

Introducción al sistema operativo GNU/Linux – Parte II

1. Introducción a los archivos	2
2. Tipos de archivos en Linux	2
3. Operaciones con archivos	3
Crear archivos	3
Ver e imprimir el contenido de un archivo	3
Cambiar el nombre a un archivo	4
Copiar y mover archivos	4
Eliminar archivos	4
4. Permisos y atributos en archivos	5
5. Compresión de archivos	6
6. Edición de textos en Linux	7
7. Algunas órdenes básicas	7
8. Comprobación de dispositivos de almacenamiento	9
Montaje y desmontaje de dispositivos Linux	9
Dar formato a dispositivos de almacenamiento	11
Manejo de discos en Linux	11
Cómo acceder a los dispositivos de almacenamiento	11
9. Gestionar hardware del equipo	12

1. Introducción a los archivos

Todo lo que hemos visto respecto de los archivos en Windows es igual para el entorno de Linux: las características de los archivos, operaciones sobre los archivos, etc. Vamos a ver cómo se manejan archivos en Linux, especialmente en entorno comando, ya que el manejo en la interfaz gráfica es igual que en Windows, con pequeñas variaciones.

Es importante destacar que el **nombre** y la **extensión** en Linux **tienen otra consideración diferente**. Son diferentes los archivos cuyo nombre esté escrito en mayúsculas o minúsculas. Los nombres de archivos en Linux tenían originalmente un límite de once caracteres más tres caracteres de extensión, que no es obligatoria.

Las operaciones que se pueden realizar en Linux con los archivos son las mismas que en Windows:

- ✓ **Creación.** Hacer que el fichero exista dentro de la estructura de archivos. Se le dará un nombre y una extensión. Cuando se crea un archivo, este tiene unas características que lo discriminan del resto, como el nombre con o sin extensión.

El formato del nombre dependerá del sistema operativo y del sistema de archivos con el que estemos trabajando. Este dato es asignado por el sistema o por el usuario. El resto de características, como atributos, hora y fecha de creación, última fecha de acceso, última fecha de modificación, clúster asignado y tamaño, son datos asignados por el sistema.

- ✓ **Consulta.** Operación que consiste en acceder a su contenido para verlo.
- ✓ **Actualización.** Modificación del contenido del archivo.
- ✓ **Borrado.** Eliminación total del archivo, dejando libre el espacio que ocupaba.
- ✓ **Renombrado.** Consiste en cambiar el nombre o la extensión al archivo.

2. Tipos de archivos en Linux

En Linux existen, básicamente, cinco tipos de archivos. Son los siguientes:

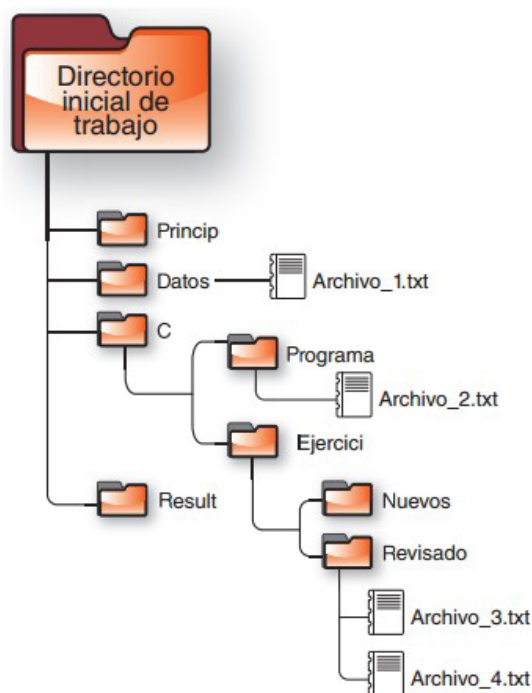
- ✓ **Archivos ordinarios.** Contienen la información con la que trabaja cada usuario. Normalmente son archivos que contienen texto, programas escritos por el usuario en lenguaje C, etc.
- ✓ **Enlaces físicos (vínculos físicos).** No es en sí mismo una clase de archivo. Son segundos nombres asignados a un mismo archivo. Cuando varios usuarios manejan uno de estos segundos nombres, abren, modifican o eliminan lo que hay en el archivo original.
- ✓ **Enlaces simbólicos (vínculos simbólicos).** Son los equivalentes a los accesos directos en Windows.
- ✓ **Directorios.** Son archivos especiales que contienen referencias a otros archivos. Contienen información sobre archivos ordinarios, subdirectorios, vínculos, vínculos simbólicos, etc.
- ✓ **Archivos especiales.** Suelen representar dispositivos físicos, como unidades de almacenamiento, impresoras, terminales, etc. Linux trata los archivos especiales como archivos ordinarios.

3. Operaciones con archivos

Crear archivos

La forma de trabajar con archivos en modo gráfico en Linux es prácticamente igual que en Windows. En primer lugar, veremos cómo podemos crear archivos de texto de forma sencilla.

Verifiquemos que tenemos la estructura de directorios de la figura. Si falta algo, lo crearemos. Además crearemos los archivos indicados.



Ver e imprimir el contenido de un archivo

Para ver el contenido de un archivo **en entorno gráfico**, en primer lugar debemos localizarlo dentro de la estructura de directorios e identificar el icono al que está asociado. Para ver los archivos (en nuestro caso de texto) en Linux, siempre hay que editarlos, y para editarlos podemos utilizar **gedit**, **nano**, etc.

Podemos actuar de varias formas: hacer doble clic sobre el icono que representa el archivo, abrir la aplicación y seleccionar el archivo o arrastrar el archivo hacia el icono que representa la aplicación.

Para imprimir cualquier tipo de archivos tendremos que editar o visualizar el archivo y luego, en la línea de menú, seleccionar la opción de Archivo. En el caso concreto de archivos de texto plano, podremos imprimirlos utilizando el comando **lpr**, que funciona igual que la orden **PRINT** del modo comando de Windows.

Situados en el directorio de trabajo, visualizamos el contenido de **archivo_1.txt**. Ejecutamos:

```
paco@ubuntupaco:~$ cat Datos/archivo_1.txt
```

El comando **cat** de Linux es **similar al comando TYPE** de Windows.

Cambiar el nombre a un archivo

La forma de actuar para cambiar el nombre a archivos en entorno gráfico es la misma que la utilizada para directorios o carpetas.

Vamos a cambiar el nombre del **archivo_1.txt** a **archivo.bas**. Después, poner al archivo el nombre original.

```
paco@ubuntupaco:~$ mv Datos/archivo_1.txt Datos/archivo.bas
```

En Linux, **hay que indicar la ruta del archivo a renombrar, así como la ruta del nuevo archivo renombrado**. Si no especificamos ruta en el archivo destino, se guardará en el directorio en el que estamos situados.

Copiar y mover archivos

En entorno gráfico esta operación se realiza igual que en Windows y de la misma forma que se hace con carpetas. Recordemos que pulsando las teclas **Control** o **Shift** podemos seleccionar múltiples archivos. También podemos realizar la copia mediante las opciones Cortar, Copiar o Pegar de la línea de menú.

Como ejemplo, vamos a copiar todos los archivos de la estructura de directorios a tu directorio personal.

```
paco@ubuntupaco:~$ cp Datos/archivo_1.txt
paco@ubuntupaco:~$ cp C/Programa/archivo_2.txt
paco@ubuntupaco:~$ cp C/Ejercici/Revisado/*.txt
```

No indicamos el destino de la copia pues estamos situados en él. La orden copia los archivos en el directorio actual (directorio activo).

La orden **XCOPY** utilizada en modo comando en Windows no tiene equivalencia en Linux, pero con **cp** **utilizada con la opción -R se obtiene el mismo efecto**. La orden **mv** equivalente a la orden **MOVE** de Windows, mueve uno o varios archivos de un directorio a otro y **permite la utilización de comodines** cuando queremos mover varios archivos a la vez.

Vamos a mover el fichero **archivo_1.txt** del directorio **Datos** al directorio **C** con el mismo nombre. Estamos situados en nuestra carpeta personal de trabajo.

```
paco@ubuntupaco:~$ mv Datos/archivo_1.txt C/
```

Para moverlo con otro nombre, por ejemplo, **nuevo.txt**, la orden es:

```
paco@ubuntupaco:~$ mv Datos/archivo_1.txt C/nuevo.txt
```

Eliminar archivos

Al igual que en Windows, seleccionaremos los archivos a eliminar y pulsaremos la tecla **Supr** o el botón derecho del ratón para elegir la opción **Mover a la papelera**.

Para eliminar los archivos sin pasar por la Papelera, tendremos que ir a las **preferencias de las carpetas**. Esta misma acción se puede realizar pulsando las teclas **Shift** + **Supr** sobre el archivo o archivos seleccionados. Se mostrará un mensaje indicando que si operamos de esta forma el archivo borrado será irre recuperable.

Dejar la estructura de directorios de tal forma que quede igual que la imagen inicial. Mover todos los archivos de la estructura de directorios al directorio C con el mismo nombre, salvo el **archivo_1.txt**, que lo moveremos con el nombre **datos_1.txt** y comprobar el resultado. Realizaremos las siguientes operaciones:

```
paco@ubuntupaco:~$ rm *.*
paco@ubuntupaco:~$ mv Datos/archivo.bas Datos/archivo_1.txt
paco@ubuntupaco:~$ mv Datos/archivo_1.txt C/datos_1.txt
paco@ubuntupaco:~$ mv C/Programa/archivo_2.txt C/
paco@ubuntupaco:~$ mv C/Ejercici/Revisado/*.* C/
paco@ubuntupaco:~$ ls -R
paco@ubuntupaco:~$ rm C/*.*
paco@ubuntupaco:~$ ls -R
```

4. Permisos y atributos en archivos

Los tipos de permisos en archivos son iguales que los utilizados en directorios. Son: **Lectura (r)**, **Escritura (w)** y **Ejecución (x)**. Todo lo visto para el manejo de atributos y permisos sobre los directorios es aplicable a los archivos. La asignación, modificación y eliminación de permisos o atributos sobre los archivos en entorno gráfico se realiza de la misma forma que para los directorios.

Por defecto, los **permisos efectivos en la creación de archivos son 666 - 022 = 644**, o lo que es lo mismo: **rw- r-- r--**.

Como ejemplo, veamos los privilegios o permisos sobre el directorio PRINCIP.

```
paco@ubuntupaco:~$ ls -lad PRINCIP
drwxr-xr-x 2 root root 4096 2008-08-02 13:24 PRINCIP
```

La gran diferencia está entre los permisos de partida que se asignan a un directorio al crearlo y los que se asignan a los ficheros. Si mostramos los permisos de un archivo cualquiera, por ejemplo del archivo_1.txt de nuestra estructura de directorios y ficheros, observamos que los permisos **no son exactamente los mismos**.

```
paco@ubuntupaco:~$ ls -la Datos/archivo_1.txt
-rw-rw-r-- 1 root root 35 2008-08-02 13:38 uno/archivo_1.txt
```

Como se puede apreciar, los permisos de un directorio y de un archivo cuando los creamos son diferentes. Vemos que para **directorios utilizaremos una máscara octal del tipo 755 y para los archivos 644**.

- El número **755** pasado a binario es **111 101 101** y su equivalente a permisos es **rwX r-x r-x**.
- El número **644** pasado a binario es **110 100 100** y su equivalente a permisos es **rw- r-- r--**.

Teniendo en cuenta esto, la asignación, modificación o eliminación de permisos a los archivos funciona exactamente igual que cuando lo hicimos con directorios.

Si por ejemplo, queremos que archivo_1.txt tenga todos los privilegios para todos los tipos de usuarios, podremos hacerlo con alguna de las siguientes órdenes:

```
paco@ubuntupaco:~$ chmod 777 Datos/archivo_1.txt
```

O de esta otra forma:

```
paco@ubuntupaco:~$ chmod ugo+rwX Datos/archivo_1.txt
```

5. Compresión de archivos

La mayoría de los sistemas operativos actuales disponen de herramientas específicas para comprimir archivos, y Linux no es menos. Ya sabemos en qué consiste la compresión de archivos. Recordemos que en Windows comprimíamos archivos con las utilidades WinRAR, WinZip, o con las herramientas del propio sistema. Vamos a ver ahora cómo en Linux se hace esta operación con herramientas propias del sistema y con alguna utilidad extra.

Para ver cómo se comprimen y descomprimen archivos en Linux, tendremos que tener en cuenta algunas consideraciones. Los archivos comprimidos en Linux pueden ser, generalmente, de alguno de estos tipos: **tar**, **zip**, **rar**, **gz** o **bz2**, aunque existen otras extensiones, de las que ahora no nos ocuparemos, ya que estas son las más comunes. Normalmente los **archivos tar** suelen usarse para **comprimir carpetas**, y **el resto para comprimir archivos y/o carpetas**.

Vamos a crear en nuestro escritorio tres archivos planos de texto. Les vamos a dar los nombres **uno.txt**, **dos.txt** y **tres.txt**, e incluyamos algo de contenido para que **el tamaño no sea de 0 bytes**.

En modo comando y para comprimir y descomprimir archivo, utilizaremos los comandos diseñados a tal efecto. Veamos cómo se comprimen los tres archivos anteriores.

a) Para comprimir y descomprimir **con tar un archivo llamado nuevo.tar**.

Comprimir o empaquetar:

```
paco@ubuntupaco:~$ tar -cvf nuevo.tar uno.txt dos.txt tres.txt
```

Descomprimir o desempaquetar:

```
paco@ubuntupaco:~$ tar -xvf nuevo.tar
```

Ver contenido:

```
paco@ubuntupaco:~$ tar -tf archivo.tar
```

b) Para comprimir y descomprimir **con zip/unzip un archivo llamado nuevo.zip**.

Comprimir o empaquetar:

```
paco@ubuntupaco:~$ zip nuevo.zip uno.txt dos.txt tres.txt
```

Descomprimir o desempaquetar:

```
paco@ubuntupaco:~$ unzip nuevo.zip
```

Ver contenido:

```
paco@ubuntupaco:~$ unzip -v nuevo.zip
```

c) Para comprimir y descomprimir con rar un archivo llamado nuevo.rar. Cuando usemos este comando, es **importante tener en cuenta que los modificadores se introducen sin guion**, a diferencia de los comandos anteriores. Por defecto, este compresor no está instalado en Linux.

Comprimir o empaquetar:

```
paco@ubuntupaco:~$ rar a nuevo.rar uno.txt dos.txt tres.txt
```

Descomprimir o Desempaquetar:

```
paco@ubuntupaco:~$ rar x nuevo.rar
```

Ver contenido:

```
paco@ubuntupaco:~$ rar v nuevo.rar
```

```
paco@ubuntupaco:~$ rar l nuevo.rar
```

d) Para comprimir y descomprimir **con gzip el archivo llamado uno.txt**.

Comprimir o empaquetar:

```
paco@ubuntupaco:~$ sudo gzip -9 uno.txt
```

Descomprimir o desempaquetar:

```
paco@ubuntupaco:~$ sudo gzip -d uno.txt.gz
```

Ver contenido:

```
paco@ubuntupaco:~$ sudo gzip -l uno.txt.gz
```

Para analizar la sintaxis de estos compresores de archivos, recomendamos utilizar la **ayuda en línea de Linux**, como **man** o introducir el nombre del comando seguido de **--help**.

6. Edición de textos en Linux

Ya vimos que en Windows, utilizando herramientas como **Wordpad** o **Bloc de notas**, podemos acceder a archivos de texto plano para trabajar con ellos. En Linux también existen editores de texto para poder trabajar con **archivos de texto plano** que no generan ningún tipo de caracteres de control especiales. Para poder editar archivos de texto en Linux utilizaremos alguno de los siguientes editores de texto:

- **vi**. Editor de textos heredado de las primeras versiones de UNIX, que no permite el manejo de las teclas de edición ni el manejo del ratón. Su uso es mínimo hoy por hoy, pero es conveniente conocer su existencia, al menos por cuestiones históricas. Lo ejecutaremos desde la línea de comandos.
- **nano**. Editor similar a vi, pero con mejor funcionalidad a la hora de editar textos planos. Este editor de texto lo podremos ejecutar desde la línea de comandos de Linux.
- **gedit**. Podemos ejecutarlo directamente desde la línea de comandos, o en entorno gráfico desde Aplicaciones > Accesorios > Editor de textos.

7. Algunas órdenes básicas

Al igual que en MS-DOS, algunas de las órdenes sencillas que el usuario puede manejar son las siguientes:

ORDEN	SINTAXIS	FUNCIÓN
clear	clear	Limpia la pantalla en modo comando.
date	date [OPTION]... [+FORMAT] date [-u --utc --universal] [MMDDhhmm[[CC]YY] [.ss]]	Presenta en pantalla el día y la hora que tiene el equipo en ese momento. Muestra, de izquierda a derecha, el día de la semana, mes, día del mes, hora, meridiano y año.
cal ncal	cal [-3jmy] [[month] year] ncal [-jJpwy] [-s country_code] [[month] year] ncal [-Jeo] [year]	Calendario perpetuo que incorpora el sistema. Nos permite ver calendarios de otros años o de un mes determinado.
who	who [options] [le ARG1 ARG2]	Muestra por pantalla una línea por cada usuario que en ese momento esté conectado. Muestra, de izquierda a derecha, el nombre de presentación del usuario, el número de terminal y la fecha y hora de presentación. Una variedad de este comando es who i am , que muestra sólo información referente al usuario.

Para probar estas órdenes ejecutamos un terminal e introducimos la orden con los parámetros y argumentos requeridos. Supongamos que se nos ha olvidado nuestro nombre de usuario (login) o bien que tenemos varias sesiones abiertas, cada una con una cuenta de usuario distinta, y que no sabemos cuál es el usuario de la sesión que estamos usando. Ejecutamos **whoami** (¿quién soy?) que nos indica quién es el usuario actual:

```
paco@ubuntupaco:~$ whoami
paco pts/0 2012-07-23 09:35 (:0.0)
```

Ahora vamos a ajustar la hora del sistema. Supongamos que son las 18 horas y 45 minutos del 25 de diciembre y que queremos actualizar esa información en el sistema. Ejecutaremos:

```
paco@ubuntupaco:~$ date 12251845
```

Ahora borraremos la pantalla:

```
paco@ubuntupaco:~$ clear
```

Y volveremos a ejecutar **date** para comprobar que la actualización ha sido correcta:

```
paco@ubuntupaco:~$ date
lun dec 25 18:45:35
```

Por último, queremos saber si el año 2002 fue bisiesto. Basta comprobar si el mes de febrero de 2002 tenía 28 o 29 días y para ello consultaremos el mes de febrero de dicho año:

```
paco@ubuntupaco:~$ cal 2 2002
febrero 2002
lu ma mi ju vi sa do
          1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28
```

Más comandos sencillos en Linux:

ORDEN	SINTAXIS	FUNCIÓN
logname	logname	Esta orden muestra el contenido de la variable logname. Esta variable contiene el nombre del usuario conectado al ordenador, es decir, el nombre de usuario con el que hemos conectado con el sistema Linux.
uname	uname [-parámetros]	Muestra información como el tipo de sistema operativo, versión, tipo de ordenador desde el que nos conectamos, etc.
id	id [-parámetros] [username]	Muestra el número de identificación y el grupo al que pertenece el usuario.
passwd	passwd [-o] [-q] [-v] [usuario [contraseña]] passwd -f [cambio de nombre completo del usuario] passwd -s [parámetros para el cambio de shell]	Esta orden se introduce sin parámetros o modificadores. Permite al usuario cambiar o asignar una contraseña, siempre que el administrador lo permita.

Vamos a cambiar nuestro password:

```
paco@ubuntupaco:~$ passwd
```



```
Cambiando la contraseña para paco.  
(actual) contraseña de UNIX:  
Introduzca la nueva contraseña de UNIX:  
Vuelva a escribir la nueva contraseña de UNIX:  
passwd: contraseña actualizada correctamente
```

Con la instrucción **uname** mostramos información del sistema en el que estamos identificados:

```
paco@ubuntupaco:~$ uname -a  
Linux ubuntupaco 2.6.24-16-generic #1 SMP Thu Apr 10 13:23:42 UTC 2008  
i686 GNU/Linux
```

Para conocer el nombre del usuario identificado en el sistema, ejecutamos:

```
paco@ubuntupaco:~$ logname  
paco
```

Respecto de la identificación del usuario y de los componentes que utiliza en el sistema, ejecutamos la orden:

```
paco@ubuntupaco:~$ id  
uid=1000 (paco) gid=1000 (paco) grupos=4 (adm) , 20 (dialout) , 24 (cdrom) , 25 (floppy) , 29 (audio) , 30 (dip) , 44 (video) , 46 (plugdev) , 107 (fuse) , 109 (lpadmin) , 115 (admin) , 1000 (paco)
```

8. Comprobación de dispositivos de almacenamiento

El método de comunicación entre un sistema Linux y los dispositivos físicos de almacenamiento de datos es a través de archivos. Estos archivos se utilizan exactamente igual que cualquier otro archivo del sistema; la diferencia está en que con ellos interactuamos con el hardware de la máquina. Cada tipo de dispositivo tiene reservada una cantidad de archivos con un nombre concreto que solo se diferencian en un número.

En Linux, los archivos de los dispositivos, ya sean para discos duros, monitores, disquetes, unidades de DVD, etc., se encuentran dentro del **directorio /dev**. Así, los discos duros físicos están asociados a archivos de dispositivos del tipo **/dev/sd**, y cada disco en particular se diferencia **del otro en un número o una letra**.

Por ejemplo, si disponemos de varios discos duros en nuestro equipo, el primero se llamará **/dev/sda**, el segundo **/dev/sdb**, etc. Y si tenemos varias particiones en el mismo disco, la primera se llamará **/dev/sda1**, la segunda **/dev/sda2**, etc. En general, los archivos especiales dentro del directorio **/dev** asociados a los dispositivos de almacenamiento son los siguientes:

DESCRIPCIÓN	DISPOSITIVO ASOCIADO
Dispositivos USB externos	sdb, sdc , etc.
Disquetes	fd0, fd1
Sistema de archivos principal	sda
Discos duros	sdb, sdc , etc.
Unidades de DVD o CD-ROM	scd0, scd1 , etc.

Montaje y desmontaje de dispositivos Linux

En Linux, para poder trabajar con dispositivos tales como discos duros, unidades de CD-ROM, DVD, grabadoras, etc., es necesario **montarlos primero en el sistema** o de lo contrario no habrá forma de

comunicarse con ellos. Montar un dispositivo consiste en **añadirlo a la estructura de archivos del sistema**. El **enlace** no es ni más ni menos que **indicar en qué punto del sistema de archivos colocamos la raíz del sistema de archivos del dispositivo** que se está montando.

Por fortuna, el montaje de dispositivos, al menos en esta versión de Ubuntu, está actualmente automatizado. Cuando introducimos un disco en la unidad de DVD o CD-ROM, o insertamos un dispositivo USB, aparecen automáticamente en el escritorio iconos que representan los nuevos dispositivos.

Las órdenes **mount** y **umount** se utilizan para montar dispositivos de almacenamiento y enlazarlos al sistema principal de archivos.

Vamos a montar y desmontar un dispositivo de almacenamiento extraíble en nuestro equipo. Antes de introducir el dispositivo USB en el equipo, analizaremos el **directorio /media** y anotaremos lo que contiene. Analizado esto, introduciremos el pen-drive. Como observaremos, nada más introducir el USB en el equipo, automáticamente aparece en el escritorio y en el lanzador un icono representativo del dispositivo añadido. Si hacemos doble clic sobre él, accederemos a la información que el mismo contiene.

Vamos ahora a desmontar el dispositivo y a montarlo en otro punto del sistema de archivos. Por defecto, dentro de la carpeta **/media**, observaremos que aparecen varios directorios, y entre ellos, uno nuevo. Este nuevo directorio será el nombre que tenga nuestro USB, si es que lo tiene, o un conjunto de números y letras asignado por el sistema, siempre y cuando no tenga etiqueta. Para desenlazar un dispositivo de almacenamiento del sistema principal de archivos ejecutaremos la orden siguiente:

```
paco@ubuntupaco:~$ sudo umount /media/[nombre]
```

Siendo **nombre** el directorio en el que se ha montado nuestro USB. Los directorios floppy y floppy0, entre otros, hacen referencia a los puntos de montaje específicos para disquetes, pero hoy en día no se utilizan.

Nosotros solamente prestaremos atención a un nombre que puede representar a nuestro dispositivo USB. Es importante tener en cuenta que si el nombre de nuestro dispositivo tiene espacios en blanco, tendremos que introducirlo entre comillas (las que están debajo de cerrar interrogación), tal y como indica la orden:

```
paco@ubuntupaco:~$ sudo umount /media/`nombre de dispositivo`
```

Una vez desmontado, lo vamos a montar, pero manualmente. Para ello, **en el directorio /media crearemos, por ejemplo, un directorio con el nombre miusb**.

```
paco@ubuntupaco:~$ cd /media
paco@ubuntupaco:/media$ mkdir miusb
```

Una vez creado el directorio procederemos a montar el dispositivo USB en este lugar. Para ello es importante saber que los dispositivos de almacenamiento, son dispositivos que son reconocidos por el sistema como sda, sbd, sdc, etc. Dentro de cada uno de estos dispositivos podremos tener varias particiones, que se reconocerán como sda1, sdb2, sda3, etcétera.

Lo que tiene que quedar claro es que **sda es en sí todo el dispositivo**, y cada **sda1, sdb2, sda3 es una partición dentro del dispositivo**. Por este motivo, cuando montemos dispositivos, **no lo montaremos completo. Montaremos cada una de las particiones que tenga el mismo, que puede ser una sola o varias**. Partiendo de la base de que nuestro USB solamente tiene una partición, el montaje lo tendremos que hacer para la partición única del dispositivo.

Si solo disponemos de un disco duro este será el sda y sus particiones sda1, sda2, etc. Es por lo que nuestro USB se habrá reconocido como sdb y la partición del mismo como sdb1. Si disponemos de un segundo disco duro en el sistema, este será reconocido como sdb, y las particiones numéricamente tantas como tenga el disco. En este otro caso, **nuestro USB será el sdc**. Suponiendo que tenemos dos discos duros en el sistema, para montar el dispositivo USB, que dispone de una sola partición, ejecutaremos la siguiente orden.

```
paco@ubuntupaco:~$ sudo mount /dev/sdc1 /media/miusb
```

Comprobaremos que de nuevo aparece en el escritorio el icono que indica que el dispositivo está habilitado. Podremos desmontarlo y montarlo tantas veces como queramos, pero **un dispositivo no debe estar montado en más de un sitio a la vez**.

Para discos duros se procede de la misma forma.

Dar formato a dispositivos de almacenamiento

Montado el dispositivo de almacenamiento, podemos proceder a darle formato. Para ello, ejecutaremos alguna de las órdenes que Linux nos proporciona.

Si lo que queremos es dar formato a un dispositivo de almacenamiento, como nuestro USB, tendremos que utilizar el comando **mkfs**. Para formatear un dispositivo, **este tiene que haber sido desmontado previamente**. Es entonces cuando ejecutaremos una orden como la siguiente, teniendo en cuenta que lo que se formatean son particiones y no todo el dispositivo.

```
paco@ubuntupaco:~$ sudo mkfs.vfat /dev/sdc1
```

Pasado algún tiempo, el dispositivo queda totalmente limpio de información y dispuesto a ser montado o extraído, o lo que corresponda.

Para formatear dispositivos de almacenamiento en entorno gráfico es necesario tener instalada una utilidad llamada **gparted**, específica para la gestión de discos, que veremos más detenidamente.

Manejo de discos en Linux

En Windows tratamos algunas operaciones que se podían realizar con los discos duros. Entre otras, vimos la opción del Desfragmentador de discos. En Linux no es necesario realizar esta operación, ya que el sistema de archivos de Linux no se fragmenta como el de Windows.

Lo que sí es necesario de vez en cuando, como ocurre en los sistemas Windows, es **comprobar la integridad de los discos duros**, tanto lógica como físicamente. De vez en cuando comprobaremos la integridad del sistema de archivos, y muy ocasionalmente podremos verificar la estructura física del disco, caras, pistas y sectores, para evitar posibles pérdidas de información.

Utilizaremos la orden **fsck**, que sirve para **recuperar archivos mal referenciados por la tabla de i-nodos del sistema de archivos**. Esta orden normalmente se ejecuta con el **modificador** o parámetro **-a**.

Otra operación que podemos hacer en Linux, de forma similar a Windows, es convertir el sistema de archivos de un disco. En el caso que nos ocupa, comprobamos el dispositivo USB ejecutando la orden:

```
paco@ubuntupaco:~$ sudo fsck /dev/sdc1 -a
```

Cómo acceder a los dispositivos de almacenamiento

Como ya hemos comentado, a excepción de la disquetera, que no viene montada, el resto de dispositivos se monta de forma automática. En concreto, los discos duros se montan durante el proceso de arranque del sistema. Los dispositivos de DVD y CD-ROM también están montados por defecto, de tal forma que cuando se introduce un disco, el sistema lo reconoce automáticamente.

En cuanto a los dispositivos de almacenamiento externo de tipo USB, indicar que **se montan automáticamente cuando se insertan en el sistema**. Los representa el icono, que aparece en el escritorio directamente al incorporar cualquier dispositivo de este tipo en el sistema, junto con el nombre del volumen o etiqueta. También sabemos que en Equipo, si hacemos doble clic en cada unidad de almacenamiento, podemos acceder a su contenido sin más.

Otra cosa es acceder al contenido de los dispositivos de almacenamiento desde el terminal de Linux. En concreto, para acceder a las unidades de almacenamiento removibles, una vez montadas las que fueran necesarias, accedemos al **directorio /media** y dentro de él a `floppy0` para referenciar el contenido del disquete.

Si queremos acceder a un dispositivo USB insertado en el sistema, y suponiendo que el nombre (**label**) que tiene es **miusb**, accederemos a él tecleando:

```
paco@ubuntupaco:~$ ls -la /media/miusb
```

Es evidente que para acceder a la unidad de disco donde tenemos instalado el sistema operativo lo haremos desde el directorio / de nuestro sistema de archivos.

Puede ocurrir que tengamos más de un disco duro u otras particiones en el disco principal además de las dos necesarias. En este caso, cuando particionemos los discos y les demos formato, operación que veremos en la unidad siguiente, las unidades serán accesibles directamente desde el entorno gráfico. Para poder acceder desde el terminal al contenido de otros discos duros instalados recientemente en el sistema, es necesario montarlos sobre algún directorio, al igual que ocurría con los disquetes o dispositivos USB.

Por ejemplo, si añadimos a nuestro sistema operativo un nuevo disco con dos particiones, tendremos que tener en cuenta que estas particiones se asocian al **dispositivo sdb**, ya que el **sda** es el **disco principal**.

En concreto, en el directorio **/dev**, el dispositivo `sdb1` será el asociado a la primera partición del segundo disco y `sdb2` el dispositivo asociado a la segunda partición de este nuevo disco. Estas asociaciones del disco físico con los dispositivos lógicos del sistema se hacen de forma automática.

9. Gestionar hardware del equipo

El **soporte plug and play** que ofrecen las **últimas versiones** de Ubuntu es muy bueno. **La mayoría del hardware del mercado es detectado e instalado automáticamente** por el sistema operativo. Si Ubuntu no reconoce automáticamente el hardware instalado, podemos empezar a tener problemas. De hecho, no todo el hardware del mercado se puede instalar en este sistema operativo.

Es sencillo obtener información del hardware instalado en nuestro equipo. Para ello, usaremos la aplicación **hardinfo**. Ejecutaremos los siguientes comandos y en modo gráfico podremos ver la información:

```
paco@ubuntupaco:~$ sudo apt-get install hardinfo
paco@ubuntupaco:~$ hardinfo
```

Si lo que queremos es añadir controladores de dispositivos hardware ya instalados o simplemente queremos actualizar los controladores existentes, tendremos que instalar en nuestro equipo lo que en Ubuntu se llaman controladores privativos.

Estos controladores hardware son **controladores suministrados por los fabricantes de hardware** para Ubuntu, de ahí su nombre de **privativos**. El **resto de controladores** son **genéricos suministrados por el propio sistema operativo**. Solamente podremos instalar nuevos controladores hardware si el fabricante, por ejemplo Nvidia o ATI, los suministran. La mayoría de los fabricantes no suministran controladores privativos.