**Fdisk**

Es uno de los comandos más importantes que deberíamos conocer, puesto que en caso de no contar con interfaz gráfica de por medio (cosa muy habitual en las distribuciones destinadas a servidores por el mejor aprovechamiento de los recursos), nos ayudará enormemente en la **gestión y administración de nuestro espacio en disco** .

Con esta herramienta podremos **crear, eliminar, redimensionar, cambiar o copiar y mover particiones** usando el sencillo menú que ofrece. El límite que existe en esta herramienta está en **4 particiones primarias como máximo por disco** , y un número de particiones extendidas o lógicas que será variable en función del tamaño de nuestro disco duro.

A continuación veremos **algunos de los comandos más usados** para gestionar la tabla de particiones de un sistema **Linux** . Recordad que deberemos estar con el **usuario root o con algún usuario con permisos similares** , para no encontrarnos continuamente con errores de permisos o comandos no encontrados.

**Lista de Comandos**

Ayuda:

DOS (MBR)

a conmuta el indicador de iniciable

b modifica la etiqueta de disco BSD anidada

c conmuta el indicador de compatibilidad con DOS

General

d borra una partición

F lista el espacio libre no particionado

l lista los tipos de particiones conocidos

n añade una nueva partición

p muestra la tabla de particiones

t cambia el tipo de una partición

v verifica la tabla de particiones

i imprime información sobre una partición

Miscelánea

m muestra este menú

u cambia las unidades de visualización/entrada

x funciones adicionales (sólo para usuarios avanzados)

Script

I carga la estructura del disco de un fichero de script sfdisk

O vuelca la estructura del disco a un fichero de script sfdisk

Guardar y Salir

w escribe la tabla en el disco y sale

q sale sin guardar los cambios

Crea una nueva etiqueta

g crea una nueva tabla de particiones GPT vacía

G crea una nueva tabla de particiones SGI (IRIX) vacía

o crea una nueva tabla de particiones DOS vacía

s crea una nueva tabla de particiones Sun vacía

**Parted**

Parted es un programa para manipular particiones de disco. Admite múltiples formatos de tabla de particiones, incluidos MS-DOS y GPT. Es útil para crear espacio.

para nuevos sistemas operativos, reorganizar el uso del disco y copiar datos a nuevos discos duros.

Los documentos de esta página del manual se separaron brevemente. La documentación completa se distribuye con el paquete en formato de información GNU.

**Lista de Comandos:**

¡Bienvenido/a a GNU Parted! Teclee «help» para ver la lista de órdenes.

(parted) help

align-check TIPO N comprueba particiones N para alineaciones TIPO(min|opc)

help [ORDEN] muestra ayuda general, o ayuda sobre ORDEN

mklabel,mktable TIPO-ETIQUETA crea una nueva etiqueta de disco (tabla de particiones)

mkpart TIPO-PART [TIPO-SF] INICIO FIN crea una partición

name NUMERO NOMBRE nombra la partición NÚMERO como NOMBRE

print [devices|free|list,all|NÚMERO] muestra la tabla de particiones, los dispositivos disponibles, el espacio libre y las particiones encontradas, o una

partición en particular

quit sale del programa

rescue INICIO FIN recupera una partición perdida entre INICIO y FIN

resizepart NUMBER END resize partition NUMBER

rm NÚMERO elimina la partición NÚMERO

select DISPOSITIVO elige el dispositivo que se editará

disk\_set FLAG STATE change the FLAG on selected device

disk\_toggle [FLAG] toggle the state of FLAG on selected device

set NÚMERO BANDERA ESTADO cambia la BANDERA en la partición NÚMERO

toggle [NÚMERO [BANDERA]] cambia el estado de BANDERA en la partición NÚMERO

unit UNIDAD establece la unidad UNIDAD como predeterminada

version muestra el número de versión y la información de copyright de GNU Parted

(parted)

**Gdisk**

es una herramienta de línea de comandos para crear y manipular tablas de particiones. Convierte automáticamente tablas de particiones MBR (*Master Boot Record*) o BSD *disklabel* (sólo si está almacenada sin anidar dentro de una partición MBR) al nuevo formato GPT (*GUID Partition Table*). De lo contrario carga la tabla GPT existente en el dispositivo.

GPT fdisk (también conocido como gdisk) es un programa basado en menú de texto para la creación

y manipulación de tablas de partición. Convertirá automáticamente un

tabla de partición de registro de arranque maestro (MBR) de estilo antiguo o etiqueta de disco BSD

almacenado sin una partición de soporte MBR al más nuevo Globally Unique

Formato de tabla de partición de identificador (GUID) (GPT), o cargará un GUID

Mesa de partición. Cuando se usa con la opción de línea de comandos -l, el programa

muestra la tabla de partición actual y luego sale.

Command (? for help): help

b back up GPT data to a file

c change a partition's name

d delete a partition

i show detailed information on a partition

l list known partition types

n add a new partition

o create a new empty GUID partition table (GPT)

p print the partition table

q quit without saving changes

r recovery and transformation options (experts only)

s sort partitions

t change a partition's type code

v verify disk

w write table to disk and exit

x extra functionality (experts only)

? print this menu

**MKFS**

Comando Linux MKFS, se  utiliza para dar formato a un dispositivo de almacenamiento de bloque con un determinado sistema de archivos.  
Un Sistema de archivos es la estructura básica de toda los datos que se guarda, edita, borra o copia etc…en el ordenador, siendo toda esta información accesada a través de gestores de archivos en sus respectivos SO.  
ext – Extended file system (sistema de archivos extendido)  
ext2 – Second extended file system (segundo sistema de archivo extendido)  
ext3 – Third extended file system (tercer sistema de archivos extendido), una forma de ext2 con registro en diario.

ext4 – Fourth extended file system (cuarto sistema de archivos extendido), una mejora de ext3 y también un sistema de archivos con registro en diario con soporte para extensiones.

## Comando Linux mkfs / Sintaxis:

mkfs [options] [-t type fs-options] device [size]  
mkfs [opciones] [-t tipo fs-opciones] dispositivo [size]

### Ejemplos prácticos mkfs:

#Crear filesystem  
mkfs -t ext4 /dev/sdd1

mke2fs -t ext4 /dev/sdb

**dd**

dd es un comando de la familia de los sistemas operativos [Unix](https://es.wikipedia.org/wiki/Unix) que permite copiar y convertir datos de archivos a bajo nivel.

Es generalmente usado para realizar operaciones avanzadas sobre dispositivos o archivos, como pueden ser:

* Transferencias de datos específicos.
* [Copias de seguridad](https://es.wikipedia.org/wiki/Copias_de_seguridad) de la información «en crudo» (*raw data*).
* Conversión de algunas codificaciones soportadas y/o caracteres predefinidos.[1](https://es.wikipedia.org/wiki/Dd_(Unix)" \l "cite_note-1)​

La invocación del comando dd tiene el siguiente formato:[3](https://es.wikipedia.org/wiki/Dd_(Unix)" \l "cite_note-3)​

dd [PARÁMETRO] ...

Los principales parámetros son:

if=*origen*

Lee desde el archivo indicado como origen. Por defecto lee de la [entrada estándar](https://es.wikipedia.org/wiki/Entrada_estándar).

of=*destino*

Escribe al archivo indicado como *destino*. Por defecto escribe en la [salida estándar](https://es.wikipedia.org/wiki/Salida_estándar).

ibs=*N*

Lee *N* bytes del archivo *origen*.

obs=*N*

Escribe *N* bytes en el archivo *destino*.

bs=*N*

Lee y escribe *N* bytes. Alternativa a usar **ibs** y **obs** con un mismo valor.

cbs=*N*

Establece en *N* bytes al tamaño del bloque de conversión para **block** y **unblock**.

skip=*N*

Se salta *N* bloques del archivo *origen* antes realizar la operación de copiado. El tamaño del bloque es indicado por **ibs**.

seek=*N*

Se salta *N* bloques del archivo *destino* antes realizar la operación de copiado. El tamaño del bloque es indicado por **obs**.

count=*N*

Copia *N* bloques del archivo *origen*, en vez de procesar hasta el final. El tamaño del bloque es indicado por **ibs**.

conv=*modo[, modo,...]*

Realiza las operaciones de conversión, según se indique. Se puede indicar más de una conversión, separándolas por comas.

Conversiones soportadas:

ascii

Convierte los caracteres [EBCDIC](https://es.wikipedia.org/wiki/EBCDIC) a [ASCII](https://es.wikipedia.org/wiki/ASCII).

ebcdic

Convierte los caracteres [ASCII](https://es.wikipedia.org/wiki/ASCII) a [EBCDIC](https://es.wikipedia.org/wiki/EBCDIC).

ibm

Convierte los caracteres [ASCII](https://es.wikipedia.org/wiki/ASCII) al [EBCDIC](https://es.wikipedia.org/wiki/EBCDIC) de [IBM](https://es.wikipedia.org/wiki/IBM).

lcase

Intercambia las mayúsculas por minúsculas.

ucase

Intercambia las minúsculas por mayúsculas.

swab

Intercambia cada par de bytes de la entrada. Para el caso especial del último byte, este se copia directamente.

noerror

No se detiene el proceso ante errores de lectura en el *origen*.

notrunc

No se trunca el archivo utilizado como *destino*.

sync

Rellena cada bloque leído con ceros, hasta el tamaño determinado por **ibs**.

block

Rellenar con espacios en blanco la línea leída, hasta el tamaño indicado por **cbs**. Reemplaza el carácter de nueva línea por espacios, convirtiendo la línea en un bloque (o registro).

unblock

Reemplazar los últimos espacios en blanco por un carácter de salto de línea del registro leído, que posea el tamaño indicado por **cbs**. Realiza la operación inversa a **block**.

**DF**

El comando df (disk free), es una herramienta CLI del tipo Unix que nos permite conocer la cantidad de espacio libre y espacio utilizado por nuestro sistema de archivos en nuestras unidades de almacenamiento. Importante tener en claro esto ya que es solo de nuestras unidades de almacenamiento, y no de nuestros archivos y directorios.

**DU**

Puede ser de mucha utilidad en Linux para saber el tamaño que tiene un fichero, una carpeta o nuestro servidor en el disco duro. El comando se escribe desde el terminal y a continuación os vamos a enseñar cómo utilizarlo.

Existen varias maneras de averiguar el peso de archivos y carpetas con este comando. A continuación vamos a ver algunos comandos sencillos para ejecutar aplicaciones que ya vienen instaladas en todos los sistemas.

Veamos este ejemplo para saber el tamaño de un .iso o una carpeta en específico, el comando du se utiliza de la siguiente manera.

$ du -bsh /fichero\_o\_carpeta

Como vemos, se puede personalizar el uso de du con añadiendo algunas de estas tres opciones:

-b [–bytes]: Mostrar en bytes.

-s [–sumarize]: Mostrar solamente el tamaño total de cada argumento.

-h [–human-readable]: Imprime los tamaños de forma leíble (e.g., 1K, 234M, 2G)

**Mount**

es un comando de sistemas basados en [**Unix**](https://es.wikipedia.org/wiki/Unix) que se utiliza para [**montar**](https://es.wikipedia.org/wiki/Montar_(informática)) [**dispositivos**](https://es.wikipedia.org/wiki/Dispositivo_de_almacenamiento_de_datos) y [**particiones**](https://es.wikipedia.org/wiki/Partición_de_disco) para su uso por el [**sistema operativo**](https://es.wikipedia.org/wiki/Sistema_operativo). Montar es hacer que el sistema operativo proyecte el contenido de ese dispositivo o partición en un enlace lógico (un directorio). Cuando se desocupa se rompe el enlace y se sigue trabajando con los mismos archivos básicos.

## **Uso**

mount [-t <tipo>] [-o <opciones>] <dispositivo> <punto\_de\_lectura>

mount -t iso9660 /dev/hdb0 /mnt/cdrom

mount -t iso9660 -o loop /home/usuario/imagen.iso /mnt/imagenes

**El tipo del** [**sistema de archivos**](https://es.wikipedia.org/wiki/Sistema_de_archivos) **puede ser:**

* auto (intenta descubrir automáticamente el sistema de archivos).
* [iso9660](https://es.wikipedia.org/wiki/Iso9660) (sistema de archivos de los CD y DVD).
* [ext2](https://es.wikipedia.org/wiki/Ext2) (sistema de archivos muy extendido en máquinas Linux).
* [ext3](https://es.wikipedia.org/wiki/Ext3) (igual que ext2 pero además añade journaling).
* [ext4](https://es.wikipedia.org/wiki/Ext4) (introdujo significativas mejoras respecto a ext3)
* [reiserfs](https://es.wikipedia.org/w/index.php?title=Reiserfs&action=edit&redlink=1) (otro sistema muy utilizado en máquinas Linux).
* msdos (para dispositivos que usen [FAT12](https://es.wikipedia.org/wiki/FAT12) o [FAT16](https://es.wikipedia.org/wiki/FAT16)).
* vfat (para dispositivos que usen [FAT32](https://es.wikipedia.org/wiki/FAT32)).
* ntfs (sistema de archivos [NTFS](https://es.wikipedia.org/wiki/NTFS) de [Windows](https://es.wikipedia.org/wiki/Windows) NT, XP, etc.).
* smbfs (sistema de archivos de [Samba](https://es.wikipedia.org/wiki/Samba_(programa))).
* nfs (sistema de archivos de red [NFS](https://es.wikipedia.org/wiki/NFS)).
* hfs y hfsplus (acceden a sistemas de archivos de [Apple](https://es.wikipedia.org/wiki/Apple_Inc.) [Macintosh](https://es.wikipedia.org/wiki/Macintosh)).

#### **Opciones comunes para todos los sistemas de archivos**

Como los sistemas de archivos de /etc/fstab serán ocasionalmente montados utilizando mount no es de extrañar que el campo de opciones contenga simplemente una lista de opciones separadas por comas que serán pasadas directamente al intentar montar un sistema de archivos.

Las opciones comunes a todos los sistemas de archivos son:

auto / noauto

Con la opción *auto*, el dispositivo será montado automáticamente durante el inicio o en caso de que el comando mount -a sea ejecutado. auto es el valor por defecto. Si no se desea que el dispositivo se monte automáticamente, se deberá substituir por *noauto*.

dev / nodev

Interpretar / no interpretar dispositivos especiales de bloques en el sistema de archivos. Los dispositivos especiales de bloques son similares a discos (donde se puede acceder a los datos dado un número de bloque, y p.ej. tiene sentido tener un caché de bloques). Todos los otros dispositivos son ficheros especiales de caracteres. Hace tiempo la distinción era diferente: la E/S a un fichero especial de caracteres sería sin búfer, y a un fichero especial de bloques, con búfer.

exec / noexec

*exec* permite ejecutar [binarios](https://es.wikipedia.org/wiki/Binario) que están en la partición, mientras que *noexec* lo impide. noexec puede resultar útil en una partición que contenga binarios que se deseen ejecutar en el sistema, o que no deban ser ejecutados. Este podría ser el caso de la partición de [Windows](https://es.wikipedia.org/wiki/Windows).

ro

Montar para sólo lectura.

rw

Montar el sistema de archivos para lectura/escritura. Utilizando esta opción, muchos nuevos usuarios confusos, por no poder escribir en sus discos o particiones de Windows, pueden solucionar el problema.

sync / async

Esta opción indica la manera en que se debe realizar la entrada y salida del sistema de archivos. *sync* especifica que se realice de manera síncrona. Para verlo en un ejemplo de un fstab, nótese que esta opción aparece en la unidad de [discos flexibles](https://es.wikipedia.org/wiki/Disquete). En particular sirve, si se está escribiendo en una unidad con la opción activada, para que los cambios sean realizados físicamente en el dispositivo a la vez que se invoca el comando correspondiente.

suid / nosuid

Permite / bloquea la operación sobre los bits [suid](https://es.wikipedia.org/wiki/Setuid) y [sgid](https://es.wikipedia.org/wiki/Setgid).

user / nouser

Permite a cualquier [usuario](https://es.wikipedia.org/wiki/Usuario) montar el sistema de archivos. Implica directamente las opciones *noexec*, *nosuid* y *nodev* a menos que se especifiquen otras. Si se utiliza la opción nouser, solo el usuario [root](https://es.wikipedia.org/wiki/Root) podrá montar el sistema de archivos.

defaults

Utiliza las opciones por defecto: rw,suid,dev,exec,auto,nouser,async.

nofail

No devuelve error si no se puede montar el dispositivo.

# **E2fsprogs**

El paquete e2fsprogs (también llamado programas e2fs) es un conjunto de utilidades para mantenimiento de los [**sistemas de ficheros**](https://es.wikipedia.org/wiki/Sistema_de_archivos) [**ext2**](https://es.wikipedia.org/wiki/Ext2), [**ext3**](https://es.wikipedia.org/wiki/Ext3) y [**ext4**](https://es.wikipedia.org/wiki/Ext4). Debido a que estos son generalmente los sistemas de archivos por defecto en las [**distribuciones Linux**](https://es.wikipedia.org/wiki/Distribución_linux), comúnmente se considera al paquete e2fsprogs software esencial.

e2fsprogs incluye:

* e2fsck, un programa [fsck](https://es.wikipedia.org/wiki/Fsck) que busca y corrige inconsistencias;
* mke2fs, usado para crear sistemas de archivos ext2, ext3, y ext4;
* resize2fs, que puede expandir y contraer sistemas de archivos ext2, ext3, y ext4;
* tune2fs, usado para modificar los parámetros en el sistema de archivos;
* dumpe2fs, que muestra la información de bloques y superbloques;
* [debugfs](https://es.wikipedia.org/wiki/Debugfs), usado para visualizar o modificar estructuras internas del sistema de archivos manualmente.

Muchas de estas utilidades están basadas en la [biblioteca](https://es.wikipedia.org/wiki/Biblioteca_(informática)) libext2fs.

A pesar de lo que su nombre pueda sugerir, los programas e2fs funcionan no solo con ext2 sino también con ext3, que está basado en este último. Aunque la capacidad de [journaling](https://es.wikipedia.org/wiki/Journaling) de ext3 puede reducir la necesidad de uso de e2fsck, en ocasiones es necesario para asistir en la protección contra bugs del [núcleo](https://es.wikipedia.org/wiki/Núcleo_(informática)) o hardware defectuoso.

Como aplicaciones de [Espacio de usuario](https://es.wikipedia.org/wiki/Espacio_de_usuario) para los controladores ext2, ext3 y ext4 del [núcleo Linux](https://es.wikipedia.org/wiki/Linux_(núcleo)), los e2fsprogs son más comúnmente usados con el sistema operativo [Linux](https://es.wikipedia.org/wiki/Linux). Sin embargo, han sido portados a otros sistemas, como [FreeBSD](https://es.wikipedia.org/wiki/FreeBSD) y [Darwin](https://es.wikipedia.org/wiki/Darwin_BSD).

**Fsck**

**fsck** (***f****ile* ***s****ystem che****ck*** o bien ***f****ile* ***s****ystem* ***c****onsistency chec****k***) es una utilidad de los sistemas [Unix](https://es.wikipedia.org/wiki/Unix) y similares, como [Linux](https://es.wikipedia.org/wiki/Linux), [AIX](https://es.wikipedia.org/wiki/AIX) y [MAC OS X](https://es.wikipedia.org/wiki/MAC_OS_X) que se utiliza ante alguna inconsistencia del [sistema de archivos](https://es.wikipedia.org/wiki/Sistema_de_archivos) para corregir los posibles errores en el sistema.

fsck se ejecuta automáticamente al inicio del sistema ante alguna anomalía, pero también puede ser utilizada manualmente por el [administrador del sistema](https://es.wikipedia.org/wiki/Administrador_de_sistemas) para forzar una verificación.

Para verificar un sistema de archivos es muy aconsejable hacerlo mientras éste está [desmontado](https://es.wikipedia.org/wiki/Montar).

La sintaxis básica de esta utilidad es:

fsck [-opciones] /dev/hdXXX (o sdXXX)

donde debemos sustituir [-opciones] por el parámetro que queramos utilizar y hdXXX por el nombre de la partición que queramos verificar.

Los parámetros básicos son:

**-a** *confirmar automáticamente. No recomendado.*  
**-c** *comprobar bloques en el disco.*   
**-f** *forzar la verificación aunque todo parezca normal.*   
**-v** *(verbose) despliega más información.*   
**-r** *Modo interactivo. Espera nuestra respuesta.*   
**-y** *asume yes de respuesta.*

**/etc/fstab**

El archivo [/etc/fstab](https://en.wikipedia.org/wiki/es:Fstab) es usado para definir cómo las particiones, distintos dispositivos de bloques o sistemas de archivos remotos deben ser montados e integrados en el sistema.

Cada sistema de archivos se describe en una línea separada. Estas definiciones se convertirán con [systemd](https://wiki.archlinux.org/index.php/Systemd_(Español)) en unidades montadas de forma dinámica en el arranque, y cuando se recargue la configuración del administrador del sistema.

El archivo es leído por la orden mount, a la cual le basta con encontrar cualquiera de los directorios o dispositivos indicados en el archivo para completar el valor del siguiente parámetro. Al hacerlo, las opciones de montaje que se enumeran en fstab también se aplicarán.

***etc/*mtab**

# **Mtab**

[Ir a la navegación](https://es.wikipedia.org/wiki/Mtab" \l "mw-head) [Ir a la búsqueda](https://es.wikipedia.org/wiki/Mtab" \l "p-search)

El fichero **mtab** (contracción de mesa de ***m****ounted* [file systems](https://es.wikipedia.org/wiki/File_systems)***tab****le*) es un archivo de información del sistema que suele encontrarse en [sistemas operativos](https://es.wikipedia.org/wiki/Sistemas_operativos) de tipo [Unix.](https://es.wikipedia.org/wiki/Unix-like) Este archivo lista los sistemas de ficheros ya montados, así como sus opciones de inicialización. **mtab** tiene una estructura muy similar a la de [fstab](https://es.wikipedia.org/wiki/Fstab), la diferencia es que el último es un archivo de configuración que contiene los sistemas de ficheros que deben ser montado en el tiempo de ejecución, así como sus puntos de montaje, mientras que el primero, simplemente lista los sistemas de ficheros montados en este momento. Es muy común copiar lineas de **mtab** en [fstab](https://es.wikipedia.org/wiki/Fstab). para mantener la misma configuración al reiniciar el sistema.​

**/dev/: archivos de dispositivos**

**/dev/disk/by-id**

by-id crea un nombre único en función del número de serie del hardware, by-path crea un nombre único en función de la ruta física más corta (de acuerdo con sysfs). Ambos contienen cadenas para indicar el subsistema al que pertenecen (es decir, pci- para by-path, y ata- para by-id) de modo que están vinculados al hardware que controla el dispositivo. Esto implica diferentes niveles de persistencia: by-path cambiará cuando el dispositivo se conecte a un puerto diferente del controlador; by-id cambiará cuando se conecte el dispositivo en un puerto de un controlador de hardware sujeto a otro subsistema. [[1]](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/5/html/Online_Storage_Reconfiguration_Guide/persistent_naming.html) Por lo tanto, ninguno de los dos son adecuados para lograr nombres persistentes tolerantes a los cambios de hardware.

Sin embargo, ambos proporcionan información importante para encontrar un dispositivo en particular en una gran infraestructura de hardware. Por ejemplo, si no se asignan manualmente etiquetas persistentes (by-label o by-partlabel) y se mantiene un directorio con utilización de puerto de hardware, by-id y by-path pueden ser utilizados para encontrar un dispositivo en particular. [[2]](http://linuxshellaccount.blogspot.in/2008/09/how-to-easily-find-wwns-of-qlogic-hba.html) [[3]](http://www.linuxquestions.org/questions/linux-server-73/how-to-find-wwn-for-dev-sdc-917269/)

by-id también crea enlaces a [nombres a nivel mundial](https://en.wikipedia.org/wiki/es:World_Wide_Name) (siglas en inglés World Wide Name) de los dispositivos de almacenamiento que lo admitan. A diferencia de otros enlaces by-id, WWNs son completamente persistentes y no cambiarán al margen del subsistema utilizado.

$ ls -l /dev/disk/by-id/

total 0

lrwxrwxrwx 1 root root 10 May 27 23:31 ata-WDC\_WD2500BEVT-22ZCT0\_WD-WXE908VF0470 -> ../../sda

lrwxrwxrwx 1 root root 10 May 27 23:31 ata-WDC\_WD2500BEVT-22ZCT0\_WD-WXE908VF0470-part1 -> ../../sda1

lrwxrwxrwx 1 root root 10 May 27 23:31 ata-WDC\_WD2500BEVT-22ZCT0\_WD-WXE908VF0470-part2 -> ../../sda2

lrwxrwxrwx 1 root root 10 May 27 23:31 ata-WDC\_WD2500BEVT-22ZCT0\_WD-WXE908VF0470-part3 -> ../../sda3

lrwxrwxrwx 1 root root 10 May 27 23:31 ata-WDC\_WD2500BEVT-22ZCT0\_WD-WXE908VF0470-part4 -> ../../sda4

lrwxrwxrwx 1 root root 10 May 27 23:31 mmc-SD32G\_0x0040006d -> ../../mmcblk0

lrwxrwxrwx 1 root root 10 May 27 23:31 mmc-SD32G\_0x0040006d-part1 -> ../../mmcblk0p1

lrwxrwxrwx 1 root root 10 May 27 23:31 wwn-0x60015ee0000b237f -> ../../sda

lrwxrwxrwx 1 root root 10 May 27 23:31 wwn-0x60015ee0000b237f-part1 -> ../../sda1

lrwxrwxrwx 1 root root 10 May 27 23:31 wwn-0x60015ee0000b237f-part2 -> ../../sda2

lrwxrwxrwx 1 root root 10 May 27 23:31 wwn-0x60015ee0000b237f-part3 -> ../../sda3

lrwxrwxrwx 1 root root 10 May 27 23:31 wwn-0x60015ee0000b237f-part4 -> ../../sda4

**Se encuentran los codigos unicos asignado a cada unidades**

**/dev/disk/by-uuid**

[UUID](https://en.wikipedia.org/wiki/es:Universally_unique_identifier) es un mecanismo para dotar a cada [sistema de archivos](https://wiki.archlinux.org/index.php/File_systems_(Español)) de un identificador único. Estos identificadores son generados por las utilidades del sistema de archivos (por ejemplo, mkfs.\*) cuando la partición se formatea y se ha diseñado de manera que evite conflictos. Todos los sistemas de archivos de GNU/Linux (incluyendo swap y las cabeceras LUKS de dispositivos cifrados en bruto) soportan UUID. Los sistemas de archivos FAT, exFAT y NTFS no son compatibles con UUID, pero todavía se enumeran en /dev/disk/by-uuid con un UID más corto (identificador único):

Ficheron donde se encuentran los UUID

$ ls -l /dev/disk/by-uuid/

total 0

lrwxrwxrwx 1 root root 10 May 27 23:31 0a3407de-014b-458b-b5c1-848e92a327a3 -> ../../sda2

lrwxrwxrwx 1 root root 10 May 27 23:31 b411dc99-f0a0-4c87-9e05-184977be8539 -> ../../sda3

lrwxrwxrwx 1 root root 10 May 27 23:31 CBB6-24F2 -> ../../sda1

lrwxrwxrwx 1 root root 10 May 27 23:31 f9fe0b69-a280-415d-a03a-a32752370dee -> ../../sda4

lrwxrwxrwx 1 root root 10 May 27 23:31 F4CA-5D75 -> ../../mmcblk0p1

El UUID de un dispositivo se puede obtener con *lsblk*:

$ lsblk -dno UUID /dev/sda1

CBB6-24F2

O con *blkid*:

# blkid -s UUID -o value /dev/sda1

CBB6-24F2

**Sistema de archivos virtual**

Un **sistema de archivos virtual** (VFS) o **conmutador de sistema de archivos virtual** es una capa de abstracción encima de un [sistema de archivos](https://es.wikipedia.org/wiki/Sistema_de_archivos) más concreto. El propósito de un VFS es permitir que las aplicaciones cliente tengan acceso a diversos tipos de sistemas de archivos concretos de una manera uniforme. Puede ser utilizada para tender un puente sobre las diferencias en los sistemas de archivos de [Windows](https://es.wikipedia.org/wiki/Windows), de [Mac OS](https://es.wikipedia.org/wiki/Mac_OS) y [Unix](https://es.wikipedia.org/wiki/Unix), de modo que las aplicaciones pudieran tener acceso a archivos en los sistemas de archivos locales de esos tipos sin tener que saber a qué tipo de sistema de archivos están teniendo acceso.

Un VFS especifica una [interfaz](https://es.wikipedia.org/wiki/Interfaz) (o un contrato) entre el [kernel](https://es.wikipedia.org/wiki/Núcleo_(informática)) y un sistema de archivos en concreto. Por lo tanto, es fácil agregar nuevos sistemas de archivos al kernel simplemente satisfaciendo el contrato. Los términos del contrato pueden volverse incompatibles de una versión a otra, lo que requeriría que sistemas de archivos concretos fuesen recompilados, y posiblemente modificados antes de la recompilación, para permitirles trabajar con un nuevo lanzamiento del sistema operativo; o el proveedor del sistema operativo pueda realizar solamente cambios retrocompatibles al contrato, de modo que un sistema de archivos concreto construido para un lanzamiento dado del sistema operativo trabaje con las versiones futuras del mismo sistema operativo.

**Sistemas de archivos ext2, ext3, ext4, xfs, vfat, ntfs**

* **ext2:** Hasta hace poco era el sistema estándar de Linux. Tiene una fragmentación muy baja, aunque es algo lento manejando archivos de gran tamaño. Fue la continuación del sistema de ficheros **ext**, implementado en 1992 e integrado en Linux 0.96. Las principales ventajas que tenía sobre ext eran las siguientes:
  + Compatible con sistemas de ficheros grandes, admitiendo particiones de disco de hasta 4TB y ficheros de hasta 2GB de tamaño.
  + Proporciona nombres de ficheros largos, de hasta 255 caracteres.
  + Tiene una gran estabilidad.
  + Actualización.
* **ext3:** Es la versión mejorada de ext2, con previsión de pérdida de datos por fallos del disco o apagones. En contraprestació, es totalmente imposible recuperar datos borrados. Es compatible con el sistema de ficheros ext2. Actualmente es el más difundido dentro de la comunidad GNU/Linux y es considerado el estándar. Sus ventajas frente a ext2 son:
  + Actualización. Debido a que los dos sistemas comparten el mismo formato, es posible llevar a cabo una actualización a ext3, incluso aunque el sistema ext2 esté montado.
  + Fiabilidad y mantenimiento.
* **ext4:** Es la última versión de la familia de sistemas de ficheros ext. Sus principales ventajas radican en su eficiencia (menor uso de CPU, mejoras en la velocidad de lectura y escritura) y en la ampliación de los límites de tamaño de los ficheros, ahora de hasta 16TB, y del sistema de ficheros, que puede llegar a los 1024PB (PetaBytes).
* **ReiserFS:** Es el sistema de ficheros de última generación para Linux. Organiza los ficheros de tal modo que se agilizan mucho las operaciones con estos. El problema de ser tan actual es que muchas herramientas (por ejemplo, para recuperar datos) no lo soportan.
* **swap:** Es el sistema de ficheros para la partición de intercambio de Linux. Todos los sistemas Linux necesitan una partición de este tipo para cargar los programas y no saturar la memoria RAM cuando se excede su capacidad. En Windows, esto se hace con el archivo pagefile.sys en la misma partición de trabajo, con los problemas que esto conlleva.
* **XFS** es un [sistema de archivos](https://es.wikipedia.org/wiki/Sistema_de_archivos) de 64 bits con **registro de bitácora** o [***journaling***](https://es.wikipedia.org/wiki/Journaling) de alto rendimiento creado por [SGI](https://es.wikipedia.org/wiki/Silicon_Graphics) (antiguamente Silicon Graphics Inc.) para su implementación de [UNIX](https://es.wikipedia.org/wiki/UNIX) llamada [IRIX](https://es.wikipedia.org/wiki/IRIX). En mayo de 2000, SGI liberó XFS bajo una [licencia de código abierto](https://es.wikipedia.org/wiki/Licencia_de_código_abierto).

XFS se incorporó a [Linux](https://es.wikipedia.org/wiki/Linux) a partir de la versión 2.4.25, cuando [Marcelo Tosatti](https://es.wikipedia.org/wiki/Marcelo_Tosatti) (responsable de la rama 2.4) lo consideró lo suficientemente estable para incorporarlo en la rama principal de desarrollo del [kernel](https://es.wikipedia.org/wiki/Núcleo_(informática)). Los programas de instalación de las [distribuciones](https://es.wikipedia.org/wiki/Distribuciones_de_Linux) de [SuSE](https://es.wikipedia.org/wiki/SuSE), [Gentoo](https://es.wikipedia.org/wiki/Gentoo), [Mandriva](https://es.wikipedia.org/wiki/Mandriva), [Slackware](https://es.wikipedia.org/wiki/Slackware), [Fedora Core](https://es.wikipedia.org/wiki/Fedora_Core), [Ubuntu](https://es.wikipedia.org/wiki/Ubuntu_(distribución_Linux)) y [Debian](https://es.wikipedia.org/wiki/Debian) ofrecen XFS como un [sistema de archivos](https://es.wikipedia.org/wiki/Sistema_de_archivos) más. En [FreeBSD](https://es.wikipedia.org/wiki/FreeBSD) el soporte para solo-lectura de XFS se añadió a partir de diciembre de 2005 y en junio de 2006 un soporte experimental de escritura fue incorporado a FreeBSD-7.0-CURRENT y luego eliminado en FreeBSD 10.0

* XFS soporta un sistema de archivos de hasta 8 [exabytes](https://es.wikipedia.org/wiki/Exabyte), aunque esto puede variar dependiendo de los límites impuestos por el [sistema operativo](https://es.wikipedia.org/wiki/Sistema_operativo). En sistemas [GNU/Linux](https://es.wikipedia.org/wiki/GNU/Linux) de 32 bits, el límite es 16 [terabytes](https://es.wikipedia.org/wiki/Terabyte)

Además de estos sistemas de ficheros, Linux también ofrece soporte para sistemas de ficheros de Windows, como FAT, FAT32 y NTFS. Tanto para FAT como para FAT32, Linux tiene soporte completo y estable de escritura y lectura, mientras que para NTFS, y con las últimas versiones del kernel, solo se puede acceder de manera estable en modo lectura. En modo escritura todavía está en fase experimental y no es estable.

[« Anterior](http://mural.uv.es/oshuso/821_caractersticas_del_sistema_de_ficheros_de_linux.html) | [Sig](http://mural.uv.es/oshuso/823_directorios_ms_importantes_en_linux.html)