

4. ADMINISTRATION LINUX

4.1. RÔLE DE L'ADMINISTRATEUR SYSTÈME

Unix étant un système d'exploitation multi-utilisateurs, la gestion du système et des utilisateurs est confiée à un super-utilisateur nommé root ou racine.

Le rôle de l'administrateur ou root est de :

- configurer le noyau du système d'exploitation ;
- sauvegarder les données et réparer les systèmes de fichier ;
- gérer les utilisateurs ;
- installer de nouveaux logiciels ;
- intégrer de nouveaux disques durs et de nouvelles partitions ;
- configurer le processus de démarrage de Linux ;
- configurer le réseau

Du fait que le super-utilisateur root possède tous les droits, il doit posséder des connaissances concernant le fonctionnement du système.

4.2. PRINCIPAUX RÉPERTOIRES SYSTÈMES

Répertoires standards :

/	Répertoire racine (ou root) contenant tous les répertoires.
/home	Répertoire contenant les répertoires personnels de tous les utilisateurs autres que root.
/root	Répertoire personnel de l'administrateur système root.

Répertoires système :

/bin	Répertoire contenant les commandes et utilitaires employés par tous les utilisateurs (ls, rm, cp, etc..)
/boot	Répertoire contenant des informations permettant le chargement de Linux.
/dev	Répertoire contenant tous les fichiers périphériques permettant d'accéder aux composants matériels.
/etc	Répertoire contenant les commandes et fichiers de paramétrages nécessaires à l'administration système.
/lib	Répertoire contenant les bibliothèques communes à tous les utilisateurs
/proc	Répertoire spécial utilisé par le système et contenant la liste des processus en cours d'exécution.
/sbin	

	Répertoire contenant les commandes et utilitaires utilisées seulement par l'administrateur système.
<code>/tmp</code>	Répertoire contenant les fichiers temporaires.
<code>/usr</code>	Répertoire composé d'un certain nombre de sous répertoires utilisés par l'ensemble des utilisateurs.
<code>/var</code>	Répertoire spécial utilisé par le système pour stocker des données souvent modifiées.

4.3. GESTION DES UTILISATEURS

Même si vous êtes le seul utilisateur, il est indispensable de créer des utilisateurs ne serait-ce que pour des raisons de sécurité. L'utilisateur root ayant tous les droits, il est recommandé de se connecter avec un autre utilisateur afin d'éviter d'effectuer de fausses manip qui pourraient avoir de lourdes conséquences sur la stabilité du système. Lorsque vous avez besoin de faire de l'administration système, vous avez toujours la possibilité de changer d'utilisateur et de devenir root à partir de la commande `su` :

```
$ su root
$ whoami
root
$ exit
$ whoami
pdrouot
```

Chaque utilisateur dispose d'un répertoire personnel sous `/home`, par exemple `/home/philippe` pour l'utilisateur philippe. Outre les fichiers personnels de l'utilisateur, son compte comprend des fichiers cachés de configuration du shell ainsi que les préférences de l'interface graphique X-Window.

4.3.1. Principe de l'ajout des utilisateurs

L'ajout d'un utilisateur consiste à :

- associer un mot de passe à l'utilisateur (ajout d'une entrée dans le fichier `/etc/passwd`) ;
- définir à quel groupe appartient l'utilisateur (ajout d'une entrée dans le fichier `/etc/group`) ;
- créer le répertoire personnel de l'utilisateur ;
- créer le fichier de configuration personnel du shell ;

Une entrée (c'est à dire une ligne) du fichier `/etc/passwd` est de la forme :

Nom : mot de passe : numéro d'utilisateur : numéro de groupe : champs spécial : répertoire personnel : shell de démarrage

Exemple d'entrée du fichier `/etc/passwd` :

```
$ cat /etc/passwd | grep philippe
philippe:x:501:100:Philippe Drouot:/home/philippe:/bin/bash
```

Une entrée (c'est à dire une ligne) du fichier `/etc/group` est de la forme :

Nom de groupe : champs spécial : numéro de groupe : membre1 , membre2, etc..

Exemple d'entrée du fichier `/etc/group` :

```
$ cat /etc/group | grep 100
users:x:100:
```

Pour ajouter un utilisateur philippe, vous devez :

1. ajouter l'utilisateur philippe dans le fichier `/etc/passwd`
2. ajouter éventuellement un nouveau groupe dans `/etc/group` (si vous souhaitez créer un groupe spécifique pour philippe)
3. créer le répertoire personnel du nouvel utilisateur (home directory), copier les fichiers de configuration du shell et changer les droits du répertoire philippe afin que l'utilisateur philippe devienne propriétaire de son répertoire personnel :

```
$ mkdir /home/philippe
$ cp /etc/skel/* /home/philippe
$ chown philippe /home/philippe
$ chgrp le_groupe_de_philippe /home/philippe
```

4. donner un mot de passe à l'utilisateur philippe par la commande :

```
$ passwd philippe
```

4.3.2. Ajout d'utilisateur et de groupe avec les commandes `useradd` et `groupadd`

Pour faciliter l'ajout d'utilisateurs et de groupe, il existe des commandes spécifiques.

Ajout d'un utilisateur avec la commande `useradd` :

Syntaxe : `useradd nom-utilisateur -g groupe -d repertoire-personnel -m`

L'option `-m` permet de recopier les fichiers de configuration du shell. On peut remplacer le shell courant par un shell spécifique avec l'option `-s` (par exemple `-s /etc/ftponly`).

```
$ useradd philippe -g users -d /home/philippe -m
$ passwd philippe
Changing password for user philippe
New UNIX password :
Retype new UNIX password :
Passwd : all authentication tokens update successfully
```

Suppression d'un utilisateur avec la commande `userdel` :

```
$ userdel -r philippe
```

L'option `-r` permet de supprimer le répertoire personnel de l'utilisateur à supprimer.

Ajout d'un groupe avec la commande `groupadd` :

```
$ groupadd ftpusers
```

Suppression d'un groupe avec la commande `groupdel` :

```
$ groupdel ftpusers
```

Gestion graphique des utilisateurs et des groupes :

Vous pouvez également ajouter un utilisateur graphiquement à partir de l'utilitaire de configuration DrakeConf fourni avec la distribution ou Mandrake (l'ajout d'un utilisateur s'accompagne de l'ajout d'un groupe du même nom) ou à partir de l'utilitaire linuxconf disponible sur toutes les distributions linux (vous pouvez alors choisir le groupe d'appartenance de l'utilisateur).

Comme pour la plupart des tâches d'administration système, vous pouvez gérer les utilisateurs et les groupes très facilement avec Webmin :

Ajout d'un utilisateur sous Webmin

Rubrique "Système" puis "Utilisateurs et groupes"
Cliquer sur un utilisateur ou un groupe pour modifier leurs propriétés.
Cliquer sur "Créer un nouvel utilisateur" pour ajouter un utilisateur.
Cliquer sur "Créer un nouveau groupe" pour ajouter un groupe.

Astuces :

- ▶ Si vous êtes amenés à ajouter régulièrement des utilisateurs (pour ajouter par exemple des comptes ftp ou des comptes mails), vous avez intérêt à créer un "utilisateur Webmin" spécialement dédié à l'ajout de comptes utilisateurs et pour lesquels les paramètres par défauts sont déjà pré-réglés (groupe utilisateur et emplacement du répertoire personnel associé au compte).
- ▶ Pour des raisons de sécurité, il est important d'interdire l'accès à la machine aux utilisateurs Internet qui n'ont pas d'autres besoins que d'effectuer du ftp ou de consulter des mails. Pour cela l'interpréteur de commande (shell) des utilisateurs doit être remplacé par `/etc/ftponly` (et non par exemple `/bin/sh` qui autoriserait un accès telnet à la machine).

Exemple de fichier `/etc/ftponly` :

```
# !/bin/sh
#
# ftponly shell
#
trap "/bin/echo Sorry ; exit 0" 1 2 3 4 5 6 7 10 15
#
Admin=root@your-domain.com
#
/bin/echo
/      b      i      n      /      e      c      h      o
"*****"
/bin/echo " You are NOT allowed interactive access."
/bin/echo
/bin/echo " User accounts are restricted to ftp and web
access."
/bin/echo
/bin/echo " Direct questions concerning this policy to
$Admin."
/      b      i      n      /      e      c      h      o
"*****"
/bin/echo
#
#
```

```
exit 0
```

De même pour un utilisateur qui n'a besoin que d'accéder à sa boîte aux lettres, la seule commande que nous lui autorisons consiste à changer son mot de passe. Le shell d'un tel utilisateur est :
/bin/passwd

Attention, pour que les shells /etc/ftponly et /bin/passwd soient autorisés par le système, il faut les ajouter dans le fichier /etc/shell.

4.4. GESTION DES PROCESSUS

Un processus est un programme chargé en mémoire et en cours d'exécution. Contrairement à Windows 98, les systèmes UNIX sont des systèmes multitâches préemptifs, c'est à dire que chaque programme ou processus tournent indépendamment. Lorsqu'un processus est planté, le système continue à tourner car les processus sont traités indépendamment. La destruction d'un processus n'a pas d'effet sur l'exécution des autres processus.

Informations associées à chaque processus :

Pour chaque processus exécuté, le système d'exploitation stocke un certain nombres d'informations :

- Numéro unique du processus PID (Process IDentification) ;
- Numéro du processus parent PPID (Parent Process Identification) ;
- Numéro d'utilisateur PID (User IDentification) ayant lancé le processus ;
- Numéro du groupe GID (Group IDentification) ayant lancé le processus ;
- Durée de traitement utilisé (temps CPU) et priorité du processus ;
- Référence au répertoire de travail courant du processus ;
- Table de référence des fichiers ouverts par le processus.

Chaque processus peut créer lui-même des processus d'où la notion de processus parent. C'est le cas par exemple du serveur Apache : lors de son lancement, le processus père crée plusieurs processus fils afin de répondre indépendamment à plusieurs clients. La destruction du processus parent (parent process) entraîne la destruction de tous les processus fils (child process).

Afficher les processus avec la commande top :

La commande `top` permet d'afficher des informations en continu sur l'activité du système (quantité de RAM et pourcentage de CPU utilisés par les processus). Pour quitter la commande `top`, il suffit de taper la lettre `p`.

Commande ps :

Il s'agit de la commande la plus employée pour obtenir des informations sur les processus exécutés par le système. Cette commande permet de connaître les processus actifs à un moment donné.

```
$ ps
PID TTY TIME CMD
26687 pts/2 00:00:00 bash
26797 pts/2 00:00:00 ps
```

La commande `ps` sans arguments ne fournit que la liste des processus associés au terminal utilisé. Pour connaître tous les processus exécuté par le système, il est nécessaire d'ajouter l'argument `-aux`

```
$ ps
```

	affiche tous les processus lancés par l'utilisateur dans le terminal
\$ ps -x	affiche tous les processus lancés par l'utilisateur
\$ ps -aux	affiche tous les processus lancés par tous les utilisateurs
\$ ps -aux grep httpd	affiche tous les processus nommés httpd lancés

Commande pstree :

La commande `ps tree` permet d'afficher les processus sous forme d'arborescence : pratique pour voir les dépendances entre processus.

Tuer un processus avec la commande kill :

La commande `kill` permet d'envoyer un signal à un processus en cours.

Syntaxe : `kill -Numéro-de-signal PID`

Pour tuer un processus, c'est à dire obliger l'application à se terminer, nous utilisons le signal de numéro 9 (SIGKILL) qui oblige le processus à se terminer : cette option permet de tuer le processus quel que soit son état et même si celui-ci est planté ou instable.

Exemple : lancement de la commande `top`, retour au shell avec les touches `Ctrl + Z`, recherche du processus correspondant à la commande `top` et destruction de celui-ci.

```
$ top
(puis appuyez sur les touches Ctrl et Z)
$ ps -aux | grep top
pdrouot 27060 1.9 0.8 1736 1060 pts/1 T 09:02 0:00 top
pdrouot 27064 0.0 0.3 1332 512 pts/1 S 09:02 0:00 grep top
$ kill -9 27060
```

Opérateur : l'opérateur `&` permet de lancer plusieurs processus en parallèle alors que l'opérateur `;` lance les processus en série.

Exercice : comparer la commande `netscape &netscape &netscape &` avec la commande `netscape ; netscape ; netscape ; netscape`

Gestion des processus sous Webmin

Rubrique "Système" puis "Gestionnaire de processus" Très pratique car cette fonction permet de connaître les processus qui tournent ainsi que leur occupation mémoire et CPU depuis votre machine ou encore depuis n'importe quel poste connecté à Internet ce qui permet de vérifier si la machine est surchargée ou non sans avoir à être physiquement devant. Pour afficher les processus parents et leurs enfants, cliquez sur "PID".

Pour classer les processus en fonction de leur occupation mémoire ou CPU, cliquez sur "Mémoire" ou sur "Processeur".

Pour tuer un processus, cliquez sur le numéro du processus, puis cliquez sur "Envoyer le signal" avec "TERM" (terminer) en paramètre.

4.5. MONTAGE DE DISQUES

4.5.1. Montage manuel

Dans les systèmes UNIX, les périphériques, les disques durs et les partitions sont gérées sous forme de fichiers contenus dans le répertoire `/dev` : chaque fichier du répertoire `/dev` correspondent à une sorte de driver.

Pour utiliser un périphérique, il faut attacher le fichier périphérique correspondant à un point de montage.

Ainsi `/dev/hda` correspond au premier disque dur IDE, `/dev/hda1` à la première partition du disque dur IDE, `/dev/hda2` à la seconde partition, `/dev/hdb` au second disque dur IDE, `/dev/hdb1` à la première partition du second disque dur, etc..

De même `/dev/sda` correspond au premier disque dur SCSI, `/dev/sda1` à la première partition du disque dur SCSI, etc..

`/dev/cdrom` correspond souvent au lecteur de cd-rom et `/dev/fd0` au premier lecteur de disquette.

Pour "monter" un périphérique ou une partition, il faut utiliser la commande `mount`.

Syntaxe : `mount -t type-du-support fichier-périphérique point-de-montage`

Traditionnellement, les périphériques tels que les disques et partitions sont montés dans le répertoire `/mnt` : par exemple `/mnt/cdrom` pour un cdrom.

Les principaux type de support sont :

`ext2` : filesystem Linux ; `msdos` : disque de type MS-DOS ; `vfat` : disque de type Windows ;
`iso9660` : CD-ROM ;

Exemples :

Montage d'un lecteur de CD-ROM : `$ mount -t iso9660 /dev/hdb /mnt/cdrom`

Montage d'une partition Windows : `$ mount -t vfat /dev/hda1 /mnt/win98`

Pour démonter un périphérique, il faut utiliser la commande `umount`.

Exemple : pour changer de CD, il faut au préalable démonter le CD courant avec la commande :

```
$ umount /mnt/cdrom
```

A noter qu'à l'issu de l'installation de la distribution Mandrake Corporate Server, le lecteur de CD-ROM est monté par défaut ainsi que l'éventuelle partition Windows.

4.5.2. Montage automatique

Le fichier `/etc/fstab` (File System Table) contient toutes les informations concernant le montage automatique des disques au démarrage du système.

```
$ cat /etc/fstab
/dev/hda5 / ext2 defaults 1 1 none
/dev/hda7 /home ext2 defaults 1 2
/dev/cdrom /mnt/cdrom auto user,noauto,nosuid,exec,nodev,ro 0 0
/dev/fd0 /mnt/floppy auto sync,user,noauto,nosuid,nodev 0 0
```

```
/dev/hda1 /mnt/windows vfat user,exec,umask=0 0 0 none /proc proc defaults 0
0
/dev/hda6 swap swap defaults 0 0
```

Liste des paramètres associés à chaque périphérique :

- device (périphérique) de la partition ;
- point de montage de la partition ;
- type de la partition ;
- options (gestion des droits) ;
- sauvegarde (si ce champ est non nul, l'utilitaire dump doit sauvegarder ce système de fichier) ;
- test et réparation (si ce champs est non nul, l'utilitaire fsck doit tester la partition avant de la monter. Le numéro correspond à l'ordre des tests).

L'ajout d'un périphérique dans le fichier `/etc/fstab` permet de le monter automatiquement au démarrage.

4.6. INSTALLATION DE NOUVEAUX LOGICIELS

L'installation de nouveaux logiciels s'effectue soit à partir des sources, soit à partir d'un binaire (application déjà compilée), soit à partir d'un paquetage rpm.

4.6.1. Installation à partir des sources

L'installation à partir des sources consiste à compiler des lignes de code (en C ou en C++) puis à installer le binaire produit. Les avantages de cette méthode sont multiples :

- un même code source peut être compilé sur n'importe quelle machine UNIX et ce quel que soit son processeur (Intel, Alpha, Risc, PowerPC, etc..) ;
- vous pouvez spécifier le répertoire où l'application doit être installée ;
- vous pouvez compiler l'application avec des options spécifiques (ajout de modules particuliers, optimisation du binaire en fonction du processeur, etc...)
- les sources étant moins volumineux que les binaires, le téléchargement des sources d'une application est beaucoup plus rapide que le téléchargement du binaire ou du paquetage rpm correspondant.

Qu'elle que soit l'application, la procédure d'installation est identique :

1. Préparation de la compilation par la commande : `$./configure -prefix=répertoire-de-destination`
2. Compilation de l'application par la commande : `$ make`
3. Installation de l'application par la commande : `$ make install`

Il ne vous reste ensuite plus qu'à exécuter le script de lancement de l'application et si la nouvelle application doit être lancée systématiquement au démarrage de la machine (cas des services Internet), il vous faut également copier le script de lancement dans le répertoire `/etc/rc.d/init.d` (cf. chapitre *Lancement de programmes au démarrage*)

4.6.2. Installation à partir d'un binaire

Pour une application donnée, il existe peut être déjà une version binaire compilée pour votre processeur. Il ne vous reste plus qu'à télécharger l'application, la décompresser puis la déplacer dans le répertoire de votre choix. A noter

que dans la désignation employée dans les distributions binaires : intel-386 désigne un processeur Intel de type 386, intel-486 de type 486, intel-586 de type Pentium, intel-686 de type Pentium II, etc...

Tout comme pour une installation à partir des sources vous devez ensuite lancer l'application et vérifier le cas échéant si celle-ci est lancée au démarrage.

4.6.3. Installation à partir d'un paquetage rpm

RPM (Red Hat Package Manager) est un puissant gestionnaire d'applications permettant d'installer, de mettre à jour, de vérifier ou de désinstaller des composants logiciels.

Pour installer un nouveau paquetage appli.rpm :

```
$ rpm -ivh appli.rpm
```

Attention, si vous installez un paquetage par cette méthode et qu'il existe déjà sur votre système dans une version inférieure, vous risquez d'avoir des problèmes pour le désinstaller (voir plus bas). Les paramètres -vh permettent d'ajouter une barre de progression.

Pour mettre à jour (upgrader) un paquetage :

```
$ rpm -Uvh appli.rpm
```

Pour supprimer un paquetage :

```
$ rpm -e appli.rpm
```

Afficher la liste de tous les paquetages installés :

```
$ rpm -qa
```

Vérifier à partir du nom si un paquetage est déjà installé :

```
$ rpm -qa | grep php
```

Lister le contenu d'un paquetage :

```
$ rpm -ql appli.rpm
```

Si vous avez besoin d'une application précise sous le format rpm, vous devriez pouvoir la trouver sans trop de difficultés sur le site <http://www.rpmfind.net>.

4.6.4. Installation à partir d'un paquetage Debian

La distribution Debian utilise son propre système de paquetage dont les fichiers sont reconnaissables par l'extension .deb .

Pour gérer les paquetages il existe deux programmes :

- dpkg
- apt-get

Pour simplifier, on peut considérer que apt est une surcouche sur dpkg.

La première chose à faire est de définir par quel moyen nous allons accéder aux paquetages, soit par CDROM soit par une liaison réseau. Le programme `apt-setup` permet de configurer les sources de paquetage de manière interactive.

Une autre manière possible est d'ajouter directement les informations dans le fichier `/etc/apt/sources.list`.

Même si on utilise les cdroms pour installer les paquetages il est quand même conseillé d'ajouter la ligne `:deb http://security.debian.org/ stable/updates main contrib non-free` au fichier `/etc/apt/sources.list` ce qui permet de télécharger les derniers paquetages qui corrigent des problèmes de sécurité. Après une modification de ce fichier il faut lancer la commande `apt-get update` pour que le système puisse construire la base de données des paquetages en prenant en compte la nouvelle source.

Commandes usuelles :

```
apt-get install Nom_Du_Paquetage : Installe un nouveau paquetage
apt-get remove Nom_Du_Paquetage : Supprime d'un nouveau paquetage
dpkg -S nom_du_fichier : Donne le paquetage auquel appartient le fichier
dpkg -i nom_du_fichier : Installe le paquetage précédemment téléchargé
dpkg -L Nom_Du_Paquetage : Liste tous les fichiers d'un paquetage
apt-get update : Mise à jour de la liste des paquetages
apt-get dist-upgrade : Mise à jour de tous les paquetages
dpkg-reconfigure Nom_de_Paquetage : reconfigure un paquetage déjà installé
apt-cache search XXXXX : cherche XXXX dans le nom ou la description des paquetages disponibles
```

Exemple d'utilisation d'apt-get :

```
debian:/home/fred# apt-get install vim<br>
Reading Package Lists... Done<br>
Building Dependency Tree... Done<br>
The following extra packages will be installed:<br>
 libgpm1 <br>
The following NEW packages will be installed:<br>
 libgpm1 vim <br>
0 packages upgraded, 2 newly installed, 0 to remove and 0 not
upgraded.<br>
Need to get 3796kB of archives. After unpacking 12.3MB will be
used.<br>
Do you want to continue? [Y/n]<br>
```

4.6.5. Lancement de programmes au démarrage

Au départ le kernel Linux lance le premier process `init`. `init` va alors lire son fichier de configuration (`/etc/inittab`). Les programmes qui vont être exécutés dépendent du `run-level` spécifié dans le fichier `inittab`. Par défaut la distribution Mandrake utilise le niveau 5 alors que Debian utilise le niveau 2.

Le `run-level` qui va être exécuté peut être fixé au lancement en donnant en paramètre `init=X` au moment du boot sur l'invite de commande `lilo`.

Chaque `run-level` correspond à une utilisation particulière, où certains services vont être lancés au démarrage ou non, suivant le choix de l'administrateur.

A chaque `run-level` est associé un répertoire `/etc/rcX.d/` ou `X` est le `run-level`. Dans ce répertoire on trouve les scripts qui vont être exécutés au démarrage. Si on examine le contenu du répertoire `/etc/rc2.d/` on voit qu'il n'y

a aucun "vrai" fichier, mais seulement des liens sur des scripts qui se trouvent dans `/etc/init.d/`. Ceci permet de centraliser les scripts de démarrage qui sont souvent commun à plusieurs run-level.

Pour lancer automatiquement une application au démarrage, il est nécessaire d'écrire un script de lancement de l'application et de le placer dans le répertoire `/etc/init.d/` puis de faire un lien (grâce à la commande `ln`) dans le ou les run-level concernés.

Par exemple voici le fichier que l'on va placer dans `/etc/init.d` :

```
#!/bin/sh
#
#Exemple d'execution d'une commande au démarrage
#
#Activation du mode DMA sur /dev/hda
/sbin/hdparm /dev/hda -d 1
```

on sauvegarde ce script sous le nom `hdparm` dans le répertoire `/etc/init.d`. Il faut le rendre exécutable :

```
chmod a+x /etc/init.d/hdparm
```

puis le mettre dans le run-level désiré :

```
ln -s /etc/init.d/hdparm /etc/rc2.d/S80hdparm
```

pour la Debian, ou :

```
ln -s /etc/init.d/hdparm /etc/rc5.d/S80hdparm
```

pour la Mandrake.