Concepto de fichero

Un fichero es un espacio de direccionamiento lógico contiguo. Es la unidad lógica de almacenamiento secundario. Se clasifican en dos grupos:

- **Ejecutables**: Son ficheros que él SO puede entender. El cargador de programas reconoce su estructura y lo prepara para ejecución.
- **De datos**: Los programas pueden estructurar la información que quieran en ellos usando caracteres o bytes

Atributos

Son metadatos (datos sobre los datos) asociados a cada fichero. Los atributos de los ficheros se mantienen en la estructura del directorio que contiene elementos como el nombre, identificador, localización, tamaño,...

Acceso a ficheros

Los ficheros se componen de bloques físicos de tamaño fijo que son la unidad de transferencia entre disco y memoria. El SO ofrece syscalls para acceder a ficheros a nivel de byte. La lectura de dicho fichero se puede realizar de forma secuencial o aleatoria.

Operaciones con ficheros

Creación, borrado, escritura/lectura, posicionamiento interno del fichero, leer y modificar atributos, abrir y cerrar ficheros

Dispositivos de almacenamiento masivo

Dispositivos de almacenamiento no volátiles. Se accede a sus celdas de memoria a través de los controladores de E/S del computador. Se clasifican en función de la tecnología utilizada:

- Cintas magnéticas: Acceso secuencial muy barato
- Discos magnéticos: Permite acceso aleatorio relativamente rápido
- Discos de estado sólido: Más resistentes y rápidos. Caros

HDDs

Estructura física

El disco está dividido en sectores que el controlador puede leer o escribir. Cada uno de estos sectores contiene 3 campos (Preamble, data, ECC). En la cabecera y la cola del sector se almacenan los metadatos del controlador del disco. Se llama cilindro a aquellos sectores accesibles sin mover la cabeza lectora

Estructura lógica

El disco se divide en particiones (exceptuando algunos casos como swap). Cada volumen es una entidad (partición, archivo, imagen de CD,...) que contiene un sistema de ficheros. Desde el punto de vista de las aplicaciones, el disco se ve como un array unidimensional de bloques lógicos identificados por su dirección lógica

Rendimiento

Si el dispositivo está ocupado, la petición de E/S se encola. Se pueden aplicar algoritmos de planificación de peticiones para optimizar el acceso al disco. Los pasos para realizar la operación de E/S son:

- Espera de cola (t_{queue})
- Esperar acción del controlador ($t_{controller}$)
- Posicionar la cabeza lectora (t_{pos})
- Esperar la rotación del disco (t_{rot})
- Realizar la transferencia del archivo (t_{trans})

$$Latencia = t_{queue} + t_{controller} + t_{pos} + t_{rot} + t_{trans}$$

> Se obtiene la menor latencia leyendo secciones grandes de memoria

SSD

Estructura física

A diferencia de sectores, el SSD utiliza páginas. Estas páginas no pueden ser sobreescritas, sino que deben ser borradas primero para reprogramarlas. Normalmente esta operación se realiza en bloques para evitar estropear el disco

Estructura lógica

El controlador del SSD se encarga del procesamiento de las páginas del disco, llevando contabilidad de que páginas deben ser borradas y cuales no para gestionar la memoria eficientemente

Comparativa SSD vs HDD

- SSD más rápido
- SSD más resistentes
- SSD siempre tarda lo mismo en acceder a memoria
- SSD menor consumo energético
- HDD más barato
- HDD mayor capacidad
- HDD puede escribir teóricamente toda la vida

Sistemas de ficheros

Es una capa del SO que abstrae la interfaz de dispositivos de bloques, como discos duros, de manera que el usuario final maneja ficheros y directorios. Para el SO, los ficheros son una colección de bytes (bloques de memoria). Un sistema de ficheros se basa en los siguientes componentes:

- **Nombre**: Interfaz para encontrar los ficheros por nombre
- **Manejo del disco**: Mantener la lista de nombres que componen un fichero
- **Protección de memoria**: Seguridad de los datos
- **Fiabilidad/durabilidad**: Se encarga de mantener la integridad de los datos a pesar de los fallos en el disco

Implementación de un sistema de ficheros

Un sistema de ficheros está, generalmente, dividido en diferentes capas encargadas de la gestión de los ficheros. Además, el SO mantiene ciertas ED almacenadas en el disco encargadas del manejo de ficheros:

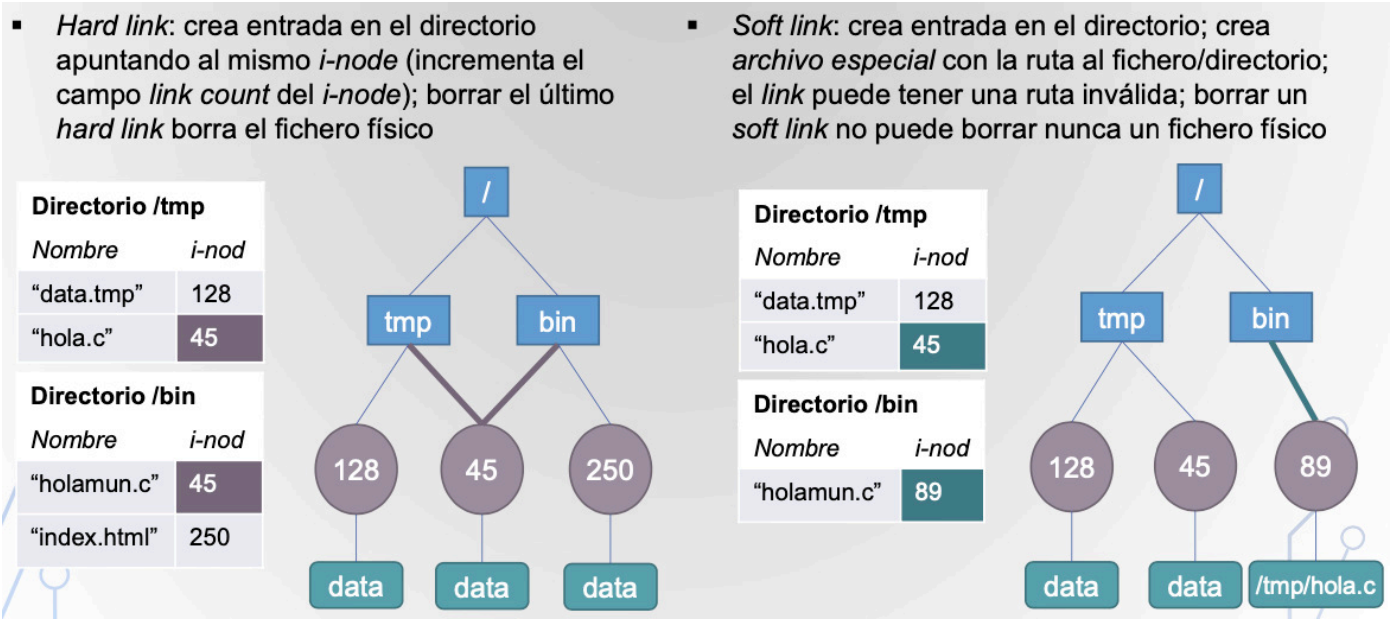
- Boot control block (por volumen): Contiene la información necesaria para arrancar un SO desde un volumen
- Volume control block (por volumen): Contiene metadatos del volumen como el número de bloques, su tamaño, información de los bloques libres, punteros a los FCB, ...
- Estructura de directorios (por sistema de ficheros): Usada para organizar los ficheros

  File control block (por fichero): Contiene metadatos del fichero. Incluye un identificador único para su asociación a un directorio

Estructura de los directorios

Los ficheros se organizan lógicamente en directorios (físicamente pueden estar en cualquier sitio). Los directorios son útiles para clasificar los ficheros almacenados de forma jerárquica. Los directorios son ficheros que almacenan otros ficheros (funcionamiento un poco especial).

Hard links vs soft links



Métodos de asignación de bloques a ficheros

Buscamos maximizar el rendimiento, facilitar el acceso y las operaciones básicas. Existen varios métodos para lograrlo

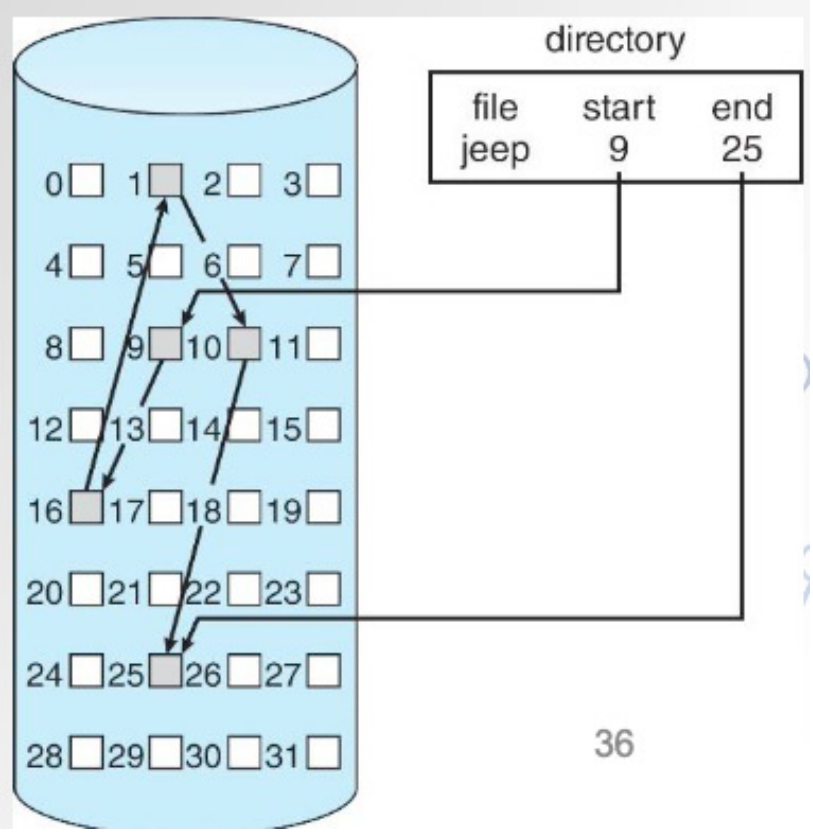
Asignación contigua

Cada fichero ocupa un conjunto contiguo de bloques en el disco. Un fichero se puede definir por la dirección del primer bloque y su tamaño en número de bloques. El acceso secuencial es rápido en HDD y el espacio libre se puede manejar con un vector de bits con política de asignación best fit o first fit

Inconvenientes: Fragmentación externa, un fichero puede tener tamaño indefinido

Asignación enlazada

Solventa los problemas de la asignación contigua. Cada fichero es una lista enlazada de bloques dispersos en el dispositivo de almacenamiento. El directorio contiene un puntero al **primer y el último bloque**, donde cada bloque tiene un puntero al siguiente. No tiene fragmentación externa pero su acceso aleatorio y secuencial empeora notablemente

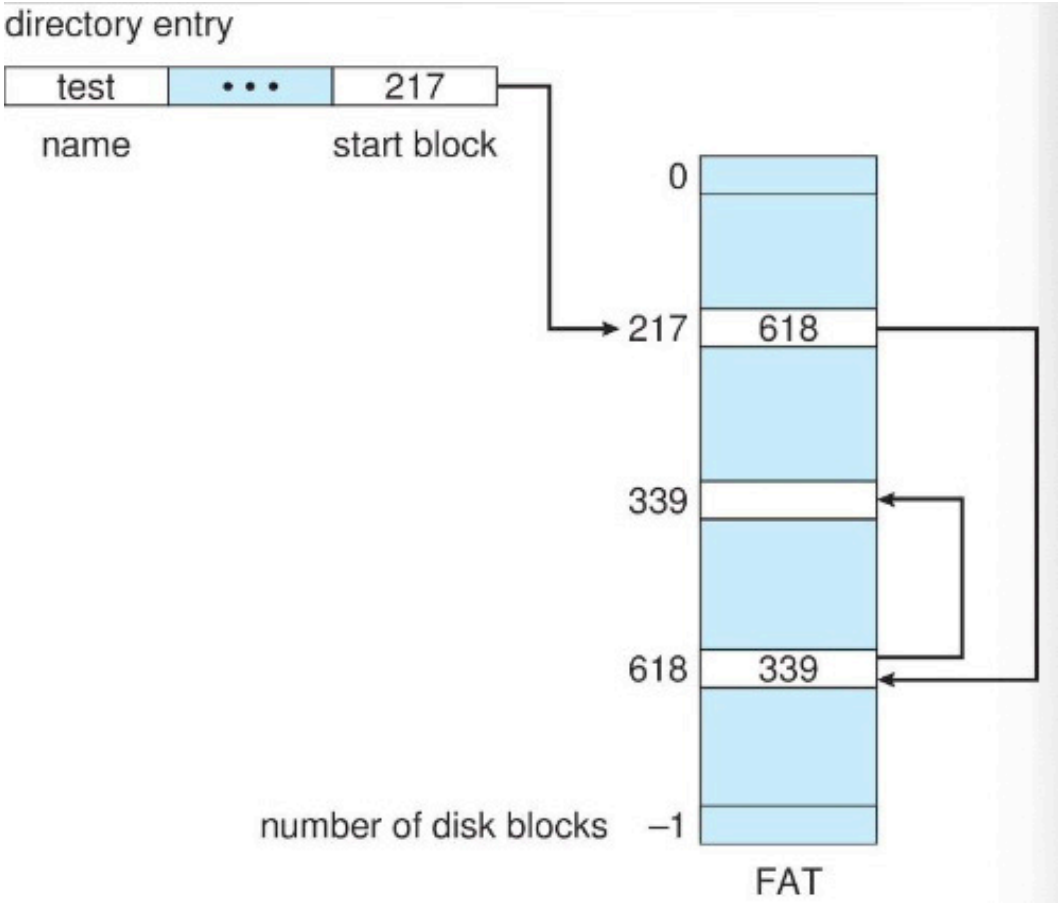


FAT

Se basa en la asignación enlazada, mejorando la velocidad de acceso aleatorio/directo. Los enlaces se mantienen en el interior de una estructura de datos (File Allocation Table) con una entrada por bloque del disco. En cada entrada se almacena un puntero al siguiente bloque del disco. El número del FAT indica el número de bits utilizados en los punteros.

> Si el puntero es negativo quiere decir que es el final del archivo

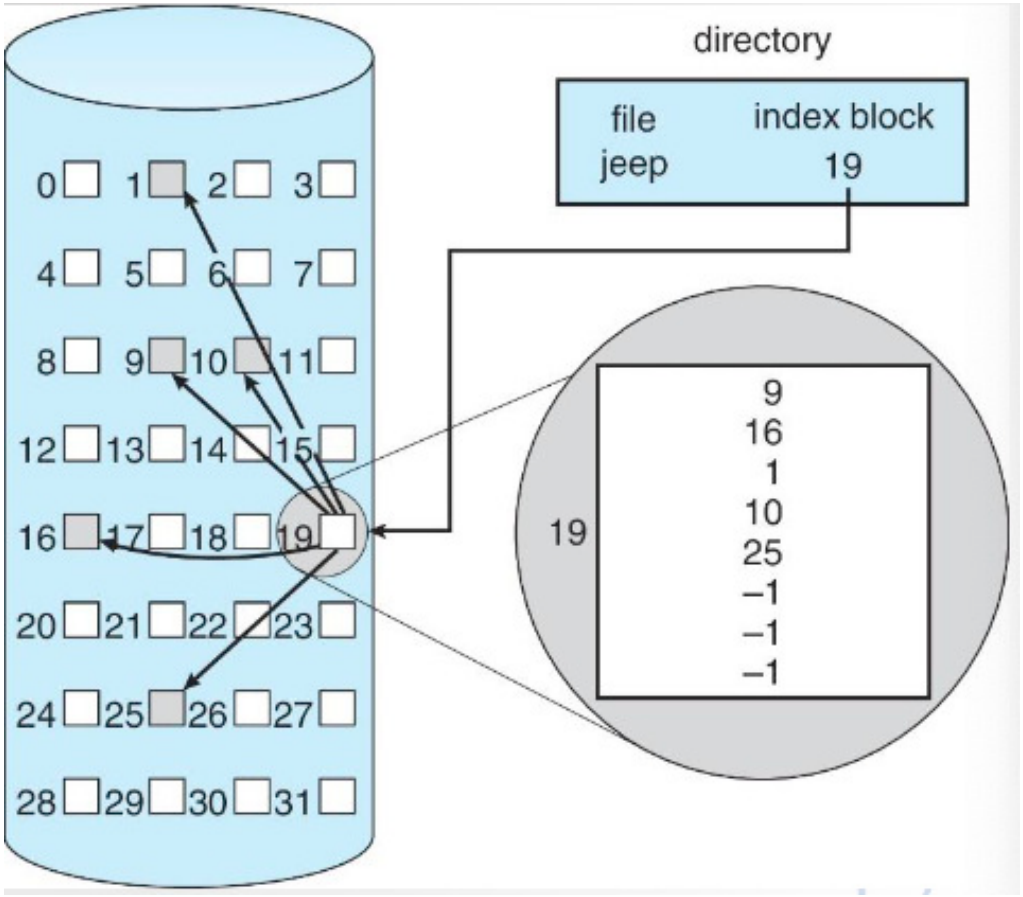
$$MAX\ TAM_{fichero} = 2^{\#} * clustersize$$



Asignación indexada

Los punteros a los bloques del fichero se mantienen en el FCB, facilitando el acceso aleatorio y dinámico.

Inconvenientes: Mucha fragmentación interna si la tabla de índices no se llena. Los bloques esparcidos por todo el disco pueden ralentizar el acceso en HDD. El tamaño de la tabla de índices puede ser insuficiente

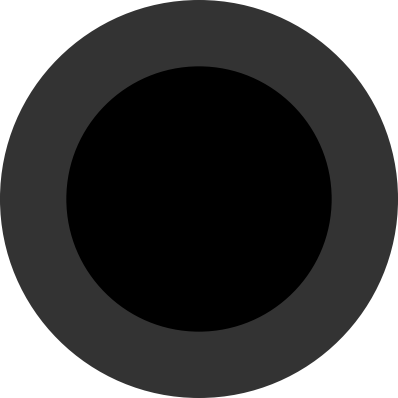


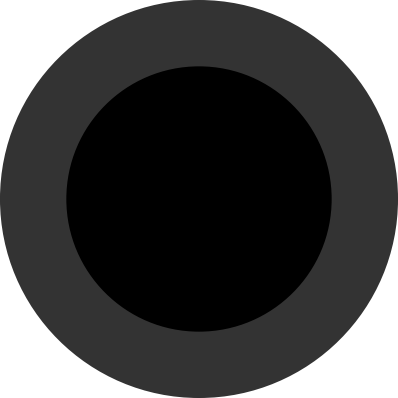
Solución:

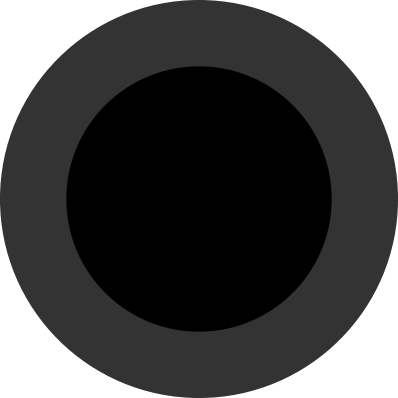
- **Enlazado**: Reservar la última entrada de la tabla de índices para que apunte a otro bloque con la extensión de la tabla de índices
- **Multinivel**: Similar a la tabla de páginas de memoria virtual
- **Combinado**: Una parte de la tabla de indices contiene índices directos, mientras que otras entradas contienen direcciones de otros niveles

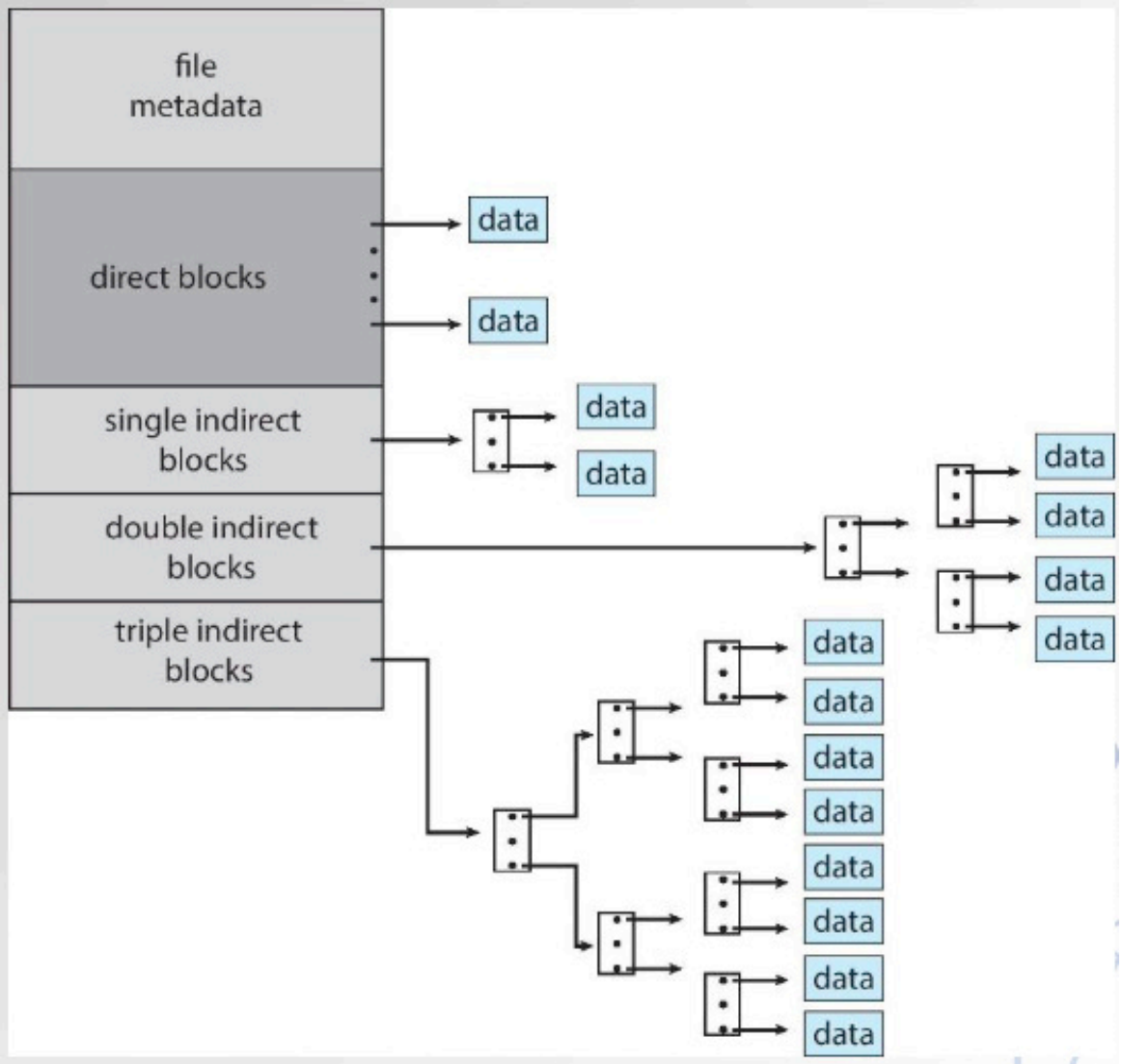
UFS (Asiganción indexada i-node)

Unix implementa un sistema de ficheros con asignación indexada combinada. Es eficiente para ficheros pequeños, permitiendo ficheros de gran tamaño. el FCB se llama i-node y contiene una tabla de índices combinada









Gestión del espacio libre

Vector de bits

El vector de bits se puede mantener en memoria y acceder con palabras. Facilita encontrar espacio libre contiguo para ficheros. Su principal problema es que puede tomar tamaños absurdamente grandes que ocupan espacio tanto en memoria como disco

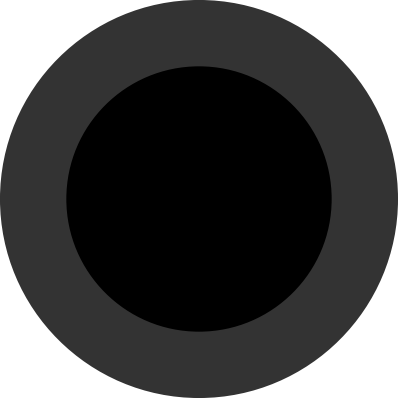
Lista enlazada

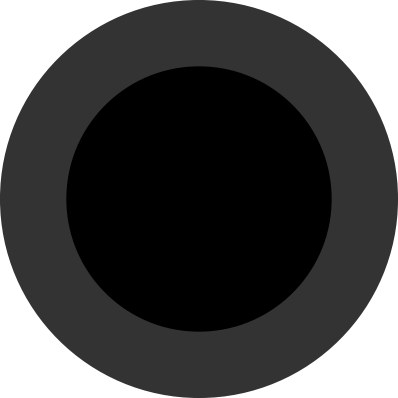
Solo mantiene en memoria el puntero al primer bloque libre. En fat, se mantiene la lista de bloques libres en la estructura File Allocation Table. Su principal inconveniente es la dificultad de acceso (lento)

Tamaño de bloque (cluster)

La elección del tamaño de bloque mínimo que se accederá en cada tracción de disco puede determinar la eficiencia y el rendimiento del sistema de ficheros:

- Tamaño grande: Más fragmentación interna, con menor tiempo de búsqueda y mejor rendimiento
- Tamaño pequeño: Más búsquedas con peor rendimiento, menor fragmentación interna





Otros

- Ficheros mapeados en memoria: Aprovechamos los accesos a disco de la tabla de páginas, por lo que mapeamos archivos dentro del espacio lógico del proceso.
- Buffer de caché unificado: El sistema de ficheros mantiene un buffer en memoria con los últimos bloques de disco accedidos. El principio de localidad dice que serán usados más veces
- FCB: File Control Block. Se encarga del nombrado, manejo de metadatos y protección de los ficheros

Ejercicios

Nombre	Descripción	Fórmula
Clusters FAT #	Número de clusters en FAT #	$TAM_D / N_{ficheros}$
Tamaño de tabla FAT #	Tamaño que ocupa una tabla FAT # en el disco	$N_{clusters} * \# / 8$
Número de ficheros máximo FAT #	El número de ficheros que se pueden almacenar como máximo en una tabla FAT dada	$2^{\#}$
Cilindro	Llamamos cilindro a cada posición que el brazo de la cabeza lectora puede tomar. Notesé que aunque existan varias cabezas lectoras, todas se mueven con el mismo brazo	
Brazo lector	Cada brazo recorre una pista secuencial de un plato. Cada plato puede tener uno o dos brazos lectores	

- Evalua el impacto de la fragmentación interna dados N archivos de tamaño A y clusters de tamaño C: No es más que calcular el espacio desperdiciado (cantidad de bytes sin utilizar en esos clusters)
- Si un tamaño de cluster es más grande que el mínimo, la tabla FAT será más pequeña
- Direccionamiento en UNIX: El sistema de ficheros mantiene direcciones indirectas y directas.
  - Direccionamiento directo: $Tam_{fichero} = N_{directos} * Tam_{cluster}$
  - Direccionamieto indirecto: $Tam_{fichero} = (\frac{Tam_{cluster}}{Tam_{puntero}})^{N_R} * Tam_{cluster}$
- La PTBR es un registro de la MMU que apunta a un marco de la memoria principal en el que se almacena la tabla de páginas

Preguntas de teoría

- Debilidad de FAT: un error en la fat afecta a multitud de datos, es enlazada (lenta) y la tabla ocupa espacio aunque el disco esté vacío
- FAT suele tener dos tablas para proporcionar tolerancia a fallos
- Un puntero negativo en FAT indica final de fichero

- Un cluster está formado por uno o más sectores
- File Control Block: Bloque de datos ubicado en el disco utilizado para indicar información sobre los ficheros
- Sistema de i-nodos: Cada directorio almacena los metadatos de los ficheros y punteros hasta los clusters que estos ocupan. Además, existe el mismo número de i-nodos que de ficheros