

11: Implementación del file system

Sistemas Operativos 1
Ing. Alejandro León Liu

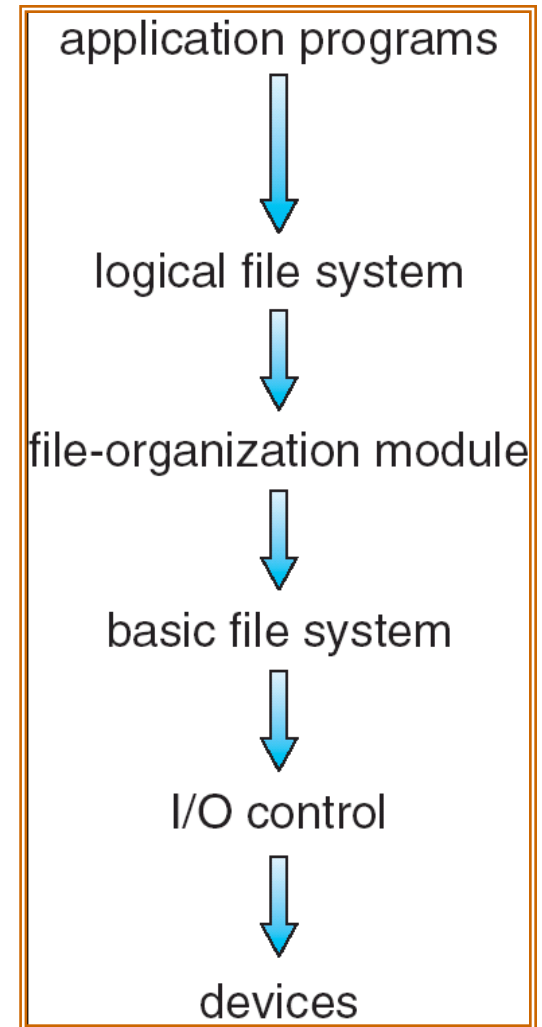


► File system

- Definir un archivo y sus atributos
- Mapear archivos a almacenamiento secundario
 - Algoritmos
 - Estructuras de datos
 - Eficiencia de I/O: agrupar bytes en bloques
 - Cada bloque es formado por uno o varios sectores de disco

IMPLEMENTACIÓN

- ▶ Programación
 - ▶ Logical file system
 - ▶ Metadata: File Control Block
 - Permisos
 - Tiempos (Creación, modificación, acceso)
 - Owner, Grupo, ACL
 - Tamaño
 - Punteros a bloques
 - ▶ File organization
 - ▶ Bloques lógicos
 - ▶ Bloques físicos
 - ▶ Manejo de bloques libres
 - ▶ Basic file system
 - ▶ Simple read y write de bloques
 - ▶ I/O
 - ▶ Dispositivos



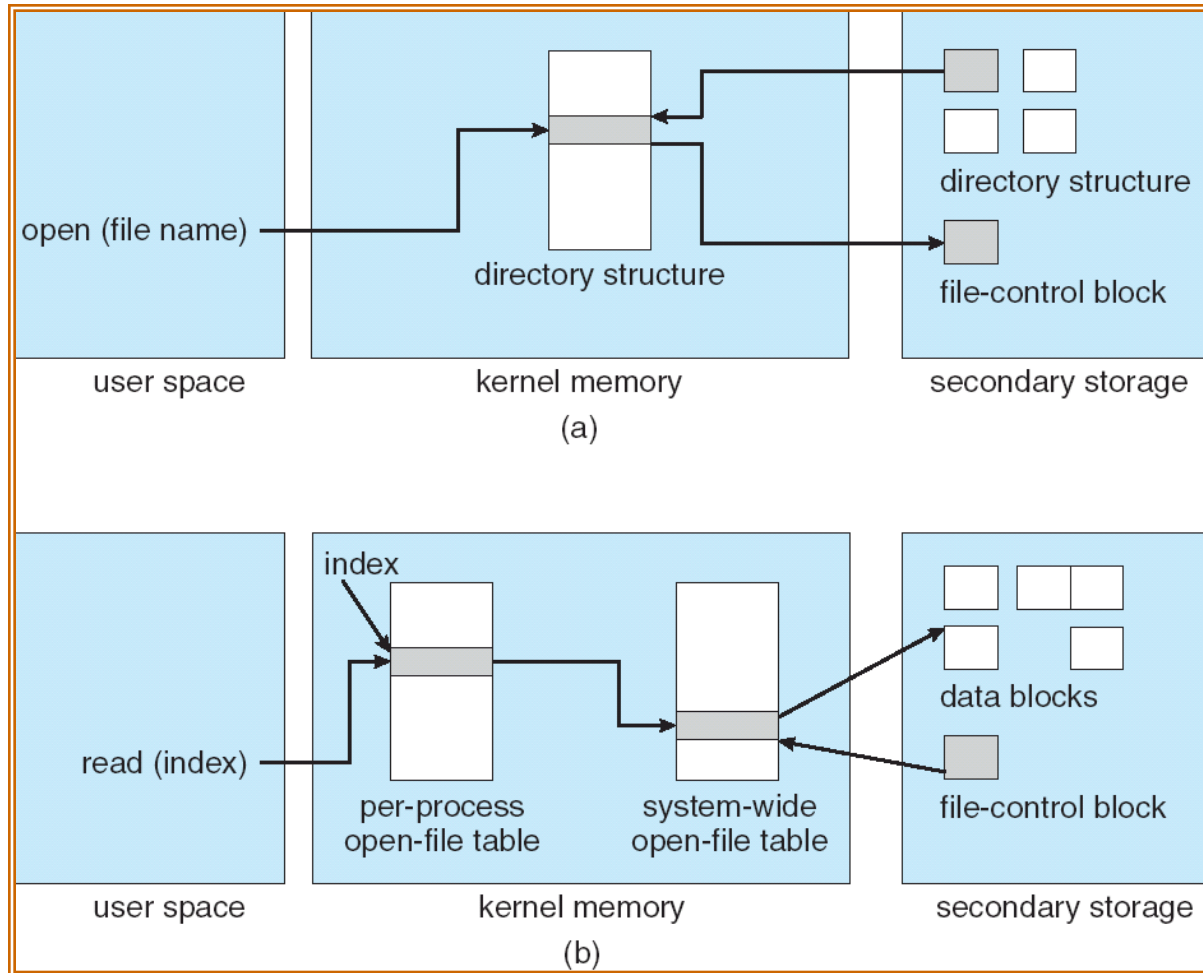
- ▶ Estructuras de datos en almacenamiento secundario
 - ▶ Boot control block
 - ▶ Información para boot un S.O.
 - ▶ Primer bloque
 - ▶ Volume control block (Partición)
 - ▶ Tamaño
 - ▶ Número de bloques
 - ▶ Bloques libres
 - ▶ FCB libres
 - ▶ Master file table
 - ▶ Directorio
 - ▶ FCB

-
- ▶ NTFS (Windows)
 - ▶ Partition boot sector
 - ▶ Master file table
 - Directorio
 - FCB
 - ▶ UFS (Unix File System)
 - ▶ Boot block
 - ▶ Superblock
 - ▶ Directorio
 - Tabla de inodes
 - ▶ Inode: FCB



► Estructuras en memoria

- Mount table: Información de las particiones montadas
- Directorio
 - Información reciente
- Open-file table
 - Contiene copias de los FCB
- Per process open-file table
 - Puntero a registro en open-file table
 - NTFS: File Handle
 - UFS: File descriptor



▶ Particiones

▶ File system

▶ Raw

▶ Sin formato

▶ Manejo especializado de disco

- ☐ Swap

- ☐ Bases de datos

▶ Boot space

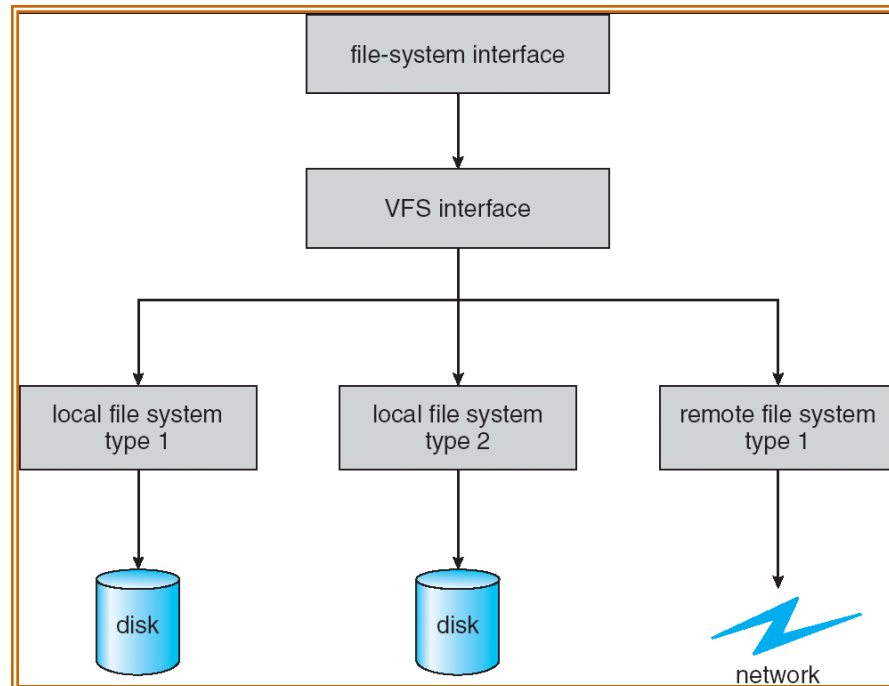
- ☐ Cargado como una imagen en memoria

- ☐ Instrucciones para cargar un S.O.

- ☐ Boot loader

▶ Virtual File System

- ▶ Misma interfaz (API) para el S.O. independiente del file system
- ▶ Permite que un S.O. interprete varios file systems
 - ▶ Remotods



DIRECTORIO

- ▶ Estructura de datos para almacenar los archivos (y carpetas) dentro de una carpeta
 - ▶ Nodos: Punteros a bloques
 - ▶ Criterios:
 - ▶ Tiempo para crear un archivo
 - ▶ Tiempo para borrar un archivo
 - ▶ Tiempo para buscar un archivo

▶ Arreglo

- ▶ Ineficiente
- ▶ Desperdicia espacio

▶ Lista lineal

- ▶ New file: búsqueda lineal de nombre repetido
- ▶ Delete file: búsqueda de archivo a borrar
- ▶ Search file: búsqueda lineal
 - ▶ Caché: archivos recientemente utilizados
 - ▶ Sorted list. Binary search
- ▶ Todo el espacio (todos los FCB) puede ser reutilizado

▶ Hash table

- ▶ Función (fileName) → lista de registros
 - ▶ Ejemplo: mod 64
 - ▶ Manejar colisiones
 - ▶ Desperdicia espacio
 - ▶ Búsqueda exacta es más rápida
-



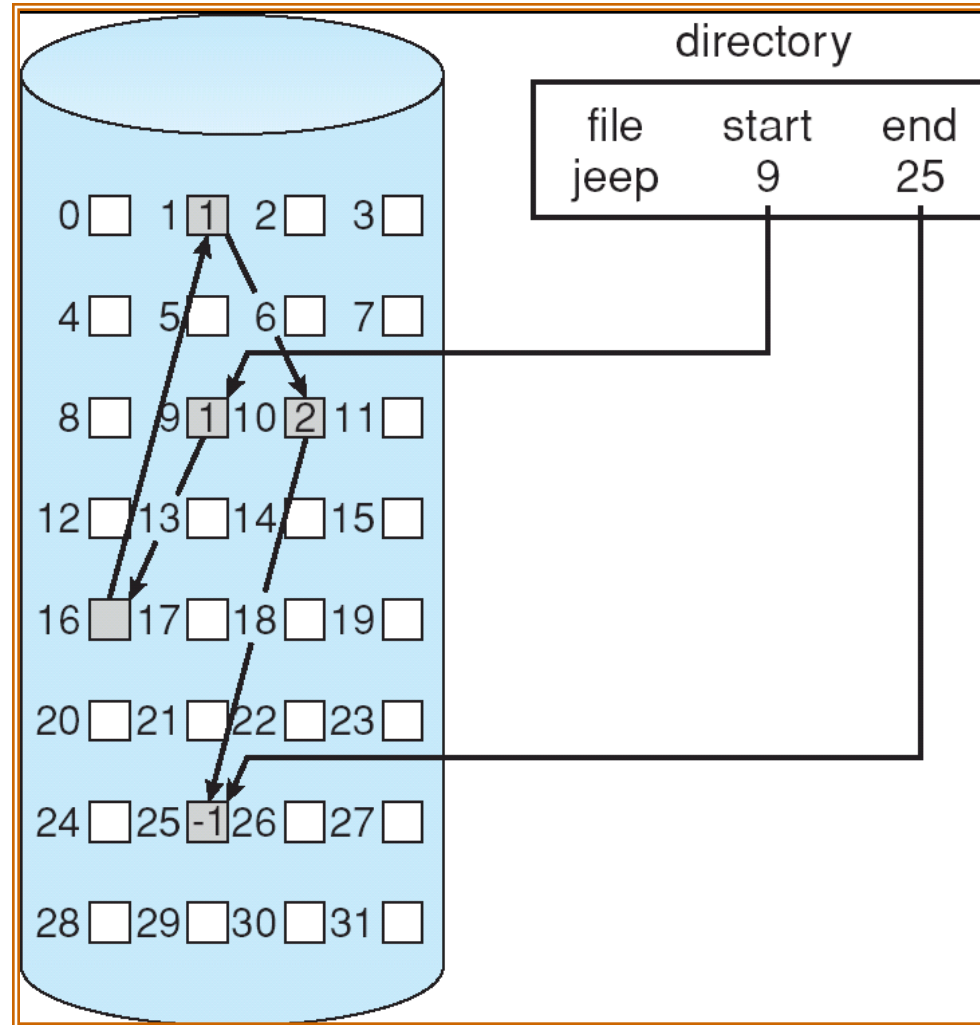
ASIGNACIÓN DE BLOQUES

- ▶ Asignación de bloques a un archivo
- ▶ Criterio de elección: Eficiencia

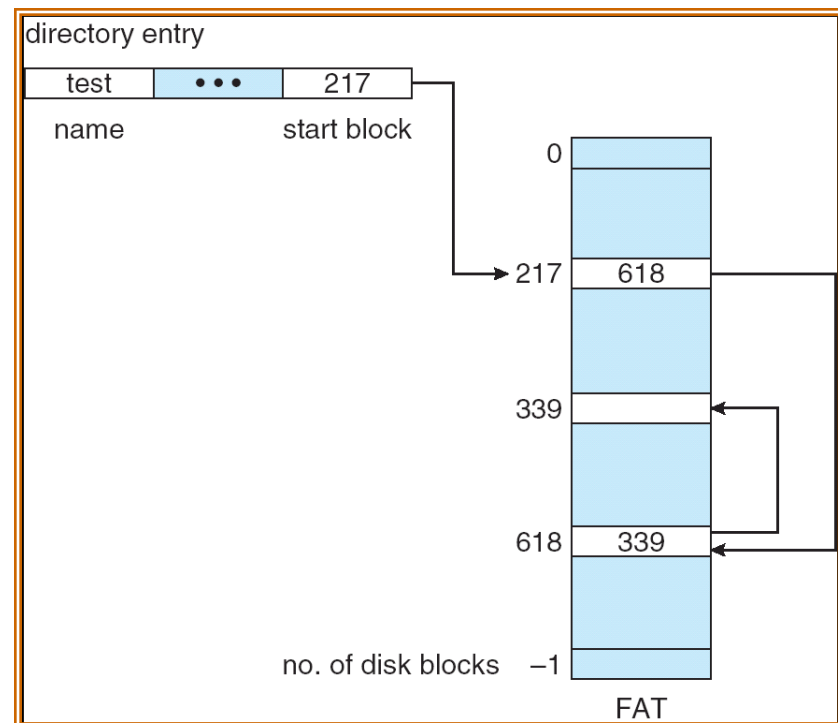
- ▶ Asignación continua
 - ▶ Lectura y escritura rápida
 - ▶ Seek time mínima
 - ▶ Difícil encontrar espacio para un nuevo archivo
 - ▶ Fragmentación externa
 - ▶ Solución requiere downtime
 - ☐ Compactar espacio libre en uno solo
 - ☐ Mover bloques de archivos
 - ▶ Difícil determinar espacio para un archivo
 - ▶ Espacio pequeño: compactar
 - ▶ Espacio grande: desperdicio

▶ Linked allocation

- ▶ Cada archivo es una lista encadenada de bloques
- ▶ FCB contiene puntero a bloque inicial
- ▶ Cada bloque contiene un puntero al siguiente bloque
- ▶ No existe fragmentación externa
- ▶ Únicamente puede ser utilizado para acceso secuencial
- ▶ Clusters
 - ▶ Asignar por clusters (varios bloques a la vez)
 - ▶ Aumenta fragmentación interna
 - ▶ Mejora performance
- ▶ Si se pierde un bloque, se pierde el resto de la información
 - ▶ Lista doblemente encadenada
 - ▶ Puntero a FCB en cada bloque

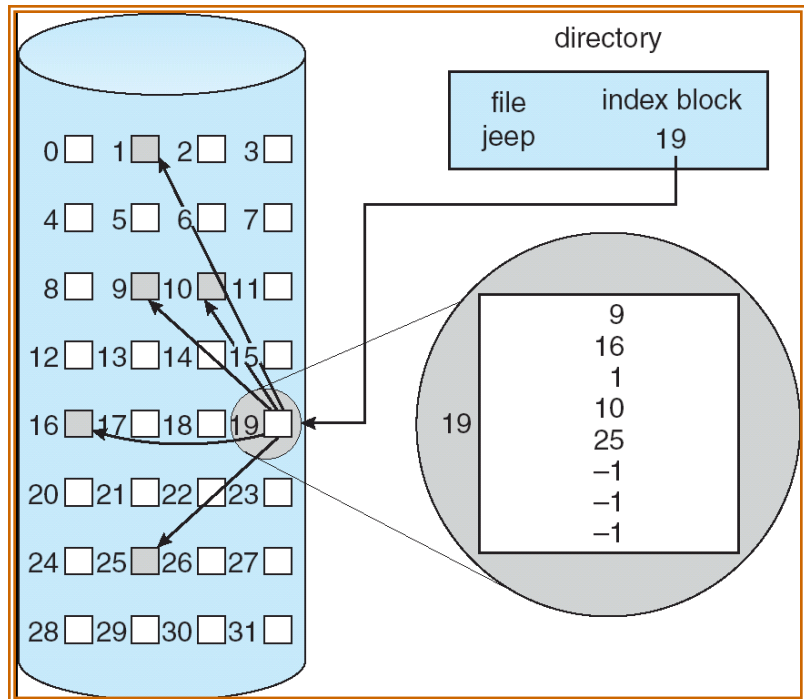


- ▶ File allocation table (FAT)
 - ▶ MSDOS, OS/2
 - ▶ Tabla al inicio de la partición
 - ▶ Contiene todos los punteros de la lista encadenada de bloques
 - ▶ Maneja espacio libre



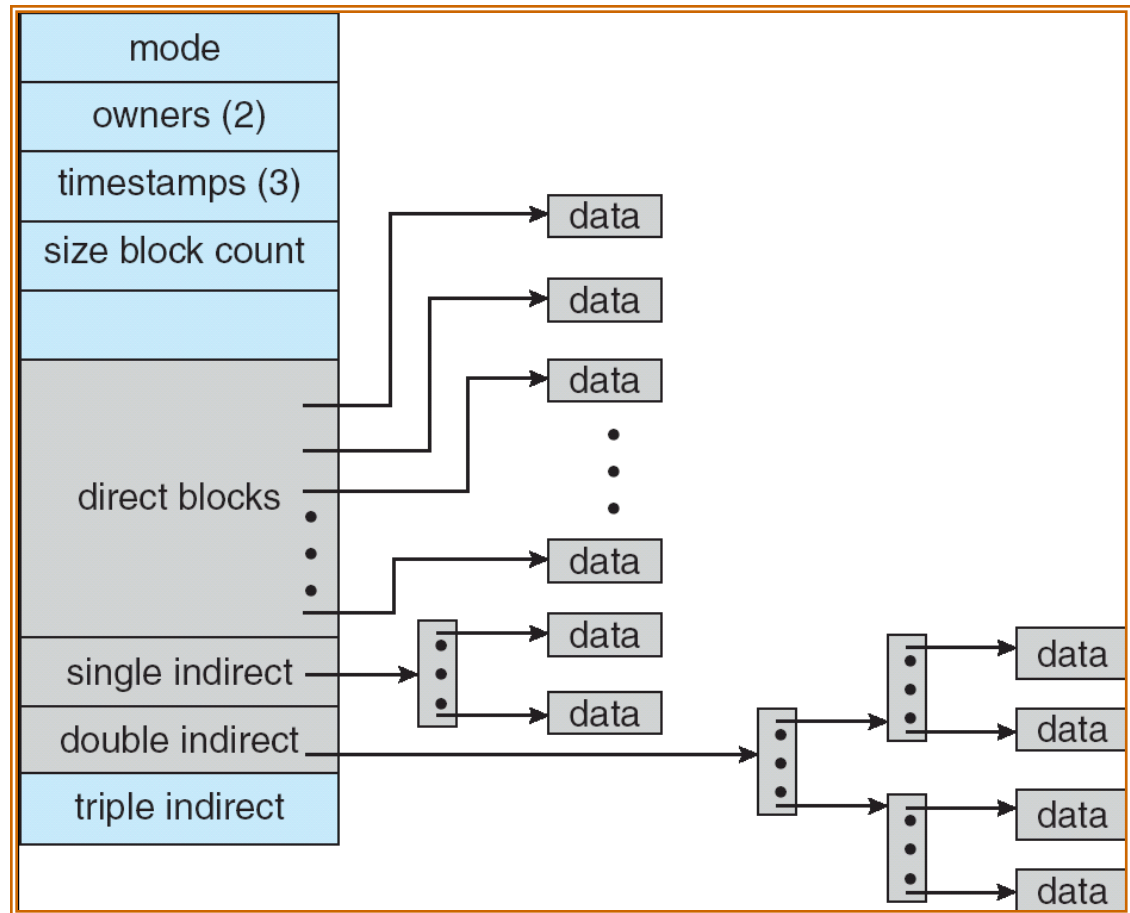
► Indexed allocation

- No hay fragmentación externa
- No es necesario declarar tamaño del archivo
- Utilizar un bloque como index block
 - Apunta a bloques de datos



► Tamaño del index block

- Lista de index blocks
- Multilevel index
- Combined (UNIX)

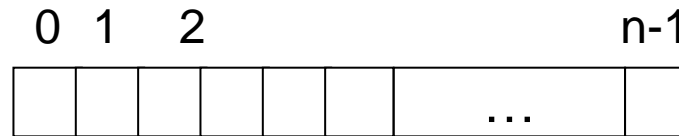


- ▶ Mejor forma de asignación de bloques?
 - ▶ Depende del tipo de acceso
 - ▶ Acceso secuencial
 - Asignación continua
 - ▶ Acceso aleatorio
 - Linked allocation
- ▶ Vale la pena eficientizar la asignación de bloques
 - ▶ 1 lectura de disco = miles de instrucciones

MANEJO DE BLOQUES LIBRES

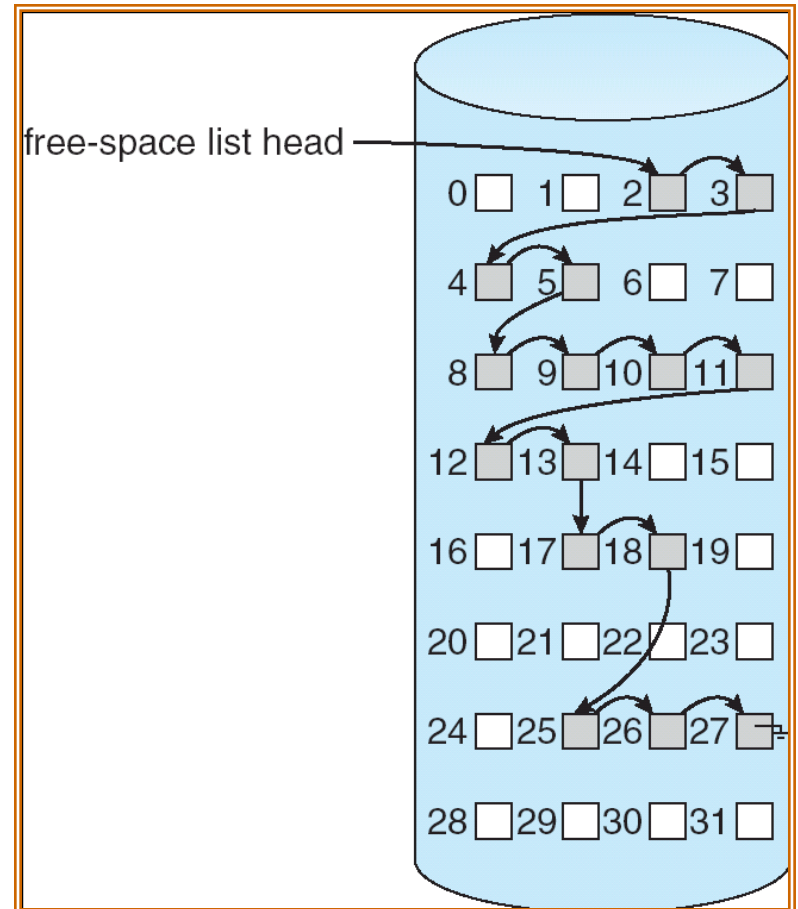
► Bit vector

- Por performance, debe estar en memoria
 - Leer bit por bit es ineficiente
- Espacio adicional



$$\text{bit}[i] = \begin{cases} 0 \Rightarrow \text{block}[i] \text{ free} \\ 1 \Rightarrow \text{block}[i] \text{ occupied} \end{cases}$$

- ▶ Linked list
 - ▶ No requiere espacio adicional
 - ▶ Algoritmo
 - ▶ Leer puntero head
 - ▶ Leer bloque libre
 - ▶ Actualizar head
- ▶ FAT
 - ▶ Maneja bloques libres



EFICIENCIA Y PERFORMANCE

- ▶ Discos: Dispositivo más lento
- ▶ Repartir inodes en todo el disco
 - ▶ Minimizar distancia entre inode y bloques
 - ▶ Minimizar seek time
- ▶ Clustering (varios bloques)
- ▶ Metadatos
 - ▶ Tiempo de último acceso
 - ▶ Una escritura adicional cada vez que se escribe sobre el archivo.
- ▶ Tamaño de punteros
 - ▶ Define el tamaño de los archivos
 - ▶ Espacio adicional

- ▶ Unified virtual memory
 - ▶ Mapear archivos a memoria virtual
 - ▶ Page cache
- ▶ Escrituras
 - ▶ Asíncronas
 - ▶ Se escriben sobre page cache
 - ▶ Escrituras calendarizadas por disco
 - ▶ Síncronas
 - ▶ No usa page cache
 - ▶ Se escriben inmediatamente
 - ▶ Respetar orden de peticiones
 - ▶ Base de datos

RECUPERACIÓN

- ▶ **Revisión de consistencia**
 - ▶ Crash → File system en estado inconsistente
 - ▶ Consistency checker
 - ▶ Revisar file system y reparar
 - ▶ Punteros muertos
- ▶ **Backup**
 - ▶ Backup incremental
 - ▶ Utilizar la última fecha de escritura de archivos
 - ▶ Día 1: copia completa
 - ▶ Día 2: copia de archivos modificados desde el Día 1

► Log

- Almacenar cambios en metadata
- Puntero apunta a las operaciones ya completadas
- Operaciones después del puntero no han sido completadas
- Recuperación