

Quelques répertoires et fichiers importants

Philippe Vignoles

Alix Mascret

Petit tour dans l'arborescence.

1. Présentation

Le but de ce document est de décrire certains dossiers de l'arborescence dont l'utilité n'est pas évidente et qui pourtant sont essentiels pour la bonne gestion du système.

Tous les dossiers de l'arborescence du système ne seront pas traités. Nous verrons notamment les dossiers `/etc`, `/mnt`, `/var`, `/dev`, `/proc` et `/tmp`.

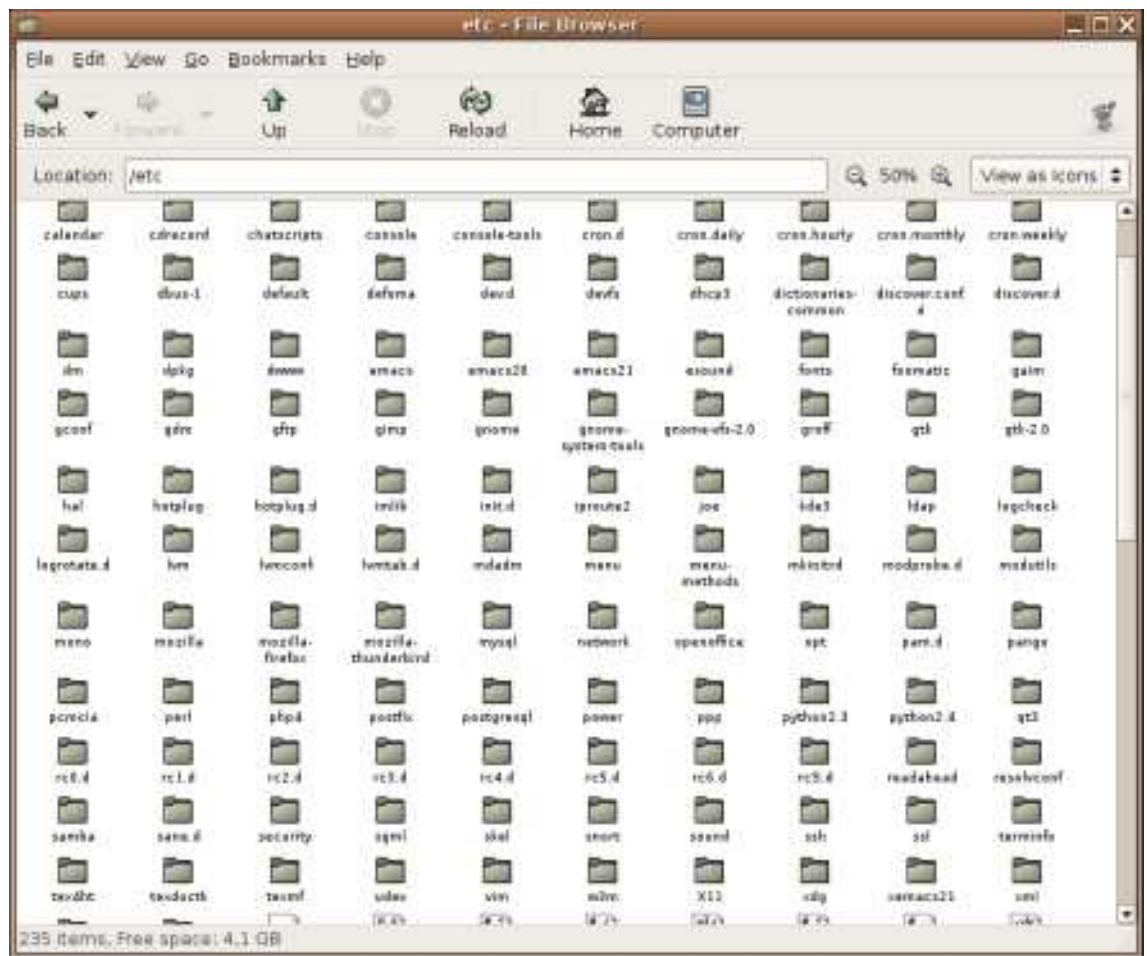
À la fin de cette activité, vous serez familiarisé avec le contenu de ces dossiers qui interviennent d'une manière incontournable dans l'administration du système.

1.1. L'arborescence de `/etc`

1.1.1. arborescence

Le dossier `/etc` regroupe la majorité des fichiers de configuration pour les différents programmes utilisables dans GNU/Linux. Ce dossier a une structure assez dense comme le montre la figure ci-dessous.

Figure 1. Arborescence de /etc



Dans ce dossier, on retrouve par exemple :

- /etc/cron.d
- /etc/cron.daily?

qui sont les dossiers contenant les fichiers de configuration du démon cron.

On retrouve aussi les dossiers contenant les scripts d'initialisation :

- /etc/init.d
- /etc/rcx.d

Ainsi que les dossiers de configuration de nombreux serveurs disponibles nativement dans les distributions de Linux :

- /etc/cups (serveur d'impression)
- /etc/cvs (serveur de gestion des versions)

- /etc/httpd (serveur web : apache)
- /etc/samba (émulation d'un serveur windows)
- /etc/ssh (shell sécurisé)
- /etc/webmin (serveur d'administration de l'environnement)
- /etc/X11 (serveur graphique)

Les fichiers de configuration contenus dans le dossier /etc (ou ses sous-dossiers) sont des fichiers textes : ils contiennent des informations littérales exploitables avec n'importe quel éditeur de texte. De plus, ils sont également souvent bien commentés ce qui simplifie leur lecture. On pourra donc éditer facilement ces fichiers de configuration grâce à des éditeurs standards comme joe, vi, emacs, nedit...

Lorsque le service à configurer est « simple » (ne nécessitant pas plusieurs fichiers de configuration), il n'y aura pas de dossier spécifique et le fichier de configuration sera présent directement dans la racine du dossier /etc.

1.2. Fichiers de configuration remarquables

Exemple de contenu pour /etc/fstab :

```
/dev/hdc5 / ext3 defaults 1 1
/dev/hdb /mnt/cdrom auto umask=0,user,iocharset=iso8859-15\
,codepage=850,noauto,ro,exec,users 0 0
none /mnt/floppy supermount dev=/dev/fd0,fs=ext2:vfat,\
--,umask=0,iocharset=iso8859-15,sync,codepage=850 0 0
/dev/hda1 /mnt/windows ntfs umask=0,iocharset=iso8859-15,\
nls=iso8859-15,ro 0 0
none /proc proc defaults 0 0
/dev/hdc1 swap swap defaults 0 0
```

Ce fichier contient les déclarations des systèmes de fichiers à attacher à l'arborescence. Par défaut, ils sont montés automatiquement au démarrage du système.

On fournira les paramètres séparés entre eux par un espace, dans l'ordre :

- Périphérique (le périphérique à attacher, déclaré par /dev/hda1, /dev/fd0, /dev/sdc1...)
- Point de montage (position dans l'arborescence où attacher le périphérique. Exemple : /mnt/disque1)
- Type du système de fichiers (ext2, ext3, vfat...)
- Options (rw pour lecture+écriture, ro pour lecture seule, noauto pour ne pas monter automatiquement ce système de fichiers?)
- Dumpfreq (pour les systèmes de fichiers nécessitant une sauvegarde. La valeur par défaut est 0)

- Passno (numéro d'ordre pour la vérification des systèmes de fichiers par fsck. 0 indique qu'il ne faut pas les vérifier)

Ce fichier contiendra aussi la déclaration des partitions d'échange. Ces partitions n'apparaîtront plus pour l'utilisateur et elles contiendront le fichier d'échange pour gérer la mémoire virtuelle du système d'exploitation.

Exemple de contenu de `/etc/mtab` :

```
/dev/hdc5 / ext3 rw 0 0
none /proc proc rw 0 0
none /proc/bus/usb usbfs rw 0 0
none /sys sysfs rw 0 0
(...)
```

Ce fichier contient les différents systèmes de fichiers actuellement montés dans l'arborescence. Ce fichier reprend globalement les mêmes informations que celles contenues dans `/etc/fstab`.

Extrait de contenu de `/etc/passwd` :

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/sh
daemon:x:2:2:daemon:/sbin:/bin/sh
adm:x:3:4:adm:/var/adm:/bin/sh
lp:x:4:7:lp:/var/spool/lpd:/bin/sh
(...)
```

Ce fichier contient les noms des utilisateurs déclarés dans le système. Les différentes informations sont séparées par des deux points. Par extension, il faudrait plutôt parler de comptes système. Ces comptes étant utilisés soit par des utilisateurs, soit par des processus. Du point de vue du système, ce sont ces comptes qui lui permet de savoir si tel processus peut ou ne peut pas utiliser une ressource.

Sans trop entrer dans le détail voici une approche plus concrète. Pour être plus concret, un compte utilisateur permettra à ce dernier d'être authentifié par le système et d'accéder, en fonction de ses droits à telle ou telle ressource.

Mais que se passe-t-il s'il s'agit d'un processus, par exemple un daemon souhaitant journaliser la dernière requête qui lui a été passée. Il faut qu'il accède à un fichier. Le système ne lui en laissera pas la possibilité si ce processus n'est pas authentifié. Chaque service utilise donc un compte système pour se faire authentifier par le système.

On trouve dans l'ordre :

- Le nom de l'utilisateur (par exemple : bin)

- Le mot de passe (ici remplacé par x car la gestion des mots de passe est délocalisée dans le fichier /etc/shadow. Sans la gestion du mot de passe protégé par shadow, on le trouverait à cet emplacement)
- L'identifiant : uid (pour bin, on trouve l'identifiant 2)
- Le groupe principal défini par son gid (le groupe principal attribué à halt a le gid égale à 0. Dans le fichier group, on trouve que le groupe ayant le gid=0 correspond au group root)
- Un commentaire (le commentaire est un texte en clair facultatif pour décrire l'utilisateur. Pour halt on trouve le commentaire halt)
- Le dossier personnel (le dossier personnel de l'utilisateur halt est /sbin)

L'interpréteur de commandes attribué à l'utilisateur (l'interpréteur de commandes attribué à halt est /sbin/hat et correspond directement à la commande qu'il faut exécuter lors de l'appel à l'utilisateur halt)

Exemple de contenu de /etc/group :

```
root:x:0:
bin:x:1:
daemon:x:2:messagebus
sys:x:3:
adm:x:4:
(...)
xgrp:x:102:xfs
```

Ce fichier contient la déclaration des noms de groupes d'utilisateurs. On distingue donc les différentes informations :

- Nom du groupe (par exemple : xgrp)
- Mot de passe (généralement non renseigné (ou plutôt invisible dans ce fichier) et remplacé par un x)
- L'identifiant du groupe (gid ; par exemple 102 pour le groupe xgrp)
- La liste des utilisateurs appartenants à ce groupe (les noms sont renseignés les uns à la suite des autres et séparés les uns des autres par une virgule. Par exemple pour le groupe xgrp, on trouve comme utilisateur déclaré : xfs)

Exemple : le groupe staff est identifié par le numéro 756 et jojo, marcel, sandrine, mickael et theboss en sont membres :

```
staff:x:756:jojo,marcel,sandrine,mickael,theboss
```

Extrait du fichier /etc/shadow :

```
root:$1$iaM6kAQDbnHg/TOEjL/v$SjOMTPcw9Z/:12788:0:99999:7:::
bin:*:12788:0:99999:7:::
daemon:*:12788:0:99999:7:::
adm:*:12788:0:99999:7:::
lp:*:12788:0:99999:7:::
sync:*:12788:0:99999:7:::
(...)
```

Ce fichier est présent dans les systèmes. Il est maintenant presque systématiquement utilisé afin de garantir la protection des comptes utilisateurs. En effet, il s'agit d'un fichier qui n'est accessible qu'à l'utilisateur root, aucun autre utilisateur n'a le droit de le manipuler. On trouvera dans ce fichier les champs suivants, séparés par les uns des autres par un deux-points :

- Nom de l'utilisateur (reprise du fichier /etc/passwd. Par exemple : root)
- Mot de passe crypté (pour root, on trouve : 1\$iaM6kAQDbnHg/TOEjL/v\$SjOMTPcw9Z/)
- Âge du mot de passe : nombre de jours écoulés depuis le 1er janvier 1970 jusqu'au dernier changement de mot de passe (pour root : 12788 jours)
- Grâce de validité : nombre de jours durant lesquels le mot de passe est encore valide (pour root : 0)
- Durée de vie du mot de passe : nombre de jours après lesquels le mot de passe doit être changé (pour root : 99999, ce qui correspond à illimité)
- Avertissement : nombre de jours avant l'expiration du mot de passe impliquant l'avertissement de l'utilisateur (pour root : 7)
- Tolérance avant désactivation : nombre de jours après l'expiration provoquant la désactivation du compte (pour root : non renseigné)
- Durée de désactivation : numéro du jour depuis le 1er janvier 1970 à partir duquel le compte a été désactivé (pour root : non renseigné)
- Champs réservé (pour root : non renseigné)

Extrait du fichier /etc/inittab :

```
id:3:initdefault:

# System initialization.
si::sysinit:/etc/init.d/rcS

# Sur certains systèmes
# si::sysinit:/etc/rc.d/rc.sysinit

10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
14:4:wait:/etc/rc.d/rc 4
15:5:wait:/etc/rc.d/rc 5
16:6:wait:/etc/rc.d/rc 6
```

```
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

Ce fichier configure le démarrage du système en appliquant le niveau de démarrage indiqué par `initdefault` (ici, il s'agit du niveau 3).

Il précise aussi quel est le script d'initialisation à exécuter en premier défini par `sysinit` (par exemple : `/etc/init.d/rcS`).

Il définit aussi les chemins d'accès aux dossiers contenant les scripts à exécuter en fonction du niveau de démarrage choisi (exemple : `l3:3:wait:/etc/rc.d/rc 3`).

Il est possible de gérer l'action appliquée par la séquence de touches `Ctrl+Alt+Suppr` (`Ctrl+Alt+Del`).

Il configure aussi le comportement du système lors d'une perte d'alimentation électrique lorsque l'ordinateur est connecté à un onduleur. Par exemple :

```
pf::powerfail:/sbin/shutdown -f -h +2 \
    "Power Failure; System Shutting Down"
```

De même, il gère l'action à réaliser si l'alimentation électrique a été rétablie avant que la procédure précédente n'ait été totalement appliquée. Par exemple :

```
pr:12345:powerokwait:/sbin/shutdown -c \
    "Power Restored; Shutdown Cancelled")
```

Enfin, il précise aussi les bureaux virtuels (tty) qui seront utilisés en fonction des niveaux de démarrage. Par exemple, on remarque que les 6 bureaux virtuels standards seront chargés pour les niveaux 2 à 5 :

```
3:23:respawn:/sbin/getty 38400 tty3
```

1.3. Autres fichiers de configuration importants

On peut aussi citer la présence des fichiers de configuration suivants :

- /etc/bashrc (configuration de l'interpréteur de commandes bash)
- /etc/filesystems (systèmes de fichiers connus par défaut)
- /etc/gshadow (équivalent du fichier /etc/shadow appliqué aux groupes)
- /etc/hosts (déclaration des ordinateurs du réseau)
- /etc/hosts.allow (déclaration explicite des ordinateurs autorisés à se connecter au système excluant tous les autres)
- /etc/hosts.deny (déclaration des ordinateurs non autorisés à se connecter au système acceptant tous les autres)
- /etc/lilo.conf (configuration du lanceur de Linux)
- /etc/php.ini (configuration de l'interpréteur de commandes php)
- /etc/rc (script appelé au changement du niveau de démarrage) (Pas sur Debian)
- /etc/rc.local (dernier script à être exécuté après tous les autres scripts exécutés au démarrage du système selon le niveau) (Pas sur Debian)
- /etc/rc.sysinit (script exécuté une seule fois au démarrage du système) sur certaines distributions comme RedHat
- /etc/resolv.conf (contient les informations relatives aux serveurs de noms utilisés pour la résolution des noms en fonction du numéro IP de l'ordinateur)
- /etc/securetty (déclaration des différents terminaux virtuels tty, vc?)
- /etc/shells (déclaration des interpréteurs de commandes utilisables par défaut : bash, sh...)
- /etc/termcap (fichier de description des types de terminaux)
- /etc/xinetd.conf (configuration du serveur de services internet)

Bien entendu, cette liste est loin d'être exhaustive?

1.4. Le dossier /mnt

1.4.1. Le contenu du dossier /mnt

Ce dossier est normalement prévu pour accueillir les points de montage des systèmes de fichiers « étrangers » ou amovibles. Ces points de montage peuvent être prédéfinis dans le fichier /etc/fstab avec montage automatique ou non au lancement du système.

Ainsi, on y trouvera les points de montage caractéristiques : /mnt/floppy, /mnt/cdrom, pour les médias amovibles ou encore /mnt/hda1 (par exemple) si le volume /dev/hda1 contient un système de fichiers indépendant de celui appartenant nativement à l'arborescence de Linux. Ce cas se rencontre souvent lorsqu'on possède sur le même disque une partition pour Windows (/dev/hda1) et une autre pour Linux (/dev/hda5).

En général /mnt/sda4 (pour les lecteurs zip). Les lecteurs zip sont gérés comme des périphériques scsi et sont très souvent associés à la 4e partition du disque /dev/sda. Le périphérique /dev/sda1 est généralement associé aux clés usb...

1.4.2. Les commandes mount et umount

La commande mount :

Le montage des systèmes de fichiers s'effectue grâce à la commande mount et le démontage grâce à la commande umount.

La commande mount peut être simple d'utilisation :

```
mount périphérique point-de-montage
```

où périphérique représente un système de fichiers existant dans le dossier /dev (par exemple /dev/hda1), point-de-montage représente le dossier où sera monté le système de fichiers.

Par exemple :

```
mount /dev/hda1 /mnt/windows
```

Le montage s'effectue en utilisant les systèmes de fichiers connus par défaut. Avec cette commande, on peut rajouter des options par exemple lorsque le système de fichiers ne semblent pas connu a priori :

```
mount -t ntfs /dev/hda1 /mnt/windows
```

avec l'option -t ntfs, on précise explicitement le type de système de fichiers à monter.

Il est possible aussi de monter un système de fichiers stocké dans un fichier image ISO. Par exemple, si on dispose du fichier appelé image.iso qui correspond à l'image du CD-ROM d'une distribution, on pourra consulter le contenu du fichier en la montant dans l'arborescence. Si on souhaite attaché le système de fichiers du fichier iso dans un dossier, il est indispensable que ce dernier existe. Si tel n'est pas le cas, il faudra au préalable créer ce dossier.

Par exemple, on veut monter cette image dans le dossier /mnt/image, on tapera la commande suivante (le fichier se trouve par exemple dans le dossier personnel de l'utilisateur ulyse) :

```
mount -o loop -t iso9660 image.iso /mnt/image
```

L'option -o permet de rajouter l'option « loopback » et on précise le type de système de fichiers avec -t iso9660 (système de fichiers généralement utilisé pour les CD-ROM).

Les options classiques de la commande mount sont les suivantes :

-a Monte tous les systèmes de fichiers listés dans /etc/fstab. Exception faite de ceux marqués comme « noauto », ou exclus par le drapeau -t, ou encore ceux qui sont déjà montés.

-d Tout effectuer à l'exception de l'appel système de montage réel. Cette option est utile conjointement avec le drapeau -v pour déterminer ce que mount est en train d'essayer de faire.

-f Force le montage d'un système de fichiers non propre (dangereux), ou force la révocation de l'accès en écriture quand on modifie l'état de montage d'un système de fichiers de l'accès lecture+écriture à l'accès lecture seule.

-F Utilisé en association avec -a. Permet de lancer un processus par montage ce qui effectue un montage parallèle des systèmes de fichiers.

-r Monte le système de fichiers en lecture seule. C'est identique à l'utilisation de l'argument ro avec l'option -o.

-t fstype Monte le système de fichiers comme étant du type de système donné, ou monte seulement les systèmes de fichiers du type donné, si l'option -a est précisée.. "ufs" est le type de système de fichiers par défaut.

-u Met à jour les options de montage sur le système de fichiers.

-v Rend la commande prolixe.

-w Monte le système de fichiers en lecture+écriture.

-o Accepte une liste d'options séparées par des virgules, dont les suivantes :

- *nodelv* Ne pas prendre en compte les périphériques spéciaux sur le système de fichiers. C'est une option de sécurité utile.
- *noexec* Ne pas autoriser l'exécution de binaires sur ce système de fichiers. C'est également une option de sécurité utile.
- *nosuid* Ne pas prendre en compte les indicateurs setuid ou setgid sur le système de fichiers. C'est également une option de sécurité utile.
- *loop* Périphérique boucle utilisé avant le montage dans le point de montage

La commande umount :

La commande umount sert pour démonter un système de fichiers monté avec la commande mount. Il suffit généralement de préciser comme paramètre le point de montage (bien que l'utilisation explicite

du périphérique est possible). Par exemple, pour démonter le système de fichiers /dev/hda1 (monté en /mnt/windows) :

```
umount /mnt/windows
ou
umount /dev/hda1
```

Pour démonter le système de fichiers ISO qui a été monté avec le périphérique loop sera démonté en précisant l'option -d (ce qui permet de libérer aussi la boucle en même temps)

```
umount -d /mnt/knoppix
```

Si un système de fichiers « résiste » au démontage, on peut tenter d'utiliser l'option -f (pour forcer le démontage) :

```
umount -fd /mnt/knoppix
```

Supermount :

L'option supermount apporte quelques améliorations. Le montage des systèmes de fichiers dans le monde Unix n'est théoriquement pas très souple car il faut les monter souvent « à la main » lorsqu'il s'agit de périphériques amovibles (disquette, CD-Rom,?).

Supermount permet de dynamiser l'accès à ces supports amovibles. Cette commande est relativement simple à utiliser :

Pour activer la prise en charge supermount pour les périphériques concernés (floppy, CD-ROM) :

```
supermount enable
```

```
none /mnt/cdrom supermount fs=udf:iso9660,\
dev=/dev/hdb,--,codepage=850,users,\
iocharset=iso8859-15,ro,nosuid,umask=0,exec,nodev 0 0
none /mnt/floppy supermount dev=/dev/fd0, \
fs=ext2:vfat,--,umask=0,iocharset=iso8859-15,\
sync,codepage=850 0 0
```

Pour retirer la prise en charge supermount aux périphériques :

```
supermount disable
```

```
/dev/hdb /mnt/cdrom auto umask=0,user,\
iocharset=iso8859-15,codepage=850,\
```

```
noauto,ro,exec,users 0 0
/dev/fd0 /mnt/floppy auto codepage=850,\
iocharset=iso8859-15,noauto,nosuid,umask=0,\
sync,user,unhide,nodev 0 0
```

L'application de la commande est temporaire. Après redémarrage de l'ordinateur, la configuration précédente sera ré-utilisée. Pour rendre la modification définitive, il faut ajouter l'option `-i` qui provoquera l'écriture directement dans le fichier de destination par défaut `/etc/fstab`.

```
supermount -i enable
```

Avec l'option `-f "nom du fichier"` il est possible de préciser également le nom du fichier de destination : `/etc/fstab` par défaut.

1.5. Le dossier /var

Le dossier `/var` contient d'une manière générale les données dont la modification est relativement fréquente. Les données qu'il contient sont liées au fonctionnement de nombreux services.

Voyons l'arborescence de `/var`. Des différences peuvent exister d'une distribution à une autre.

1.5.1. Le dossier /var/cache

On constate que dans cette arborescence, se trouve un dossier `/var/cache`. Ce dossier regroupe les fichiers traités fréquemment et donc placés en cache afin d'accélérer leurs traitements par le service (par exemple les fichiers servis par le serveur web sont contenus dans le dossier `/var/cache/apache` ; ceux utilisés par le service traitant les pages manuels sont stockés dans `/var/cache/man` ; même remarque pour samba avec le dossier `/var/cache/samba`).

1.5.2. Le dossier /var/ftp

Ce dossier 'peut' (cela dépend de la configuration du serveur ftp), contenir les fichiers gérés par un serveur ftp lors de l'utilisation du compte `anonymous`.

1.5.3. Le dossier /var/lib

On y trouve de nombreux dossiers de configuration de divers services tels que : `dhcp` (serveur de noms dynamique), `mysql` (serveur de base de données), `postgresql` (autre serveur de base de données), `samba` (serveur de partage de fichiers)...

1.5.4. Le dossier /var/lock

Ce dossier contient les fichiers de verrous des applications.

1.5.5. Le dossier /var /run

Ce dossier contient l'ensemble des identifiants de processus actifs (pid).

1.5.6. Le dossier /var/www

De la même manière que pour le serveur ftp, on trouvera aussi un dossier relatif au serveur web. Le dossier /var/www va contenir les pages des différents sites web gérés par le serveur web (souvent un serveur apache).

1.6. Les fichiers log

1.6.1. Le dossier /var/log

Dans le dossier /var, il est important de détailler le dossier /var/log. En effet, ce dossier contient de nombreux fichiers de suivi de l'activité des services et du système.

On trouvera donc des fichiers relatifs à l'authentification des utilisateurs dans le système : */var/log/auth.log* ou des informations liées au démarrage du système : */var/log/boot.log*, */var/log/dmesg*

On y trouvera aussi des fichiers traçant l'activité du système: */var/log/messages*, */var/log/syslog*

D'autres fichiers permettront de surveiller tous types d'activité.

1.7. Les autres dossiers caractéristiques

1.7.1. Le dossier /dev

On trouvera dans ce répertoire l'ensemble des périphériques gérés par le système d'exploitation. Il s'agit d'interfaces de communication entre le système et les périphériques. On y trouve par exemple :

```
fd (floppy disk : disquette)
fd0, fd1
hda (hard disk : disque dur 1 IDE1)
hda1, hda2, ...
hdb (hard disk : disque dur 2 IDE1)
```

cdrom (qui est généralement un lien vers le périphérique identifié dans la chaîne IDE, SCSI ou USB comme étant le lecteur de CD-ROM : si le CD-ROM correspond au périphérique hdc, alors cdrom est un lien symbolique vers hdc).

dvd (même remarque que pour cdrom)

sr1, sr2 (périphérique correspondant au graveur de CD-ROM/DVD-ROM)

loop (périphérique de bouclage)

tty1, tty2... (périphériques de terminaux virtuels)

mouse (périphérique de pointage : souris. Il s'agit d'un lien vers le périphérique correspondant sur le port série : psaux ou usb : usbmouse)

ram0, ram1... (périphériques mémoires)

sda, sda1, sda2... (périphériques SCSI)

1.7.2. Le dossier /proc

Ce répertoire correspond à un point de montage créé par le système (déclaré dans /etc/fstab).

On y trouve des fichiers fournissant des informations sur le matériel, la configuration du noyau et sur les processus en cours d'exécution. Par conséquent, chaque processus possédera un dossier dont le nom reprend l'identifiant de processus (pid). Chaque dossier de processus contient au moins les fichiers suivants

- cmdline (contient la ligne de commande qui a créé le processus)
- status (information sur l'état du processus : en attente, en exécution, propriétaire)
- exe (lien symbolique vers le fichier exécutable utilisé par le processus)

Dans le dossier /proc, les fichiers donnant des informations générales sur le système sont par exemple :

- uptime (donne le temps de fonctionnement du système)
- stat (statistiques sur l'utilisation du système : utilisation du processeur, mémoire..)
- meminfo (récapitule l'utilisation de la mémoire)
- cpuinfo (description des processeurs du système)

1.8. Le dossier /tmp

Ce dossier permet de stocker des données temporaires. Il s'agit d'un dossier qui reçoit des données provenant de l'activité des services lancés par les utilisateurs.

Chaque données écrites dans ce dossier possèdera donc les droits de son propriétaire. Par conséquent, à part de rares exceptions, les utilisateurs ne devraient pas pouvoir consulter les fichiers ne leur appartenant pas.

2. Copyright

Cette documentation est soumise aux termes de la Licence de Documentation Libre GNU (GNU Free Documentation License) (<http://www.gnu.org/copyleft/fdl.html>).

Les programmes sont soumis aux termes de la Licence Générale Publique GNU (GNU General Public Licence) (<http://www.gnu.org/copyleft/gpl.html>).