

L'arborescence Linux

Table des matières

L'arborescence Linux.....	1
Introduction.....	1
Organisation.....	1
Principaux répertoires.....	3
1. /bin, /sbin, /lib.....	3
2. /boot.....	3
3. /dev.....	3
4. /home.....	5
5. /root.....	5
6. /tmp.....	5
7. /lost+found.....	6
8. /mnt.....	6
9. /proc.....	6
10. /sys.....	7
11. /usr.....	7
12. /var.....	8
13. /run.....	9
14. /etc.....	10
Montages et accès aux médias amovibles.....	10
1. Commande mount.....	11
2. Commande umount.....	12
3. Boîte à outils mtools.....	13
Exercices.....	14

Introduction

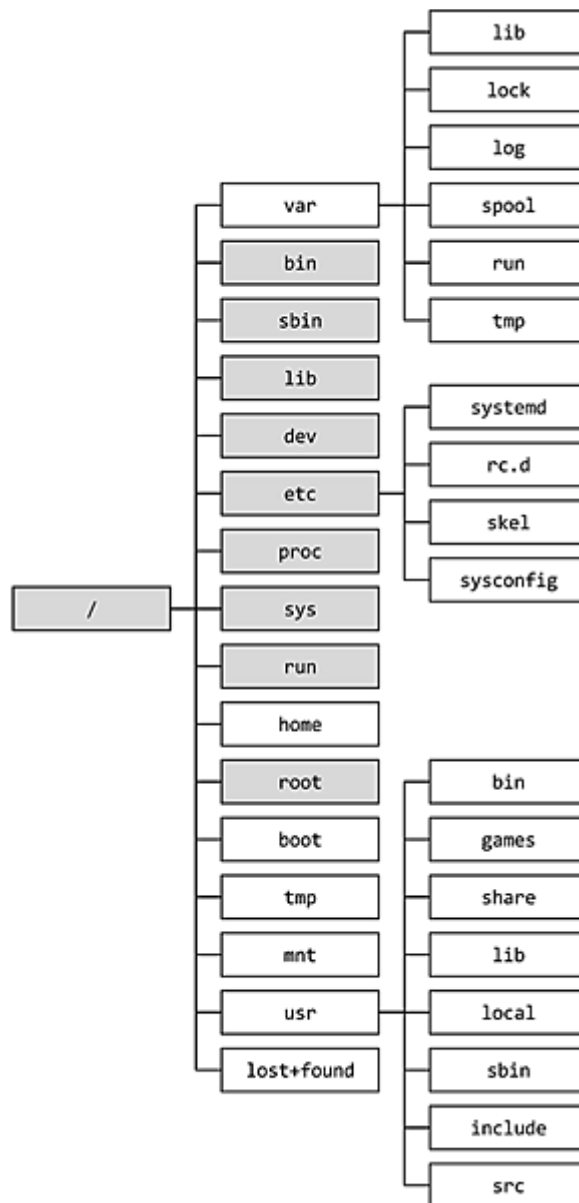
Le plus difficile, lorsqu'on débute sous Linux, est sans doute de connaître l'emplacement des fichiers et l'utilité des répertoires présents dans l'arborescence.

Organisation

Présenté de façon arborescente, le système de fichiers Linux est une hiérarchie de répertoires ayant pour racine unique / (slash).

Lors de l'installation d'une distribution Linux, il est possible de créer, en plus de la partition principale contenant /, des partitions dédiées à certains répertoires de l'arborescence. Cependant, les répertoires indispensables au démarrage du

système doivent être sur la même partition que / et ne peuvent donc pas être installés sur une partition séparée ; ces répertoires essentiels apparaissent en grisé dans le schéma suivant.



Dans la mesure du possible, une analogie sera faite entre l'arborescence Linux et les répertoires classiques du système Microsoft

Windows ; ceci permettra aux utilisateurs connaissant ce système grand public de retrouver leurs marques.

L'arborescence de fichiers Linux respecte, à quelques répertoires près, le FHS (Filesystem Hierarchy Standard) mis en place dans le but d'homogénéiser la structure des systèmes de fichiers Unix. Ce document détaille le nom et le contenu des répertoires ; il est disponible à l'adresse suivante :

<http://www.pathname.com/fhs>

Principaux répertoires

1. /bin, /sbin, /lib

Le répertoire /bin contient les exécutables (binaires) de base nécessaires au fonctionnement du système ; les commandes telles que **date** s'y trouvent.

Toutes les commandes d'administration système essentielles sont, quant à elles, dans /sbin ("super" binaires). On y trouvera, par exemple, les commandes de partitionnement et de gestion des périphériques réseau.

Les binaires compilés pour Linux font appel à des bibliothèques de fonctions, ce qui permet notamment d'alléger la taille des fichiers puisque plusieurs exécutables pourront utiliser la même portion de code contenue dans l'une de ces bibliothèques. Le répertoire /lib (libraries) regroupe les bibliothèques utilisées par les binaires contenus dans /bin et /sbin. Sur les systèmes acceptant des binaires compilés aussi bien en 32 bits qu'en 64 bits et qui nécessitent des bibliothèques déclinées dans ces deux formats, on peut trouver aussi les répertoires /lib32 et /lib64 pour les distinguer.

En comparaison avec un système Windows, ces trois répertoires correspondent aux exécutables contenus dans C:\windows avec leurs bibliothèques (fichiers .dll) se trouvant dans C:\windows\system.32 et dans C:\Windows\SysWOW64.

Ces trois répertoires sont vitaux pour le système et doivent être sur la même partition que /.

2. /boot

Ce répertoire contient le noyau Linux et d'autres fichiers lancés à l'amorçage du système.

Ce répertoire peut faire l'objet d'une partition distincte.

3. /dev

Une des particularités de Linux est de présenter, sous forme de fichiers, les périphériques attachés au système.

Ce répertoire contient des fichiers comme /dev/fd0 et /dev/cdrom qui représentent respectivement le lecteur de disquettes et le lecteur de CD-ROM.

Les clés ou disques USB ont un fichier dont le nom commence par sd dans ce même répertoire. Pour déterminer le nom de ce périphérique après l'avoir

connecté physiquement à la machine, on peut utiliser la commande **dmesg** qui retourne les derniers messages du noyau :

```
[ 2457.930953] usb 1-1: USB disconnect, device number 8
[ 2477.743199] usb 1-1: new high-speed USB device number 9 using ehci-pci
[ 2477.874973] usb 1-1: New USB device found, idVendor=0951, idProduct=168a
[ 2477.874978] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 2477.874980] usb 1-1: Product: DT Micro
[ 2477.874981] usb 1-1: Manufacturer: Kingston
[ 2477.874982] usb 1-1: SerialNumber: 00241D8CE459BC3129070006
[ 2477.882106] usb-storage 1-1:1.0: USB Mass Storage device detected
[ 2477.882270] scsi host9: usb-storage 1-1:1.0
[ 2478.965448] scsi 9:0:0:0: Direct-Access Kingston DT Micro PMAP PQ: 0 ANSI: 0 CCS
[ 2478.968571] sd 9:0:0:0: Attached scsi generic sg2 type 0
[ 2480.238273] sd 9:0:0:0: [sdb] 30712320 512-byte logical blocks: (15.7 GB/14.6 GiB)
[ 2480.242268] sd 9:0:0:0: [sdb] Write Protect is off
[ 2480.242274] sd 9:0:0:0: [sdb] Mode Sense: 23 00 00 00
[ 2480.246337] sd 9:0:0:0: [sdb] No Caching mode page found
[ 2480.246347] sd 9:0:0:0: [sdb] Assuming drive cache: write through
[ 2480.288176] sdb: sdb1
[ 2480.309030] sd 9:0:0:0: [sdb] Attached SCSI removable disk
```

Dans l'exemple précédent, l'avant-dernière ligne avec **sdb: sdb1** nous indique que le disque s'appelle /dev/sdb et possède une partition /dev/sdb1.

En fonction du formatage effectué par le constructeur, cette dernière partition /dev/sdb1 n'est pas obligatoirement présente et il faut travailler directement avec le disque /dev/sdb.

Une autre façon d'arriver à la même conclusion que l'exemple précédent est de lister le contenu du répertoire /dev/disk/by-id qui, comme son nom l'indique, classe les périphériques de type disque par leur identifiant unique :

```
[nicolas]$ ls -ltr /dev/disk/by-id/
total 0
lrwxrwxrwx. 1 root root 10 23 févr. 14:30 lvm-pv-uuid-8YXpYn -> ../../sda2
lrwxrwxrwx. 1 root root 9 23 févr. 14:30 ata-IDE_CDRom_Drive_101 -> ../../sr0
lrwxrwxrwx. 1 root root 10 23 févr. 14:30 dm-uuid-LVM-Sue9iLz06V8 -> ../../dm-1
lrwxrwxrwx. 1 root root 10 23 févr. 14:30 dm-name-fedora-swap -> ../../dm-1
lrwxrwxrwx. 1 root root 10 23 févr. 14:30 dm-uuid-LVM-SueogpfaV2o -> ../../dm-0
lrwxrwxrwx. 1 root root 10 23 févr. 14:30 dm-name-fedora-root -> ../../dm-0
lrwxrwxrwx. 1 root root 9 23 févr. 15:12 usb-Kingston_DT_Micro_00241D06-0:0
-> ../../sdb
lrwxrwxrwx. 1 root root 10 23 févr. 15:12 usb-Kingston_DT_Micro_00241D06-0:0-part1
-> ../../sdb1
```

Dans cet exemple, la commande **ls** affiche tout en bas le dernier disque connecté au système. On y retrouve les liens vers /dev/sdb et /dev/sdb1.

Toujours dans le domaine des fichiers spéciaux, il existe dans ce répertoire des fichiers comme /dev/null ; ce dernier est en quelque sorte une poubelle dans

laquelle on peut rediriger toutes les données que l'on ne veut pas garder. Son utilisation sera détaillée dans la partie sur le Shell Bash.

La spécificité des fichiers présents dans /dev n'est pas en rapport avec leurs noms mais avec leurs numéros majeurs et mineurs. Ces deux numéros indiquent au noyau Linux le type de périphérique attaché au fichier ; ils sont affichés avec la commande **ls -l**.

Les fichiers de ce répertoire servant à accéder aux périphériques ou à mettre en œuvre des fonctionnalités importantes pour le système, /dev doit apparaître sur la même partition que /.

4. /home

Le répertoire /home contient les répertoires personnels des utilisateurs de la machine. Ainsi, pour les comptes utilisateur **nicolas**, **linus** et **richard** existeront les répertoires /home/nicolas, /home/linus et /home/richard.

C'est dans son répertoire personnel qu'un utilisateur stocke ses fichiers de données et que viennent s'inscrire les fichiers de configuration propres à l'utilisateur. Par exemple, le paramétrage de l'éditeur Vi, pour l'utilisateur **nicolas**, se trouve dans le fichier /home/nicolas/.exrc.

Le répertoire /home sous Linux peut être comparé au dossier C:\Documents and Settings ou C:\Users d'un système Windows.

À l'installation du système, l'administrateur dédie habituellement une partition à ce répertoire afin de séparer physiquement les données utilisateur et les données système.

5. /root

Ce répertoire a le même rôle que le précédent, mais il est réservé à l'utilisateur **root** qui est l'administrateur de la machine.

Il ne se trouve pas dans /home pour des raisons de sécurité et parce qu'il ne peut faire l'objet d'une partition distincte comme /home.

6. /tmp

Comme son nom l'indique, le répertoire /tmp est prévu pour accueillir des fichiers temporaires.

Tous les utilisateurs du système peuvent écrire dans ce répertoire mais un nettoyage automatique de ce répertoire est généralement planifié à intervalles

réguliers par l'administrateur. La durée de vie d'un fichier présent dans ce répertoire est estimée à quelques heures, voire quelques jours.

7. /lost+found

Ce répertoire est créé automatiquement sur chaque partition Linux au démarrage du système. Il est utilisé par l'outil de vérification **fsck** (File System Check) pour stocker les fichiers récupérés suite à un incident sur le système ; après une coupure intempestive de l'alimentation, par exemple.

Ce répertoire devrait, en toute logique, rester vide.

8. /mnt

/mnt est un répertoire vide ou contenant une série de répertoires vides prédéfinis. Il est réservé au montage de systèmes de fichiers tiers.

On peut y trouver ou y créer les sous-répertoires /mnt/usbdisk et /mnt/cdrom, prévus pour accéder respectivement aux disques USB et aux CD-ROM.

Les distributions récentes prévoient aussi des points de montage dans /media pour les systèmes de fichiers provenant de médias amovibles (clés USB, CD-ROM, disques externes...), et dans /misc pour les arborescences montées automatiquement par le monteur automatique système.

9. /proc

/proc est un système de fichiers virtuel représentant l'état du système en cours d'exécution.

Il ne prend aucune place sur le disque ; il n'existe qu'en mémoire centrale (RAM) et chaque fichier qu'il contient donne accès à des informations système telles que l'utilisation actuelle de la mémoire, la liste des périphériques détectés sur les différents bus de la machine (ISA, PCI, USB...) ou encore le routage de paquets réseau.

L'analyse des fichiers présents dans /proc est du ressort de l'administrateur système et peut être ignorée dans le cadre de l'utilisation quotidienne du poste de travail.

De plus, pour chaque processus lancé sur le système, il existe un sous-répertoire le caractérisant dans /proc. Des commandes comme **ps** et **top**, abordées plus loin, utilisent ces informations.

Toutes ces données étant présentées dans des fichiers "virtuels", aucune partition particulière n'est réservée à /proc.

10. /sys

À l'instar de /proc, /sys est un système de fichiers virtuel n'occupant pas de place sur le disque. Son but est de représenter sous forme de fichiers les différents périphériques et d'indiquer caractéristiques et états matériels.

11. /usr

C'est le répertoire le plus volumineux créé lors de l'installation du système ; en effet, /usr contient tous les programmes qui ne sont pas dans /bin et /sbin. On y retrouve aussi la documentation et les sources des logiciels.

Ce répertoire correspond approximativement au dossier c:\program files présent sous Windows.

Il est possible de créer une partition spécialement pour ce répertoire lors de l'installation du système.

/usr/bin, /usr/sbin, /usr/lib

Ces trois sous-répertoires contiennent des binaires (et bibliothèques associées) supplémentaires mais non essentiels pour l'utilisation ou l'administration de base du système.

/usr/games

Comme son nom l'indique, les jeux installés sur le système devraient figurer ici.

/usr/include

On trouve ici les définitions des différentes bibliothèques utilisées en programmation. Le contenu de ce répertoire n'est nécessaire que lors de la compilation de programmes.

/usr/local

Ce répertoire est dédié aux applications et documents propres à la machine locale ; les logiciels compilés spécifiquement pour le système se trouvent donc ici.

Ce répertoire est décomposé en sous-répertoires bin, lib...

/usr/share

En plus des fichiers d'internationalisation contenant la traduction des messages affichés par les applications (/usr/share/locale), ce sous-répertoire renferme notamment la documentation (/usr/share/man, /usr/share/info et /usr/share/doc).

Les ressources partagées destinées aux applications graphiques se trouvent dans le sous-répertoire /usr/share/X11.

/usr/src

La plupart des logiciels utilisés étant développés sous licence libre (GPL), il n'est pas rare de disposer des sources de ceux-ci pour les recompiler de manière spécifique.

Si tel est le cas, les fichiers sources de ces programmes sont stockés ici. Par exemple, les sources du noyau Linux se trouveront dans le répertoire /usr/src/linux.

12. /var

Autre répertoire volumineux d'un système de fichiers Linux, /var regroupe tous les fichiers de données "variables" dans le temps : les files d'attente d'imprimante, les boîtes aux lettres électroniques des utilisateurs et les journaux système.

Un journal ou fichier de "log" contient l'historique de l'activité d'un programme. Il existe par exemple un fichier de log traçant l'heure de connexion et de déconnexion des utilisateurs sur la machine.

La taille de ce répertoire dépend fortement des services mis en œuvre sur le système ; ainsi, si la machine fait office de serveur de messagerie, de fichiers et d'imprimante en même temps, ce répertoire sera plus volumineux que sur une machine servant uniquement de poste de travail.

Enfin, une partition spécifique à ce répertoire peut lui être réservée.

/var/lib

Ce sous-répertoire contient généralement les fichiers des systèmes de gestion de bases de données (SGBD) comme MySQL.

/var/log

C'est ici que sont regroupés tous les journaux système et applicatifs. La consultation de fichiers de log permet à l'administrateur de comprendre - et donc de résoudre - un grand nombre de problèmes système et applicatifs.

/var/lock

Les applications serveur (serveur web, serveur ftp, planificateur de tâches) créent chacune un fichier dans ce répertoire pour témoigner de leur exécution. Appelés fichiers "verrous", ils permettent par leur présence d'éviter qu'un de ces programmes soit lancé plusieurs fois.

Ces fichiers sont, en toute logique, effacés lors de l'arrêt des programmes correspondants.

/var/run

En plus des fichiers verrous (/var/lock), les services lancés sur la machine créent généralement un fichier qui porte leur nom et contient l'identifiant du processus (PID) correspondant dans /var/run.

Ces fichiers permettent de retrouver plus facilement les identifiants des processus par les scripts de contrôle.

/var/spool

Ce sous-répertoire regroupe les files d'attente (spool) de différents services : impression (/var/spool/lpd et /var/spool/cups), messagerie électronique (/var/spool/mail), travaux planifiés (/var/spool/cron)...

/var/tmp

À l'instar du répertoire /tmp, tous les utilisateurs du système peuvent stocker des fichiers temporaires dans /var/tmp.

La différence entre ces deux emplacements est la durée de vie généralement attribuée par l'administrateur à ces fichiers ; de quelques jours à plusieurs semaines pour /var/tmp.

13. /run

Tout comme /proc et /sys, le répertoire /run, apparu récemment, est un système de fichiers virtuel n'occupant pas de place sur le disque.

Il a pour objectif de regrouper les fichiers applicatifs résidant en mémoire comme les fichiers verrous ou contenant le PID des services lancés, ou encore toute autre donnée d'exécution ne pouvant prendre le risque d'être effacée dans /tmp. Sa mise en place facilite en plus la gestion du démarrage avec **systemd** et des périphériques avec **udev**.

14. /etc

Le répertoire /etc ("et cætera") contient les fichiers ne trouvant pas leur place dans les autres répertoires, c'est-à-dire tous les fichiers de configuration et

scripts de démarrage du système.

Il est décomposé en sous-répertoires ; on notera l'existence de :

- /etc/rc.d qui accueille les scripts de démarrage et de contrôle des services.
- /etc/skel (répertoire "squelette") qui contient les fichiers qui seront copiés dans le répertoire de l'utilisateur lors de la création de son compte Linux.
- /etc/sysconfig qui contient principalement la configuration des périphériques.

Ce répertoire correspond, en quelque sorte, à la base de registres du système Windows.

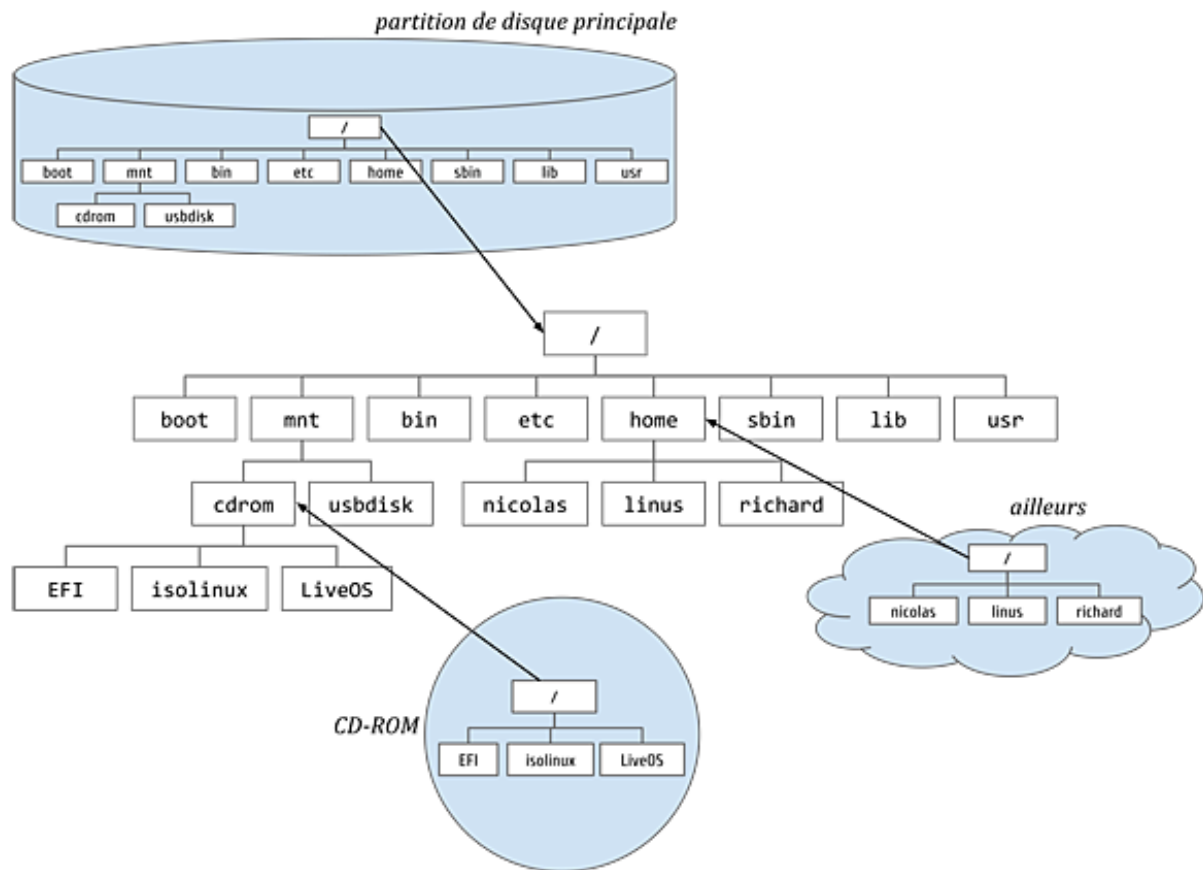
Primordial au système d'exploitation, ce répertoire se trouve obligatoirement sur la partition principale

Montages et accès aux médias amovibles

Considérée dans son ensemble jusqu'à présent, l'arborescence Linux peut être composée de plusieurs systèmes de fichiers stockés sur des supports différents.

En fait, chaque partition ou média amovible abrite un système de fichiers "physique". Une fois rassemblés, ils forment l'arborescence - ou système de fichiers - Linux que nous avons décrite précédemment. Pour cela, une partition sur le disque dur de la machine est désignée comme principale et accueille le système de fichiers racine (celui contenant /) auquel viendront se greffer les autres systèmes de fichiers.

Par exemple :



1. Commande mount

Pour accéder au contenu d'un système de fichiers physique, il faut "accrocher" celui-ci à l'arborescence Linux. Cette opération s'appelle "montage" et s'effectue à l'aide de la commande **mount**. Sa syntaxe générale est la suivante :

mount [-t <type>] [-o options[,...]] <périphérique> <répertoire>

Une page de manuel détaille cette commande (commande : `man mount`) ; seuls les montages de systèmes de fichiers présents sur CD-ROM ou disques USB seront exposés ici.

Comme indiqué précédemment, les fichiers associés au lecteur de CD-ROM et à un disque USB sont `/dev/cdrom` et `/dev/sdb1` dans notre exemple de début.

Pour monter les systèmes de fichiers présents sur les médias insérés dans ces périphériques, il faut saisir les commandes **mount** suivantes :

mount /dev/cdrom

mount /dev/sdb1

La commande **mount** invoquée seule nous indique le répertoire utilisé dans l'arborescence Linux pour tous les montages de système de fichiers ; les répertoires de montage associés par défaut aux lecteurs de CD-ROM et aux disques USB sont définis dans le fichier /etc/fstab ; pour plus d'informations concernant ce fichier : **man 5 fstab**.

Suivant les droits accordés par l'administrateur dans le fichier /etc/fstab, les utilisateurs peuvent ne pas être autorisés à effectuer de montage. Il faudra alors se connecter en tant que **root**.

Par exemple, pour visionner le contenu d'un CD-ROM :

```
[root]# mount
/dev/hda3 on / type ext3 (rw)
none on /proc type proc (rw)
[root]# ls /mnt/floppy/
[root]# mount /dev/fd0
[root]# mount
/dev/hda3 on / type ext3 (rw)
none on /proc type proc (rw)
/dev/fd0 on /mnt/floppy type vfat (rw,nosuid,nodev)
[root]# ls /mnt/floppy/
command.com ega2.cpi io.sys      keybrd2.sys mode.com
config.sys  ega3.cpi  keyb.com   keybrd3.sys msdos.sys
display.sys ega.cpi   keyboard.sys keybrd4.sys
```

Dans cet exemple, la deuxième commande **mount**, invoquée seule, indique que le système de fichiers présent sur le CD-ROM (/dev/cdrom) est monté sous le répertoire /mnt/cdrom. Les appels de la commande **ls -l** affichent quant à eux le contenu du répertoire /mnt/cdrom avant et après le montage effectué avec **mount /dev/cdrom**.

Sur les distributions récentes, un système d'automontage est mis en œuvre. Ceci permet à l'utilisateur d'accéder aux systèmes de fichiers présents sur les disques USB et CD-ROM sans invoquer la commande **mount**, uniquement en accédant aux répertoires associés aux périphériques dans le fichier /etc/fstab.

2. Commande umount

De la même manière, lorsqu'on ne désire plus accéder à un système de fichiers sous Linux, il faut le "démonter" avant de pouvoir extraire le média de son lecteur. Dans le cas d'un CD-ROM, le tiroir du lecteur est bloqué tant que cette opération n'est pas réalisée.

Même s'il est possible d'extraire mécaniquement une clé USB de son port, il est

tout de même obligatoire de démonter le système de fichiers monté préalablement, sous peine de perdre les données stockées sur le média.

La commande **umount**, suivie du nom du périphérique ou du répertoire de montage, permet le démontage du système de fichiers ; par exemple, pour le CD-ROM :

```
# umount /mnt/cdrom
```

ou :

```
# umount /dev/cdrom
```

Ceci dit, il n'est pas possible de démonter un système de fichiers utilisé ; il faudra prendre soin de ne pas se trouver dans le répertoire de montage lors de l'appel à **umount**.

3. Boîte à outils mtools

Installé par défaut avec la plupart des distributions Linux, le paquetage logiciel mtools est un ensemble de commandes permettant d'accéder aux médias de la même manière que sur la ligne de commande DOS.

Ces commandes portent le même nom que les commandes DOS préfixées de la lettre "m".

Par exemple, pour lire une disquette DOS et copier un de ses fichiers dans le répertoire /tmp sans pour autant effectuer de montage, on utilisera les commandes **mdir** et **mcopy** comme suit :

```
[root]# mdir a:
Volume in drive A has no label
Volume Serial Number is 2A87-6CE1
Directory for A:/

EGA2  CPI   58870 06-08-2000 17:00
EGA3  CPI   58753 06-08-2000 17:00
EGA   CPI   58870 06-08-2000 17:00
KEYB  COM    21607 06-08-2000 17:00
KEYBOARD SYS  34566 06-08-2000 17:00
KEYBRD2 SYS  31942 06-08-2000 17:00
KEYBRD3 SYS  31633 06-08-2000 17:00
KEYBRD4 SYS  13014 06-08-2000 17:00
MODE  COM    29239 06-08-2000 17:00
COMMAND COM   93040 06-08-2000 17:00
DISPLAY SYS  17175 06-08-2000 17:00
CONFIG SYS    33 03-01-2004 19:16
    12 files           448 742 bytes
                   888 832 bytes free
```

```
# mcopy a:\command.com /tmp
# ls /tmp
command.com
```

Les lettres des lecteurs émulés sont **A** et **B** pour les disquettes, **J** pour un lecteur Jaz, et **Z** pour un disque Zip.

Attention, les commandes mtools ne peuvent accéder aux médias déjà montés avec la commande **mount**.

Exercices

Exercice 1

Insérez un CD-ROM ou un DVD-ROM, clé USB et consultez les fichiers présents dessus.

Solution

Si vous êtes dans un environnement graphique, il est très probable que le système monte automatiquement le média détecté ou vous propose de le faire. Dans ce cas, utilisez la commande mount seule pour déterminer le point de montage automatique et explorez les fichiers qui s'y trouvent.

Si ce n'est pas le cas ou si vous êtes en environnement texte, utilisez la commande mount /dev/cdrom /mnt et consultez les fichiers présents sous ce point de montage.

Pour cette dernière opération, il faudra probablement se connecter en tant qu'administrateur pour avoir les autorisations nécessaires.

Exercice 2

Démontez et éjectez le média de l'exercice précédent.

Solution

Veillez préalablement à ne pas avoir de shell ou d'explorateur de fichiers graphique actuellement positionné sur un des répertoires de l'arborescence du média à démonter.

En environnement graphique, une entrée du menu contextuel avec le clic droit sur l'icône du média monté automatiquement permet d'éjecter et de démonter le périphérique en une seule opération.

En environnement texte, l'emploi de la commande umount /dev/cdrom démontera le média optique.