

Chapitre 4

L'arborescence Linux

1. Introduction

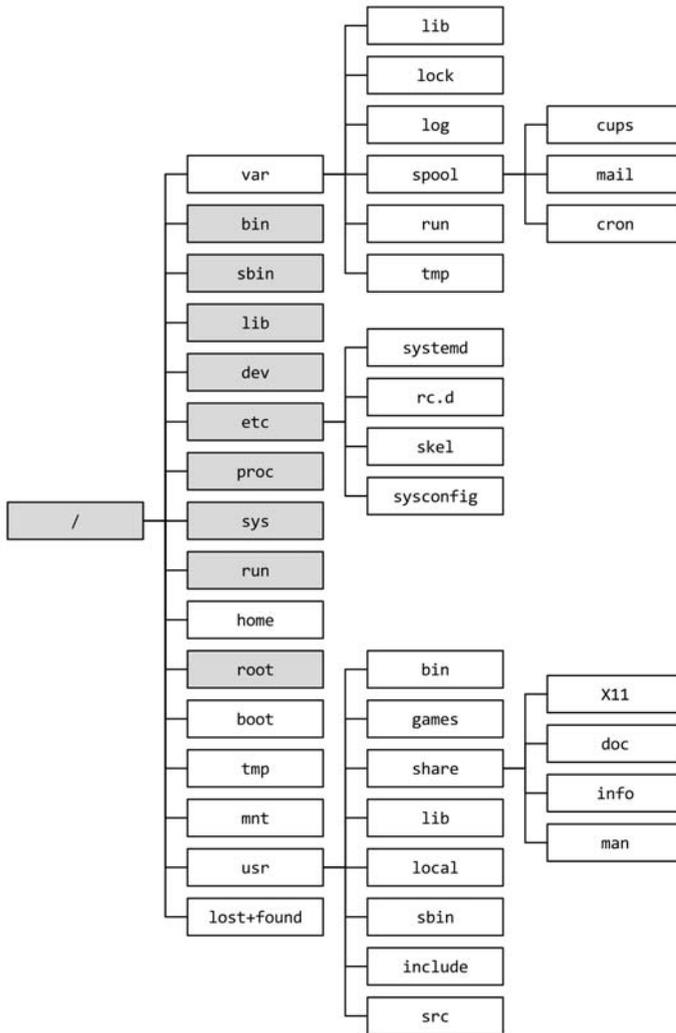
Le plus difficile, lorsqu'on débute sous Linux, est sans doute de connaître l'emplacement des fichiers et l'utilité des répertoires présents dans l'arborescence.

Ce chapitre propose donc un tour d'horizon du système de fichiers Linux.

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.



Remarque

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. Maintenu par *The Linux Foundation*, ce document détaille le nom et le contenu des répertoires ; il est disponible à l'adresse suivante : <https://wiki.linuxfoundation.org/lsb/fhs>

2. Principaux répertoires

2.1 /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.

■ Remarque

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\system32` et dans `c:\windows\SysWOW64`.

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

2.2 /boot

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

Pour des raisons non détaillées dans cet ouvrage, ce répertoire peut faire l'objet d'une partition distincte.

2.3 /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/tty0` et `/dev/cdrom` qui représentent respectivement le premier terminal texte 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: 00241D8CE459EC3129070006
[ 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`.

Remarque

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`.

Remarque

La syntaxe et les options de la commande `ls` sont détaillées dans le chapitre *Manipulation de fichiers*.

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 le chapitre 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 `/`.

2.4 /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*.

■ Remarque

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.

2.5 /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*.

2.6 /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.

2.7 /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.

2.8 /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.

■ Remarque

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.

2.9 /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 (SCSI, PCI, USB...) ou encore le routage de paquets réseau.