

# **Gestión de Archivos**

# Gestión de archivos

- En un sistema informático, aunque la memoria principal es indispensable para ejecutar procesos, no es un medio válido para el almacenamiento de los programas y los datos a largo plazo por dos motivos fundamentales: **es volátil y no es suficientemente grande.**
- Para solventar estas carencias, **se utilizan los dispositivos de memoria secundaria**, como discos duros, unidades SSD, unidades USB, discos ópticos, etc.
- El sistema de archivos (en inglés, filesystem) es el componente del sistema operativo que se encarga de organizar el modo en el que se guardan los datos dentro de los dispositivos de almacenamiento secundario.

# Sistemas de Archivo por categorías

- **De disco**
- **De red:** para que un equipo pueda trabajar con un sistema de archivos de otro equipo como si fuera local (nfs) o compartir archivos por la red (smbfs)
- **Especiales:** swap, para la partición de intercambio de memoria en Linux.
- **Virtuales:** para que una determinada aplicación pueda tener acceso a diferentes sistemas de archivos (virtual FUSE, vfat, ...)

- Para llevar a cabo su tarea, desde un punto de vista lógico utiliza dos conceptos diferentes:
- Archivo (fichero)
  - Es una serie de bytes almacenados en un dispositivo de almacenamiento externo que, en conjunto, forman una unidad lógica. Cada archivo suele estar identificado en el sistema mediante un nombre y una extensión. Normalmente, el nombre sirve para identificar el contenido del archivo y la extensión para identificar el tipo al que pertenece (si es un documento, una imagen, etc.). Para evitar ambigüedades, no pueden existir dos archivos que tengan el mismo nombre y extensión dentro de la misma ubicación.
- Carpeta (directorio)
  - Es un modo de agrupar archivos, según el criterio del usuario, para facilitar su organización. Igual que los archivos, las carpetas tienen un nombre que las identifica. Para un sistema de archivos, una carpeta no es más que un archivo que contiene información sobre el modo en el que se organizan los datos. Como en el caso de los archivos, para evitar ambigüedades, no puede haber dos carpetas con el mismo nombre en la misma ubicación.

- La unidad de información con la que trabaja un sistema de archivos es, precisamente, el archivo.
- Es frecuente que cada familia de sistemas operativos tenga sus propios sistemas de archivos.

Sistemas de archivos más habituales		
Nombre	Sistema operativo	Otros sistemas operativos compatibles
FAT12, FAT16 y FAT32	Toda la familia de sistemas operativos de Microsoft (FAT32 a partir de DOS 7.1)	Linux y Mac OS
ExtFAT (versión FAT para memorias USB)	Toda la familia de sistemas operativos de Microsoft a partir de XP SP2	Mac OS X a partir de la versión 10.6.5
NTFS	Toda la familia de sistemas operativos de Microsoft a partir de NT (en los sistemas servidores) y XP (en los sistemas de escritorio)	Linux con núcleo a partir de la versión 2.2 y Mac OS X pero sólo en modo lectura.
HFS	Mac OS (con soporte de sólo lectura a partir de la versión OS X 10.6)	Linux
HFS Plus	Mac OS desde la versión 8.1	
ext2	Linux	Mac OS X
ext3	Linux	Mac OS X
ext4	Linux con núcleo a partir de la versión 2.6.28	Mac OS X (con la extensión ExtFS)
ReiserFS	Linux con un núcleo parcheado	

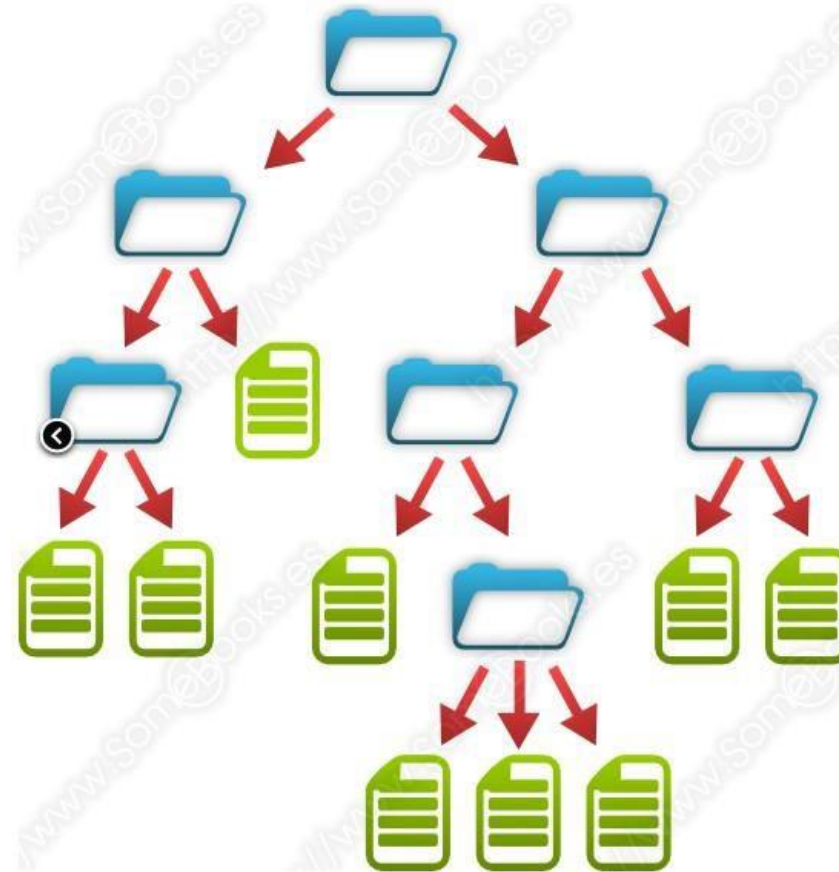
- A pesar de todo, existe cierto nivel de **compatibilidad entre los diferentes sistemas operativos en cuanto a los sistemas de archivos**, bien directamente como se muestra en la tabla, bien a través de herramientas complementarias.
- Es muy frecuente que **los datos se almacenen en un dispositivo de bloque**. En esos casos, un **archivo no será más que un conjunto de sectores de disco**. Estos **sectores no se leerán o escribirán de forma individual, sino que lo harán en grupos llamados clústers**.
- Por lo tanto, un sistema de archivos se encargará de aspectos como:
  - **Organizar** de forma lógica los sectores del dispositivo para constituir archivos y directorios.
  - **Asignar espacio** de almacenamiento a los archivos y mantener el control sobre los sectores que pertenecen a cada archivo.
  - Ofrecer los **procesos** que permitan crear nuevos archivos, cambiarles el nombre y/o la ubicación, modificar su contenido o eliminarlos.
  - Mantener la **estructura jerárquica** del sistema de directorios.
  - **Controlar el acceso** seguro a los archivos, es decir, que sólo puedan acceder a los datos los usuarios autorizados.
  - **Controlar** qué **sectores** permanecen **disponibles** para ser ocupados en cualquier momento.

# Atributos y permisos

- El sistema operativo debe tener la capacidad de controlar qué usuario puede acceder a cada uno de sus recursos (directorios, impresoras, conexiones de red, etc.). Para lograrlo, cada uno de estos recursos suele tener asociada una Lista de Control de Acceso o ACL (del inglés, Access Control List), en la que se relacionan los diferentes usuarios que pueden acceder y bajo qué condiciones (lectura, escritura, ejecución, ...)
- Por otro lado, un determinado archivo puede tener asociados diferentes atributos, que informan sobre ciertas características del archivo o del modo en el que el sistema operativo debe tratarlo. Así, un archivo puede tener atributos como: directorio, oculto, de sistema, cifrado, etc.

# Organización del sistema de archivos: nombres y rutas

- A día de hoy, prácticamente todos los sistemas de archivos que podemos encontrar organizan los archivos de forma jerárquica, permitiendo creación de un árbol de directorios que facilitan la organización y clasificación de su contenido.
- Además, para evitar ambigüedades, los sistemas operativos no permiten que, dentro mismo directorio, existan dos archivos con el mismo nombre.



la

del mismo



- Para expresar la ubicación exacta de un archivo o un directorio se utiliza su ruta (en inglés, path). Aunque la forma de expresar una ruta puede cambiar según el sistema operativo que estemos usando, en general se forman indicando la lista jerárquica de directorios que representa el camino que debemos recorrer para llegar hasta un archivo o directorio. En esa relación, el último elemento será el propio archivo o directorio referenciado. Al escribir una ruta, deberemos utilizar un carácter que separe cada elemento del siguiente. **Este carácter separador suele ser una barra invertida (\) en los sistemas Microsoft y una barra inclinada (/) en la familia de sistemas Unix, como es el caso de GNU/Linux.**

Además de todo lo dicho hasta ahora, también existen dos modos diferentes de escribir rutas:

- De forma **absoluta**
  - Una ruta de este tipo hará referencia a un archivo o directorio a partir del directorio raíz. En los sistemas de la familia Unix, una ruta absoluta comienza por una barra inclinada. Por ejemplo:  
`/home/Alicia/Documentos/informe.odt`  
En los sistemas Microsoft, es necesario comenzar la ruta absoluta con la letra de unidad a la que hace referencia, seguida de dos puntos y una barra invertida. Por ejemplo:  
`C:\Usuarios\Alicia\Documentos\informe.odt`

- De forma **relativa**

- Una ruta de este tipo hará referencia a un archivo o directorio tomando como punto de partida el directorio en el que nos encontramos. Por ejemplo:

`../../Jacinto/Documentos/memoria.odt`

Como puede apreciarse, el ejemplo es válido en sistemas de la familia Unix, pero bastaría con cambiar el sentido de las barras para que fuese válido en sistemas Microsoft.

- **NOTA**

- Las rutas que identifican recursos compartidos en una red suelen hacer uso de la nomenclatura UNC (en inglés, Universal Naming Convention), que fue definida por Microsoft y que tiene la siguiente sintaxis general:

`\\Servidor\RecursoCompartido\ruta\archivo`

## Caracteres comodín

- Un carácter comodín es aquél que tiene la capacidad de representar a cualquier otro carácter o conjunto de caracteres. Se utilizan para sustituir parte del nombre de un archivo o de alguno de los directorios que forman su ruta. Esto nos permite hacer referencia a un archivo del que no recordamos su nombre completo o bien a un grupo de archivos que tengan en común una parte de su nombre. Los dos caracteres comodín mas utilizados son:
  - El asterisco (\*), que representa a cualquier combinación de caracteres e incluso cubre la posibilidad de que no haya ningún carácter.
  - El signo de interrogación (?), que representa a un único carácter.

# Tipos de sistemas de archivos y sus características

A grandes rasgos, en los sistemas operativos Microsoft podemos encontrar dos tipos de sistemas de archivos totalmente diferentes:

- **FAT (del inglés, File Allocation Table)**

Se trata de un sistema de archivos muy simple que acompañaba a los primeros sistemas operativos del fabricante y no tiene ninguna característica de seguridad, salvo la básica que aportan los atributos de los archivos (sólo lectura, ocultos y de sistema) y sus archivos se fragmentan con relativa facilidad.

Sus diferentes versiones acompañaron a las versiones de escritorio de los sistemas operativos Microsoft hasta Windows Me.

- Las diferentes variantes, en orden cronológico son:

- **FAT12**

Se implementó por primera vez en QDOS en 1980. Era capaz de direccionar un máximo de 32 MB y usaba nombres de archivo con un máximo de 8 caracteres y 3 de extensión.

- **FAT16**

Se implementó sobre MS-DOS en 1980. Direccionaba hasta 90 GB usando clusters de 32 KB.

- **VFAT**

Se implementó sobre Windows 3.11, que usaba el modo protegido de 32 bits de los procesadores de la época y accedía directamente al hardware y a la memoria caché, lo que aceleraba su funcionamiento.

En Windows 95 se añadió una mejora que permitía el uso de nombres largos (hasta 255 caracteres).

- **FAT32**

Apareció con Windows 95 OSR2. Por diseño podría direccionar casi 8 TB, pero Microsoft lo limitó primero a 128 GB y, más adelante, con Windows XP a 32 GB. El tamaño máximo de archivo es de 4 GB.

- **ExFat**

Se presentó con Windows Embedded CE 6.0. Se trata de una mejora, adaptada a memorias USB.

- **NTFS (del inglés, New Technology File System)**

Comparado con FAT, ofrece un mayor rendimiento, seguridad y fiabilidad (incluso puede recuperarse de algunos errores de disco de forma automática). Además, aplica técnicas de journaling. El único inconveniente es que sus estructuras de datos son demasiado grandes en dispositivos de almacenamiento pequeños.

Apareció con las versiones profesionales de los sistemas Microsoft (Windows NT), aunque se incorporó a las versiones de escritorio a partir de Windows XP.

Está basado en HPFS, el sistema de archivos desarrollado entre IBM y Microsoft para el sistema operativo OS/2 y permite direccionar volúmenes de hasta 16 TB (aunque puede llegar hasta los 16 EB (1 Exabyte =  $1024 \times 1024$  TB)).

Existen 5 versiones:

v1.0: Publicada con Windows NT 3.1 en 1993 v1.1:  
Publicada con Windows NT 3.5 en 1994 v1.2: Publicada  
con Windows NT 3.51 en 1995 v3.0: Publicada con  
Windows 2000 en 2000 v3.1: Publicada con Windows XP  
en 2001



# Sistemas de archivos de la familia Apple

A lo largo del tiempo, la familia de ordenadores del fabricante Apple ha incorporado dos sistemas de archivos diferentes:

- **MFS (Macintosh File System)**

Era el sistema incluido en el Macintosh 128K (enero de 1984). Soportaba nombres de archivo con 255 caracteres (aunque sólo se podían localizar por los primeros 63).

El tamaño máximo de volumen era de 256 MB, pudiendo tener archivos del mismo tamaño.

- **HFS (Hierarchical File System)**

También llamado Mac OS estándar: Se desarrolló para el sistema operativo Mac OS y podía usarse en discos duros, disquetes y CD-ROMs. Su principal objetivo era aumentar el rendimiento en dispositivos de almacenamiento de mayor capacidad (hasta 2TB, con archivos de 2GB como máximo). Aunque mantiene la restricción de localizar archivos sólo por los primeros 63. Reemplazó a MFS en septiembre de 1985.

- **HFS+ o HFS Plus**

También llamado Mac OS extendido: Se introdujo en enero de 1998. Utiliza técnicas de journaling, emplea Unicode para los nombres de archivo y directorio, y es capaz de direccionar 8 EB (y admite archivos del mismo tamaño)

# Sistemas de archivos de la familia GNU/Linux

Por su parte, la familia de sistemas GNU/Linux soporta una gran variedad de sistemas de archivos, aunque los más comunes son los siguientes:

- **ext2 (second extended filesystem)**

Se introdujo en enero de 1993 como una evolución de un sistema de archivos anterior, llamado ext. Admitía dispositivos de hasta 16 TB, con archivos de hasta 2 TB y nombres de 256 caracteres.

- **ext3 (third extended filesystem)**

Apareció en noviembre de 2001 con la versión 2.4.15 del núcleo de GNU/Linux. Mantenía su compatibilidad con ext2, pero añadía soporte para un registro por diario (journaling) y se apoyaba en un árbol binario balanceado que le otorgaba un mayor rendimiento. Admitía dispositivos con un máximo de 32 TB, con archivos que podían llegar a los 2 TB

- **ext4 (fourth extended filesystem)**

Se publicó en octubre de 2006 con la versión 2.6.19 del núcleo de GNU/Linux y añade mejoras de velocidad y de uso de la CPU. Además, admite dispositivos de hasta 1 EB, con archivos que pueden llegar a 16 TB.

- **ReiserFS**

Apareció a principios de 2001 con la versión 2.4.1 del núcleo de GNU/Linux. Fue el primer sistema de archivos soportado por el núcleo de GNU/Linux que incluía journaling. Admitía dispositivos de hasta 16 TB, con archivos de hasta 8 TB y nombres de 256 caracteres.

- **XFS**

Apareció en 1994 sobre IRIX, el Unix de la empresa Silicon Graphics Inc (actualmente SGI), pero su código fuente se liberó en mayo de 2000. Se añadió al núcleo de GNU/Linux en la versión 2.4.25, en el año 2004. Fue el primer sistema de archivos que ofreció journaling. Admite dispositivos de hasta 16 EB, con archivos que pueden llegar a 8 EB.

## **Transacciones. Sistemas transaccionales.**

En general, cuando hablamos de transacción, nos referimos a un acuerdo entre dos partes en la que se produce un intercambio. Normalmente, una transacción estará formada por varias acciones individuales, que será necesario completar en su conjunto para que el conjunto de la transacción pueda darse por terminada. Como es lógico, si la transacción se interrumpe antes de completarse satisfactoriamente, debe anularse en su totalidad.

El journaling, o registro diario, es el modo de implementar transacciones en un sistema de archivos. Se basa en crear un registro de diario (en inglés, journal) donde se anotan los datos necesarios para deshacer las modificaciones que se hayan hecho durante la transacción por si ésta fallase.

Aplicando journaling, se evita la corrupción del sistema de archivos ante cortes en la alimentación eléctrica o cualquier otro fallo inesperado. Cuando el sistema arranque después de una de estas situaciones, sólo tendrá que deshacer las transacciones inacabadas para volver a disponer de un sistema de archivos íntegro.