

GNU/Linux - Resumen comandos y ficheros

[Información sobre el documento.....1](#)

[1. - Inicio y cierre de sesión..... 1](#)

[2. -Páginas de manual..... 2](#)

[3. -Comandos básicos de sistema de ficheros..... 2](#)

[Gestión de ficheros y directorios.....2](#)

[Tuberías.....7](#)

[Redirecciones.....7](#)

[Permisos.....7](#)

[Gestión de procesos.....11](#)

[4. -Sistema de ficheros en Linux..... 12](#)

[Directorios importantes..... 13](#)

[Dispositivos: ficheros especiales..... 14](#)

[Sistema de ficheros: montaje y desmontaje de dispositivos..... 15](#)

Información sobre el documento

El objetivo de este documento es enseñar, compartir conocimientos para facilitar el aprendizaje. Este documento es mejorable, y será actualizado si es preciso.

Este documento puede ser utilizado para uso personal, no comercial, como se presenta, respetando una serie de **condiciones de uso basadas en Licencia Creative Commons**, como aparece en el logotipo, que se describe a continuación



- El **documento se proporciona como está**, por tanto *no se pueden realizar modificaciones*, ni en el **formato** ni en el **contenido**, ni trabajos derivados, sin la autorización expresa del autor.
- Se debe **mencionar al autor** del mismo, por supuesto sin modificar los enlaces o imágenes introducidas por éste en el documento.
- **No se puede utilizar** este material **con fines lucrativos, comerciales o cualquier uso que pueda proporcionar, directa o indirectamente, un beneficio económico de terceros**, sin la autorización expresa del autor.
- Si se desea publicar el documento en algún sitio web, se debe hacer a través de un **enlace al documento en el sitio del autor** (www.educativa.es o cursos.educativa.es)

Si se desea publicar como un recurso dentro de un sitio web, sin utilizar un enlace al material en el sitio web del autor, se debe solicitar autorización expresa y, en cualquier caso, referenciar el sitio web del autor (www.educativa.es).

Espero sea de provecho ;)

1. - Inicio y cierre de sesión

Inicio de sesión. Para iniciar sesión en una terminal en modo texto, el sistema solicita el nombre del usuario con el que se desea iniciar sesión (**login**). Una vez introducido el nombre del usuario, el sistema solicita la contraseña de dicho usuario (**password**) para autentificarlo. No aparecerá ningún texto mientras se escribe la contraseña, por motivos de seguridad.



Si todo el usuario y contraseña son correctos, se iniciará una sesión en el sistema, pudiendo interactuar con el sistema a través del interprete de comandos.

Cierre de sesión. Una vez se haya terminado de trabajar con el sistema es importante finalizar la sesión, de forma que los procesos lanzados se cierren correctamente y se vuelquen los datos en memoria a fichero.

Para cerrar la sesión bastará con ejecutar el comando **exit**. Si todo ha ido bien, deberá aparecer de nuevo la solicitud de inicio de sesión (login).

2. -Páginas de manual

Las páginas de manual proporcionan información sobre comandos, aplicaciones, ficheros de configuración, etc. Para consultar las páginas de manual se utiliza el comando **man**.

```
man <nombre_comando>
Muestra un manual de ayuda sobre el comando mv
man mv
```

Algunos comandos tienen varias secciones de ayuda entre los que navegar.

```
man -a mv # Muestra todas las páginas de manual existentes para el comando mv secuencialmente.
man 7 man # Muestra la página de manual para man de la sección 7
```

3. -Comandos básicos de sistema de ficheros

Gestión de ficheros y directorios

ls muestra el contenido de directorios (ficheros y directorios)

Sintaxis

```
ls [opciones] [fichero..]
```

```
ls #Muestra el contenido del directorio actual
ls practicas #Muestra el contenido del directorio practicas
ls /home/pepe/practicas/file1 #Muestra el nombre del fichero file1
ls -l #Listado extendido o largo del directorio actual
ls -t #Clasifica los ficheros y directorios por edad
ls -R #Muestra recursivamente el contenido del directorio actual y de todos su subdirectorios
```

cd Cambia el directorio de trabajo actual.

Sintaxis

```
cd <ruta nuevo directorio>
```

```
cd /usr/bin # cambia al directorio /usr/bin (ruta absoluta)
cd practicas # cambia al directorio practicas, contenido en el directorio actual (ruta relativa)
cd # cambia al directorio de inicio o "home" del usuario actual
```

cat Concatena los archivos que se pasan por parámetro y los muestra en pantalla.

```
cat file1 # muestra el contenido de file1 en pantalla
cat file1 file2 #muestra el contenido de file1 uniéndolo con el de file2 en pantalla.
```

more Permite visualizar, por páginas completas, el contenido de ficheros de texto. Permite movernos por el texto solo hacia adelante.

```
more file1 #Muestra el contenido del fichero file1
```

Para salir pulsar la tecla “q”

less Permite visualizar y moverse a través del contenido de ficheros de texto. Versión más avanzada que el comando more, puesto que permite moverse por todo el texto.

```
less file1 #Muestra el contenido del fichero file1
```

Para salir pulsar la tecla “q”

nano. Editor de texto para consola de texto. Permite crear y editar ficheros de texto plano desde el interprete de comandos.

```
nano #Lanza la aplicación nano con un documento de texto vacío.
nano file1 #Abre el fichero file1 con el editor de texto nano.
```

cp Copia archivos y directorios. Necesita que se especifique el origen u orígenes y el destino.

Sintaxis

```
cp [opciones] fichero destino
cp [opciones] fichero [fichero2 fichero3 ... ] directorio_destino
```

Si el destino no existe, lo crea. Si existe, lo sobrescribe.

```
cp file1 file2 #Copia el fichero file1 en file2. Si no existe file2 lo crea y si existe lo sobrescribe
cp file1 .. #Copia el fichero1 en el directorio anterior
cp file1 file2 file3 practicas #Copia los ficheros file1 file2 y file3 en el directorio practicas.
```

Opciones

- r -R** Permite copiar un directorio y su contenido de manera recursiva.
- i** Pide confirmación al usuario antes de sobrescribir un fichero.
- f** Borra los ficheros destinos existentes en caso necesario (sobrescribir)
- a** Preserva la estructura y atributos originales en la medida de lo posible

mkdir Crea directorios

Sintaxis

```
mkdir [opciones] directorio1 [directorio2 ...]
```

```
mkdir prueba #Crea el directorio prueba
```

Se pueden especificar varios directorios a crear en un mismo comando.

```
mkdir dir1 dir2 dir3 #Crea los directorios dir1 dir2 y dir3
```

Opciones:

-p Permite crear los directorios padres necesarios para crear un directorio determinado.

```
mkdir -p /home/pepe/practicass # Creará las directorios necesarios para crear dicho directorio
```

rmdir Elimina directorios

Sintaxis

```
rmdir [opciones] directorio1 [directorio2 ...]
```

Solo elimina los directorios si están vacíos. En caso contrario, da un error por pantalla.

-p permite eliminar los directorios padres necesarios para eliminar completamente una rama de directorios.

```
rmdir -p /home/pepe/practicass
```

#Elimina todos los directorios necesarios para terminar con el directorio home

mv Mueve (renombra) ficheros.

Sintaxis

```
mv [opciones] origen destino
```

```
mv [opciones] origen... destino
```

Parte de la información obtenida a partir de los manuales de los comandos de GNU/Linux expuestos

Se encarga de mover el fichero o ficheros origen al fichero o directorio destino.

Si hay más de un fichero origen, el destino debe ser obligatoriamente un directorio.

Si el fichero destino existe, lo sobrescribe. Si no existe, lo crea. (se produce un renombrado).

Se pueden mover directorios enteros (renombrar).

```
mv file1 file2 #Renombra el fichero file1 a file2
```

```
mv dir1 dir2 #Renombra el directorio dir1 al dir2
```

```
mv file1 dir1 #Mueve el fichero file1 al directorio dir1 (si éste existe, sino crea el fichero dir1)
```

```
mv file1 file2 file3 dir1 #Mueve los ficheros file1, file2 y file3 al directorio dir1
```

Opciones

-f No pide confirmación para llevar a cabo la operación

-i Pide confirmación para llevar a cabo la operación

rm Borra ficheros o directorios

Sintaxis

```
rm [opciones] fichero...
```

Borra los ficheros proporcionados por parámetro

```
rm file1 file2 file3 #Elimina los ficheros file1, file2 y file3
```

Para borrar directorios, necesita que se especifique la opción -r (borrado recursivo)

```
rm -r dir1 #Elimina el directorio dir1 y todo su contenido.
```

Opciones

- i Pide confirmación
- f No pide confirmación del borrado

tail Muestra la última parte de ficheros

Sintaxis

```
tail [opciones] [ficheros]
```

Muestra las 10 últimas líneas de cada fichero pasado por parámetro por la salida estándar.

```
tail file1 #Muestra las 10 últimas líneas del fichero file1  
tail file1 file2 #Muestra las 10 últimas líneas del fichero file1 y file2.  
tail -n 5 file1 #Muestra las 5 últimas líneas del fichero file1
```

head Muestra la primera parte de ficheros

Sintaxis

```
head [opciones] [ficheros]
```

Muestra las 10 primeras líneas de cada fichero pasado por parámetro por la salida estándar.

```
head file1 #Muestra las 10 primeras líneas del fichero file1  
head file1 file2 #Muestra las 10 primeras líneas del fichero file1 y file2  
head -n 5 file1 #Muestra las 5 primeras líneas del fichero file1
```

grep Muestra las líneas que concuerdan con un patrón dado.

Sintaxis

```
grep [opciones] patron [fichero...]
```

```
grep lista file1
```

#Muestra las líneas del fichero donde aparezca la palabra lista, aunque forme parte de otra palabra.

Opciones.

- e patron,** permite determinar un patrón. Es bastante útil si el patrón a buscar comienza con "-".
- i** No tiene en cuenta la diferencia entre mayúsculas y minúsculas.
- v** Invierte la selección de las líneas, mostrando las que no correspondan con el patrón.
- f** Fichero permite determinar un fichero donde estén todos los patrones.

find Permite encontrar un archivo que cumpla con determinadas características.

Lleva a cabo una búsqueda del fichero determinado tomando como base una expresión que ayuda a encontrarlo

Sintaxis

```
find [ruta..] [expresión]
```

```
find / -name a*a #Busca cualquier fichero que comience con a y termine con a
```

Comodines

Linux permite el uso de comodines, que son caracteres especiales que permiten definir el nombre de ficheros o directorios en base a patrones.

*	Cualquier nombre, por ejemplo: *.jpg --> Ficheros cuyo nombre termine en ".jpg"
?	Cualquier carácter, por ejemplo: *.?pg --> Ficheros cuyo nombre termine en "." y cualquier carácter seguido de "pg"

Estos comodines, pueden utilizarse como parte de la ruta de cualquier fichero o directorio.

```
ls *.jpg
```

#Muestra todos los ficheros que terminen por ".jpg", es decir con extensión jpg.

```
tail -n 5 *.???
```

#Muestra las cinco últimas líneas de cualquier fichero que tenga una extensión de tres caracteres.

Tuberías

Se trata de una de las características más potentes del interprete bash. Permite que la salida de un comando sea la entrada de otro comando.

```
ls -l | tail -n 5
ls -l | head -n 5
ls -l | grep lista
ls -l | grep lista | tail -n 5
```

Redirecciones

Permite que la salida estándar provocada por un comando se redirija a un fichero. Existen dos tipos de redirecciones.

C o m a n d o > fichero	Redirección de sobrescritura.
----------------------------	-------------------------------

C o m a n d o >> fichero	Redirección de adicción.
-----------------------------	--------------------------

```
ls -r > lista.txt #Crea o sobrescribe el fichero lista.txt escribiendo el resultado del comando ls -r
ls >> lista.txt #Crea o añade al final del fichero lista.txt el resultado del comando ls.
```

Permisos

Para mostrar los permisos de archivos podemos utilizar el comando ls con la opción -l

```
ls -l #Muestra información extra (permisos, propietario, grupo, fecha, tamaño) de ficheros.
```

```
ls -lh writer.pdf
-rw-r--r-- 1 jaws users 376K 2005-04-26 21:03 writer.pdf
```

Tipo y permisos	T	propietario	grupo	fecha y hora ultima modificación	nombre del fichero
-rw-r--r--	1	jaws	users	2005-04-26 21:03	writer.pdf

T: Representa el número de enlaces del fichero.

Tipo y permisos	1	2	3	4
	-	rwX	rwX	rwX

- 1: Tipo de archivo
- 2: Permisos para el **propietario**.
- 3: Permisos para usuario miembro del **grupo** del archivo.
- 4: Permisos para el **resto de usuarios**.

Tipo	Descripción
-	archivo normal
d	Directorio
l	Enlace simbólico
c	Dispositivo de carácter (archivo especial)
b	Dispositivo de bloque (archivo especial)
s	Socket (proceso de escucha de red)
p	Canalización con nombre

El indicador de propietario y grupo de los archivos y directorios, permite establecer distintos permisos de acceso para los distintos usuarios del sistema: propietario, grupo y resto de usuarios.

También se pueden expresar los permisos con un número de tres dígitos en octal. Cada dígito octal proporciona información de permisos de propietario, grupo o resto de usuarios.

0	-- -	3	- wx	6	rw -
1	-- x	4	r- -	7	rw x
2	- w-	5	r- x		
750			rwxr-x---		

chmod Permite establecer los permisos de acceso de ficheros y directorios.

Sintaxis

```
chmod [opciones] modo fichero...
```

Opciones

- R Cambia recursivamente los permisos de los directorios y sus contenidos.
- c Muestra un mensaje mostrando los ficheros que se han visto afectados.

modo [ugoa] [[+|=][rwxXstugo]][,...]

Valor	Descripción
u	Usuario propietario
g	Otros usuarios pertenecientes al grupo del fichero
o	Resto de usuarios del sistema
a	Todos los usuarios (propietario, pertenecientes al grupo y resto)
+	Permisos se añaden a los existentes
-	Permisos se eliminan de los existentes
=	Establece solo los permisos definidos
r	Permiso de lectura
w	Permiso de escritura o modificación
x	Permiso de ejecución o de paso para directorios
X	Ejecución si el fichero
s	Establece el identificador de usuario o grupo (UID o GID) a los del propietario del archivo cuando éste está en ejecución. Sobre un directorio determina que el propietario y grupo de los archivos creados serán el del directorio en lugar del usuario que lo cree.
t	Activa el bit de persistencia. Se usa en directorios. Solo el propietario de un archivo puede eliminar el archivo.
u	Permisos que el usuario propietario tiene
g	Permisos que el grupo tiene
o	Permisos que el resto de usuarios tiene

Establecer el identificador de usuario o grupo permite que un archivo se pueda ejecutar como si lo estuviera ejecutando su propietario y no quien realmente lo lanzo.

```
chmod go+rx file1
```

#Añade los permisos de lectura y ejecución sobre el fichero file1 para los usuarios del grupo y el resto de usuarios.

```
chmod a=rx file1
```

Establece los permisos de lectura y ejecución sobre el fichero file1 a todos los usuarios. Se eliminan los permisos que tuvieran.

chown Cambia el propietario (y grupo) de archivos y directorios. (Solo superusuario)

Sintaxis

```
chown [opciones] usuario[:grupo] ficheros...
```

Opciones.

-R Cambia recursivamente el propietario de directorios y sus contenidos.

```
chown pepe file1 file2 dir3 #Cambia el propietario de file1, file2 y dir3 al usuario pepe.
```

```
chown pepe:users file1 #Cambia el propietario y grupo de file1 al usuario pepe y grupo users.
```

```
chown -R pepe dir3 #Cambia el propietario del directorio dir3 y todo su contenido al usuario pepe.
```



chgrp Cambia el grupo al que pertenecen los archivos y directorios. (Solo superusuario)

Sintaxis

```
chgrp [opciones] usuario ficheros...
```

Opciones

-R Cambia recursivamente el grupo de directorios y sus contenidos.

Usuarios y grupos

who Muestra los usuarios que tienen una sesión iniciada.

su Permite *convertirse* en otro usuario dentro de una sesión iniciada. Si se utiliza sin parámetro trata de convertirse en el superusuario (root)

whoami Muestra el usuario (identificador efectivo) con el que se está trabajando.

groups Muestra el grupo primario y el resto de grupos a los que pertenece un usuario.

```
groups [usuario...]
```

Si no recibe ningún parámetro, toma como usuario al usuario de la sesión actual.

adduser Añade usuarios al sistema (y grupos). Permite añadir usuarios de manera interactiva, contestando a una serie de preguntas. También permite usar argumentos que definan la operación.

useradd y **grpadd** Añaden usuarios y grupos al sistema.

addgroup Añade grupos al sistema a través de opciones interactiva o a través de parámetro.

passwd Cambia la contraseña del usuario actual.

Archivos y directorios relacionados con la gestión de usuarios y grupos

/etc/passwd Almacena información sobre los usuarios del sistema. En cada línea de dicho fichero se almacena información para un usuario distinto. Esta información está dividida en campos, separados por el carácter ":". Ver apuntes de "Información de cuentas de usuario en Linux" para completar.

```
nombre de la cuenta:contraseña:UID:GID:datos usuario:home:interprete
```

/etc/shadow Almacena información acerca de las claves de los usuarios del sistema. Se utiliza como medida de seguridad complementando al fichero `/etc/passwd`. En cada línea de dicho fichero se almacena información para un usuario distinto. Esta información está dividida en campos, separados por el carácter ":". Ver apuntes de "Información de cuentas de usuario en Linux" para completar.

/etc/group Almacena información sobre los grupos de usuarios en el sistema. En cada línea de dicho fichero se almacena información sobre cada uno de los grupos en el sistema. La información está dividida en campos separados por el carácter ":".

```
nombre del grupo:contraseña:GID:Lista de usuarios (separada por comas)
```

En la lista de usuarios aparecen los usuarios de dicho grupo cuyo grupo principal es distinto al grupo.

/etc/login.defs Almacena las características predeterminadas para las cuentas de usuario en el sistema.

/etc/skel Directorio que almacena el "perfil" predeterminado para las cuentas de usuario creadas en el sistema. Al crear una cuenta de usuario a través de las herramientas proporcionadas por el SO, se utiliza como base el perfil almacenado en este directorio, copiándose los archivos y directorios que contiene a los directorios principales o de inicio de los nuevos usuarios.

archivos de configuración Se trata de archivos de configuración de las aplicaciones de los usuarios. Comienzan por `.`, de manera que el sistema operativo, por defecto, no los muestra.

```
ls .* #Muestra todos los archivos que comiencen por . en el directorio actual.
```

Gestión de procesos

ps Muestra información sobre los procesos en ejecución en el sistema en el instante en el que se ejecuta.

```
ps #Muestra información de los procesos en ejecución de la sesión actual
```

```
ps -A #Muestra información de todos los procesos en ejecución
```

top Muestra información en tiempo real (actualizándose) de los procesos en ejecución del sistema. La información se actualiza en pantalla cada cierto periodo de tiempo.

kill Envía una señal a uno o varios procesos. Esta señal puede ser una señal de terminación (-9). Para identificar al proceso se utiliza su PID (Identificador de Proceso).

```
kill -9 1000 #Envía una señal de terminación al proceso 1000
```

```
kill -l #Muestra todas las señales disponibles.
```

killall Termina con los procesos que tengan el nombre que se pasa por parámetro.

Fin de sesión, reinicio y apagado (terminal)

login Comienza una nueva sesión en el sistema

logout y **exit** Termina la sesión actual.

reboot Reinicia el sistema (hardware y software)

halt Apaga el sistema (hardware y software)

shutdown Permite reiniciar o apagar el sistema completo o tan solo reiniciar el S.O.

Sintaxis

```
shutdown [-t segundos] [opciones] tiempo [mensaje de alerta]
```

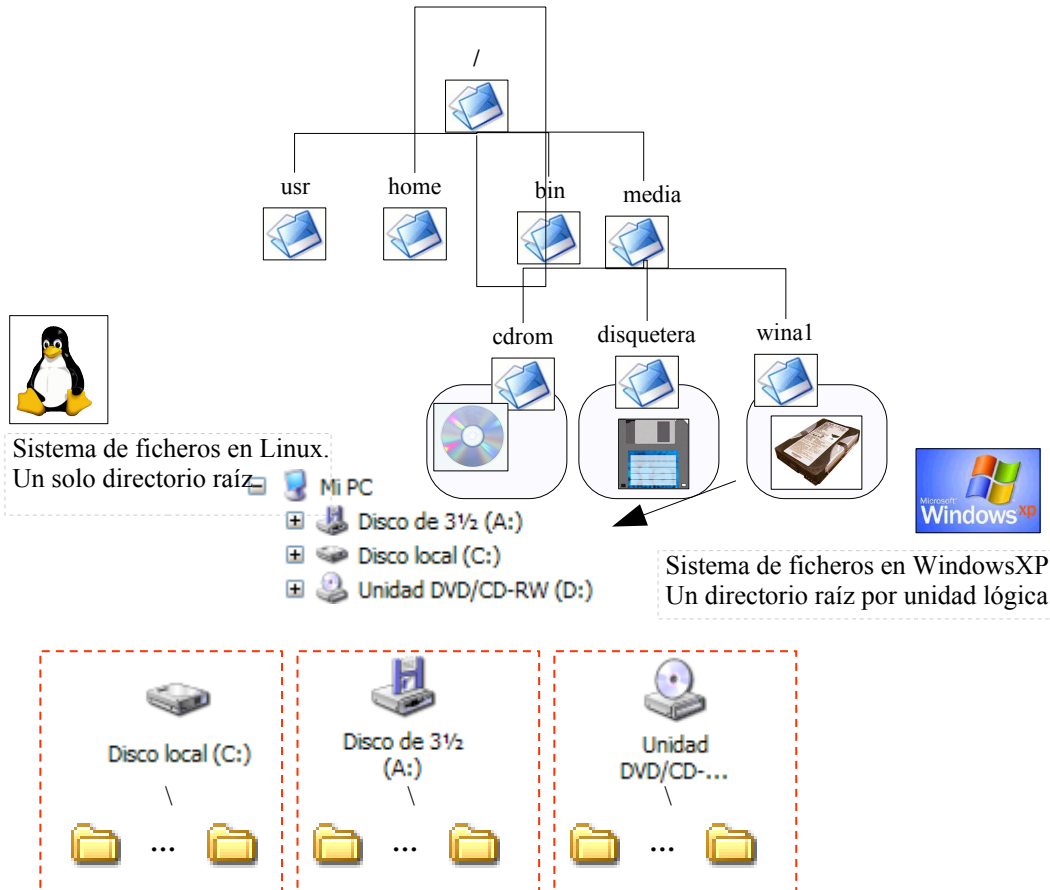
```
shutdown "now" Cierre del sistema por mantenimiento #Reinicia el sistema ahora con un mensaje.
```

```
shutdown 00:05 #Reinicia el sistema en cinco minutos
```

4. -Sistema de ficheros en Linux

La estructura de directorios de Linux está basada en una jerarquía en forma de árbol, similar a la utilizada en MS DOS o Windows XP. Sin embargo, presenta un par de diferencias importantes con respecto a estos sistemas operativos:

- 1) Todo el sistema de archivos en Linux se encuentra incluido en un único árbol de directorios con un único directorio principal o raíz, el directorio /. Este directorio es el padre de todos los directorios del sistema.
- 2) En un sistema de archivos Linux solo existen ficheros y directorios



No existe el concepto de unidades lógicas de CD, DVD, disquetera o discos duros como en Windows, donde cada unidad tiene su propio directorio raíz. En Linux, se puede acceder a estos elementos montándolos en un directorio y accediendo a ellos como si de un directorio más del sistema de ficheros se tratase.

Directorios importantes

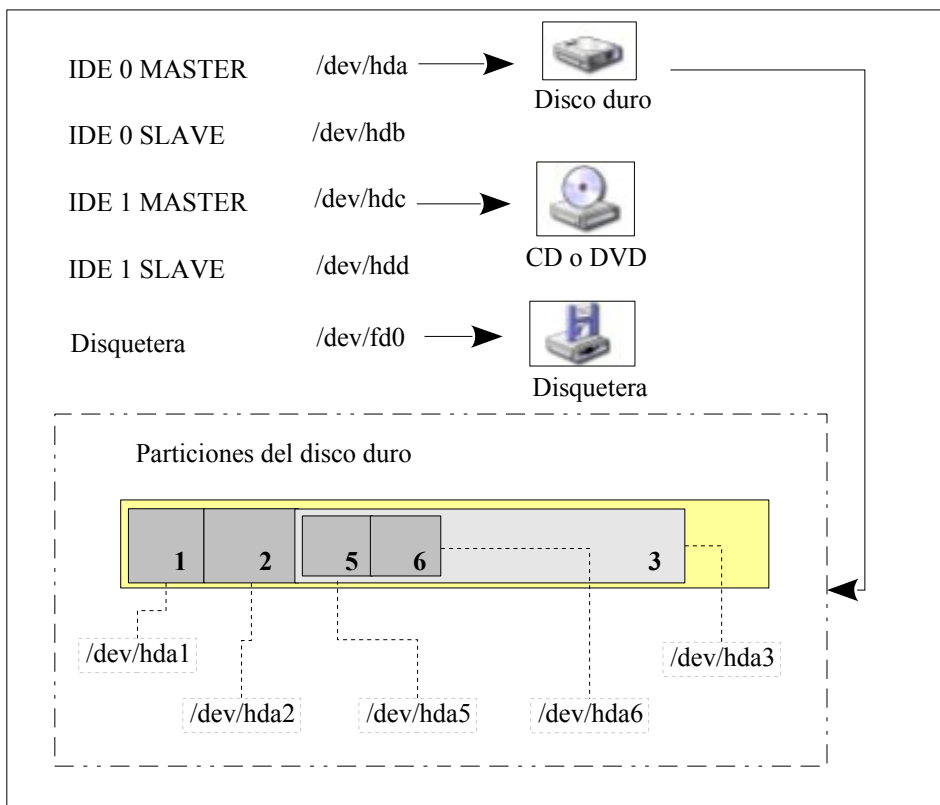
Existe un convenio para definir la organización básica de la estructura de archivos y directorios de un sistema Linux. Proyecto FHS (Estándar de jerarquías de sistemas de ficheros) trata de evitar un caos de ficheros y directorios.

Directorios comunes en todos los S.O. Linux, que tienen un objetivo determinado y común. A continuación vamos a ver algunos de estos directorios.

dir	Contenido
bin	Aplicaciones binarias ejecutables. Herramientas del sistema, aplicaciones de usuario, etc.
boot	Elementos necesarios para llevar a cabo el proceso de arranque del sistema operativo, entre ellos el gestor de arranque grub.
dev	Todas las definiciones de dispositivos conocidos por el sistema operativo. Estas definiciones están representadas por ficheros especiales.
etc	Ficheros relacionados con la configuración del sistema. Puede contener subdirectorios que contienen ficheros de configuración específicos con una finalidad común.
home	Los datos de los usuarios del sistema. Contiene un directorio por cada usuario en el sistema, su directorio principal o de inicio, donde guarda sus datos y aplicaciones.
lib	Las librerías o bibliotecas comunes que puedan requerir las aplicaciones del sistema.
lost+found	Almacena i-nodos perdidos que pueden ser utilizados por herramientas del sistema para recuperar archivos y corregir errores.
mnt	Directorio que se proporciona como base para los puntos de montaje utilizados en el sistema de ficheros. No es obligatorio utilizarlo como base de los puntos de montaje
opt	Directorio donde se almacenan los complementos o plugins de aplicaciones.
proc	Contiene, en archivos especiales, información detallada sobre el sistema, como procesos en ejecución o sobre el hardware del Sistema.
root	Directorio principal o de inicio (HOME) del usuario root. A diferencia del resto de usuarios del sistema, el directorio principal del usuario root no está dentro del directorio home.
sbin	Contiene utilidades de administración y gestión del sistema. (fdisk, parted, mkfs, etc.)
tmp	Almacena archivos temporales de las aplicaciones.
usr	Almacena todos los datos, de solo lectura, que serán compartidos por los usuarios del sistema. (documentación /usr/doc, código fuente del núcleo /usr/src, etc.)
var	Su finalidad es mantener información susceptible de cambiar (variable). Algunos ejemplos son, colas de impresión, de correo, etc.

Dispositivos: ficheros especiales

IDE	FD	SCSI / SATA / USB	CD-ROM	Varios
/dev/hda /dev/hdb /dev/hdc /dev/hdd	/dev/fd0	/dev/sda /dev/sdb... /dev/sc0	/dev/cdrom /dev/cdrom0 /dev/cdrom1 /dev/dvd	Ratón Primer puerto serie Segundo puerto serie Primer puerto paralelo Terminal virtual (salida)

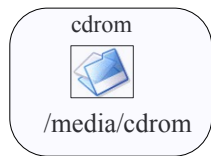


En muchas ocasiones estos dispositivos son enlaces simbólicos a otros dispositivos. Comprobar que es realmente el fichero /dev/cdrom utilizando el comando file.

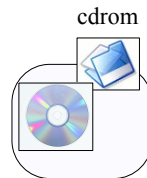
Sistema de ficheros: montaje y desmontaje dispositivos

Para poder acceder a los datos almacenados en un dispositivo, y más concretamente a su sistema de ficheros, es preciso montarlo previamente.

Directorio normal
(punto de montaje)



CD ROM montado en directorio



`mount /dev/cdrom /media/cdrom`



`umount /media/cdrom`



Dispositivo



`/dev/cdrom`

mount Monta un sistema de ficheros.

```
mount # Muestra los sistemas de ficheros montados actualmente.
mount -a # Monta todos los dispositivos mencionados en /etc/fstab
mount dispositivo # Monta el dispositivo en el directorio definido en /etc/fstab
mount directorio # Monta el dispositivo enlazado con el directorio definido en /etc/fstab
mount [opciones] dispositivo directorio # Monta el dispositivo en el directorio (debe existir)
```

Normalmente es el usuario root el que puede montar dispositivos, sin embargo también puede hacerlo un usuario si así se define en `/etc/fstab`.

umount Desmonta un sistema de ficheros.

```
umount directorio # Desmonta el sistema de ficheros montado en el directorio dado.
umount dispositivo # Desmonta el sistema de ficheros almacenado en el dispositivo.
umount /media/cdrom # Desmonta el sistema de ficheros montado en el directorio cdrom.
```

Solo el usuario que ha montado el sistema de ficheros podrá desmontarlo.

fichero `/etc/mtab` Almacena información sobre los sistemas de ficheros montados actualmente en el sistema. Se actualiza en tiempo real.

fichero `/etc/fstab` Almacena información sobre los sistemas de ficheros que el SO debe montar en el inicio o carga del sistema. Cada línea determina información sobre el dispositivo, el punto de montaje (directorio) y las opciones de montaje.

file Determina el tipo de un fichero.

ln Crea un enlace a un fichero o directorio (similar a los accesos directorios de Windows). Existen dos tipos de enlaces:

1. *simbólicos o blando*. Son similares a los accesos directos de Windows. Se trata de un fichero que referencia a otro fichero (contiene un enlace con la ruta de otro fichero).

2. **físico o duro**. Crea un nuevo nombre de archivo que apunta a los mismos datos. Ambos archivos, original y enlace duro, comparten la misma información en disco (mismos bloques).

df Informa sobre el espacio utilizado en los sistemas de ficheros.

Sintaxis

```
df [opciones] [fichero..]
```

```
df #Informa del espacio utilizado en todos los sistemas de ficheros montados actualmente.  
df file1 #Informa sobre el sistema de ficheros que contiene al fichero file1  
df /dev/hda1 #Informa sobre el sistema de ficheros que contienen la primera partición de hda  
df -h /dev/hda1 #Informa sobre el sistema de ficheros en hda1 con información más legible
```

du Muestra la cantidad de espacio en disco utilizada por los archivos especificados. Si entre los archivos hay directorios, calcula el espacio ocupado por los archivos contenidos por cada uno de los directorios. Por defecto, el tamaño mostrado es en bloques de 512 Bytes, aunque se puede configurar.

```
du #Informa sobre el espacio utilizado por los archivos en el directorio actual.  
du -k #Igual que el anterior, pero usando bloques de 1024KB  
du -h #Igual que el anterior, pero mostrando información de tamaño más adecuada para humanos.
```

fdisk Permite gestionar las particiones en los discos duros en el sistema. También se puede utilizar para consultar la tabla de particiones de los distintos dispositivos de almacenamiento masivo disponibles.

```
fdisk /dev/hda #Gestiona el disco duro instalado en el primer IDE  
fdisk -l #Muestra información de la tabla de particiones y sale. Toma la información del fichero /proc/partitions  
fdisk -s /dev/hda1 #Muestra el tamaño de la partición en bloques.
```

mkfs Crea un sistema de ficheros Linux en un dispositivo. Se puede utilizar el archivo del dispositivo o el punto de montaje como argumento.

fsck Comprueba y arregla un sistema de ficheros Linux. Si no se especifica el sistema de ficheros a comprobar, tomará la información de los sistemas de ficheros disponibles de /etc/fstab.