

Les deux modes (bloc et caractère) correspondent aux deux types b et c. D'autre part, à l'emplacement habituel de la taille des fichiers, on remarque deux nombres séparés par une virgule. Il s'agit de deux informations baptisées **majeur** et **mineur**. Le majeur (20) désigne le programme "driver" dans les tables internes du noyau. Le mineur (0 ou 1) est un argument transmis au majeur. Il permet d'identifier soit un périphérique particulier, soit un mode d'utilisation du périphérique (ici, il s'agit d'une densité pour la disquette).

Parfois, plusieurs noms de fichiers correspondent au même majeur et au même mineur. Dans l'exemple, **/dev/fd0.18** et **/dev/fd0h** sont associés au même couple "20,1". Ceci traduit simplement deux noms équivalents pour désigner une même utilisation. Ces équivalences sont liées soit à un besoin de simplification des noms, soit à la préoccupation de compatibilité ascendante lors des évolutions de version. Les deux noms de l'exemple sont deux liens classiques (on remarque leur compteur de liens égal à 2).

La commande **mknod** peut aussi servir à créer un fichier spécial. On lui fournit le nom souhaité, le type (b ou c), le majeur et le mineur.

Cette commande est rarement appelée directement. Elle constitue plutôt la première étape interne de la mise en œuvre des périphériques.

On pourrait cependant l'employer ici pour créer un troisième nom de disquette :

```
mknod /dev/diskette b 20 1
```

Particularité Solaris

Les noms associés aux périphériques Solaris sont de deux types :

- Noms physiques (noms internes)

Ces noms sont situés dans le répertoire **/devices** où chaque contrôleur est représenté par un sous-répertoire contenant les véritables fichiers spéciaux.

- Noms logiques (utilisés dans les commandes)

Ils sont situés, de manière classique, dans le répertoire **/dev** mais ce sont des liens symboliques vers le répertoire **/devices**.

Exemple

```
# ls -l /dev/mouse
lrwxrwxrwx  1  root  other  32  dec 6 20:24
              /dev/mouse -> ../devices/pseudo/consms@0:mouse
# cd /devices/pseudo
# ls -l consms@0:mouse
crw-----  1  root  other  143,0  dec 6 20:39  consms@0:mouse
#
```

Mode bloc

L'accès à ces périphériques se fait à l'aide de requêtes au système de gestion de fichiers du noyau. Celui-ci gère un ensemble de **tampons caches** permettant de différer les écritures physiques et d'anticiper éventuellement les lectures.

Un périphérique géré en mode bloc donne la possibilité de création d'un filesystem. Concrètement, ce mode correspond aux disques (partitions et volumes logiques d'un disque dur, disquettes, CD-Rom).

Comme déjà précisé dans le chapitre Disques et systèmes de fichiers, ces périphériques peuvent aussi être utilisés en mode **raw bloc**, sans passer par le gestionnaire du noyau. Certaines commandes ou applications peuvent ainsi les manipuler directement. De ce fait, les périphériques en mode bloc ont deux entrées dans le répertoire **/dev**, l'une de type b (bloc) et l'autre de type c (**raw bloc**). On se rappellera que les versions Linux ne conservent pas ce principe de la double entrée mais proposent une commande **raw** pour associer une entrée **raw bloc** à la seule entrée **bloc** en cas de besoin.

Mode caractère

Ce mode ne donne pas la possibilité de création d'un filesystem. Les entrées/sorties sont directes (sans utilisation de **tampons caches** du noyau) et le pilote sait gérer des caractères de validation ou d'interruptions.

Concrètement, ce mode correspond aux terminaux, bandes magnétiques, imprimantes...

2. Recensement des périphériques présents

Les noms des fichiers spéciaux sont spécifiques aux versions, voire aux types de matériels de chaque constructeur. Chaque version propose cependant des commandes d'identification rapide des périphériques.

2.1 Périphériques AIX

AIX stocke beaucoup d'informations système dans une base de données spécifique baptisée **ODM** (*Object Data Manager*) ; en particulier, tout ce qui concerne la gestion des périphériques. La configuration courante se décline par rapport à une base de périphériques supportés. La commande **lscfg** permet d'afficher la configuration (un maximum de détails est obtenu avec l'option **-v**).

Les noms de périphériques sont très simples car ils sont constitués d'un **préfixe** et d'un **numéro d'ordre** lors de la configuration. De ce fait, le nom de fichier spécial ne précise pas directement l'emplacement physique. Cette adresse physique est affichée par la commande **lscfg** et s'interprète suivant le type de périphérique (adaptateur, périphérique SCSI, port série...). Une commande plus interne, **lsdev**, permet aussi d'obtenir des informations détaillées.

Les préfixes des périphériques usuels sont :

- *hdisk* : disques durs.
- *fd* : disquettes.
- *rmt* : bandes.
- *tty* : terminaux.
- *lp* : imprimantes.
- *cd* : CD-Rom.

Bien que très spécifique, la configuration se fait de manière intuitive via les menus de l'utilitaire **smit**. La terminologie de ces menus est la suivante :

États

Supported	Périphérique dont les caractéristiques sont connues.
Undefined	État virtuel : périphérique encore inconnu.
Defined	Périphérique déclaré mais non opérationnel.
Available	Périphérique opérationnel.

Transition d'états

Add a device	supported ou undefined vers available .
Configure a device	defined vers available .
Remove a device	Suppression complète ou retour à l'état defined .

2.2 Périphériques HP-UX

La commande **ioscan** permet d'obtenir une description complète de la configuration matérielle.

Elle est dotée de nombreuses options, dont par exemple :

-C class

Restriction de l'affichage à une certaine classe de matériel.

-f

Liste détaillée.

-n

Affichage des noms des fichiers spéciaux.

Exemple

```
# ioscan -fn -Cdisk
Class I HW Path Driver SW State Type Description
=====
disk 0 2/0/1.4.0 sdisk CLAIMED DEVICE TOSHIBA CD-ROM XM-5701TA
      /dev/dsk/c0t4d0 /dev/rdisk/c0t4d0
disk 1 2/0/1.5.0 sdisk CLAIMED DEVICE MICROP 2112
      /dev/dsk/c0t5d0 /dev/rdisk/c0t5d0
disk 2 2/0/1.6.0 sdisk CLAIMED DEVICE CONNER CFP2107S 2.14GB
      /dev/dsk/c0t6d0 /dev/rdisk/c0t6d0
#
# ioscan -f
Class I HW Path Driver SW State Type Description
=====
bc 0 root CLAIMED BUS_NEXUS
.....
ext_bus 0 2/0/1 c720 CLAIMED INTERFACE Built-in SCSI
.....
disk 0 2/0/1.4.0 sdisk CLAIMED DEVICE TOSHIBA CD-ROM XM-5701TA
disk 1 2/0/1.5.0 sdisk CLAIMED DEVICE MICROP 2112
disk 2 2/0/1.6.0 sdisk CLAIMED DEVICE CONNER CFP2107S 2.14GB
.....
lan 0 2/0/2 lan2 CLAIMED INTERFACE Built-in LAN
tty 0 2/0/4 asio0 CLAIMED INTERFACE Built-in RS-232C
tty 1 2/0/5 asio0 CLAIMED INTERFACE Built-in RS-232C
ext_bus 1 2/0/6 CentIf CLAIMED INTERFACE Built-in Parallel
.....
#
```

2.3 Périphériques Solaris

La commande **prtconf** affiche la configuration matérielle sous une forme peu claire :

```
# prtconf
System Configuration: Sun Microsystems i86pc
Memory size: 127 Megabytes
System Peripherals (Software Nodes):
i86pc
  +boot (driver not attached)
.....
  isa, instance #0
    motherboard (driver not attached)
```

```

    asy, instance #0
    asy, instance #1
    lp (driver not attached)
    fdc, instance #0
    .....
    pci, instance #0
    pci1028,4094 (driver not attached)
    display, instance #0
    .....

```

Deux commandes plus internes, **getdev** et **devattr**, peuvent fournir des informations plus concrètes :

```

# getdev type=disk
disk1
# devattr -v disk1
alias='disk1'
bdevice='/dev/dsk/c0d0s2'
capacity='0'
cdevice='/dev/rdisk/c0d0s2'
desc='Disk Drive'
dpartlist='dpart100,dpart102,dpart103,dpart104,dpart105'
part='true'
removable='false'
type='disk'
#

```

2.4 Périphériques Linux

La commande **dmesg** affiche tous les messages concernant le démarrage du système. On y trouve une bonne description des périphériques reconnus.

Exemple

On extrait ici tous les messages concernant les disques (**hd**).

```

# dmesg | grep hd
Kernel command line: ro root=/dev/hda5
    ide0: BM-DMA at 0xfcd0-0xfcd7, BIOS settings: hda:pio, hdb:pio
    ide1: BM-DMA at 0xfcd8-0xfcdf, BIOS settings: hdc:pio, hdd:pio
    hda: FUJITSU MHD2032AT, ATA DISK drive
    hdb: SR200S, ATAPI CD/DVD-ROM drive

```