Introducción a los Sistemas Operativos

Administración de Archivos - I

I.S.O.

- ☑ Versión: Noviembre 2017
- ☑ Palabras Claves: Archivo, Directorio, File System,

Algunas diapositivas han sido extraídas de las ofrecidas para docentes desde el libro de Stallings (Sistemas Operativos) y el de Silberschatz (Operating Systems Concepts). También se incluyen diapositivas cedidas por Microsoft S.A.

Porque necesitamos archivos?

- ☑ Almacenar grandes cantidades de datos
- ☑ Tener almacenamiento a largo plazo
- ☑ Permitir a distintos procesos acceder al mismo conjunto de información

Archivo

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

Archivos - Punto de vista del Usuario

- ☑ Que operaciones se pueden llevar a cabo
- ☑ Como nombrar a un archivo
- ☑ Como asegurar la protección
- ☑ Como compartir archivos
- ☑ No tratar con aspectos físicos
- ☑ Etc.

Archivos - Punto de vista del Diseño

- ☑ Implementar archivos
- ☑ Implementar directorios
- ☑ Manejo del espacio en disco
- ☑ Manejo del espacio libre
- ☑ Eficiencia y mantenimiento

Sistema de Manejo de Archivos

- Conjunto de unidades de software que proveen los servicios necesarios para la utilización de archivos
  - Crear
  - Borrar
  - Buscar
  - Copiar
  - Leer
  - Escribir
  - Etc.

Sistema de Manejo de Archivos (cont.)

- ☑ Facilita el acceso a los archivos por parte de las aplicaciones
- ☑ Permite la abstracción al programador, en cuanto al acceso de bajo nivel (el programador no desarrolla el soft de administración de archivos)

Objetivos del SO en cuanto a archivos

- Cumplir con la gestión de datos
- Cumplir con las solicitudes del usuario.
- Minimizar / eliminar la posibilidad de perder o destruir datos
  - Garantizar la integridad del contenido de los archivos
- Dar soporte de E/S a distintos dispositivos
- Brindar un conjunto de interfaces de E/S para tratamiento de archivos.

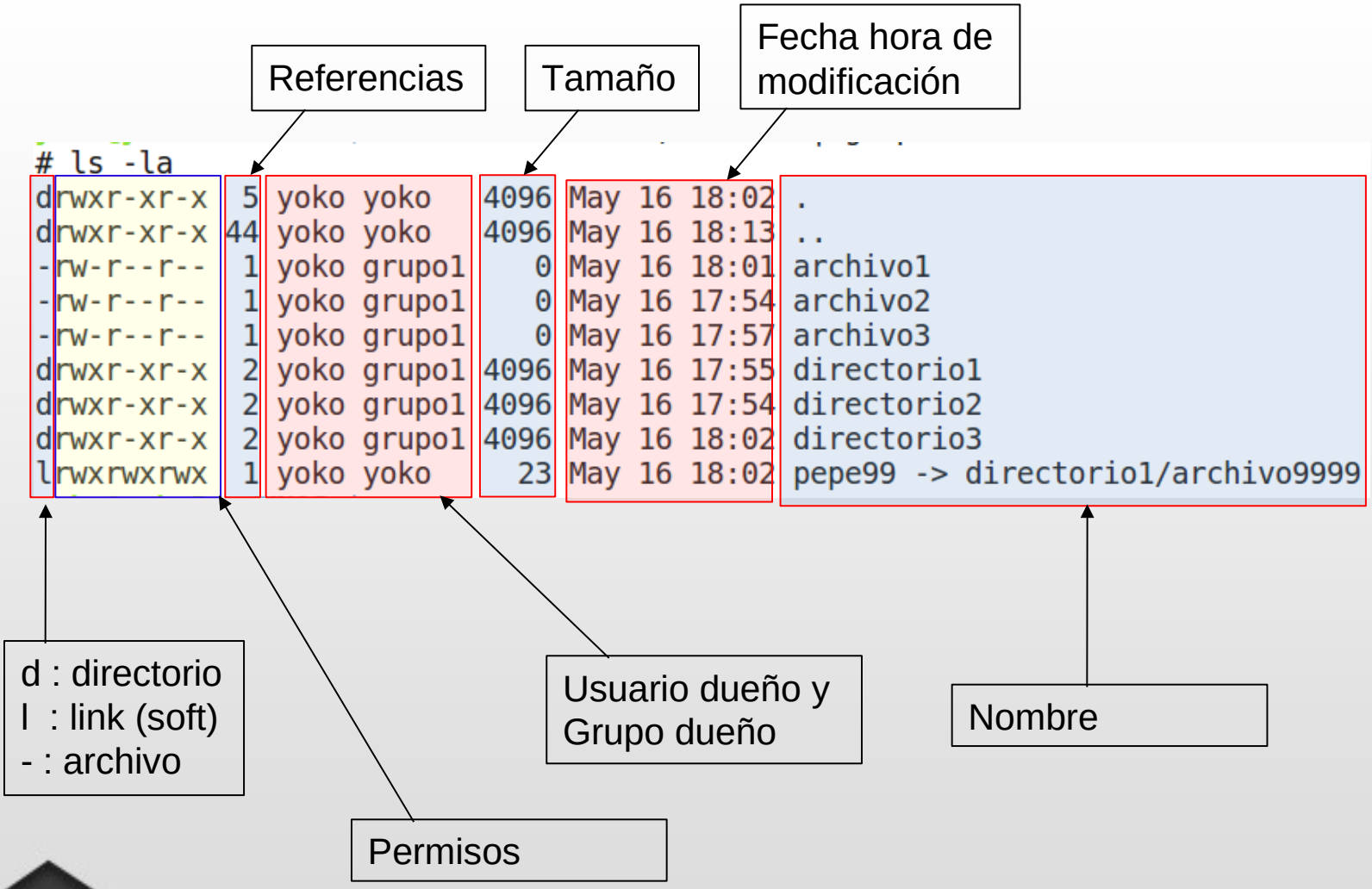
Tipos de Archivos

- Archivos Regulares
  - Texto Plano
    - Source File
  - Binarios
    - Object File
    - Executable File
- Directorios
  - Archivos que mantienen la estructura en el FileSystem

Atributos de un Archivo

- ☑ Nombre
- ☑ Identificador
- ☑ Tipo
- ☑ Localización
- ☑ Tamaño
- ☑ Protección, Seguridad y Monitoreo
  - ✓ Owner, Permisos, Password
  - ✓ Momento en que el usuario lo modifico, creo, accedio por ultima vez
  - ✓ ACLs

Ej: Tipos de archivos y atributos



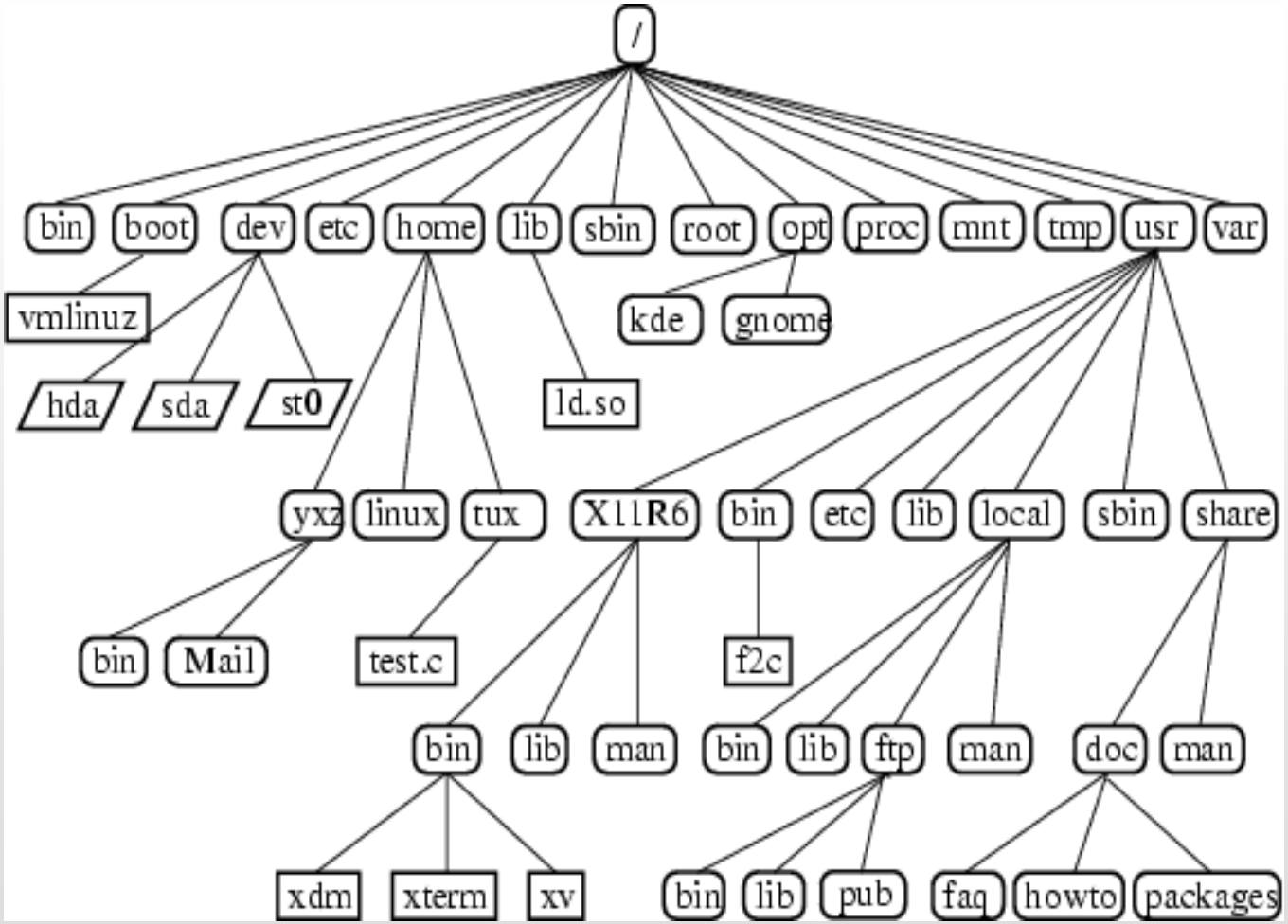
Directorios

- Contiene 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.
- Operaciones en directorios:
  - Buscar un archivo
  - Crear un archivo (entrada de directorio)
  - Borrar un archivo
  - Listar el contenido
  - Renombrar archivos
  - Etc.

Directorios de Archivos (cont.)

- El uso de los directorios ayuda con:
  - La eficiencia: Localización rápida de archivos
  - Uso del mismo Nombre de archivo:
    - Diferentes usuarios pueden tener el mismo nombre de archivo
  - Agrupación: Agrupación lógica de archivos por propiedades/funciones:
    - Ejemplo: Programas Java, Juegos, Librerias, etc.

Estructura de Dir. Jerárquica o Arbol



Estructura de Directorios

- Los archivos pueden ubicarse siguiendo un path desde el directorio raíz y sus sucesivas referencias (**full pathname** del archivo o **PATH absoluto**)
- Distintos archivos pueden tener el mismo nombre pero el fullpathname es único

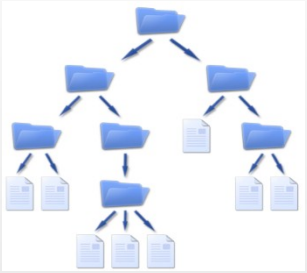
Estructura de Directorios

- El directorio actual se lo llama “directorio de trabajo (working directory)
- Dentro del directorio de trabajo, se pueden referenciar los archivos tanto por su **PATH absoluto** como por su **PATH relativo** indicando solamente la ruta al archivo desde el directorio de trabajo.

Identificación absoluta y relativa

Tanto archivos como directorios se pueden identificar de manera:

- **Absoluta.** El nombre incluye todo el camino del archivo.
  - /var/www/index.html
  - C:\windows\winhelp.exe
- **Relativa.** El nombre se calcula relativamente al directorio en el que se esté
  - si estoy en el directorio /var/spool/mail/
  - Entonces es: ../../www/index.html



Compartir archivos

- En un ambiente multiusuario se necesita que varios usuarios puedan compartir archivos
- Debe ser realizado bajo un esquema de protección:
  - Derechos de acceso
  - Manejo de accesos simultáneos

Protección

- El propietario/administrador debe ser capaz de controlar:
  - Que se puede hacer
    - Derechos de acceso
  - Quien lo puede hacer

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 permisos.

Derechos de acceso (cont.)

- Execution
  - El usuario puede ejecutar
- Reading
  - El usuario puede leer el archivo,
- Appending
  - El usuario puede agregar datos pero no modificar o borrar el contenido del archivo

Derechos de acceso (cont.)

- Updating
  - El usuario puede modificar, borrar y agregar datos. Incluye la creación de archivos, sobreescribirlo y remover datos
- Changing protection
  - El usuario puede modificar los derechos de acceso
- Deletion
  - El usuario puede borrar el archivo

Derechos de acceso

- Owners (propietarios)
  - Tiene todos los derechos
  - Pueder dar derechos a otros usuarios. Se determinan clases:
    - Usuario específico
    - Grupos de usuarios
    - Todos (archivos públicos)

Ejemplo - Protección en UNIX

- Derechos de acceso son definidos independientemente para:
  - (u) user – Owner (creator) of a file
  - (g) group – Group
  - (o) other – all other users of the UNIX system

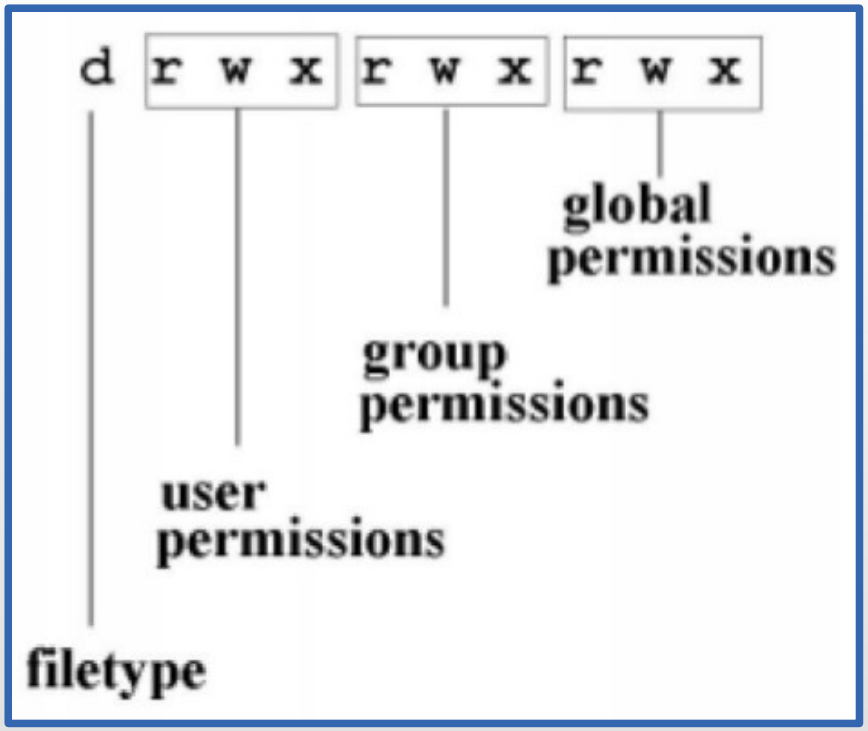
- **Derechos de Acceso:**
  - **(r) Read access right; List right for directory**
  - **(w) Write access right; Includes delete/append rights**
  - **(x) Execute access right; Traverse right for directories**

- Binary representation:
  - (x): Bit 0 (+1)
  - (w): Bit 1 (+2)
  - (r): Bit 2 (+4)
- Rights can be combined
  - Read+Write access right: 6
  - Read+Execute access right: 3
  - Read-only: 2

Ejemplo - Protección en UNIX

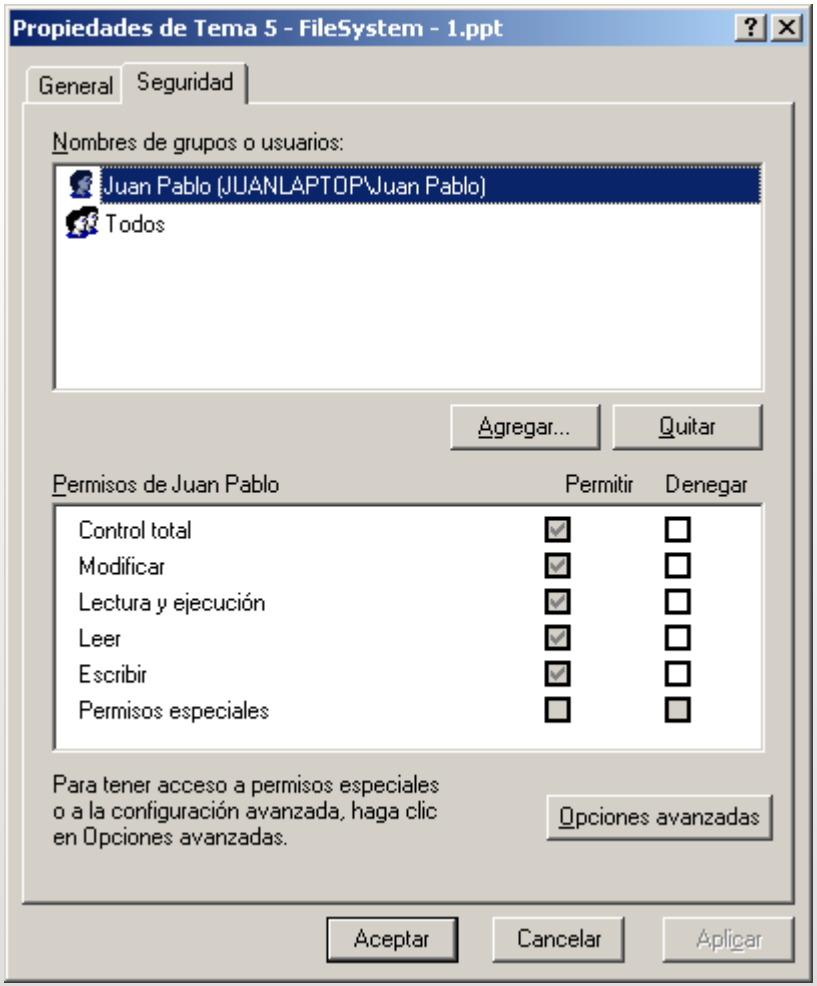
Los permisos que se pueden dar o quitar son:

- r - de lectura
- w - de escritura
- x - de ejecución




```
$ ls -l
drwxrwxr-x 4 www     www     ..
-rw-rw-r-- 1 www     www     x_windows.tex
lrwxrwxrwx 1 lee     lee     img -> ../linux/img/
-rw-rw-r-- 1 lee     lee     test.log
```

Ejemplo - Protección en Windows



Introducción a los Sistemas Operativos

Administración de Archivos - II

I.S.O.

- ☑ Versión: Noviembre 2017
- ☑ Palabras Claves: Archivo, Directorio, File System, Asignación, Espacio Libre

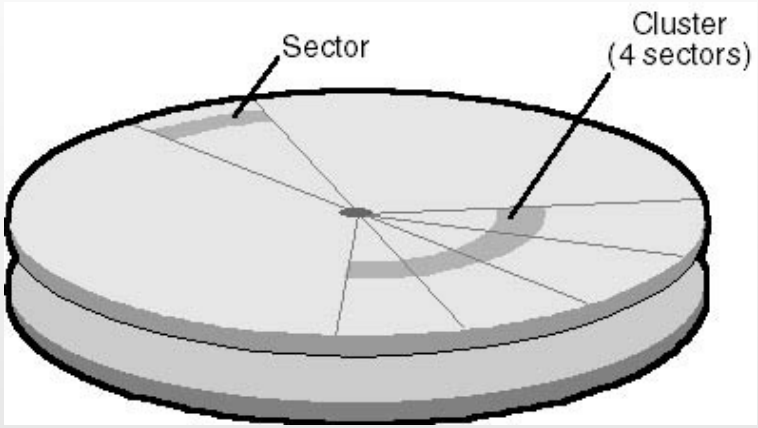
Algunas diapositivas han sido extraídas de las ofrecidas para docentes desde el libro de Stallings (Sistemas Operativos) y el de Silberschatz (Operating Systems Concepts). También se incluyen diapositivas cedidas por Microsoft S.A.

Metas del Sistema de Archivos

- ☑ Brindar espacio en disco a los archivos de usuario y del sistema.
- ☑ Mantener un registro del espacio libre. Cantidad y ubicación del mismo dentro del disco.

Conceptos

- Sector
  - Unidad de almacenamiento utilizada en los Discos Rígidos
- Bloque/Cluster
  - Conjuntos de sectores consecutivos
- File System
  - Define la forma en que los datos son almacenados
- FAT: File Allocation Table
  - Contiene información sobre en que lugar están alocados los distintos archivos



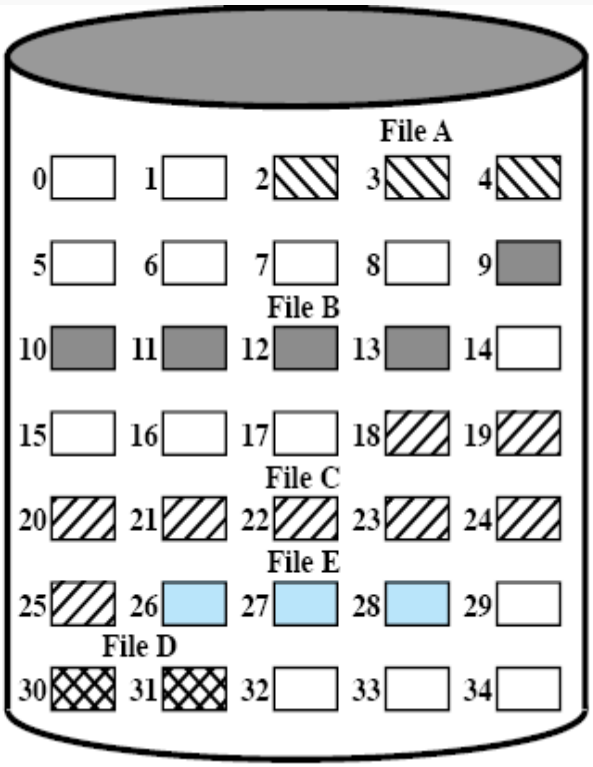
Pre-asignación

- Se necesita 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
- Posibilidad de utilizar sectores contiguos para almacenar los datos de un archivo
- Qué pasa cuando el archivo supera el espacio asignado?

Asignación Dinámica

- El espacio se solicita a medida que se necesita
- Los bloques de datos pueden quedar de manera no contigua

Formas de Asignación - Continua



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?

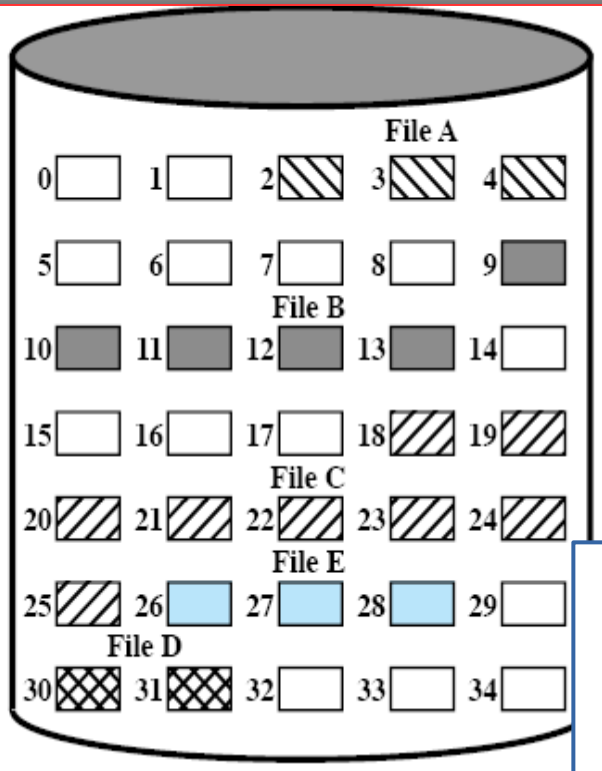
Formas de Asignación - Continua

- Conjunto continuo de bloques son utilizados
- Se requiere una pre-asignación
  - Se debe conocer el tamaño del archivo durante su creación
- File Allocation Table (FAT) es simple
  - Sólo una entrada que incluye Bloque de inicio y longitud
- El archivo puede ser leído con una única operación
- Puede existir fragmentación externa
  - Compactación

File Allocation Table

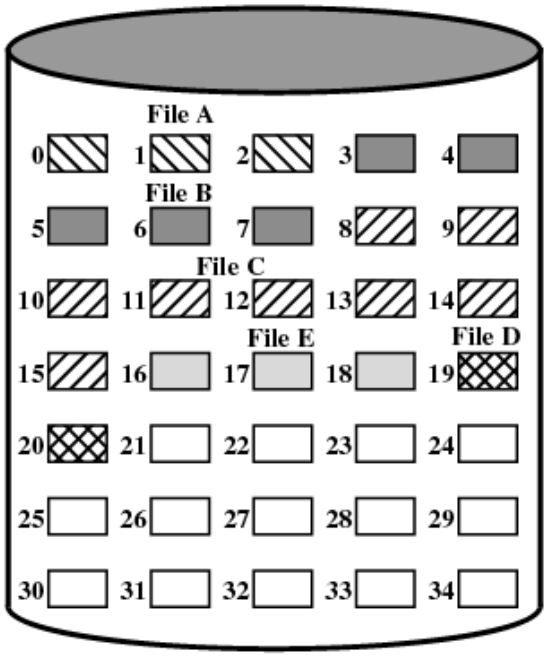
File Name	Start Block	Length
File A	2	3
File B	9	5
File C	18	8
File D	30	2
File E	26	3

Compactación



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



File Allocation Table

File Name	Start Block	Length
File A	0	3
File B	3	5
File C	8	8
File D	19	2
File E	16	3

Figure 12.8 Contiguous File Allocation (After Compaction)

Formas de Asignación - Continua

- Problemas de la técnica
  - Encontrar bloques libres continuos en el disco
  - Incremento del tamaño de un archivo

Formas de Asignación - Encadenada

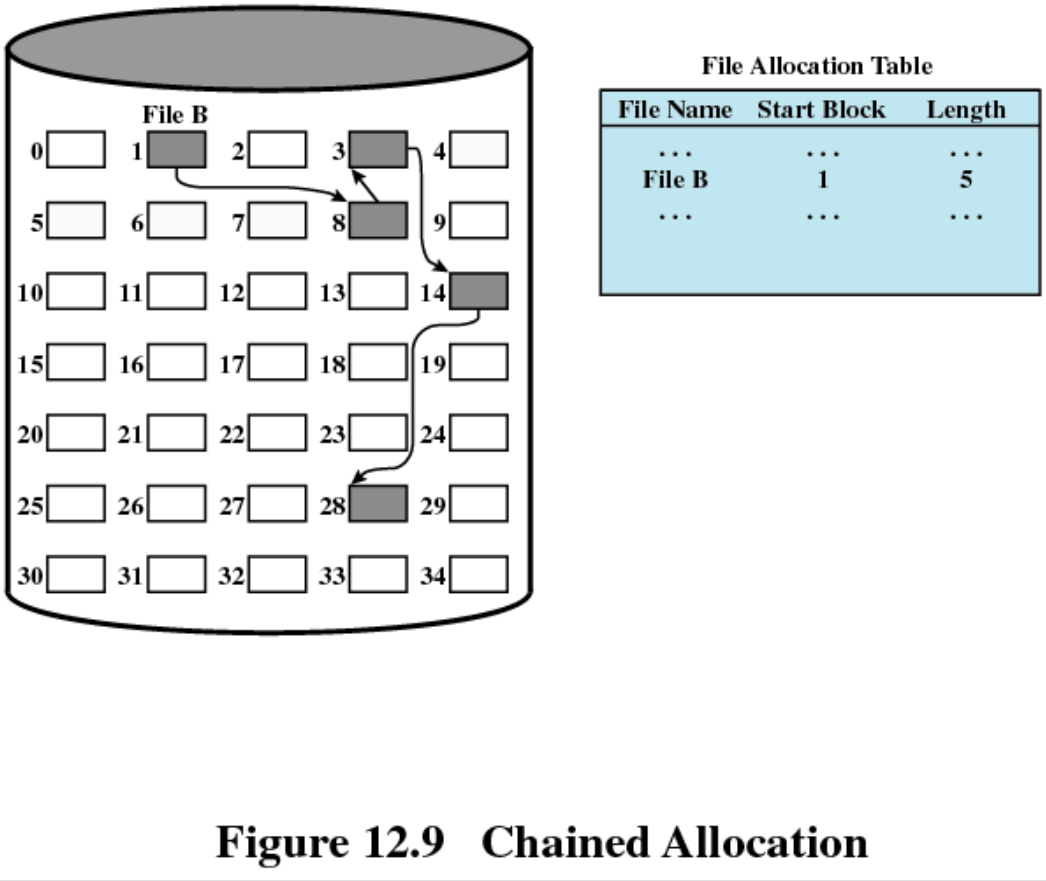
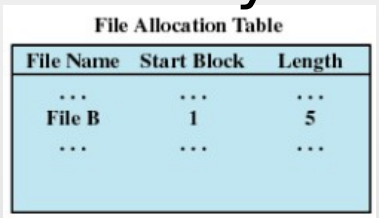


Figure 12.9 Chained Allocation

Formas de Asignación - Encadenada

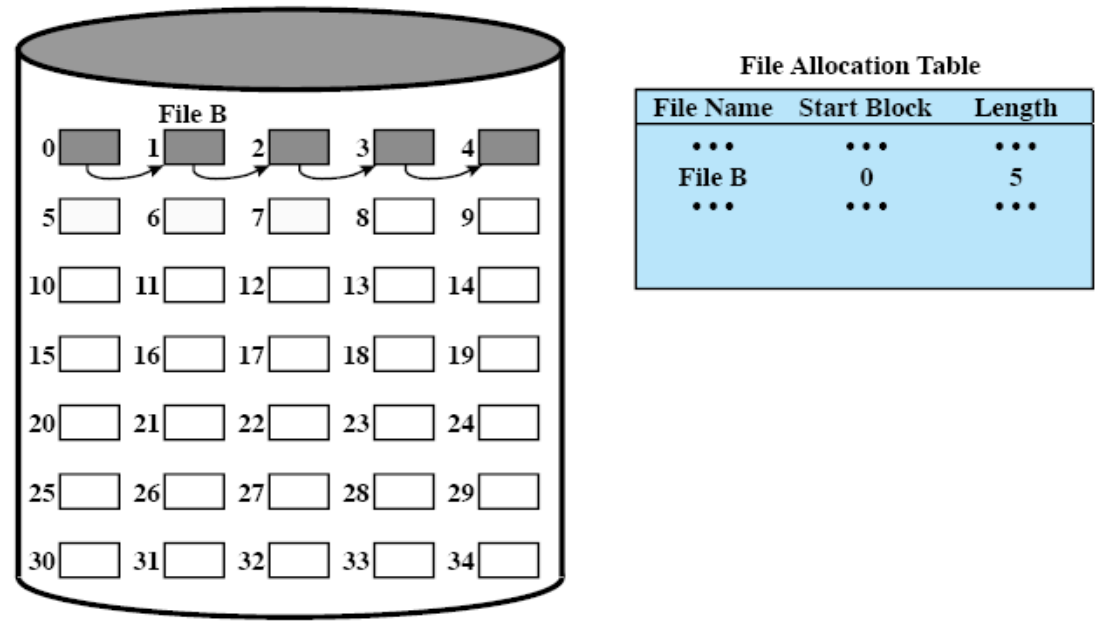
- Asignación en base a bloques individuales
- Cada bloque tiene un puntero al próximo bloque del archivo
- File allocation table
  - Única entrada por archivo: Bloque de inicio y tamaño del archivo



- No hay fragmentación externa
- Útil para acceso secuencial (no random)
- Los archivos pueden crecer bajo demanda
- No se requieren bloques contiguos

Formas de Asignación - Encadenada

- Se pueden consolidar los bloques de un mismo archivo para garantizar cercanía de los bloques de un mismo archivo.



Formas de Asignación - Indexada

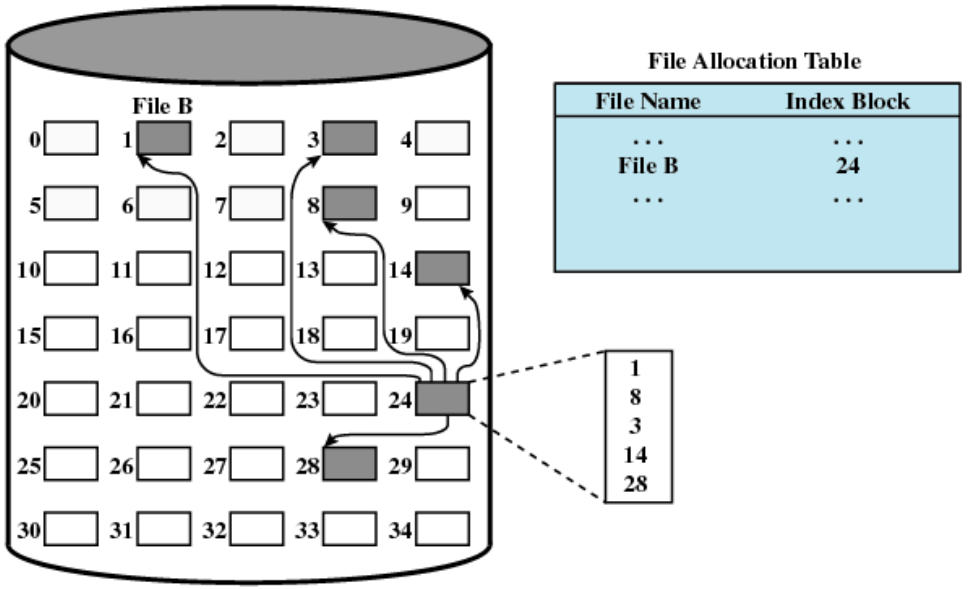
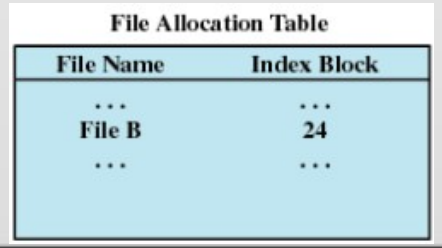


Figure 12.11 Indexed Allocation with Block Portions

- La FAT contiene un puntero al bloque índice
- El bloque índice no contiene datos propios del archivo, sino que contiene un índice a los bloques que lo componen

Formas de Asignación - Indexada

- Asignación en base a bloques individuales
- No se produce Fragmentación Externa
- El acceso “random” a un archivo es eficiente
- File Allocation Table
  - Única entrada con la dirección del bloque de índices (index node / i-node)



Formas de Asignación - Indexada

- Variante: asignación por secciones
- A cada entrada del bloque índice se agrega el campo longitud
- El índice apunta al primer bloque de un conjunto almacenado de manera contigua

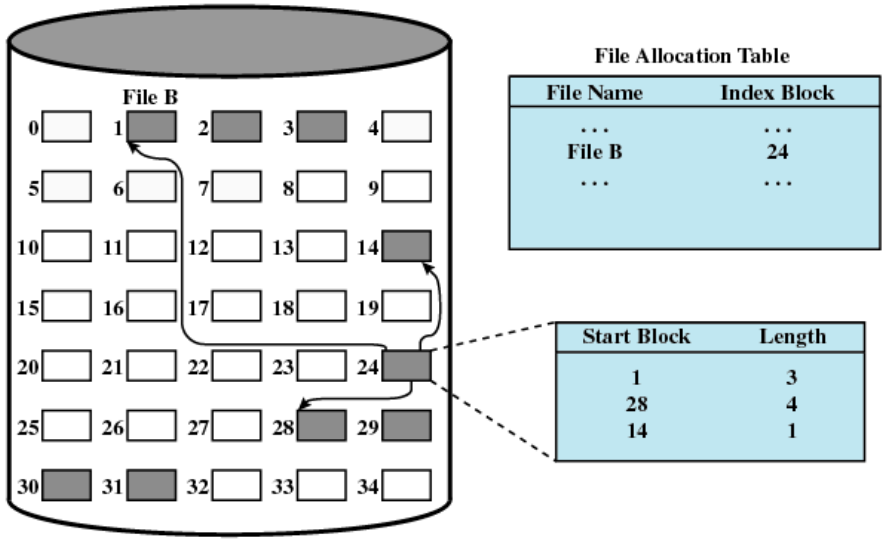
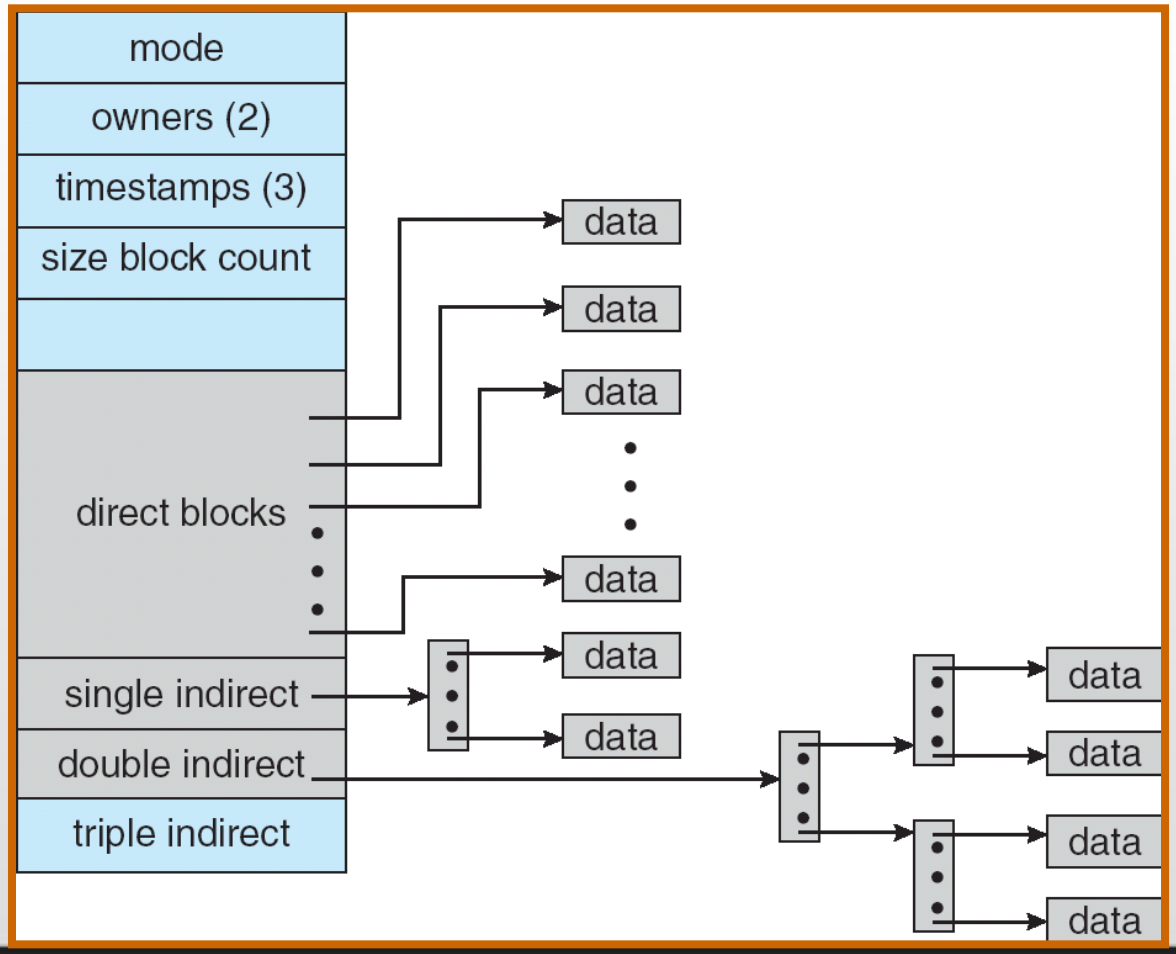


Figure 12.12 Indexed Allocation with Variable-Length Portions

Formas de Asignación - Indexada

- Variante: niveles de indirección
  - Existen bloques directos de datos
  - Otros bloques son considerados como bloque índices (apuntan a varios bloques de datos)
  - Puede haber varios niveles de indirección



Asignación Indexada - Ejemplo

Cada I-NODO contiene 9 direcciones a los bloques de datos, organizadas de la siguiente manera:

- 7 de direccionamiento directo.
- 1 de direccionamiento indirecto simple
- 1 de direccionamiento indirecto doble

Si cada bloque es de 1KB y cada dirección usada para referenciar un bloque es de 32 bits:

- ✓ ¿Cuántas referencias (direcciones) a bloque pueden contener un bloque de disco?

  1 KB / 32 bits = 256 direcciones

- ✓ ¿Cuál sería el tamaño máximo de un archivo?

  $(7 + 256 + 256^2) * 1\ KB = 65799\ KB = 64{,}25\ MB$

Gestión de Espacio Libre

- Control sobre cuáles de los bloques de disco están disponibles.
- *Alternativas*
  - *Tablas de bits*
  - *Bloques libres encadenados*
  - *Indexación*

Espacio Libre - Tabla de bits

- Tabla (vector) con 1 bit por cada bloque de disco
- Cada entrada:
  - 0 = bloque libre 1 = bloque en uso
- Ventaja
  - Fácil encontrar un bloque o grupo de bloques libres.
- Desventaja
  - Tamaño del vector en memoria

    tamaño disco bytes / tamaño bloque en sistema archivo

    Eje: Disco 16 Gb con bloques de 512 bytes ➔ 32 Mb.

Espacio Libre - Tabla de bits (cont.)

- Ejemplo

  00111

  00001

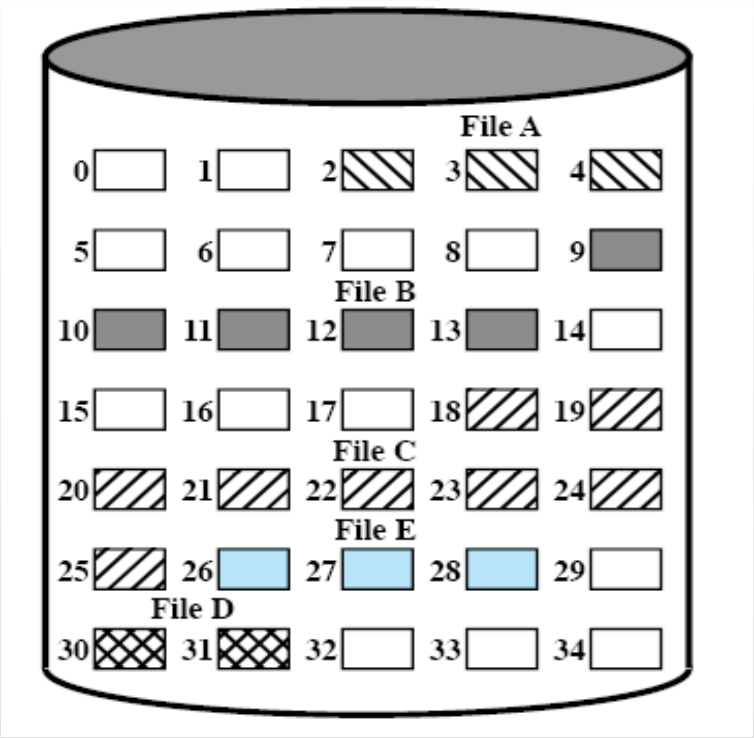
  11110

  00011

  11111

  11110

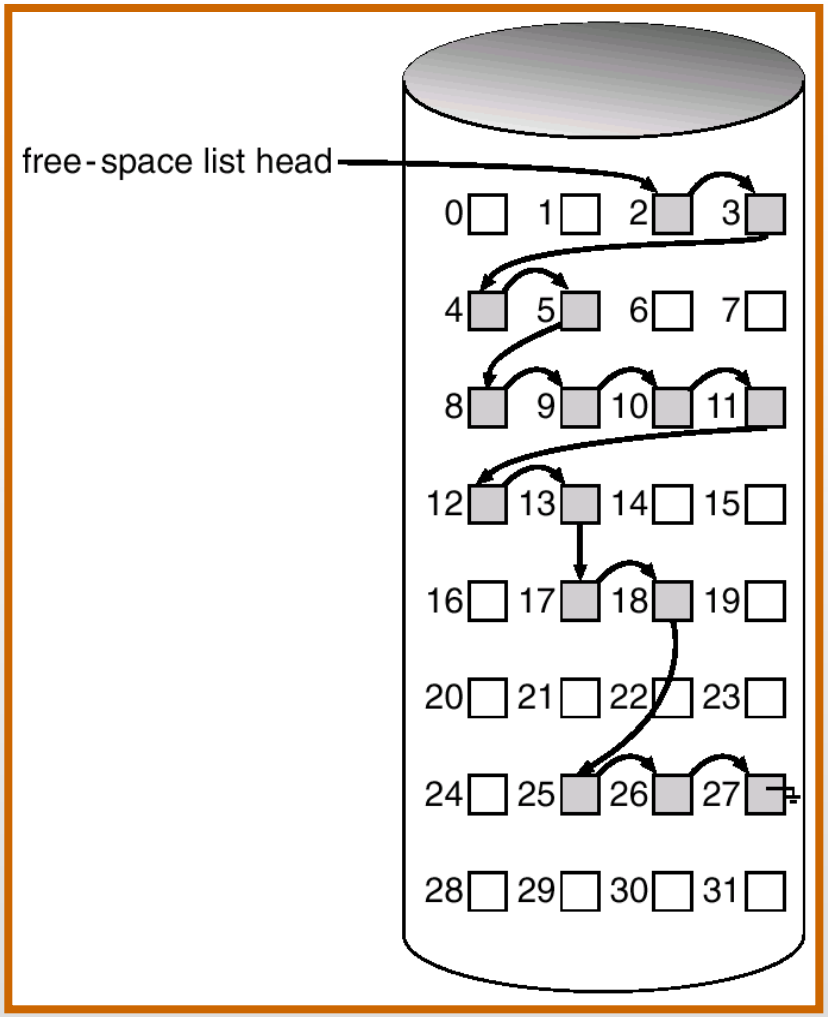
  11000



Espacio Libre - Bloques Encadenados

- Se tiene un puntero al primer bloque libre.
- Cada bloque libre tiene un puntero al siguiente bloque libre
- Ineficiente para la búsqueda de bloques libres ➔ Hay que realizar varias operaciones de E/S para obtener un grupo libre.
- Problemas con la pérdida de un enlace
- Difícil encontrar bloques libres consecutivos

Espacio Libre - Bloques Encadenados



Espacio Libre - Indexación (o 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.

Introducción a los Sistemas Operativos

Administración de Archivos - III

I.S.O.

- ☑ Versión: Noviembre 2017
- ☑ Palabras Claves: Archivo, File System, Directorio, UNIX, I-NODO, Windows, FAT

Algunas diapositivas han sido extraídas de las ofrecidas para docentes desde el libro de Stallings (Sistemas Operativos) y el de Silberschatz (Operating Systems Concepts). También se incluyen diapositivas cedidas por Microsoft S.A.

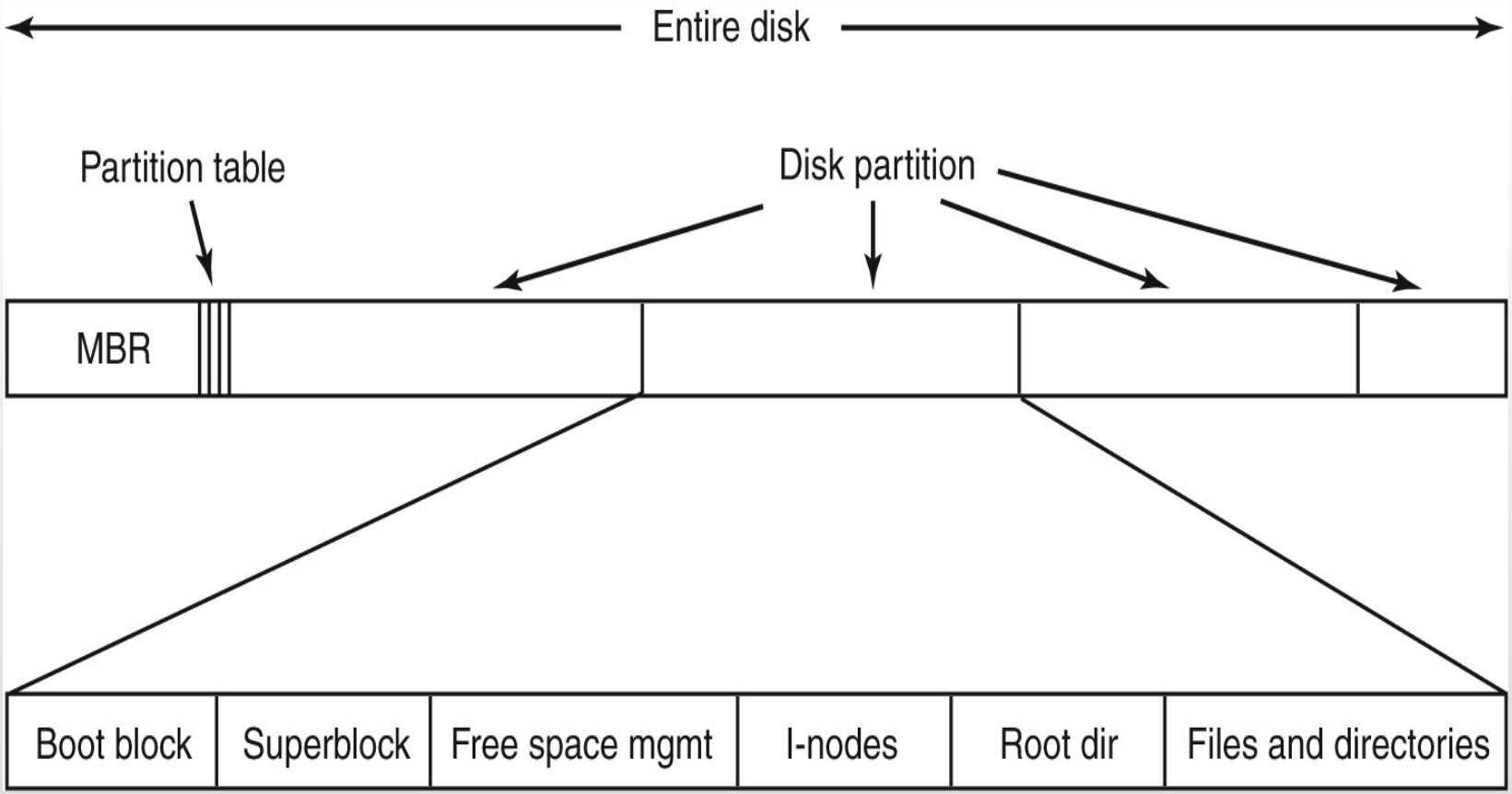
UNIX - Manejo de archivos

- Tipos de Archivos
  - Archivo común
  - Directorio
  - Archivos especiales (dispositivos /dev/sda)
  - Named pipes (comunicación entre procesos)
  - Links (comparten el i-nodo, solo dentro del mismo filesystem)
  - Links simbólicos (tiene i-nodo propio, para filesystems diferentes)

UNIX - Estructura del Volumen

- Cada disco físico puede ser dividido en uno o mas volúmenes. Cada volumen o partición contiene un sistema de archivos. Cada sistema de archivos contiene:
  - Boot Block: Código para bootear el S.O.
  - Superblock: Atributos sobre el File System
    - Bloques/Clusters libres
  - I-NODE Table: Tabla que contiene todos los I-NODOS
    - I-NODO: Estructura de control que contiene la información clave de un archivo
  - Data Blocks: Bloques de datos de los archivos

UNIX - Estructura del Volumen

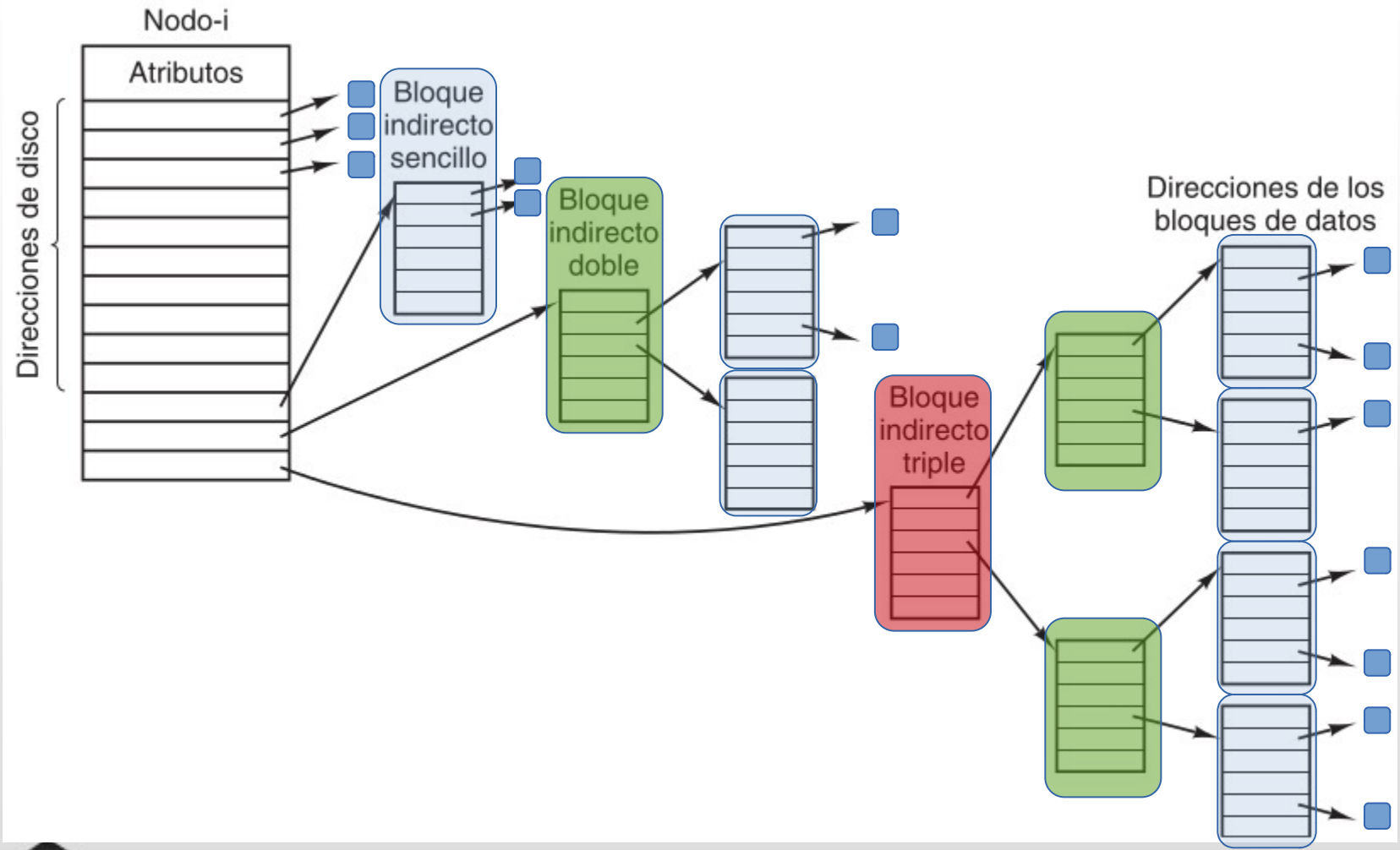


UNIX – Información del i-nodo

Table 12.4 Information in a UNIX Disk-Resident Inode

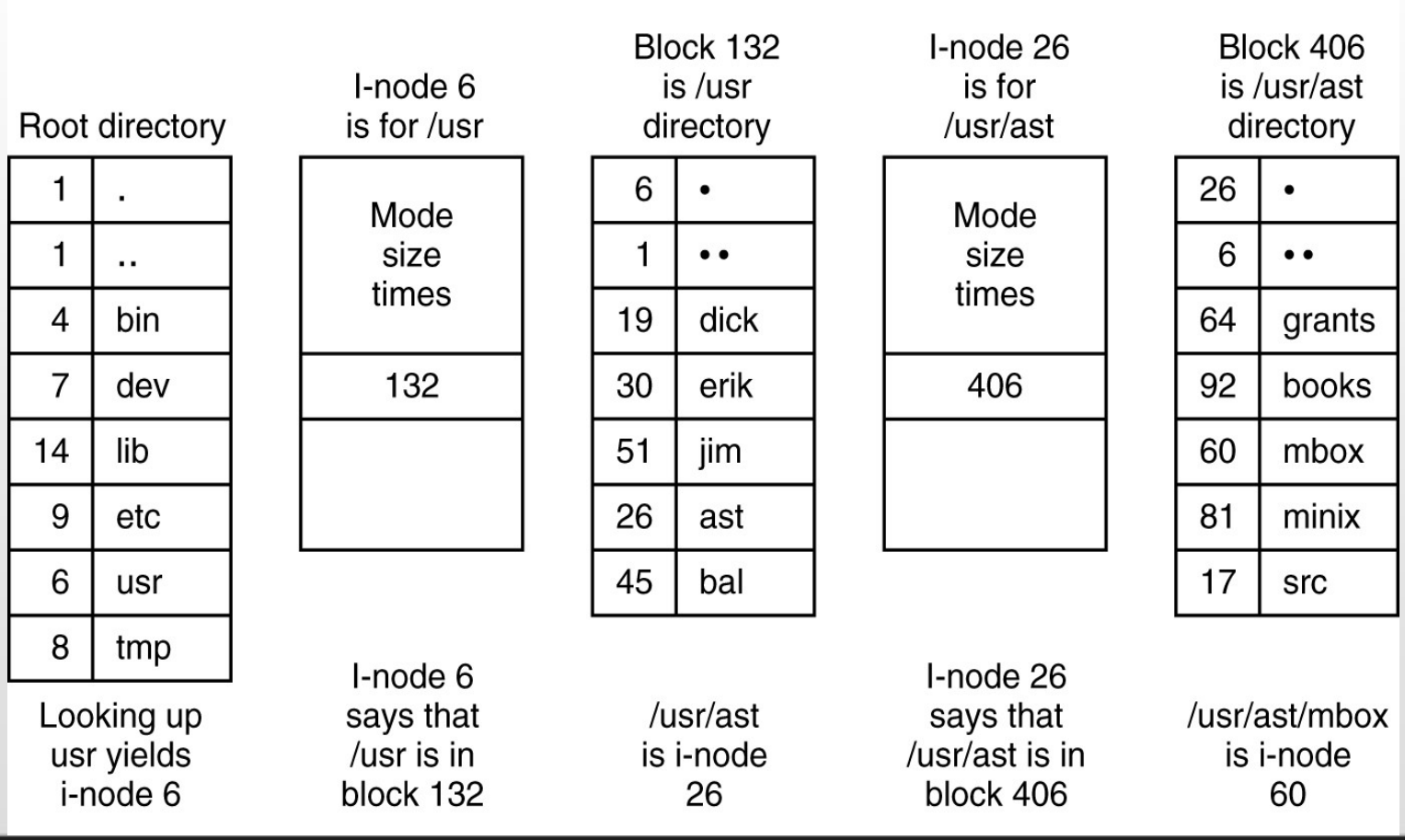
File Mode	16-bit flag that stores access and execution permissions associated with the file. 12-14 File type (regular, directory, character or block special, FIFO pipe 9-11 Execution flags 8 Owner read permission 7 Owner write permission 6 Owner execute permission 5 Group read permission 4 Group write permission 3 Group execute permission 2 Other read permission 1 Other write permission 0 Other execute permission
Link Count	Number of directory references to this inode
Owner ID	Individual owner of file
Group ID	Group owner associated with this file
File Size	Number of bytes in file
File Addresses	39 bytes of address information
Last Accessed	Time of last file access
Last Modified	Time of last file modification
Inode Modified	Time of last inode modification

UNIX - I-NODO


Nodo-i
Atributos
Direcciones de disco
Bloque indirecto sencillo
Bloque indirecto doble
Bloque indirecto triple
Direcciones de los bloques de datos

UNIX - Directorios (cont)

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



Windows - File Systems Soportados

- ☑ CD-ROM File System (CDFS) → CD
- ☑ Universal Disk Format (UDF) → DVD, Blu-Ray
- ☑ File Allocation Table
  - FAT12 → MS-DOS v3.3 a 4.0 (año 1980), floppy
  - FAT16 → MS-DOS 6.22, nombres cortos de archivo
  - FAT32 → MS-DOS 7.10, nombres largos pero no soportados en MS-DOS
- ☑ New Technology File System (NTFS)

Windows - FAT

- FAT (File Allocation Table) es un sistema de archivos utilizado originalmente por DOS y Windows 9x
- ¿Porqué Windows aun soporta FAT file systems?:
  - Por compatibilidad con otro SO en sistemas multiboot
  - Para permitir upgrades desde versiones anteriores
  - Para formato de dispositivos como diskettes
- Las distintas versiones de FAT se diferencian por un número que indica la cantidad de bits que se usan para identificar diferentes bloques o clusters:
  - FAT12
  - FAT16
  - FAT32

Windows - FAT

- Se utiliza un mapa de bloques del sistema de archivos, llamado FAT.
- La FAT tiene tantas entradas como bloques.
- La FAT, su duplicado y el directorio raiz se almacenan en los primeros sectores de la partición

FAT format organization

Windows - FAT

- ☑ Se utiliza un esquema de ASIGNACION ENCADENADA.
- ☑ La única diferencia es que el puntero al proximo bloque está en la FAT y no en los bloques
- ☑ Bloques libres y dañados tienen codigos especiales

DIRECTORIO

Nombre		1er bloque	Tamaño
FICH_A		7	4
FICH_B		4	1
FICH_C		2	3

FAT

Tamaño del disco	0
	1
6	2
14	3
EOF	4
EOF	5
5	6
3	7
EOF	8
LIBRE	9
LIBRE	10
LIBRE	11
LIBRE	12
DAÑADO	13
8	14
LIBRE	15
...	...

Windows - FAT12

- En sistemas FAT12, al utilizarse 12 bits para la identificación del sector, la misma se limita a 2^{12} (4096) sectores
  - Windows utiliza tamaños de sector desde los 512 bytes hasta 8 KB, lo que limita a un tamaño total de volume de 32 MB → $2^{12} * 8$ KB
  - Windows utiliza FAT12 como Sistema de archivos de los disketts de 3,5 y 12 pulgadas que pueden almacenar hasta 1,44 MB de datos

Windows - FAT16

- FAT16 al utilizar 16 bits para identificar cada sector puede tener hasta 2^{16} (65.536) sectores en un volúmen
  - En windows el tamaño de sector en FAT16 varía desde los 512 bytes hasta los 64 KB, lo que limita a un tamaño máximo de volume de 4 GB.
  - El tamaño de sector dependía del tamaño del volume al formatearlo

Windows - FAT32

- FAT32 fue el Filesystem mas reciente de la línea (posteriormente salió exFAT que algunos lo conocen como FAT64)
- FAT32 utiliza 32 bits para la identificación de sectores, pero reserva los 4 bits superiors, con lo cual efectivamente solo se utiliazan 28 bits para la identificación:
  - El tamaño de sector en FAT 32 puede ser de hasta 32 KB, con lo cual tiene una capacidad teórica de direccionar volúmenes de hasta 8 TB
  - El modo de identificación y acceso de los sectores lo hace mas eficiente que FAT16. Con tamaño de sector de 512 bytes, puede direccionar volúmenes de hasta 128 GB.

Windows - FAT

Block size	FAT-12	FAT-16	FAT-32
0.5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB

Windows - NTFS

- NTFS es el filesystem nativo de Windows desde Windows NT
- NTFS usa 64-bit para referenciar sectores
  - Teoricamente permite tener volumenes de hasta 16 Exabytes (16 billones de GB)
- ¿Porqué usar NTFS en lugar de FAT? FAT es simple, mas rápido para ciertas operaciones, pero NTFS soporta:
  - Tamaños de archivo y de discos mayores
  - Mejora performance en discos grandes
  - Nombres de archivos de hasta 255 caracteres
  - Atributos de seguridad
  - Transaccional