

## ISO, TEORÍA 5: FILESYSTEM

### Que es un archivo?

- Entidad abstracta con nombre.
- Espacio lógico continuo y que es direccionable.
- Provee a los programas con datos (entrada).
- Permite a los programas guardar datos (salida).
- El programa mismo es información que debe ser guardada.

### Punto de vista del usuario sobre un archivo

- Que operaciones puedo llevar a cabo sobre el archivo.
- Como debo nombrarlo.
- Como asegurar la protección del mismo.
- Como compartirlo.
- El usuario no trata con aspectos físicos del HW.

### Punto de vista sobre el diseño

- Como implementamos la construcción lógica para crear un archivo.
- Como implementamos directorios para contener y organizar archivos.
- Manejo del espacio en disco.
- Manejo del espacio libre.
- Eficiencia y mantenimiento.

### Que es un sistema de archivos

- Conjunto de unidades de software (que requieren CPU) que proveen los servicios necesarios para la utilización de archivos. La idea es que cada operación sea eficiente ya que la CPU no puede ser malgastada.
  - Crear.
  - Borrar.
  - Buscar.
  - Copiar.
  - Leer.
  - Escribir.
  - Etc.
- Permite la facilitación el acceso a los archivos por parte de las aplicaciones.
- Permite la abstracción al programador sobre el bajo nivel (el programador no desarrolla el software de administración de archivos).

### Objetivos del SO en cuanto a archivos

- Cumplir con la gestión de datos.
- Cumplir las solicitudes del usuario.

## ISO, TEORÍA 5: FILESYSTEM

- Minimizar o eliminar la posibilidad de perder o destruir datos de forma que sea garantizada la integridad del contenido de los archivos.
- Dar soporte de E/S a distintos dispositivos.
- Brindar un conjunto de interfaces de E/S para tratar archivos.

### **Tipos de archivos**

- Archivos de tipo regular
  - Texto plano
    - Source file
  - Binarios
    - Object file
    - Executable file
- Directorios (los directorios son archivos)
  - Su contenido son relaciones con otros archivos y mantienen la estructura en el Filesystem

### **Atributos de un archivo**

- Nombre.
- Identificador unívoco.
- Tipo.
- Localización, toda la información necesaria para ubicar el contenido del archivo en disco.
- Tamaño.
- Protección, seguridad y monitoreo
  - Dueño, permisos, contraseña.
  - Momento en que el usuario lo modificó, creó, accedió por ultima vez (FECHAS)
  - ACLs → Access control list.

### **Directorios**

- Archivos que contienen información acerca de archivos y directorios que están dentro de él.
- El directorio es en si mismo un archivo.
- Interviene en la resolución entre el nombre y el archivo mismo, ósea que proporciona la ruta concreta para el archivo específico (el cuál puede estar en otros directorios).
- Operaciones en directorios:
  - Buscar un archivo.
  - Crear un archivo.
  - Borrar un archivo.

## ISO, TEORÍA 5: FILESYSTEM

- Listar el contenido.
- Renombrar.
- El uso de los directorios ayuda con:
  - Eficiencia -> ubicar rápidamente un archivo.
  - Uso del mismo nombre de archivo:
    - Diferentes usuarios pueden tener el mismo nombre de archivo.
  - Agrupar: se agrupa un conjunto de archivos de forma lógica en base a propiedades o funciones:
    - Ejemplo: programas java, juegos, librerías.

### Estructura de directorios

- Los archivos pueden ubicarse siguiendo un path desde el directorio raíz y sus sucesivas referencias (path absoluto).
- Distintos archivos pueden tener el mismo nombre pero el full path name es único.
- El directorio actual del proceso se lo llama working directory.
- Dentro del directorio de trabajo se pueden referenciar archivos tanto por su PATH absoluto como por su PATH relativo indicando solamente la ruta al archivo DESDE el working directory.
  - Absoluto: el path incluye todo el camino del archivo.
  - Relativo (al working directory actual): el nombre se calcula relativamente al directorio en el que se esté.

### Compartir archivos

- En ambientes multiusuario es necesario que varios usuarios puedan compartir archivos.
- La compartición se debe realizar bajo un esquema de protección:
  - Derechos de acceso:
    - Los directorios también tienen permisos, los cuales pueden permitir el acceso al mismo para que el usuario pueda usar el archivo siempre y cuando tenga los permisos necesarios.
    - Permiso de ejecución → el usuario puede ejecutar.
    - Permiso de lectura → el usuario puede leer el archivo.
    - Permiso de agregar → el usuario puede agregar datos al archivo (no modificar).
    - Permiso de actualización → el usuario puede modificar, borrar y agregar datos. Incluye crear archivos, sobrescribirlos, remover datos de ellos.

## ISO, TEORÍA 5: FILESYSTEM

- Permiso de cambio de protección → el usuario puede modificar los derechos de acceso al archivo.
- Permiso de borrado → el usuario puede borrar el archivo.
- **Owners:**
  - Posee todos los derechos sobre el archivo.
  - Puede dar derechos a otros usuarios. Se determinan clases:
    - Derechos para un usuario específico.
    - Derechos para grupos de usuarios.
    - Derechos para todos (archivos públicos).
- Manejo de accesos simultáneos.
- El esquema de protección:
  - El propietario/administrador del archivo debe ser capaz de controlar que se puede hacer (derechos de acceso) y quien puede hacer.

## ISO Tema 5, FILESYSTEM

### Metas del Sistema de archivos

- Brindar espacio en disco a los archivos del usuario y del sistema.
- Mantener un registro del espacio libre. Cantidad de espacio libre y donde se ubica dentro del disco.

### Conceptos

- Sector
  - Unidad de almacenamiento utilizada en los discos rígidos.
- Bloque o Cluster
  - Conjuntos de sectores consecutivos.
  - Nos sirve para archivos que necesitan más de un sector.
- File system
  - Define la forma en que los datos se almacenan.
- FAT: File Allocation Table
  - Tabla que contiene información sobre en que lugar están alocados los distintos archivos.
- Se puede provocar fragmentación interna en un disco.

### Pre-asignación: forma de asignar espacio

- Es necesario saber cuanto espacio va a ocupar el archivo en el momento de su creación.
- Se tiende a definir espacios mucho más grandes que lo necesario.
- Es posible usar sectores contiguos para almacenar los datos de un archivo.
- Y si el archivo supera el espacio que le asignaron?
- Asignación continua aplica la pre-asignación.

### Asignación dinámica: forma de asignar espacio

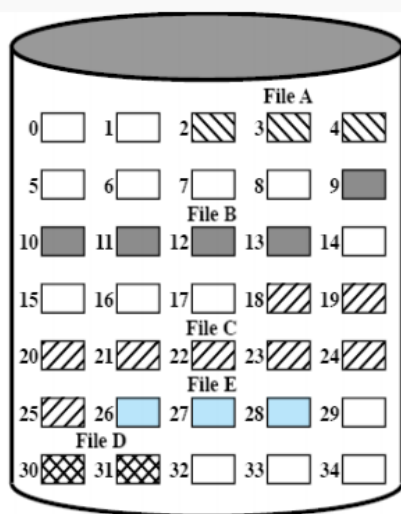
- El espacio es solicitado a medida que se necesita.
  - Los bloques de datos pueden quedar de forma no contigua.
  - La tendencia de hoy día es para asignaciones dinámicas.
- 
-

## ISO Tema 5, FILESYSTEM

### Veremos formas de asignación

#### Forma continua

- Para guardar un archivo se utiliza un conjunto contiguo de bloques.
- Es necesario preasignar, ósea conocer el tamaño del archivo al ser creado.
- El formato de la FAT es simple, ya que utiliza una sola entrada que incluye el bloque en el cuál empieza el archivo y la longitud (bloques usados).
- El archivo puede ser leído con una única operación.
- Puede existir fragmentación externa que se puede resolver compactando aunque esto es MUY costoso.
- La técnica presenta problemas:
  - Encontrar bloques libres continuos en el disco (y que tengan el tamaño necesario).
  - El incremento del tamaño de un archivo.



FAT  
File Allocation Table

File Name	Start Block	Length
File A	2	3
File B	9	5
File C	18	8
File D	30	2
File E	26	3

Que sucedería si necesitamos agregar un nuevo archivo de 6 bloques?

RTA: No podría agregarlo a menos que realice una desfragmentación.

#### Forma encadenada

- Se asigna en base a bloques individuales.
- Cada bloque tiene un puntero a la ubicación del próximo bloque del archivo → se debe implementar una estructura de enlaces.
- FAT tiene una única entrada, bloque inicial y tamaño del archivo.
- No hay fragmentación externa al usar esta forma.
- Útil para acceso secuencial.
- Los archivos pueden crecer bajo demanda (asignación dinámica).
- No se requieren bloques contiguos.

ISO Tema 5, FILESYSTEM

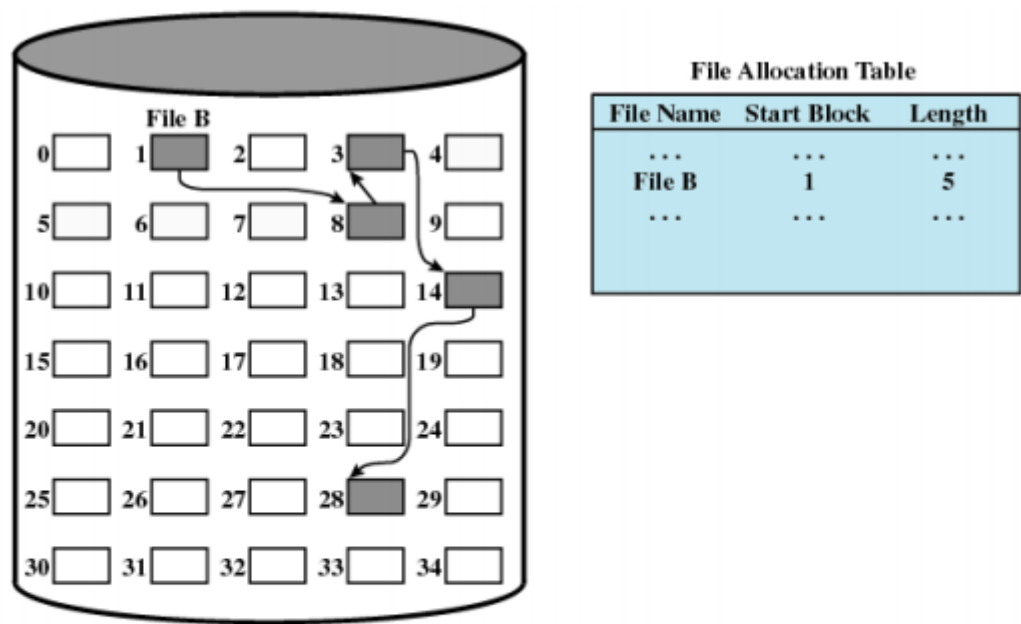
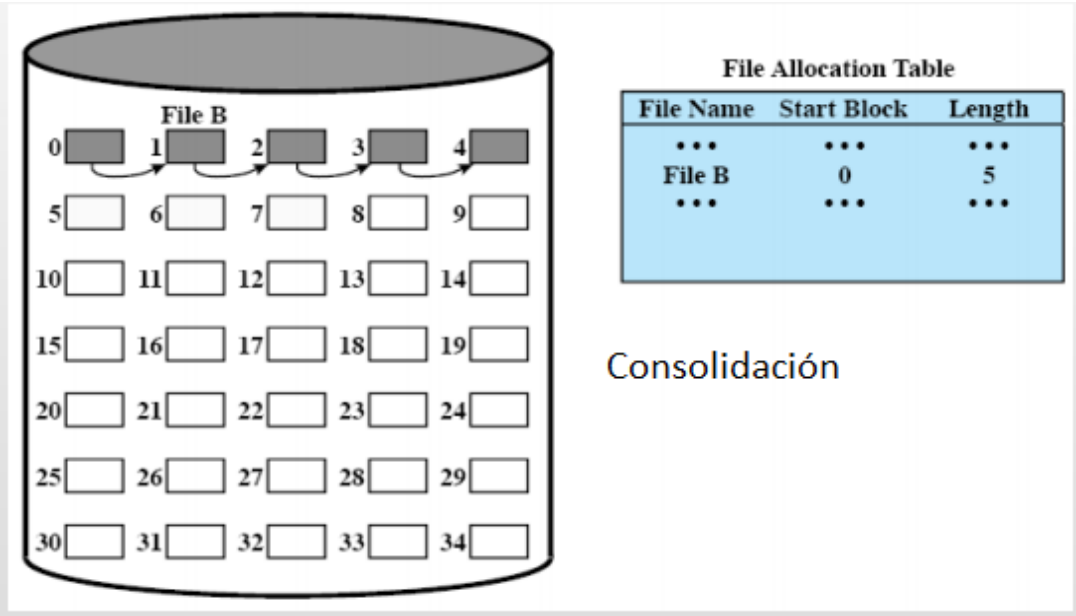


Figure 12.9 Chained Allocation

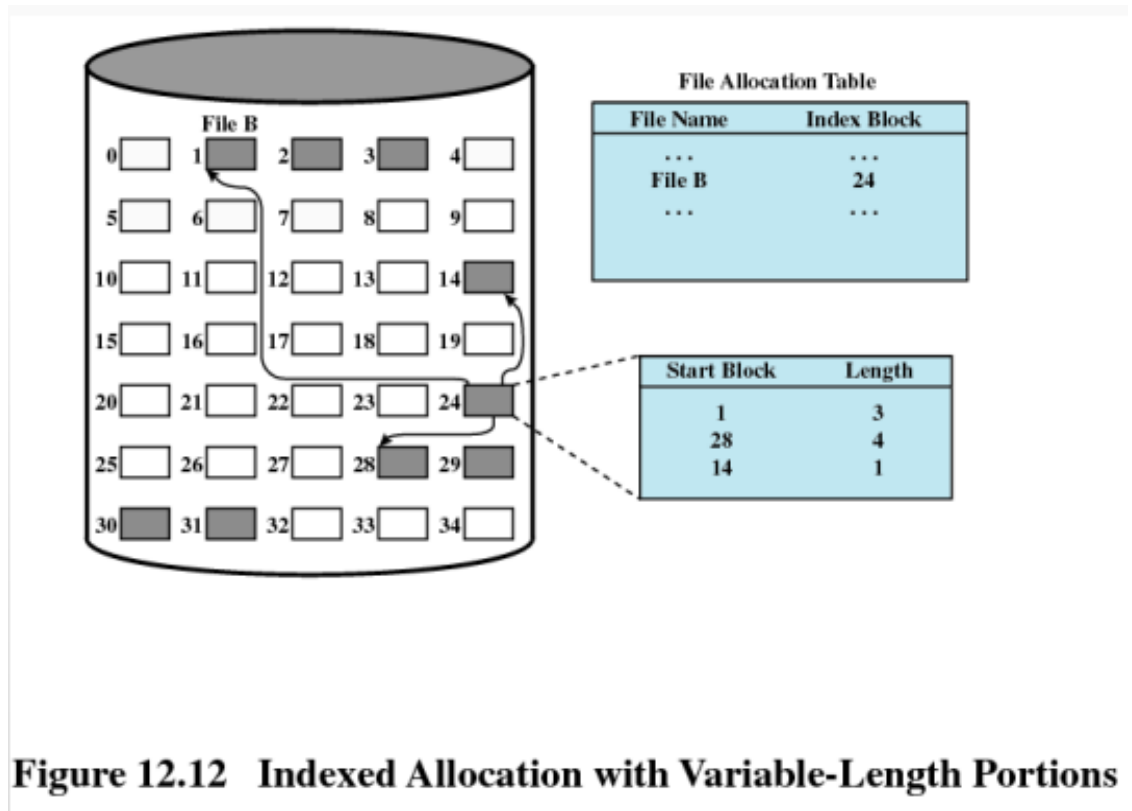
- 
- Es posible mover los bloques de un mismo archivo de forma que queden contiguamente, esto ayuda a que haya una cercanía de bloques para agilizar operaciones a la hora de leer el disco (desfragmentadores de discos).



## ISO Tema 5, FILESYSTEM

### Forma indexada:

- Se asigna en base a bloques individuales.
- No se produce fragmentación externa.
- El acceso random a un archivo se vuelve eficiente.
- FAT
  - Una sola entrada con la dirección del bloque de índices (puntero).
  - El bloque índice NO contiene datos del archivo, sino que contiene una lista de punteros a los bloques que componen el archivo. (índice)
- Tiene dos variantes
- **ASIGNACIÓN POR SECCIONES**
  - A cada entrada del bloque índice se agrega el campo longitud.
  - Cada índice del bloque índice apunta al primer bloque de un conjunto que se almacena sí o sí de manera contigua.



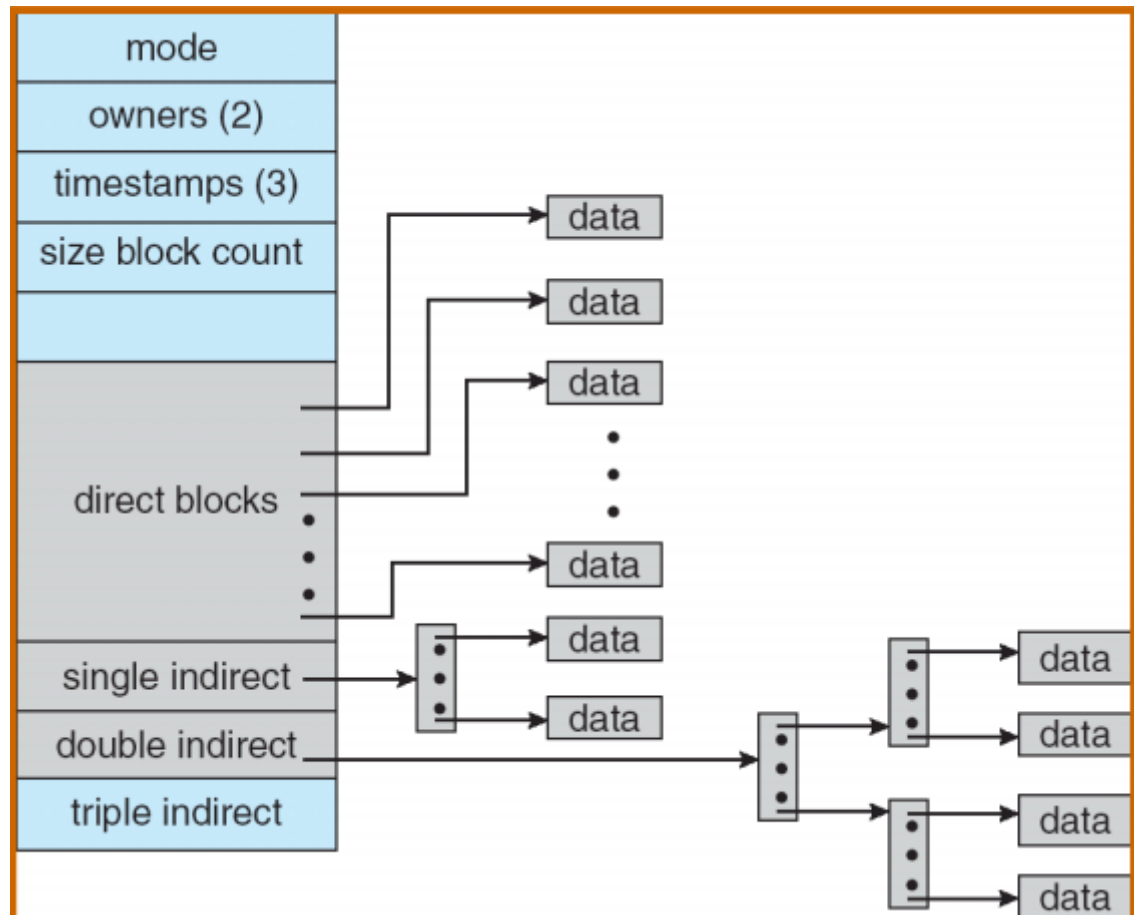
**Figure 12.12 Indexed Allocation with Variable-Length Portions**

- 
- **NIVELES DE INDIRECCIÓN**
  - Existen bloques directos con datos guardados dentro.
  - Otros bloques son considerados como bloque índice (apuntan a varios bloques de datos).
    - Direccionamiento indirecto simple: un bloque índice solo que apunta a los bloques de datos.



## ISO Tema 5, FILESYSTEM

- Direccionamiento indirecto doble: un bloque índice que apunta a otro bloque índice y este último apunta a los bloques de datos.
- Pueden haber varios niveles de indirección



- 
- Conocido como INODO

---

### Gestión de espacio libre

- Consiste en el control sobre cuáles de los bloques de disco están disponibles.
- Tres alternativas:
  - Tablas de bits.
  - Bloques libres encadenados.
  - Indexación.

#### Tablas de bits

- Vector con 1 bit por CADA bloque de disco.
  - 0 = bloque libre, 1 = bloque ocupado.
- Ventaja
  - Es fácil hallar un bloque o grupo de bloques que estén libres.

## ISO Tema 5, FILESYSTEM

- Desventaja
  - Tamaño que llega a ocupar el vector en la memoria.
    - Se calcula: tamaño del disco bytes / tamaño bloque en sistema de archivos.
    - Ej: Disco de 16 GB con bloques de 512 bytes cada uno → el vector pesa 32 MB.

### ✓ Ejemplo

00111

00001

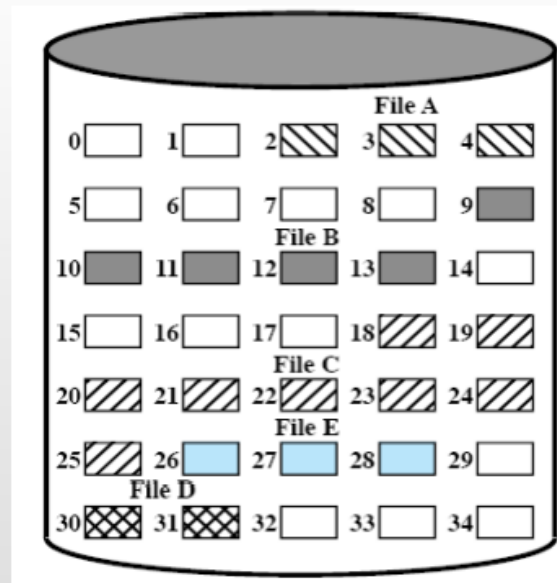
11110

00011

11111

11110

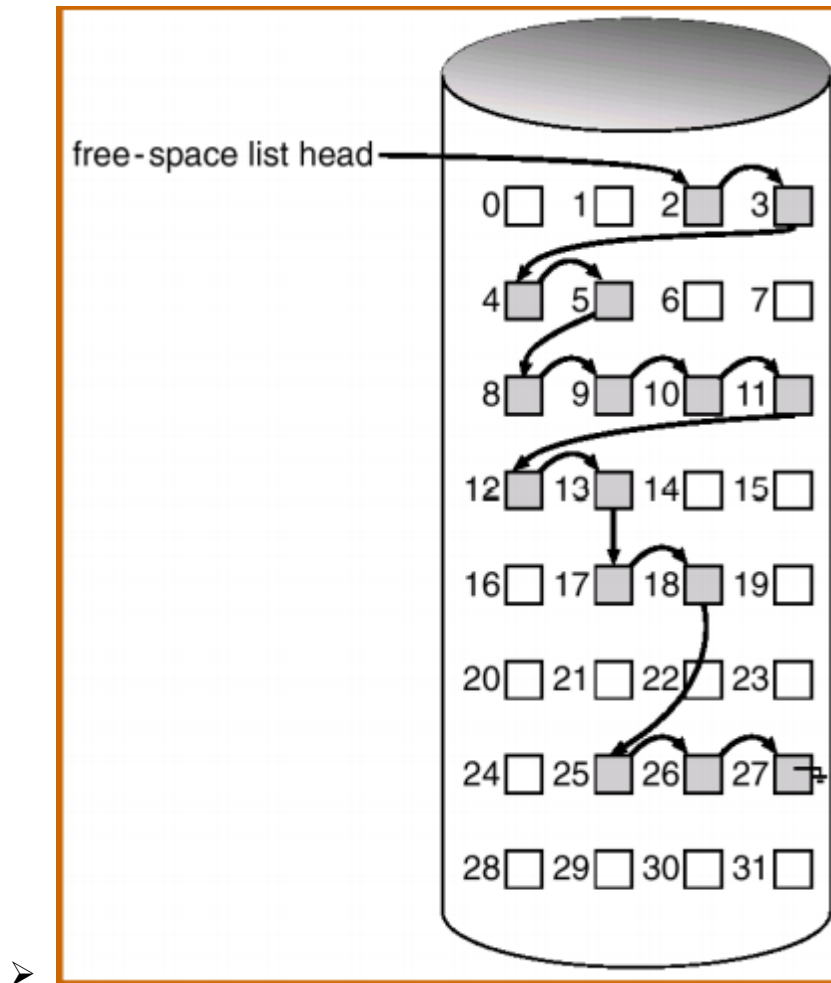
11000



### Bloques encadenados

- Se tiene un puntero al primer bloque que esté libre.
- Cada bloque libre tiene un puntero al siguiente bloque libre.
- Resulta ser ineficiente para la búsqueda de bloques libres ya que hay que realizar varias operaciones de E/S para obtener un grupo libre.
- Si se pierde un enlace se generan problemas.
- Es difícil hallar bloques libres consecutivos.

## ISO Tema 5, FILESYSTEM



### Indexación (agrupamiento)

- Variante de bloques libres encadenados.
- El primer bloque libre contiene las direcciones de N bloques libres.
- Las N-1 primeras direcciones son bloques libres.
- La N-ésima dirección referencia otro bloque con N direcciones de bloques libres.

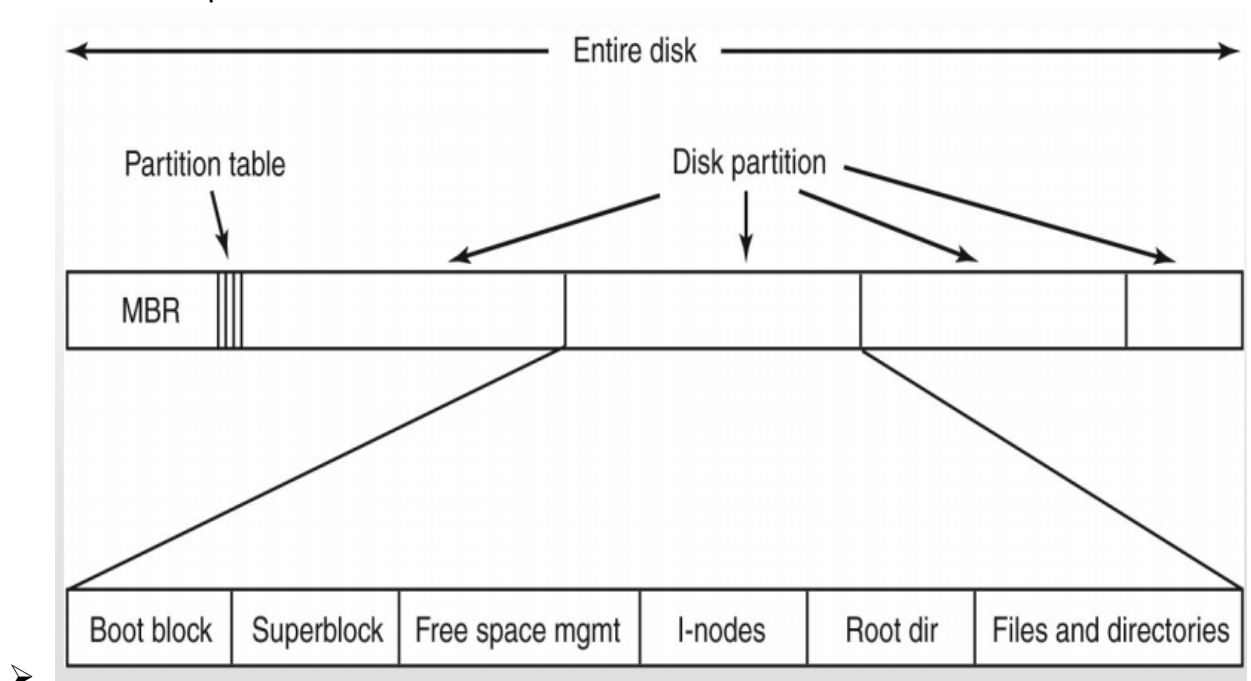
## ISO TEMA 5- FILESYSTEM

### Unix – Manejo de archivos

- Tipos de archivos
  - Archivo común.
  - Directorio.
  - Archivos especiales (dispositivos).
  - Named pipes (comunicación entre procesos).
  - Links (comparten el i-nodo, solo dentro del filesystem).
  - Links simbólicos (como si fueran accesos directos, tiene i-nodo propio, para filesystems diferentes).

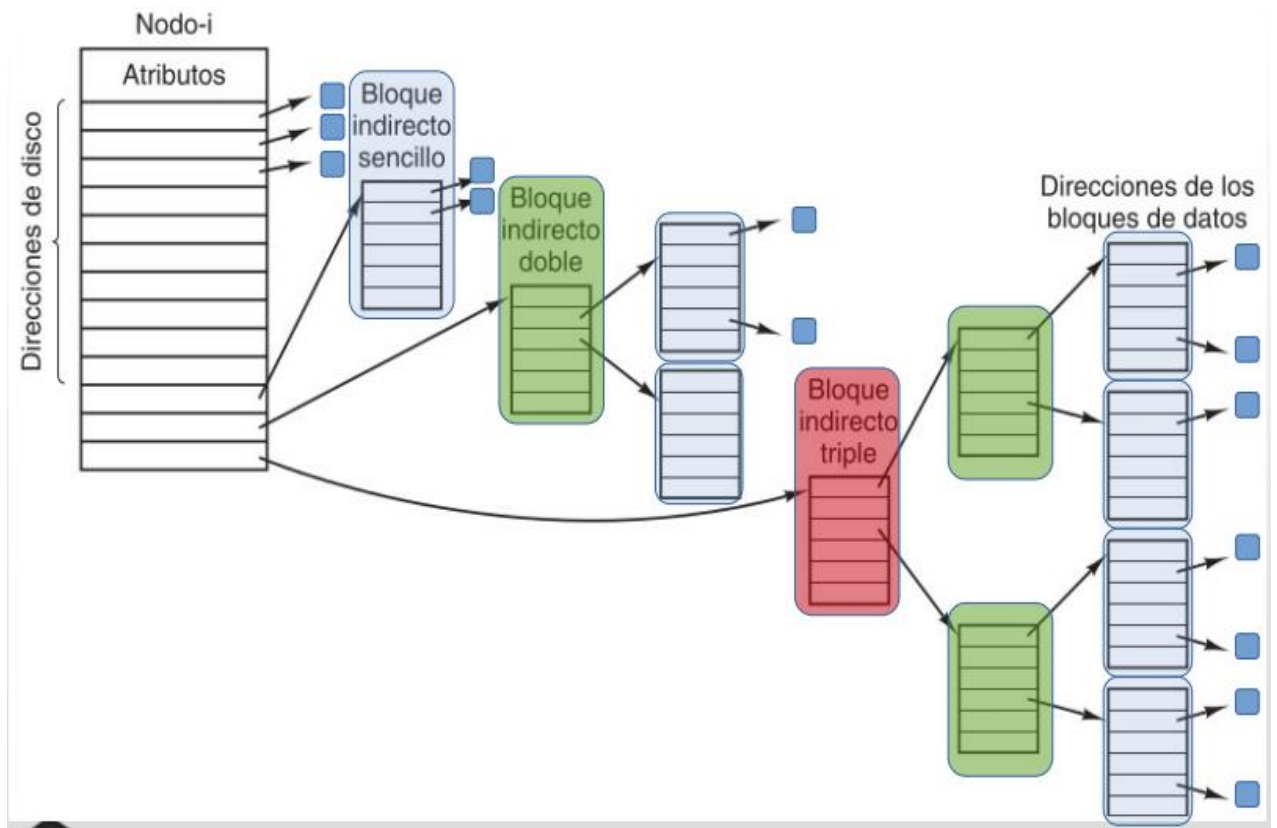
### Unix – Estructura del volumen

- Cada disco físico puede ser dividido en uno o más volúmenes (particiones). Cada volumen contiene un filesystem. Cada filesystem contiene.
  - Boot block: código para bootear el S.O
  - Superblock: atributos sobre el filesystem que incluyen que bloques/clusters están libres.
  - Tabla de Inodos: tabla que contiene todos los inodos.
    - Inodo: estructura de control que contiene la información clave de un archivo.
  - Bloques de datos: de los archivos.

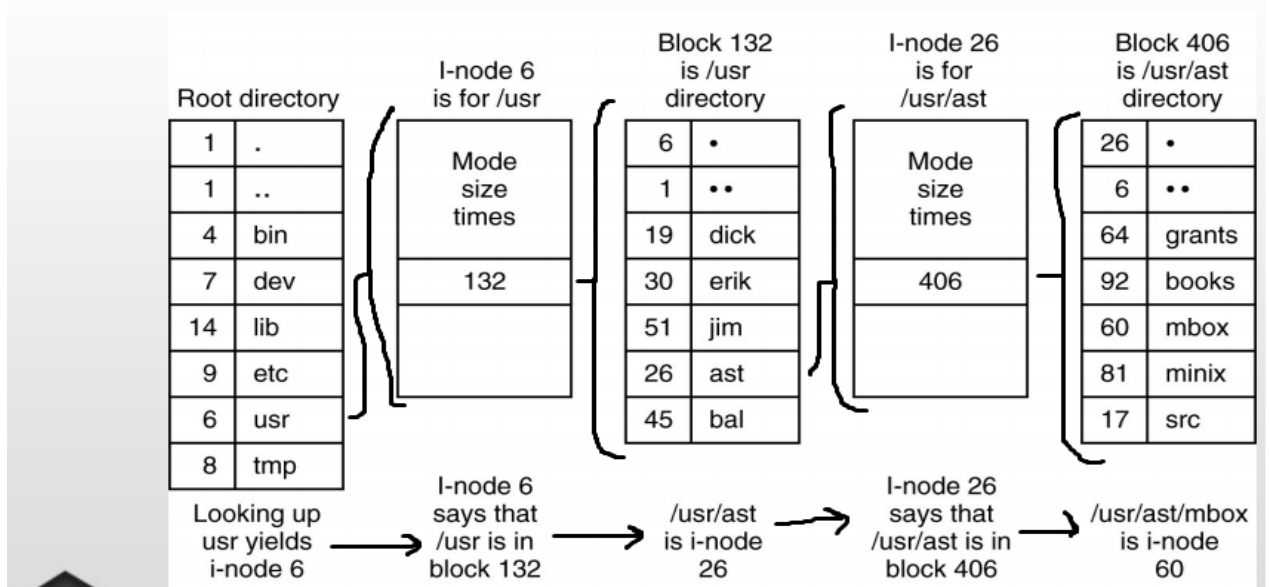


### Unix - INODO (Como se ve y un ejemplo de búsqueda)

## ISO TEMA 5- FILESYSTEM



### Buscar el i-nodo del archivo /usr/ast/mbox



Las tablas que tienen "Mode size times" especifican en que bloque se encuentra la información.

## ISO TEMA 5- FILESYSTEM

### Windows – File Systems soportados

- CD-ROM filesystem (CDFS) → CD
  - Universal Disk Format (UDF) → DVD, Blu-Ray
  - File Allocation Table
    - FAT12
    - FAT16 → nombres cortos de archivos
    - FAT32 → nombres largos de archivos pero no soportados en MS-DOS
  - New Technology File System (NTFS)
- 

### Windows – FAT

- FAT es un sistema de archivos utilizado originalmente por DOS y Windows 9x
- Por qué Windows aun soporta FAT?
  - Proporciona compatibilidad con otro SO en sistemas multiboot.
  - Permite upgrades desde versiones anteriores.
  - Para formato de dispositivos como diskettes.
- Las distintas versiones de FAT son diferenciadas por un número que indica la cantidad de bits que son usados para identificar diferentes bloques o clusters.
  - FAT12
  - FAT16
  - FAT32
- Es utilizado un mapa de bloques del FILESYSTEM, llamado FAT.
- La FAT tiene cantidad de entradas = cantidad de bloques.
- La FAT, su duplicado, y el directorio raíz se almacenan en los primeros sectores de la partición.



- 
- En FAT se utiliza la asignación encadenada.
- El puntero al próximo bloque está en la FAT y no en los bloques.
- Los bloques libres y dañados tienen códigos especiales.

### Windows – FAT12

- FAT12 utilizan 12 bits para la identificación del sector, se limita a  $2^{12}$  (4096) sectores.
  - Windows utiliza tamaños de sector que van desde los 512 bytes hasta los 8kb, esto limita el tamaño total del volumen a 32 MB → 4096 sectores \* 8 KB

## ISO TEMA 5- FILESYSTEM

- FAT12 se usa como filesystem para diskettes de 3,5 y 12 pulgadas que son capaces de almacenar hasta 1,44 MB de datos (probablemente lo que pesen 3 o 4 resúmenes míos de la materia).

### Windows – FAT16

- FAT16 usa 16 bits para identificar cada sector. Puede haber  $2^{16}$  (65.536) sectores en un volumen.
  - El tamaño del sector varía entre 512 bytes hasta los 64 KB lo que limita el tamaño máximo de la partición a 4 GB ( $64 \text{ KB} * 65.536$ )
  - El tamaño de sector dependía del tamaño de la partición al formatearla.

### Windows – FAT32

- FAT32 fue el filesystem más reciente de la línea esa (posteriormente salió exFAT que algunos lo conocen como FAT64).
- FAT32 usa 32 bits para la identificación de sectores, PERO reserva los 4 bits superiores. Efectivamente se usan 28 bits para identificar.
  - El tamaño de sector en FAT32 puede ser de hasta 32KB con lo cual tiene una capacidad teórica de direccionar particiones de hasta 8 TB.
  - El modo de identificación y acceso a los sectores lo hace de forma más eficiente que FAT16. Con tamaño de sector de 512 bytes, puede direccionar volúmenes de hasta 128 gb.

### Windows – NTFS

- NTFS es el filesystem nativo de Windows desde Windows NT.
- Usa 64 bits para referenciar sectores.
  - Teóricamente permite tener volúmenes de hasta 16 Exabytes (16 billones de GB).
- Por qué usar NTFS en lugar de FAT?. A pesar de que FAT es simple y más rápido para ciertas operaciones, NTFS soporta:
  - Tamaños de archivo y de disco mayores.
  - Mejora performance en discos grandes.
  - Nombres de archivos de hasta 255 caracteres.
  - Atributos de seguridad.
  - Es transaccional.