

TEMA 6: Gestión de ficheros

1. El sistema de gestión de ficheros

Es la parte del Sistema Operativo que se encarga del manejo y la organización de los ficheros. Administra la memoria secundaria y estructura el Sistema de Ficheros sobre ella. También implementa los algoritmos y estructuras de datos necesarios para hacer corresponder la visión del usuario con el sistema físico de almacenamiento secundario.

Las funciones del sistema de gestión de ficheros son:

- Interfaz de llamadas al sistema (recibir llamadas al sistema sobre gestión de ficheros).
- Gestionar ficheros (localizar los bloques del disco que contienen los datos de un fichero, conectando la imagen lógica del fichero con la física del disco).
- Gestionar directorios (localizar ficheros en la jerarquía de directorios y proporcionar sistemas de nombrado lógico de los ficheros).
- Gestionar el espacio libre de los discos (localizar bloques libres para asignarlos a nuevos ficheros o liberar bloques cuando se borra un fichero).
- Gestionar la caché de disco localizada en memoria.
- Proporcionar protección (efectuar el control de acceso a los ficheros).
- Comunicarse con el sistema de Gestión de E/S (manejador de dispositivos) para indicarle la operación a realizar sobre los bloques físicos del disco.

2. Ficheros

Los ficheros son abstracciones que utiliza el Sistema Operativo para almacenar datos de manera persistente. Hay muchos tipos de ficheros, pero hoy en día son secuencias de bytes. Los métodos de acceso son los siguientes:

- Secuencial (la información se procesa de forma secuencial)
- Directo (por dirección) (permite el acceso directo a cualquier dirección del mismo)

El sistema guarda información del fichero en un **Descriptor de Fichero** que suele ser: **i-nodo** de Unix, **registro de MFT** de Windows NT, directorios de MS-DOS.

El disco se organiza en bloques (bloques físicos del dispositivo).

Los ficheros se dividen en bloques (bloques lógicos del fichero).

Hay dos posibilidades de asignación:

- Asignación contigua (se guardan todos los bloques del fichero en bloques contiguos del disco).
- Asignación no contigua (los bloques del fichero no están almacenados consecutivamente en disco) Por ejemplo en los discos duros, dónde será necesario utilizar una estructura que relacione bloque lógico-físico.

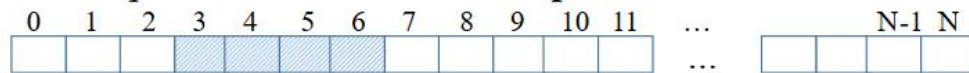
Para cada fichero, hará falta almacenar en su descriptor:

- Bloque de inicio del fichero.
- Número de bloques que ocupa.

En caso de aumento del tamaño del fichero, si no hay espacio libre a continuación del que tiene asignado no podrá crecer.

En un sistema con alto grado de volatilidad provocan fragmentación, con lo que habrá que compactar periódicamente.

Fichero: bloques 3, 4, 5, 6. En el descriptor: 3



Los bloques de cada fichero no tienen que estar almacenados en posiciones contiguas. Aunque la fragmentación no es desde el punto de vista lógico un problema, sí puede serlo en cuanto al rendimiento.

Para localizar los bloques que pertenecen a cada fichero puede utilizarse una lista con los bloques que pertenecen al fichero. Esta lista puede implementarse de la siguiente forma:

- Asignación con lista enlazada en los propios bloques con datos.
- Asignación FAT.
- Asignación indexada con tabla de índices.
- Asignación indexada con árbol equilibrado.

A continuación se explica cada una de las implementaciones.

1. Asignación enlazada

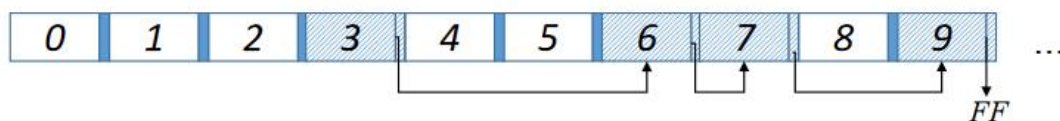
Dentro de cada bloque de datos existe un puntero al siguiente bloque de datos.

En el descriptor se almacena la dirección del primer bloque del fichero.

Hay que recorrer los bloques anteriores para llegar a uno dado.

El puntero ocupa un espacio y está incluido en el fichero

Fichero: bloques 3, 6, 7, 9. En el descriptor: 3



2. Asignación FAT (File Allocation Table)

El sistema mantiene en el disco una tabla con una entrada para cada bloque del disco.

En el descriptor se almacena la dirección del primer bloque del fichero.

En la entrada FAT correspondiente al primer bloque del fichero se almacena la dirección del segundo. En la entrada correspondiente a la del segundo, la del tercero, ...

En la entrada correspondiente al último bloque del fichero se almacena una marca de fin de fichero.

Para acceder directamente a un bloque hay que hacer un recorrido por la FAT, no por los bloques. FAT16 utiliza 16 bits para cada entrada de la FAT.

Fichero: bloques 3, 6, 7, 9. En el descriptor: 3

0	1	2	3	4	5	6	7	8	9	...
---	---	---	---	---	---	---	---	---	---	-----

Bloque	Siguiente
1	
2	
3	6
4	
5	
6	7
8	

Bloque	Siguiente
9	FF
10	
11	
12	
13	
14	
15	

3. Asignación indexada con tabla de índices

Se guardan todos los números de bloque (índice) de un fichero en una estructura llamada tabla de índices. Esta tabla se almacena en bloques de disco.

En la i-ésima entrada se localiza el índice i-ésimo bloque del fichero.

Existe en los discos de dos tipos de bloques:

- De datos
- De índices (bloques con números bloques físicos del fichero)

En cada bloque de índices caben muchos índices de bloques:

- Tamaño del bloque / nº de bytes del índice
- Ej: 4K / 4 Bytes = 1024 índices caben en un bloque de 4K

Fichero: bloques 3, 6, 7, 9. En el descriptor: 1

0	1	2	3	4	5	6	7	8	9	...
---	---	---	---	---	---	---	---	---	---	-----

3
6
7
FF

4. Asignación indexada con Árboles equilibrados

Representa la estructura de un fichero mediante árboles equilibrados (con número fijo de hijos en cada nivel).

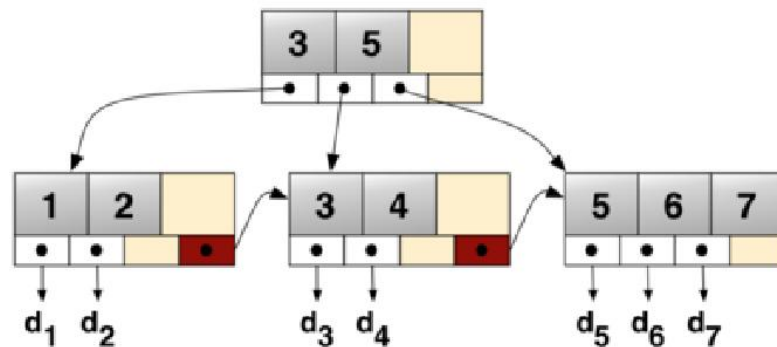
Cada bloque contiene datos y punteros a los hijos.

La raíz del árbol se almacena en el descriptor del fichero.

En cada nodo puede haber datos y enlaces a otros nodos.

La clave de búsqueda es el número lógico del bloque.

Los accesos directos por dirección pueden ser muy rápidos.



Ejemplo de árbol B

Además de todo esto, el SO tiene que saber qué bloques del disco están libres, para poder asignarlos a ficheros cuando sea necesario. Hay varias posibilidades:

- Usar un mapa de bits, con una entrada para cada bloque.
- En los sistemas con FAT cada entrada de la propia FAT se utiliza para indicar si el bloque está libre.
- Usar una lista conteniendo los bloques libres. Esta lista puede estar implementada de una forma similar a lo visto para los ficheros (lista enlazada, por índices, ...).

3. Directorios

Es un objeto que relaciona el nombre simbólico de un fichero con el descriptor interno usado por el SO.

Hay varias estructuras de los directorios:

- De uno o dos niveles.
- De árbol:
 - Nodo raíz, nodos internos son directorios y hojas son ficheros.
- De grafos acíclicos:
 - Útil para compartición de directorios por usuarios diferentes.
 - Se crean enlaces a archivos compartidos.

El nombre del fichero lo marca el camino desde el directorio raíz hasta el fichero.

La estructura y almacenamiento de los directorios se caracteriza por:

- En ficheros contiguos, almacenan atributos y número del primer bloque y tamaño.
- En ficheros FAT, almacenan atributos, número del primer bloque y tamaño.
- En ficheros indexados, identificador del descriptor del fichero.

4. Estructura general del Sistema de Ficheros en disco

Los discos se dividen en una o varias particiones o volúmenes, llamada, Partición, que es una porción del disco a la que se le dota de entidad propia y puede ser manipulada por el SO como entidad independiente.

Algunos SO permiten construir particiones extendidas que abarcan varios discos.

Sistema de fichero: modo de organización de la información en el disco, en un formato inteligible para el SO.

El bloque es la unidad de transferencia mínima (1 o más sectores).

El sistema de ficheros (partición) se divide en bloques numerados lógicamente del 0 al máximo del sistema de ficheros.

Los ficheros también se dividen en bloques (de igual tamaño) numerados lógicamente del 0 al máximo del fichero.

A cada bloque lógico del fichero se le asigna un bloque lógico del sistema de ficheros (denominado en ocasiones bloque físico).

El tamaño del bloque influye en:

- Uso del disco. Cada fichero tiene un numero entero de bloques.
- Velocidad de acceso.
- Tamaños grandes producen fragmentación interna, pero son más rápidos.
- Tamaños típicos (512, 1K, 2K).

Información almacenada en los bloques del disco (en general):

- Información de arranque del SO (boot)
- Metainformación sobre sistema de ficheros
- Bloques de datos