

2.6- EXPLORANDO EL KERNEL DE LINUX (LINUX)

Cristóbal Suárez Abad

ADMINISTRACIÓN DE SISTEMAS OPERATIVOS - 2º ASIR

Índice

| | | |
|----|--|----|
| 1) | Comprobación inicial de módulos | 3 |
| a) | Lista todos los módulos cargados actualmente en tu equipo. | 3 |
| b) | Cuenta cuántos módulos hay disponibles en el núcleo que estás usando. | 3 |
| c) | Anota la versión exacta del kernel..... | 3 |
| 2) | Observación del sistema ante nuevos dispositivos | 4 |
| a) | Conecta un lápiz USB (o simula el evento montando un dispositivo de bloque) (sudo modprobe usb/usbcore/loop_storage -- sudo udevadm trigger). | 4 |
| b) | Observa los mensajes generados por el sistema utilizando dmesg. | 4 |
| c) | Identifica qué módulos o controladores se han cargado automáticamente.... | 5 |
| 3) | Gestión de módulos..... | 6 |
| a) | Elimina un módulo correspondiente a un dispositivo no esencial y comprueba qué ocurre..... | 6 |
| b) | Vuelve a cargarlo correctamente..... | 6 |
| c) | Registra en un log los comandos usados y los mensajes obtenidos. | 6 |
| 4) | Configuración del arranque | 7 |
| a) | Selecciona un módulo que se cargue automáticamente en tu equipo..... | 7 |
| b) | Configura el sistema para que no se cargue al iniciar | 7 |
| c) | Reinicia y verifica que la configuración se mantiene..... | 7 |
| 5) | Gestión del módulo loop | 8 |
| a) | Carga el módulo loop..... | 8 |
| b) | Consulta su información y averigua para qué se utiliza. | 8 |
| c) | Lista el contenido del directorio /sys/module/loop/parameters/. | 8 |
| d) | Configura el sistema para permitir un máximo de 12 dispositivos loop en el próximo arranque. | 8 |
| 6) | Parámetros del sistema | 9 |
| a) | Cambia provisionalmente la política de swappiness para que la swap solo se active cuando la RAM supere el 90% de uso | 9 |
| b) | Haz que el cambio sea permanente tras el reinicio..... | 9 |
| 7) | IPv6 y reenvío de paquetes | 10 |
| a) | Muestra el valor actual del bit de <i>forwarding</i> para IPv6..... | 10 |
| b) | Interpreta su significado y posibles implicaciones. | 10 |
| 7) | Compilación dinámica con DKMS..... | 11 |

| | |
|---|----|
| a) Instala y compila el módulo OpenZFS mediante DKMS | 11 |
| b) Comprueba que el módulo se carga correctamente..... | 11 |
| (Opcional) Añade un disco nuevo y crea un sistema de archivos ZFS para comprobar su funcionamiento..... | 12 |
| Comandos extras: | 13 |

1) Comprobación inicial de módulos

- a) Lista todos los módulos cargados actualmente en tu equipo.

`lsmod`

```
[root@server2asir usuario]$ lsmod
Module           Size  Used by
tls              114688  0
dummy            16384   0
binfmt_misc     24576   1
joydev           32768   0
input_leds       16384   0
serio_raw        20480   0
mac_hid          16384   0
qemu_fw_cfg      20480   0
sch_fq_codel    20480   2
dm_multipath    40960   0
scsi_dh_rdac    20480   0
scsi_dh_emc     16384   0
scsi_dh_alua    20480   0
msr              16384   0
efi_pstore       16384   0
ip_tables        32768   0
x_tables         53248   1 ip_tables
autofs4          49152   2
btrfs             1564672  0
```

- b) Cuenta cuántos módulos hay disponibles en el núcleo que estás usando.

`find /lib/modules/$(uname -r)/kernel/ -type f -name "*.ko*" | wc -l`

```
[root@server2asir usuario]$ find /lib/modules/$(uname -r)/kernel/ -type f -name "*.ko*" | wc -l
6025
Tu Nombre: martes 25 noviembre 2025 11:12
```

- c) Anota la versión exacta del kernel.

`uname -r`

```
[root@server2asir usuario]$ uname -r
5.15.0-161-generic
Tu Nombre: martes 25 noviembre 2025
```

2) Observación del sistema ante nuevos dispositivos

- a) Conecta un lápiz USB (o simula el evento montando un dispositivo de bloque) (sudo modprobe usb/usbcore/loop_storage -- sudo udevadm trigger).

sudo modprobe usbcore

sudo udevadm trigger

```
[root@server2asir usuario]$sudo modprobe usbcore
Tu Nombre martes 25 noviembre 2025 11:14
[root@server2asir usuario]$sudo udevadm trigger
Tu Nombre martes 25 noviembre 2025 11:14
```

- b) Observa los mensajes generados por el sistema utilizando dmesg.

dmesg -w

```
[root@server2asir usuario]$dmesg -w
[    0.000000] Linux version 5.15.0-161-generic (buildd@lcy02-amd64-072) (gcc (Ubuntu 11.4.0-1ubuntu1~2
2.04.2) 11.4.0, GNU ld (GNU Binutils for Ubuntu) 2.38) #171-Ubuntu SMP Sat Oct 11 08:17:01 UTC 2025 (Ub
untu 5.15.0-161.171-generic 5.15.189)
[    0.000000] Command line: BOOT_IMAGE=/vmlinuz-5.15.0-161-generic root=/dev/mapper/ubuntu--vg-ubuntu-
-lv ro
[    0.000000] KERNEL supported cpus:
[    0.000000]   Intel GenuineIntel
[    0.000000]   AMD AuthenticAMD
[    0.000000]   Hygon HygonGenuine
[    0.000000]   Centaur CentaurHauls
[    0.000000]   zhaoxin Shanghai
[    0.000000] BIOS-provided physical RAM map:
[    0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
[    0.000000] BIOS-e820: [mem 0x0000000000009fc00-0x000000000009ffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000f0000-0x00000000000ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000000100000-0x000000000bffd9fff] usable
[    0.000000] BIOS-e820: [mem 0x000000000bf000-0x000000000bfffffff] reserved
[    0.000000] BIOS-e820: [mem 0x000000000feffc000-0x000000000fefffff] reserved
[    0.000000] BIOS-e820: [mem 0x000000000ffffc0000-0x000000000fffffff] reserved
[    0.000000] NX (Execute Disable) protection: active
[    0.000000] SMBIOS 2.8 present.
[    0.000000] DMI: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qe
mu.org 04/01/2014
[    0.000000] Hypervisor detected: KVM
[    0.000000] kvm-clock: Using msrs 4b564d01 and 4b564d00
[    0.000000] kvm-clock: cpu 0, msr b1601001, primary cpu clock
[    0.000003] kvm-clock: using sched offset of 10744438797 cycles
[    0.000007] clocksource: kvm-clock: mask: 0xffffffffffffffff max_cycles: 0x1cd42e4dff, max_idle_ns:
881590591483 ns
```

- c) Identifica qué módulos o controladores se han cargado automáticamente.

```
dmesg | grep -i "usb\|loop\|storage\|driver"
```

```
[root@server2asir usuario]$dmesg | grep -i "usb\|loop\|storage\|driver"
[ 0.126185] Calibrating delay loop (skipped) preset value.. 4588.49 BogoMIPS (lpj=9176992)
[ 0.272476] Performance Events: unsupported Netburst CPU model 107 no PMU driver, software events on
ly.
[ 0.318723] aciphp: ACPI Hot Plug PCI Controller Driver version: 0.5
[ 0.434685] ACPI: bus type USB registered
[ 0.434685] usbcore: registered new interface driver usbfs
[ 0.434685] usbcore: registered new interface driver hub
[ 0.434685] usbcore: registered new device driver usb
[ 0.501465] Block layer SCSI generic (bsg) driver version 0.4 loaded (major 243)
[ 0.503519] shpchp: Standard Hot Plug PCI Controller Driver version: 0.4
[ 0.514910] Serial: 8250/16550 driver, 32 ports, IRQ sharing enabled
[ 0.522821] loop: module loaded
[ 0.526518] tun: Universal TUN/TAP device driver, 1.6
[ 0.526644] PPP generic driver version 2.4.2
[ 0.526833] VFIO - User Level meta-driver version: 0.3
```

3) Gestión de módulos

- a) Elimina un módulo correspondiente a un dispositivo **no esencial** y comprueba qué ocurre.

```
sudo rmmod pcspkr
```

```
lsmod | grep pcspkr # Comprueba si ha desaparecido
```

```
Tu Nombre martes 25 noviembre 2025 11:22
[root@server2asir usuario]$sudo rmmod pcspkr
rmmod: ERROR: Module pcspkr is not currently loaded
Tu Nombre martes 25 noviembre 2025 11:22
[root@server2asir usuario]$lsmod | grep pcspkr
Tu Nombre martes 25 noviembre 2025 11:23
[root@server2asir usuario]$sudo modprobe pcspkr
```

- b) Vuelve a cargarlo correctamente.

```
sudo modprobe pcspkr
```

```
lsmod | grep pcspkr # Comprueba que se ha cargado
```

```
Tu Nombre martes 25 noviembre 2025 11:23
[root@server2asir usuario]$sudo modprobe pcspkr
Tu Nombre martes 25 noviembre 2025 11:23
[root@server2asir usuario]$lsmod | grep pcspkr
pcspkr           16384  0
Tu Nombre martes 25 noviembre 2025 11:23
```

- c) Registra en un log los comandos usados y los mensajes obtenidos.

```
echo "## Log de Gestión de Módulos ##" >> modulo_log.txt
date >> modulo_log.txt
echo "Comando: sudo rmmod pcspkr" >> modulo_log.txt
sudo rmmod pcspkr 2>&1 | tee -a modulo_log.txt # 2>&1 redirige errores
echo "Estado tras rmmod:" >> modulo_log.txt
lsmod | grep pcspkr >> modulo_log.txt
echo "Comando: sudo modprobe pcspkr" >> modulo_log.txt
sudo modprobe pcspkr 2>&1 | tee -a modulo_log.txt
echo "Estado tras modprobe:" >> modulo_log.txt
lsmod | grep pcspkr >> modulo_log.txt
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]
```

4) Configuración del arranque

- Selecciona un módulo que se cargue automáticamente en tu equipo.

Los ejemplos más comunes son: loop, lp, pcspkr, floppy y usb_storage.

- Configura el sistema para que **no se cargue al iniciar**.

Actualmente está cargado:

```
Tu Nombre martes 25 noviembre 2025 11:29
[root@server2asir usuario]$lsmod | grep pcspkr
pcspkr                16384  0
Tu Nombre martes 25 noviembre 2025 11:30
[...]
```

Para evitar que se cargue, lo podemos "poner en lista negra" (blacklist):

```
echo "blacklist pcspkr" | sudo tee /etc/modprobe.d/blacklist-pcspkr.conf
```

```
Tu Nombre martes 25 noviembre 2025 11:29
[root@server2asir usuario]$echo "blacklist pcspkr" | sudo tee /etc/modprobe.d/blacklist-pcspkr.conf
blacklist pcspkr
Tu Nombre martes 25 noviembre 2025 11:30
[...]
```

- Reinicia y verifica que la configuración se mantiene.

```
reboot now
```

Una vez reiniciada la máquina, podemos comprobar que no se carga:

```
lsmod | grep pcspkr
```

```
Tu Nombre martes 25 noviembre 2025 11:33
[usuario@server2asir ~]$lsmod | grep pcspkr
Tu Nombre martes 25 noviembre 2025 11:33
[usuario@server2asir ~]$
```

Si queremos deshacer el cambio, solo tenemos que eliminar el archivo anterior:

```
sudo rm /etc/modprobe.d/blacklist-pcspkr.conf
```

```
Tu Nombre martes 25 noviembre 2025 11:33
[usuario@server2asir ~]$sudo rm /etc/modprobe.d/blacklist-pcspkr.conf
[sudo] password for usuario:
Tu Nombre martes 25 noviembre 2025 11:34
[usuario@server2asir ~]$
```

5) Gestión del módulo loop

a) Carga el módulo loop

sudo modprobe loop

b) Consulta su información y averigua para qué se utiliza.

modinfo loop

```
[usuario@server2asir ~]$modinfo loop
name:          loop
filename:      (builtin)
alias:         devname:loop-control
alias:         char-major-10-237
alias:         block-major-7-*
license:       GPL
file:          drivers/block/loop
parm:          max_loop:Maximum number of loop devices
parm:          max_part:Maximum number of partitions per loop device (int)
```

c) Lista el contenido del directorio /sys/module/loop/parameters/.

ls -l /sys/module/loop/parameters/

```
[usuario@server2asir ~]$ls -l /sys/module/loop/parameters/
total 0
-r--r--r-- 1 root root 4096 nov 25 11:41 max_loop
-r--r--r-- 1 root root 4096 nov 25 11:41 max_part
Tu Nombre martes 25 noviembre 2025 11:41
```

d) Configura el sistema para permitir un máximo de **12 dispositivos loop** en el próximo arranque.

echo "options loop max_loop=12" | sudo tee /etc/modprobe.d/loop-config.conf

```
Tu Nombre martes 25 noviembre 2025 11:41
[usuario@server2asir ~]$echo "options loop max_loop=12" | sudo tee /etc/modprobe.d/loop-config.conf
options loop max_loop=12
Tu Nombre martes 25 noviembre 2025 11:42
```

6) Parámetros del sistema

- a) Cambia **provisionalmente** la política de swappiness para que la swap solo se active cuando la RAM supere el **90% de uso**.

El valor predeterminado de *swappiness* suele ser 60.

```
sysctl vm.swappiness
```

```
[Tu Nombre martes 25 noviembre 2025 11:42]
[usuario@server2asir ~]$sysctl vm.swappiness
vm.swappiness = 60
[Tu Nombre martes 25 noviembre 2025 11:42]
```

Para que solo se use *swap* al 90% de uso de RAM, debes reducir el valor a **10**.

```
sudo sysctl vm.swappiness=10
```

```
Tu Nombre martes 25 noviembre 2025 11:43
[usuario@server2asir ~]$sudo sysctl vm.swappiness=10
vm.swappiness = 10
Tu Nombre martes 25 noviembre 2025 11:44
[usuario@server2asir ~]$sysctl vm.swappiness
vm.swappiness = 10
Tu Nombre martes 25 noviembre 2025 11:44
```

- b) Haz que el cambio sea **permanente** tras el reinicio.

```
echo "vm.swappiness = 10" | sudo tee -a /etc/sysctl.conf
```

```
[Tu Nombre martes 25 noviembre 2025 11:46]
[root@server2asir usuario]$echo "vm.swappiness = 10" | sudo tee -a /etc/sysctl.conf
vm.swappiness = 10
[Tu Nombre martes 25 noviembre 2025 11:46]
```

Para aplicar los cambios sin reiniciar:

```
sudo sysctl -p
```

```
[root@server2asir usuario]$sudo sysctl -p
vm.swappiness = 10
[Tu Nombre martes 25 noviembre 2025 11:46]
```

7) IPv6 y reenvío de paquetes

- a) Muestra el valor actual del bit de *forwarding* para IPv6.

```
sysctl net.ipv6.conf.all.forwarding
```

```
[root@server2asir usuario]$sysctl net.ipv6.conf.all.forwarding
net.ipv6.conf.all.forwarding = 0
```

- b) Interpreta su significado y posibles implicaciones.

Valor 0: El reenvío (*forwarding*) está **deshabilitado**. El sistema no actuará como un *router* para paquetes IPv6, sino que solo procesará paquetes destinados a sus propias interfaces. Esta es la configuración por defecto para un *host* normal.

Valor 1: El reenvío está **habilitado**. El sistema reenviará paquetes IPv6 que lleguen a una interfaz y estén destinados a otra red/interfaz.

Implicaciones:

Si es **0**: Es más seguro para un *host* final, ya que no reenvía tráfico no solicitado.

Si es **1**: Es **necesario** si el servidor actúa como **router IPv6** entre diferentes redes o interfaces.

7) Compilación dinámica con DKMS

a) Instala y compila el módulo **OpenZFS** mediante **DKMS**.

`sudo apt update`

`sudo apt install zfsutils-linux dkms build-essential linux-headers-$(uname -r)`

```
[root@server2asir usuario]$ sudo apt install zfsutils-linux dkms build-essential linux-headers-$(uname -r)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
linux-headers-5.15.0-161-generic ya está en su versión más reciente (5.15.0-161.171).
fijado linux-headers-5.15.0-161-generic como instalado manualmente.
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  libpcp-pmda-perl libpcp-trace2 libpcp-webl libpfm4 pcp
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  bzip2 cpp cpp-11 cpp-12 dctrl-tools dpkg-dev fakeroot fontconfig-config fonts-dejavu-core g++
  g++-11 gcc gcc-11 gcc-11-base gcc-12 libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libasan8 libatomic1 libc-dev-bin libc-devtools libc6-dev libcc1-0
  libcrypt-dev libdeflate0 libdpkg-perl libfakeroot libfile-fcntllock-perl libfontconfig1
  libgcc-11-dev libgcc-12-dev libgd3 libgomp1 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8
  liblsan0 libmpc3 libnsl-dev libnvpair3linux libquadmath0 libstdc++-11-dev libtiff5 libtirpc-dev
  libtsan0 libtsan2 libubsan1 libutil3linux libwebp7 libxpm4 libzfs4linux libzpool5linux
  linux-libc-dev lto-disabled-list make manpages-dev rpcsvc-proto zfs-zed
Paquetes sugeridos:
  bzip2-doc cpp-doc gcc-11-locales gcc-12-locales cpp-12-doc debtags menu debian-keyring g++-multilib
  g++-11-multilib gcc-11-doc gcc-multilib autoconf automake libtool flex bison gdb gcc-doc
  gcc-11-multilib gcc-12-multilib gcc-12-doc glibc-doc bzr libgd-tools libstdc++-11-doc make-doc
  nfs-kernel-server samba-common-bin zfs-initramfs | zfs-dracut
Se instalarán los siguientes paquetes NUEVOS:
  build-essential bzip2 cpp cpp-11 dctrl-tools dkms dpkg-dev fakeroot fontconfig-config
  fonts-dejavu-core g++ g++-11 gcc gcc-11 gcc-11-base gcc-12 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libasan8 libatomic1 libc-dev-bin
  libc-devtools libc6-dev libcc1-0 libcrypt-dev libdeflate0 libdpkg-perl libfakeroot
  libfile-fcntllock-perl libfontconfig1 libgcc-11-dev libgcc-12-dev libgd3 libgomp1 libisl23 libitm1
  libjbig0 libjpeg-turbo8 libjpeg8 liblsan0 libmpc3 libnsl-dev libnvpair3linux libquadmath0
  libstdc++-11-dev libtiff5 libtirpc-dev libtsan0 libtsan2 libubsan1 libutil3linux libwebp7 libxpm4
  libzfs4linux libzpool5linux linux-libc-dev lto-disabled-list make manpages-dev rpcsvc-proto zfs-zed
  zfsutils-linux
0 actualizados, 65 nuevos se instalarán, 0 para eliminar y 87 no actualizados.
Se necesita descargar 106 MB de archivos.
Se utilizarán 354 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] |
```

b) Comprueba que el módulo se carga correctamente.

`lsmod | grep zfs`

```
Tu Nombre martes 25 noviembre 2025 15:15
[root@server2asir usuario]$ lsmod | grep zfs
zfs                      4112384  6
zunicode                  348160  1 zfs
zzstd                     491520  1 zfs
zlua                      163840  1 zfs
zavl                      16384   1 zfs
icp                       311296  1 zfs
zcommon                   106496  2 zfs,icp
znvpair                   98304   2 zfs,zcommon
spl                      118784   6 zfs,icp,zzstd,znvpair,zcommon,zavl
Tu Nombre martes 25 noviembre 2025 15:15
[root@server2asir usuario]$
```

(Opcional) Añade un disco nuevo y crea un sistema de archivos ZFS para comprobar su funcionamiento.

Antes de nada. Apagamos la VM y le añadimos un nuevo disco virtual.

1. Crear un pool ZFS simple llamado 'mypool'

```
sudo zpool create mypool /dev/sdb
```

2. Crear un nuevo sistema de archivos dentro del pool

```
sudo zfs create mypool/data
```

3. Comprobar que se ha montado (por defecto en /mypool/data)

```
df -h | grep mypool
```

```
[Tu Nombre martes 25 noviembre 2025 15:25
[root@server2asir usuario]$sudo zpool create mypool /dev/sdb
[Tu Nombre martes 25 noviembre 2025 15:32
[root@server2asir usuario]$sudo zfs create mypool/data
[Tu Nombre martes 25 noviembre 2025 15:32
[root@server2asir usuario]$df -h | grep mypool
mypool                                9,3G  128K  9,3G   1% /mypool
mypool/data                            9,3G  128K  9,3G   1% /mypool/data
[Tu Nombre martes 25 noviembre 2025 15:32
[root@server2asir usuario]$
```

Comandos extras:

Listar todos los dispositivos de bloque: lsblk

```
[root@server2asir usuario]$lsblk
NAME           MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0          7:0    0   64M  1 loop /snap/core20/2379
loop1          7:1    0 63,8M  1 loop /snap/core20/2682
loop2          7:2    0 91,4M  1 loop /snap/lxd/35819
loop3          7:3    0 91,4M  1 loop /snap/lxd/36558
loop4          7:4    0 38,8M  1 loop /snap/snapd/21759
loop5          7:5    0 50,9M  1 loop /snap/snapd/25577
sda             8:0    0   20G  0 disk 
└─sda1          8:1    0   1M  0 part 
└─sda2          8:2    0  1,8G  0 part /boot
└─sda3          8:3    0 18,2G  0 part 
  └─ubuntu--vg-ubuntu--lv 253:0  0   10G  0 lvm  /
sdb             8:16   0   10G  0 disk 
└─sdb1          8:17   0   10G  0 part 
└─sdb9          8:25   0   8M  0 part 
sr0            11:0   1 1024M 0 rom
```

Información más detalla: fdisk -l