
Dispositivos

Diseño y Administración de Sistemas y Redes

Juan Céspedes <cespedes@gsync.escet.urjc.es>



Curso 2005–2006



El *hardware* en un sistema UNIX

- Al instalar un periférico, es necesario configurar el sistema operativo para que lo reconozca y poder hacer uso de él.
- En los primeros sistemas, implicaba casi siempre la generación de un nuevo núcleo que incluya los nuevos *drivers*.
- Actualmente, la mayor parte de los sistemas UNIX son modulares y se pueden añadir dispositivos sin necesidad de tener que apagar el sistema.

¿Qué es un dispositivo?

- Todos los periféricos en UNIX están representados por un fichero en el directorio `/dev`.
- Cuando el sistema operativo detecta que un proceso está accediendo a un fichero de dispositivo, convierte la referencia en órdenes *hardware*.
- Los ficheros de `/dev` no son *realmente* el dispositivo; no tienen tamaño ni contenido, solo son referencias para que el sistema operativo nos ponga en contacto con el *hardware*.

Tipos de dispositivos

- **Dispositivos de caracteres**
Se refieren a periféricos cuyo acceso a los datos se realiza de forma secuencial, y no se utilizan los *buffers* del sistema.
Ejemplo: `/dev/ttyS0`
- **Dispositivos de bloques**
Acceso aleatorio y por bloques a los datos, uso del *buffer cache* del sistema.
Ejemplo: `/dev/hda1`

Dispositivos no asociados a *hardware*

- La mayor parte de los ficheros de dispositivos están asociados a un periférico, pero no todos.
- Ejemplos:
 - `/dev/null`: Al leerlo, se comporta como un fichero vacío. Al escribir, es como un “agujero negro”: se descarta todo lo que recibe.
 - `/dev/zero`: Al leerlo, es una fuente inagotable de bytes 0.
 - `/dev/tty`: *Pseudo-terminal*: se comporta como una línea serie entre dos procesos, de manera que uno de ellos tenga la impresión de estar conectado con una terminal real.

Características de los dispositivos

Cada fichero de dispositivo tiene una entrada en el directorio y un inodo, pero sin bloques de datos asociados. Por lo tanto, tienen:

- Nombre
- Propietario y grupo
- Permisos de acceso
- Fecha de creación, última modificación y último acceso

Adicionalmente, hay otros tres datos que los identifican:

- Tipo de dispositivo (de **bloques** o de **caracteres**)
- *Major number*
- *Minor number*

Características de los dispositivos (2)

Tanto el *major number* como el *minor number* son números sin signo de 8 bits (esto es, con valores del 0 al 255).

- El *major number* indica el tipo de dispositivo.
- El *minor number* indica el número de unidad de todos los dispositivos del mismo tipo.

Por ejemplo, el fichero:

```
brw-rw---- 1 root disk 3, 1 2005-04-11 11:55 /dev/hda1
```

indica que `/dev/hda1` es un dispositivo de bloques con *major number*=3 (correspondiente con el primer disco duro IDE del sistema) y *minor number*=1 (primera partición de ese disco duro).

Dispositivos en Linux

Como en cualquier otro sistema operativo, el *kernel* de Linux tiene que estar configurado para los dispositivos que reconoce.

En Linux, la lista de dispositivos que reconoce actualmente el *kernel* está en el fichero `/proc/devices`, divididos en dispositivos de caracteres y de bloques, e indicando el *major number* de cada tipo.

Para cambiar algún dispositivo de esta lista, es necesario añadir o eliminar un *driver* del *kernel*, y eso se puede hacer:

- Cambiando la configuración del *kernel* y recompilándolo.
- Añadiendo o borrando algún módulo al *kernel* actual.

Módulos del *kernel*

- Lista de módulos activos: `/proc/modules`. Se puede consultar con la orden `lsmod`.
- La lista de módulos disponibles para el *kernel* que se está ejecutando está en `/lib/modules/`uname -r``. Se puede consultar con la orden `modprobe -l`.
- Para insertar un módulo en el *kernel*, se utilizan las órdenes `insmod módulo` o `modprobe módulo`.
- Para eliminar un módulo del *kernel*, se usa `rmmmod módulo` o `modprobe -r módulo`.

Uso del nuevo *hardware*

Para usar un periférico nuevo, no basta con que el *kernel* lo reconozca: el *kernel* se comporta simplemente como un interfaz entre los procesos de usuario y el periférico; necesitamos que haya procesos que lo configuren y controlen.

Ejemplo: un disco duro. Además de que el *kernel* lo reconozca, hay que:

- Elegir y crear las particiones adecuadas.
- Darle formato a cada partición que vayamos a usar.
- Elegir un punto de montaje para cada partición.
- Configurar el sistema para que monte las nuevas particiones en el arranque.

Dispositivos sin fichero asociado

A pesar de lo dicho, la realidad es que existen ciertos dispositivos que se manejan de otra manera, y no tienen ningún fichero asociado en `/dev`:

- Teclado
Englobado en otros dispositivos, las *consolas virtuales*.
- Tarjetas gráficas
En modo texto, son parte de las *consolas virtuales*; en modo gráfico, son controladas por procesos de usuario (las X).
- Tarjetas de red
Controlados completamente por el *kernel*, pero con un interfaz propio y distinto al del resto de los periféricos.

Recursos usados por el hardware

Independientemente de si el *kernel* reconoce un determinado dispositivo o no, hay algunos que se muestran como pertenecientes a un *bus* y se puede consultar su tipo, nombre, función, y algunas cosas más:

- Dispositivos PCI: mediante la orden `"lspci"`.
- Dispositivos USB: mediante la orden `"lsusb"`.
- Dispositivos PCMCIA: usando la orden `"cardctl status"`.

Además, para los reconocidos por el *kernel*, tenemos información de los recursos que utilizan en los ficheros `/proc/interrupts` (lista de interrupciones *hardware*), `/proc/ioports` (puertos de entrada-salida) y `/proc/iomem` (puertos *mapeados* en memoria).