# PRÁCTICA 00

## *Administración Básica de Linux*

Servicios Telemáticos

Departamento de Ingeniería Telemática

# ÍNDICE

# 1. Objetivos

## 1.1. Introducción

El ordenador se suministra al alumno con los sistemas operativos ya instalados. En el Anexo de esta práctica se recogen algunas de las operaciones más significativas usadas en ese proceso de instalación.

Esta práctica pretende servir de ***repaso*** de los conocimiento de Administración de Linux que el alumno ha debido adquirir en asignaturas previas. Existen muchas distribuciones de Linux (SUSE, Caldera, Slackware, Mandriva, Debian, Gentoo, Ubuntu, Guadalinex, etc). En el laboratorio se utilizará la distribución CentOS, perteneciente a la rama Linux Red Hat. La versión de la distribución que actualmente se está utilizando puede obtenerse mirando el contenido del archivo `/etc/centos-release`.

El sistema tiene dos usuarios creados: “root” (superusuario, con clave “root”) y “dit” (usuario normal adecuado cuando no haya que realizar tareas de administración, con clave “dit”). Por seguridad, el acceso directo al sistema (por consola gráfica o texto) como "root" está desaconsejado. Para trabajar como administrador se recomienda autenticarse con un usuario normal, y usar entonces el comando "su -" desde un intérprete de comandos. No modifique las contraseñas de los usuarios “root” ni “dit” pues serán usados por otros compañeros del laboratorio durante el curso.

Como ejercicio, entre al sistema, por ejemplo a través del entorno gráfico, usando el usuario “dit”. Posteriormente, abra una consola y cambie al usuario “root” usando el comando “su -“.

## 1.2. Documentación

- Página web de Linux Fedora Core: http://fedoraproject.org/
- “*Linux Network Administrator's Guide, 3rd Edition*”. Tony Bautts, Terry Dawson, Gregor N Purdy. Editorial: O'Reilly, 2005. ISBN: 0-596-00548-2
- Página Web de tcpdump: http://www.tcpdump.org/
- Página Web de Wireshark: http://www.wireshark.org/

- Repositorios CentOS: https://centoshelp.org/resources/repos/, https://www.centos.org/ download/mirrors/, http://mirror.centos.org/centos/7/os/x86_64/, http://mirrorlist.centos.org/?release=7&arch=x86_64&repo=os

RECUERDE: Dado que los equipos de trabajo son compartidos entre varias sesiones, **antes de comenzar con el desarrollo de esta práctica, ejecute como usuario "root" el comando siguiente** para asegurar que la configuración de su equipo es la adecuada:

```
/mnt/servicios/P00/P00inicio.sh
```

**NOTA** (leer en caso de que el anterior script no pueda ejecutarse): Con objeto de facilitar la escritura de los ficheros, ***los distintos archivos empleados por cada práctica se encuentran disponibles en el servidor del laboratorio "ait08.us.es" vía NFS.*** *Haciendo uso de la capacidad de automontaje (autofs) instalada en su sistema, para acceder a dichos ficheros basta con ejecutar el siguiente comando, el cual montará automáticamente la información remota en el directorio local indicado en el comando (si necesitase modificar alguno de los ficheros, copie dicho fichero a su equipo y realice sobre dicha copia los cambios que requiera)**:***

```
cd  /mnt/servicios/P00/
```

siendo "xx" el número de la práctica en cuestión.

NOTA: Si hubiese problemas con el servicio `autofs`, no funcionando el comando anterior, puede montar la carpeta de prácticas manualmente mediante (siendo "`IP_servidor`" la IP "172.16.17.126" en subred A y "172.16.17.254" en subred B):

```
umount /mnt/servicios/
pkill -9 automount; service autofs stop
mkdir /mnt/servicios/
mount -t nfs IP_servidor://lt/exportado/servicios/
             /mnt/servicios/
```

## 2. Intérprete de comandos

El interprete de comandos o shell de Linux recoge los comandos que un usuario introduce por el teclado y los envía al sistema operativo para que los ejecute. Para introducir comandos locales se pueden utilizar las consolas virtuales o los emuladores

de terminal en el entorno gráfico. En su sistema Linux el interprete de comandos por defecto es el programa bash.

A continuación se recuerdan distintas herramientas del intérprete de comandos que debe conocer y recordar de asignaturas anteriores.

**IMPORTANTE**: Recuerde emplear el usuario "root" con los comandos de administración (adduser, userdel, ...).

## 2.1. Consolas Virtuales y comandos básicos

Las consolas virtuales proporcionan varias "pantallas" a través de las cuales el usuario puede entrar al sistema y ejecutar programas. Para poder conmutar entre las diferentes consolas se utiliza la combinación de teclas [Control]-[Alt]-[Fz] (donde Fz puede ser F1, F2, etc.). En su sistema, normalmente dispondrá de una consola la consola gráfica (en la que se ejecuta un servidor X) en la consola "1", estando las consolas de texto de de la "2" a la "7". Conmute entre las distintas consolas y abra sesiones en cada una de ellas.

Abra un terminal con el usuario "dit" y pruebe a ejecutar algunos comandos sencillos como los propuestos a continuación:

```
ls
ls –l
ls –la
man man
man date
pwd
cd /
pwd
ls –la /usr/lib/
ls –la /usr/lib/ | more
cd
pwd
stat /etc
stat /etc/passwd
```

De las distintas variables de entorno, resulta especialmente importante la variable PATH. Ésta contiene los directorios separados por ":" donde se buscan por defecto (al no indicar ninguna ruta) los comandos que el usuario quiere ejecutar. Puede ver el valor de dicha variable con el comando:

```
echo $PATH
```

## 2.2. Comandos para Manejo de Cuentas de Usuario

Entre otros, recuerde el uso de los siguientes comandos:

```
su, sudo, useradd, userdel, passwd, usermod, groupadd,
groupdel, id, groups, finger, users, who, w, last
```

Como ejercicio, mediante el comando "`useradd`" cree dos usuarios (usuario1 y usuario2). Asígnele una contraseña a dichos usuarios empleando el comando "`passwd usuario`". Utilice el comando "su" para cambiar a dichos usuarios. Por último, elimine ambos usuarios.

Los ficheros donde se recogen los usuarios y grupos que hay en el sistema así como sus contraseñas son: /etc/passwd, /etc/shadow, /etc/group y /etc/gshadow (en "/etc/passwd" se define el grupo principal al que pertenece cada usuario; en "/etc/group", adicionalmente se puede asociar dicho usuario a otros grupos).

Como ejercicio, usando el usuario root, vea el contenido de los anteriores ficheros, los cuales debe conocer de asignaturas anteriores.

## 2.3. Comandos de Procesos

Entre otros, recuerde el uso de los siguientes comandos:

```
ps, pstree, top, nice, pidof, strace, ldd, lsof, fuser,
pgrep, kill, pkill, killall, bg, jobs, fg, sleep, exec,
nohup, poweroff (systemctl poweroff), reboot (systemctl
reboot), pm-hibernate (systemctl hibernate), pm-suspend
(systemctl suspend), halt (systemctl halt), shutdown, watch
```

Como ejercicio básico, cree un proceso en segundo plano (por ejemplo, puede ejecutar el comando "`xclock &`"), y observe como el sistema muestra ese proceso con los comandos "`ps`", "`ps ax`" y "`pstree`". Obtenga el PID del proceso y luego elimínelo usando dicho PID.

Como segundo ejemplo, ejecute en una consola virtual (una terminal del entorno gráfico) el siguiente comando:

```
watch  cat /tmp/file
```

En otra consola virtual del entorno gráfico (intente distribuir en pantalla ambas consolas para que se visualicen simultáneamente), vaya ejecutando los siguientes comandos, viendo la salida que aparece en la consola anterior tras cada comando (evalúe que está haciendo el comando "watch"):

```
echo 1  > /tmp/file
echo 2  > /tmp/file
echo 34 >> /tmp/file
```

Para terminar el comando "`watch`" debe pulsar "`Ctrl-C`". Entre otros usos, el comando "watch" es útil para ver interactivamente los registros (logs) que va generando un servicio. Por ejemplo, para ver interactivamente los registros que el sistema va generando en el fichero "/var/log/messages" ejecute en una consola:

```
watch -n0,1 "cat /var/log/messages | tail -10"
```

En otra consola (intente nuevamente que ambas consolas gráficas se visualicen simultáneamente en pantalla) use por ejemplo el comando "`su`" para cambiar entre usuarios. Observe como van visualizándose los registros del sistema en la consola del comando "`watch`".

**ADVERTENCIA SOBRE EL FORMATO DEL TIEMPO DE REFRESCO**

El formato para indicar el tiempo de refresco en el comando "watch" depende del idioma en el que esté trabajando la consola, configurado con la variable de entorno "LANG" (puede ver su valor con "echo $LANG"). Si el idioma es:

| Idioma | LANG | Formato del tiempo |
|---|---|---|
| Inglés | LANG="en_US.utf8" | Decimales con punto: `watch -n0.1 comando` |
| Español | LANG="es_ES" | Decimales con coma: `watch -n0,1 comando` |

Si lo desea, puede cambiar el valor de la variable LANG usando el comando "export LANG="valor") (afectando a la consola actual) o en el fichero "/etc/profile" (afectando a las consolas que se habran en el futuro).

**NOTA**: Para imprimir permanentemente el contenido de un fichero, alternativamente al comando "`watch cat fichero`" o "`watch tail fichero`", puede usarse directamente el comando "`tail -f fichero`".

El comando "lsof" permite obtener información de los procesos actuales, tal como:

| | |
|---|---|
| Lista de todos los recursos de red (sockets) y locales (ficheros/carpetas) en uso | `lsof \| grep xxx` |
| Procesos que están usando un determinado fichero o directorio (por omisión, listaría todos los procesos con sus ficheros) | `lsof /mnt/servicios`<br>`lsof \| grep /mnt/servicios` |
| Eliminar todos los procesos que están usando una carpeta | `lsof +D /dir \`<br>`  \| awk '{print $2}' \`<br>`  \| tail -n +2 \`<br>`  \| xargs kill -9` |
| Procesos que están usando ficheros dentro de una determinada carpeta | `lsof +D /usr/sbin/` |
| Ficheros abiertos por un determinado proceso, incluidos sockets (se indica el PID del proceso) | `lsof -p PID1,PID2` |
| Procesos que tienen abiertos sockets Unix | `lsof -i`<br>`lsof -i -P` |
| Procesos que tienen abiertas sockets de red (IP4/6), de escucha o de conexión. "-P" imprime puertos numéricamente | `lsof -i`<br>`lsof -i -P` |
| Proceso que tiene abierto determinado socket de red (IP4/6), de escucha o conexión | `lsof -i [4/6][TCP/UDP][IP][:port]` |
| Proceso que tiene abierto determinado puerto (en escucha o conexión) | `lsof -i -P :80`<br>`netstat -anp \| grep :80`<br>`ss -anp \| grep :80` |
| Proceso que tiene abierto determinado socket de red TCP/UDP en el estado indicado (LISTEN, ESTABLISHED, ...) | `lsof -i TCP:22 -sTCP:LISTEN`<br>`lsof -i UDP -p` |
| Sockets de red (IP) que tiene abiertos determinado proceso/s (PID/s). La opción "-a" indica "AND" | `lsof -i -a -p PID1,PID2` |
| Procesos abiertos por un determinado usuario | `lsof -u root` |

**RECUERDE**: Para que el comando "`lsof`" pueda buscar entre ***todos*** los procesos del sistema, debe ejecutarlo como superusuario (root). En otro caso, sólo buscará entre los procesos del usuario actual.

Como ejercicio de los comandos "ss" y "lsof":

1º Abra un navegador "firefox" y acceda a "https://www.google.es/". Obtenga su PID (comando "`ps ax`" o "`pidof firefox`"). Tras ello, usando el comando "lsof", consulte:

- Sockets Unix que tiene abiertos: `lsof -U -p PID`
- Sockets de red que tiene abiertos: `lsof -P -i -a -p PID`

2º Conéctese vía SSH a su propio equipo (puerto "22"):

```
ssh dit@localhost
```

Para obtener los puertos que está usando su servidor SSH (ejecutable "`/usr/sbin/sshd`", proceso "sshd"), puede ejecutar:

```
lsof -i -P -a -p PID
```

siendo PID el número de proceso del servidor (obtenido con "`ps ax | grep sshd`"). Puede automatizarlo usando programación POSIX:

```
for i in $(pidof sshd); do lsof -P -i -a -p $i; done
```

Usando el comando "lsof", consulte los procesos que están usando el puerto 22 (compruebe cuales son los PIDs de los procesos indicados por cada comando):

- Sockets (de escucha y conexión):

```
lsof -P -i :22
ss -anp 'sport =  22'
```

- Sólo sockets TCP de escucha:

```
lsof -P -i :22 -sTCP:LISTEN
ss -lnp 'sport =  22'
```

- Sólo sockets TCP de conexión:

```
lsof -P -i :22 -sTCP:ESTABLISHED
ss -tnp 'sport =  22'
```

De forma complementaria, el comando "fuser" permite obtener el PID/s del proceso/s que estén usando actualmente determinado fichero o sistema de archivos (con un carácter tras el PID indica si lo tiene abierto "f", abierto para escritura "F", lo está ejecutando "e", ...). Ejemplos:

```
fuser /bin/bash
fuser /dev/sda7
```

Con el comando "`ps ax | grep PID_proceso`" puede consultar los detalles del proceso correspondiente.

Por último, si se desea analizar el funcionamiento interno de una aplicación, puede usarse:

- "`strace`": muestra las llamadas al sistema que realiza. Por ejemplo, puede usarse para comprobar si una aplicación usa determinados ficheros. Ejecute los siguientes comandos y razone los resultados.

```
strace   ls    / 2>&1    | grep passwd
strace   ls -l / 2>&1    | grep passwd
strace   ls    / 2>&1    | grep group
strace   ls -l / 2>&1    | grep group
```

- "`ldd`": muestra las librerías compartidas del sistema que usa la aplicación. Ejemplos:

```
ldd /usr/sbin/sshd
ldd /usr/sbin/sshd   | grep libwrap
ldd /usr/sbin/xinetd | grep libwrap
```

## 2.4. Comandos del Sistema de Archivos

Entre otros, recuerde el uso de los siguientes comandos:

```
ln, chown, chmod, stat, find, whereis, which, locate, lsattr,
chattr, rename, namei, file
```

Como ejercicio, cree un enlace simbólico a un fichero regular y otro a un directorio, y analice los resultados con el comando "`ls -l`". Repita la prueba, pero intentando usar

ahora enlaces duros (a un fichero regular y a un directorio). Compare y analice los resultados.

**AYUDA**: Para editar la dirección a la que apunta un enlace simbólico, sin borrarlo previamente, puede usarse:

```
ln  -sfn   fichero   enlace
```

Como segundo ejercicio, cree dos enlaces simbólicos a un directorio, y haga uso de ellos. Después, intente cambiar los permisos y usuario/grupo propietarios del directorio real y de los enlaces simbólicos, observando de que modo los cambios sobre uno afectan sobre los demás.

Para ver los permisos y propietarios de todos los directorios implicados hasta una ruta, puede usar el comando "`namei`". Por ejemplo:

```
namei -l  /usr/share/emacs
```

O para consultarlo para la ruta actual:

```
namei -l  $(pwd)
```

Adicionalmente, para imprimir los permisos de un fichero o directorio en formato numérico (octal), puede usar el comando "`stat`". Por ejemplo:

```
stat                              /etc/passwd
stat -c "%a  %A  %U:%G  %n"  /home/dit/
stat -c "%a  %A  %U:%G  %n"  /home/dit/*
stat -c "%a  %A  %U:%G  %n"  /etc/passwd
```

**AYUDA**: "`stat`" imprime permisos (octal y simbólico), usuario/grupo propietarios (U/GID y nombre) y fechas del último acceso (Access), modificación de su contenido (Modify) y cambio de sus metadatos, tales como permisos, propietarios, tamaño, ... (Change).

Para ver el tipo (PDF, JPG, ASCII, ...) de contenido de un fichero puede usar el comando "`file`". Por ejemplo:

```
file  /etc/fstab
```

## 2.5. Comandos para ficheros de texto

Además de los editores de texto (vi, emacs, nano, mcedit, gedit, nedit, ...), recuerde el uso de los siguientes comandos básicos:

```
touch, diff, comm, cmp, more, less, cat, tac, dos2unix,
unix2dos, wc, strings
```

Adicionalmente, resultan especialmente útiles los comandos de "Filtros de texto", que permiten obtener (filtrar) una parte de la información recogida en un texto (esté guardado en un fichero o volcado en pantalla):

```
head, tail, cut, grep, egrep, sed, awk
```

Por ejemplo, para imprimir el número de sesiones actualmente abiertas por el usuario "root" podría usarse el comando:

```
who | grep -e "^root " | wc -l
```

o para imprimir una lista con los usuarios definidos en el sistema y su shell asociado:

```
cut -f1,7 -d ":" /etc/passwd
```

Como ejercicio sencillo, use el comando adecuado para crear un fichero vacío, y luego use el comando "cat" para comprobar que efectivamente existe y no tiene contenido.

Posteriormente, usando redirecciones, cree un fichero que contenga la línea "L1" y luego una segunda línea "L2". Usando este fichero, analice la utilidad de los comandos "head", "tail" y "grep".

## 2.6. Comandos de empaquetado y compresión

Entre otros, recuerde el uso de los siguientes comandos:

```
tar, 7za, gzip, gunzip, zcat, bzip2, bunzip2
```

Especialmente, debe recordar el uso del comando "tar", cuya sintaxis básica para comprimir y descomprimir en formato gzip, respectivamente es (para formato "bzip2" basta cambiar el atributo "z" por "j"):

```
tar  cpfvz   fichero.tar.gz  carpeta1  carpeta2
```

```
tar xpfvz fichero.tar.gz [-C carpeta_destino]
```

Como ejercicio, usando estos comandos comprima en un fichero "tar.gz" el contenido de la carpeta "/etc/init.d/" y posteriormente descomprímalo en el directorio "/tmp/comp/".

Repita la prueba anterior, pero usando los parámetros "cpfvj" (para comprimir) y "xpfvj" (para descomprimir), con objeto de crear un fichero "tar.bz2".

## 2.7. Comandos de conexión remota

Entre otros, recuerde el uso de los siguientes comandos:

```
ssh, scp, sftp, telnet, ftp
```

Como ejercicio, conectese a otro equipo del laboratorio (todos ellos tienen los mismos usuarios) y ejecute los comandos remotos necesarios para crear el fichero "/tmp/remoto". Posteriormente, usando "sftp", recupere dicho archivo, copiándolo en su ordenador en la carpeta "/home/dit/".

## 2.8. Comandos de dispositivos de almacenamiento

Entre otros, recuerde el uso de los siguientes comandos:

```
mount, umount, df, du, lsusb, fdisk, sfdisk, parted, cfdisk,
gparted, mkfs.ext3, mkfs.vfat, mkfs.ntfs, e2fsck, blkid,
lsblk, lshw, lspci, lsusb, lsscsi, hdparm, smartctl
```

Como operaciones básicas sobre los discos, realice las siguientes consultas:

| | |
|---|---|
| Dispositivos (incluidos discos) conectados al equipo de tipo SCSI/STA, USB | `lsscsi`<br>`lsusb` |
| Discos y particiones detectados | `fdisk -l [/dev/sda]`<br>`sfdisk -l [/dev/sda]` |
| Particiones detectadas con su información básica (UUID, etiqueta, ...): | `blkid` |

| | |
|---|---|
| Sistema de ficheros específico de las distintas particiones de los discos detectados | `lsblk -f` |
| Particiones detectadas con tamañ, tipo de sistema de ficheros y partición de arranque | `parted -l` |
| Manejo de particiones mediante menú en modo consola o gráfico | `cfdisk`<br>`gparted` |
| Espacio usado y libre (en MB, GB, ...) de las particiones montadas (“h” humano) | `df -h` |
| Tamaño de ciertas carpetas en MB ("s" total, "h" humano) | `du -sh /etc /opt` |
| Información detallada sobre un disco (modelo, tamaño, cilindros, ...). Paquete “smartmontools” | `smartctl -a /dev/sda` |

Especialmente, recuerde el uso del comando “mount” para el montaje de discos, cuya sintaxis básica es “`mount –t tipo dispositivo dir`”. El comando “`umount dispositivo`” o “`umount dir`” realiza el proceso de desmontaje, no pudiendo estar en uso la carpeta a desmontar (puede comprobar si algún proceso está actualmente usando la carpeta montada con el comando “`lsof dir`”). Para ver qué particiones de un disco (por ejemplo, el disco “/dev/sda”) están actualmente montadas puede usar, por ejemplo, los comandos “`mount`” o “`mount | grep sda`”.

Aunque no necesita hacer uso de la misma, para particionar los discos resulta especialmente útil la utilidad gráfica Gparted. Arránquelo (comando “`gparted &`”) para ver su aspecto. Tras ello, cierre la ventana sin aplicar ninguna operación.

Como ejercicio, usando el comando “mount”, monte en la carpeta “/tmp/linux/” una partición del disco duro de su equipo que use sistema de ficheros “ext3” (o “ext4”) y que actualmente no se encuentre montada. Acceda a dicha carpeta y compruebe que puede ver y modificar el contenido de esa partición. Posteriormente, desmóntela.

Por último, si dispone de un disco USB, conéctelo al equipo, móntelo con el comando “mount”, compruebe que tiene acceso a su contenido y, por último, desmontelo con el comando “umount”. Esto le resultará útil a lo largo de la asignatura para poder copiar ficheros del ordenador, por lo que se le aconseja que haga esta tarea.

# 3. Administración de red

En este apartado se recuerdan los conceptos más básicos para controlar la configuración de red de un equipo.

**AYUDA**: Consulte el documento "Anexo-Linux_Configuracion_Red.pdf", incluido en la Documentación de Apoyo de la Asignatura, para una mayor información sobre los ficheros y comandos de administración de red en Linux.

## 3.1. Configuración de red del sistema

En la distribución Fedora Core, la configuración de las tarjetas de red se realiza en el directorio "`/etc/sysconfig`":

| | |
|---|---|
| `/etc/sysconfig/network` | Parámetros generales de red |
| `/etc/sysconfig/network-scripts/ifcfg-ethX` | Configuración de cada tarjeta de red |

Un posible ejemplo de fichero de configuración basado en DHCP sería el siguiente:

`/etc/sysconfig/network-scripts/ifcfg-eth0`

```
DEVICE=eth0          # Dispositivo dinámico eth0
ONBOOT=yes           # Activar en arranque sistema
BOOTPROTO=dhcp       # Configurar interfaz mediante DHCP
NM_CONTROLLED=no     # Network Manager no configurará esta
                     tarjeta
```

Como scripts relevantes en la gestión de la red se pueden destacar los siguientes "shell-scripts" (puede observar su contenido para analizar su funcionalidad):

- Scripts para desactivar una interfaz de red o activarla con los parámetros de los ficheros de configuración "ifcfg-ethX" (eth0, eth1, ...):

  ```
  /sbin/ifdown  ethX
  /sbin/ifup    ethX
  ```

- Script para des/activar todas las interfaces de red (cada una con los parámetros de su fichero de configuración "ifcfg-ethX"):

  ```
  /etc/rc.d/init.d/network stop
  /etc/rc.d/init.d/network restart
  ```

Como ejercicio, haga un ping a “www.google.es” y compruebe que funciona. Desactive la interfaz “eth0” y repita el ping. Reactive las tarjetas de red con el script “network” y vuelva a repetir el ping.

## 3.2. Comandos de red

A continuación se recuerdan los principales comandos para el uso de la red, muchos de los cuales debe conocer de asignaturas anteriores.

### 3.2.1. Comandos “ip” y “ss”

En las distribiuciones Linux actuales, los comandos clásicos de red "`arp, ifconfig, route, netstat`" (entre otros) se encuentran en estado obsoleto (desarrollo abandonado, funcionalidad no garantizada y eliminación próximamente), por lo que se desaconseja su uso. En su lugar, se recomienda emplear los comandos "ip" (Internet Protocol) y "ss" (Socket Statistics):

| Paquete | Comando clásico (obsoleto) | Funcionalidad | Sustituto | Paquete |
|---|---|---|---|---|
| net-tools | `arp` | Gestionar caché ARP | `ip neighbour` | iproute2 |
| | `ifconfig`<br>`netstat -i` | Configurar tarjetas de red cableadas | `ip link, ip address` | |
| | `iptunnel` | Crear túneles IP | `ip tunnel` | |
| | `ipmaddr`<br>`netstat -g` | Gestionar direcciones multicast | `ip maddr` | |
| | `route`<br>`netstat -r` | Gestionar Tabla de encaminamiento principal (FIB) | `ip route` | |
| | `netstat` | Gestionar sockets locales | `ss` | |
| | `nameif` | Renombrar tarjetas de red | `ip link` | |
| | | | `ifrename` | ifrename |
| | `mii-tool` | Controlar hardware y drivers de red | `ethtool` | ethtool |
| wireless -tools | `iwconfig`<br>`iwlist` | Configurar tarjetas de red inalámbricas | `iw` | iw |

El comando "ip" es un comando multinivel que cubre la funcionalidad de los comandos clásicos junto a una gran cantidad adicional de funcionalidades de red avanzadas.

En esta práctica se presentan las funcionalidades básicas de ambos comandos "ip" y "ss", junto a sus equivalencias con los comandos clásicos.

**SINTAXIS COMANDO "ip"**: El comando "ip" usa la sintaxis:

```
ip  [opciones] objeto  comando  [parametros]
```

Tanto para el "objeto" (neighbour, link, address, route, ...) como para el "comando" ("list", "show", ...) se permite escribir la palabra completa o cualquier sub-prefijo de la misma. Por ejemplo, en lugar de "link" puede escribirse, indistintamente, "link", "lin", "li", "l". Normalmente no hay duplicidad, aunque en algunos casos sí; por ejemplo, para "ip link show" e "ip link set", si se escribe "ip link s" habría ambigüedad, interpretándose como "show". En algunos casos también existen alias abreviados, como por ejemplo "ip link list" e "ip link ls". Asimismo, para cada "objeto" se define un "comando" por omisión; por ejemplo, son equivalentes "ip address list" e "ip address".

Puede consultar las opciones del comando "ip" usando:

```
man ip
man ip-objeto
ip objeto help
```

https://wiki.linuxfoundation.org/networking/iproute2
https://www.policyrouting.org/iproute2-toc.html

Como ejercicio de sintaxis del comando "ip", ejecute los siguientes comandos:

```
ip neighbour
ip neigh
ip ne
ip n
ip address
ip address list
ip address show
ip a       l
ip a       l     ethX
ip route   ls
ip r       show
```

### 3.2.2. Comandos "ip neighbour" (o "arp") y "arping"

El protocolo ARP permite obtener la dirección física (MAC) correspondiente a la dirección de red de una máquina existente en la misma red local, siendo controlado por el kernel. El comando "ip neighbour" (o su equivalente clásico, el comando "`arp`") manipula la caché ARP del kernel (no tiene ningún control sobre el protocolo ARP).

Como ejercicio, ejecute y analice los siguientes comandos (consulte los manuales "`man ip`" y "`man ip-neighbour`"):

```
arp
ip -r neighbour
ping 172.16.17.126
arp -n
ip n
arp IP_otro_equipo_de_su_subred_encendido
ip n show IP_otro_equipo_de_su_subred_encendido
ping IP_otro_equipo_de_su_subred_encendido
arp -n
ip n
```

Recuerde que es posible crear entradas manuales (flag "M") en la caché arp con el parámetro "-s" y borrarlas con el parámetro "-d". Ejecute los siguientes comandos (observé que algunos funcionarán y otros no, según la subred del laboratorio en la que se encuentre):

```
arp -d 172.16.17.126 -i eth0
ip n del 172.16.17.254 dev eth0
arp -s 172.16.17.126 00:01:02:03:04:05 -i eth0
ip n add 172.16.17.254 lladdr 00:01:02:03:04:05 dev eth0
```

**NOTA**: Si al ejecutar el anterior comando "`ip n add 172.16.17.254 ...`" obtiene el mensaje "RTNETLINK answers: File exists", normalmente se deberá a que la entrada en la caché ARP para la IP 172.16.17.254 no ha sido eliminada. Ello se debe normalmente a que, al ser la pasarela, cualquier tráfico IP hacia el exterior del laboratorio auto-regenera dicha entrada. Para lograr que el comando funcione (asignando dicha MAC manual) pruebe lo siguiente:

1º Cierre cualquier navegador que esté usando, como Firefox o Chrome (estos usan conexiones persistentes, enviando tráfico de forma constante).

2º Fuerce el borrado total (no sólo la MAC como hace el comando "`ip n flush...`") de todas las entradas de la interfaz eth0 en la caché ARP ejecutando:

```
ip link set eth0 arp off; ip link set eth0 arp on
```

3º Rápidamente, ejecute el comando:

```
ip n add 172.16.17.254 lladdr 00:01:02:03:04:05 dev eth0
```

Compruebe que ya no le es posible acceder a Internet. Determine el motivo. Elimine las entradas:

```
arp -d 172.16.17.126 -i eth0
ip n del 172.16.17.254 dev eth0
```

y compruebe que vuelve a tener acceso a Internet.

Si necesitase borrar todas las entradas automáticas de la caché ARP, puede usar el comando "`ip neigh flush all`".

Asimismo, recuerde que puede usarse un fichero, normalmente guardado como "/etc/ethers", para cargar entradas manuales. Cree dicho archivo (en caso de que no exista) y escriba en él la entrada "`00:01:02:03:04:05 IP`" (siendo "IP" alguna dirección IP de su subred) y úselo para añadir una entrada manual a la caché ARP mediante el comando "arp -f /etc/ethers". Compruebe con el comando "arp" como dicha entrada manual (flag "M") se ha añadido correctamente a la caché ARP.

El orden en el que se añaden las entradas en la caché ARP no sigue un criterio fijo. En cualquier caso, ello no resulta relevante, dado que no puede haber dos entradas que posean el mismo par "IP" e "interfaz".

**Comando "arping"**

A diferencia del comando "ip neigh" (o su equivalente clásico "arp"), que sólo opera sóbre la caché ARP, el comando "arping" sólicita al kernel el envío de mensajes ARP Request (o mensajes "ARP Reply" con el parámetro "-A"). El envío y recepción de esos mensajes ARP no afectará, en general, a la caché ARP (no añade nuevas entradas).

Tras activar en Wireshark la captura sobre la interfaz eth0, ejecute los siguientes comandos y analice los resultados y el tráfico generado:

```
arping -I eth0 172.16.17.126
arping -I eth0 172.16.17.254
ip n
ip n flush dev eth0
ip n
arping -f   -I eth0 172.16.17.126
arping -c 1 -I eth0 172.16.17.254
ip n
```

Además de para obtener la MAC de una IP; puede usarse para comprobar si algún equipo de la red está usando nuestra misma dirección IP (detección de duplicidad de direcciones IP o "Duplicate Address Detection, DAD", RFC 5227 y RFC 2131/4.4.1):

```
arping -D -I eth0 IP_de_su_equipo_a_comprobar
```

Este mecanismo es usado automáticamente por las tarjetas de red al arrancar (`ifup ethX`), para que, antes de comenzar a usar la dirección IP configurada, comprobar que dicha dirección no está ya en uso en la red. El funcionamiento se basa en enviar un ARP Request (por la interfaz ethX indicada) preguntando por dicha IP; si se obtiene ARP Reply, indicará que la dirección ya está en uso. Usando Wireshark, monitorice la red para comprobar los mensajes intercambiados tanto con arping (usando -D) como al activar una tarjeta de red. Analice los resultados.

### 3.2.3. Comandos "ip link" e "ip address" (o "ifconfig")

Los comandos “ip link” e “ip address” (o su equivalente clásico “ifconfig”) se utilizan para configurar las interfaces de red del equipo.

Debe advertir que, a diferencia del script “`ifup`” que configura todos los parámetros de la interfaz establecidos en su fichero de configuración (pasarela, ...), el comando “`ip link set up eth0`” (o su equivalente clásico “`ifconfig eth0 up`”) sólo activa la interfaz indicada asignándole su IP y máscara (no el resto de parámetros). Como ejercicio, ejecute los siguientes comandos y analice el resultado (consulte los manuales “`man ip-link`” y “`man ip-address`”)

```
/sbin/ifconfig
ifconfig
ip address show up
ip link show up
ifconfig -a
ip address
ip link
ifconfig eth0
ip a l eth0
ip l l eth0
ping su_propia_dirección_IP
ping IP_de_ait08_en_su_subred
ping IP_de_un_equipo_de_la_misma_subred_encendido
ping IP_de_un_equipo_de_la_otra_subred_encendido
ip link set down eth0
ping IP_de_un_equipo_de_la_misma_subred_encendido
ping su_propia_dirección_IP
ifdown eth0
ping su_propia_dirección_IP
ip link set up eth0
ip a add IP_de_SU_equipo_en_la_subred_del_laboratorio/25 dev
eth0
ping su_propia_dirección_IP
ping IP_de_un_equipo_de_la_misma_subred_encendido
```

```
ping IP_de_un_equipo_de_la_otra_subred_encendido
ifup eth0
ping IP_de_un_equipo_de_la_otra_subred_encendido
ifdown eth0; ifup eth0
ping IP_de_un_equipo_de_la_otra_subred_encendido
```

**NOTA**: Si al ejecutar los comandos:

```
ifdown ethX; ifup ethX
```

obtiene el mensaje:

```
Determining IP information for eth0...dhclient(6394) is
already running - exiting.
```

Ello se debe a que la interfaz "ethX" está configurada para usar DHCP, y el comando "ifdown" no ha podido cerrar (intenta un cierre suave) el proceso "dhclient" (el comando "ifup", al detectar que el proceso está activo, termina directamente su ejecución con ese mensaje). Para solucionarlo, basta forzar el cierre del proceso:

```
pkill -9 dhclient
```

y luego volver a ejecutar los comandos anteriores.

Como ejemplo de uso de cambio de dirección de la interfaz, realice las siguientes pruebas:

```
ip addr add     10.0.0.1/24 dev eth1
ifconfig  eth1  10.0.0.2/24
ip addr add     10.0.0.3/24 dev eth1
ifconfig  eth1  10.0.0.4 netmask 255.255.255.0
ifconfig
ip a
ifdown eth1
```

Analice los resultados.

**AYUDA**: los comandos "`ip addr add IP/mascara dev ethX`" y "`ifconfig eth1 IP/mascara`" NO son 100% equivalentes. El comando:

- "`ip`": va asociando (añadiendo) múltiples direcciones a la tarjeta de red (si ejecuta varios "`ip a add...`", podrá ver con "`ip a`" las múltiples direcciones asignadas (para eliminarlas, debe usarse "`ip a del...`"). Asimismo, este comando sólo modifica la dirección IP y máscara de la interfaz, pero no su estado "up/down".

- "`ifconfig`": sólo opera sobre (modifica y muestra) una de las direcciones asignadas a la interfaz (pero realmente la interfaz usa todas las direcciones que muestra "`ip a`"; por ejemplo, puede comprobar como un "`ping IP`" a cualquiera de esas direcciones se envía a través de la interfaz "lo"). Además, a diferencia del equivalente "ip", con este comando "ifconfig" automáticamente se cambia el estado de la interfaz a "up".

El comando "`ifup ethX`" usa internamente el comando "`ip addr add IP/nn dev ethX`". Consecuentemente, si realizasen las operaciones:

```
# Se cambia IP en ifcfg-ethX a: IP1
ifup ethX
# Se cambia IP en ifcfg-ethX a: IP2
ifup ethX
# Se cambia IP en ifcfg-ethX a: IP3
ifup ethX
# Se cambia IP en ifcfg-ethX a: IP4
ifup ethX
```

al ejecutar:

- "`ifconfig`": se muestra solamente "IP1".

- "`ip a`": se muestran las 4 IPs con las que realmente trabaja la interfaz ethX (IP1, IP2, IP3, IP4).

Esto es un ejemplo de alguno de los motivos por los que se recomienda abandonar el uso del comando clásico "`ifconfig`".

**NOTA**: El script "`ifup eth1`" usa internamente el comando "`ip addr add IP/nn dev ethX`". Consecuentemente, si el fichero "ifcfg-ethX" tiene configurada para la interfaz una dirección distinta a las que actualmente tiene asignadas (`ip a`), al ejecutar el comando "`ifup ethX`" se le asignara a la interfaz como otra dirección adicional la indicada en "`ifcfg-ethX`" (este funcionalidad la realiza el comando "`ifup`" independientemente de que la interfaz se encuentre activada o desactivada).

El comando "`ifdown ethX`" usa internamente el comando "`ip addr flush dev eth0 scope global`" para desconfigurar completamente los valores de la interfaz.

### 3.2.4. Comando "ip route" (o "route")

El comando "ip route" permite manipular las tablas de encaminamiento del equipo (o su equivalente clásico "route" sólo permite manipular la tabla de encaminamiento principal o FIB) . Como ejercicio, ejecute los siguientes comandos y analice el resultado (consulte "`man ip-route`"):

```
route -n
ip r ls
route add -net 172.16.17.0 netmask 255.255.255.128 dev eth0
ip route add  172.16.17.128/25                      dev eth0
ip route add   default via 172.16.17.253            dev eth0
ip r ls
ping 172.16.17.2
ifdown eth0; ifup eth0
```

### 3.2.5. Comandos "ping" y "traceroute"

a) "ping": permite hacer pruebas de accesibilidad a un equipo usando el protocolo ICMP. Como ejercicio, ejecute los siguientes comandos y analice el resultado:

```
ping su_propia_dirección_IP                    (parar con [Ctrl-c])
ping -c 4 -i 3 su_propia_dirección_IP
ping -w 5 su_propia_dirección_IP
ping www.google.es
ping -r www.google.es
```

b) "traceroute": posibilita el seguimiento de la ruta que sigue un paquete hasta alcanzar su destino. Como ejercicio, ejecute los siguientes comandos y analice el resultado::

```
traceroute su_propia_dirección_IP
traceroute ait08.us.es
traceroute www.google.es
traceroute -r www.google.es
```

### 3.2.6. Comandos "ss" (o "netstat") y "nmap"

a) "ss" (equivalente clásico "netstat"): permite analizar los sockets locales del equipo. Como ejercicio, ejecute los siguientes comandos y analice el resultado (consulte "`man ss`"):

```
netstat
ss
netstat -l
ss -l
netstat -l -p
ss -l -p
netstat -t
ss -t
netstat -u
ss -u
```

b) "nmap": herramienta para el escaneo del estado de los puertos y servicios en una red (existen otras similares, tales como "hping3" o "nping"). Como ejercicio, ejecute los siguientes comandos y analice el resultado:

```
nmap -sn 172.16.17.* 192.168.0.*
nmap -p 80,8080 su_propia_dirección_IP
nmap -sS su_propia_dirección_IP
nmap -sS  172.16.17.* | grep ssh
```

### 3.2.7. Comandos "host/nslookup" y "dig"

a) "host" y "nslookup": utilidades DNS para realizar conversiones entre nombres de red y direcciones IP. Los nombres resueltos localmente se encuentran configurados en el fichero "`/etc/hosts`". Como ejercicio, ejecute los siguientes comandos y analice el resultado:

```
host www.esi.us.es
nslookup ait08.us.es
host 193.147.162.169
nslookup 172.16.17.254
```

Los servidores DNS que usa su equipo están configurados en el fichero "/etc/resolv.conf". Observe el contenido de dicho archivo.

b) "dig": muestra información relacionada con los servidores DNS. Los nombres resueltos localmente se encuentran configurados en el fichero "`/etc/hosts`". Como ejercicio, ejecute los siguientes comandos y analice el resultado (consulte "`man dig`"):

```
dig www.google.es
dig www.google.es NS
dig us.es +trace
```

## 3.3. Monitorización de red

En esta práctica se utilizarán las siguientes herramientas de monitorización para redes de área local tipo Ethernet, las cuales debe conocer de asignaturas anteriores:

- "tcpdump": Rastreador (sniffer) en modo texto. Para capturar todo el tráfico recibido (independientemente de la MAC destino), requiere activar manualmente el modo promiscuo de la tarjeta de red (puede usarse, por ejemplo, el comando "`ip link set ethX promisc on`").

- "wireshark": Analizador de protocolos gráfico.

Ambas se basan en la librería de captura "pcap" (libpcap en sistemas Linux y winpcap en sistemas Windows), presentando la misma sintaxis de filtros de captura, la cual también ha estudiado en cursos anteriores.

Como ejercicio, realice las siguientes tareas con una de dichas aplicaciones:

a) Descubra si su equipo está utilizando el servicio DNS y, en dicho caso, obtenga la dirección IP del servidor o servidores DNS que se estén usando.

b) Capture todo el tráfico en su interfaz local "`lo`". Durante la captura, realice un ping hacia su propia dirección de red "`172.16.17.x`" y observe los mensajes capturados. **Analice si la dirección IP de esos mensajes coincide con la dirección IP de la interfaz "lo", y razone el motivo.**

c) Conexión HTTP:

   1º Ejecute el analizador "wireshark" con el siguiente filtro:

```
host SU_DIR_IP and (tcp and port 80)
```

2º Abra una página Web (que use el puerto 80) usando un navegador.

3º Pare la captura de datos y observe las tramas capturadas. Seleccione una de las tramas recibidas (cualquiera), pulse con el botón derecho del ratón y seleccione "`Follow TCP Stream`". Comprobará que se puede ver todo lo que se ha transmitido y recibido, lo que indica la nula seguridad de este protocolo.

d) Conexión HTTPS: repita el apartado anterior pero abriendo una página https. Comprobará que los datos aparecen encriptados y no son visibles.

e) Conexión FTP:

1º Conéctese vía SSH a otro equipo del laboratorio que esté encendido.

2º Arranque en este equipo un servidor FTP ejecutando como superusuario el siguiente comando: `/sbin/service vsftpd start`

3º En su equipo, ponga el analizador "wireshark" a capturar todo el tráfico FTP. Tras ello, desde su equipo, acceda con un cliente ftp al servidor FTP remoto que acaba de arrancar (abra la sesión ftp con el usuario "dit"). Tras ello, active el filtro de visualización (distinto a filtro de captura) "`frame.cap_len <= 200` " para que sólo se muestren los paquetes de longitud no superior a 200 bytes.

4º Por último, realice los pasos necesarios para obtener la clave usada en esa conexión FTP.

| | |
|---|---|
| **RECUERDE**: | ***NUNCA debe apagar el equipo directamente***; hágalo de forma ordenada mediante los comandos "`poweroff`" (apagar) ó "`reboot`" (reiniciar) (o bien con su equivalente gráfico). Compruebe que **tanto el equipo como el monitor** quedan correctamente apagados cuando finalice la práctica. |

# A. Configuración de Linux CentOS

La información de este Anexo es informativa, no necesita realizar lo aquí explicado en esta práctica. La principal utilidad de este anexo es su aplicación en la realización del Trabajo de la asignatura.

## A.1. Recompilación del kernel

En el sistema pueden encontrarse instalados múltiples kernels, ubicados en la carpeta "`/boot`". Para saber la versión del kernel con el que se ha arrancado, puede usarse el comando:

```
uname -r
```

La compilación de un nuevo kernel puede ser necesaria en algunas ocasiones, tales como

- Instalar una nueva versión de kernel no disponible en el sistema de paquetes de la distribución.
- Aplicar un parche al kernel.
- Activar parámetros del kernel sólo disponibles en tiempo de compilación.
- Por limitaciones hardware, necesidad de preparar un kernel ligero y adaptado a la máquina concreta en la que se encuentra el sistema.

Para la selección del kernel a instalar debe tenerse en cuenta que:

- Existen múltiples versiones del kernel:

  https://www.kernel.org/category/releases.html

  La aparición de una nueva versión no implica el abandono de las anteriores, sino que las penúltimas versiones se siguen "corrigiendo" para arreglar fallos.

- Una versión de kernel posterior al instalado por el sistema de paquetes de la distribución puede haber eliminado características que usen algunas aplicaciones, por lo que el uso de una versión más alta debe hacerse con precaución.

- Una ventaja de usar el kernel del sistema de paquetes de la distribución es que dicho kernel se actualizará automáticamente con el sistema de paquetes. En cambio, si se usa un kernel compilado manualmente, el administrador será el encargado de ir actualizando dicho kernel.

### A.1.1. Obtención del código fuente del kernel

Para la compilación de un nuevo kernel, el primer paso es obtener el código fuente del mismo (suele ubicarse en la carpeta "/usr/src/kernels/"), existiendo dos tipos de código fuente de kernel:

a) Código fuente oficial: descargable de la URL:

   http://www.kernel.org/

   Por ejemplo, podrían usarse los siguientes comandos:

```
cd /usr/src/
wget https://www.kernel.org/pub/linux/kernel/v3.x/linux-3.10.tar.xz
tar xf linux-3.10.tar.xz
ln -s  /usr/src/linux-3.18.4/  /usr/src/linux/
```

b) Código fuente de la distribución (con múltiples parches y adaptaciones realizadas por la distribución): para la distribución CentOS, el código fuente en formato "*.src.rpm" está disponible en:

   http://vault.centos.org/7.N.YYMM/os/Source/SPackages/
   http://vault.centos.org/7.N.YYMM/updates/Source/SPackages/

   siendo (puede obtenerse la rama y versión actualmente usados por el sistema con el comando "`cat /etc/centos-release`"):

   - "7.N": rama principal "7" (CentOS 7) y número de versión dentro de esa rama "N".
   - "YYMM": año y mes de esa versión.

   Los pasos para extraer el código fuente de estos paquetes de detallan en:

   http://wiki.centos.org/HowTos/I_need_the_Kernel_Source

pudiendo resumirse en los siguientes para la rama CentOS 7:

```
# Como superusuario "root"
yum install rpm-build redhat-rpm-config unifdef
mkdir -p ~/rpmbuild/{BUILD,BUILDROOT,RPMS,SOURCES,SPECS,SRPMS}
echo '%_topdir %(echo $HOME)/rpmbuild' > ~/.rpmmacros
yum install rpm-build redhat-rpm-config asciidoc bison
yum install hmaccalc perl-ExtUtils-Embed pesign xmlto
yum install audit-libs-devel binutils-devel
yum install elfutils-devel elfutils-libelf-devel
yum install ncurses-devel newt-devel numactl-devel qt-devel
yum install pciutils-devel python-devel zlib-devel
# Como usuario normal "dit":
rpm -i http://vault.centos.org/7.1.1503/updates/Source/SPackages/

/kernel-3.10.0-229.4.2.el7.src.rpm
cd ~/rpmbuild/SPECS
rpmbuild -bp --target=$(uname -m) kernel.spec
```

Tras ello, el código fuente del kernel se encontrará disponible en la carpeta:

```
~/rpmbuild/BUILD/kernelXXX/linuxXXX/
```

Si se desea mover a la carpeta "/usr/src/", podría ejecutarse como superusuario:

```
cd ~/rpmbuild/BUILD/kernel-3.10.0-229.4.2.el7/
mkdir -p /usr/src/kernels/
mv ./linux*/   /usr/src/kernels/
ln     -s     /usr/src/kernels/linux-3.10.0-229.4.2.el7/
/usr/src/linux
```

### A.1.2. Compilación del kernel

Una vez obtenido el código fuente, el siguiente paso es configurar y compilar el kernel. Asumiendo que el código fuente del kernel a compilar se encuentra en "`/usr/src/linux/`", los pasos serían los siguientes:

1º Ajustar la configuración del kernel a la deseada (son necesarios los paquetes "gcc" y "`ncurses-devel`"):

```
cd /usr/src/linux
cp .config ./config-bck
make menuconfig
```

accediendo al menú de parámetros del kernel (si desea usar el menú de configuración gráfico "`make xconfig`", deberá tener instalado el paquete "`qt-`

`devel`"). Tras modificar todos los valores deseados, se usa "`Save`" para guardar los cambios y "`Exit`" para terminar. La configuración será almacenada en el fichero:

```
/usr/src/linux/.config
```

Comentarios útiles sobre esto son:

- Si dispone de una preconfiguración del kernel almacenada en un fichero, bastará copiarlo con el nombre anterior, recomendándose hacer previamente una copia de seguridad del mismo. Suele disponerse de estos ficheros de configuración en:
  - "/boot/config-xxx": fichero de configuración usado para configurar dicho kernel "xxx" ya compilado en "/boot".
  - "/usr/src/linux/configs/": ficheros de configuración ofrecidos en el código fuente del kernel de la distribución.
  - "/usr/src/linux/arch/x86/configs/x86_64_defconfig": ficherfichero de configuración usada de plantilla base para construir el ".config" de compilación si no hay ninguno.
- Si se copia el ".config" de un kernel para usarlo de preconfiguración de partida para compilar otro kernel, puede ejecutarse el comando:

  ```
  make oldconfig
  ```

  encargado de preguntar al usuario como configurar las opciones del kernel a compilar que no existián en el kernel de partida.

2º Opcionalmente, puede definirse un sufijo (e.g. "-lt") a añadir al nombre del kernel (y módulos) a compilar, con objeto de no eliminar el actual (necesario si se desea mantener el kernel actual como posibilidad en el arranque). Para ello, edite la siguiente línea en el fichero "`/usr/src/linux/Makefile`":

```
EXTRAVERSION = -lt
```

3º Compilar el kernel y los módulos haciendo uso de la configuración establecida: basta ejecutar los siguientes comandos:

```
cd /usr/src/linux/
```

```
make
make modules_install
make install
```

Estos comandos, además de compilar el kernel, añaden en el gestor de arranque GRUB una opción para arrancar con él.

De forma alternativa a los comandos anteriores, la compilación del kernel puede realizarse de forma más controlada mediante los siguientes pasos (pueden visualizarse las distintas opciones disponibles para la compilación del kernel con "`make help`"):

a) Limpiar los ficheros de compilación:

```
make clean
```

b) Compilar la imagen del kernel y colocarla en el directorio de arranque (no reemplazar la imagen anterior, mantenerla por seguridad hasta comprobar que el nuevo kernel arranca correctamente):

```
make bzImage
cp /usr/src/linux/arch/i386/boot/bzImage /boot/vmlinuz-3.10.0-lt
```

c) Compilar e instalar los módulos (resulta necesario tener instalado el paquete "`module-init-tools`" para que el sistema pueda cargar los módulos):

```
cd /usr/src/linux/
make modules; make modules_install
```

d) Crear la imagen ramdisk de arranque (el segundo parámetro corresponde al nombre del subdirectorio en "`/lib/modules/`" donde se han instalado los módulos compilados):

```
mkinitrd  /boot/initramfs-3.10.0-lt.img  3.10.0-lt
```

e) Actualizar la tabla de símbolos correspondiente al nuevo kernel compilado:

```
cp /usr/src/linux/System.map /boot/System.map-3.10.0-lt
```

g) Si no se ha usado la opción "`make install`", se deberá preparar el gestor de arranque (GRUB en nuestro caso) para que ofrezca la opción de arrancar con este nuevo kernel: ver subapartado siguiente.

***Una vez que arranque con el nuevo kernel compilado,*** se recomienda ejecutar el siguiente comando para calcular las dependencias entre los módulos del nuevo kernel:

```
/sbin/depmod -a
```

Tras todo ello, cuando haya comprobado que el nuevo kernel ("`3.10.0-1t`") funciona correctamente, puede proceder a eliminar la versión antigua (borrando las imágenes del kernel en "`/boot`", el directorio de módulos en "`/lib/modules/`", así como la entrada en el gestor de arranque GRUB).

### A.1.3. Preparar un paquete rpm con el kernel compilado

En el caso de usar el código fuente de la distribución, si se desea preparar un paquete rpm con el nuevo kernel compilado (facilitando su uso en varios equipos y su control mediante las herramientas de paquetes), pueden seguirse las instrucciones indicadas en los siguientes enlaces: (por motivos de seguridad y estabilidad del sistema, se aconseja realizar el proceso mediante un usuario normal no root):

http://wiki.centos.org/HowTos/I_need_the_Kernel_Source
http://wiki.centos.org/HowTos/Custom_Kernel
http://wiki.centos.org/HowTos/RebuildSRPM

### A.1.4. Compilación de un nuevo módulo

Si se desea añadir un nuevo módulo al sistema, pueden seguirse las instrucciones recogidas en:

http://wiki.centos.org/HowTos/BuildingKernelModules

Tal como se indica en dicho enlace, según el sistema de módulos, para añadir un nuevo módulo pueden darse dos opciones:

a) Basta usar las cabeceras (headers) del kernel, sin requerir todo su código fuente: pueden instalarse con:

```
yum install kernel-devel
```

o descargar e instalar manualmente el rpm, por ejemplo:

```
rpm -i http://vault.centos.org/7.0.1406/os/x86_64/
```

```
/Packages/kernel-devel-3.10.0-123.el7.x86_64.rpm
```

Pueden verse los paquetes del kernel instalados con:

```
rpm -qa | grep kernel
```

El código fuente de los módulos del kernel se encuentran en:

```
/usr/src/linux/xxx/
```

b) Resulta necesario usar el código fuente del kernel completo, pudiendo obtenerlo conforme se ha indicado anteriormente.

Los módulos suelen instalarse en la carpeta:

```
/lib/modules/
```

y la configuración para su carga suele encontrarse en:

```
/etc/modprobe.d/*.conf
/etc/modules-load.s/*.conf
```

## A.2. Configuración del gestor de arranque

El arranque de su sistema está gestionado por la aplicación GRUB (otra posibilidad sería instalar el gestor LILO). Dicho gestor de arranque está preparado para leer en cada arranque del equipo el fichero:

```
/boot/grub2/grub.cfg
```

ofreciendo según su contenido un determinado menú de arranque. Para que el menú de arranque ofrezca la posibilidad de arrancar con el nuevo kernel compilado, bastaría añadir en dicho fichero las líneas siguientes:

```
/etc/grub2/grub.cfg
```

```
...
menuentry '2) Linux 1 (CentOS Linux (3.10.0-lt)' --class
rhel fedora --class gnu-linux --class gnu --class os
--unrestricted $menuentry_id_option
'gnulinux-3.10.0-lt-advanced-d8943fcc-6a3e-460d-8bb2-a142163
15e8c' {
  load_video
  set gfxpayload=keep
  insmod gzio
  insmod part_msdos
  insmod ext2
  set root='hd0,msdos2'
  search --no-floppy --label "linux1" --set=root
  linux16 /boot/vmlinuz-3.10.0-lt root=LABEL=linux1 ro
           crashkernel=auto rhgb quiet net.ifnames=0
           biosdevname=0 LANG=es_ES.UTF-8 systemd.debug
  initrd16 /boot/initramfs-3.10.0-lt.img
}
...
```

donde:

- "hd0,msdos2": indica la partición en la que se encuentra esta carpeta "/boot/" que contiene los ficheros de Grub. ("hd0" es el primer disco duro, normalmente equivalente a "/dev/sda"; y "msdos2" sería la partición de tipo DOS en dicho disco, equivalente a "/dev/sda2").

- "linux1" es la etiqueta asignada a la partición "/dev/sda2" (pueden verse las etiquetas con el comando "`blkid`").

Tras ello, si reinicia su equipo, deberá aparecer el nuevo kernel en el menú de arranque.

Sobre lo anterior, deben realizarse las siguientes aclaraciones:

a) Para instalar GRUB en el sistema se usa el comando "grub2-install". Por ejemplo, para instalarlo en el MBR del disco /dev/sda se ejecutaría:

   `grub2-install [--root-directory=/] /dev/sda`

   donde "`--root-directory`" permite indicar la carpeta bajo la que se encuentra el directorio /boot con la configuración de grub (por omisión, el directorio raíz "/").

b) Por motivos de seguridad, en algunos casos al fichero "/boot/grub2/grub.cfg" se le asigna el atributo especial de bloqueo "i" ("i"nmutabilidad) que impide pueda ser modificado. Pude comprobarlo con:

```
lsattr /boot/grub2/grub.cfg
```

y si está aplicado, eliminarlo con (puede volverlo a aplicar con "+i"):

```
chattr -i /boot/grub2/grub.cfg
```

c) La sintaxis concreta del fichero de configuración "/boot/grub2/grub.cfg" se encuentra explicada en el manual de grub:

http://www.gnu.org/software/grub/manual/

Tal como recoge dicho manual, puede evitarse modificar directamente dicho fichero, configurando GRUB mediante la modificación de los ficheros de shell-script:

```
/etc/grub.d/*
/etc/default/grub
```

y la posterior ejecución del comando:

```
grub-mkconfig
```

que genera automáticamente el fichero "/boot/grub2/grub.cfg" a partir de los shell-script anteriores. No obstante, el propio manual advierte que en algunos casos, (como añadir nuevas entradas en el menú y controlar su orden), la modificación de los anteriores shell-scripts puede ser más compleja que editar directamente el fichero "grub.cfg".

## A.3. Montaje de particiones

Al arrancar el sistema se lleva a cabo el montaje del sistema de archivos raíz y demás dispositivos siguiendo las líneas especificadas en el fichero "`/etc/fstab`". Para ajustarlo a la configuración de la máquina que se está instalando, debe editarse dicho fichero para que su contenido sea el siguiente:

```
/etc/fstab
```

```
#LABEL=windows    /mnt/sda1       ntfs-3g defaults           0  0
LABEL=linux1      /               ext3    errors=remount-ro  0  0
#LABEL=linux2     /mnt/sda6       ext3    errors=remount-ro  0  0
LABEL=swap        none            swap    sw                 0  0
LABEL=lprogramas  /opt/programas  ext3    errors=remount-ro  0  0
#LABEL=wprogramas /mnt/sda8       ntfs-3g defaults           0  0
/dev/sr0          /media/cdrom0   udf,iso9660   user,noauto  0  0
```

**NOTA**: Se recomienda bloquear cualquier modificación, incluso para el superusuario, sobre el fichero de particiones, mediante el comando:

```
chattr +i /etc/fstab
```

## A.4. Instalación de software y Repositorios

En Linux existen varias maneras de instalar software: paquetes, ficheros binarios y código fuente.

Un paquete contiene un programa software con todos los archivos necesarios así como información de donde copiarlos. Existen distintos formatos de paquetes y cada distribución utiliza uno u otro. CentOS (y la rama Red Hat en general) utiliza los paquetes tipo RPM. Cada paquete RPM es específico de un kernel y una distribución. Un paquete RPM creado para otra distribución probablemente no se instalará bien, ya que suelen ser archivos compilados con las características y librerías específicas de una distribución.

El comando para gestionar paquetes en CentOS (instalar, actualizar, desinstalar, etc.) es "`rpm`" (similar a "`dpkg`" en Debian). Se trata de una herramienta de gestión de paquetes de software que se puede utilizar para instalar, verificar, actualizar y borrar otros paquetes de software de forma individual. También permite comprobar si un paquete está instalado y de qué versión se trata.

| | |
|---|---|
| Instalar un paquete | `rpm –i fichero.rpm` |
| Forzar la reinstalación de un paquete | `rpm –i --force fichero.rpm` |

| | |
|---|---|
| Desinstalar un paquete | `rpm –e nombre_reducido_del_paquete` |
| Actualizar un paquete | `rpm –U fichero.rpm` |
| Comprobar si un paquete está instalado | `rpm –q nombre_reducido_del_paquete` |
| Lista de paquetes instalados | `rpm –qa` |

Los distintos paquetes ".rpm" y ".src.rpm" de las distribuciones CentOS se encuentran disponibles en (recuerde que puede obtener la versión de CentOS usada en su sistema con el comando "`cat /etc/centos-release`"):

http://mirror.centos.org/centos/
http://vault.centos.org/

La instalación de software mediante paquetes rpm requiere ocuparse de gestionar adecuadamente ciertas tareas tales como la ubicación del paquete a instalar o la detección de dependencias. Con objeto de facilitar la instalación de paquetes rpm existen herramientas basadas en repositorios (almacenes de paquetes) que localizan automáticamente los paquetes, así como sus dependencias. Entre otras herramientas, están "`yum`" (equivalente a "`apt`" en Debian) o la interfaz gráfica "`yumex`" (similar a "`synaptic`" en Debian).

La herramienta "`yum`" (Yellowdog Updater Modifier), propia de sistemas Red Hat, permite facilitar considerablemente la instalación de paquetes, **resolviendo automáticamente las dependencias** (puede consultarse la ayuda "`man yum`"). Para usarla, se basa en una serie de repositorios creados en la carpeta "`/etc/yum.repos.d/`", que contienen, entre otros datos, las URLs en las que se encuentran los ficheros "rpm". Además de manualmente, es posible instalar repositorios usando el comando "rpm", por ejemplo:

```
rpm -i http://mirror.symnds.com/distributions/fedora-epel/7/x86_64/
```

A través de "yum", la instalación de un paquete rpm (y sus dependencias) que se encuentre en los repositorios, como por ejemplo la utilidad "`qt-devel`", puede realizarse simplemente mediante el comando (si "`yum`" detecta alguna variación en la configuración de repositorios, actualizará automáticamente la cache cuando busque algún paquete rpm):

```
yum update
yum install qt-devel
```

| | |
|---|---|
| Instalar un paquete | `yum install nombre_reduc_paquete[=vers]` |
| Forzar la reinstalación de un paquete | `yum reinstall nombre_reducido_paquete` |
| Desinstalar un paquete | `yum erase nombre_reducido_paquete` |
| Actualizar un paquete (o todos) | `yum update [nombre_reducido_paquete]` |
| Buscar los paquetes que en su nombre/descripción contienen la palabra "cadena" | `yum search cadena` |
| Buscar el paquete que instala el comando "comando" | `yum provides comando` |
| Lista de paquetes disponibles en los repositorios | `yum list` |
| Lista de paquetes instalados | `yum list installed` |
| Lista de paquetes de los que depende un paquete | `yum deplist nombre_reducido_paquete` |