GESTION DE FICHEROS (I)

Sistema de ficheros

- Un sistema de ficheros es un conjunto de software de sistema que proporciona servicios a los usuarios y aplicaciones en el uso de ficheros.
- Proporciona al sistema una forma consistente y bien definida de controlar el recurso de los ficheros.
- Cada sistema operativo posee su propia organización lógica del disco para poder almacenar la información, y la usará normalmente, pero además puede tener la posibilidad de usar particiones propias de otros sistemas.

Ejemplos de tipos de sistema de ficheros

- **ext2**: linux nativo. Soporta características avanzadas: propietarios, permisos, enlaces, etc.
- **ext3**: linux nativo con *journaling*. Similar a ext2 pero con transacciones (registro por diario) para evitar que apagados accidentales puedan deteriorar el sistema de ficheros.
- **msdos**: diseñado para un sistema monousuario. En la actualidad sólo se utiliza en ciertos dispositivos como cámaras digitales debido a su limitación en el nombre de ficheros (8 caracteres. + 3 de extensión)
- **VFAT** (FAT12, FAT16 y FAT32): ampliación de msdos, con soporte para nombres largos de ficheros. Son monousuario y los permisos son muy limitados.
- **NTFS**: sistema de ficheros de Windows NT/XP. Es un sistema de ficheros con características avanzadas y es multiusuario.
- **iso9660**: es el sistema de ficheros de los CDs.

Problemas de diseño

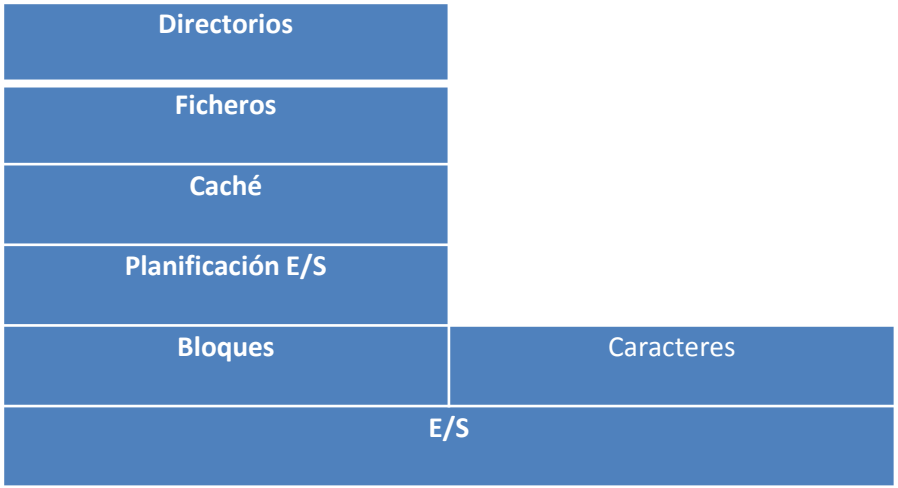
- Definir cómo debe ver el usuario el sistema de ficheros
  - Definir un fichero y sus atributos
    - Nombre (cadena de caracteres)
    - Tipo de archivo (necesario en sistemas que reconocen distintos tipos)
    - Ubicación en el dispositivo
    - Tamaño
    - Información de protección
    - Fechas, horas e identificación del usuario

Problemas de diseño

- Definir las operaciones permitidas sobre un fichero
  - Abrir / Cerrar
  - Crear / Eliminar
  - Leer/ Escribir
  - Posicionar un puntero
  - Cambiar atributos
- Definir la estructura de directorios

- Definir los algoritmos y estructuras de datos que deben crearse para establecer la correspondencia entre el sistema de ficheros lógico y los dispositivos físicos donde se almacenan

Arquitectura de un sistema de ficheros



- Con el objetivo de aumentar la eficiencia E/S las transferencias entre la memoria y el disco se efectúan en unidades de **bloques** (uno o más sectores).

Funciones de un sistema de ficheros

- Identificar y localizar un fichero seleccionado
- Usar un directorio para describir la ubicación de todos los ficheros y sus atributos
- En un sistema compartido, describir el control de acceso de usuario
- Convertir en bloques los registros de un fichero
- Asignar ficheros a bloques libres de almacenamiento secundario
- Conocer qué bloques de almacenamiento secundarios están disponibles para nuevos ficheros y cuáles utilizados en ficheros existentes

Métodos de asignación de ficheros

- Asignación contigua
- Asignación enlazada
- Asignación indexada

Métodos de asignación de ficheros

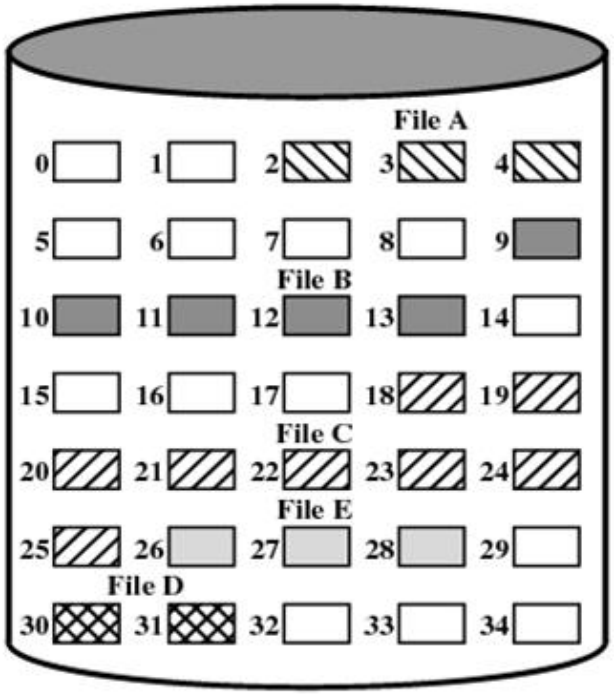
- **Asignación contigua**
- Asignación enlazada
- Asignación indexada

Métodos de asignación de ficheros

- Asignación contigua
  - Se asigna un único conjunto contiguo de bloques en tiempo de creación de los ficheros.
  - Se realiza una pre-asignación que utiliza porciones de tamaño variable.
  - Se usa en CD-ROMs y DVDs.

Métodos de asignación de ficheros

- Asignación contigua (cont.)



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

Siendo DL la dirección lógica (offset), y TB el tamaño de bloque:

- bloque a acceder = DL/TB + bloque inicial
- desplazamiento dentro del bloque = DL%TB

Métodos de asignación de ficheros

- Asignación contigua (cont.)
  - **Ventajas:**
    - Sencillo: La tabla de asignación de ficheros contiene sólo una entrada para cada fichero indicando el bloque inicial y la longitud del fichero.
    - Adecuado tanto para el acceso secuencial como para el directo.
  - **Desventajas:**
    - Es necesario declarar el tamaño del fichero en el tiempo de creación.
    - Fragmentación externa (derroche de espacio, dificultad de encontrar bloques contiguos de espacio de suficiente longitud).
    - Requiere compactación para liberar espacio.

Métodos de asignación de ficheros

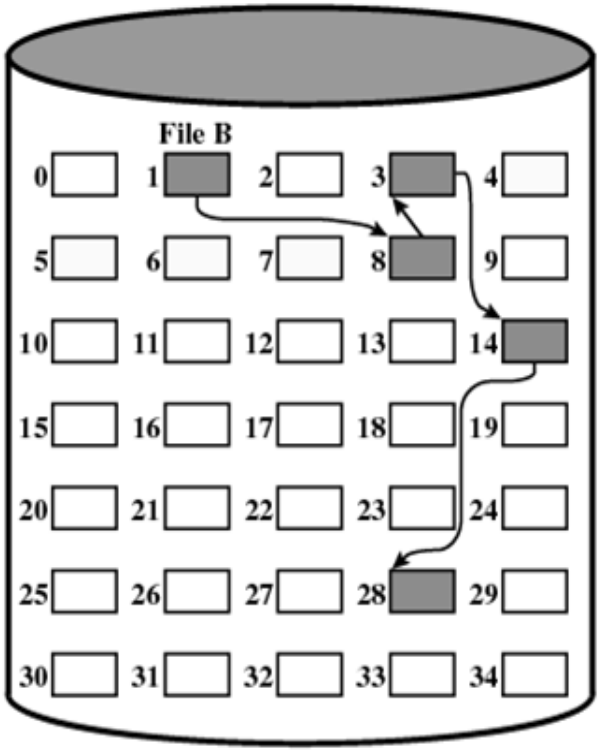
- Asignación contigua
- **Asignación enlazada**
- Asignación indexada

Métodos de asignación de ficheros

- Asignación enlazada
  - La asignación se realiza a nivel de bloques individuales. Los bloques pueden estar dispersos en el disco.
  - Cada bloque contiene un puntero al siguiente bloque en la cadena.

Métodos de asignación de ficheros

- Asignación enlazada (cont.)



File Allocation Table

File Name	Start Block	Length
...	...	...
File B	1	5
...	...	...

Métodos de asignación de ficheros

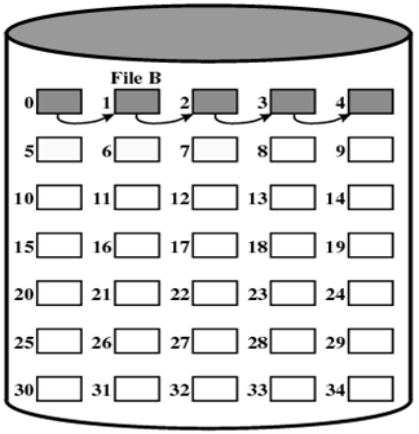
- Asignación enlazada (cont.)
  - **Ventajas:**
    - Sencillo: La tabla de asignación de ficheros contiene sólo una entrada para cada fichero indicando el bloque inicial y la longitud del fichero (o el bloque final).
    - Se pueden asignar bloques a medida que se necesitan, no requiere pre-asignación.
    - Evita la fragmentación externa.
    - Los ficheros pueden crecer mientras exista espacio en el disco.

Métodos de asignación de ficheros

- Asignación enlazada (cont.)
  - **Desventajas:**
    - Aunque es muy adecuado para el acceso secuencial, no lo es para el acceso directo ya que hay que leer todos los bloques para recorrer la cadena de enlaces hasta el destino.
    - Se requiere un espacio adicional para los punteros de enlace.
      - Solución: agrupación de bloques.
    - Problema fiabilidad, la pérdida de un bloque de fichero supone la pérdida de todos los datos del fichero que van detrás de dicho bloque.
      - Solución: lista doblemente enlazada (overhead)

Métodos de asignación de ficheros

- Asignación enlazada (cont.)
  - **Desventajas:**
    - No existe principio de proximidad (si es necesario traer varios bloques de fichero a la vez, se requiere una serie de accesos a diferentes partes del disco).
      - Solución: consolidar ficheros periódicamente:

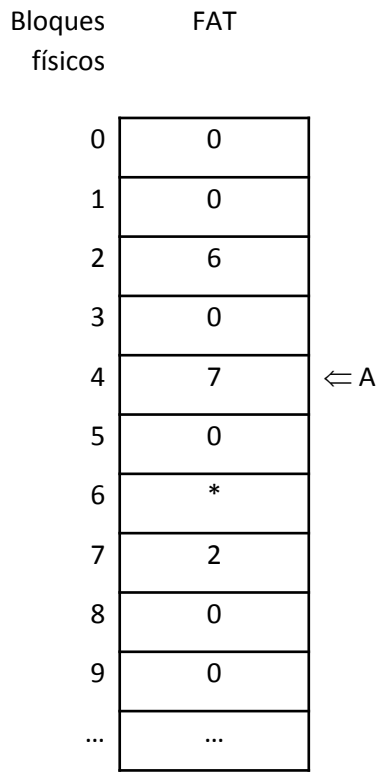


File Allocation Table

File Name	Start Block	Length
...	...	...
File B	0	5
...	...	...

Métodos de asignación de ficheros

- Variación del método enlazado: **Tabla de Asignación de Archivos (FAT)**
  - Los punteros de los bloques de ficheros se guardan en una tabla (FAT) que contiene una entrada por cada bloque del disco y está indexada por nº de bloque de disco
  - La cadena continúa hasta el último bloque, que tiene un valor especial de fin de archivo como entrada de la tabla.
  - Los bloques desocupados se indican con valor cero en la tabla



Métodos de asignación de ficheros

- Variación del método enlazado: **Tabla de Asignación de Archivos (FAT)**
  - Para localizar un bloque sólo se necesita leer en la FAT
  - Se reserva una sección del disco al comienzo de la partición para la FAT.
  - Hay versiones de punteros de 16 bits (FAT16) y de 32 bits (FAT32).
    - Nº máx. ficheros FAT16 (2^{16}): 65.536
    - Nº máx. ficheros FAT32 (2^{28}): 268.435.456

Métodos de asignación de ficheros

- Variación del método enlazado: **Tabla de Asignación de Archivos (FAT)**
  - Limitación para gestionar archivos grandes.
    - Una posible solución es aumentar el tamaño de bloque.
  - Simple y eficiente si está en caché
  - Problema fiabilidad.
    - Solución: doble copia de la FAT
  - Utilizado antiguamente en Windows y actualmente en memorias flash

Métodos de asignación de ficheros

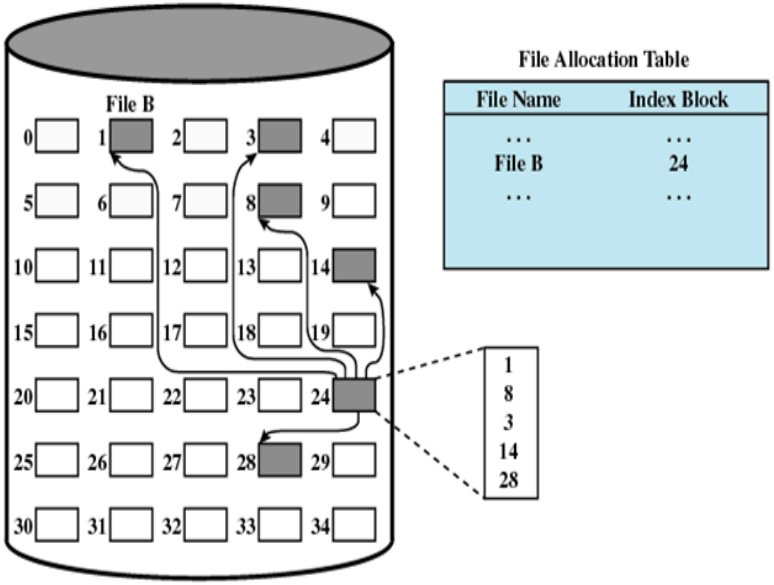
- Asignación contigua
- Asignación enlazada
- **Asignación indexada**

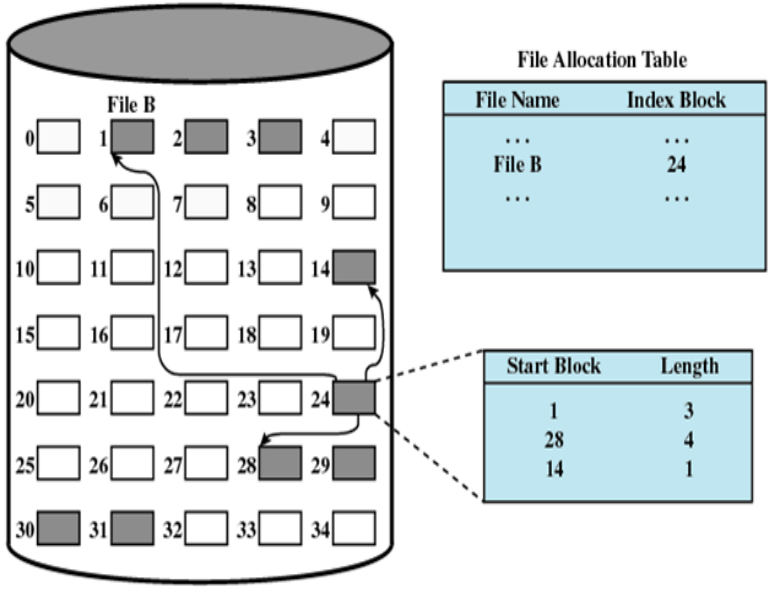
Métodos de asignación de ficheros

- Asignación indexada
  - Hay un índice de punteros por cada fichero.
  - El índice tiene una entrada por cada porción asignada al fichero y se guarda en un bloque separado.
  - La tabla de asignación de ficheros contiene para cada archivo la localización de su índice.
  - La asignación puede realizarse mediante bloques de tamaño fijo (para leer el i-ésimo bloque buscamos el puntero en la i-ésima entrada del bloque índice) o variable.

Métodos de asignación de ficheros

- Asignación indexada (cont.)





Métodos de asignación de ficheros

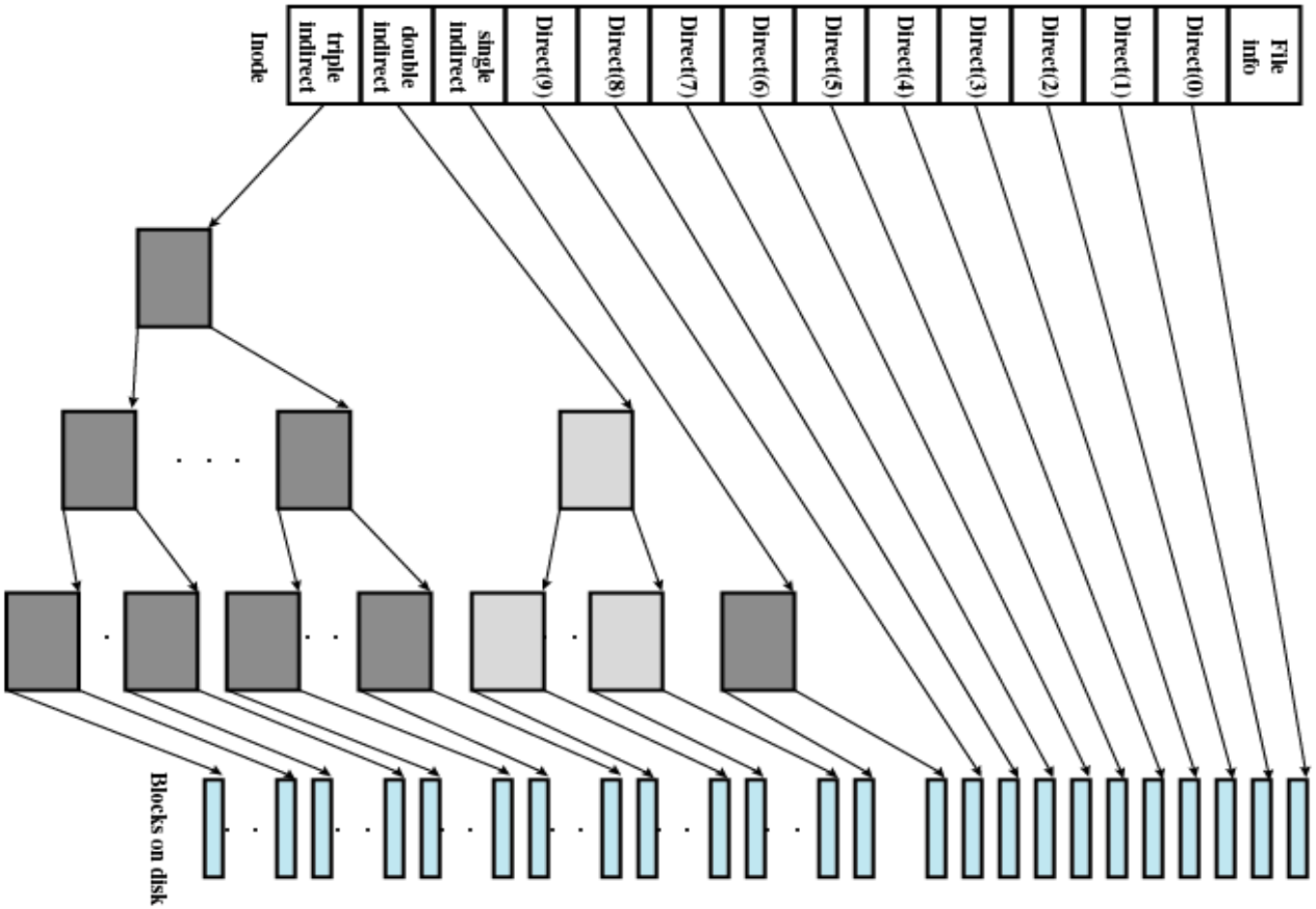
- Asignación indexada (cont.)
  - **Ventajas:**
    - Buen acceso directo y secuencial.
    - La asignación por bloques no produce fragmentación externa.
    - La asignación por porciones variables mejora la proximidad.
  - **Desventajas:**
    - Posible desperdicio de espacio en los bloques índices.
    - Si es el bloque índice es demasiado pequeño (normalmente ocupa un bloque de disco), no podrá contener suficientes punteros como requiere un archivo grande.

Métodos de asignación de ficheros

- Asignación indexada (cont.)
  - Soluciones:
    - Esquema enlazado: Para manejar archivos grandes podríamos enlazar varios bloques índice.
    - Índice multinivel: Usar un bloque índice de primer nivel que apunte a un conjunto de bloques índice de segundo nivel.
    - Esquema combinado (UNIX): Guardamos los primeros punteros del bloque índice en el bloque índice del archivo. Los punteros apuntarán a bloques directos, bloques indirectos simples, bloques indirectos dobles, ...

Métodos de asignación de ficheros

- Asignación indexada (cont.)



Gestión del espacio libre

- El sistema mantiene una lista de los bloques que están libres.
- La FAT no necesita ningún método.
- Mapa o vector de bits
  - Cada bloque se representa con un bit (0: libre, 1: ocupado)
  - Fácil encontrar un bloque libre o n bloques libres consecutivos
  - Fácil tener archivos en bloques contiguos
  - Ineficiente si no se mantiene en memoria principal

Gestión del espacio libre

- Lista de libres enlazada
  - Enlaza todos los bloques libres del disco, guarda un puntero al 1er bloque
  - No derrocha espacio
- Lista de secciones libres enlazadas
  - Puntero + tamaño
  - Obtener muchas direcciones de bloques libres es rápido
- Indexación (recuento)
  - Cada entrada de la lista: una dirección de bloque libre y un contador del nº de bloques libres que le sigue

Directorios

- **Operaciones:**
  - Buscar un archivo por nombre
  - Crear archivos
  - Borrar Archivos
  - Renombrar archivos
  - Listar el directorio
  - Recorrer el sistema de ficheros

Directorios

- **Contenido de una entrada de directorio:**
  - Nombre de archivo + atributos + dirección de los bloques de datos (MS-DOS).

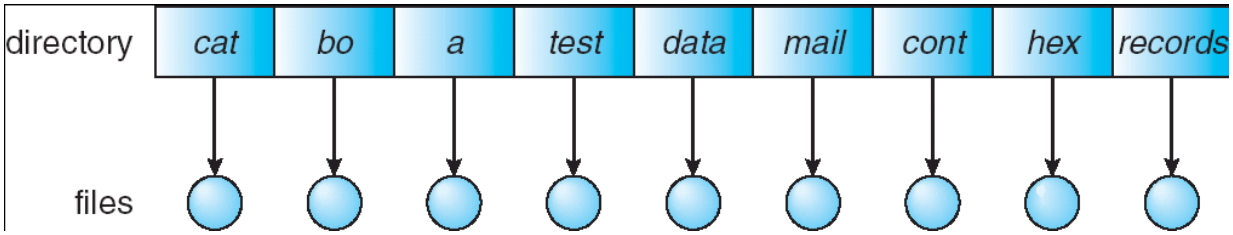
bytes	8	3	1	10	2	2	2	4
	Nombre de archivo	Tipo de archivo (extensión)	Atributos	reservado	Tiempo	Fecha	Nº 1er bloque	Tamaño

  - Nombre de archivo + puntero a una estructura de datos que contiene toda la información relativa al archivo (UNIX).

puntero iNodo	Nombre de archivo

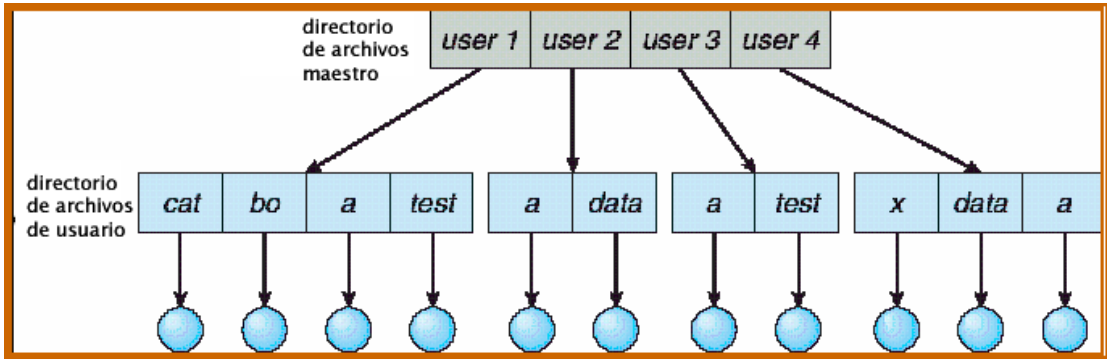
Directorios. Organización

- Único nivel (espacio plano)
  - Todos los archivos se guardan en el mismo directorio
  - Conflictos de nombres (al aumentar el nº de ficheros)
  - Mala organización en sistemas multiusuario
  - El tiempo de búsqueda tiende a aumentar



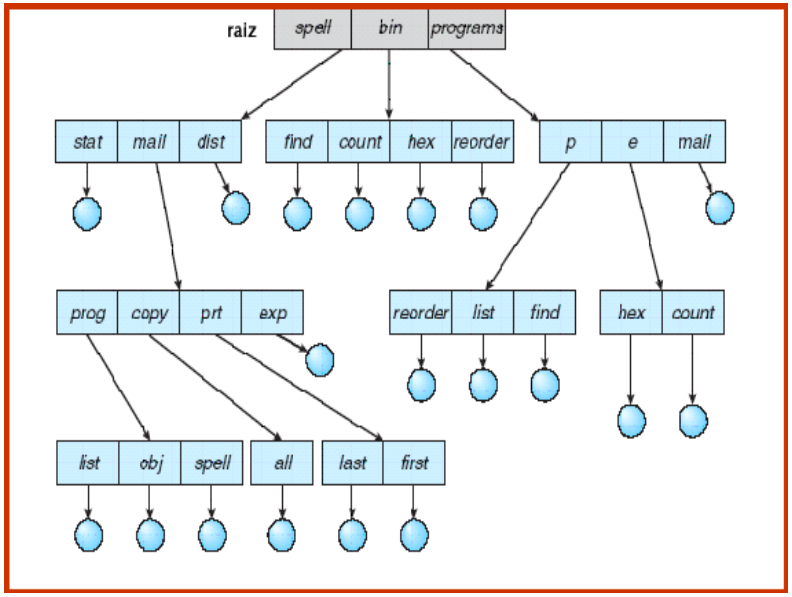
Directorios. Organización

- Dos niveles
  - Un directorio por usuario
  - Se resuelve el problema de conflictos de nombres (entre diferentes usuarios)
  - Problema: compartir información (algunos sistemas impiden el acceso a los directorios de otros usuarios)
  - Aparece el concepto de ruta (path)
  - Archivos de sistema $\Rightarrow$ usuario especial



Directorios. Organización

- Árbol
  - Altura arbitraria (subdirectorios)
  - Se reducen al mínimo los conflictos de nombres
  - Directorio ⇒ archivo que se trata de manera especial
  - Los nombres de los archivos son rutas sobre el árbol de directorios: rutas absolutas o relativas
  - Aparece el concepto de "directorio de trabajo" (CWD).
  - Política a seguir para la eliminación de directorios
  - "Caminos de búsqueda " (facilita el hecho de que varios usuarios quieran compartir ficheros)



Directorios. Organización

- Grafo (enlaces)
  - Permite a los directorios tener subdirectorios y archivos compartidos
  - Enlace o link (es un puntero a otro archivo o directorio). Se crea una nueva entrada en el directorio y se copia la dirección de la estructura de datos con la información del archivo. Requiere un contador de enlaces.