

---

## Prérequis

---

- Savoir utiliser le shell et ses commandes.
- Disposer d'un éditeur de texte.
- Avoir le mot de passe root.
- Disposer d'un support de stockage physique ou virtuel modifiable.
- Connaître les bases du fonctionnement matériel d'un ordinateur.

---

## Objectifs

---

À la fin de ce chapitre, vous serez en mesure :

- De reconnaître les disques et les partitions.
- De manipuler les paramètres des périphériques.
- De choisir parmi différents systèmes de fichiers.
- De créer des partitions.
- De créer un système de fichiers.
- De créer des points de montage et modifier fstab.
- De contrôler, modifier et réparer un système de fichiers.
- De créer des espaces de swap.
- De mettre en place des quotas.
- De gérer les permissions et les propriétaires des fichiers.
- De connaître le principe des hard links.

## A. Représentation des disques

Note préalable : les unités de mesure de stockage utilisées dans ce chapitre et dans l'ensemble de ce livre utilisent la représentation de l'usage traditionnel en kilo-octets, selon la règle 1 ko = 1024 octets ( $2^{10}$ ), sauf indication contraire. Cette représentation se nomme théoriquement Kio (kibiocet).

### 1. Nomenclature

Ceci est un petit rappel des points déjà rencontrés dans le chapitre Présentation de Linux. Suivant le type de contrôleur et d'interface sur lesquels les disques sont connectés, Linux donne des noms différents aux fichiers spéciaux des périphériques disques.

Chaque disque est représenté par un fichier spécial de type bloc. Chaque partition aussi.

#### a. IDE

Cette section est conservée pour des raisons historiques, la norme SATA ayant remplacé la norme IDE sur la quasi-totalité des ordinateurs de bureau et portables depuis plus de dix ans. Les disques reliés à des contrôleurs IDE (appelés aussi PATA (*Parallel ATA*) ou ATAPI) se nomment hdX :

- hda : IDE0, Master
- hdb : IDE0, Slave
- hdc : IDE1, Master
- hdd : IDE1, Slave
- etc.

Contrairement aux idées reçues, il n'y a pas de limites au nombre de contrôleurs IDE, sauf le nombre de ports d'extension de la machine (slots PCI). De nombreuses cartes additionnelles et convertisseurs existent permettant de lire d'anciens disques IDE. Au-delà de quatre disques ou lecteurs, les fichiers se nomment hde, hdf, hdg, etc.

Les lecteurs CD-Rom, DVD et graveurs de type IDE/ATAPI sont vus comme des disques IDE et respectent cette nomenclature.

Les noyaux Linux utilisent maintenant par défaut une API appelée libata pour accéder à l'ensemble des disques IDE, SCSI, USB, Firewire, etc. La nomenclature reprend celle des disques SCSI, abordée au point suivant.

#### b. SCSI, SATA, USB, FIREWIRE, etc.

Les disques reliés à des contrôleurs SCSI, SCA, SAS, FibreChannel, USB, Firewire, thunderbolt (et probablement d'autres interfaces exotiques) se nomment sdX. L'énumération des disques reprend l'ordre de détection des cartes SCSI et des adaptateurs (hosts) associés, puis l'ajout et la suppression manuelle des autres via hotplug ou udev.

- sda : premier disque SCSI
- sdb : deuxième disque SCSI
- sdc : troisième disque SCSI
- etc.

La norme SCSI fait une différence entre les divers supports. Aussi les lecteurs CD-Rom, DVD, HD-DVD, Blu-ray et les graveurs associés n'ont pas le même nom. Les lecteurs et graveurs sont en srX (sr0, sr1, etc.). Vous pouvez aussi trouver scd0, scd1, etc. mais ce sont généralement des liens symboliques vers sr0, sr1, etc.

Au-delà de sdz, l'énumération redémarre à sdaa, sdab, etc.

La commande **lsscsi** permet d'énumérer les périphériques SCSI. Notez que les disques sont bien en sdX, tandis que le lecteur dvd est en srX.

```
$ lsscsi
[4:0:0:0]   disk      ATA          ST380011A    8.01        /dev/sda
[5:0:0:0]   cd/dvd    LITE-ON     COMBO SOHC-4836V S9C1        /dev/sr0
[31:0:0:0]  disk      USB2.0     Mobile Disk  1.00        /dev/sdb
```

## 2. Cas spéciaux

### a. Contrôleurs spécifiques

Certains contrôleurs ne suivent pas cette nomenclature. C'est par exemple le cas de certains contrôleurs RAID matériels. C'est du cas par cas. Un contrôleur Smart Array sur un serveur HP, utilisant le pilote cciss, place ses fichiers de périphériques dans /dev/cciss sous les noms cXdYpZ, où X est le slot, Y le disque et Z la partition. Les nouveaux contrôleurs utilisent le pilote hpsa, exploitant la couche SCSI du noyau et donc un nommage standard des périphériques.

### b. Virtualisation

La représentation des disques des systèmes invités (*guests*) virtualisés dépend du type de contrôleur simulé. La plupart sont de type IDE ou SCSI, et dans les deux cas bien souvent avec la libata ils sont vus comme du SCSI. Cependant certains systèmes comme KVM ou XEN (ainsi que les environnements cloud les utilisant, comme AWS) proposant de la paravirtualisation offrent un contrôleur spécifique présentant les disques sous le nom vdX (virtual disk x), ou xvdx :

- vda : premier disque virtualisé, ou xvda,
- vdb : deuxième disque virtualisé, ou xvdb,
- etc.

### c. SAN, iSCSI, multipathing

Les disques raccordés via un SAN (*Storage Area Network*, généralement en fibre optique) ou par iSCSI sont vus comme des disques SCSI et conservent cette nomenclature. Cependant les systèmes de gestion des chemins multiples (*multipathing*) se plaçant par-dessus fournissent d'autres noms. Powerpath nommera les disques emcpowerx (emcpowera, emcpowerb, etc.) tandis que le système par défaut de Linux appelé multipath les nommera mpathx (mpath0, mpath1, etc.) ou tout autre nom choisi par l'administrateur.

## B. Manipulations de bas niveau

### 1. Informations

La commande **hdparm** permet d'effectuer un grand nombre de manipulations directement sur les périphériques disques gérés par la bibliothèque libata, c'est-à-dire tous les disques SATA, ATA (IDE) et SAS. La commande **sdparm** peut faire à peu près la même chose pour les disques SCSI. Notez que bien que les noms de périphériques de la libata soient identiques à ceux du SCSI, il est fort probable que de nombreuses options de configuration de **hdparm** ne fonctionnent pas sur des disques SCSI, la réciproque étant vraie pour **sdparm** avec les disques SATA ou IDE. La suite se base sur **hdparm**.

Pour obtenir des informations complètes sur un disque, utilisez les paramètres **-i** ou **-I**. Le premier récupère les informations depuis le noyau et obtenues au moment du boot, le second interroge directement le disque. Préférez le **-I** qui donne des informations très détaillées.

```
# hdparm -I /dev/sda

/dev/sda:

ATA device, with non-removable media
  Model Number:          VBOX HARDDISK
  Serial Number:         VB91a2e953-933cdc65
  Firmware Revision:    1.0
Standards:
  Used: ATA/ATAPI-6 published, ANSI INCITS 361-2002
  Supported: 6 5 4
Configuration:
  Logical          max          current
  cylinders        16383       16383
  heads           16           16
  sectors/track   63           63
  --
  CHS current addressable sectors:    16514064
  LBA  user addressable sectors:     63152320
  LBA48 user addressable sectors:     63152320
  Logical/Physical Sector size:       512 bytes
  device size with M = 1024*1024:     30836 MBytes
  device size with M = 1000*1000:     32333 MBytes (32 GB)
  cache/buffer size = 256 KBytes (type=DualPortCache)
Capabilities:
  LBA, IORDY(cannot be disabled)
  Queue depth: 32
  Standby timer values: spec'd by Vendor, no device specific minimum
  R/W multiple sector transfer: Max = 128          Current = 128
  DMA: mdma0 mdma1 mdma2 udma0 udma1 udma2 udma3 udma4 udma5 *udma6
      Cycle time: min=120ns recommended=120ns
  PIO: pio0 pio1 pio2 pio3 pio4
      Cycle time: no flow control=120ns IORDY flow control=120ns
Commands/features:
```

```

Enabled      Supported:
*           Power Management feature set
*           Write cache
*           Look-ahead
*           48-bit Address feature set
*           Mandatory FLUSH_CACHE
*           FLUSH_CACHE_EXT
*           Gen2 signaling speed (3.0Gb/s)
*           Native Command Queuing (NCQ)
Checksum: correct

```

## 2. Modification des valeurs

Plusieurs paramètres des disques peuvent être modifiés. Attention cependant ! Certaines options de `hdparm` peuvent se révéler être dangereuses tant pour les données contenues sur le disque que pour le disque lui-même. La plupart des paramètres sont en lecture et écriture. Si aucune valeur n'est précisée `hdparm` affiche l'état du disque (ou du bus) pour cette commande. Voici quelques exemples d'options intéressantes.

- `-c` : largeur du bus de transfert EIDE sur 16 ou 32 bits. 0=16, 1=32, 3=32 compatible.
- `-d` : utilisation du DMA. 0=pas de DMA, 1=DMA activé.
- `-x` : modifie le mode DMA (mdma0 mdma1 mdma2 udma0 udma1 udma2 udma3 udma4 udma5). Vous pouvez utiliser l'un des modes précédents ou des valeurs numériques : 32+n pour les modes mdma (n variant de 0 à 2) et 64+n pour les modes udma.
- `-C` : statut de l'économie d'énergie sur le disque (unknown, active/idle, standby, sleeping). L'état peut être modifié avec `-S`, `-y`, `-Y` et `-Z`.
- `-g` : affiche la géométrie du disque.
- `-m` : indique ou modifie l'état du Automatic Acoustic Management (AAM). 0=off, 128=quiet et 254=fast. Tous les disques ne le supportent pas.
- `-r` : passe le disque en lecture seule.
- `-T` : bench de lecture du cache disque, idéal pour tester les performances de transfert entre Linux et le cache du disque. Il faut relancer la commande deux ou trois fois.
- `-t` : bench de lecture du disque, hors cache. Mêmes remarques que l'option précédente.

Ainsi la commande suivante passe le bus de transfert en 32 bits, active le mode DMA en mode Ultra DMA 5 pour le disque `sda` :

```
# hdparm -c1 -d3 -X udma5 /dev/sda
```

Voici quelques autres exemples :

```

# hdparm -c /dev/sda
/dev/sda:
  IO_support      = 0 (default 16-bit)
# hdparm -C /dev/sda

```

```
/dev/sda:
drive state is: active/idle

# hdparm -g /dev/sda

/dev/sda:
geometry          = 3931/255/63, sectors = 63152320, start = 0

# hdparm -T /dev/sda

/dev/sda:
Timing cached reads:   23868 MB in  2.00 seconds = 11950.45 MB/sec
# hdparm -t /dev/sda

/dev/sda:
Timing buffered disk reads: 308 MB in  3.02 seconds = 101.87 MB/sec
```

## C. Choisir un système de fichiers

### 1. Principe

#### a. Définition

L'action de « formater » un disque, une clé ou tout support de données consiste uniquement à créer sur un support de mémoire secondaire (volume de stockage) l'organisation logique permettant d'y placer des données. Le mot « formatage » sous Linux est utilisé pour décrire la création d'un système de fichiers. On parle donc de système de fichiers qui est à la fois l'organisation logique des supports au niveau le plus bas comme au niveau de l'utilisateur.

Les informations ne sont pas écrites n'importe comment sur les disques. Une organisation est nécessaire pour y placer tant les informations sur les fichiers qui y sont stockés que les données. Ce sont le système de fichiers et les pilotes associés qui définissent cette organisation. Si les principes de base sont souvent les mêmes entre les divers systèmes présents sous Linux, les implémentations et les organisations logiques des données sur le disque varient fortement. Aussi il n'existe pas un type de système de fichiers, mais plusieurs, au choix de l'utilisateur ou de l'administrateur système.

Tous les systèmes de fichiers Linux doivent respecter les normes POSIX. Comme POSIX définit un ensemble de règles de base, un système de fichiers peut aller au-delà de cette norme en proposant des extensions. La plupart de celles-ci concernent des éléments de sécurité, comme les ACL ou selinux.

Le principe de base d'un système de fichiers est d'associer un nom de fichier à son contenu et d'y permettre l'accès : création, modification, suppression, déplacement, ouverture, lecture, écriture, fermeture. Suivant ce principe, le système de fichiers doit gérer ce qui en découle : mécanismes de protection des accès (les permissions, les propriétaires), les accès concurrents, etc.

---

## Prérequis

---

Les connaissances nécessaires à la certification LPIC-1 :

- Notions de base sur les périphériques, partitions et systèmes de fichiers.
- Édition de fichiers.
- Commandes de gestion de répertoires et de fichiers.

---

## Objectifs

---

À la fin de ce chapitre, vous serez en mesure de :

- Déterminer les caractéristiques des différents types de systèmes de fichiers Linux.
- Créer et gérer les systèmes de fichiers Linux.
- Configurer et gérer l'espace de swap.
- Administrer le système de fichiers global Linux et superviser les périphériques SMART.
- Configurer et administrer l'automontage de systèmes de fichiers de périphériques ou via le réseau.
- Créer des systèmes de fichiers pour les CD-ROM et autres périphériques.
- Connaître les caractéristiques des systèmes de fichiers chiffrés.

## A. Système de fichiers et périphériques

Ce sujet est divisé en trois parties de poids différents.

### 1. Administration du système de fichiers Linux

<b>Poids</b>	4
<b>Objectifs</b>	Configurer et gérer le système de fichiers standard Linux, en combinant le montage de plusieurs systèmes de fichiers de types différents.

#### a. Compétences principales

- Configuration par le fichier `fstab`.
- Outils de gestion de l'espace de swap.
- Identification et montage des systèmes de fichiers par leur UUID.
- Compréhension des unités de montage de `systemd`.

#### b. Éléments mis en œuvre

- `/etc/fstab`
- `/etc/mtab`
- `/proc/mounts`
- `mount`
- `umount`
- `blkid`
- `sync`
- `swapon`
- `swapoff`

### 2. Maintenance du système de fichiers Linux

<b>Poids</b>	3
<b>Objectifs</b>	Administrer le système de fichiers Linux avec les outils système pour gérer les types de systèmes de fichiers standards et superviser les périphériques SMART.

#### a. Compétences principales

- Outils de gestion des systèmes de fichiers de type ext2, ext3 et ext4.
- Outils pour la gestion de base des systèmes de fichiers de type Btrfs, y compris les sous-volumes et les snapshots.
- Outils de gestion des systèmes de fichiers de type XFS.
- Connaissance des principes généraux des systèmes de fichiers de type ZFS.



**b. Éléments mis en œuvre**

- mkfs (mkfs.\*)
- mkswap
- fsck (fsck.\*)
- tune2fs, dumpe2fs et debugfs
- btrfs, btrfs-convert
- xfs\_info, xfs\_check, xfs\_repair, xfsdump et xfsrestore
- smartd, smartctl

**3. Création et configuration de systèmes de fichiers optionnels**

<b>Poids</b>	2
<b>Objectifs</b>	Configuration de l'automontage de systèmes de fichiers de périphériques ou via le réseau, par <code>autofs</code> . Création de systèmes de fichiers pour les CD-ROM et autres périphériques. Notions de système de fichiers chiffré.

**a. Compétences principales**

- Fichiers de configuration d'`autofs`.
- Unités `automount`.
- Outils UDF et ISO 9660.
- Autres systèmes de fichiers pour CD-ROM (HFS).
- Extensions pour les systèmes de fichiers CD-ROM (Joliet, Rock Ridge, El Torito).
- Notions de base du chiffrement de données (`dm-crypt`, LUKS).

**b. Éléments mis en œuvre**

- `/etc/auto.master`
- `/etc/auto.[dir]`
- `mkisofs`
- `cryptsetup`

## B. Administration du système de fichiers Linux

Pour stocker des fichiers et les mettre à disposition des applications, le système Linux utilise une arborescence globale, organisée en répertoires qui peuvent contenir d'autres répertoires et/ou des fichiers.

Les applications peuvent accéder à l'ensemble des fichiers du système de fichiers Linux, sous réserve du contrôle d'accès, sans avoir à connaître l'organisation physique du stockage de ces fichiers (sur un ou plusieurs disques durs, une ou plusieurs partitions de disques, des volumes logiques, sur la machine locale ou une machine distante, etc.).

L'administrateur configure l'organisation de l'arborescence globale du système de fichiers Linux, en utilisant un ou plusieurs systèmes de fichiers pour constituer cette arborescence.

Certains espaces de stockage sont en permanence accessibles à travers l'arborescence, d'autres sont amovibles et peuvent être rendus accessibles dynamiquement, par une commande du système ou de façon automatique (automontage).

Il est possible d'inclure dans l'arborescence globale du système de fichiers Linux des espaces de stockage se trouvant sur d'autres systèmes.

D'autre part, une partie de l'espace physique de stockage des informations doit être réservée pour les opérations de swap de la mémoire vive.

Le sujet 203.1 a pour objectif de s'assurer que le candidat a la maîtrise des opérations de configuration et de gestion du système de fichiers Linux, en combinant des systèmes de fichiers de types différents, permanents ou amovibles, éventuellement chiffrés, ainsi que de la gestion du swap.

### 1. Composants du système de fichiers Linux

L'arborescence du système de fichiers Linux a un répertoire de départ, le répertoire **racine** (*root*), noté */*. Elle est constituée d'un ou plusieurs systèmes de fichiers autonomes, chacun monté sur un des répertoires de l'arborescence.

Un système de fichiers permet de structurer un espace de stockage, sous forme de fichiers et de répertoires. Cette organisation interne à l'espace de stockage est rendue accessible aux applications, sous forme de répertoires et de fichiers, par l'opération de **montage** sur un répertoire de l'arborescence globale.

Un système de fichiers non monté est vu comme un ensemble d'octets non structuré, accessible comme un tout, sans qu'il soit possible d'utiliser son organisation en répertoires et fichiers.

Les systèmes de fichiers peuvent être montés au démarrage du système sur l'arborescence globale du système de fichiers, ou pendant son fonctionnement, manuellement ou automatiquement.

Il existe différents types de systèmes de fichiers, qui peuvent être combinés entre eux au sein de l'arborescence globale du système de fichiers Linux.

### 2. Systèmes de fichiers physiques

Ces systèmes de fichiers sont associés à de l'espace de stockage physique. Ils sont organisés en répertoires et fichiers, qui sont stockés sur divers supports physiques : partitions de disques durs, volumes logiques, périphériques amovibles (CD-ROM, clé USB, etc.), espaces disques gérés par des systèmes distants (NFS, CIFS, SAN, NAS, etc.).

☞ Il existe de nombreux types de systèmes de fichiers physiques supportés par Linux, en particulier *ext2*, *ext3*, *ext4*, *XFS*, *Btrfs* et *ZFS*, qui seront décrits dans le cadre du sujet 203.2.

### 3. Systèmes de fichiers virtuels

Certains types de systèmes de fichiers sont dits **virtuels**, car ils ne sont pas associés à un espace de stockage, mais sont directement gérés en mémoire vive. Ils doivent être montés pour donner accès à leurs répertoires et fichiers, mais leur durée de vie est limitée à leur période de montage : quand ils sont démontés (explicitement ou suite à l'arrêt du système), leur contenu est perdu.

Le type de système de fichiers virtuel *tmpfs* est conçu pour permettre de gérer en mémoire vive les fichiers temporaires. Il est utilisé par défaut par certaines distributions pour le contenu du répertoire */tmp*. Ce mécanisme permet d'optimiser les temps d'accès pour ces fichiers qui n'ont pas vocation à être stockés de façon permanente.

Linux propose deux systèmes de fichiers virtuels standards, **proc** et **sys**, dont le but est de faciliter la communication avec le noyau. Ce dernier expose dynamiquement, sous forme de fichiers, des éléments de sa configuration et des objets qu'il gère (processus, fichiers, etc.). Ces deux systèmes de fichiers sont montés par défaut sur les répertoires */proc* et */sys* de l'arborescence globale du système de fichiers.

#### a. Le système de fichiers virtuel **proc**

Ce système de fichiers virtuel, de type **proc**, est monté par défaut sous le répertoire */proc*. Il est géré dynamiquement par le noyau pour permettre de suivre l'état des processus actifs. Chaque processus est associé à un répertoire dont le nom est l'identifiant du processus. Quand le processus se termine, le répertoire est supprimé. Ce mécanisme permet de suivre les différents attributs des processus du système.

Ce système de fichiers virtuel donne également accès à de nombreuses tables de contrôle du noyau : partitions, montages, utilisation de la mémoire, etc. Il fournit également des informations concernant les caractéristiques du matériel pris en charge par le noyau.

#### Exemples

On lance l'éditeur Vim depuis une connexion SSH et on visualise les caractéristiques du processus ainsi créé depuis une autre connexion, dans le répertoire */proc* :

```
vi /etc/hosts
```

Depuis un autre terminal :

```
ps -ef | grep vim
```

```
pba      2348  2242  1 18:26 pts/0    00:00:00 vim /etc/hosts
```

On cherche le répertoire correspondant au processus de PID 2348, dans */proc* :

```
ls -ld /proc/2348
```

```
dr-xr-xr-x. 9 pba pba 0 25 févr. 18:26 /proc/2348
```

*Contenu du répertoire :*

```
ls /proc/2348
```

```
attr          exe          mounts      projid_map   status
autogroup    fd          mountstats  root         syscall
auxv         fdinfo      net         sched        task
cgroup       gid_map     ns          schedstat    timers
clear_refs   io          numa_maps   sessionid    timerslack_ns
cmdline      limits     oom_adj     setgroups    uid_map
comm         loginuid    oom_score   smaps        wchan
coredump_filter map_files  oom_score_adj smaps_rollup
cpuset       maps        pagemap     stack
cwd          mem         patch_state stat
environ      mountinfo   personality  statm
```

*Tous ces fichiers contiennent des informations sur le processus exécutant Vim. Par exemple, on peut lire la ligne de commande exécutée par le processus :*

```
cd /proc/2348
```

```
cat cmdline
```

```
vim/etc/hosts
```

*On peut connaître le répertoire courant du processus :*

```
ls -ld cwd
```

```
lrwxrwxrwx. 1 pba pba 0 25 févr. 18:29 cwd -> /home/pba
```

*cwd est un lien symbolique vers le répertoire courant du processus.*

*On sort du répertoire associé au processus et on quitte Vim dans l'autre terminal :*

```
cd ..
```

```
ls 2348
```

```
ls: impossible d'accéder à '2348': Aucun fichier ou dossier de ce type
```

*Dès que le processus est terminé, le noyau supprime le répertoire correspondant dans le système de fichiers virtuel proc.*

*Dans le système de fichiers virtuel proc, on dispose de fichiers et de répertoires concernant différents éléments matériels et logiciels gérés par le noyau :*

*Pour connaître la version du noyau :*

```
cat version
```

```
Linux version 4.18.0-147.el8.x86_64 (mockbuild@kbuilder.bsys.centos.org)
(gcc version 8.3.1 20190507 (Red Hat 8.3.1-4) (GCC)) #1 SMP Wed Dec 4 21:51:45
UTC 2019
```

Pour les informations concernant le (ou les) processeur(s) :

```
cat cpufreq
```

```
processor : 0
vendor_id : GenuineIntel
cpu family : 6
model : 142
model name : Intel(R) Core(TM) i3-7020U CPU @ 2.30 GHz
[...]
```

## b. Le système de fichiers virtuel sys

Ce système de fichiers virtuel, de type **sysfs**, est monté par défaut sous le répertoire `/sys`. Il fournit des informations concernant les caractéristiques des différents périphériques. Certains des pseudo-fichiers sont accessibles en écriture (sous contrainte des droits d'accès) et permettent de modifier dynamiquement des paramètres.

### Exemple

Sur une distribution Debian :

```
ls /sys
```

```
block class FscSpecials fs kernel power
bus dev firmware hypervisor module
```

Ces répertoires contiennent différents répertoires et fichiers, permettant de connaître les différents éléments matériels et logiciels gérés par le noyau.

```
ls /sys/dev/block
```

```
11:0 254:0 254:1 254:2 254:3 254:4 8:0 8:1 8:2 8:5
```

Chaque répertoire de `/sys/dev/block` correspond à un périphérique associé à un fichier spécial en mode bloc. Le nom est composé du numéro de majeur et du numéro de mineur du périphérique :

```
cd /sys/dev/block/11:0
```

```
cat device/model
```

```
CD-ROM
```

Le fichier spécial de majeur 11 et de mineur 0 est associé au lecteur de CD-ROM de la machine.

## 4. Identification des systèmes de fichiers

L'identification des systèmes de fichiers peut se faire de différentes manières : fichier spécial associé, chemin d'accès réseau, chemin d'accès de fichier, label, UUID.

### a. Fichier spécial en mode bloc

C'est la méthode traditionnelle Unix, reprise dans Linux. Elle permet d'identifier de façon unique le système de fichiers contenu dans l'espace de stockage associé au fichier spécial en mode bloc.

☞ Dans certains cas, la dénomination du fichier spécial correspondant au périphérique peut changer, par exemple si un disque physique est changé de connecteur, ou si l'ordre de détection des périphériques réseau est modifié.