

Gestionnaire de périphériques UDEV

1. Découverte des périphériques

a. Enregistrement d'un périphérique

Le répertoire `/dev` contient, sous la forme d'un fichier spécial, tout périphérique système que l'on appelle nœud (*node* en anglais). Concrètement, cela signifie que le contenu peut y être lu suivant la catégorie du périphérique. Le mode bloc, où les données sont inscrites en groupes continus sur un support persistant ; et le mode caractère, où les données sont lues octet par octet en fonction d'un flux.

Note : toutes les commandes sont à faire en mode console et avec les droits d'administration (en `root`).

Démonstration pour le mode bloc

- Tapez la commande `less -f /dev/sda`

Le périphérique `sda` représente votre disque (changez par `hda` si vous disposez d'un disque IDE). Le contenu affiché par la commande `less` ou l'option `-f` force l'ouverture d'un fichier non-texte, représente tout simplement le contenu de votre disque. Les caractères en ASCII sont d'ailleurs lisibles.

- Quittez par la lettre `q`.

Démonstration pour le mode caractère

La souris est un périphérique usuel pour le mode caractère. Dans une console texte, il faut installer le paquet `gpm` pour disposer de la souris (avec les présentations semi-graphiques et la bibliothèque `curse`).

- Tapez la commande `aptitude install gpm`

Une fois installé, en bougeant la souris, le pavé se déplace à l'écran.

- Tapez la commande `cat /dev/mouse`

En bougeant cette fois la souris, des caractères s'affichent en temps réel, correspondant aux caractères liés au protocole de la souris.

- Quittez par `[Ctrl] c`.

b. Découverte dynamique

L'affichage par la commande `ls` du répertoire `/dev` retourne une longue liste de périphériques définis par le système ; certains étant utilisés, d'autres pas. La gestion du répertoire `/dev`, celle des droits et des permissions, de l'identification des fichiers de périphériques, autrefois du ressort de **devfs** (*DEVICE File System*) est maintenant dévolu à **udev** (*Userspace implementation od DEVfs*). Udev apporte :

- l'emploi de règles au niveau de l'utilisateur à la différence de `devfs` fonctionnant, lui, au niveau du noyau.
- une gestion dynamique des périphériques réellement présents avec la prise en compte de la connexion "à chaud" (`hotplug`).
- une association avec `sysfs`, le système de fichiers du noyau à partir de la version 2.6.

2. Fonctionnement d'UDEV

a. Fichiers de base

Udev crée et enlève les nœuds dans `/dev`, à partir des événements envoyés par le noyau en cas de découverte de périphériques. La commande `df` montre bien `udev` comme système de fichiers pour `/dev`. La liste des périphériques toujours attendus et gérés par `udev` se trouve dans `/lib/udev/devices` :

```
root@ubuntu:~# ls -l /lib/udev/devices/
total 12
crw----- 1 root root    5, 1 2008-05-16 08:31 console
lrwxrwxrwx 1 root root    11 2008-05-16 08:31 core -> /proc/kcore
lrwxrwxrwx 1 root root    13 2008-05-16 08:31 fd -> /proc/self/fd
crw-r----- 1 root kmem    1, 2 2008-05-16 08:31 kmem
brw----- 1 root root    7, 0 2008-05-16 08:31 loop0
lrwxrwxrwx 1 root root    13 2008-05-16 08:31 MAKEDEV -> /sbin/MAKEDEV
drwxr-xr-x 2 root root 4096 2008-05-16 08:31 net
crw----- 1 root root    1, 3 2008-05-16 08:31 null
crw----- 1 root root 108, 0 2008-05-16 08:31 ppp
drwxr-xr-x 2 root root 4096 2008-04-11 14:21 pts
drwxr-xr-x 2 root root 4096 2008-04-11 14:21 shm
lrwxrwxrwx 1 root root    24 2008-05-16 08:31 sndstat -> /proc/asound/oss/sndstat
lrwxrwxrwx 1 root root    15 2008-05-16 08:31 stderr -> /proc/self/fd/2
lrwxrwxrwx 1 root root    15 2008-05-16 08:31 stdin -> /proc/self/fd/0
lrwxrwxrwx 1 root root    15 2008-05-16 08:31 stdout -> /proc/self/fd/1
root@ubuntu:~#
```

Les noms persistants donnés par `udev` sont situés dans `/dev/disk` :

```
root@ubuntu:~# ls -l /dev/disk
total 0
drwxr-xr-x 2 root root 120 2008-05-16 10:16 by-id
drwxr-xr-x 2 root root 140 2008-05-16 10:16 by-path
drwxr-xr-x 2 root root 80 2008-05-16 10:16 by-uuid
root@ubuntu:~# ls -lR /dev/disk
/dev/disk:
total 0
drwxr-xr-x 2 root root 120 2008-05-16 10:16 by-id
drwxr-xr-x 2 root root 140 2008-05-16 10:16 by-path
drwxr-xr-x 2 root root 80 2008-05-16 10:16 by-uuid

/dev/disk/by-id:
total 0
lrwxrwxrwx 1 root root    9 2008-05-16 10:16 edd-int13_dev80 -> ../../sda
lrwxrwxrwx 1 root root   10 2008-05-16 10:16 edd-int13_dev80-part1 -> ../../sda1
lrwxrwxrwx 1 root root   10 2008-05-16 10:16 edd-int13_dev80-part2 -> ../../sda2
lrwxrwxrwx 1 root root   10 2008-05-16 10:16 edd-int13_dev80-part5 -> ../../sda5

/dev/disk/by-path:
total 0
lrwxrwxrwx 1 root root   10 2008-05-16 10:16 pci-0000:00:07.1-scsi-1:0:0:0 -> ../../scd0
lrwxrwxrwx 1 root root    9 2008-05-16 10:16 pci-0000:00:10.0-scsi-0:0:0:0 -> ../../sda
lrwxrwxrwx 1 root root   10 2008-05-16 10:16 pci-0000:00:10.0-scsi-0:0:0:0-part1 -> ../../sda1
lrwxrwxrwx 1 root root   10 2008-05-16 10:16 pci-0000:00:10.0-scsi-0:0:0:0-part2 -> ../../sda2
lrwxrwxrwx 1 root root   10 2008-05-16 10:16 pci-0000:00:10.0-scsi-0:0:0:0-part5 -> ../../sda5

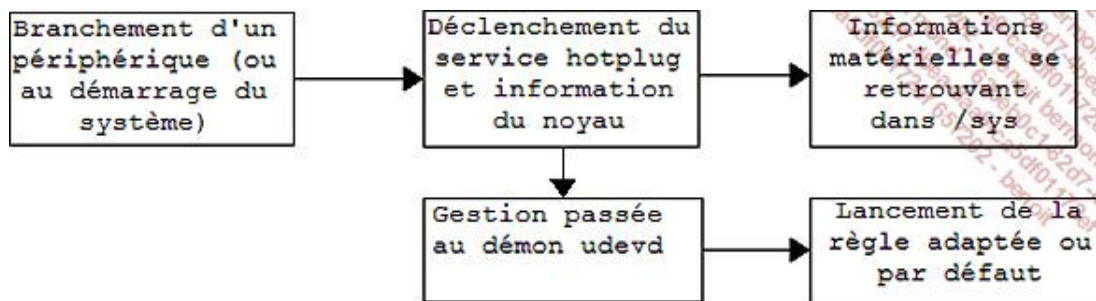
/dev/disk/by-uuid:
total 0
lrwxrwxrwx 1 root root   10 2008-05-16 10:16 49b50a9d-3569-4f46-97e3-6775106ceb54 -> ../../sda5
lrwxrwxrwx 1 root root   10 2008-05-16 10:16 98186f7c-8f1e-4225-aabc-05eedc89b1f8 -> ../../sda1
root@ubuntu:~#
```

On voit bien par exemple les UUID de chaque partition du disque dur dans le sous-répertoire `/dev/disk/by-uuid`. Sur Ubuntu, toutes les règles d'`udev` (extension `rules`) sont quant à elles situées dans `/etc/udev/rules.d` :

```
root@duncan:~# ls -l /etc/udev/rules.d/
total 20
-rw-r--r-- 1 root root 208 2010-04-30 13:56 10-vboxdrv.rules
-rw-r--r-- 1 root root 55 2010-04-30 14:27 51-android.rules
-rw-r--r-- 1 root root 854 2010-04-30 11:26 70-persistent-cd.rules
-rw-r--r-- 1 root root 786 2010-04-30 10:51 70-persistent-net.rules
-rw-r--r-- 1 root root 1157 2010-04-19 11:30 README
```

b. Principe d'une règle

Le fonctionnement d'udev déterminant une règle s'établit dans le schéma suivant :



Les règles s'appliquent normalement dans l'ordre de numérotation des fichiers. Une règle comporte une série de clés sous la forme de paires clés/valeurs et s'applique si et seulement si toutes les assertions données par les valeurs sont exactes. Exemples de clés :

- **KERNEL** : nom donné au périphérique par le noyau.
- **SUBSYSTEM** : nom donné au sous-système du périphérique.
- **DRIVER** : pilote concerné par le périphérique.

Les renseignements sont donnés par la commande `udevinfo` dont voici un extrait d'une sortie sur un disque SCSI :

```
root@ubuntu:/etc/udev/rules.d# udevinfo -a -p /sys/class/scsi_device/1:0:0:0/

Udevinfo starts with the device specified by the devpath and then
walks up the chain of parent devices. It prints for every device
found, all possible attributes in the udev rules key format.
A rule to match, can be composed by the attributes of the device
and the attributes from one single parent device.

looking at device '/class/scsi_device/1:0:0:0':
  KERNEL=="1:0:0:0"
  SUBSYSTEM=="scsi_device"
  DRIVER=""

looking at parent device '/devices/pci0000:00/0000:00:07.1/host1/target1:0:0:1:0:0:0':
  KERNELS=="1:0:0:0"
  SUBSYSTEMS=="scsi"
  DRIVERS=="sr"
  ATTRS{device_blocked}=="0"
  ATTRS{type}=="5"
  ATTRS{scsi_level}=="6"
  ATTRS{vendor}=="NECUMWar"
  ATTRS{model}=="VMware IDE CDR10"
  ATTRS{rev}=="1.00"
  ATTRS{state}=="running"
```

Grâce à ces règles, on peut gérer les permissions de chaque périphérique, exécuter des programmes en fonction d'évènements, etc. L'exemple typique d'administration des périphériques USB concerne l'attribution d'un nom persistant et ce, en fonction du fichier spécial dans `/dev`. Une clé USB peut se voir ainsi attribuer un nom associé au périphérique `/dev/sdc` quel que soit le nombre de périphériques USB branché.

Pour plus de renseignements sur les commandes d'udev, reportez-vous au manuel en ligne et à l'adresse : <http://www.kernel.org/pub/linux/utils/kernel/hotplug/udev.html>

c. Problème en cas de changement de carte réseau

Le mieux est parfois l'ennemi du bien et un problème se pose pour la carte réseau. La règle d'attribution d'udev pour une interface réseau indique naturellement l'utilisation de son adresse MAC. Or dans certains cas, lors du changement de carte réseau ou dans le cadre de la virtualisation de serveur, cette attribution de nom deviendra obsolète. Une nouvelle carte ou un changement verra le nom basé sur l'adresse MAC attribué à `eth1` et non `eth0`.

Les services, comme par exemple un serveur DHCP, basés sur `eth0` ne fonctionneront plus.

Le fichier des règles Ubuntu concerné est :

- `75-persistent-net.generator.rules` : règle établissant les associations des interfaces réseau avec les adresses MAC et situé dans le répertoire `/lib/udev/Rules.d`.
- `70-persistent-net.rules` : généré par la règle ci-dessus.

Solution n°1

Détruire la (ou les) ligne(s) de la règle du fichier `70-persistent-net.rules` et redémarrer le système. Ce fichier reconstruit contiendra la bonne association.

Solution n°2

Encore plus radical : détruire les deux fichiers de règles. Ceci est à éviter pour respecter les règles de noms des périphériques dans le cadre d'une gestion par `udev`.