

Sistemas Informáticos

TEMA 4

Gestión de la Información

GESTIÓN DE LA INFORMACIÓN

1. EL SISTEMA DE ARCHIVOS

- El **sistema de archivos** determinará la estructura, nombre, forma de acceso, uso y protección de los archivos que se guardarán en el disco.
- Cada sistema operativo dispone de su propio sistema de archivos, pero el objetivo y función de todos ellos es el mismo: permitir al usuario un manejo fácil y lógico de sus archivos abstrayéndose de las particularidades de los dispositivos físicos empleados.

GESTIÓN DE LA INFORMACIÓN

1. EL SISTEMA DE ARCHIVOS

- En un sistema de archivos hay dos tipos fundamentales de objetos: los **directorios** y los **archivos**.
- Los **archivos** son los objetos encargados de contener los datos, mientras que los **directorios** son los objetos cuya misión principal es permitir una mayor organización de los archivos dentro del disco. Un directorio es un contenedor que puede contener archivos y, a su vez, otros directorios dentro de él..
- A los directorios también se les denomina **carpetas**.

GESTIÓN DE LA INFORMACIÓN

2. LOS ARCHIVOS

- Los archivos son la forma de almacenar información en el disco y poder volverla a leer más adelante sin que el usuario tenga que preocuparse por la forma y lugar físico de almacenamiento de la información así como del funcionamiento real de los discos.
- La estructura típica del nombre de un archivo en los diferentes sistemas de archivo es:
nombre.extensión.

GESTIÓN DE LA INFORMACIÓN

2. LOS ARCHIVOS

- Junto con el nombre del archivo, el sistema operativo almacena también unos atributos que califican al archivo. Estos **atributos** varían también de un sistema operativo a otro y, entre ellos, se pueden encontrar:
 - **S: atributo de sistema** (*system*): indica si el archivo pertenece al sistema operativo o no.
 - **H: atributo de oculto** (*hidden*): indica si el archivo está oculto. En este caso, no se visualizará al hacer un listado del directorio.
 - **R: atributo de solo lectura** (*read only*): indica si el archivo es de solo lectura o se permite también su escritura.
 - **A: atributo de archivo**: este atributo se suele cambiar cuando se modifica el archivo. Su mayor utilidad es poder determinar qué archivos se modificaron desde la última copia de seguridad.
 - **Fecha**: es el atributo que almacena la fecha de creación o modificación del archivo.
 - **Hora**: es el atributo que almacena la hora de creación o modificación del archivo.
 - **Tamaño**: almacena el tamaño que ocupa el archivo.

Hay sistemas operativos (como Linux) que utilizan también atributos para indicar la pertenencia del archivo a un determinado usuario o grupo, y los permisos que el propietario del fichero tiene sobre dicho archivo (lectura, escritura, ejecución) así como el tipo de archivo que es: archivo normal, enlace físico, enlace lógico, directorio.

GESTIÓN DE LA INFORMACIÓN

2. LOS COMODINES

En cualquier sistema de archivos existen formas de recortar y facilitar las cosas más usuales, entre las que se encuentra facilitar la selección de ficheros.

Para ello, se disponen de los **comodines** que son de dos tipos:

- “*”: sustituye a TODOS los caracteres (delante, detrás o en medio del nombre).
- “?”: sustituye a un CARÁCTER para que coincida con el resto que esté escrito.

GESTIÓN DE LA INFORMACIÓN

2. TIPOS DE ARCHIVOS

Los archivos se pueden dividir en dos grandes grupos: los ejecutables (están creados para funcionar por sí mismos) y los no ejecutables o archivos de datos (que almacenan información que tendrá que ser utilizada con ayuda de algún programa).

Dentro de los archivos de datos se pueden crear categorías, por su temática o clase de información que almacenen. Pueden ser de:

- Sistema.
- Audio.
- Vídeo.
- Imágenes.
- Comprimidos.
- Texto.
- Imágenes de CD/DVD.
- Programas.

GESTIÓN DE LA INFORMACIÓN

2. OPERACIONES COMUNES CON ARCHIVOS

Entre las operaciones comunes que se pueden realizar con los archivos se encuentran :

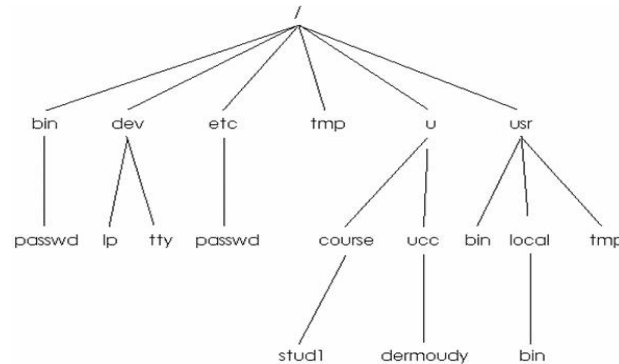
- **Crear:** con esta operación se añade un nuevo archivo.
- **Abrir:** en esta operación el método de acceso localiza e identifica un archivo existente para que los usuarios o el propio sistema operativo pueda operar con él.
- **Cerrar:** esta operación se utiliza para indicar que se va a dejar de utilizar un archivo determinado. Mediante esta operación el método de acceso finaliza la conexión entre el programa de usuario y el archivo, garantizando la integridad de los registros. Al ejecutar esta operación, el sistema se encarga de escribir en el dispositivo de almacenamiento aquella información que contienen los búferes asociados al archivo y se llevan a cabo las operaciones de limpieza necesarias.
- **Copiar:** con esta operación se crea una copia del archivo en otra ubicación quedando el archivo en el lugar de origen.
- **Mover:** con esta operación se quita el archivo del lugar de origen y se lleva a otra ubicación de destino.
- **Renombrar:** con esta operación se le cambia el nombre al archivo.
- **Eliminar:** con esta operación se borra el archivo de la ubicación en la que se encuentre.

GESTIÓN DE LA INFORMACIÓN

3. LOS DIRECTORIOS

Los directorios son una división lógica de almacenamiento de archivos u otros subdirectorios.

En todo sistema de archivos hay un directorio especial llamado **raíz** (*root* en inglés) que es el directorio que contiene todos los demás directorios y archivos (también se le puede identificar con la barra inclinada).



Desde este directorio es desde el que se parte cuando se busca un archivo mediante una **ruta de acceso absoluta**. En Windows, las rutas de acceso están separadas por el carácter `\`, mientras que en Linux se utiliza el carácter `/`. Así pues, una ruta absoluta en Linux tendrá el siguiente aspecto `/usr/bin/calc` y en Windows, `\pepe\lunes\programa.exe`.

Cuando se usa una **ruta de acceso relativa**, el archivo se busca partiendo del directorio en el que se esté trabajando o directorio activo. Las rutas relativas carecen de carácter inicial y, por tanto, no parten del directorio raíz sino del activo. Por ejemplo: `bin/calc` o `lunes\programa.exe`.

GESTIÓN DE LA INFORMACIÓN

3. LOS DIRECTORIOS

Muchos de los sistemas operativos que implementan un sistema jerárquico de directorios tienen dos entradas especiales en cada uno de sus directorios «.» y «..» que hacen referencia respectivamente al directorio activo y a su padre, es decir, al propio directorio y al directorio de nivel superior que contiene a éste.

Los nombres de los directorios pueden tener extensión al igual que el nombre de un archivo y es recomendable que sea lo más descriptivo posible de los archivos que contiene.

Al igual que ocurría con los archivos, el sistema operativo almacena también unos atributos que califican al directorio.

Hay sistemas operativos (como Linux) que utilizan también atributos para indicar la pertenencia del directorio a un determinado usuario o grupo, y los permisos que el propietario del directorio tiene sobre él (lectura, escritura, ejecución).

Otros sistemas operativos (como Windows) permiten indicar si los archivos de cada directorio estarán cifrados o comprimidos.

GESTIÓN DE LA INFORMACIÓN

4. IMPLEMENTACIÓN DEL SISTEMA DE ARCHIVOS

Un **bloque** está compuesto por un determinado número de sectores que se asocian a un único archivo. Un archivo, por tanto, se almacena en uno o más bloques de sectores.

Un aspecto muy importante es la elección del tamaño del bloque, para esto hay que entender que si el tamaño del bloque es muy grande, aun cuando el archivo sea de un tamaño muy pequeño, se le asignará el bloque entero con lo que se desperdiciará gran parte de la capacidad del disco.

Por otra parte, si el tamaño del bloque es demasiado pequeño para almacenar un archivo, harán falta muchos bloques con lo que se producirá un retraso en la lectura del archivo al tener que localizar en el disco todos los bloques que componen dicho archivo. Una vez más, se ha de llegar a una solución de compromiso, eligiendo un tamaño del bloque lo suficientemente pequeño para no desperdiciar capacidad de disco pero lo suficientemente grande como para no ralentizar en exceso la lectura de los archivos.

Si se elige un tamaño de bloque de, por ejemplo, 2 KB en un disco cuyo sector tiene 512 bytes, cada bloque estará compuesto por cuatro sectores.

GESTIÓN DE LA INFORMACIÓN

4. IMPLEMENTACIÓN DEL SISTEMA DE ARCHIVOS

Para manejar los bloques asociados a cada archivo, se pueden utilizar varias técnicas. La primera de ellas consiste en almacenar los archivos mediante bloques adyacentes en el disco (de esta forma, en el directorio únicamente se tendrá que guardar la dirección en la que comienza el primer bloque, ya que los demás se encuentran a continuación). A esta técnica se la denomina **asignación adyacente**. Su gran ventaja es su fácil implementación, pero tiene el gran problema de que es necesario conocer con anterioridad el número de bloques que ocupará el fichero y esto, en general, no ocurre. Además genera una gran fragmentación del disco, que produce una pérdida de espacio.

Otra técnica que solventa algunas de las carencias de la asignación adyacente es la **asignación en forma de lista ligada**. En esta técnica, el directorio contiene la dirección del primer bloque y cada bloque contiene, a su vez, la dirección del siguiente bloque o el valor *null* (nulo) en caso de que sea el último bloque del fichero. Con esta técnica se consigue aprovechar todos y cada uno de los bloques del disco y se evita perder capacidad por la fragmentación.

GESTIÓN DE LA INFORMACIÓN

4. IMPLEMENTACIÓN DEL SISTEMA DE ARCHIVOS

Una tercera técnica es la llamada **asignación mediante una lista ligada y un índice**, que intenta eliminar los defectos de la anterior. En esta técnica se crea una tabla con un registro por cada uno de los bloques del disco, en cada registro se indica si dicho bloque está libre (*null*) o cuál es la dirección del siguiente bloque (en caso de que ese bloque pertenezca a un determinado archivo). De esta forma, en el directorio se asocia con el nombre del archivo el número de bloque en el que comienza dicho archivo; con este dato y mediante la tabla, se puede averiguar la dirección de todos los bloques que componen dicho archivo simplemente siguiendo la lista ligada.

Con esta organización, todo el bloque estará disponible para los datos. Además, el acceso a un determinado bloque es mucho más rápido, ya que aunque también haya que seguir la cadena de bloques como en la asignación en forma de lista ligada, al estar la tabla en memoria, estas consultas son mucho más rápidas y no es necesario acceder a disco.

Con esta organización, todo el bloque está disponible para los datos. Además, el acceso a un determinado bloque es mucho más rápido ya que, aunque también haya que seguir la cadena de bloques como en la asignación en forma de lista ligada, al estar la tabla en memoria, estas consultas son mucho más rápidas y no es necesario acceder al disco.

GESTIÓN DE LA INFORMACIÓN

5. TIPOS DE SISTEMAS DE ARCHIVOS

- **FAT16** (también denominado **FAT**): se puede acceder a este sistema de archivos desde *Windows 95* y sucesivos. Permite trabajar con particiones de hasta 2 GB, las unidades de asignación son de 32 KB, el tamaño máximo de un archivo es de 2 GB, los volúmenes pueden llegar hasta 2 GB, no distingue entre mayúsculas y minúsculas en los nombres de archivos/directorios.
- **FAT32**: se puede acceder a este sistema de archivos desde *Windows 98* y sucesivos. Permite trabajar con particiones mayores de 2 GB, las unidades de asignación son de 4 KB, el tamaño máximo de un archivo es de 4 GB, los volúmenes pueden llegar hasta 2 TB (en *Windows 2000/XP/Vista/2003/2008* solo hasta 32 GB por decisión de Microsoft, aunque hay utilidades que permiten sobrepasar dicho límite), no distingue entre mayúsculas y minúsculas en los nombres de archivos/directorios.
- **NTFS 5** (*NT File System 5*): se puede acceder a este sistema de archivos desde *Windows 2000* y sucesivos. Permite nombres de archivo de hasta 256 caracteres, ordenación de directorios, atributos de acceso a archivos, reparto de unidades en varios discos duros, reflexión de discos duros y registro de actividades, Se puede acceder al Directorio Activo, dominios de *Windows 2000/2003/2008*, utilizar cuotas en disco para cada usuario, cifrado y compresión de archivos, almacenamiento remoto, dispone de una herramienta de desfragmentación y utilización de enlaces de archivos similares a los realizados en *Linux*. Sus volúmenes pueden llegar hasta 16 TB menos 64 KB y el tamaño máximo de un archivo solo está limitado por el tamaño del volumen. Distingue entre mayúsculas y minúsculas en los nombres de archivos/directorios.

GESTIÓN DE LA INFORMACIÓN

5. TIPOS DE SISTEMAS DE ARCHIVOS

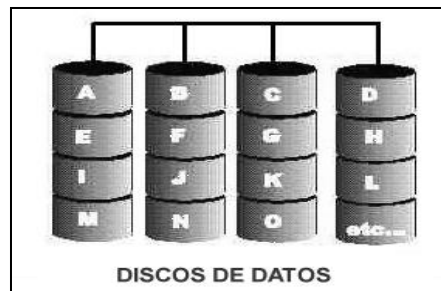
- **Sistema de Archivos Extendido 3 (*ext3fs*)**: es uno de los más eficientes y flexibles sistemas de archivos. Se puede acceder desde *Linux*, permite hasta 256 caracteres en los nombres de los archivos, el tamaño máximo de un volumen es de 32 TB y el tamaño máximo de un archivo es de 2 TB. Distingue entre mayúsculas y minúsculas en los nombres de archivos/directorios. Así mismo, dispone de un **registro de diario** que permite almacenar la información necesaria para restablecer los datos afectados por una transacción en caso de que ésta falle.
- **Sistema de Archivos Extendido 4 (*ext4fs*)**: es uno de los más eficientes y flexibles sistemas de archivos. Se puede acceder desde *Linux*, es compatible con **ext3**, permite hasta 256 caracteres en los nombres de los archivos, el tamaño máximo de un volumen es de 1 EB y el tamaño máximo de un archivo es de 16 TB. Distingue entre mayúsculas y minúsculas en los nombres de archivos/directorios. Así mismo, dispone de un **registro de diario** que permite almacenar la información necesaria para restablecer los datos afectados por una transacción en caso de que ésta falle.
- **HPFS (*High Performance File System*)**: en su origen se creó para *OS/2* y resolvía los problemas del sistema de archivos *FAT*. Permite hasta 256 caracteres en los nombres de los archivos, el tamaño máximo de un volumen es de 2 TB y el tamaño máximo de un archivo es de 2 GB.

GESTIÓN DE LA INFORMACIÓN

6. LA TOLERANCIA A FALLOS

El término **RAID** significa *Redundant Array of Independent Disks* y su filosofía consiste en disponer de varias unidades de disco conectadas entre sí, por medio de controladoras, software o combinación de ambos, de manera que, cuando una unidad física de disco falle o se venga abajo, los datos que se encontraran en dicha unidad no se pierdan sino que se reconstruyan usando la paridad de los mismos (el sistema operativo ve a la matriz como si ésta fuese una sola).

- **RAID 0:** la información se divide entre todos los discos del sistema, de forma que no se establece ningún tipo de redundancia.

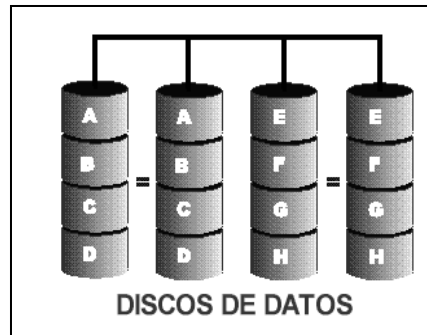


- **Ventajas:** proporciona alto rendimiento, tiempos de acceso muy bajos y posibilidad de acceso en paralelo. No tiene coste adicional. Se emplea toda la capacidad del disco.
- **Inconvenientes:** no es verdaderamente un disco RAID ya que no presenta integridad de los datos. Un error en uno de los discos implica la pérdida total de los datos.

GESTIÓN DE LA INFORMACIÓN

6. LA TOLERANCIA A FALLOS

- **RAID 1:** también conocido como **MDA** (*Mirrored Disk Array*). En esta configuración los discos se asocian por parejas y cada una de ellas almacenará la misma información. Cada pareja está formada por un disco *primario*, donde se leen y se escriben los datos, y un disco *espejo*, donde solamente se escriben las modificaciones y en el que se leerán datos cuando el primario falle.



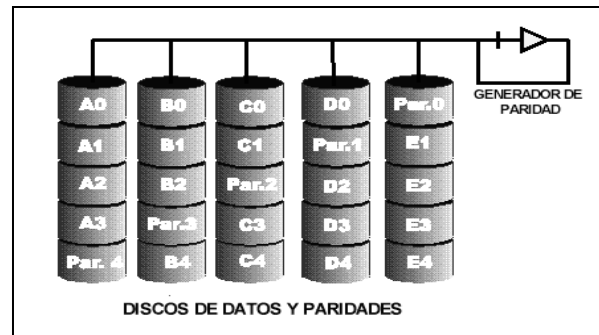
- **Ventajas:** en caso de error de uno de los discos se recuperan todos los datos. Es la arquitectura más rápida que presenta tolerancia a fallos. Con un mínimo de dos discos es suficiente.
- **Inconvenientes:** es bastante caro, ya que se emplea el doble de espacio del necesario.

GESTIÓN DE LA INFORMACIÓN

6. LA TOLERANCIA A FALLOS

- **RAID 5:** es un sistema de discos independientes con integración de códigos de error mediante paridad, en donde los datos y la paridad se guardan en los mismos discos, por lo que se consigue aumentar la velocidad de demanda.

La paridad nunca se guarda en los discos que contienen los datos que han generado dicha paridad, ya que, en el caso de que uno de ellos se estropeará, bastaría con regenerar los discos para que el dato pudiera volver a reestablecerse.



- **Ventajas:** alto rendimiento en aplicaciones con gran demanda de velocidad. No se desaprovecha ningún disco exclusivamente para almacenar códigos de paridad. Se pueden recuperar los datos.
- **Inconvenientes:** bajo rendimiento en escrituras. Se requiere un mínimo de tres discos.

Bibliografía

Sistemas Informáticos. José Luis Raya Cabrera, Laura Raya González, Javier S. Zurdo.
Ed. Ra-Ma 2011