de l'**un des domaines** puis **emplacement relatif** de la **boite associé** (par rapport à **virtual_mailbox_base**). Si le **nom** se **termine par /**, format **maildir** (**sinon mailbox**). **Maildir** emploie un **répertoire complet** pour **représenter** la **boite aux lettres**, **mbox** la **stocke** dans **un seul fichier** (chaque **ligne débutant** par **''From'' suivi** d'un **espace marque** le **début** d'un **nouveau message électronique**)

```
Fichier : /etc/postfix/mailbox
# le courrier de jean est stocké au format maildir
jean@upf.org upf.org/jean/
# le courrier de sophie au format mbox
sophie@upf.pf upf.pf/sophie
```

Restrictions à la réception et à l'envoi

Restreindre l'accès en fonction de l'adresse IP

la directive **smtpd_client_restrictions contrôle** les **machines autorisées** à communiquer avec le serveur de courrier électronique.

```
Exemple : Restrictions en fonction de l'adresse du client
smtpd_client_restrictions = permit_mynetworks, warn_if_reject,
reject_unknown_client, check_client_access,
hash:/etc/postfix/access_clientip, reject_rbl_client sbl-
xbl.spamhaus.org, reject_rbl_client list.dbsl.org
```

Cette **liste de règle** est **évaluée dans l'ordre** (de la première à la dernière). Chacune peut accepter, **refuser** ou **laisser** poursuivre le message. **L'inversion** de 2 règles peut mettre en place un **comportement très différent**.
permit_mynetwork accepte inconditionnellement **toute machine du réseau local**.
La **deuxième refuse** normalement les machines **dépourvues** de **configuration DNS** totalement **valide** (valide : **résolution inverse fonctionnelle et nom DNS pointant sur l'IP**). Cette **restriction** est assez **élevé**, de nombreux serveurs ne disposant pas de DNS inverse (d'où la **directive warn_if_reject** qui **transforme** le **refus** en simple **avertissement** enregistré dans les logs).
La **troisième** permet de mettre en place une **liste blanche de serveurs** de courriers électroniques stockés dans **/etc/postfix/access_clientip** (qui sera dispensé des règles suivantes).
Les **deux dernières régles refusent** tout **message** provenant d'un **serveur** présent dans l'une des différentes **''listes noires''** (RBL pour Remote Black Lists).
Table access : exemple de table (avec critéres de restriction) dans **/etc/postfix/access** (documenté).
access_clientip : liste des adresses IP et réseau
access_helo : noms de machines et domaines
access_sender : précise les adresses électroniques
après modifications, en faire une table de hachage avec la commande : **postmap /etc/posfix/fichier**.

```
Vérifier la validité de la commande EHLO ou HELO
Exemple : Restrictions sur le nom annoncé lors du EHLO
smtpd_helo_restrictions = permit_mynetworks, reject_invalid_hostname,
check_helo_access hash:/etc/postfix/access_helo,
reject_non_fqdn_hostname, warn_if_reject reject_unknown_hostname
```

La règle **reject_invalid_hostname** refuse tout courrier dont l'annonce EHLO indique un nom de machine syntaxiquement incorrect. La règle **reject_non_fqdn_hostname** refuse tout message dont le nom de machine annoncé n'est pas complètement qualifié (cad avec nom de domaine). La règle **reject_unknown_hostname** refuse le courrier si la machine n'est **pas dans** la **base du DNS**. Elle est **atténuée par warn_if_reject** pour évaluer son impact. **Permit_mynetworks** au début **permet** de n'**appliquer les règles suivantes qu'à des machines extérieures au réseau local**. Il est ainsi **possible** de **mettre** en **liste noire** tous **ceux qui s'annoncent membres du réseau upf.pf**... ce qui s'effectue en ajoutant la **ligne upf.pf REJECT You're not in our network!** au fichier **/etc/postfix/access_helo**.

Accepter ou refuser en fonction de l'émetteur (annoncé)

Chaque **message** est **associé** à un **expéditeur** grâce à : **MAIL FROM** du **protocole SMTP**.

```
Exemple : vérifications sur l'expéditeur
smtpd_sender_restrictions =
check_sender_access hash:/etc/postfix/access_sender,
reject_unknown_sender_domain, reject_unlisted_sender,
reject_non_fqdn_sender
```

La table **access_sender** associe des **traitements particuliers** à certains **expéditeurs** (en général liste blanche ou noire).
La règle **reject_unknown_sender_domain requiert** un **domaine d'expéditeur valide**, **nécessaire** à **une adresse valide**. **Reject_unlisted_sender refuse** les **expéditeurs locaux si** leur **adresse n'existe pas**. **Personne ne peut envoyer** de **courrier issu** d'une **adresse invalide dans** le **domaine upf.pf**.

Enfin, **reject_non_fqdn_sender refuse** les **adresses dépourvues** de **domaine complètement qualifié**.

Accepter ou refuser en fonction du destinataire (annoncé)

Chaque **courrier** compte **un** ou **plusieurs destinataires** communiqué **grâce à RCP TO**.

```
Exemple : vérification sur le destinataire
smtpd_recipient_restrictions = permit_mynetworks,
reject_unauth_destination, reject_unlisted_recipient,
reject_non_fqdn_recipient
```

reject_unauth_destination est la **règle** de base **imposant** à **tout courrier provenant** de l'**extérieur de nous être déstiné**. **Sans** cette **règle**, votre **serveur** est un **relais ouvert** (**exploitable par** les **spammeurs**).
reject_unlisted_recipient refuse les **messages** à **destination d'utilisateurs locaux inexistants**.
reject_non_fqdn_recipient refuse les **adresses électroniques non qualifiées**.

Restrictions associées à la commande DATA

Elle **précède** l'**envoi** des **données contenues dans** le **message**.

```
Exemple :
smtpd_data_restrictions = reject_unauth_pipelining
```

Cette **règle refuse** le **message si** le **correspondant envoie** une **commande sans avoir attendu** la **réponse à la commande précédente** (les **spammers n'attendent** généralement **pas** de **réponse pour aller plus vite)**.

Application des restrictions

Le **refus réel** n'est **signifié par Postfix que lors** de la **réponse à** la **commande RCPT TO** (annonce du destinataire).

Filtrer en fonction du contenu du message

Le **système** de **vérification** ne serait **pas complet sans moyen de réagir au contenu**. **2 types** de **vérification** existent : sur les **en-têtes et** sur le **corps**.

```
Exemple : activation des filtres de contenu
header_checks = regexp:/etc/postfix/header_checks
body_checks = regexp:/etc/postfix/body_checks
```

Les **2 fichiers contiennent** une **liste d'expressions rationnelles**(regexp). **Chacune** est **associé à** une **action à exécuter**.

```
Exemple : /etc/postfix/header_checks
/^X-Mailer: GOTO Sarbacane/ REJECT I fight spam (GOTO Sarbacane)
/^Subject: * Your email contains VIRUSES/ DISCARD virus notification
```

La **première vérifie** l'**entête** en **indiquant** le **logiciel de courrier électronique envoyé**: si elle trouve **GOTO Sarbacane** (un logiciel d'envoi en masse de courriers), elle refuse le message.
La **seconde inspecte** le **sujet s'il indique** une **notification de virus sans intêret**, elle **accepte** le **message mais** le **supprime immédiatement**.

Intégration d'un antivirus

L'**antivirus libre** est **clamav** (**paquet clamav, arj, unzoo, unrar** et **lha**, ces derniers pour les archives).
Pour **interfacer** cet **antivirus** au **serveur** de **messagerie** : **amavisd-new**, mini-serveur qui **écoute** le **port 10024**. Les **courriers reçus** sont **analysés** (**paramétrage** du **comportement** dans

/etc/amavis/amavisd.conf).
Après l'**installation** de **amavisd-new**, il faut **ajouter** l'**utilisateur clamav** au **groupe amavis** pour qu'**amavisd** puisse **piloter clamav**.
Ensuite **paramètrez /etc/amavis/amavisd.conf** et y **indiquer** le **nom** de **domaine principal** et la **liste** des **domaines locaux** (y compris virtuels).

```
Exemple :
$mydomain = ''upf.pf''
@local_domains_acl = (''.$mydomain'', ''.exemple.tm.fr'', ''.upf.org'')
Ensuite il faut préciser la manière de faire suivre les messages :
# where to forward checked mail
$forward_method = 'smtp:127.0.0.1:10025';
# where to submit notifications
$notify_method = $forward_method;
```

Le **paramètre $banned_filename_refuse refuse** par défaut les **fichiers joints dotés** d'une **double extension** (**comme document.doc.pif**). On peut encore **durcir** cette **règle** en **interdisant** tout **exécutable**.
Les **paramètres *_lovers limitent** les **vérifications effectuées** en **fonction** des **destinataires**.
Le **comportement par défaut** d'**amavisd** est d'**envoyer** un **message** au **postmaster** pour **signaler** chaque **message mis en quarantaine** pour cause de virus. On peut **supprimer** ces **notifications** en **commentant** la ligne contenant **$virus_admin**.

Configuration de Postfix avec l'antivirus

Il faut **configurer Postifix pour** lui faire **relayer** le **courrier** à **amavis**, **qui** lui **renverra par** un **autre canal** (**port 10025**). Les **instructions** de **configuration** sont **détaillées** dans **/usr/share/doc/amavisd-new/README.postfix.gz**.
On **modifiera** le **fichier /etc/postfix/master.cf** en ajoutant :

```
smtp-amavis unix - - n - 2 1mtp
      -o lmtp_data_done_timeout=1200
      -o lmtp_send_xforward_command=yes
      -o disable_dns_lookups=yes
127.0.0.1:10025 inet n -n - - smtpd
      -o content_filter=
      -o receive_override_options=no_header_body_checks,
       no_unknown_recipient_checks
      -o local_recipient_maps=
      -o relay_recipient_maps=
      -o smtpd_restriction_classes=
      -o smtpd_client_restrictions=
      -o smtpd_helo_restrictions=
      -o smtpd_sender_restrictions=
      -o smtpd_recipient_restrictions=permit_mynetworks, reject
      -o mynetworks=127.0.0.0/8
      -o strict_rfc821_envelopes=yes
      -o smtpd_error_sleep_time=0
      -o smtpd_soft_error_limit=1001
      -o smtpd_hard_error_limit=1000
```

Il faut encore **dévier** les **courriers entrants vers amavis**.

```
Exemple : Activation du filtre extérieur sur le contenu
content_filter = smtp-amavis:[127.0.0.1]:10024
```

Les **messages traités** par **Postfix passent** désormais **par** un **détecteur-filtre antivirus**.

Serveur web(HTTP)

Liste complète des **modules** d'Apache : **http://httpd.apache.org/docs/mod/index.html**
Installation du paquet **apache provoque** de **nombreuses questions** de configuration.
suExec est une **option utile lorsque** des **inconnus** peuvent **exécuter** des **scripts CGI**: les **droits** sont alors **limités** à **ceux** de leur **propre compte** (au lieu de www-data).
Ensuite, **choisissez** les divers **modules voulues**. Il est toujours **possible** d'en **activer** d'**autres plus tard** avec la **commande modules-config apache**.
Entrez le **nom du serveur** (www.upf.pf), l'**adresse du webmaster** (webmaster@upf.pf), le **répertoire racine** du site web (**/var/www par défaut**) et le **port** sur lequel **Apache attend** les **connexions** (**80 par défaut**).

Prise en charge SSL

Il faut installer le **paquet libapachemod-ssl** qui propose de **créer** un **certificat**. Les changements de **configuration** étant quelque peu **complexes**, il faut suivre les instructions données dans **/usr/share/doc/libapache-mod-ssl/README.Debian.**

Configuration d'hôtes virtuels

C'est une **identité supplémentaire assumée** par le **serveur Web**. Il existe **deux types d'hôtes virtuels** : **ceux** qui se **basent** sur l'**adresse IP** et **ceux** qui **reposent sur** le **nom DNS** du **serveur Web**. La **première nécessite** une **adresse IP différente** pour **chaque site** tandis que la **seconde différencie** les **sites par** le **nom d'hôte communiqué par** le **client HTTP** (ce qui ne fonctionne qu'avec la version 1.1 du protocole HTTP déjà employée par tous les navigateurs web).
La **rareté** des **adresses IPv4 fait** en général **privilégier** cette **deuxième méthode**. **Elle est** cependant **impossible** si **chacun** des **hôtes virtuels** a **besoin** de **HTTPS**.
Pour **mettre en place** des **hôtes virtuels basés sur** le **nom**, il faut **insérer dans /etc/apache/httpd.conf**, **juste avant** la **directive incluse /etc/apache/conf.d. :**

```
NameVirtualHost *
<VirtualHost *>
ServerName www.upf.pf
ServerAlias upf.com
DocumentRoot /srv/www/www.upf.com
</VirtualHost>
```

Chaque hôte virtuel **supplémentaire est** ensuite **décrit par** un **fichier placé dans** le **répertoire /etc/apache/conf.d**. **Ainsi** la **mise en place** du **domaine falcot.org se résume** à **créer** le **fichier ci-dessous.**

```
Exemple : /etc/apache/conf.d/www.upf.org.conf
<VirtualHost *>
ServerName www.upf.org
ServerAlias upf.org
DocumentRoot /srv/www/www.upf.org
</VirtualHost>
```

Le **serveur Apache** est ici **configuré pour** n'**utiliser** qu'**un seul fichier** de **log par hôte** virtuel (ce qu'on peut **modifier** en utilisant la **directive CustomLog**). On peut **personnaliser** son **format** avec **LogFormat.**

```
Exemple :
# Nouveau format de log avec virtual host (vhost)
LogFormat ''%v %h %1 %u %t \''%r\'' %>s %b \''%{Referer}i\'' \''%{User-
Agent}i\'' '' vhost
# On emploie le format vhost en standard
CustomLog /var/log/apache/access.log vhost
```

```
Directives courantes
Exemple : Bloc Directory
<Directory /var/www>
Options Includes FollowSymlinks
AllowOverride All
DirectoryIndex index.php index.html index.htm
</Directory>
```

La directive **DirectoryIndex** précise la **liste** des **fichiers à essayer** pour **répondre** à une **requête sur** un **répertoire**. Le **premier fichier** existant est **appelé pour générer** la **réponse**.
La directive **Options** est **suivie** d'une **liste d'options à activer**. L'**option None désactive toutes** les **options**. **Inversement**, l'**option All** les **active toutes sauf MultiViews**. **Voici** les **options existantes** :

- **ExecCGI** indique qu'il est **possible** d'**exécuter** des **scripts CGI**
- **FollowSymlinks** indique qu'il doit **suivre** les **liens symboliques** et donc **effectuer** la **requête sur** le **fichier réel** qui en est la cible
- **SymlinksIfOwnerMatch** a le **même rôle** mais **impose** la **restriction supplémentaire** de ne **suivre** le **lien** que **si** le **fichier pointé appartient** au **même propriétaire**
- **Includes active** les **inclusions côté serveur** (Server Side Includes ou SSI). Il s'agit de **directives directement intégrées dans** les **pages HTML** et **exécutées** à la volée **à chaque requête**
- **Indexes autorise** le **serveur** à **retourner** le **contenu** du **dossier si** la **requête HTTP pointe sur** un **répertoire dépourvu** de **fichier d'index** (**tous** les **fichiers** de la **directive DirectoryIndex** ayant été tentés en vain)
- **MultiViews active** la **négociation** de contenu, ce qui **permet** au **serveur** de **renvoyer** la **page Web** correspondant à la langue annoncée par le navigateur.

La directive **AllowOverride donne toutes** les **options** qu'on peut **activer** ou **désactiver par** l'intermédiaire d'un **fichier .htaccess**. Il est **souvent important** de **contrôler** l'option **ExecCGI pour rester maître** des **utilisateurs autorisés** à **exécuter** un **programme au sein** d'un **serveur Web** (**sous** l'identifiant **www-data**).
Le **fichier .htaccess contient** des **directives** de **configuration** d'**Apache prises en compte** à **chaque fois** qu'une **requête concerne** un **élément** du **répertoire où il est stocké**. Sa **portée embrasse** également les **fichiers** de **toute l'arborescence** qui en est **issue**.
La **plupart** des **directives** d'un bloc **Directory** peuvent se **trouver dans** un fichier **.htaccess**.

Requérir une authentification

Il est nécessaire de **restreindre** l'**accès à une partie** d'un **site**. Les **utilisateurs légitimes doivent** alors **fournir** un **identifiant et** un **mot de passe** pour **accéder** à son **contenu**.

```
Exemple : fichier .htaccess requérant une authentification
Require valid-user
AuthName ''Répertoire privé''
AuthType Basic
AuthUserFile /etc/apache/authfiles/htpasswd-prive
```

Le **fichier /etc/apache/authfiles/htpasswd-prive contient** la **liste** des **utilisateurs et** leur **mots de passe**; **on** le **manipule avec** la **commande htpasswd**. Pour **ajouter**, un **utilisateur** ou **changer** un **mot de passe**, on **exécutera** la **commande** suivante:

```
# htpasswd /etc/apache/authfiles/htpasswd-prive utilisateur
New password:
Re-type new password:
Adding password for user utilisateur
```

Ce **système d'authentification** (Basic) a une **sécurité très faible**, **puisque** les **mots de passe circulent**

sans protection (uniquement codés en **base64**). Il est donc **préférable** d'**utiliser SSL.**

Restrictions d'accès

Les directives **Allow from** (**autoriser** en **provenance de**) et **Deny from** (**refuser** en **provenance de**), qui **s'appliquent** à un **répertoire et** à **toute l'arborescence** qui en est **issue**, **paramètrent** les **restrictions d'accès**.
La directive **Order indique** dans quel **ordre évaluer** les **directives Allow from et Deny from**. Concrètement, **Order deny, allow autorise l'accès si aucune** des **règles Deny from ne s'applique**. Inversement, **Order allow, deny refuse l'accès si aucune directive Allow from ne l'autorise**.
Les directives **Allow from et Deny from peuvent être suivies** d'une **adresse IP**, **d'un réseau** (ex: 192.168.0.0/255.255.255.0,192.168.0.0/24 et même 192.168.0), d'un **nom** de **machine** ou de **domaine** ou du **mot clé all** désignant tout le monde.

```
Exemple : interdire par défaut mais autoriser le réseau local
Order deny,allow
Allow from 192.168.0.0/16
Deny from all
```

Analyseur de logs

AWStats (Advanced Web Statitics ou statistiques Web avancées) **pour analyser** les **fichiers** de log d'**Apache**. La **première étape** de la configuration consiste à **créer** le **fichier /etc/awstats/awstats.conf**.
Il est recommandé **d'adapter** le **modèle /usr/share/doc/awstats/examples/awstats.model.conf.gz**.

```
Exemple :
LogFile=''/var/log/apache/access.log''
LogFormat = ''%virtualname %host %other %logname %time1 %methodur1 %code
%bytesd %refererquot %uaquot''
SiteDomain=''www.upf.pf''
HostAliases=''upf.pf REGEX[^.*\.upf\.pf$]''
DNSLookup=1
DirData=''/var/lib/awstats''
DirIcons=''/awstats-icon''
DirLang=''/usr/share/awstats/lang''
LoadPlugin=''tooltips''
```

LogFile et **LogFormat indique** l'**emplacement** du **fichier** de **log et** les **informations qu'il contient**. **SiteDomain** et **HostAliases indiquent** les **différents noms associés** au **site Web principal**.
Pour les **sites** à **fort trafic**, il est **déconseillé** de **positionner DNSLookup** à **1**. Cependant, ce **réglage** permet d'**avoir** des **rapports** plus **lisibles** qui **emploient** les **noms complets** des machines **plutôt que** l'**IP**.
On activera **AWStats pour** d'**autres hôtes virtuels**, **en créant** un **fichier spécifique** par hôte.

```
Exemple : /etc/awstats/awstats.www.upf.org.conf
Include ''/etc/awstats/awstats.conf''
SiteDomain=''www.upf.org''
HostAliases=''upf.org''
```

On peut **protéger** les **statistiques** en **donnant** la **liste** des **adresses IP autorisées grâce au paramètre AllowAccessFromWebToFollowingIPAddresses**.
Pour faire **prendre en compte** ce **nouvel hôte**, il faut **modifier** le **fichier /etc/cron.d/awstats** et y **ajouter /usr/lib/cgi-bin/awstats.pl -config=www.upf.org -update**

Rotation de logs

Pour **prendre en compte tous** les **logs**, il est **impératif** d'**invoquer AWStats avant** la **rotation**. On ajoute

dans **/etc/logrotate.d/apache la directive prerotate :**

```
/var/log/apache/*.log {
      weekly
      missingok
      rotate 52
      compress
      delaycompress
      notifempty
      create 644 root adm
      sharedscripts
      prerotate
      su - www-data -c ''/usr/lib/cgi-bin/awstats.pl -config=awstats
-update > /dev/null''
      su - www-data -c ''/usr/lib/cgi-bin/awstats.pl -config=www.upf.org
-update > /dev/null''
      endscript
      postrotate
         /etc/init.d/apache reload > /dev/null
      endscript
}
```

Il faut s'assurer que les **fichiers** de **log** soient **lisibles par tout le monde (create 644 root adm).**
Pour les **icônes** de **awstats (/usr/share/awstats/icon), ajoutez:**

```
Alias /awstats-icon/ /usr/share/awstats/icon/
```

Après quelques minutes, le résultat est accessible en ligne:
http://www.upf.pf/cgi-bin/awstats.pl

Serveur de fichier NFS

NFS (Network File System) est un **protocole** qui **permet** d'**accéder à un système de fichiers à distance, pris en charge par tous** les **systèmes Unix. Pour windows**, il faut **employer Samba**.
NFS est **fort utile** mais il ne faut pas oublier ses **limitations**, surtout en **termes de sécurité** : toutes les **données circulent en clair** sur le réseau (un **sniffer peut** donc **les intercepter**). Le **serveur restreint** l'**accès** en fonction de l'**adresse IP du client (vulnérable au spoofing)**. On peut **sécuriser** ce **protocole avec SSH** :
http://nfs.sourceforge.net/nfs-howto/
RPC (Remote Procedure Call) est un **standard Unix pour** des **services distants**. **NFS** est un **service RPC**. Les services **RPC s'enregistrent dans un annuaire**, le **portmapper** (**port 111 en TCP ou UDP**) qui **indiquent** où **se trouve** le **serveur** (**généralement port 2049**).

Sécuriser NFS (au mieux)

Le **pare-feu** doit **prohiber** le **spoofing** et les **différents ports employés** doivent être **restreints** aux **machines devant accéder aux partages NFS.**
D'autres services RPC sont **nécessaires** au **fonctionnement** optimal de **NFS**, notamment **rpc.mountd, rpc.statd et lockd**. Malheureusement, ils **emploient** par défaut un **port aléatoire assigné par** le **portmapper** et il est donc **difficile** de **filtrer** ce qui leur est destiné.
Les **deux premiers services** sont **démarrés respectivement** par **/etc/init.d/nfs-kernel-server et /etc/init.d/nfs-common**. Pour employer les **options adéquates**, il faut **modifier /etc/default/nfs-kernel-server et /etc/default/nfs-common.**

```
Exemple : /etc/default/nfs-kernel-server
# Number of servers to start up
RPCNFSDCOUNT=8
# Options for rpc.mountd
```

```
RPCMOUNTDOPTS=''-p 2048''

Exemple : /etc/default/nfs-common
# Options for rpc.statd.
# Should rpc.statd listen on a specific port?
# If so, set this variable to a statd argument like: ''--port 1000''.
STATDOPTS=''-p 2046 -o 2047''
# Are you sure that your kernel does or does not need a loackd daemon?
# If so, set this variable to either ''yes'' or ''no''
NEED_LOCKD=
```

Après ces **modifications** et un **redémarrage** des **services**, **rpc.mountd emploie** le **port 2048**; **rpc.statd écoute** le **port 2046 et** utilise le **port 2047 pour** les **connexions sortantes**.
Le service **lockd** est **géré par** un **thread** (processus léger) **noyau**, fonctionnalité **compilée sous forme de module dans les noyaux Debian.** Ce module dispose de **deux options pour** choisir systématiquement **le même port** : **nlm_udpport et nlm_tcpport**. Pour les employer systématiquement, il faut **créer** un **fichier /etc/modutils/lockd puis exécuter update-modules**. **Pour** un **noyau 2.6**, il faut **créer /etc/modprobe.d/.**

```
Exemple : Fichier /etc/modutils/lockd ou /etc/modprobe.d/lockd
options lockd nlm_udpport=2045 nlm_tcpport=2045
```

Avec tous ces **paramétrages**, il est maintenant **possible** de **contrôler** plus **finement** les **accès** au **service NFS grâce à un pare-feu**. Ce sont les **ports 111 et 2045 à 2049 (en UDP et en TCP).**

Serveur NFS

Pour l'**activer automatiquement** à chaque démarrage, il faut **installer nfs-kernel-server** qui **contient** les **scripts d'initialisation** adéquats. Le fichier **/etc/exports donne** les **répertoires exportés** à l'extérieur. A **chaque partage NFS** sont **associées** des **machines** qui ont le **droit d'y accéder**. Un certain nombre d'**options permettent** de dicter **quelques règles d'accès**.

```
Exemple : /etc/exports
/repertoire/a/partager machine1(option1,option2,...) machine2(...) ...
```

Chaque **machine** est **identifiée par** son **nom DNS ou** son **adresse IP**. On peut **spécifier** un **ensemble** de **machine avec *.upf.pf** ou en **décrivant** une **plage complète** d'**adresses IP** (ex: 192.168.0.0/255.255.255.0,192.168.0.0/24,10.2.2.0/24).
Un **partage par défaut** n'est **accessible** qu'**en lecture seule** (option **ro**). L'option **rw donne accès** en **lecture/ecriture**. Les **clients NFS** doivent se **connecter** depuis un **port réservé** à **root (inférieur** à **1024) à moins** que l'**option insecure** n'**ait** été **employée**.
Le **serveur** ne **répond** à une **requête** NFS **que** lorsque **l'opération** sur **disque** a été **complétée** (**option sync**). L'**option async** (asynchrone) **désactive** cette **fonctionnalité** et **améliore** quelque peu les **performances**, au **détriment de** la **fiabilité** (**si** le **serveur crash**, les **données acquittées par** le **serveur NFS** n'auront **pas** été **sauvegardé**).
Pour ne **pas donner** un **accès root à n'importe quel client**, **toutes** les **requêtes provenant** de **root** sont **transformées** en **requêtes provenant** de l'**utilisateur anonymous** (**option root_squash activé** par défaut). Les **options anonuid=uid et anongid=gid permettent** d'**employer** un **autre utilisateur**.

Client NFS

```
Exemple : montage manuel avec la commande mount
mount -t nfs -o rw,nosuid rangi.etudiant.upf.pf:/home /home
Exemple : Entrée NFS dans /etc/fstab
rangi.etudiant.upf.pf:/home /home nfs rsize=8192,wsize=8192,intr,timeo=14
La page de manuel nfs(5) détaille toutes les options
```

Partage Windows avec Samba

Samba est une **suite** d'**outils** qui **permettent** de **gérer** le **protocole SMB** (maintenant **appelé ''CIFS''**) **sous Linux**. Ce **dernier** est **employé par Windows** pour **accéder** aux **partages réseaux** et **aux imprimantes partagées.**
Samba sait également **jouer** le **rôle** de **contrôleur de domaine NT.**

Samba en serveur

Le **paquet** Debian **samba contient** les **deux principaux serveurs** de **Samba 3** (**smbd et nmbd**).
Authentifier à l'aide d'un serveur Windows
Winbind permet **d'utiliser** un **serveur Windows NT comme serveur** d'**authentification et** s'**intègre** à **PAM et** à **NSS**. Il est ainsi **possible** de **mettre en place des machines Linux où tous** les **utilisateurs** d'un **domaine NT disposeront** automatiquement **d'un compte.**
Voir /usr/share/doc/samba-doc/htmldocs/howto/winbind.html

Configuration avec debconf

On peut **reconfigurer samba avec dpkg-reconfigure samba-common samba.**
La **première information** est le **nom** du **groupe de travail** (**UNIV** ou **ETUDIANT**). Il **faut** aussi **utiliser** des **mots de passe chiffrés pour fonctionner avec** les **clients Windows** les plus **récents**.
On **peut utiliser samba comme serveur Wins**. On peut aussi **utiliser inetd pour** lancer les **démons si** le serveur **samba** est **utilisé** de **manière occasionnelle**.
Enfin, **debconf** propose d'**utiliser** le **fichier /var/lib/samba/passdb.tdb** qui est **bien plus efficace qu'un fichier texte standard /etc/samba/smbpasswd.**
Administrer Samba avec SWAT (Samba Web Administration Tool) **est une interface Web permettant** de **configurer Samba**. Pour **activer** l'**interface** de **configuration** par **défaut** : **update-inetd --enable swat. SWAT** est alors **accessible à** l'URL **http://localhost:901**. Il faut **employer** le **compte root et** le **mot de passe habituel.**
Si on veut **interfacer samba avec** un **serveur d'authentification Windows Server 2003**, lire **/usr/share/doc/samba-doc/htmldocs/guide/index.html** qui **traite** d'un **cas concret évoluant** au **fil de la croissance de l'entreprise.**

Configuration manuelle

```
Exemple : /etc/samba/smb.conf
[global]
## Browsing/Identification ##
# Change this to the workgroup/NT-domain name your Samba server will part
of
workgroup = UNIV
# server string is the equivalent of the NT Description field
server string = %h server (Samba %v)
# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable its WINS
Server
wins support = yes //1
# ''security=user'' is always a good idea. This will require a Unix
account
# in this server for every user accessing the server. See /usr/share/doc/
samba-doc/htmldocs/ServerType.html in the samba-doc package for details.
security = user //2
# You may wish to use password encryption. See the section on 'encrypt
passwords' in the smb.con(5) manpage before enabling.
encrypt passwords = true
```

```
# If you are using encrypted passwords, Samba will need to know what
password database type you are using.
passdb backend = tdbsam guest

########## Printing #########
# If you want to automatically load your printer list rather than setting
them up individually then
# you'll need this
load printers = yes //3

# lpr (ng) printing. You may wish to override the location of the
printcap file
; printing = bsd
; printcap name = /etc/printcap

# CUPS printing. See also the cupsaddsmb(8) manpage in the cupsys-client
package.
printing = cups //4
printcap name = cups

######## File sharing ########
# Name mangling options
; preserve case = yes
; short preserve case = yes
unix charset=ISO8859-1 //5
```

1 : indique que **Samba** doit jouer le **rôle** de **serveur** de **nom Netbios (WINS)**
2 : c'est la **valeur par défaut**. Chaque **utilisateur** doit **s'authentifier** avant de pouvoir accéder au moindre partage
3 : demande à samba de **partager automatiquement** toutes les **imprimantes** existantes en local dans **Cupsys**.
4 : documente le **système d'impression** employé, **en l'occurence Cupsys.**
5 : indique le **jeu de caractères employé** (sous Linux) **dans** les **noms** de **fichiers**.
Valeur par défaut : UTF8 (Unicode)

Ajout des utilisateurs

Chaque **utilisateur** ayant **besoin** d'un **compte sur** le **serveur**, il faut **créer** les **comptes unix** (**avec adduser**) **puis enregistrer** chaque **utilisateur dans** la **base de données Samba**.
L'**ajout** d'un **utilisateur dans** la **base de données Samba s'effectue** avec **smbpasswd -a utilisateur** qui **demande** le **mot de passe interactivement.**
On **supprime** un **utilisateur** avec **smbpasswd -x utilisateur**. On peut **geler** quelque temps un **compte avec smbpasswd -d utilisateur**, puis **réactivé** avec **smbpasswd -e utilisateur.**

Transformation en contrôleur de domaines

Permet d'**offrir** des **profils errants** (les **utilisateurs retrouvent** leur **bureau** quelle que soit la machine)
Il faut **rajouter** dans la **section** [**global**] :

```
domain logons = yes //1
preferred master = yes
logon path = \\%L\profiles\%U //2
logon script = scripts/logon.bat //3
```

1 : **active** la fonctionnalité de **contrôleur** de **domaine**
2 : **indique** l'**emplacement** des **répertoires personnels** des **utilisateurs** (**stockés sur** un **partage dédié**

pour l'**option profile acls**)
3 : **script batch exécutés à chaque ouverture** de **session sur** la **machine Windows** cliente. En **l'occurence**, **/var/lib/samba/netlogon/scripts/logon.bat**. Il doit être au **format DOS**, **après modification**, **lancez** la **commande unix2dos** sur le fichier.

```
Exemple : fichier logon.bat
net time \\SERVEUR /set /yes
net use H: /home
net use U: \\SERVEUR\utils
```

2 partages supplémentaires et leurs **répertoires associés** ont aussi été **crées** :

```
[netlogon]
comment = Network Logon Service
path = /var/lib/samba/netlogon
guest ok = yes
writable = no
share modes = no

[profiles]
comment = Profile Share
path = /var/lib/samba/profiles
read only = no
profile acls = Yes
```

Il faut également **créer** les **répertoires personnels** de tous les **utilisateurs** (**/var/lib/samba/profiles/<utilisateur>**) qui doivent en **être propriétaire**.

Samba en client

Paquets Debian **smbfs** et **smbclient**.
Le programme **smbclient interroge** tous les **serveurs SMB**. L'option **-U utilisateur permet** de se **connecter sous** une **autre identité**.
smbclient //serveur/partage accède au **partage** de manière interactive.
smbclient -L serveur donne la **liste** des **partages disponibles**

Monter un partage Windows

smbmount permet de **monter** un **partage** dans l'**arborescence Linux.**

```
Exemple :
smbmount //draco/miepas /media/miepas -o credentials=/usr/local/etc/smb-
credentials
```

Le **fichier /usr/local/etc/smb-credentials** ne sera **pas visible par** les **utilisateurs et respectera** le **format suivant** :

```
username = utilisateur
password = mot_de_passe
```

D'**autres options** existent détaillées **dans** la page de **manuel smbmount(1). Deux options** intéressantes permettent de **forcer l'utilisateur** (uid) **et** le **groupe** (gid) **propriétaire** des **fichiers accessibles sur** le **montage afin** de ne **pas restreindre l'accès à root.**
Le programme **smbumount démonte** un **partage SMB**.

Utiliser mount pour un partage Windows :

```
mount -t smbfs -o credentials=/usr/local/etc/smb-credentials
//serveur/partage /partage
```

Dans /etc/fstab

```
//serveur/partage /partage smbfs credentials=/usr/local/etc/smb-credentials
```

Imprimer sur une imprimante partagée

Cupsys est une **solution élégante pour imprimer sur** une **imprimante partagée par** une **machine Windows depuis** un **poste Linux (smbclient doit être installé)**.
Voici les étapes à suivre :

- **Allez** dans **Cupsys** : **http://localhost:631/admin**
- **Cliquez sur ''ajouter imprimante''** puis **saisissez** les **données**
- Lors du **choix** du **périphérique**, choisissez **''Windows Printer via SAMBA''**
- L'**URI décrivant l'imprimante** doit avoir la **forme suivante** : **smb://utilisateur:motdepasse@serveur/imprimante**

Mandataire HTTP/FTP

Un **proxy** est un **intermédiaire pour** les **connexion HTTP et/ou FTP**. Son **rôle** est **double**:

- Celui de **serveur cache** : il **garde** une **copie** des **documents téléchargés pour éviter de** les **rapatrier plusieurs fois**
- Celui de **serveur filtrant s'il est obligatoire et** que les **connexions sortantes** sont **par ailleurs bloqués**.

Installation

Le **paquet** Debian **squid** n'est qu'un **mandataire modulaire**. **Pour** le **transformer** en **serveur filtrant**, il faut **adjoindre** le **paquet squidguard**. Le **paquet squid-cgi** permet **d'interroger et d'administrer Squid**.
Préalablement à l'installation, il faut **vérifier** que le **système** est **capable** d'**identifier son nom complet** (**hostname -f**). Sinon **modifier /etc/hosts** (**nom_machine.upf.pf**).

Configuration d'un cache

Modifier /etc/squid/squid.conf

```
Exemple : extrait de squid.conf
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
# Example rule allowing access from your local networks. Adapt to list
your (internal) IP networks
# from where browsing should be allowed
acl our_networks src 192.168.0.1/24 192.168.2.0/24
http_access allow our_networks
http_access allow localhost
# And finally deny all other access to this proxy
http_access deny all
```

Configuration d'un filtre

Grâce à **squidguard**, **ligne** à **ajouter dans squid.conf** :
redirect_program /usr/bin/squidGuard -c /etc/squid/squidGuard.conf

Il faut également **installer** le ***CGI* /usr/lib/cgi-bin/squidGuard.cgi** à partir du **fichier squidGuard.cgi.gz dans** le **répertoire /usr/share/doc/squidguard/examples/**. On **modifiera** ce **script** en **changeant** les **variables $proxy (nom du serveur mandataire) et $proxymaster (courrier électronique de contact de l'administrateur)**. Les variables **$image et $redirect** devront **pointer sur** des **images existantes symbolisant** le **refus d'accéder à la page** demandée.
La commande **/etc/init.d/squid reload active** le **filtre**. **Pour définir** le **filtrage**, **personnalisez**

/etc/squid/squidGuard.conf. Après **chaque modification**, **lancez** la commande **update-squidguard**.
Le **format** est **documenté sur** :
http://www.squidguard.org/config
Le paquet **chastity-list offre** un **ensemble** de **règles prêtes à l'emploi pour filtrer** les **sites pornographiques, de piratage**...
Ajouter **dans squid.conf** :

```
redirect_program /usr/bin/squidGuard -c
/etc/chastity/squidGuard/chastity.conf
```

Annuaire LDAP

OpenLDAP implémente le **protocole LDAP**, **base de données** pour **gérer** des **annuaires**. L'emploi de **LDAP** permet de **centraliser** la **gestion** des **comptes** des **utilisateurs et** des **droits associés**. De plus, LDAP est **facile à dupliquer** ce qui **permet** de **mettre en place plusieurs serveurs LDAP synchronisés**. En cas de croissance rapide, on peut donc **répartir** la **charge sur plusieurs serveurs**.
Les **données** sont **structurées** et **hiérarchisées**, les **schémas définissent** les **objets** que la **base peut stocker avec** la **liste** de tous les **attributs possibles**.

Installation

le **paquet slapd contient** le **serveur OpenLDAP**. Le **paquet ldap-utils renferme** les **utilitaires en ligne de commande**.
Plusieurs questions sont posées **avec debconf**. Ne **pas ignorer** la **configuration** de **slapd**, **indiquer** le **nom de domaine (upf.pf)**, le **nom de l'organisation (UPF)**, saisissez le **mot de passe administrateur** pour la base, ne **pas supprimer** la **base en cas de suppression de slapd**, ne **pas autoriser LDAPv2 (si compatible en LDAPv3)**.
On peut **interroger directement** la **base** : **ldapsearch -x -b dc=upf,dc=pf**
Un **fichier LDIF** (LDAP Data Interchange Format) est un **fichier textuel décrivant** le **contenu** d'un **base** LDAP afin de **pouvoir** l'**intégrer dans n'importe quel serveur**.

Remplissage de l'annuaire

Le **paquet migrationtools offre** un **ensemble** de **scripts** qui permettent de **récupérer** les **informations depuis** les **annuaires Unix** standards **(/etc/passwd, /etc/group, /etc/services, /etc/hosts,etc.) puis** de les **intégrer dans** la **base** de données.
Après **installation**, **modifier /usr/share/migrationtools/migrate_common.php pour activer** les **options IGNORE_UID_BELOW, IGNORE_GID_BELOW (qu'il suffit de décommenter)**.
La **mise à jour** à proprement se fait **en exécutant migrate_all_online.sh** comme suit:

```
cd /usr/share/migrationtools
LDAPADD=''/usr/bin/ldapadd -c'' ETC_ALIASES=/dev/null
./migrate_all_online.sh
```

Plusieurs questions sont alors **posées** : **X.500 naming context : dc=upf,dc=pf, LDAP server hostname : localhost, Manager DN : cn=admin,dc=upf,dc=pf, bind credentials : mot de passe administrateur, Create DUAConfigProfile : no**
La **migration** du fichier **/etc/aliases** est **volontairement ignorée**. **S'il** est **nécessaire**, il faut **ajouter /etc/ldap/schema/misc.schema dans** le fichier **/etc/ldap/slapd.conf**.

Utiliser LDAP pour gérer les comptes

Cette section explique **comment paramétrer** un **système Linux** afin que les **différents annuaires emploient** la base de données **LDAP** de manière transparente.

Configuration de NSS

Le système **NSS** (Name Service Switch) est un **système modulaire pour définir** ou **récupérer** les **informations** des **annuaires systèmes**. **Pour** utiliser **LDAP comme** une **source** de données **NSS**, il faut **mettre** en place le **paquet libnss-ldap.**
Au cours de l'installation, **nom du serveur ldap : ldap.upf.pf; nom** de la **base** de recherche : **dc=upf, dc=pf; version** de **LDAP** : **3**; **demande** de **login** : **non**; **fichier** de configuration doit-il être **restreint** en **lecture** : **non**.
Il faut ensuite **modifier /etc/nsswitch.conf pour** lui **indiquer** d'employer le **module ldap.**

```
Exemple : fichier /etc/nsswitch.conf
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionnality.
# If you have the 'glibc-doc' and 'info' packages installed, try:
# 'info libc ''Name Service Switch'' ' for information about this file.
passwd: ldap compat
group: ldap compat
shadow: ldap compat

hosts: files dns ldap
networks: ldap files
protocols: ldap db files
services: ldap db files
ethers: ldap db files
rpc: ldap db files
netgroup: files
```

Le **module ldap**, systématiquement **ajouté** au **début** est donc **consulté** en **premier**. Le **service hosts** fait **exception** puisque pour **contacter** le serveur **LDAP**, il faut **consulter** le **DNS** au **préalable** (pour résoudre ldap.upf.pf).
Si l'on souhaite que le serveur **LDAP** soit la **référence unique** (et ne pas prendre en compte les fichiers locaux employés par le module files) il est possible de **configurer** chaque **service avec** la syntaxe :
service: ldap [NOTFOUND=return] files
Si l'entrée n'existe pas, la réponse n'existera pas même si la ressource existe dans l'un des fichiers locaux.

Configuration de PAM

Le **changement** de la configuration **PAM** standard est une **opération sensible**. En cas de **mauvaise manipulation**, il peut être **impossible** de **s'authentifier**, pensez à **garder un shell root ouvert**.
Installez libpam-ldap
nom du serveur ldap : ldap.upf.pf; nom de la base : dc=upf, dc=pf; version de LDAP : 3; administrateur local root, administrateur de la base : oui; la base demande un login : non;
Root Login Account : cn=admin, dc=upf, dc=pf; Root Login Password : mot de passe admin LDAP; local crypt to use : crypt.

Après installation et configuration du module, il reste à **adapter** la **configuration** PAM par défaut en **modifiant** les **fichiers /etc/pam.d/common-auth, /etc/pam.d/common-password et /etc/pam.d/common-account** :

```
Exemple : fichier /etc/pam.d/common-auth
#
# /etc/pam.d/common-auth - authentication settings common to all services
#
# This file is included from other service-specific PAM config files,
```

```
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.). The default is to use the
traditional Unix authentication # mechanisms.
auth  sufficient pam_ldap.so
auth  required   pam_unix.so try_first_pass nullok_secure

Exemple : fichier /etc/pam.d/common-password
#
# /etc/pam.d/common-password - password-related modules common to all
services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define the services to be
used to change user passwords.
# The default is pam_unix
password sufficient pam_ldap.so

# The ''nullok'' option allows users to change an empty password, else
empty passwords are treated
# as locked accounts.
# (Add 'md5' after the module name to enable MD5 passwords)
# The ''obscure'' option replaces the old 'OBSCURE_CHECKS_ENAB' option in
login.defs. Also
# the ''min'' and ''max'' options enforce the length of the new password
password   required   pam_unix.so nullok obscure min=4 max=8 md5
# Alternate strength checking for password. Note that this requires the
libpam-cracklib package to
# be installed. You will need to comment out the password line above and
uncomment the next two
# in order to use this.
# (Replaces the 'OBSCURE_CHECKS_ENAB', 'CRACKLIB_DICTPATH')
# password required  pam_cracklib.so retry=3 minlen=6 difok=3
# password required  pam_unix.so use_authtok nullok md5

Exemple : fichier /etc/pam.d/common-account
# /etc/pam.d/common-account - authorization settings common to all
services
# This file is included from other service-specific PAM config files,
# and should contain a list of the authorization modules that define the
central access policy for use # on the system. The default is to only
deny service to users whose accounts are expired in
# /etc/shadow
account sufficient   pam_ldap.so
account requiredpam_unix.so try_first_pass
```

Services mal configurés

Les **différents fichiers /etc/pam.d/common-*** sont **prévus** pour être **employés** de **manière standard par tous** les **services** (**grâce à** une **directive @include) mais** ça **n'est pas** le **cas** de **tous**. **Sudo** par exemple est une **exception à la règle** et continuera d'**utiliser pam_unix.so**. Dans ce cas, les **services** deviennent **non fonctionnels** puisque la **base shadow renvoyée par** le **module libnss-ldap** ne **mentionne aucun mot de passe chiffré** (la connexion à **LDAP étant anonyme)**.
Il faut donc **reconfigurer** ces **services** manuellement (en **modifiant** le **fichier /etc/pam.d/<service>**) pour **employer** également le **module pam_ldap.so**.
L'autre solution est de **positionner** la **variable rootbinddn dans /etc/libnss-ldap.conf** ce qui **permet** au

moins **aux processus diposant** de **droits roots** de **récupérer** les **mots de passe chiffrés** via NSS.

Sécuriser les échanges de données LDAP

LDAP est **transporté en clair** sur le réseau. Il convient d'**employer** une **couche supplémentaire** de chiffrement.

Configuration côté serveur

La 1er étape consiste à **créer** une **paire** de **clés publique et privée** pour LDAP. Il faut **installer** le **paquet openssl puis** exécuter la **commande /usr/lib/ssl/misc/CA.pl -newcert** qui **pose plusieurs questions** banales (**lieu, nom de l'organisation**...). Il est **impératif** de **répondre à** la question ''**Common Name**'' le **nom complet** du **serveur LDAP; ldap.upf.pf**.
La **commande génère** un **fichier newreq.pem**. Il faut **séparer** la **clé publique de la clé privée avec openssl rsa -in newreq.pem -out newkey.pem puis** en **supprimant dans** le fichier **newreq.pem** le **bloc RSA PRIVATE KEY**.
Il reste à **installer** ces **clés dans** un **emplacement standard** :

```
mv newkey.pem /etc/ssl/private/ldap-key.pem
chmod 0600 /etc/ssl/private/ldap-key.pem
mv newreq.pem /etc/ssl/certs/ldap-cert.pem
```

Indiquons à **slapd** qu'il doit **employer** ces **clés** dans le **cadre** du **chiffrement**. Il faut **ajouter** les **directives** ci-dessous **dans /etc/ldap/slapd.conf**.

```
Exemple : configuration de slapd pour la prise en charge du chiffrement
# TLS support
TLSCipherSuite HIGH
TLSCertificateFile /etc/ssl/certs/ldap-cert.pem
TLSCertificateKeyFiles /etc/ssl/private/ldap-key.pem
```

La **dernière étape** pour **activer** la **mise en place** du **chiffrement** est de **modifier** la **variable SLAPD_SERVICES** du fichier **/etc/default/slapd**.

```
Exemple : fichier /etc/default/slapd
# Default location of the slapd.conf file
SLAPD_CONF=
# System account to run the slapd server under. If empty the server will
run as root.
SLAPD_USER=
# System group to run the slapd server under. If empty the server will
run in the primary group of
# its user.
SLAPD_GROUP=
# Path to the pid file of the slapd server. If not set the init;d script
will try to figure it out from
# $SLAPD_CONF (/etc/ldap/slapd.conf)
SLAPD_PIDFILE=
# Configure if the slurpd daemon should be started. Possible values :
# - yes: Always start slurpd
# - no: Never start slurpd
# - auto: Start slurpd if a replica option is found in slapd.conf
# (default)
SLURPD_START=auto
# slapd normally serves ldap only on all TCP-ports 389. slapd can also
service requests on TCP-port
# 636 (ldaps) and requests via unix sockets.
```

```
# Example usage:
SLAPD_SERVICES=''ldaps:/// ldapi:///''
# Additional options to pass to slapd and slurpd
SLAPD_OPTIONS='' ''
SLURPD_OPTIONS='' ''
```

Configuration côté client

Il faut **modifier** la **configuration** des **modules libpam-ldap et libnss-ldap en ajoutant ssl** on **à /etc/pam_ldap.conf et /etc/libnss-ldap.conf**
Les **clients doivent** également **authentifier** le **serveur** en **connaissant** la **clé publique**. Il est **nécessaire** d'en **installer** une **copie** (**/etc/ssl/certs/ldap-cert.pem**) **et** de la **référencer** depuis **/etc/ldap/ldap.conf**.

```
Exemple : /etc/ldap/ldap.conf
BASE dc=upf,dc=pf
URI ldaps://ldap.upf.pf
#SIZELIMIT 12
#TIMELIMIT 15
#DEREF never
TLS_CACERT /etc/ssl/certs/ldap-cert.pem
```

Station de travail

Configuration de XFree86

Avec **debconf associé** au paquet **xserver-xfree86.** Pour **déterminer** les **valeurs** : **discover, mdetect et read-edid**. Pour le **pilote video** : **discover --data-path=xfree86/server/device/driver display.** On peut aussi **détecter automatiquement** la **souris avec** : **mdetect -x.** Enfin les **caractéristique** de **l'écran** sont **données avec** le **protocole DDC** : **get-edid | parse-edid.**

Script de configurations

Pour **reconfigurer l'interface graphique** : **dpkg-reconfigure xserver-xfree86**. Choix du serveur X : **xserver-xfree86, pour** le **pilote vidéo reprendre** celui de la **commande discover (ati pour ATI, mga pour Matrox, nv pour nVidia...).**
A **défaut** d'un **pilote adéquat**, on peut **employer vesa** qui **convient pour** la **quasi totalité des cartes.**
Clavier : **pc105** (si touche windows) et **disposition** des **touches** (keyboard layout) : **fr. Souris** : souvent **/dev/psaux** (PS/2), **pour** un noyau **2.6** on peut **répondre /dev/input/mice et ImPS/2.**
Ecran : le **mode simple demande** la **taille** réelle de **l'écran** (pour CRT), le **mode moyen** permet d'**indiquer** une **liste** de **résolutions et** de **fréquence**. Le **mode expert** permet de **spécifier** des **intervalles horizontaux** et **verticaux différents** (notice de l'écran).

Personnalisation de l'interface

Gestionnaire d'écran répandus : **gdm, kdm et xdm**. Pour **gdm dans /etc/gdm/gdm.conf**, on peut **permettre** à l'**utilisateur** d'**éteindre et de redémarrer** la machine **sans** le **mot de passe admin** :

```
Greeter=/usr/bin/gdmgreeter
SystemMenu=true
SecureSystemMenu=false
Welcome=Bienvenue sur %s
Use24Clcok=true
UseCirclesInEntry=true
GraphicalTheme=happygnome
```

Gestionnaire de fenêtres

GNOME emploie metacity et **KDE exploite kwm**. **Xfce** dispose de **XFwm**. **Pour** des **anciens postes**, des **gestionnaires moins gourmands** existent : **WindowMaker (wmaker), afterstep, fvwm2, icewm ou encore blackbox.** Dans ce cas, il **peut être intéressant** de **modifier** le **choix x-window-manager grâce à update-alternatives --config x-window-manager.**

Gestion des menus

Pour avoir des **menus à jour**, Il existe une **base centrale où** chaque **nouvelle application s'enregistre**. **Chaque** nouveau **paquet** installé **s'ajoute dans cette base**. Cet **infrastructure** est **offerte par** le **paquet menu**. Chaque **paquet dépose** un **fichier dans /usr/lib/menu** qui **décrit** les **capacités** de l'**application et** l'**emplacement qui** lui **convient** le mieux. Le script de **post-installation appelle update-menus.**
L'**administrateur** peut **intervenir dans** le **processus** pour **influencer** les **menus générés**. Pour **supprimer** un **élément** du **menu**, il faut **placer dans /etc/menu** un **fichier vide portant** le **nom du paquet**. On peut **réorganiser** le **menu** en **changeant** le **nom** de **certaines sections ou** les **regrouper** : **/etc/menu-methods/translate_menus**. Pour **ajouter** des **éléments**, **/etc/menu/local.<element>.**

GNOME

```
apt-get install gnome-desktop-environment
```

Pour les administrateurs, **GNOME semble** être **mieux préparé** à des **déploiements massifs**. La **configuration** des **applications** est **gérée par GConf**, une **sorte** de **base** de **registre interrogeable** et **modifiable par** l'utilitaire **gconftool-2.**
Ressource en ligne : **http://www.gnome.org/learn/admin-guide/latest**

KDE

apt-get install kde
Qt n'étant **pas libre** (bientôt) **sous Windows aucune application libre KDE** ne **peut être porté sous** ce **système**. Le **langage C++** est **obligatoire pour développer** une **application KDE.**

XFCE

Bureau graphique **simple et allégé**. **apt-get install xfce4**
Contrairement aux précédents, il ne **comprend** que:
un gestionnaire de fichiers
un gestionnaire de fenêtres
un panneau démarreur d'applications
Xfce en **entreprise** :
http://www.xfce.org/index.php?page=documentation&lang=fr#corporate

Outils

Les **paquets populaires**, des sondages sont réalisés (grâce notamment au paquet popularity-contest), elles **sont sur http://popcon.debian.org**, si vous **hésitez entre deux paquets**, **choisissez le plus populaire.**

Courrier électronique

Evolution

Il **gère** un **agenda**, un **carnet d'adresses**, une **liste de tâches et un puissant système d'indexation des messages**. Une **extension permet** de l'**intégrer à Microsoft Exchange** : il s'agit de **Ximian Connector du paquet debian evolution-exchange**.

KMail

fait parti de la **suite KDE-PIM comprenant** le **carnet d'adresses, l'agenda**... Un **excellent client** de messagerie.

Thunderbird

De la **suite** logicielle **Mozilla**. **Paquets** debian **mozilla-thunderbird, mozilla-thunderbird-locale-fr et mozilla-thunderbird-enigmail (pour le chiffrement)**.

Navigateurs Web

Epihany sous **GNOME** (moteur gecko). **Paquet epiphany-browser**.
Konqueror (moteur KHTML). **Choisi par Apple pour** son navigateur **Safari**.
Mozilla-firefox allie la **puissance de gecko** a une **interface légère et extensible**.

Développement

Outils pour GTK+ sur Gnome

Ajunta est un **environnement** de **développement GTK+**. **Glade** (du paquet éponyme) est **capable** de **créer** des **interfaces utilisateurs et** de les **enregistrer dans** un **fichier** (au **format XML**). La bibliothèque **libglade** permet de **recréer dynamiquement** les **interfaces sauvegardés**.

Outils pour QT sur KDE

KDevelop (kdevelop3). **Qtdesigner** est un **logiciel facilitant** la **conception** d'**interface graphique**.

Travail collaboratif

Groupware

PHPGroupware regroupe de **nombreuses fonctionnalités** de **travail collaboratif**:
messagerie électronique, messagerie instantanée, carnet d'adresses, wiki, système de gestion de contenus, base de connaissance, annuaire de liens, système de chat, gestion de projet
http://www.phpgroupware.org
alternative : egroupware, apt-cache search egroupware

Vidéoconférence avec gnomemeeting

Application **phare sous Linux**. Si l'on souhaite **placer un** seul **client GnomeMeeting derrière** un **pare-feu**, il faut **forwarder** le **port 1720 en tcp** (écoute des connexions entrantes), les **ports 30000-30010 en TCP** (contrôle des connexions ouvertes) et les **ports 5000-5013 en UDP** (audio, vidéo et enregistrement dans un proxy H323). **Si** l'on souhaite placer **plusieurs clients**, il faut **installer** un

''proxy H323'' (paquet gatekeeper).

Messagerie instantanée

Jabber utilise un **protocole standardisé** et **ne démérite pas en** terme de **fonctionnalités**.

Configuration du serveur

apt-get install jabber jabber-muc jabber-jud. MUC (Multi User Chat), JUD (Jabber User Directory).
Modifier /etc/jabber/jabber.xml. Toutes les **références** à **localhost** doivent être **remplacés par** le **nom officiel du serveur** (jabber.upf.pf). On peut **personnaliser** les **informations** de **vCard** et **modifier** les **messages textuels standards**. On corrige le **nom** de **machine** (variable JABBER_HOSTNAME) **dans /etc/jabber/jabber.cfg**.
Il faut ensuite **ajouter** un **bloc service** au **fichier jabber.xml** et **activer** le **module** en **modifiant /etc/default/jabber-muc et /etc/default/jabber-jud** et **positionner** la **variable ENABLED à 1** et **personnaliser /etc/jabber/jabber-muc.xml (et jabber-jud.xml)** de manière concordante (les **mots de passe indiqués dans <secret> et** les **identifiants de service** doivent **correspondre** au **contenu** du **fichier jabber.xml).**
A savoir, les **jid** sont **uniques** (**même si** le **port diffère,** chaque **service** doit **recevoir** un **jid différent** qui **soit** un **nom DNS valide**), c'est pourquoi **conference.jabber.upf.pf, jud.jabber.upf.pf et jabber.upf.pf pointent tous trois sur la même machine.**
`Lire /usr/share/doc/jabber-muc/README.Debian et /usr/share/doc/jabber-jud/README.Debian.`

Clients Jabber

Gnome dispose de **gossip** et **Kde de Kopete.**

Travail collaboratif avec GForge

GForge est en réalité une **agglomération d'outils** permettant de **gérer**, **suivre** et **animer** des **projets** classés en **trois catégories :**
communication : forums de discussion sur le Web, gestionnaire de listes de diffusion par messagerie électronique, systèmes de nouvelles permettant à un projet de publier des brèves
suivi : gestionnaire de tâches permettant le contrôle de leur progrès et leur ordonnancement, pisteurs pour les bogues
partage : gestionnaire de documentation, mise à disposition de fichiers générique, espace web dédié à chaque projet
A cela **s'ajoute CVS** (Concurrent Versions System) qui **contient** un **historique** des **différentes versions**, une **trace de chaque changements** et **fusionne** les **modifications apportées indépendamment**.
La **plupart** de ces **outils** sont **accessibles** (voire gérés) **par** une **interface Web avec** un **système** de **gestion** des **permissions et** des **notifications par courrier électronique.**

Suites bureautiques

OpenOffice.org est devenu une **référence** (avec une **meilleure compatibilité avec** les **formats word et excel que gnome-office ou koffice**). Les **paquets** à installer : **openoffice.org, openoffice.org-l10n-fr, openoffice.org-help-fr et openoffice.org-spellcheck-fr-fr.**

L'émulation Windows : Wine, VMWare, VNC, QEMU...

Le plus simple **pour utiliser Wine** est de **disposer** d'une **partition comportant** déjà une **installation** de

Microsoft **Windows sous** un **système** de fichier **FAT** (le seul accessible sous Linux).
Il faut **s'assurer** que la **partition** est **montée (sous /windows)** et **accessible** en **lecture/ecriture à l'utilisateur qui emploie wine**.
Dans fstab:

```
/dev/hda1 /windows fat defaults,uid=1000,gid=100,umask=002,nls=iso8859-1
0 0
```

Installons tous les paquets nécessaires :

```
apt-get install wine winesetuptk msttcorefonts libwine-print wine-utils
wine-doc
```

Il faut ensuite **exécuter winesetup** et **accepter** les **réglages par défaut**. On pourra ensuite **exécuter** les **programmes Windows** en **exécutant wine /windows/.../program.exe**.
Pour **l'émulation machine**, les **paquets QEMU** ou **Bochs** sont des **solutions libres**.

Windows Terminal Server ou VNC

On peut **installer** les **applications Windows** à conserver **sur** un **serveur central Windows Terminal Server** et **exécutées à distance avec** le paquet **rdesktop**. Ce programme **gère** le **protocole RDP** (Remote Desktop Protocol) **employé par Windows NT/2000**.

Recompiler un paquet depuis ses sources

On peut avoir besoin de **fonctionnalité** du **logiciel** qui **implique** de **recompiler** le **programme ou** de prendre une **version plus récente** (testing ou unstable) à **faire fonctionner sous stable (rétroportage)**.

Récupérer les sources

```
apt-get update
apt-get source nom-paquet-source (nécessite une ligne deb-src dans
/etc/apt/sources.list)
```

Si vous souhaitez **récupérer** une **version particulière** (non disponible avec apt), récupérer **2 ou 3 fichiers (*.dsc, *.tar.gz, *.diff.gz, ce dernier existe que si .tar.gz contient .orig.tar.gz)** puis **exécuter** :

```
dpkg-source -x fichier.dsc
```

Effectuer les modifications

les **sources** du paquet sont maintenant **disponibles dans** un **répertoire portant** le **nom du paquet source et** sa **version** (**ex: samba-3.0.2**). Il faut **changer** le **numéro de version pour distinguer** les **paquets recompilés** des paquets originaux. Supposons que la **version** soit **3.0.2-2**, nous pouvons **créer** la **version 3.0.2-2.upf1** pour **indiquer** l'**origine du paquet**.
Pour **effectuer** ce **changement**, **utiliser dch** du **paquet devscripts** en **saisissant dch -v 3.0.2-2.upf1** qui **lancera** un **éditeur** permettant de **changer** le **fichier debian/changelog**.
Si **modification** des **options de compilation**, il faut **modifier** le **fichier debian/rules.** Il **contient** les **lignes concernant** la **configuration initiale** (./configure ...) **ou déclenchant** la **compilation** ($(MAKE) ... ou make ...).
Il convient **parfois** de **s'occuper** du **fichier debian/control** qui **renferme** la **description** des **paquets générés**. Il peut être **intéressant de** la **changer** pour qu'elle **reflète** les **changements apportés**. Ce fichier **contient aussi** des champs **Build-Depends** qui **donnent** la **liste** des **dépendances** de **génération**

de paquets (voir apt-get.org).
apt-get permet d'**installer tous** les **paquets cités dans** ce champ, **apt-get build-dep paquet-source**

Démarrer la recompilation

Ce processus **nécessite** théoriquement les **droits root**, par **sécurité**, l'**utilitaire fakeroot permet** de **s'en passer**. **Tout** ce **processus** est **contrôlé par dpkg-buildpackage.**

```
Exemple : recompilation d'un paquet
dpkg-buildpackage -r fakeroot -us -uc
```

Cette **commande peut échouer si** le **champ Build-Depend**s n'a **pas** été **corrigé** ou les **dépendances** n'ont **pas** été **installées**. On **peut outrepasser** cette **vérification avec** l'**option -d**.
Le **programme debuild** (du **paquet devscripts**) **fera suivre** l'**appel de dpkg-buildpackage par** l'**exécution** d'un **programme chargé** de **vérifier** le **paquet généré**.
Le programme **pbuilder** permet de **recompiler** un **paquet dans** un **environnement chrooté** : il **crée** un **répertoire temporaire contenant** un **système minimal nécessaire** à la **reconstruction** du **paquet** (en se **basant sur Build-Depends**). **Grâce à chroot**, ce **répertoire sert** ensuite **de racine** (/) **lors** du **processus** de **recompilation**. Cette **technique permet** de **détecter rapidement** les **manques éventuels dans** les **dépendances et** de **compiler** un **paquet pour** une **version Debian différente du système** (compilation unstable sur une Debian stable).

Construire son premier paquet

Faux paquet ou méta-paquet

Paquet vide permettant de **satisfaire les dépendances** lorsque le logiciel en question a été installé manuellement. C'est en fait une collection de paquets par le biais de ses dépendances que son installation installera toutes. On peut recourir aux programmes **equivs-control** et **equivs-build** (du paquet **equivs**). La commande **equivs-control fichier crée** un **fichier contenant** des **en-têtes** de paquet **Debian** qu'on modifiera pour indiquer le nom du paquet souhaité, son numéro de version, le nom du mainteneur, ses dépendances, sa description. Tous les autres champs dépourvus de valeur par défaut sont optionnels et peuvent être supprimés. Les champs **Copyright**, **Changelog**, **Readme** et **Extra-files** sont **propres** à **equivs-build** et **disparaîtront des en-têtes réels** du paquet généré.

```
Exemple : Fichier d'en-têtes d'un faut paquet libxml-libxml-perl
Section:  perl
Priority: optional
Standards-Version: 3.5.10

Package: libxml-libxml-perl
Version: 1.57-1
Maintainer: Prenom Nom <prenom.nom@upf.pf>
Depends: libxml2 (>= 2.6.6)
Architecture: all
Description: Fake package - module manually installed in site_perl
This is a fake package to let the packaging system believe that this
Debian package is installed.
.
In fact, the package is not installed since a newer version of the module
has been manually compiled & installed in the site_perl directory
```

Ensuite, il suffit de **générer** le **paquet Debian** en invoquant **equivs-build** fichier.

Simple archive de fichiers

Si l'on veut **déployer** facilement un **ensemble de documents sur** un **grand nombre de machines**, on peut **créer un paquet**.
http://www.debian.org/doc/maint-guide/index.fr.html

On **crée** un **répertoire upf-data-1.0** qui **abritera** le **paquet source** ayant pour nom upf-data et pour numéro de version 1.0. On **place** ensuite les **fichiers dans** un sous **répertoire data**. On **invoque** la **commande dh_make pour ajouter** les **fichiers requis** par le processus de **génération** d'un **paquet** (tous contenus **dans** un **sous-répertoire debian**) :

```
cd upf-data-1.0
dh_make
Type of package: single binary, multiple binary, library or kernel
module?
[s/m/l/k] s
Maintainer name: Prenom Nom
Email-Adress: prenom.nom@upf.pf
Date: Thr, 11 May 2006 11:15:12 +0100
Package Name: upf-data
Version: 1.0
Type of Package: single
```

Le **type** de paquet **single binary indique** que **ce paquet source** ne **générera** qu'**un seul paquet binaire**.
Mutliple-binary est à employer pour un **paquet source générant plusieurs paquets binaires**.
Library est un **cas particulier pour** les **bibliothèques partagées** qui doivent suivre des règles de mise en paquet très strictes. Il en est de **même pour kernel module** réservé aux **paquets contenant** des **modules** de **noyau**.
Le **sous répertoire Debian** généré **contient** de **nombreux fichiers**. Les **fichiers nécessaires** sont **rules**, **control**, **changelog** et **copyright**. Les **fichiers .ex** sont des **exemples** qu'on peut **modifier** et **rebaptiser ou supprimer**. Le **fichier compat** est **nécessaire au** bon **fonctionnement** des **programmes** de l'ensemble **appelé debhelper dont** les **noms commencent par** le **préfixe dh_** .
Copyright doit **contenir** les **auteurs** des **documents inclus dans** le **paquet et** la **licence logicielle associée**. Le fichier **changelog par défaut convient** relativement bien. Le fichier **control** a changé : la **section** a désormais **pour valeur misc** et le **champ Architecture** est **passée de any à all** puisque le paquet abrite des documents. Le champ **Depends** a été **changé en mozilla-browser | www-browser** pour **garantir** la **présence** d'un **navigateur Web** capable de consulter les documents ainsi diffusés.

```
Exemple : le fichier control
Source: upf-data
Section: misc
Priority: optional
Maintainer: Prenom Nom <prenom.nom@upf.pf>
Build-Depends: debhelper (>= 4.0.0)
Standards-Version: 3.6.0

Package: upf-data
Architecture: all
Depends: mozilla-browser | www-browser
Description: Documentation interne de UPF
Ce paquet fournit plusieurs documents décrivant la structure interne de
l'UPF. Cela comprend:
-  l'organigramme
-  les contacts pour chaque département
.
Ces documents NE DOIVENT PAS sortir de la société. Ils sont réservés à un
```

```
usage interne

Exemple : le fichier changelog
upf-data (1.0-1) internal; urgency=low
•  Inital Release
•  Commençons avec peu de documents:
   - la structure interne de la société
   - les contacts de chaque département

Exemple : le fichier copyright
This package was debianized by Prenom Nom <prenom.nom@upf.pf> on Thr, 11
May 2006 11:15:12 +0100.
Upstream Author(s): UPF
Copyright:
Copyright 2006 UPF - all rights reserved
```

Le fichier **rules contient** un ensemble de **règles employées** pour **configurer**, **compiler** et **installer** le **logiciel dans** un **sous répertoire dédiés**. Le **contenu** de ce **sous-répertoire est** ensuite **intégré** au **paquet Debian** comme si il était à la racine du système de fichiers. Dans notre cas, les **fichiers** seront **installés dans** le répertoire **/usr/share/upf-data** . Le fichier **rules** est de **type makefile avec** quelques **cibles standardisées** (notamment **clean** et **binary** pour **nettoyer** et **produire** le **binaire**). Seul **build**, **install** et **clean** ont été **modifiées**. Les **références** à **$(MAKE)** ont été **supprimées**. La cible **install** a **reçu** les **commandes** permettant de **créer /usr/share/upf-data et** d'y **copier** la **documentation** :

```
build:build-stamp
build-stamp: configure-stamp
      dh_testdir
      # Add here commands to compile the package.
      touch build-stamp
clean:
      dh_testdir
      dh_testroot
      rm -f build-stamp configure-stamp
      # Add here commands to clean up after the build process.
      dh_clean
install: build
      dh_testdir
      dh_testroot
      dh_clean -k
      dh_installdirs
      # Add here commands to install the package into debian/upf-data
      mkdir -p $(CURDIR)/debian/upf-data/usr/share/upf-data
      cp -a data/* $(CURDIR)/debian/upf-data/usr/share/upf-data
```

Un fichier **makefile** est un **script détaillant** au **programme make** les **règles nécessaires pour reconstruire** des **fichiers issus** d'un **réseau** de **dépendances**. Il **contient** la **liste** de ces **règles** en **respectant** le **format suivant**:

```
cible: sources
      commandes
```

Si l'un des **fichiers** de **sources** est **plus récent que** le **fichier cible**, il faut **exécuter** les **commandes pour régénérer** la **cible** à partir des sources.
Un **caractère** de **tabulation** doit **impérativement précéder toutes** les **commandes**, **si** la **ligne** de commande **commence par** un **moins** (-), la **commande peut échouer sans que tout** le **processus avorte**. **A ce stade**, il est déjà **possible** de **créer** le **paquet**. Nous allons toutefois y ajouter une dernière touche.

Pour que ces **documents** soient facilement **accessibles depuis** les **menus Aide (ou Help)** des **bureaux graphiques**, il faut **créer** une **entrée dans** le système de **menus Debian**. Pour cela, on **modifie** le **fichier debian/menu.ex** et on l'enregistre sans extension.

```
Exemple : le fichier menu
?package(upf-data) :needs=X11|wm section=Help\
title=''Documentation interne à UPF'' \
command=''/usr/bin/x-www-browser /usr/share/upf-data/index.html''
?package(upf-data) :needs=text section=Help\
title=''Documentation interne à UPF''\
command=''/usr/bin/www-browser /usr/share/upf-data/index.html''
```

Le champ **needs positionné** à **X11|wm** indique que cette **entrée** de **menu** n'a de sens **que dans** l'**interface graphique**. Le champ **section précise** l'**emplacement** de l'**entrée dans** le **menu**. **Dans notre cas**, elle sera intégrée dans le **sous menu d'aide Help**. Le champ **title** est le **texte que** les **utilisateurs verront** dans le menu. Enfin, **command décrit** la **commande à exécuter**. La **deuxième entrée** est le **pendant** de la **première** mais **adaptée au mode texte** d'une console Linux.
Après rédaction du fichier menu, il s'agit de l'**installer au bon endroit**. Nous **déléguerons** cette **tâche** au **programme dh_installmenu** en **décommentant** la **ligne correspondante dans** la **cible binary-arch** du **fichier rules**.
Le paquet source étant prêt, on **génère** le **paquet binaire** en se **plaçant dans** le **répertoire upf-data-1.0** et en **exécutant dpkg-buildpackage -r fakeroot -us -uc**
L'**organisation** des **menus Debian suit** une **structure précise**, documentée dans le texte suivant:
http://www.debian.org/doc/packaging-manuals/menu-policy/

Créer une archive de paquets pour APT

Pour **faciliter** le **déploiement** de **paquets modifiés ou** de **programmes**, on peut **intégrer** une **archive** de paquets **directement utilisable par APT**. On **sépare** les **paquets internes des paquets officiels** :

```
deb http://packages.upf.pf/ updates/
deb http://packages.upf.pf/ internal/
```

Il suffit de **configurer** un **hôte virtuel sur** le **serveur HTTP** interne. La **racine** de l'**espace Web associé est /srv/vhosts/packages/**. Pour **gérer** les **archives**, on peut **employer** le **programme mini-dinstall**. Celui-ci **scrute** un **répertoire** d'**arrivée incoming** (en l'occurence, **/srv/vhosts/packages/mini-dinstall/incoming**) **pour** y **récupérer tout paquet Debian déposé** et l'**installer dans** une **archive Debian** (dont le répertoire est **/srv/vhosts/packages**). Ce **programme fonctionne en traitant** les **fichiers .changes crées lors de** la **génération** d'un **paquet Debian**.