



# SISTEMAS DE ARCHIVOS

DEPARTAMENTO DE CIENCIAS E INGENIERÍA  
DE LA COMPUTACIÓN

UNIVERSIDAD NACIONAL DEL SUR



# SISTEMAS DE ARCHIVOS

## REQUERIMIENTOS ESENCIALES

1. Debe ser posible almacenar gran cantidad de información.
2. La información debe sobrevivir a la finalización del proceso que está utilizándola.
3. Múltiples procesos pueden acceder simultáneamente a la información.

# SISTEMAS DE ARCHIVOS INTERFAZ

---

# SISTEMA DE ARCHIVOS: INTERFAZ

1. Concepto de archivos
2. Métodos de Acceso
3. Estructura de Directorio
4. Montaje de Sistemas de Archivos
5. Archivos Compartidos

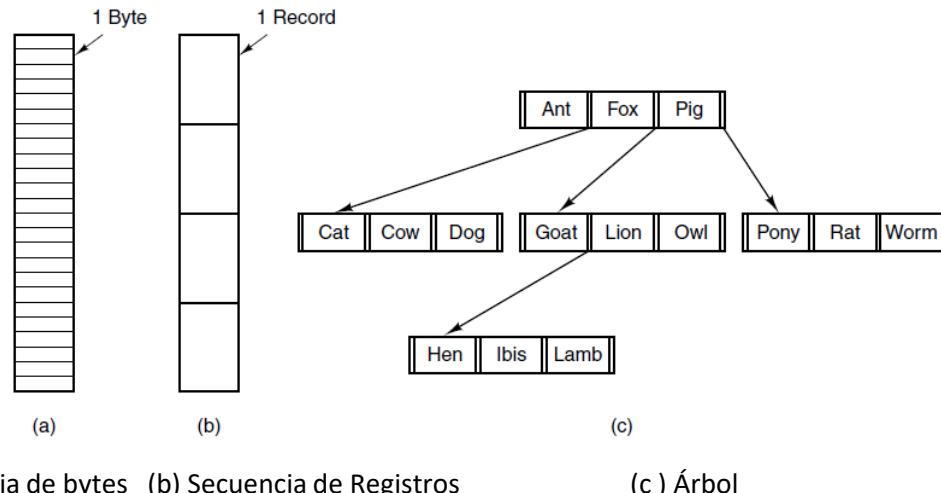
# CONCEPTO – ESTRUCTURA. ARCHIVO

Concepto:

- Espacio de direcciones lógicas contiguas. Tipos: Datos ó Programas

Estructura:

- Ninguna – secuencia de palabras, bytes
- Estructura de registros simple
- Estructuras Complejas



(a) Secuencia de bytes (b) Secuencia de Registros

(c) Árbol

# ARCHIVO: ATRIBUTOS Y OPERACIONES

## ATRIBUTOS

- Nombre
- Tipo
- Locación
- Tamaño
- Protección
- Tiempo, fecha, e identificación de usuario

## OPERACIONES

- creación
- escritura
- lectura
- reposición puntero corriente
- borrado
- truncado
- $\text{open}(F_i)$
- $\text{close } (F_i)$

### Información requerida para administrarlos

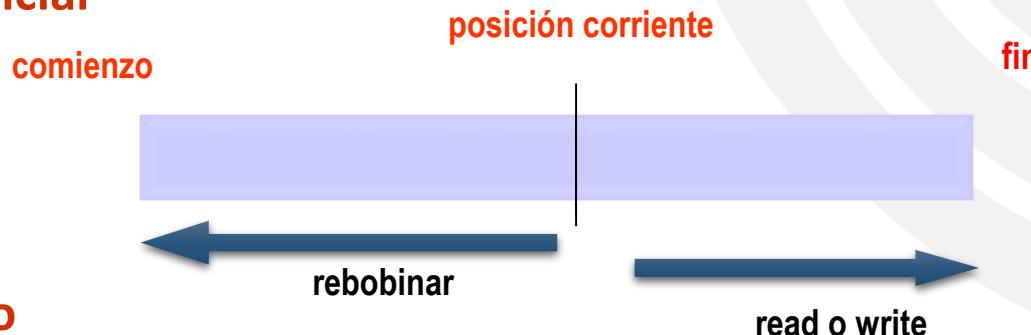
- Puntero corriente del archivo
- Cuenta de archivo abierto
- Locación en el disco del archivo
- Derechos de acceso

# LOCKING DE ARCHIVOS ABIERTOS

- Provisto por algunos sistemas operativos y sistemas de archivos
- Media en el acceso al archivo
- Mandatorio flexible:
  - **Mandatorio** – el acceso es rechazado dependiendo de los locks que se tienen y requeridos. Ejemplo: Windows
  - **Flexible** – los procesos verifican el estado de los locks y decide que hacer. Ejemplo: UNIX

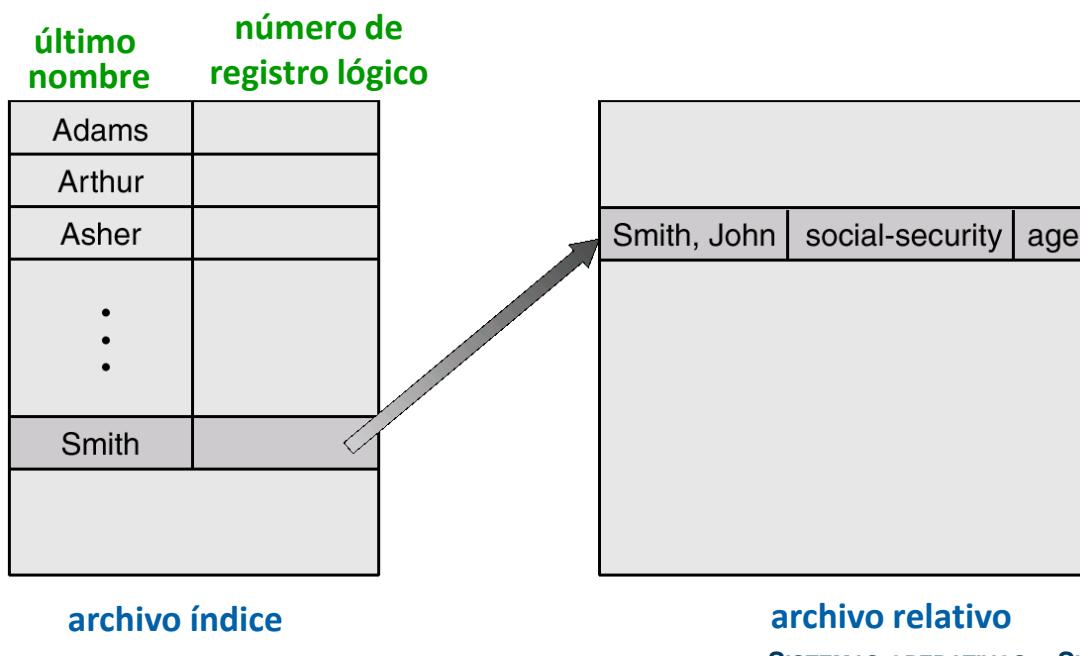
# MÉTODOS DE ACCESOS

- **Acceso Secuencial**



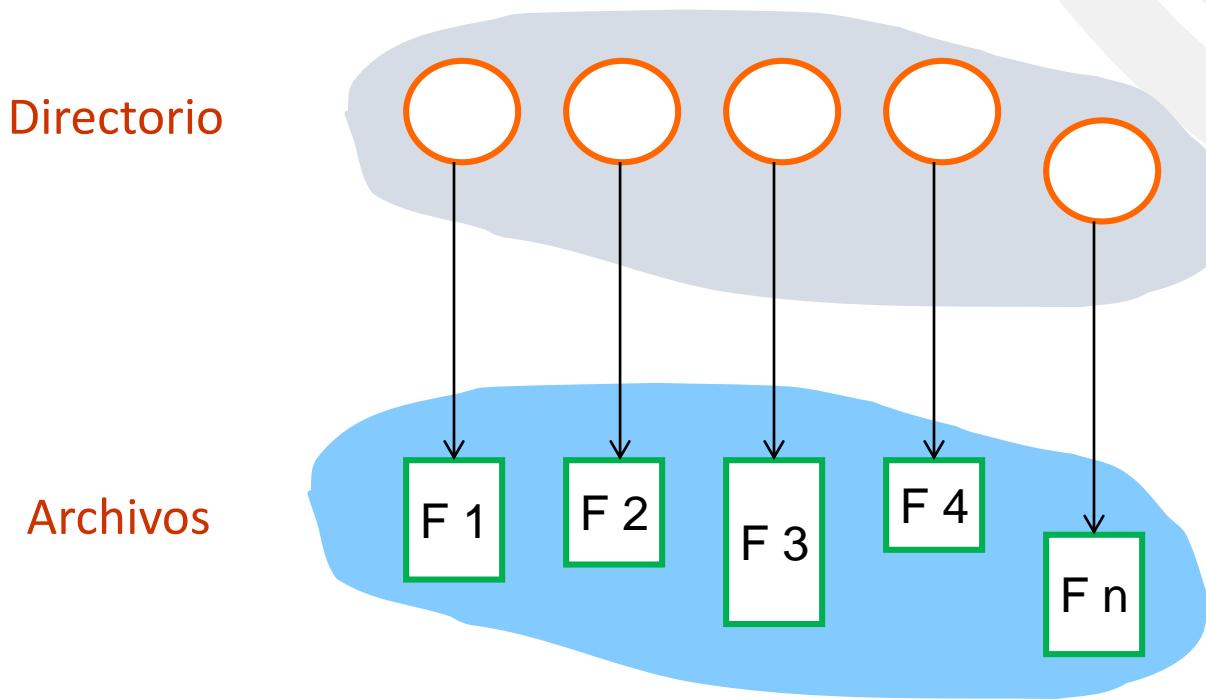
- **Acceso Directo**

- **Acceso Indexado**



# ESTRUCTURA DE DIRECTORIO

- Una colección de nodos conteniendo información sobre todos los archivos.



- La estructura de directorio y los archivos residen en el disco.

# DIRECTORIO

## INFORMACIÓN

- Nombre
- Dirección
- Longitud corriente
- Máxima longitud
- Fecha del último acceso
- Fecha de la última actualización (para vuelco)
- Tipo
- ID del dueño
- Información de protección

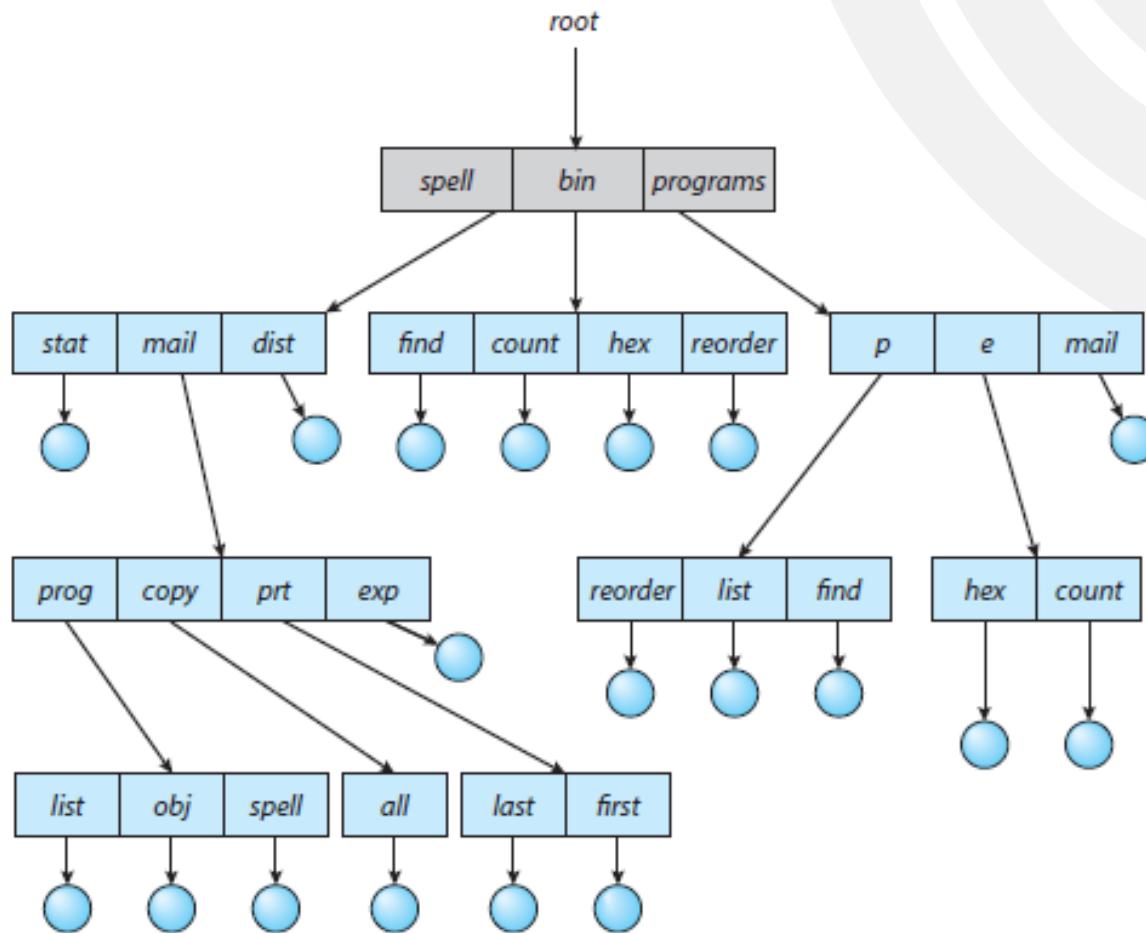
## OPERACIONES

- Búsqueda de un archivo
- Creación de un archivo
- Borrado de un archivo
- Listado de un directorio
- Renombrado de un archivo
- Atravesar un sistema de archivos

# ORGANICE EL DIRECTORIO (LÓGICAMENTE) PARA OBTENER:

- Eficiencia – localizar un archivo rápidamente.
- Nombres – conveniente para los usuarios.
- Agrupamiento – agrupamiento lógico de archivos por propiedades, (p.e., todos los programas C, todos los juegos, ...)

# ESTRUCTURA ARBÓREA DE DIRECTORIOS

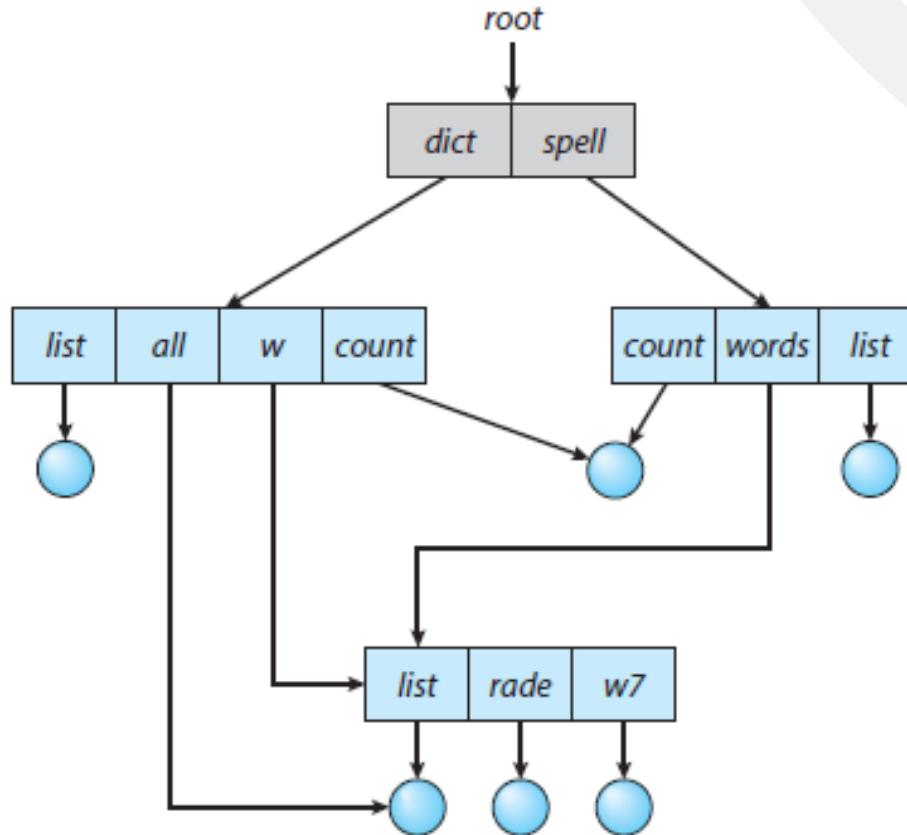


# ESTRUCTURA ARBÓREA DE DIRECTORIOS

- Búsqueda eficiente
- Capacidad de agrupamiento
- Directorio corriente (directorio de trabajo)
- Camino de nombres absoluto o relativo

# GRAFO ACÍCLICO DE DIRECTORIOS

- Puede compartir subdirectorios y archivos.



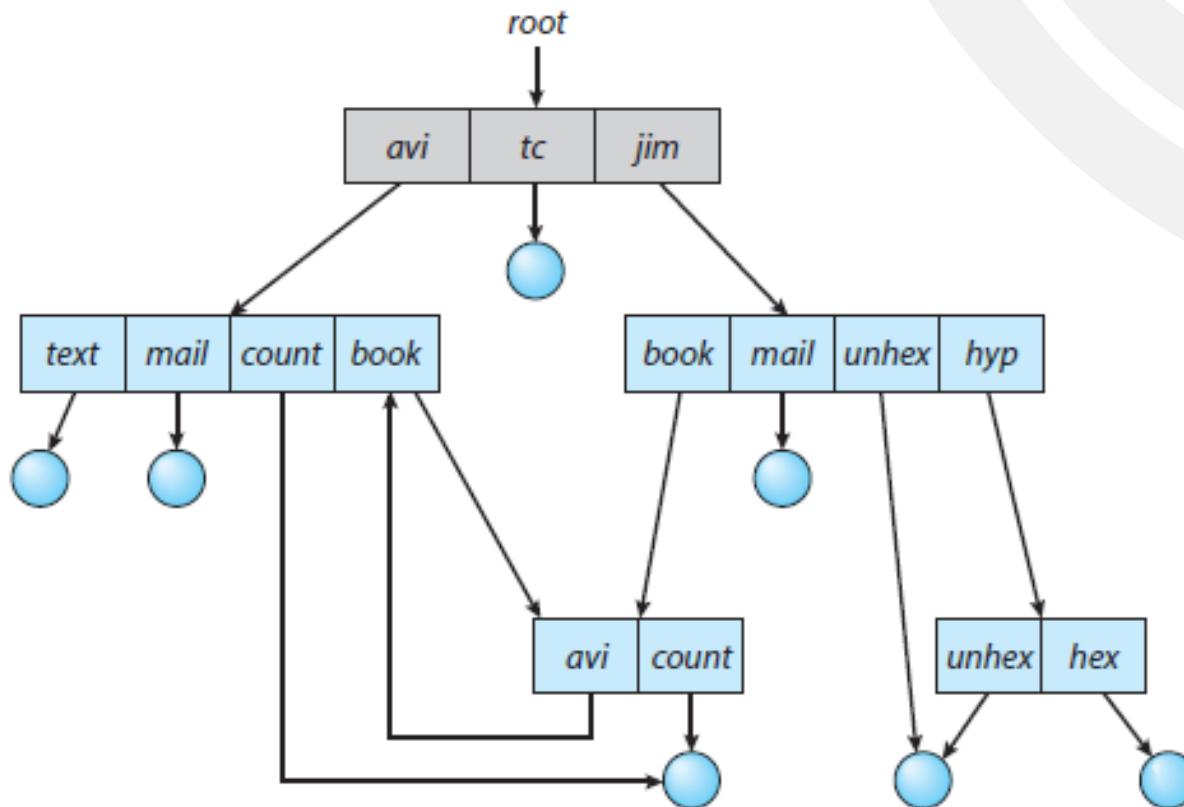
# GRAFO ACÍCLICO DE DIRECTORIOS

- Dos nombres diferentes (alias)
- Si *dict* borra *list* ⇒ quedan punteros solitarios.

Soluciones:

- ▶ Punteros hacia atrás, así se pueden borrar todos los punteros. Los registros de tamaño variable son un problema.
- ▶ Punteros hacia atrás usando una organización “cadena margarita”.
- ▶ Contador de entradas al archivo.
- Nueva entrada en el directorio
  - ▶ Link – Otro nombre (puntero) a un archivo existente
  - ▶ Resuelva el link – siga el puntero hasta localizar el archivo

# GRAFO GENERAL DE DIRECTORIO



# GRAFO GENERAL DE DIRECTORIO

¿Cómo se garantiza que no haya ciclos?

- Permita enlaces (links) a archivos y no a sudirectorios.
- “Garbage collection”.
- Cada vez que se agrega un nuevo enlace (link) se usa un algoritmo de detección de ciclos para determinar si está bien.

# PROTECCIÓN

- El **creador/dueño** del archivo debería poder controlar:
  - ▶ que cosas pueden hacerse
  - ▶ por quién
- Tipos de acceso
  - ▶ Read
  - ▶ Write
  - ▶ Execute
  - ▶ Append
  - ▶ Delete
  - ▶ List

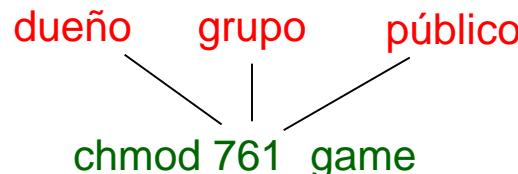
# LISTAS DE ACCESO Y GRUPOS

- Modos de acceso: read, write, execute
- Tres clases de usuarios

RWX

		RWX
a) acceso dueño	7	$\Rightarrow 111$
b) acceso grupos	6	$\Rightarrow 110$
c) acceso público	1	$\Rightarrow 001$

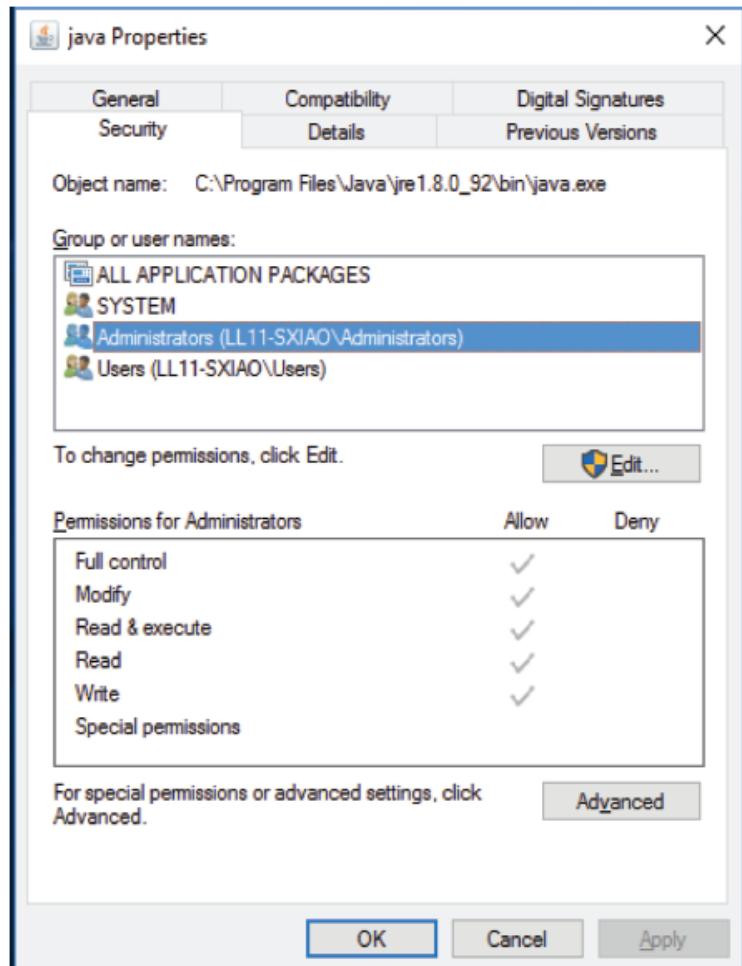
- Pedir al administrador crear un grupo (único nombre), sea G, y adicionar algún usuario al mismo.
- Para un archivo particular (sea *game*) o subdirectorio, definir un acceso apropiado.



Agregar un grupo a un archivo

chgrp *G* *game*

# EJEMPLOS: WINDOWS Y UNIX-LINUX



-rw-rw-r--	1	pbg	staff	31200	Sep 3 08:30	intro.ps
drwx-----	5	pbg	staff	512	Jul 8 09:33	private/
drwxrwxr-x	2	pbg	staff	512	Jul 8 09:35	doc/
drwxrwx---	2	jwg	student	512	Aug 3 14:13	student-proj/
-rw-r--r--	1	pbg	staff	9423	Feb 24 2017	program.c
-rwxr-xr-x	1	pbg	staff	20471	Feb 24 2017	program
drwx--x-x	4	tag	faculty	512	Jul 31 10:31	lib/
drwx-----	3	pbg	staff	1024	Aug 29 06:52	mail/
drwxrwxrwx	3	pbg	staff	512	Jul 8 09:35	test/

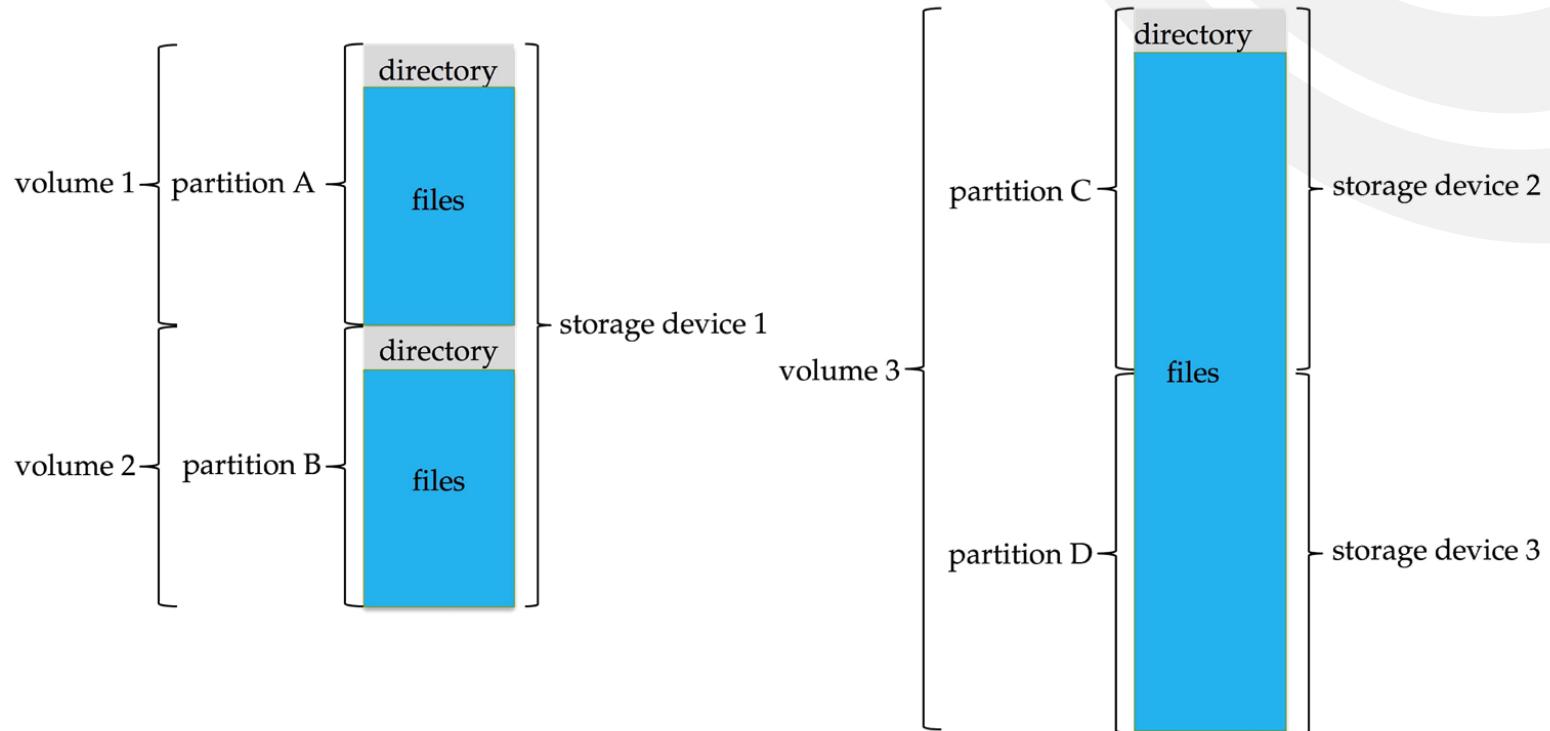
# SISTEMAS DE ARCHIVOS INTERNAL

---

# SISTEMA DE ARCHIVOS

- Las computadoras de uso general pueden tener múltiples dispositivos de almacenamiento.
- Los dispositivos se pueden dividir en particiones, que contienen volúmenes
- Los volúmenes pueden abarcar múltiples particiones.
- Cada volumen generalmente formateado en un sistema de archivos

# UNA ORGANIZACIÓN TÍPICA DE UN SISTEMA DE ARCHIVOS



# ESTRUCTURA ALMACENAMIENTO SOLARIS

/	ufs
/devices	devfs
/dev	dev
/system/contract	ctfs
<b>/proc</b>	<b>proc</b>
/etc/mnttab	mntfs
/etc/svc/volatile	tmpfs
/system/object	objfs
/lib/libc.so.1	lofs
/dev/fd	fd
/var	ufs
/tmp	tmpfs
/var/run	tmpfs
/opt	ufs
/zpbge	zfs
/zpbge/backup	zfs
/export/home	zfs
/var/mail	zfs
/var/spool/mqueue	zfs
/zpbg	zfs
/zpbg/zones	zfs

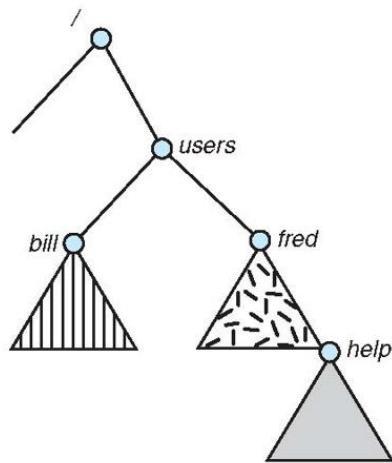
En un sistema operativo puede mantener múltiples sistemas de archivos

Sistema de archivo virtual con información de todos los procesos

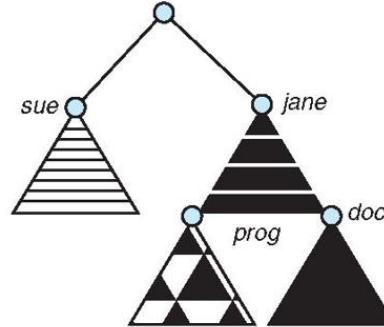
Sistemas de archivo de propósito general

# MONTAJE DE SISTEMA DE ARCHIVOS (MOUNTING)

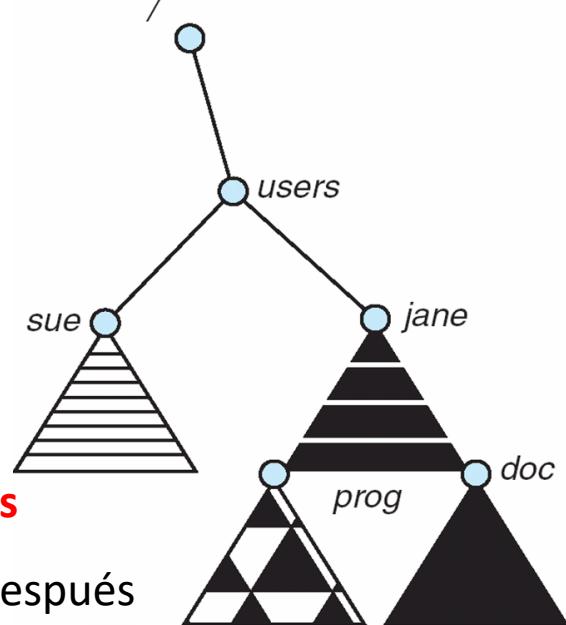
- Un sistema de archivos debe ser **montado** antes de que pueda ser accedido.
- Un sistema de archivos sin montar es montado en un **PUNTO DE MONTAJE (MOUNT POINT)**.



Antes



Montaje del  
Sistema de Archivos



Después

# ARCHIVOS COMPARTIDOS

- La acción de compartir debe ser hecha por medio de un esquema de *protección*.
- En sistemas distribuidos los archivos pueden ser compartidos a través de la red.
- Network File System (NFS) es un método común de compartir archivos distribuidos.
- Los **User IDs** identifican usuarios, admiten permisos y protección por usuarios.
- Los **Group IDs** admite agrupar usuarios en grupos, permitiendo asignar al mismo derechos de acceso.

# ARCHIVOS COMPARTIDOS – SEMÁNTICA DE CONSISTENCIA

- La **semántica de consistencia** especifica como múltiples usuarios acceden a un archivo compartido simultáneamente
  - **SEMÁNTICA UNIX.** El sistema de archivos Unix (UFS) implementa:
    - Las escrituras a un archivo abierto son visibles inmediatamente a los otros usuarios que comparten el mismo archivo abierto
    - El puntero a archivos compartidos permite que múltiples usuarios lean y escriban concurrentemente
  - **SEMÁNTICA DE SESIÓN.** AFS tiene una semántica de sesión
    - Las escrituras son solo visibles solo después que la sesión termina.
  - **SEMÁNTICA DE ARCHIVOS COMPARTIDOS INMUTABLES**

# SISTEMAS DE ARCHIVOS IMPLEMENTACIÓN

---

# OBJETIVOS

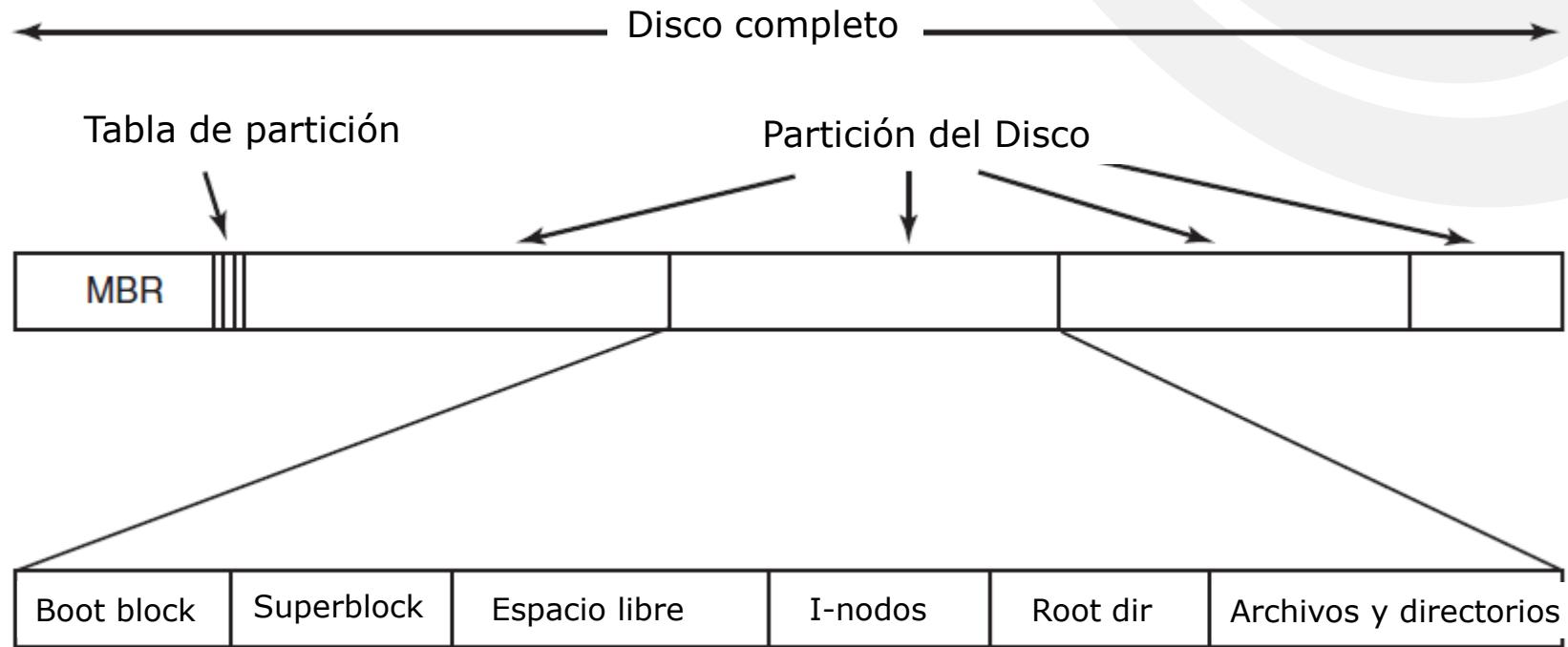
- Describir los detalles locales de la implementación del sistema de archivos y estructuras de directorios
- Discutir algoritmos de alocación de bloques y bloques libres y compromisos

# DISEÑO DEL SISTEMA DE ARCHIVOS

- El sistema de archivos se almacena en disco.
- Los discos pueden dividirse en varias particiones.
- MBR (Master Boot Record) se encuentra en el sector 0 del disco.
- Tabla de particiones.
- En el almacenamiento, el sistema de archivos puede contener las siguientes estructuras:
  - Bloque de control de booteo (Boot Control Block por volumen), ubicado típicamente en el primer bloque. En UFS: boot block, en NTFS: sector de partición de booteo (partition boot sector)
  - Bloque de control de volumen (por volumen). In UFS: superbloque y en NTFS: almacenado en el master file table.
  - Una estructura de directorio (por cada sistema de archivos)
  - Para cada archivo una estructura de FCB

# DISEÑO DEL SISTEMA DE ARCHIVOS

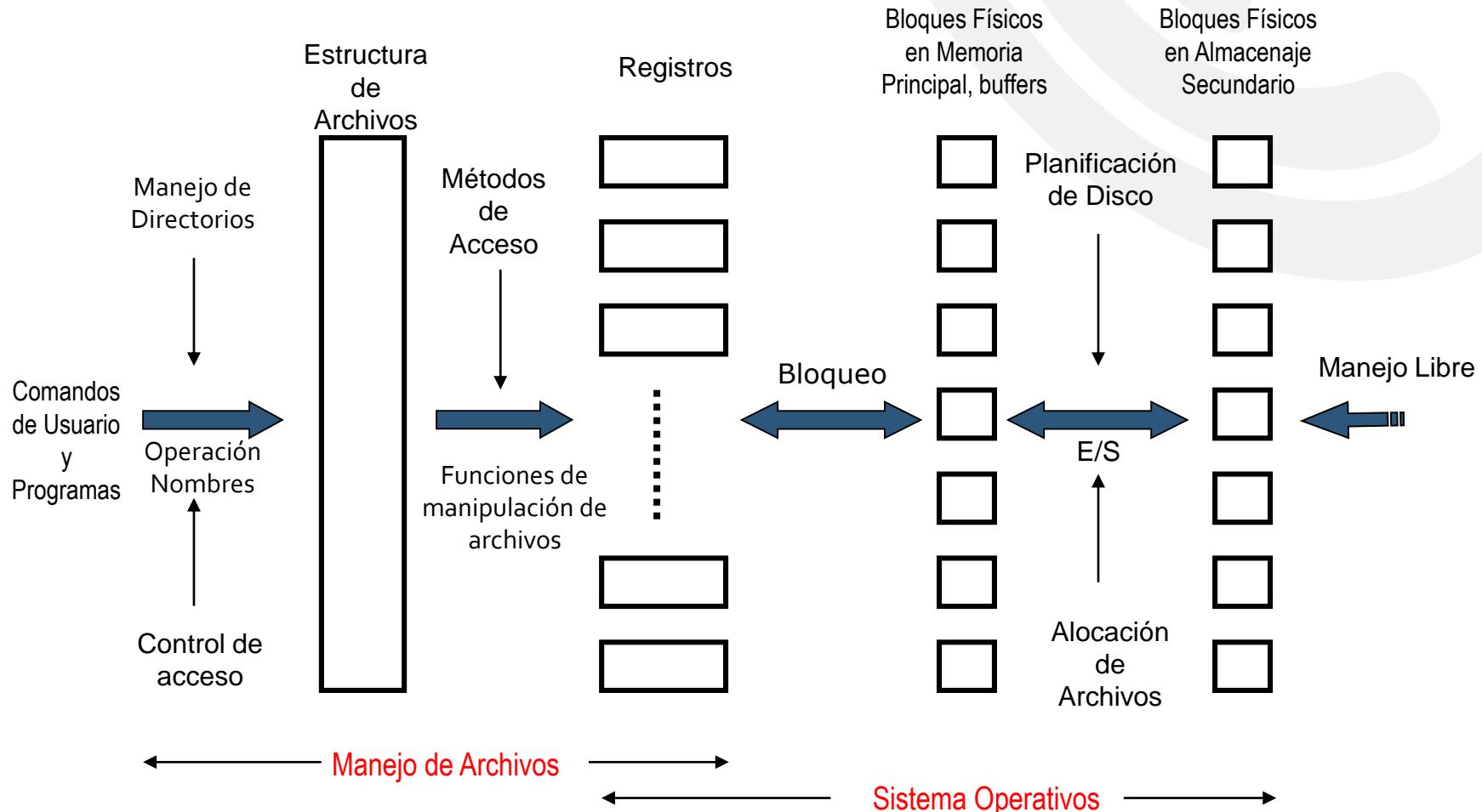
## Ejemplo de un posible diseño



# ESTRUCTURA DEL SISTEMA DE ARCHIVOS

- Estructura de Archivo
  - Unidad Lógica de almacenamiento
  - Colección de información relacionada
- El sistema de archivos reside en almacenamiento secundario (discos).
- El sistema de archivo está organizado en capas.
- **FILE CONTROL BLOCK (FCB)** – estructura de almacenaje consistente de información sobre el archivo.

# SISTEMA DE ARCHIVOS EN GENERAL



# UN BLOQUE DE CONTROL DE ARCHIVOS TÍPICO

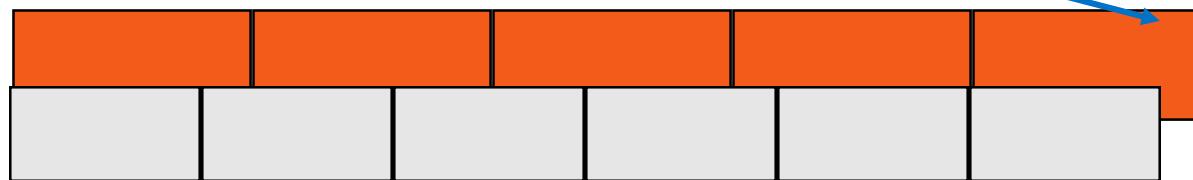
## FCB: File Control Block

file permissions
file dates (create, access, write)
file owner, group, ACL
file size
file data blocks or pointers to file data blocks

# ESTRUCTURAS DE ARCHIVO

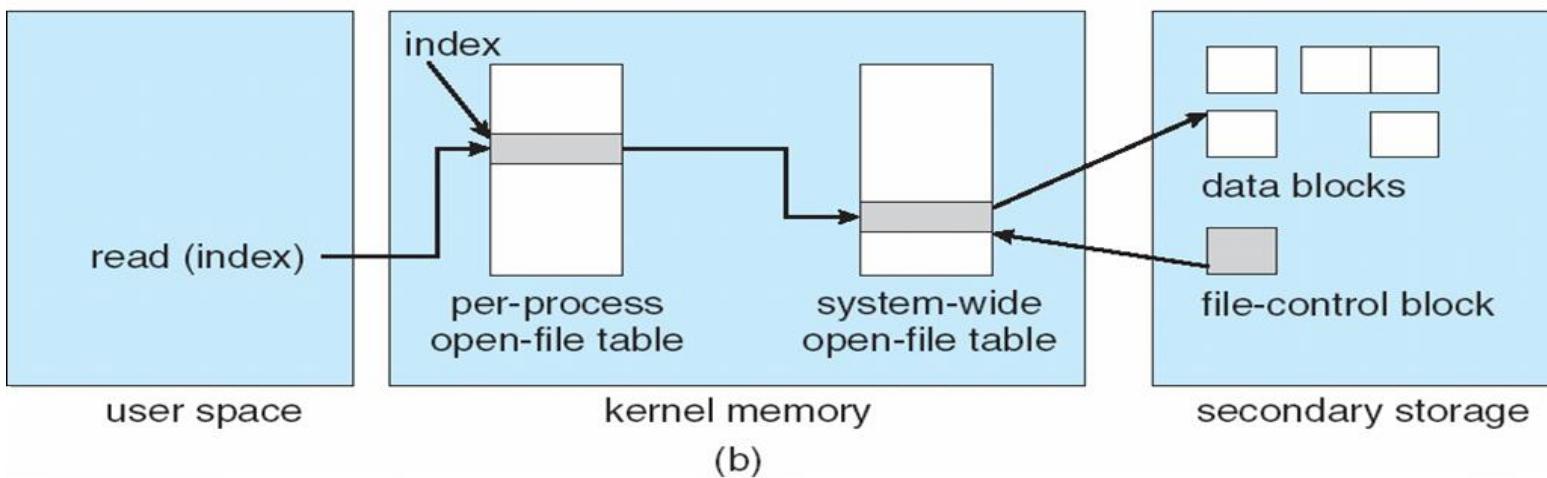
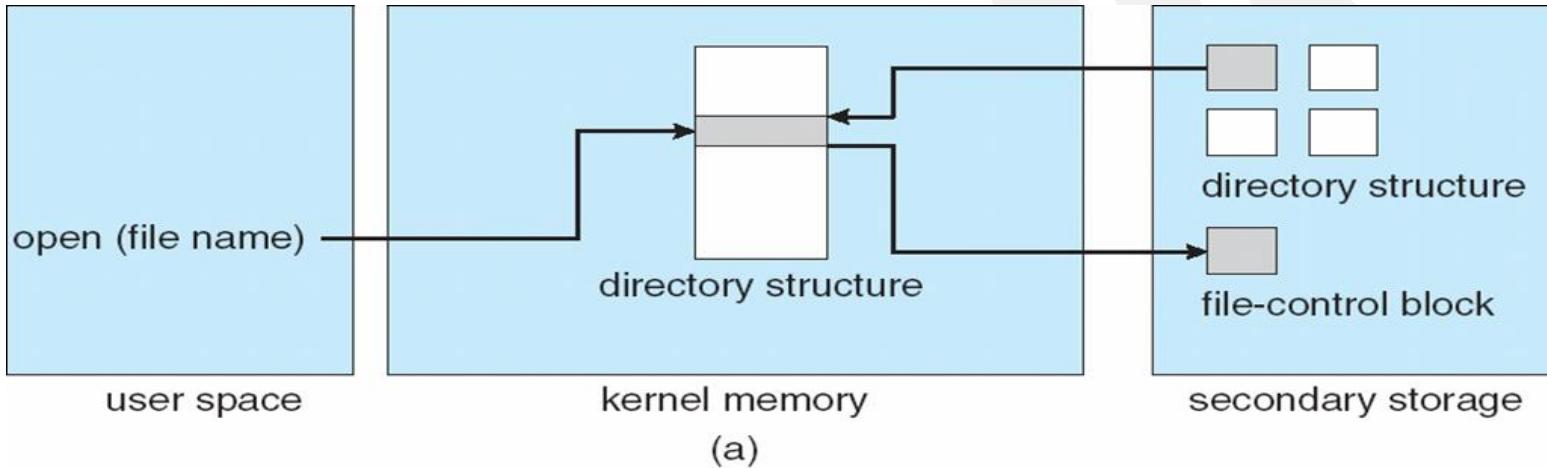
- Bloques Físicos
- Registros Lógicos
- Fragmentación

Bloques  
físicos



Registros lógicos

# ESTRUCTURAS DEL SISTEMA DE ARCHIVOS EN MEMORIA



## a) Apertura de un archivo

## b) Lectura de un archivo

# IMPLEMENTACIÓN DE DIRECTORIO

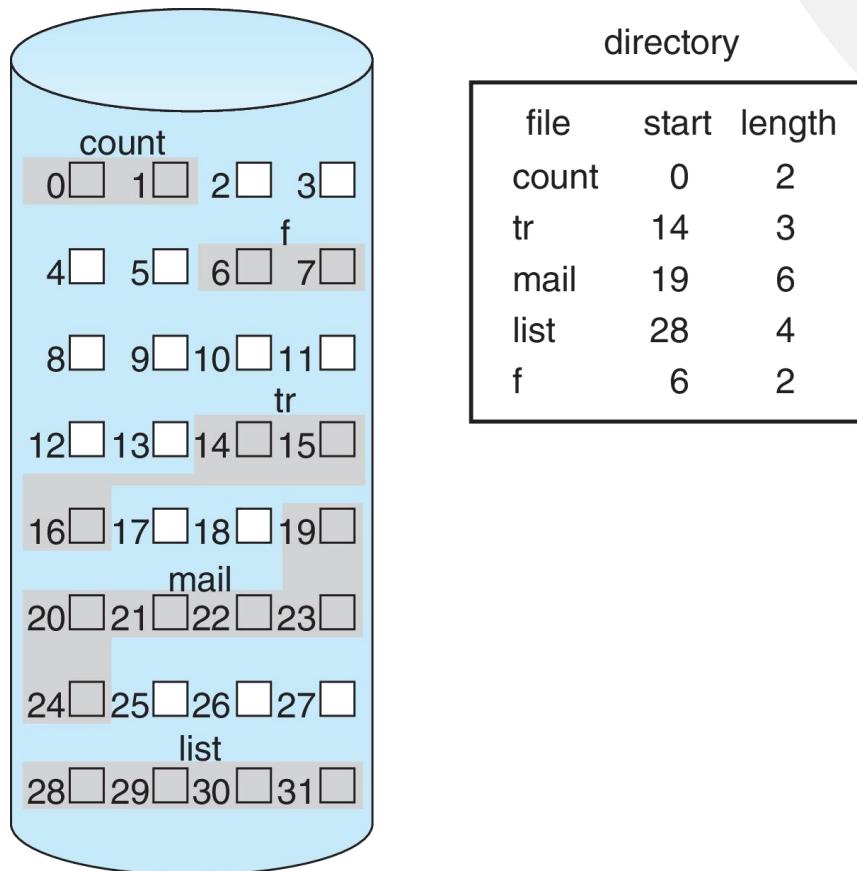
- Lista lineal de nombres de archivos con punteros a los bloques de datos.
- Tabla *hash* – Lista lineal con estructura de datos *hash*.

# MÉTODOS DE ALOCACIÓN

Un método de alocación se refiere a cómo los bloques de disco de un archivo son ubicados:

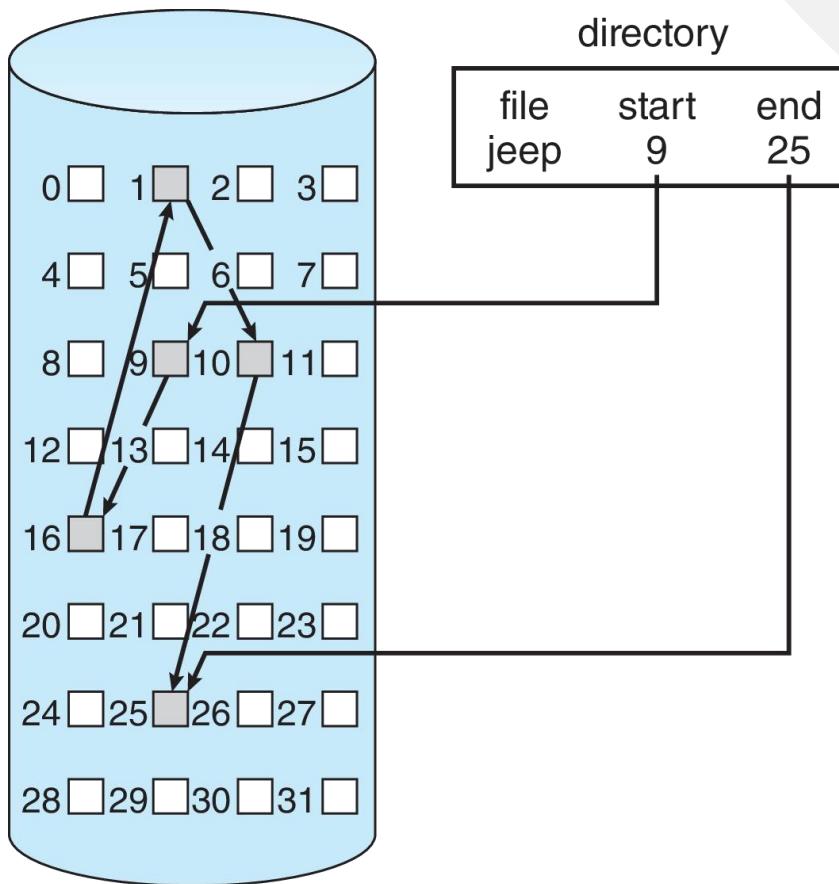
- Alocación Contigua
- Alocación Enlazada
- Alocación Indexada

# ALOCACIÓN CONTIGUA



# ALOCACIÓN ENLAZADA

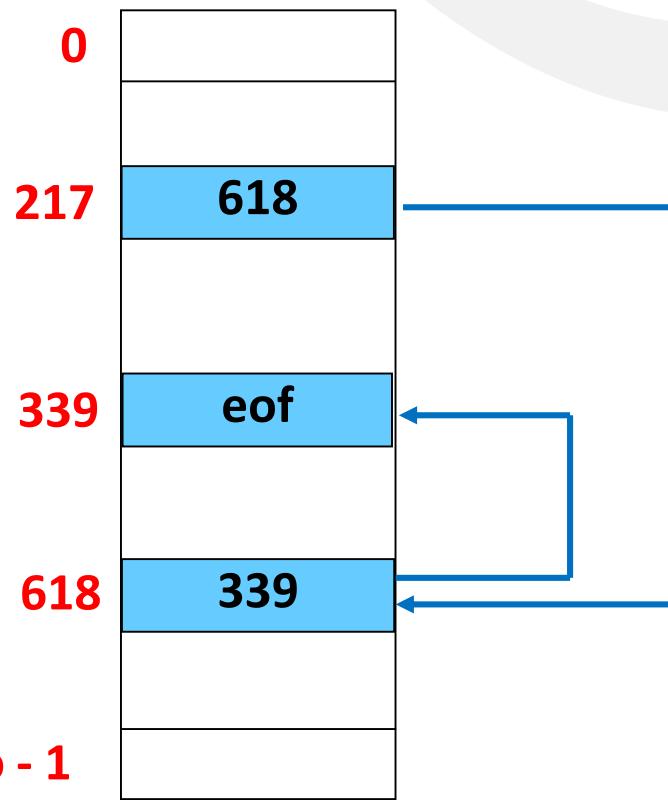
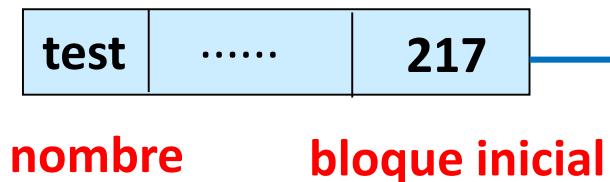
- Cada archivo es una lista enlazada de bloques de disco: los bloques pueden estar en cualquier lugar del disco.



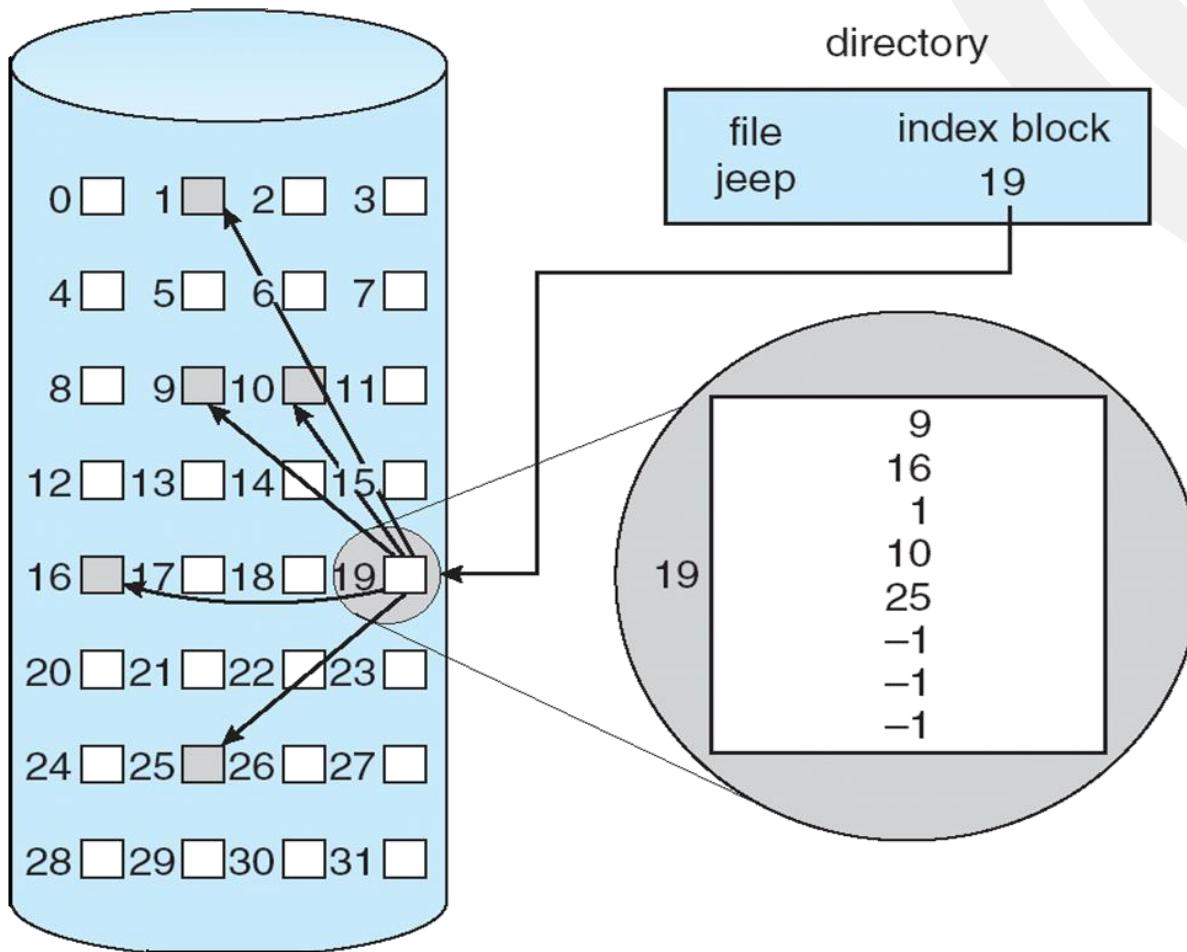
# ALOCACIÓN ENLAZADA

- *File-Allocation Table (FAT)* – alocación de espacio de disco usado en MS-DOS y OS/2.

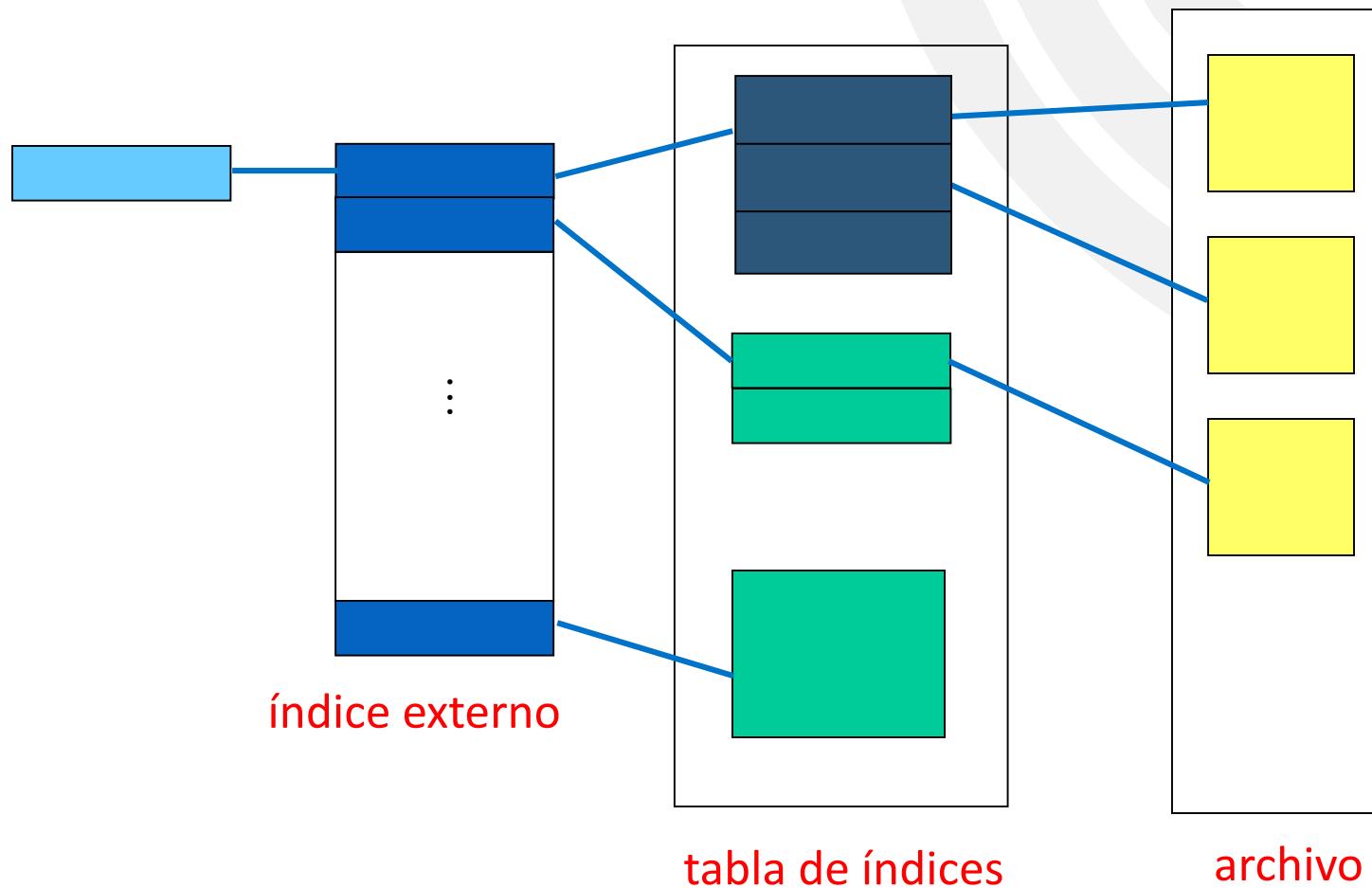
entrada de directorio



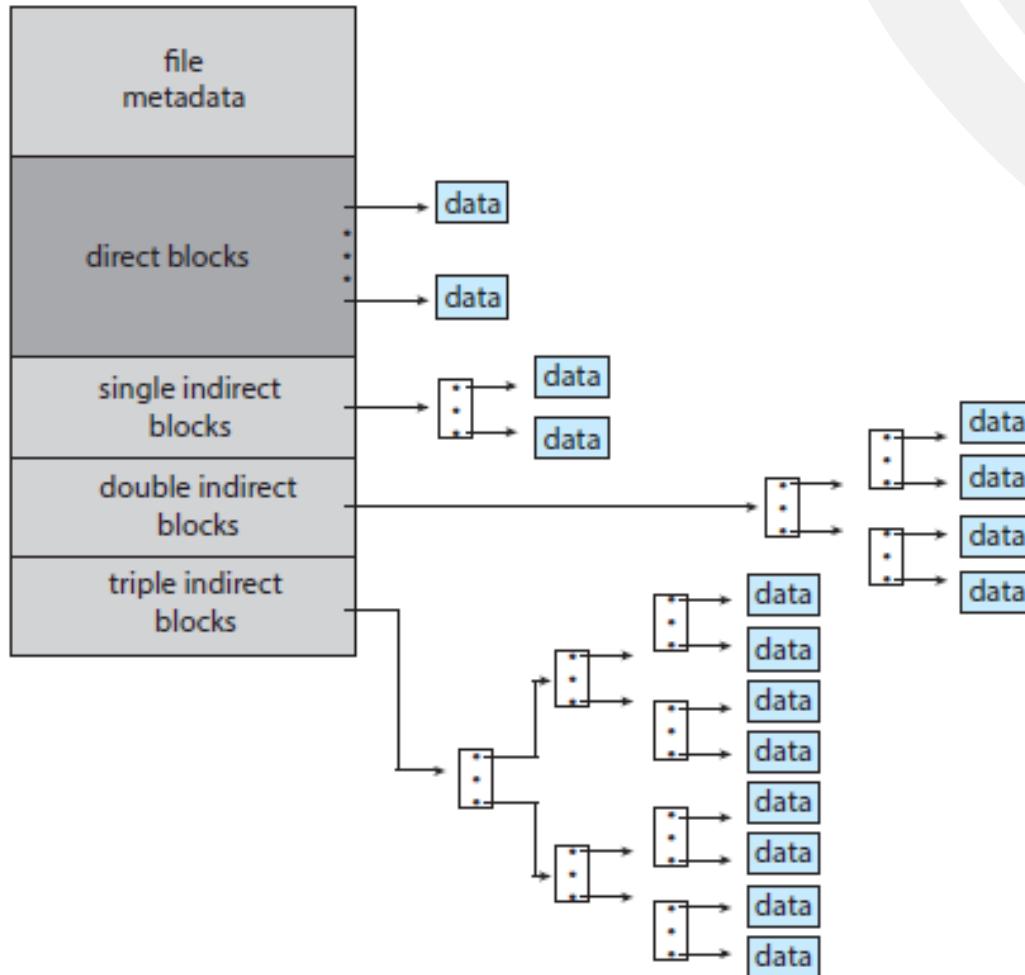
# ALOCACIÓN INDEXADA



# ALOCACIÓN INDEXADA – MAPEO



# ESQUEMA COMBINADO: UNIX (4K BYTES POR BLOQUE)



# ESQUEMA COMBINADO: UNIX

Los pasos para buscar un archivo */usr/ast/mbox*.

1	.
1	..
4	bin
7	dev
14	lib
9	etc
6	usr
8	tmp

Looking up  
usr yields  
i-node 6

Mode	size	times
132		

I-node 6  
says that  
*/usr* is in  
block 132

6	*
1	..
19	dick
30	erik
51	jim
26	ast
45	bal

*/usr/ast*  
is i-node  
26

Block 132  
is */usr*  
directory

Mode	size	times
406		

I-node 26  
says that  
*/usr/ast* is in  
block 406

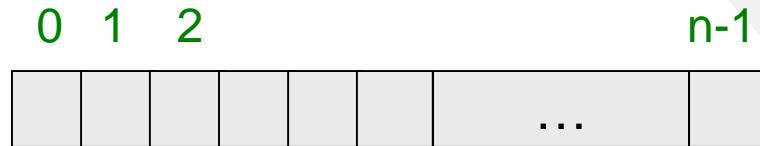
I-node 26  
is for  
*/usr/ast*

26	*
6	..
64	grants
92	books
60	mbox
81	minix
17	src

*/usr/ast/mbox*  
is i-node  
60

# ADMINISTRACIÓN DE ESPACIO LIBRE

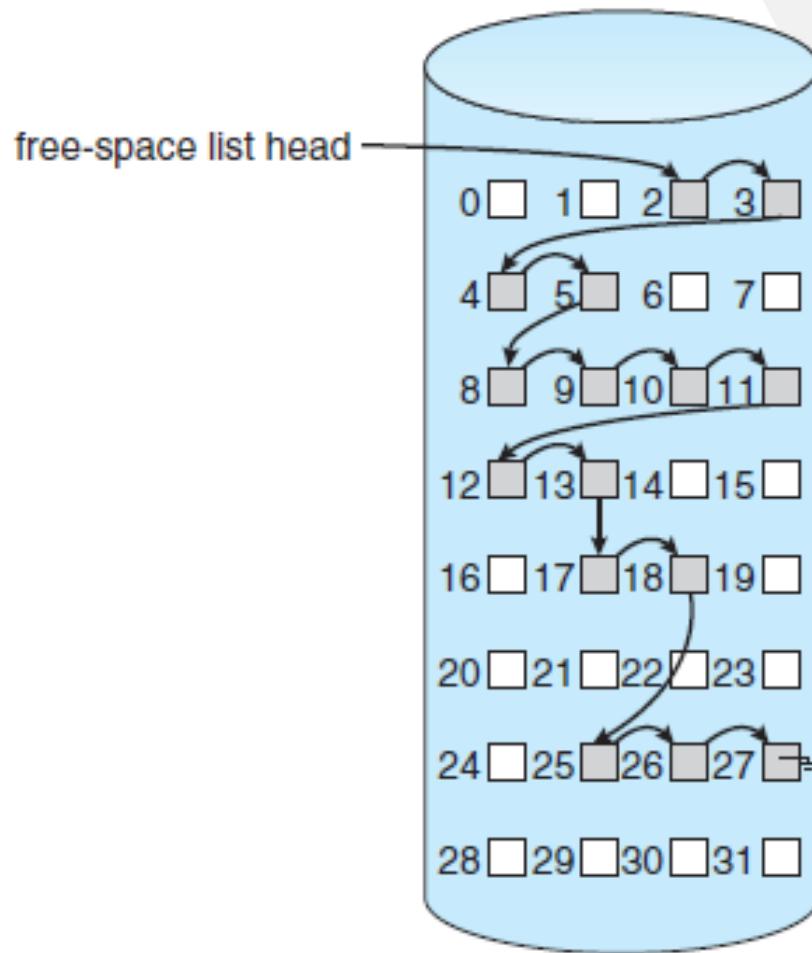
- Vector de Bits – bit map ( $n$  bloques)


$$\text{bit}[i] = \begin{cases} 1 \Rightarrow \text{bloque}[i] \text{ libre} \\ 0 \Rightarrow \text{bloque}[i] \text{ ocupado} \end{cases}$$

- Cálculo del número de bloque

(número de bits por palabra) \*  
(número de palabras con valor 0) +  
offset del primer bit 1

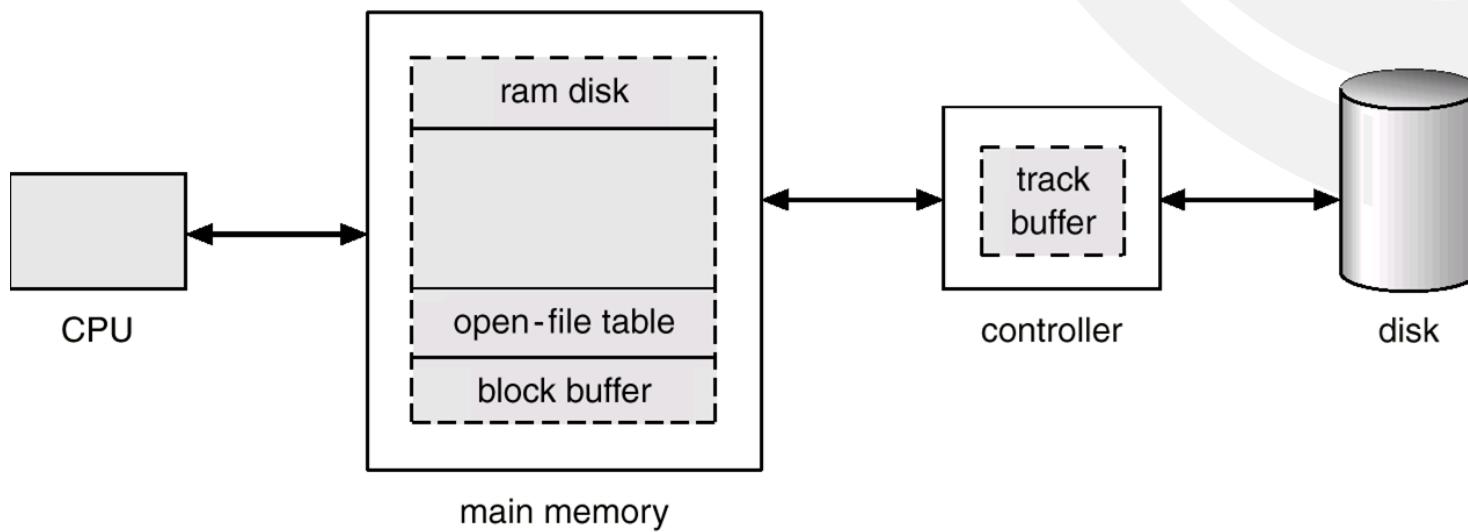
# ADMINISTRACIÓN DE ESPACIO LIBRE



# EFICIENCIA Y DESEMPEÑO

- La eficiencia depende de:
  - alocacion en el disco y algoritmos de directorio
  - tipos de datos mantenidos en la entrada de directorio del archivos
- Desempeño
  - caché de disco – sección separada de memoria principal para bloques frecuentemente usados
  - *free-behind* y *read-ahead* – técnicas para optimizar el acceso secuencial
  - mejora del desempeño de la PC dedicando una sección de la memoria como disco virtual, o disco RAM.

# DISTINTAS LOCACIONES DEL CACHE DE DISCO

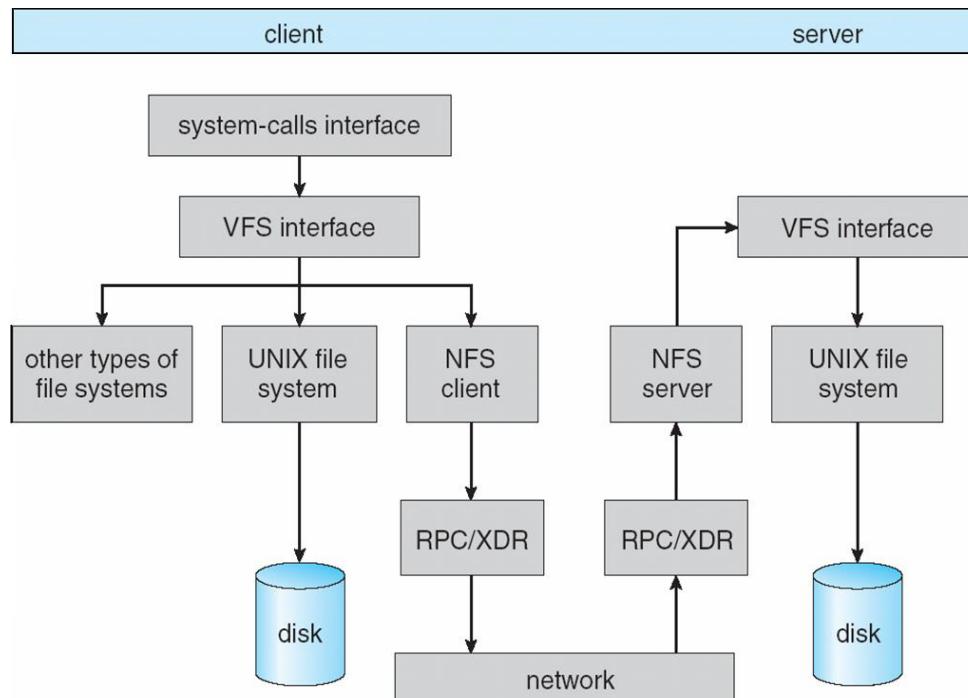


# RECUPERACIÓN

- Verificador de Consistencia – compara datos en la estructura de directorio con bloques de datos en el disco, y trata de reparar inconsistencias.
- Uso de programas de sistema para **respaldar** (*back up*) datos del disco a otro dispositivo de almacenaje (cinta magnética, optical, etc).
- Se recuperan archivos perdidos o disco por **restauración** de datos del backup.

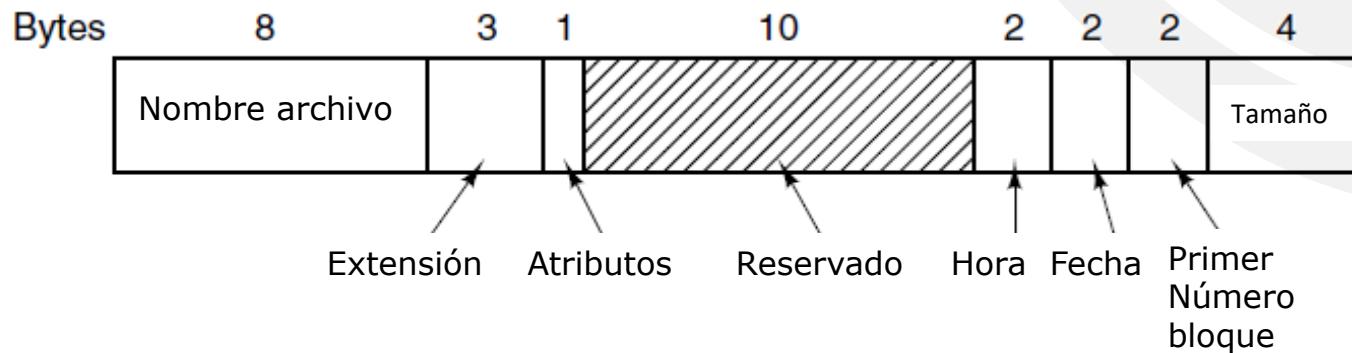
# EL SISTEMA DE ARCHIVOS DE RED DE SUN (NFS)

- Es una implementación y una especificación de un sistema de software para acceder a archivos remotos a través de LANs (o WANs).
- La implementación es parte de los sistemas operativos Solaris y SunOS que corre sobre estaciones de trabajo Sun usando un protocolo no confiable datagrama (protocolo UDP/IP) y Ethernet.



# SISTEMA DE ARCHIVO – MS-DOS

- Entrada de directorio



Sistema de Archivos FAT tiene tres versiones:

- ✓ FAT-12
- ✓ FAT-16
- ✓ FAT -32

# SISTEMA DE ARCHIVO – MS-DOS

- Máximo tamaño de la partición para diferentes tamaños de bloques

Tam. Bloque	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

## Bibliografía:

- Silberschatz, A., Gagne G., y Galvin, P.B.; "*Operating System Concepts*", 7<sup>ma</sup> Edición 2009, 9<sup>na</sup> Edición 2012, 10<sup>ma</sup> Edición 2018.
- Tanenbaum, A.; "*Modern Operating Systems*", Addison-Wesley, 3<sup>ra</sup> Edición 2008, 4<sup>ta</sup>. Edición 2014.
- Stallings, W. "*Operating Systems: Internals and Design Principles*", Prentice Hall, 6<sup>ta</sup> Edición 2009, 7<sup>ma</sup> Edición 2011, 9<sup>na</sup> Edición 2018.