

UF3 - Sistemas operativos. Gestión de archivos y almacenamiento

SISTEMAS INFORMÁTICOS

Índice

Introducción

Sistemas de archivos

- FAT
- exFAT
- NTFS
- APFS
- ext4
 - i-nodo

Estructura de directorios

- Directorios en Linux
- Directorios en Windows

Gestión de archivos por CLI en Linux

Entrada/salida por CLI en Linux

Comandos CLI en Linux

Gestión de almacenamiento en Linux

- Desfragmenación
- Chequeo
- RAID

Introducción

La **gestión de archivos** es uno de los pilares fundamentales de cualquier sistema operativo, junto a la gestión de procesos y la de usuarios.

La gestión de la información contenida en los medios de almacenamiento es posible gracias a los **sistemas de archivos**, que proveen la forma de almacenar los datos, así como mecanismos que permiten realizar operaciones en ellos.

Existen multitud de sistemas de archivos con diferentes características que repercuten en la seguridad, rendimiento y gestión de los datos.

En esta unidad estudiaremos diferentes sistemas de ficheros, centrándonos en los más usados: ext4 y NTFS.

Introducción

Los sistemas operativos proveen herramientas para la gestión del almacenamiento, ya sea por línea de comandos o por interfaz gráfica, para realizar particiones, formatear, chequear, desfragmentar, montar y desmontar sistemas de archivos, buscar información e incluso crear diferentes esquemas RAID.

En esta unidad y las siguientes trabajaremos con herramientas GUI de Windows y, de forma más avanzada, comandos CLI de Linux.

Sistemas de archivos

Los sistemas de archivos emplean los **archivos (ficheros)** como herramientas de abstracción lógica de la información que permiten que el usuario no tenga que conocer la estructura interna de los medios de almacenamiento.

Otro elemento empleado por los sistemas de archivos son las **carpetas (directorios)**, ficheros que actúan de contenedores lógicos de ficheros y directorios, por lo que contienen los atributos y localización física de estos.

Sistemas de archivos

Los sistemas de ficheros tienen como objetivo:

- Acceder a la información de los ficheros
- Crear, modificar y eliminar ficheros
- Acceder a los ficheros mediante protocolos de red o mediante otros ficheros
- Facilitar el acceso multiusuario
- Facilitar el acceso a múltiples medios de almacenamiento
- Realizar copias de seguridad
- Utilizar herramientas de recuperación de la información
- Aportar eficiencia y seguridad de acceso a la información
- Maximizar el rendimiento de las operaciones sobre archivos
- Permitir la monitorización y contabilidad sobre ficheros
- Administrar el espacio de almacenamiento y gestionar la asignación de espacio libre

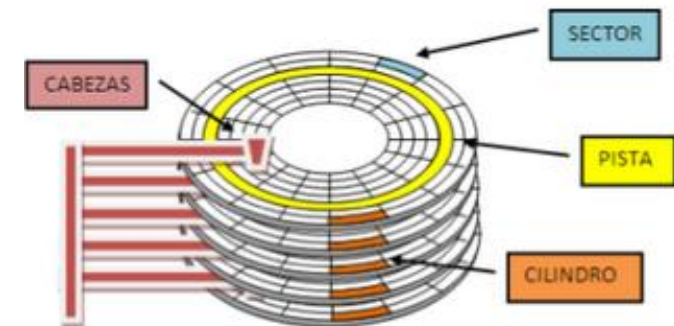
Sistemas de archivos

Para administrar el espacio libre y el espacio ocupado, se definen los espacios de asignación o **bloques**, que son las unidades mínimas de almacenamiento gestionadas por los sistemas de archivos.

Se puede definir el **tamaño de los bloques** durante la creación del sistema de archivos. Así, cada archivo ocupará un número entero de bloques según el contenido del archivo.

- En un sistema de archivos con tamaño de bloque de 4KB, un archivo que contenga 10 caracteres (bytes) ocupará 1 bloque entero en el disco (4KB), y un archivo de 15000 bytes ocupará 4 bloques (16KB).
- Lo ideal es elegir un tamaño de bloque adecuado según el tamaño medio de los ficheros, el tamaño total del volumen y el uso que se les va a dar.

A nivel físico, tanto las unidades flash como los discos duros mecánicos se dividen en bloques o sectores, pero no se trata de los mismos que los bloques del sistema de archivos.



FAT

FAT (*File Allocation Table*) es un sistema de ficheros creado para el sistema operativo MS-DOS.

FAT es compatible con la mayoría de los sistemas operativos, por lo que se suele emplear en dispositivos extraíbles como pendrives, tarjetas SD, etc.

Su administración es muy sencilla, pero tiene importantes limitaciones:

- Imposibilidad de gestionar particiones superiores a 8TB (32GB en Windows).
- El tamaño máximo de un archivo es de 4GB.
- Bajo rendimiento.
- Inseguro: no permite cifrado, sus atributos y permisos son limitados y no permite *journaling*.

La versión de FAT usada actualmente es FAT32.

exFAT

exFAT es una evolución del sistema de archivos FAT32 que elimina sus principales limitaciones, aunque mantiene la ligereza frente a sistemas de archivos más avanzados como NTFS y APFS.

Sigue resultando inseguro, pero es ideal para medios flash extraíbles con gran capacidad.

Es compatible con la mayoría de los sistemas operativos.

NTFS

Es el sistema de archivos estándar de Windows desde Windows NT. Sus mejoras con respecto a FAT32 son considerables:

- Emplea *journaling*, favoreciendo una pronta recuperación ante errores inesperados.
- Permite cifrado y compresión.
- Reduce significativamente la fragmentación y aumenta la velocidad de búsqueda de archivos.
- Puede gestionar volúmenes de hasta 16 EB y archivos de hasta 16 TB.
- Emplea Unicode para el nombre de archivos, con hasta 255 caracteres.

APFS

Sistema de archivos empleado por Apple para sus medios de almacenamiento, que supone una versión mejorada de su predecesor HFS+.

Sus características son similares a NTFS y ext4, y puede administrar archivos y volúmenes de hasta 8EB. Permite cifrado y está optimizado para almacenamiento flash.

ext4

ext4 (*extended file system 4*) es el sistema de ficheros predeterminado para sistemas operativos basados en Linux. Incluye journaling, maneja archivos de hasta 16TB y volúmenes de hasta 1EB.

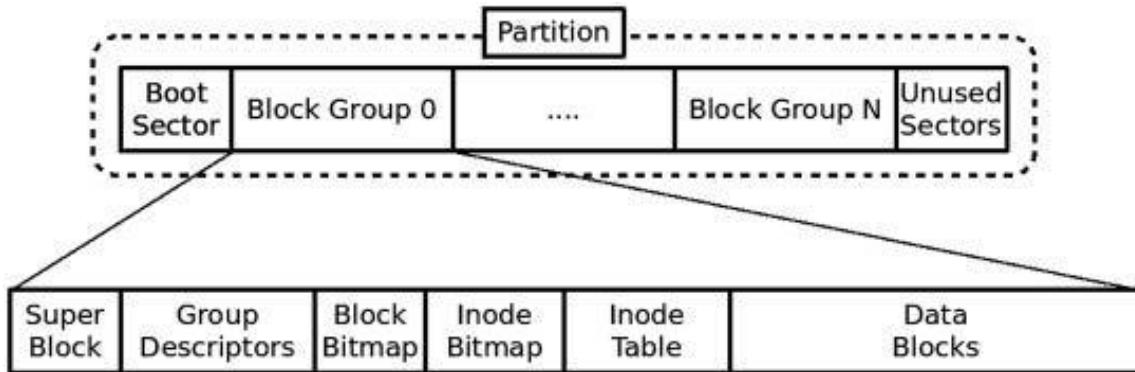
A diferencia de NTFS, ext4 no emplea extensiones como parte del nombre de los archivos para asociarlas con programas para ejecutar el archivo. Un archivo en Linux puede terminar con una extensión (.c, .txt, .html, ...), pero es frecuente ver archivos sin extensión.

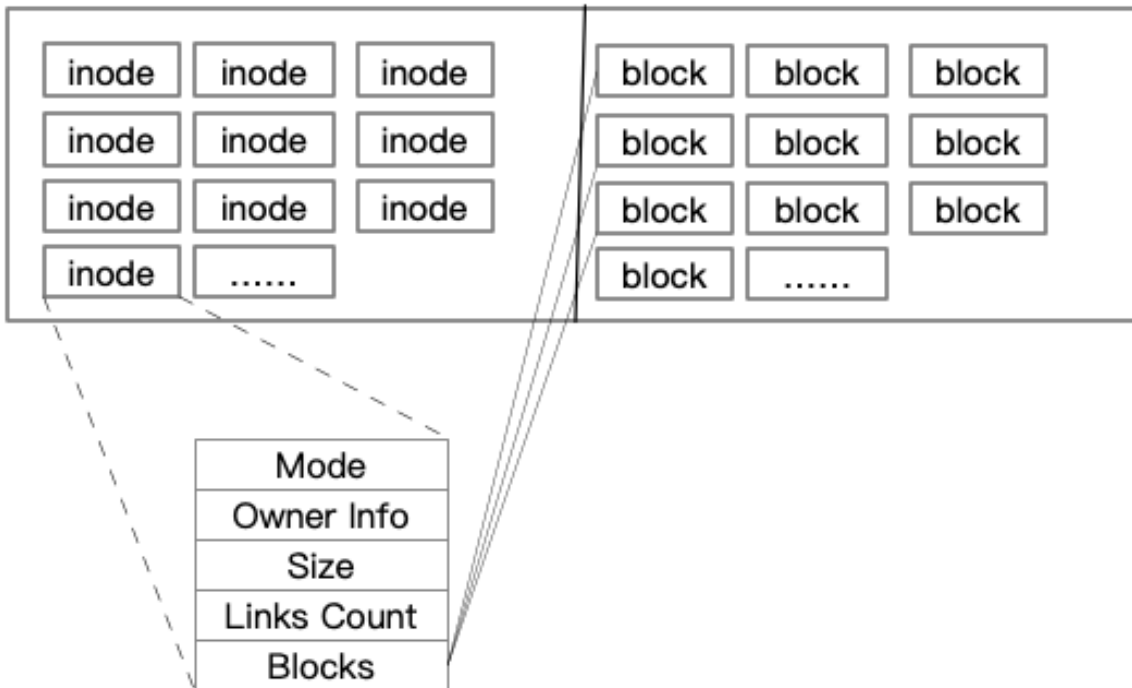
Windows no reconoce los sistemas de archivos ext4 ni APFS, pero la mayoría de distribuciones Linux sí reconocen NTFS.

ext4

Una partición con sistema de archivos ext4 se divide en **grupos de bloques**, los cuales se dividen en:

- **Superbloque:** contiene la información más relevante del grupo de bloques.
- **Descriptor de grupo:** almacena la información más importante del resto de bloques.
- **Mapa de bits de bloques:** donde se representa cada bloque y su estado (libre/ocupado).
- **Mapa de bits de i-nodos:** donde se representa cada i-nodo junto a su estado (libre/ocupado).
- **Tabla de i-nodos:** contiene una entrada por cada i-nodo, con los metadatos de cada archivo.
- **Bloques de datos:** bloques que alojan contenidos de archivos. Cada bloque es libre o está asociado a un archivo.





i-nodo

La estructura fundamental del sistema de archivos ext4 es el **i-nodo** o nodo índice.

Cada archivo se asocia a un i-nodo y este almacena los metadatos asociados del archivo.

i-nodo

Un i-nodo está compuesto por:

- **Número de i-nodo**: es un identificador único del i-nodo en su mismo sistema de archivos.
- **Tipo de archivo**: fichero regular, directorio, enlace simbólico, dispositivo por bloques o caracteres, etc.
- **Permisos** de escritura, lectura y ejecución para el usuario propietario, grupo propietario y otros.
- **Tamaño del fichero** en bytes.
- **Número de enlaces duros** que apuntan a este i-nodo.
- Identificadores de usuario (**UID**) y grupo (**GID**) propietarios del fichero.
- **Fechas** de último acceso (lectura), modificación (de contenido), cambio (de metadatos) y creación.
- Número y localización de los **bloques de datos**.

Estructura de directorios

Cuando un sistema operativo es instalado en un medio de almacenamiento, crean un conjunto de directorios donde se despliegan los archivos y subdirectorios del sistema operativo en forma de árbol invertido del sistema de archivos (**árbol de directorios**), es decir, un directorio del que cuelgan archivos y directorios de los que cuelgan archivos y directorios y así sucesivamente.

Para hacer referencia a un archivo o directorio dentro del árbol, se emplea el término **ruta** (*path*):

- En **Linux**, las **rutas absolutas** empiezan con la raíz (/) y bajan por el árbol carpeta a carpeta separándolas mediante /. Por ejemplo, `/usr/bin/cat`.
- En **Windows**, las **rutas absolutas** empiezan por la unidad o volumen (C:\, D:\, etc.) y bajan por el árbol separando las carpetas con \. Por ejemplo, `C:\Windows\system32\drivers`.

Las **rutas relativas** no empiezan por la raíz ni por ninguna unidad de disco, sino que parten del **directorio de trabajo actual**.

Directorios

Tanto en Windows como en sistemas tipo Unix, todos los directorios vienen con dos directorios especiales:

- `.`: se refiere al propio directorio en el que está.
- `..`: se refiere al directorio padre.

En Linux, `pwd` (o la variable `PWD`) muestra el directorio de trabajo actual.

El comando `cd` permite cambiar de directorio de trabajo.

Directorios de Linux

Los directorios más importantes de la mayoría de las distribuciones Linux son:

- `/bin`: contiene archivos ejecutables a nivel de usuario.
- `/sbin`: contiene archivos ejecutables a nivel de administrador.
- `/boot`: contiene ejecutables y configuración para el arranque del sistema.
- `/dev`: contiene ficheros que representan discos, particiones y componentes del sistema.
- `/etc`: almacena los archivos de configuración globales del sistema.
- `/home`: aloja los directorios personales de los usuarios normales.
- `/lib`: contiene bibliotecas y módulos del *kernel* del sistema operativo.
- `/media`: se emplea para montar medios de almacenamiento.
- `/mnt`: se emplea para montar medios de almacenamiento temporales.

Directorios de Linux

- `/proc`: guarda información relativa a los procesos y el *kernel*.
- `/sys`: guarda información relativa al *kernel*, *drivers* y dispositivos.
- `/tmp`: directorio con permisos de escritura abiertos donde las aplicaciones y usuarios pueden crear ficheros temporales de corta duración. Su contenido se borra al inicio.
- `/usr`: almacena archivos de solo lectura de la mayoría de las aplicaciones y utilidades instaladas en el sistema.
- `/opt`: contiene el resto de las aplicaciones no almacenadas en `/usr`.
- `/srv`: directorio encargado de alojar ficheros y carpetas para servidores.
- `/var`: contiene información varia: *backups*, correo, *logs*, caché, sitios web, etc.

Directorios de Windows

El directorio raíz de una partición con Windows instalado dispone de:

- `\Archivos de programa` o `\Program files`: se encuentran los programas instalados (en los sistemas de 64 bits, incluye solo los programas de 64 bits).
- `\Archivos de programa (x86)` o `\Program files (x86)`: solo para los sistemas de 64 bits, incluye solo los programas de 32 bits.
- `\PerfLogs`: puede contener registros de rendimiento del sistema.
- `\ProgramData`: contiene datos de programas para todos los usuarios.
- `\Usuarios` o `\Users`: carpeta que contiene una subcarpeta por cada usuario del sistema, además de Acceso público y Default.
- `\Windows`: contiene el grueso de los archivos del sistema operativo. Tiene las subcarpetas:
 - `System32`: contiene bibliotecas DLL de 32 bits (en sistemas de 32 bits) o 64 bits (en sistemas de 64 bits).
 - `SysWOW64`: contiene bibliotecas DLL de 32 bits. Solo para sistemas de 64 bits.
 - `WinSxS`: almacén de componentes de Windows, con archivos utilizados para la instalación y actualizaciones del sistema.

Gestión de archivos por CLI en Linux

Los principales comandos de gestión de ficheros en Linux son:

- `ls`: para listar el contenido de un directorio.
- `cd/pwd`: para cambiar/ver el directorio actual.
- `rm`: para eliminar ficheros o directorios (`-r`).
- `mkdir`: para crear directorios.
- `rmdir`: para eliminar directorios vacíos.
- `cp`: para copiar archivos o directorios (`-r`).
- `mv`: para mover archivos o directorios.
- `stat`: para ver los metadatos de un fichero.
- `ln`: para crear un enlace duro o simbólico (`-s`) de un fichero.
- `less/more`: para acceder al contenido de un fichero.
- `nano`: para ver y editar un fichero.

Entrada y salida por CLI en Linux

Los siguientes comandos aceptan argumentos para abrir archivos, pero si no se les pasa ningún argumento, leen de la entrada estándar:

- `cat`: saca el contenido de un archivo por la salida estándar.
- `head/tail`: saca las primeras/últimas líneas de un archivo por la salida estándar.
- `wc`: cuenta el número de líneas, palabras y bytes del fichero.
- `sort`: saca el contenido de un fichero con las líneas ordenadas.
- `tac`: muestra el contenido de un archivo pero con el orden de las líneas invertido.
- `rev`: muestra el contenido de un archivo pero con las líneas invertidas (carácter a carácter).
- `cut`: muestra secciones concretas de cada línea de un archivo.
- `grep`: busca una **expresión regular** en un archivo y solo muestra las líneas que lo incluyen.

Comandos CLI en Linux

En esta unidad se explican algunos conceptos teóricos y comandos de la gestión de ficheros en Linux, pero para mayor completitud se recomienda revisar el PDF *Chuleta Shell* en las siguientes secciones:

- **Ficheros:** tipos de ficheros.
- **Directorios:** árbol de directorios, rutas, búsqueda de archivos con `find`.
- **Otros Ficheros:** todo lo referido a enlaces físicos y simbólicos.
- **Entrada/Salida:** tipos de entradas/salidas estándar y de error, redirecciones.
- **Expresiones regulares y edición de flujo:** expresiones regulares.
- **Administración de discos.**

El PDF desarrolla todos los comandos vistos aquí, por lo que para conocer su uso se recomienda echarle un vistazo, abrir el manual (`man <comando>`) o buscar en internet.

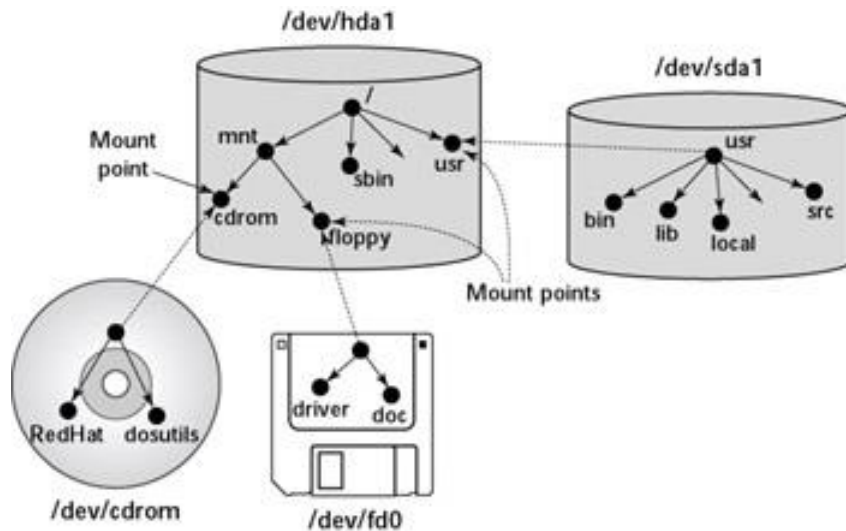
Gestión de almacenamiento en Linux

La estructura de directorios de Linux incluye los diferentes discos y particiones que el sistema operativo puede gestionar.

Cuando una partición contiene un sistema de archivos, Linux puede tratar con ella anexando su árbol de directorios al árbol de directorios raíz mediante un directorio común llamado punto de montaje. Así, este directorio permite acceder al árbol de directorios del subsistema de archivos.

Para montar y desmontar se usa:

- `mount`: este comando monta un sistema de archivos.
- `umount`: este comando desmonta un sistema de archivos.
- `/etc/fstab`: contiene montajes que suelen hacerse al inicio.



Gestión de almacenamiento en Linux

Los dispositivos de almacenamiento se administran a través del directorio `/dev`, el cual contiene archivos que se agrupan por tipos. Los más importantes son:

- `/dev/hd*`: interfaz para unidades de disco duro IDE (* es una letra).
- `/dev/sd*`: interfaz para discos SCSI, SATA y unidades de almacenamiento flash (SSDs, pen drives, etc.) (* es una letra).
- `/dev/tty*`: consolas o terminales físicos. Para cambiar entre consolas, se utiliza la combinación de teclas `Ctrl+Alt+F1..F6` (para volver al entorno gráfico, `Ctrl+Alt+F7`) (* es un número).
- `/dev/ttyS*`: puertos serie (* es un número).
- `/dev/sr*` y `/dev/scd*`: interfaz para unidades cd CD/DVD (* es un número).

Ejemplos: `/dev/hda`, `/dev/sdc`, `/dev/tty3`, `/dev/sr0`, etc.

Para identificar una partición, añadimos un número (de 1 en adelante) al dispositivo.

- Por ejemplo, si el disco `/dev/sdc` tiene 2 particiones, pueden llamarse `/dev/sdc1` y `/dev/sdc2`.

Gestión de almacenamiento en Linux

Para ver y administrar las tablas de particiones de los dispositivos de almacenamiento existen los comandos `parted` y `fdisk`:

```
# fdisk <dispositivo>           # parted <dispositivo>
# fdisk -l [<dispositivo>]      # parted -l [<dispositivo>]
```

El comando `lsblk` nos permite ver los dispositivos, sus particiones y su punto de montaje (si lo hay):

```
# lsblk [<dispositivo>]
```

Para formatear una partición:

```
# mkfs.<formato> <partición>
```

Por ejemplo:

```
#mkfs.ext4 /dev/sdb3
```

Desfragmentación

La **fragmentación** en un sistema de archivos se entiende como el esparcimiento de los datos relacionados entre sí en el medio de almacenamiento.

Aunque para los discos duros SSD no es un problema, los discos duros mecánicos pueden verse penalizados por la fragmentación, ya que el cabezal de lectura y escritura tendrá que oscilar continuamente para seguir los bloques esparcidos de un archivo.

Al proceso de unión de los bloques de datos de un mismo archivo se le conoce como **desfragmentación**, y es común en sistemas de ficheros NTFS y FAT para mejorar su rendimiento, aunque no en los sistemas de archivos de Linux, ya que emplean sistemas inteligentes de asignación de bloques.

Existen aplicaciones GUI de desfragmentación para Windows, como la que viene incluida en el propio SO. En Linux se puede usar el comando `e4defrag`.

Chequeo

Los discos duros suelen ser víctima de diversos problemas a lo largo de su vida útil, originados internamente o por agentes externos, como:

- Malware
- Fallos en componentes electrónicos
- Fluctuaciones de tensión eléctrica
- Daños físicos
- Errores del firmware o de actualización de drivers

La mayoría de discos duros emplean la tecnología SMART (*Self Monitoring Analysis and Reporting Technology*) con capacidad para detectar e informar de errores y fallos.

Cuando se emplea esta tecnología junto con software compatible (firmware BIOS/UEFI o software de terceros), obtenemos información valiosa sobre el estado actual del disco.

Chequeo

En Linux, existe el comando genérico `fsck` para hacer chequeos de cualquier sistema de archivos y el comando `e2fsck` para hacer chequeos de sistemas de archivos de la familia ext.

Es muy recomendable (e incluso puede ser obligatorio) desmontar un sistema de archivos antes de su chequeo.

RAID

Los discos duros están sujetos a fallos. Los discos duros mecánicos, al tener partes móviles, son especialmente susceptibles a perder los datos debido a fallos mecánicos, pero los SSD también pueden sufrir fallos que hagan perder la información.

Para asegurar la información que almacenan los discos duros, existe el concepto de RAID, que es una de las formas más eficaces de evitar la pérdida de datos por fallos de los dispositivos de almacenamiento.

- RAID no sirve para prevenir pérdidas por fallos del usuario o intencionadas. Para ello se usan las copias de seguridad.

RAID (*Redundant Array of Independent Disks*) consiste en establecer una configuración de un grupo de discos de almacenamiento para aumentar la integridad, la capacidad de almacenamiento, la velocidad de transferencia y/o disminuir el riesgo a fallos.

- Esta configuración se puede hacer mediante **software** o mediante **hardware** específico.

Tipos de RAID

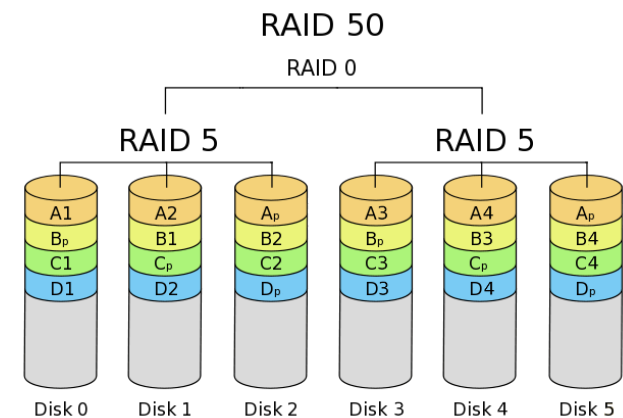
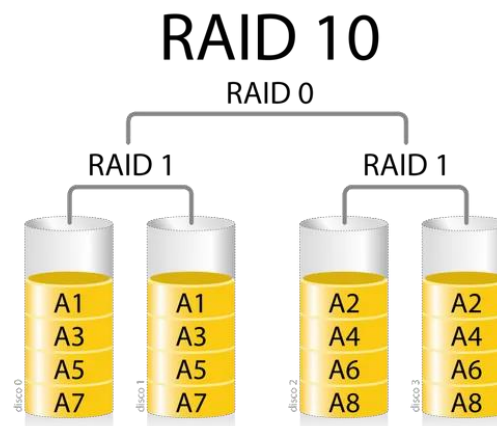
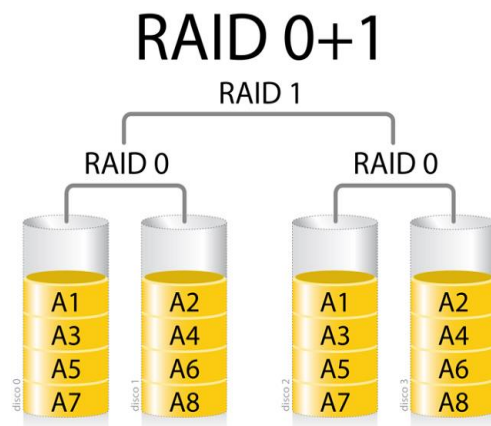
Los niveles RAID más empleados son:

- **RAID 0:** distribuye los datos en dos o más (N) discos duros sin duplicar la información, es decir, sin redundancia de datos. No aumenta la integridad, pero sí la velocidad de lectura y escritura.
 - Esta configuración **no permite que ningún disco falle** sin perder datos.
- **RAID 1:** establece una copia exacta entre dos o más (N) discos duros. Esto permite aumentar la integridad de los datos y la velocidad de lectura (si hay muchos procesos solicitando lecturas en paralelo).
 - Esta configuración **permite que N-1 discos fallen** sin perder datos.
- **RAID 5:** requiere tres o más (N) discos. Realiza una distribución de los bloques de datos en los discos, como RAID 0, pero por cada N-1 bloques de datos en N-1 discos, genera en el disco restante un bloque de **paridad** o de **detección de error** (CDC) calculado a partir de los otros bloques. Esto aumenta la integridad sobre RAID 0, pero reduce la velocidad de las escrituras, al tener que calcular una paridad.
 - Esta configuración **permite que 1 disco falle** sin perder datos, ya que un bloque de paridad puede suplir un bloque perdido.

Tipos de RAID

Se pueden combinar los niveles RAID:

- **RAID 01:** se establece una copia exacta (RAID 1) en dos o más configuraciones RAID 0. Se requieren 4 o más discos.
- **RAID 10:** los datos se distribuyen (RAID 0) en dos o más configuraciones RAID 1. Se requieren 4 o más discos.
- **RAID 50:** los datos se distribuyen (RAID 0) en dos o más configuraciones RAID 5. Se requieren 6 o más discos.



Administración de RAID

En Linux, la gestión y administración de RAID software se realiza mediante el paquete `mdadm` (*Multiple Device Administrator*).

Para comprobar si existe algún dispositivo RAID en el sistema, visualizamos el archivo `/proc/mdstat`, y en la línea `Personalities` podemos comprobar los tipos de RAID soportados.

La creación de RAID se puede hacer sobre dispositivos o sobre particiones, incluso de distinto tamaño (aunque lo óptimo es usar discos de la misma capacidad).

Administración de RAID

Para crear un RAID:

```
# mdadm --create /dev/mdX --level=Y --raid-devices=Z <dispositivos...>
```

Donde:

- /dev/mdX será el nombre del nuevo dispositivo, siendo X un número.
- Y es el nivel RAID para aplicar (raid0, raid1, raid5, raid10, etc.).
- Z es el número de dispositivos que se usarán para el RAID, y <dispositivos...> es la lista de dispositivos.

Para añadir un dispositivo al RAID, quitarlo y examinar el estado de un RAID:

```
# mdadm /dev/mdX --add <dispositivo>
# mdadm /dev/mdX --remove <dispositivo>
# mdadm --detail /dev/sdX
```

Administración de RAID

Para eliminar un RAID (ejemplo con /dev/md0):

- Desmontar el dispositivo RAID si está en uso (si está en `/etc/fstab`, eliminar la línea):

```
# umount /dev/md0
```

- Detener el RAID:

```
# sudo mdadm --stop /dev/md0
```

- Borrar el superbloque de cada dispositivo que constituía el RAID:

```
# sudo mdadm --zero-superblock <dispositivo>
```

Búsqueda de archivos

Los sistemas de archivos suelen ofrecer herramientas para localizar archivos según diferentes criterios (fecha de creación, de modificación, tamaño, nombre, tipo, etc.).

En Windows, el explorador de archivos provee una barra de búsqueda y un menú para hacer búsquedas por nombre o por otros criterios.

En Linux, se usa el comando `find`, que permite localizar archivos e imprimir la lista de las rutas o realizar alguna acción con ellos:

```
$ find <ruta> <criterio> <acción>
```

<ruta> es la ruta donde se comienza a buscar.

<acción> puede ser: `-print` para mostrar la lista (se usa por defecto), `-delete` para borrar los ficheros, `-exec` para ejecutar un comando personalizado.