



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

udev: definir el nombre de los /dev para dispositivos removibles (34243 lectures)

Per **Ricardo Galli Granada**, [gallir](http://mnm.uib.es/gallir/) (<http://mnm.uib.es/gallir/>)

Creado el 14/11/2004 13:43 modificado el 14/11/2004 13:43

De algunos comentarios en mi [artículo anterior](#)⁽¹⁾, está claro que la gente prefiere unos artículos técnicos a mis... ergh... [profundos ensayos filosóficos](#)⁽²⁾... si queréis más circo, tengamos circo ;-). El tema es que tengo una Palm (cuya memory stick puede accederse vía usb-storage), un par de pendrives, un par de cámaras digitales, y me molestaba mucho tener que buscar el nombre del dispositivo en /dev cada vez que los enchufaba para luego hacer el mount. A grandes remedios grandes soluciones, instalé el [udev](#)⁽³⁾ y me aprendí el language de [reglas](#)⁽⁴⁾ para definir los dispositivos. Es más fácil de lo que aparente, y potentísimo (ya veréis los avances espectaculares que se producirán en temas de removibles sobre GNU/Linux).

Como ya comenté, hay que instalar el [udev](#)⁽³⁾. En Debian es directo, `apt-get install udev` y funciona sin problemas desde ese momento (suponiendo que tenéis el [kernel correctamente configurado](#)⁽⁵⁾), aunque os recomiendo fuertemente que reiniciéis el sistema antes de trastear con él (al arrancarse crea unos dispositivos básicos que deben ser los que usa el kernel).

El udev es muy potente, básicamente crea un sistema de ficheros en /dev (del tipo tmpfs) y allí crea dinámicamente los dispositivos a medida que el kernel los reconoce. Lo potente no solo está en que funciona como una seda, sino que además casi todo lo hace en *userspace*, así que hacer que el usuario [pueda definir como denominar](#)⁽⁴⁾ a los dispositivos o indicar que cree enlaces simbólicos es sólo un paso.

Esas especificaciones van en el directorio `/etc/udev/rules.d/` y se tienen que acabar con la extensión `rules`. Os explicaré brevemente mis tres reglas en `/etc/udev/rules.d/mis_removibles.rules` me permitieron tener un `fstab` como el siguiente:

`/etc/fstab`

```
...
/dev/sonyl      /media/sony    auto    defaults,user,noauto 0 0
/dev/jetflash   /media/jetflash auto    defaults,user,noauto 0 0
...
```

Fichero `/etc/udev/rules.d/mis_removibles.rules`

Regla 1: Crea un enlace simbólico al primer dispositivo ttyUSB?

BUS="usb", SYSFS{manufacturer}="Palm. Inc.", KERNEL="ttyUSB[024]", NAME="%k", SYMLINK="pilot"

Regla 2: Cuando el Palm se conecta como usb-storage crea los `/dev/sony*`
con todas las particiones

BUS="usb", SYSFS{product}="Sony PEG Mass Storage", NAME{all_partitions}="sony"

Regla 3: Crea el `/dev/jetflash` cuando se inserta my pendrive habitual

BUS="usb", SYSFS{manufacturer}="JetFlash", KERNEL="sd?1", NAME="%k", SYMLINK="jetflash"

en este caso es equivalente a:

BUS="scsi", SYSFS{vendor}="JetFlash", NAME="%k", KERNEL="sd?1", SYMLINK="jetflash"

El sistema de reglas es bastante sencillo, se deben cumplir las condiciones (en rojo) para que se ejecuten las acciones (en negro). Hay un tutorial [bastante completo](#)⁽⁴⁾, pero en mi caso es obvio de imaginarse y/o bastante sencillo de explicar. Por ejemplo, `BUS`, `SYSFS`, y `KERNEL` son condiciones que se deben cumplir. `NAME` indica el nombre que deberá tener el dispositivo, `SYMLINK` si se crea un enlace simbólico y el `%k` es el nombre que le da el kernel por defecto.



Explicaré brevemente qué hace cada regla y finalmente como he creado la regla 3 (que es similar a las demás) con la ayuda del `udevinfo`.

Regla 1

Cada vez que se enchufe el Palm se crean dos dispositivos `/dev/ttyUSB?`. El Kpilot usa siempre el primero y me interesaba llamarlo `/dev/pilot`, así que crea ese enlace hacia el primero de los `ttyUSB` (que es el 0, 2, 4...).

```
$ ls -l /dev/ttyUSB* /dev/pilot
lrwxrwxrwx 1 root root          7 Nov 14 12:39 /dev/pilot -> ttyUSB0
crw-rw---- 1 root dialout 188, 0 Nov 14 12:40 /dev/ttyUSB0
crw-rw---- 1 root dialout 188, 1 Nov 14 12:39 /dev/ttyUSB1
```

Regla 2

El Sony Clié puede tener una *memory stick* que puede ser montada como un `usb-storage` normal. Tenía un pequeño gran problema. Cuando se conecta el Palm, éste no envía el tamaño de la memoria, por lo tanto el kernel no detecta inmediatamente la primera partición y el dispositivo no se crea en `/dev`. Usando el `NAME{all_partitions}` obligo al udev a que cree todas las particiones inmediatamente.

```
$ ls -l /dev/sony*
brw-rw---- 1 root root 8, 32 Nov 14 12:38 /dev/sony
brw-rw---- 1 root root 8, 33 Nov 14 12:38 /dev/sony1
brw-rw---- 1 root root 8, 42 Nov 14 12:38 /dev/sony10
brw-rw---- 1 root root 8, 43 Nov 14 12:38 /dev/sony11
brw-rw---- 1 root root 8, 44 Nov 14 12:38 /dev/sony12
brw-rw---- 1 root root 8, 45 Nov 14 12:38 /dev/sony13
brw-rw---- 1 root root 8, 46 Nov 14 12:38 /dev/sony14
brw-rw---- 1 root root 8, 47 Nov 14 12:38 /dev/sony15
brw-rw---- 1 root root 8, 34 Nov 14 12:38 /dev/sony2
brw-rw---- 1 root root 8, 35 Nov 14 12:38 /dev/sony3
brw-rw---- 1 root root 8, 36 Nov 14 12:38 /dev/sony4
brw-rw---- 1 root root 8, 37 Nov 14 12:38 /dev/sony5
brw-rw---- 1 root root 8, 38 Nov 14 12:38 /dev/sony6
brw-rw---- 1 root root 8, 39 Nov 14 12:38 /dev/sony7
brw-rw---- 1 root root 8, 40 Nov 14 12:38 /dev/sony8
brw-rw---- 1 root root 8, 41 Nov 14 12:38 /dev/sony9
```

Regla 3

Esta regla la uso para crear el enlace simbólico `/dev/jetdrive` que apunta al dispositivo SCSI que sea crea al enchufarlo (normalmente `/dev/sd+`). Veréis que hay una condición adicional, es para verificar que se trata del dispositivo de bloques SCSI y no uno de los dispositivos de emulación que se crean adicionalmente (`/dev/sg?`). La regla `KERNEL=sd?1` indica que debe ser la partición 1 de uno de esos dispositivos (`sda`, `sdb`, `sdc`...).

```
$ ls -l /dev/jet*
lrwxrwxrwx 1 root root 4 Nov 14 12:22 /dev/jetflash -> sdb1
```

Cómo crear las reglas

Para poder crear esas reglas necesitáis saber un mínimo de información. El `udevinfo` os la dará. En el caso del `jetflash`, lo enchufé y veo con el comando `dmesg` que crea por defecto los dispositivos `/dev/sdc` y `/dev/sdc1`. Lo primero que hay que hacer es encontrar el *path* dentro del `/sys` para obtener la información completa del dispositivo. Para ello hacemos:

```
$ udevinfo -q path -n /dev/sdc1
/block/sdc/sdc1
```

Con esa información ejecutamos ahora el comando a continuación. Marqué en negrita la información relevante para crear las reglas:

```
$ udevinfo -a -p /block/sdc/sdc1
```



```
...
looking at class device '/sys/block/sdc/sdc1':
SYSFS{dev}="8:33"
SYSFS{size}="250666"
SYSFS{start}="62"
SYSFS{stat}="          0          0          0          0"

follow the class device's "device"
looking at the device chain at '/sys/devices/pci0000:00/0000:00:1d.7/usb5...':
  BUS="scsi"
  ID="5:0:0:0"
  ...
  SYSFS{type}="0"
  SYSFS{vendor}="JetFlash"
  ...

looking at the device chain at '/sys/devices/pci0000:00/0000:00:1d.7/usb5...':
  BUS="usb"
  ID="5-4"
  SYSFS{bConfigurationValue}="1"
  ...
  SYSFS{manufacturer}="JetFlash"
  SYSFS{maxchild}="0"
  SYSFS{product}="TS128MJF2A"
  SYSFS{speed}="480"
  SYSFS{version}=" 2.00"
  ...
```

Supongo que ya os habéis dado cuenta de dónde he obtenido la información para especificar la regla 3. Lo bueno del udev es que podéis modificar el fichero de reglas en cualquier momento y **sólo tenéis que volver a enchufar el dispositivo para que ya tengan efectos vuestras modificaciones**. No hace falta rearrancar nada... flexible y potente el udev ¿no?

Lista de enlaces de este artículo:

1. <http://bulma.net/body.phtml?nIdNoticia=2113>
2. <http://bulma.net/body.phtml?nIdNoticia=2099>
3. <http://www.kernel.org/pub/linux/utils/kernel/hotplug/udev.html>
4. <http://www.reactivated.net/udevrules.php>
5. http://webpages.charter.net/decibelshelp/LinuxHelp_UDEVPrimer.html#Steps

E-mail del autor: gallir_ARROBA_uib.es

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=2114>