

# Administración de Servidores

## El núcleo Linux

### Gestión del tiempo de ejecución del núcleo y resolución de problemas

Creado por

Iñaki Fernández de Viana y González

Huelva, octubre 2020

# Sobre Nosotros

## Iñaki Fernández de Viana y González



Despacho 128. Escuela Técnica  
Superior de Ingeniería



Dpto. De Tecnologías de la Información



[i.fviana@dti.uhu.es](mailto:i.fviana@dti.uhu.es)



+34 959217378



# Objetivos

- **Peso:** 4
- **Descripción:** Candidates should be able to manage and/or query a 2.6.x, 3.x or 4.x kernel and its loadable modules. Candidates should be able to identify and correct common boot and run time issues. Candidates should understand device detection and management using udev. This objective includes troubleshooting udev rules.

# Objetivos (II)

- **Áreas clave de conocimiento:** Use command-line utilities to get information about the currently running kernel and kernel modules; Manually load and unload kernel modules; Determine when modules can be unloaded; Determine what parameters a module accepts; Configure the system to load modules by names other than their file name; /proc filesystem; Content of /, /boot, and /lib/modules; Tools and utilities to analyse information about the available hardware; udev rules

# Objetivos (III)

- **Términos y utilidades:** /lib/modules/kernel-version/modules.dep; module configuration files in /etc/; /proc/sys/kernel/; /sbin/depmod; /sbin/rmmod; /sbin/modinfo; /bin/dmesg; /sbin/lspci; /usr/bin/lsdev; /sbin/lsmmod; /sbin/modprobe; /sbin/insmod; /bin/uname; /usr/bin/lsub; /etc/sysctl.conf, /etc/sysctl.d/; /sbin/sysctl; udevmonitor; udevadm monitor; /etc/udev/



# Índice

1. Introducción
2. Módulos del núcleo
3. Parámetros
4. Obteniendo información del hardware
5. Udevd

# Introducción

---

1

---

# Introducción

- ★ El núcleo es un programa modular, estos módulos se van cargando/descargando según las necesidades.
- ★ Además, el comportamiento del núcleo se puede modificar cambiando el valor de ciertos parámetros.
- ★ Estos parámetros pueden afectar a módulos particulares o al *core* del núcleo.
- ★ Los valores de estos parámetros se pueden modificar en tiempo de ejecución
- ★ Una vez cargados los módulos se realizan una serie de acciones gestionadas por udev.



# Módulos del núcleo

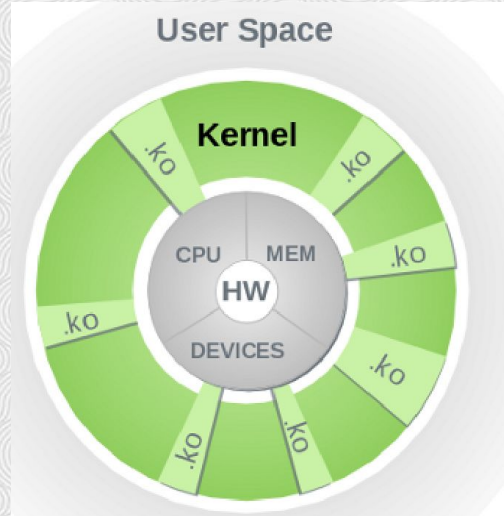
---

2

---

# Introducción

- ★ El núcleo es capaz de gestionar adecuadamente sus módulos.
- ★ Hay veces que tenemos que indicarle los módulos a cargar y qué configuraciones (opciones y/o parámetros) deben tener.



## Los ficheros

- ★ Los módulos de cada núcleo se almacenan en `/lib/modules/`.
- ★ Cada núcleo tiene su propia subcarpeta:

```
★ [root@localhost /lib/modules]# ls
★ 5.4.42-20.1ts2019 5.4.71-71.1ts2019 5.9.0-991.native
```

- ★ Para saber qué núcleo está actualmente en ejecución usamos el comando `uname`:

```
[root@localhost /lib/modules]# uname -a
Linux localhost 5.9.0-991.native #1 SMP Mon Oct 12 08:10:54 PDT 2020 x86_64 GNU/Linux
```

## Los ficheros (II)

★ El comando `uname` devuelve:

- Nombre del núcleo (**-s**)
- Nombre de la máquina (**-n**)
- Versión del núcleo (**-r**)
- Información adicional sobre el núcleo (**-v**)
- Arquitectura (**-m**)
- Sistema operativo (**-o**)

```
[root@localhost /lib/modules]# uname -a  
Linux localhost 5.9.0-991.native #1 SMP Mon Oct 12 08:10:54 PDT 2020 x86_64 GNU/Linux
```

## Los ficheros (III)

- ★ Si sabemos el núcleo en ejecución, podemos obtener mostrar sus módulos:

```
[root@localhost /lib/modules]# ls $(uname -r)
build          kernel          modules.alias.bin  modules.builtin.bin
modules.dep    modules.devname modules.softdep     modules.symbols.bin
extra          modules.alias   modules.builtin     modules.builtin.modinfo
modules.dep.bin modules.order    modules.symbols
```

- ★ En el directorio `/lib/modules/$(uname -r)/kernel` están los módulos por categorías:

```
[root@localhost /lib/modules]# ls $(uname -r)/kernel
arch crypto drivers fs kernel lib mm net security sound
```

## Los ficheros (IV)

- ★ Para consultar los sistemas de ficheros soportados, consultamos `/lib/modules/$(uname -r)/kernel/fs`:

```
[root@localhost /lib/modules]# ls $(uname -r)/kernel/fs
9p  cachefiles  ceph  cifs  cramfs  dlm  ecryptfs  exfat  fscache  fuse  hfsplus
isofs  lockd  nfs  nfs_common  nfsd  nls  overlayfs  pstore  udf  vboxsf
```

- ★ Existen dependencias entre módulos esta información se guardan en los fichero `modules.dep` y lo genera el comando `depmod`.

```
[root@localhost /lib/modules]# ls $(uname -r)
[...]
modules.dep
```



## Los ficheros (V)

- ★ Para consultar los sistemas de ficheros soportados, consultamos `/lib/modules/$(uname -r)/kernel/fs:`

```
[root@localhost /lib/modules]# head modules.dep
kernel/arch/x86/events/amd/power.ko:
kernel/arch/x86/kvm/kvm-amd.ko: kernel/drivers/crypto/ccp/ccp.ko
kernel/arch/x86/kernel/cpu/mce/mce-inject.ko:
kernel/arch/x86/crypto/des3_edc-x86_64.ko: kernel/lib/crypto/libdes.ko
kernel/arch/x86/crypto/blowfish-x86_64.ko: kernel/crypto/blowfish_common.ko
kernel/arch/x86/crypto/cast5-avx-x86_64.ko: kernel/crypto/cast5_generic.ko
kernel/crypto/cast_common.ko
[...]
```

## Mostrar módulos cargados

- ★ Podemos mostrar el listado de módulos que están actualmente cargados si usamos el comando **lsmod**.
- ★ Muestra el nombre, tamaño y “cosas” que depende de él

```
[root@localhost /lib/modules]# lsmod
Module                Size  Used by
[...]
vboxpci               23236  0
vboxnetadp            25670  0
vboxnetflt            27612  1
vboxdrv               320274  5 vboxpci,vboxnetadp,vboxnetflt
rfcomm                47864  12
```

## Cargar módulos

- ★ El núcleo es el encargado de cargar los módulos según vaya detectando eventos.
- ★ En caso de querer cargar módulos manualmente podemos usar `insmod` o `modprobe`.
- ★ Es recomendable usar `modprobe` y no `insmod`

## insmod

- ★ El comando **insmod** Debemos indicar la ruta completa del fichero del módulo a cargar:

```
[root@localhost /lib/modules]# insmod  
/lib/modules/3.8.0-30-generic/kernel/drivers/net/ethernet/8390/axnet_cs.ko  
[root@localhost /lib/modules]# lsmod | grep axnet_cs  
Module          Size      Used by  
axnet_cs         26072      0
```

- ★ **insmod**, además, no resuelve las dependencias entre módulos por lo que es necesarios cargarlas manualmente

```
[root@localhost /lib/modules]# grep axnet_cs /lib/modules/`uname -r`/modules.dep  
kernel/drivers/net/ethernet/8390/axnet_cs.ko:  
    kernel/drivers/pcmcia/pcmcia.ko kernel/drivers/pcmcia/pcmcia_core.ko
```

## modprobe

- ★ El comando `modprobe` permite cargar módulos, los busca dentro del directorio que contiene los módulos y consulta el fichero `modules.dep` para cargar las dependencias:

```
[root@localhost /lib/modules]# modprobe axnet_cs
```



## Descargar módulos

- ★ Descargamos módulos por:
  - Liberar espacio en memoria
  - Evitar conflictos con otros módulos
  - Reducir consumo
- ★ Para poder descargar un módulo, los recursos que gestiona deben estar libres.
- ★ Una vez que se descargan, los recursos estarán inaccesibles para el SO
- ★ Podemos usar los comandos `rmmod` y `modprobe`



## rmmod

- ★ El comando **rmmod** permite descargar módulos cargados en memoria, hay que indicarle el nombre el módulo a descargar:

```
[root@localhost /lib/modules]# rmmod axnet_cs
```

- ★ Si el módulo está en uso, dará un error. Para esperar a que el módulo quede libre usamos la opción **-w**

```
[root@localhost /lib/modules]# rmmod -w axnet_cs
```

- ★ **rmmod** no gestiona las dependencias entre módulos, podemos forzar la descarga usando la opción **-f** (no recomendado)

```
[root@localhost /lib/modules]# rmmod -f axnet_cs
```

## modprobe

- ★ El comando **modprobe**, gracias a la opción -r, permite descargar un módulo y los que dependan de él:

```
[root@localhost /lib/modules]# modprobe -r fat
```

# Información sobre los módulos

- ★ Podemos obtener información sobre los módulos usando el comando **modinfo**

```
[root@localhost /lib/modules]# modinfo cdrom
filename:          /lib/modules/5.9.0-991.native/kernel/drivers/cdrom/cdrom.ko
license:           GPL
retpoline:         Y
intree:            Y
name:              cdrom
vermagic:          5.9.0-991.native SMP mod_unload
parm:              debug:bool
parm:              autoclose:bool
parm:              autoeject:bool
parm:              lockdoor:bool
parm:              check_media_type:bool
parm:              mrw_format_restart:bool
```

## Información sobre los módulos (ii)

- ★ Las tuplas **parm** nos indican los parámetros que admite el módulo.
- ★ Podemos establecer los valores de estos parámetros, temporalmente, al cargar el módulo

```
[root@localhost /lib/modules]# modprobe cdrom lockdoor=0
```

# Parámetros

---

3

---

# Introducción

- ★ El núcleo y sus módulos pueden modificar su comportamiento atendiendo a los valores asignados a una serie de parámetros.
- ★ Estos valores pueden tener un carácter permanente o temporal.



# Módulos

- ★ El núcleo y sus módulos pueden modificar su comportamiento atendiendo a los valores asignados a una serie de parámetros.
- ★ En el caso de los módulos, los valores de los parámetros se indican, temporalmente, al cargar el módulo:

```
[root@localhost /lib/modules]# modprobe cdrom lockdoor=0
```

- ★ Si queremos que estos valores tengan un carácter permanente, usaremos el directorio `/etc/modprobe.d`
- ★ Este directorio contiene ficheros de texto con extensión `.conf` en el que se indican los parámetros de los módulos

## Módulos (II)

- ★ El núcleo y sus módulos pueden modificar su comportamiento atendiendo a los valores asignados a una serie de parámetros.

```
[root@localhost /lib/modules]# # ls -l /etc/modprobe.d/
-rw-r--r--  1 root root  325 abr 10  2014 blacklist-ath_pci.conf
-rw-r--r--  1 root root 1603 abr 10  2014 blacklist.conf
-rw-r--r--  1 root root  210 abr 10  2014 blacklist-firewire.conf
-rw-r--r--  1 root root  677 abr 10  2014 blacklist-framebuffer.conf
-rw-r--r--  1 root root  583 abr 10  2014 blacklist-rare-network.conf
-rw-r--r--  1 root root 1077 abr 10  2014 blacklist-watchdog.conf
-rw-r--r--  1 root root  456 abr 14  2014 fbdev-blacklist.conf
-rw-r--r--  1 root root  347 abr 10  2014 iwlmwifi.conf
-rw-r--r--  1 root root  104 abr 10  2014 mlx4.conf
```

## Módulos (III)

- ★ Para indicar los valores de los parámetros de un módulo usamos la opción **options**:

```
[root@localhost /lib/modules]# cat /etc/modprobe.d/v4l2loopback.conf
options v4l2loopback devices=1 exclusive_caps=1 video_nr=2 card_label="v4l2loopback"
```

- ★ También podemos usar la opción **install** y/o **remove** para indicar los comando que se ejecutarán al cargar y/o descargar un módulo:

```
[root@localhost /lib/modules]# cat /etc/modprobe.d/nfsd.conf
install nfsd /sbin/modprobe --first-time --ignore-install nfsd && { /bin/mount -t
nfsd /proc/fs/nfsd > /dev/null 2>&1 || ;;}
remove nfsd { /bin/umount /proc/fs/nfsd > /dev/null 2>&1 || ;; } ;
/sbin/modprobe -r --first-time --ignore-remove nfsd
```

## Módulos (IV)

- ★ En ocasiones, versiones muy antiguas, la configuración de los módulos la encontramos en el fichero `/etc/modprobe.conf`.
- ★ Este fichero se genera a partir de `/etc/modprobe.d` usando el comando `modules-update`

## Módulos (V)

- ★ También podemos usar la opción **alias** para crear un nick asociado al módulo:

```
[root@localhost /lib/modules]# cat /etc/modprobe.d/ne2k-pci.conf
alias eth0 ne2k-pci
options eth0 irq=10
```

- ★ Que un módulo se encuentre en **/etc/modprobe.d** no quiere indicar que se cargue al arrancar. Si queremos esto, tenemos que crearlo en **/etc/modules-load.d**

```
[root@localhost /lib/modules]# ls /etc/modules-load.d/
akvcam.conf  v4l2loopback.conf
```



## Core (I)

- ★ Al núcleo se le pasan parámetros en el momento de la carga gracias al *boot loader*.
- ★ Algunos parámetros se pueden modificar dinámicamente, estos valores se pueden consultar mediante el comando `sysctl` y la opción `-a`:

```
[root@localhost /lib/modules]# sysctl -a | more
abi.vsyscall32 = 1
debug.exception-trace = 1
dev.hpet.max-user-freq = 64
dev.i915.oa_max_sample_rate = 100000
dev.i915.perf_stream_paranoid = 1
dev.raid.speed_limit_max = 200000
dev.raid.speed_limit_min = 1000
```



## Core (II)

- ★ Para modificar temporalmente el valor de alguno de ellos, usamos la opción **-w** del comando **sysctl**:

```
[root@localhost /lib/modules]# sysctl -w kernel.hostname=miequipo
kernel.hostname=miequipo
```

- ★ Si queremos que estos cambios sean permanentes, tenemos que añadirlos al fichero **/etc/sysctl.conf**:

```
[root@localhost /lib/modules]# echo "kernel.hostname=miequipo" >> /etc/sysctl.conf
```

- ★ O crear un fichero, **.conf**, en el directorio **/etc/sysctl.d**:

```
[root@localhost /lib/modules]# ls /etc/sysctl.d
40-max-user-watches.conf
```

## Core (III)

- ★ Podemos cargar la configuración de `/etc/sysctl.conf` con la opción `-p` del comando `sysctl`:

```
[root@localhost /lib/modules]# sysctl -p
```

# Información del hardware

---

4

---

# Introducción

- ★ Existen distintos comandos que nos permiten obtener información sobre el hardware específico de una máquina.
- ★ Algunos de los más conocidos son:
  - `lspci`
  - `lsdev`
  - `lsusb`
  - `lshw`
- ★ Estas herramientas acceden a los dispositivos gracias a los ficheros de dispositivo gestionados por udev.

# lspci

- ★ El comando **lspci** muestra información sobre los buses PCI del sistema y los dispositivos conectados a ellos:

```
[root@localhost /lib/modules]# lspci
04:00.0 Ethernet controller: Intel Corporation I211 Gigabit Network Connection (rev 03)
05:00.0 Network controller: Intel Corporation Wi-Fi 6 AX200 (rev 1a)
09:00.0 VGA compatible controller: Advanced Micro Devices, Inc. [AMD/ATI] Baffin
[Radeon RX 550 640SP / RX 560/560X] (rev cf)
09:00.1 Audio device: Advanced Micro Devices, Inc. [AMD/ATI] Baffin HDMI/DP Audio
[Radeon RX 550 640SP / RX 560/560X]
0b:00.4 Audio device: Advanced Micro Devices, Inc. [AMD] Starship/Matisse HD Audio
Controller
0c:00.0 RAID bus controller: Advanced Micro Devices, Inc. [AMD] Device 7916 (rev 51)
```

## lspci (II)

- ★ Gracias a las opciones **-v**, **-vv**, **-vvv** obtenemos más información:

```
[root@localhost /lib/modules]# lspci -v
04:00.0 Ethernet controller: Intel Corporation I211 Gigabit Network Connection (rev
03)

Subsystem: Gigabyte Technology Co., Ltd Device e000
Flags: bus master, fast devsel, latency 0, IRQ 98
Memory at fc800000 (32-bit, non-prefetchable) [size=128K]
I/O ports at f000 [size=32]
Memory at fc820000 (32-bit, non-prefetchable) [size=16K]
Capabilities: [40] Power Management version 3
[...]
Kernel driver in use: igb
Kernel modules: igb
```



## lspci (III)

- ★ La información relacionada con los módulos usados por dispositivo también la obtenemos con **-k**

```
[root@localhost /lib/modules]# lspci -k
0b:00.3 USB controller: Advanced Micro Devices, Inc. [AMD] Matisse USB 3.0 Host
Controller
    Subsystem: Gigabyte Technology Co., Ltd Device 5007
    Kernel driver in use: xhci_hcd
0b:00.4 Audio device: Advanced Micro Devices, Inc. [AMD] Starship/Matisse HD Audio
Controller
    Subsystem: Gigabyte Technology Co., Ltd Device a0c3
    Kernel driver in use: snd_hda_intel
    Kernel modules: snd_hda_intel
```

# lsdev

- ★ El comando **lsdev** obtiene información sobre interrupciones, puertos de entrada salida y dma

```
[root@localhost /lib/modules]# lsdev
Device          DMA    IRQ    I/O Ports
-----
0000:00:1d.0           d800-d81f
0000:00:1d.1           d000-d01f
0000:00:1d.2           d400-d41f
0000:00:1f.0           4000-407f 4080-40bf
0000:00:1f.1           f000-f00f
0000:00:1f.3           5000-501f
0000:00:1f.5           e000-e0ff e400-e43f
0000:01:05.0           c000-c0ff
```

## Lsdev (II)

- ★ El comando **lsdev** obtiene esta información desde `/proc/interrupts`, `/proc/dma/` y `/proc/ioports`.

# lsusb

- ★ El comando **lsusb** muestra información sobre los buses USB y los dispositivos conectados:

```
[root@localhost /lib/modules]# lsusb
Bus 006 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 005 Device 004: ID 0424:274c Microchip Technology, Inc. (formerly SMSC)
Bus 005 Device 003: ID 046d:082d Logitech, Inc. HD Pro Webcam C920
Bus 005 Device 002: ID 0424:2734 Microchip Technology, Inc. (formerly SMSC) USB2734
Bus 005 Device 010: ID 03f0:042a HP, Inc LaserJet M1132 MFP
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 003: ID 0d8c:0014 C-Media Electronics, Inc. Audio Adapter (Unitek
Y-247A)
```

## Lsusb (II)

- ★ Podemos usar las opciones -v para obtener información detallada y -d para indicar el dispositivo

```
[root@localhost /lib/modules]# lsusb -v -d 046d:082d
Bus 005 Device 003: ID 046d:082d Logitech, Inc. HD Pro Webcam C920
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB                 2.00
  bDeviceClass           239 Miscellaneous Device
  bDeviceSubClass         2
  bDeviceProtocol         1 Interface Association
  bMaxPacketSize0         64
```

# Udev

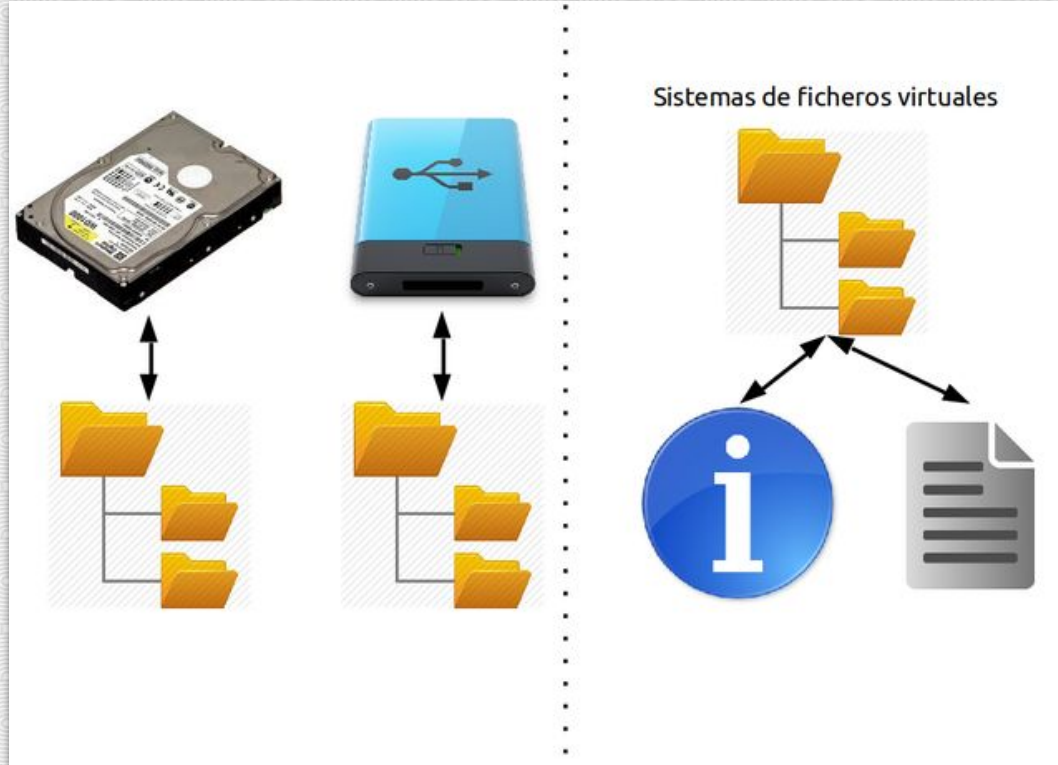
---

5

---

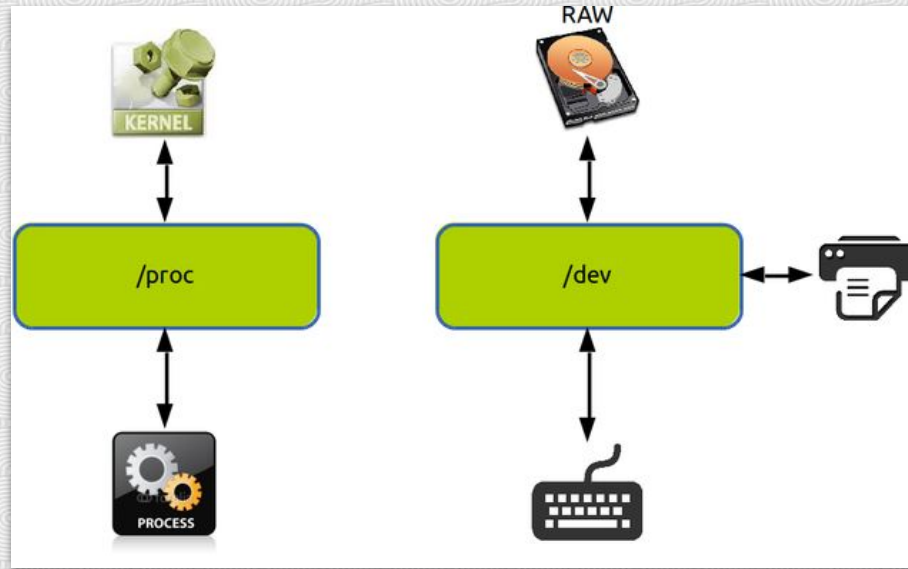


# Sistemas de ficheros virtuales



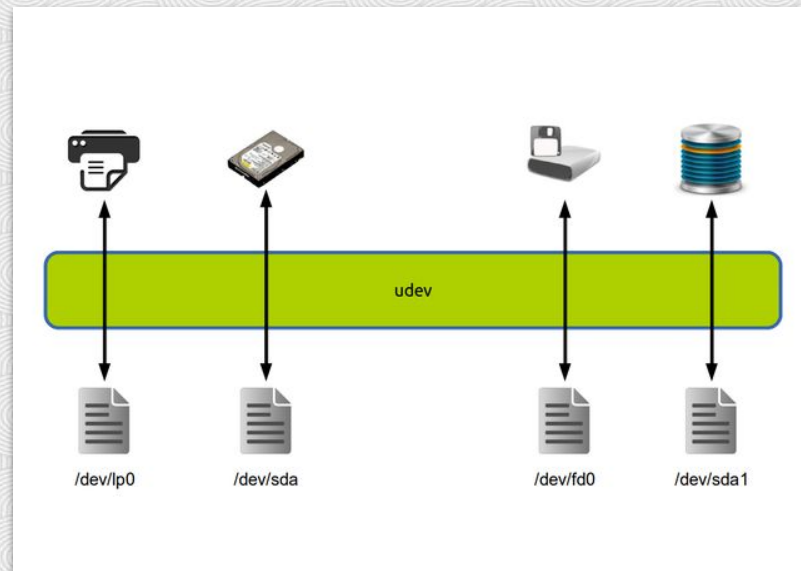
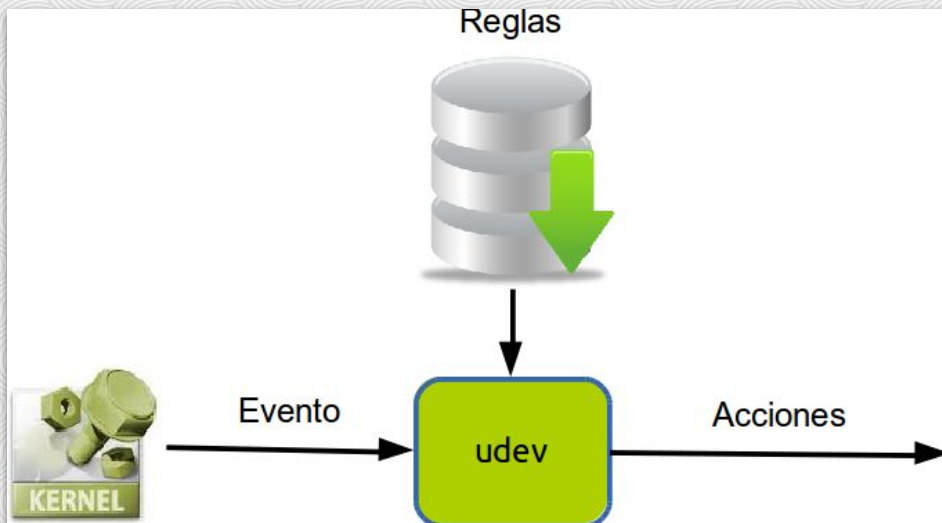
## Sistemas de ficheros virtuales (II)

- ★ El directorio `/dev` contiene ficheros de dispositivos. Antes tenía una gran cantidad de ficheros, ahora sólo los de los dispositivos conectados.



## ¿Qué es udev?

- ★ El denominado **udev** trabaja junto al núcleo para crear estos dispositivos



# Ficheros de dispositivos

Fichero	Dispositivo
<code>/dev/sdA</code>	Un disco duro completo (SCSI o SATA)
<code>/dev/hdA</code>	Disco duro o disco óptico (IDE)
<code>/dev/sdaA#</code>	Una partición de un disco duro SCSI.
<code>/dev/hdaA#</code>	Una partición de un disco duro IDE
<code>/dev/fd#</code>	Disquete flexible
<code>/dev/sr#</code>	Disco óptico SCSI
<code>/dev/lp#</code>	Puerto paralelo

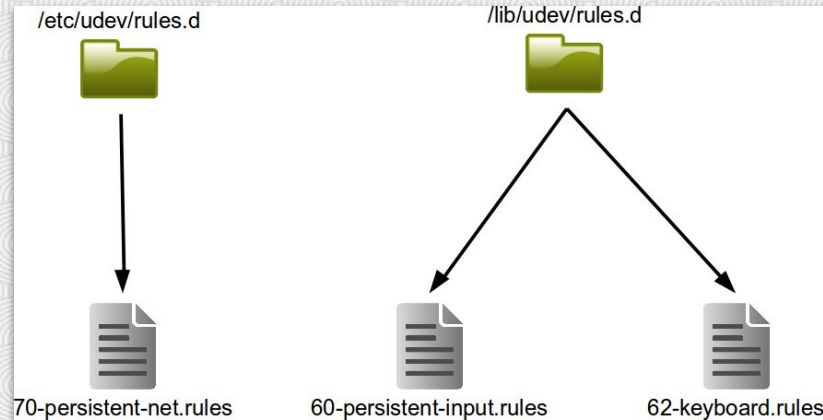
# Ficheros de dispositivos

Fichero	Dispositivo
<code>/dev/usb/lp#</code>	Impresora usb
<code>/dev/ttyS#</code>	Puerto serie
<code>/dev/tty#</code>	Consola de login (texto)
<code>/dev/pts/#</code>	Sesión en modo texto (Consola X)
<code>/dev/bus/usb/</code>	Acceso a dispositivos usb
<code>/dev/snd/</code>	Hardware de sonido
<code>/dev/input/*</code>	Dispositivos de entrada ( <code>/dev/input/mice</code> , <code>/dev/input/mouse#</code> ).



# Reglas

- ★ Las reglas le indican a **udev** lo que hacer teniendo en cuenta el evento generado.
- ★ Estas reglas son ficheros de texto (**.rules**) que se almacenan en **/etc/udev/rules.d** o **/lib/udev/rules.d** y tiene un orden de ejecución.





## Ejemplo

- ★ Las reglas contienen **acciones/condiciones**, **operadores**, **claves** y **valores**.

```
SUBSYSTEM!="usb_device", ACTION!="add", GOTO="minolta_rules_end"  
# Minolta|DiMAGE Scan Elite 5400  
ATTR{idVendor}=="0686", ATTR{idProduct}=="400e", SYMLINK+="scan5400"  
MODE="0660", OWNER="lisa", GROUP="scanner"  
LABEL="minolta_rules_end"  
# PCI device 0x8086:0x1030 (e100)  
SUBSYSTEM=="net", ACTION=="add", ATTR{address}=="00:03:47:b1:e3:d8", \  
KERNEL=="eth*", NAME="eth0"
```

## Ejemplo (II)



### Condición/Acción

```
KERNEL=="sd*[0-9]|dasd*[0-9]", ENV{ID_SERIAL}=="?*", \
SYMLINK+="disk/by-id/${env{ID_BUS}-${env{ID_SERIAL}}-part%n"
```

### Operadores

```
KERNEL==sd*[0-9]|dasd*[0-9]", ENV{ID_SERIAL}==?*", \
SYMLINK+=disk/by-id/${env{ID_BUS}-${env{ID_SERIAL}}-part%n"
```

### Claves

```
KERNEL=="sd*[0-9]|dasd*[0-9]", ENV{ID_SERIAL}=="?*", \
SYMLINK+="disk/by-id/${env{ID_BUS}-${env{ID_SERIAL}}-part%n"
```

### Valores

```
KERNEL=="sd*[0-9]|dasd*[0-9]", ENV{ID_SERIAL}=="?*" \
SYMLINK+="disk/by-id/${env{ID_BUS}-${env{ID_SERIAL}}-part%n"
```

# Operadores

Clave	Tipo	Descripción
<b>==</b>	Matching	Igualdad
<b>!=</b>	Matching	Distinto
<b>=</b>	Assignment	Asignación
<b>+=</b>	Assignment	Añadir un valor a una clave
<b>:=</b>	Assignment	Añadir un valor a una clave y evitar futuros cambios

```
SUBSYSTEM!="usb_device", ACTION!="add", GOTO="minolta_rules_end"  
# Minolta|DiMAGE Scan Elite 5400  
ATTR{idVendor}=="0686", ATTR{idProduct}=="400e", SYMLINK+="scan5400"
```

# Claves de control

Clave	Descripción
<b>LABEL</b>	Etiqueta para gestionar un flujo
<b>GOTO</b>	Redireccionar el flujo a una determinada etiqueta
<b>WAIT_FOR</b>	Espera hasta que un determinado fichero se cree

```
SUBSYSTEM!="usb_device", ACTION!="add", GOTO="minolta_rules_end"  
# Minolta|DiMAGE Scan Elite 5400  
ATTR{idVendor}=="0686", ATTR{idProduct}=="400e", SYMLINK+="scan5400"  
MODE="0660", OWNER="lisa", GROUP="scanner"  
LABEL="minolta_rules_end"
```

# Claves de comparación

Clave	Descripción
<b>ACTION</b>	Nombre acción tomada por udev
<b>KERNEL</b>	Nombre dado por el kernel para el dispositivo
<b>DEVPATH</b>	Camino dado por el kernel para el dispositivo
<b>SUBSYSTEM</b>	Nombre dado por el kernel para el subsistema
<b>DRIVER</b>	Nombre del driver del dispositivo
<b>PROGRAM</b>	Ejecuta programa externo
<b>RESULT</b>	Salida de la última invocación a PROGRAM
<b>ATTR{string}</b>	Nombre de atributo asociado a un dispositivo



# Claves de asignación

Clave	Descripción
NAME	Nombre del fichero de dispositivo a crear
SYMLINK	Nombre del enlace simbólico a crear
OWNER	Propietario del fichero de dispositivo
GROUP	Grupo propietario del fichero de dispositivo
MODE	Permisos del fichero de dispositivo
RUN	Nombre de un programa externo (relativo a /lib/udev) a ejecutar
OPTIONS	Establece opciones específicas del dispositivo
ATTR{string}	Nombre de atributo específico del dispositivo



## udevadm

- ★ Para gestionar udevd podemos usar la utilidad **udevadm**, podemos consultar las acciones que realiza **udev** ejecutando **udevadm monitor**

```
[root@localhost /lib/modules]# udevadm monitor
monitor will print the received events for:
UDEV - the event which udev sends out after rule processing
KERNEL - the kernel uevent

KERNEL[64334.399863] add
/devices/pci0000:00/0000:00:08.1/0000:0b:00.3/usb5/5-4/5-4.3 (usb)
KERNEL[64334.401670] bind
/devices/pci0000:00/0000:00:08.1/0000:0b:00.3/usb5/5-4/5-4.3 (usb)
UDEV [64334.401890] add /devices/virtual/workqueue/scsi_tmf_9 (workqueue)
```

## udevadm (II)

- ★ Con **udevadm**, también podemos consultar los valores asignables a las claves:

```
[root@localhost /lib/modules]# udevadm info -a -p $(udevadm info -q path -n  
/dev/input/mouse1)  
looking at device '/class/input/input6/mouse1':  
KERNEL=="mouse1"  
SUBSYSTEM=="input"  
DRIVER==""  
looking at parent device '/class/input/input6':  
KERNELS=="input6"  
SUBSYSTEMS=="input"  
ATTRS{name}=="Logitech Trackball"  
ATTRS{phys}=="usb-0000:00:13.0-3/input0"
```

# dmesg

- ★ Los eventos del núcleo registrados por **udevadm** también pueden ser consultados en el fichero **/var/log/messages** o usando el comando **dmesg**

```
[root@localhost /lib/modules]# dmesg
[64331.517104] usb 5-4.3: new high-speed USB device number 11 using xhci_hcd
[64331.612678] usb 5-4.3: New USB device found, idVendor=090c, idProduct=1000,
bcdDevice=11.00
[64331.613000] usb 5-4.3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[64331.613285] usb 5-4.3: Product: USB Flash Disk
[64331.613462] usb 5-4.3: Manufacturer: General
[64331.613630] usb 5-4.3: SerialNumber: 0362216120011545
[64331.614823] usb-storage 5-4.3:1.0: USB Mass Storage device detected
[64331.615181] usb-storage 5-4.3:1.0: Quirks match for vid 090c pid 1000: 400
```