

# Sistemas Operativos

## Gestión de la Entrada/Salida

[SGG]: Cap. 10 al 13

[Sta]: Cap 10 a 12

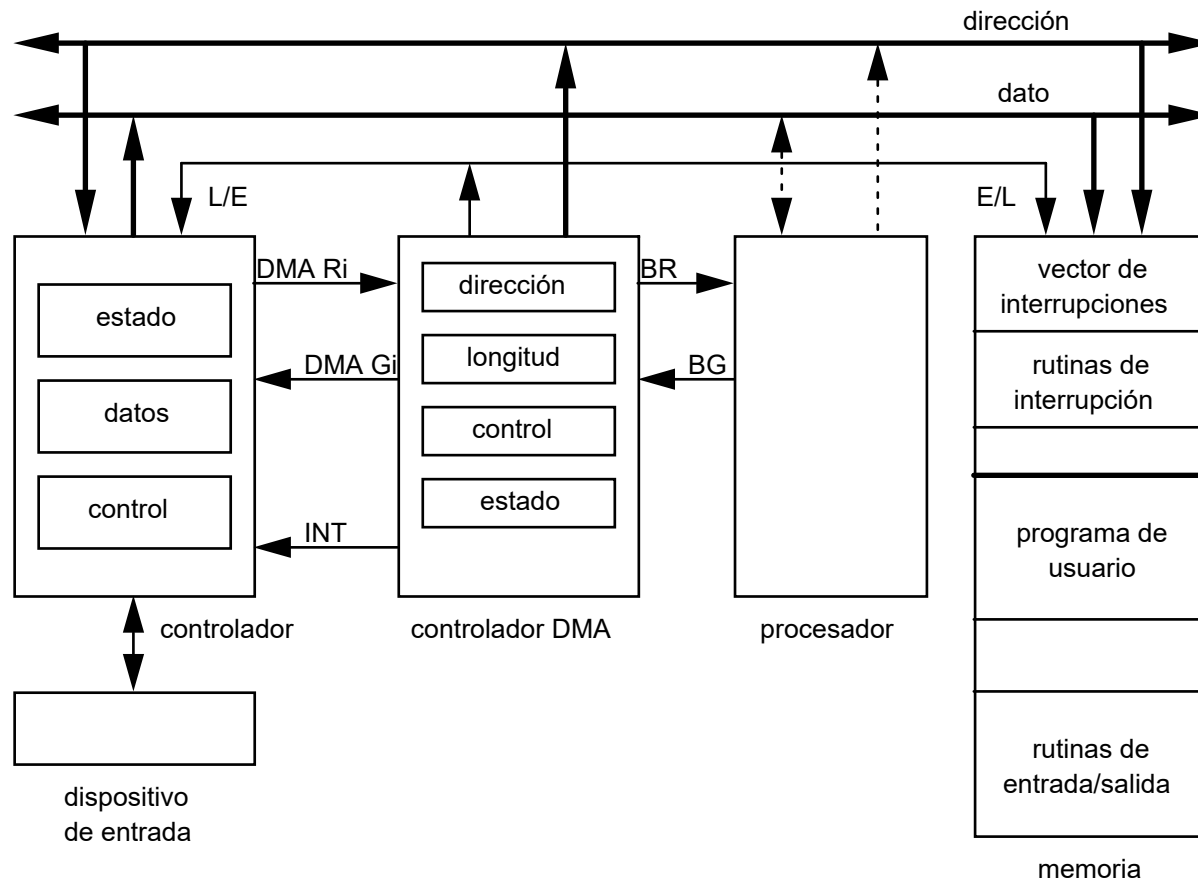
[Ste05]: Cap. 3 a 5



# Gestión de la entrada/salida

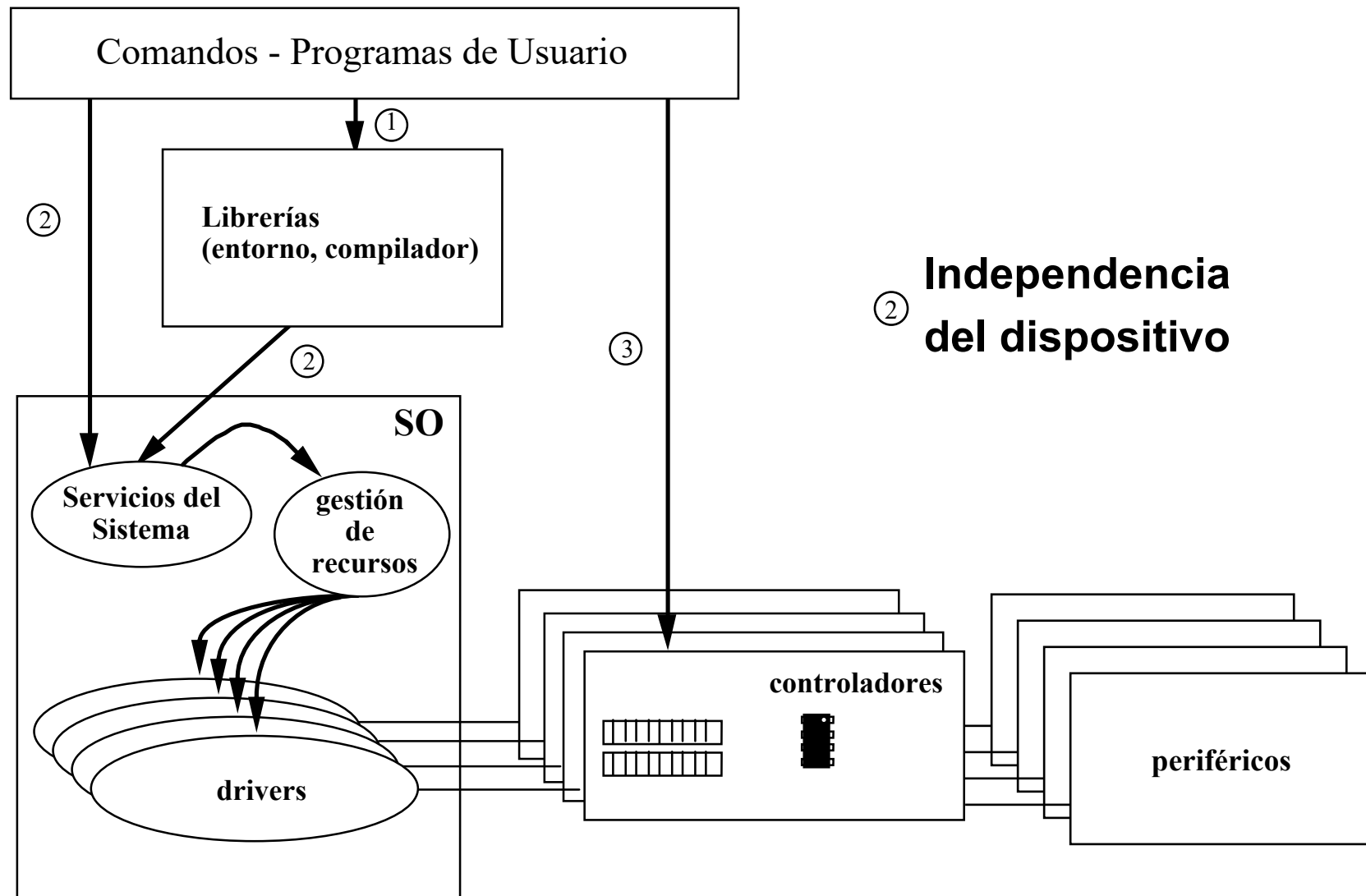
- *Hardware* de e/s. Recordatorio
- *Software* de la e/s. *Drivers*
- Tipos de dispositivos
- *Disco*: directorio, fichero, asignación de espacio
- Sistema de Ficheros UNIX
- UNIX: Estructura de un disco
- Tablas en memoria del Sistema de Ficheros

# Fundamentos del *hardware* de e/s

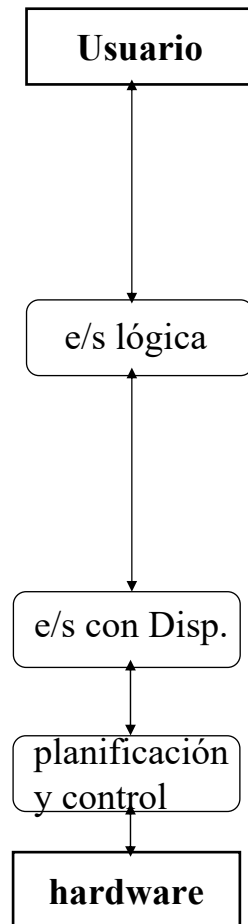


- **Sincronización por INTerrupciones**
- **Transferencia por Direct Memory Access**

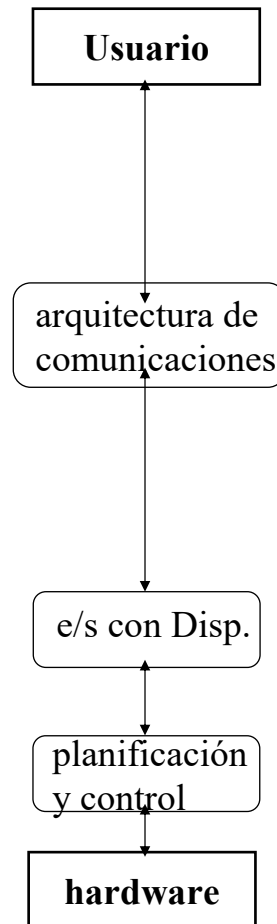
# Análisis del *software* de la e/s



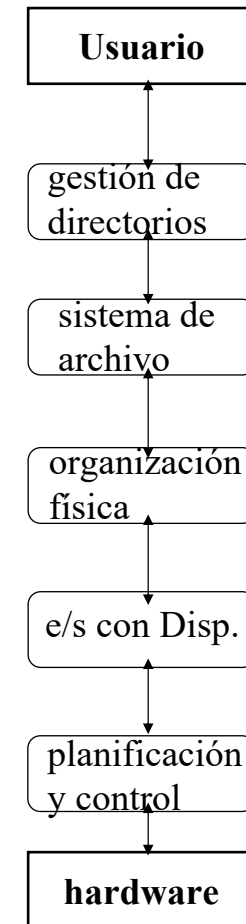
## Software de la e/s (2 de 2)



Periférico local



Puerto de comunicaciones



Sistema de Archivos

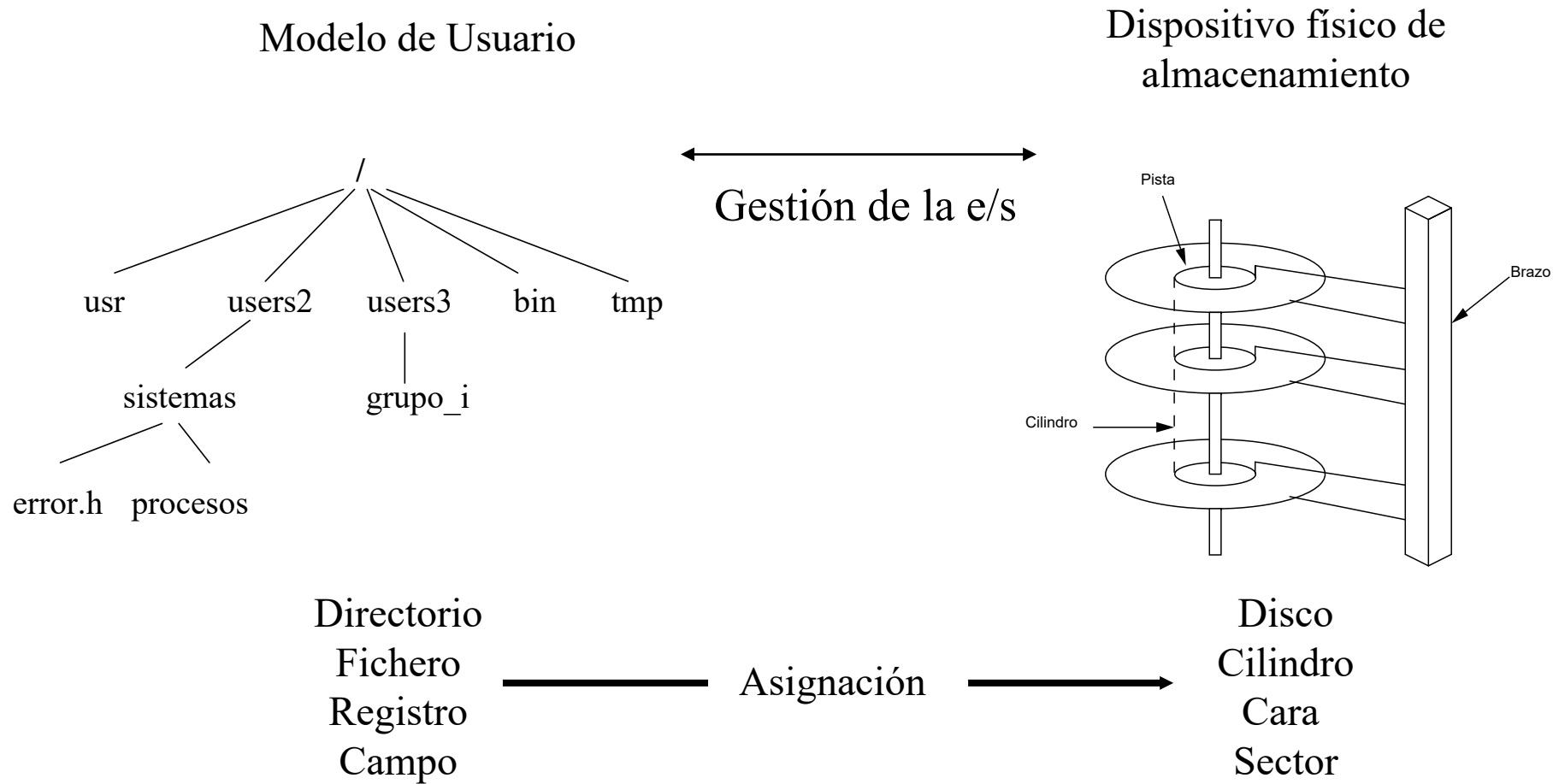
# *Drivers*

- Dispositivos lógicos mantenidos por el SO
- *Software* que trabaja con la e/s dependiente del dispositivo
- Un *driver* por cada dispositivo o por una clase si están muy relacionados
  - UNIX: número mayor, número menor
- Emite los comandos a los registros del  $K_D$
- Verifica que se ejecuten bien
- Conoce todos los detalles sobre el D y el  $K_D$ 
  - @ de registros del  $K_D$ , utilidad de los registros, mecánica del D, sectores, pistas...

# Tipos de Dispositivos

- De bloque. *Discos*
  - direccionable (bloque), con operaciones de localización
- De caracteres. *Terminales, impresoras, ratón...*
  - no direccionables, sin operaciones de localización

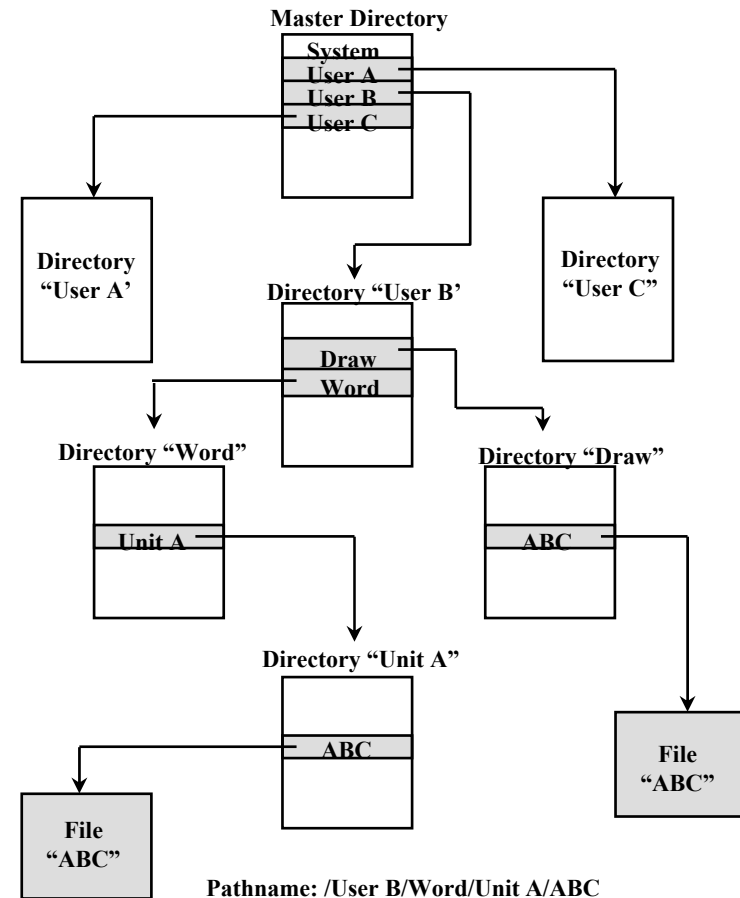
# Abstracción del disco





# Directorio: Espacio de nombres

- Un nivel
  - Todos los ficheros en un único directorio
- Dos niveles
  - Un directorio por usuario
- Jerarquía
  - Cada usuario puede organizar sus ficheros en subdirectorios



# Fichero

- Unidad lógica de almacenamiento secundario
  - Tipo abstracto de dato def. e implementado por el S.O.
  - Colección de información relacionada
  - Para el usuario es la unidad mínima de almac. secund.
- Atributos :
  - nombre, identificador, tipo, propietario, protección, tamaño, fechas, localización, etc.
- Operaciones:
  - crear, borrar, leer, escribir, truncar, reposicionar
- Tipos de acceso a la información:
  - secuencial, directo, indexado, hash, ...

# Fichero: operaciones

- Operaciones vs. Tipo de acceso
  - Secuencial:
    - rebobinar, leer\_siguiente, escribir\_siguiente
  - Directo:
    - leer(n), escribir(n), posicionar(n)
    - leer\_siguiente, escribir\_siguiente
  - Indexado, Hash
    - Leer(clave), escribir(clave), posicionar(clave)
    - leer\_siguiente, escribir\_siguiente

# Asignación de espacio

- A cada fichero se le debe asignar espacio físico en un dispositivo
- El sistema debe guardar la información necesaria para recuperar el contenido del fichero
- El sistema debe controlar el espacio libre de cada dispositivo
- El espacio se asigna como una o varias porciones

# Tamaño de las porciones

- Fijo/Variable
  - Variable minimiza el espacio perdido
  - Fijo simplifica la asignación de espacio
  - Se usa fijo: BLOQUE
- Grande/Pequeño
  - Grande minimiza tiempo de acceso
  - Grande minimiza información de acceso
  - Pequeño minimiza el espacio perdido

# Métodos de Asignación

- Asignación contigua
- Asignación encadenada
- Asignación indexada

# Asignación contigua

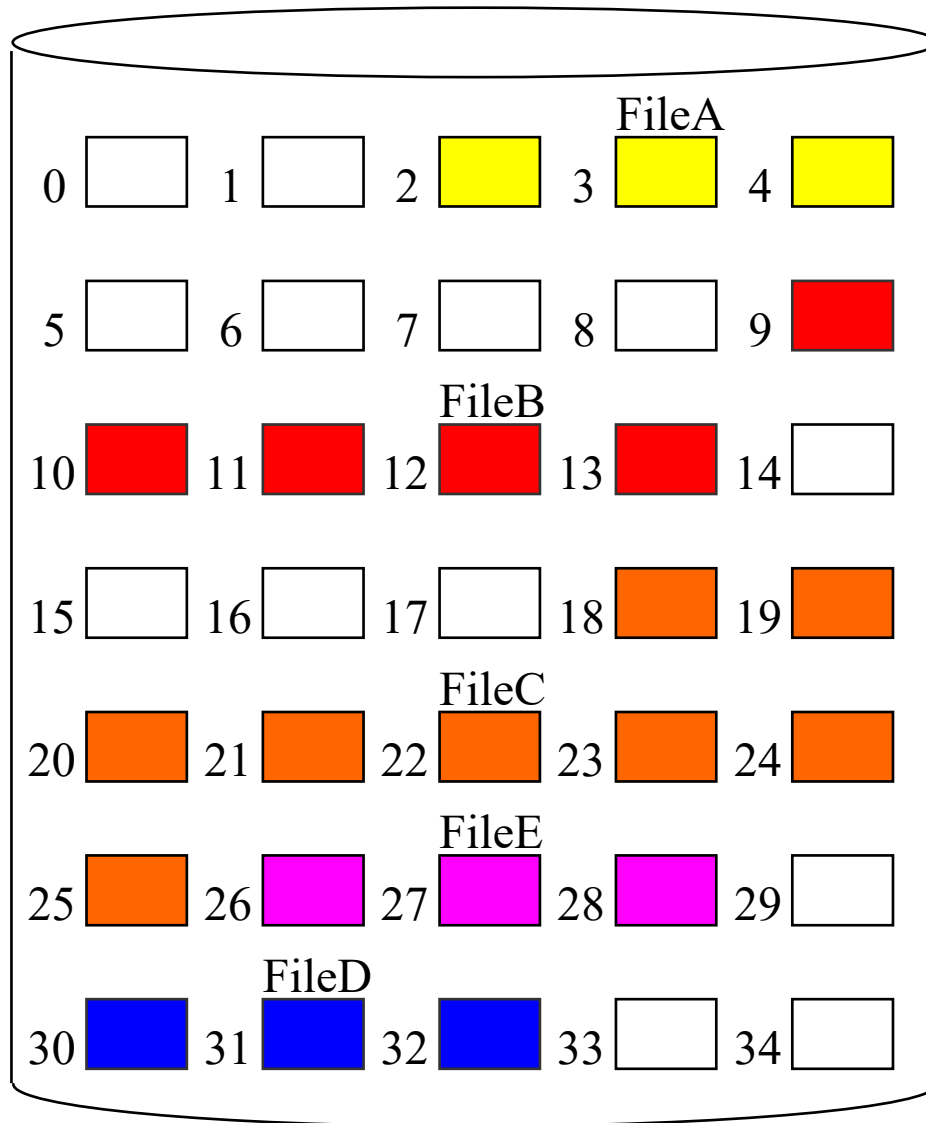


Tabla de localización

File Name	Start Block	Length
FileA	2	3
FileB	9	5
FileC	18	8
FileD	30	3
FileE	26	3

- Fácil localización: inicio y tamaño
- Fácil acceso secuencial y directo
- Difícil asignación
- Pérdida de espacio
- Problemas con cambios de tamaño

# Asignación encadenada

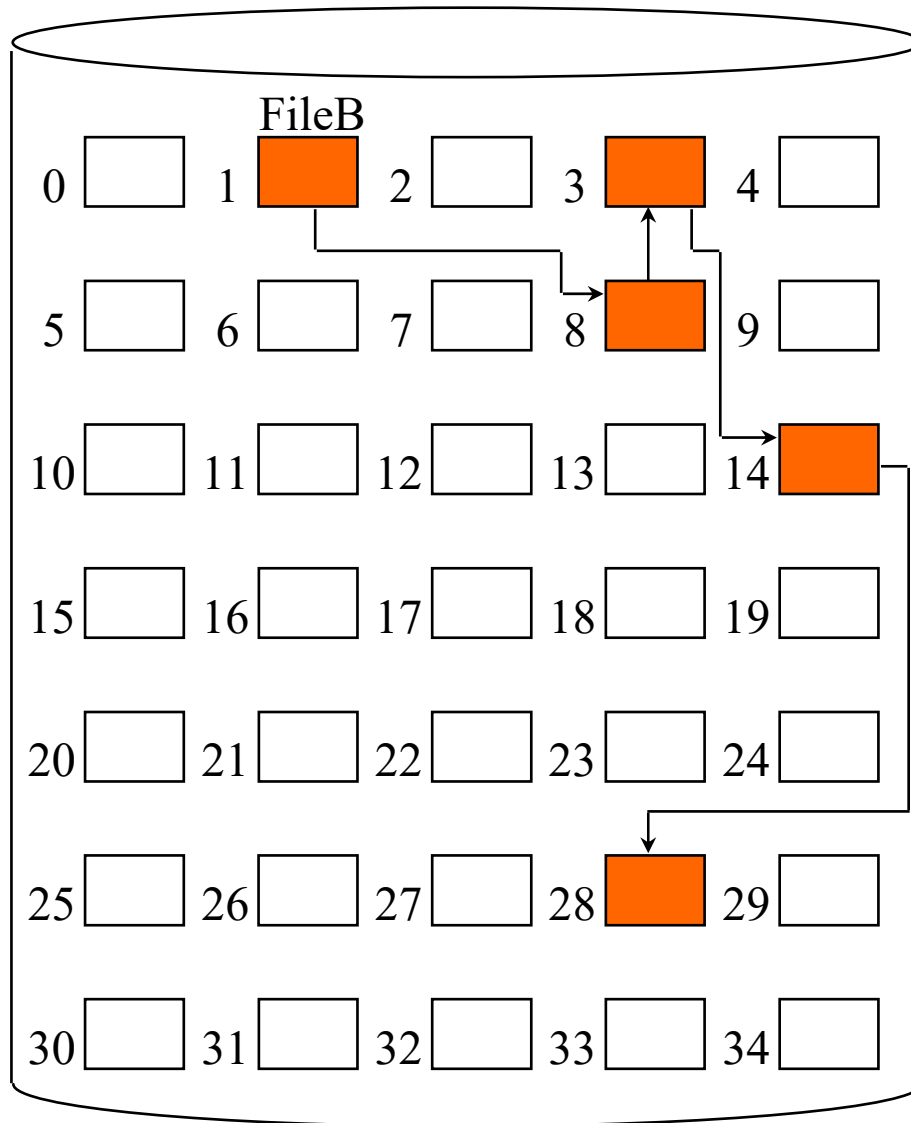


Tabla de localización

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

- Localización: inicio y tamaño
- Acceso secuencial
- Imposible acceso directo
- Fácil asignación
- Sin pérdida de espacio

La información se puede centralizar  
Modelo DOS: FAT



# Asignación indexada

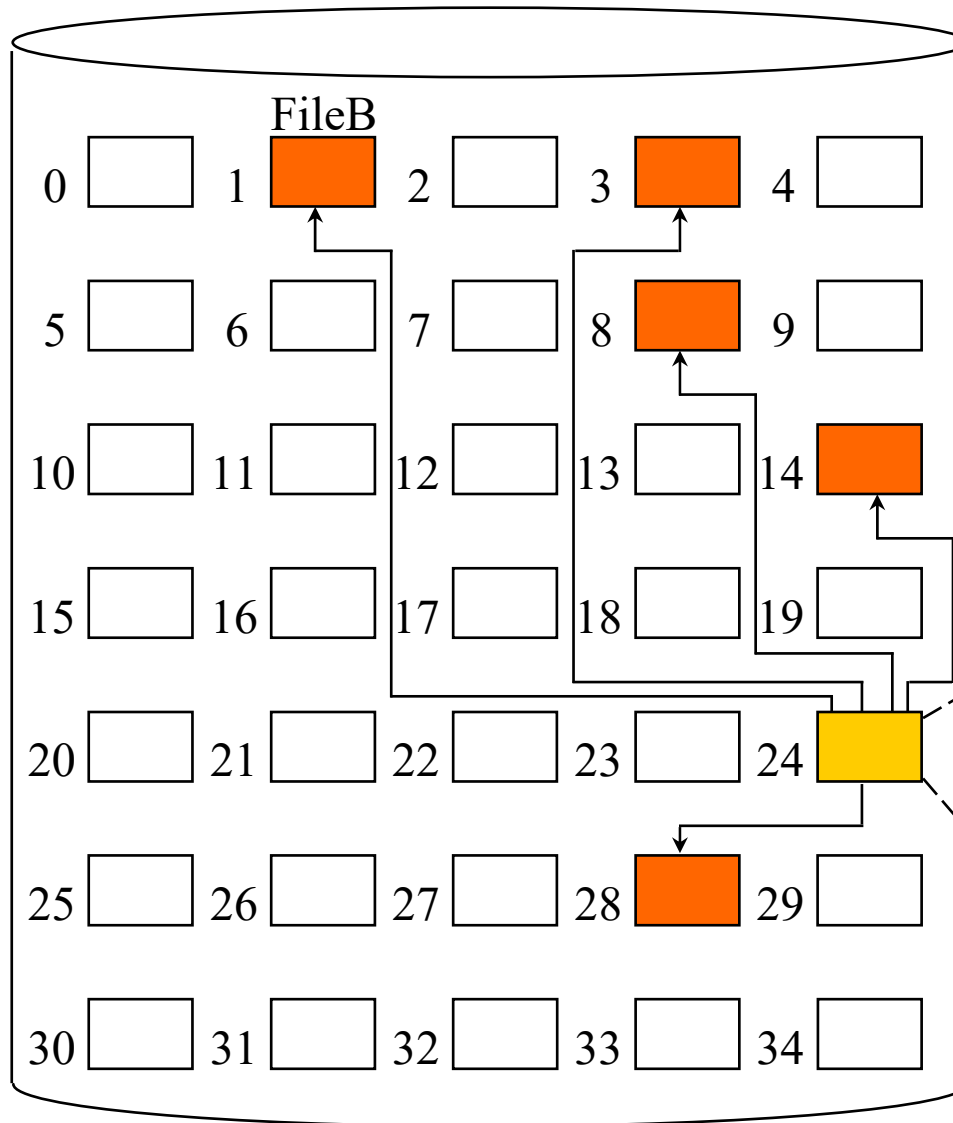


Tabla de localización

File Name	Index Block
...	...
FileB	24
...	...

1  
8  
3  
14  
28

- Localización: índice
- Acceso secuencial
- Acceso directo
- Fácil asignación
- Sin pérdida de espacio

Problema: tamaño del índice

# Sistema de ficheros UNIX

- Ficheros sin estructura
  - Son secuencias de bytes
- Tipos de ficheros
  - Ordinario: definido por el usuario
  - Directorio: contiene una lista de parejas nombres de fichero -- Inodo
  - Dispositivo: usado para el acceso a dispositivos de entrada/salida
  - FIFO, pipes o “named pipes”: Ficheros para comunicación entre procesos

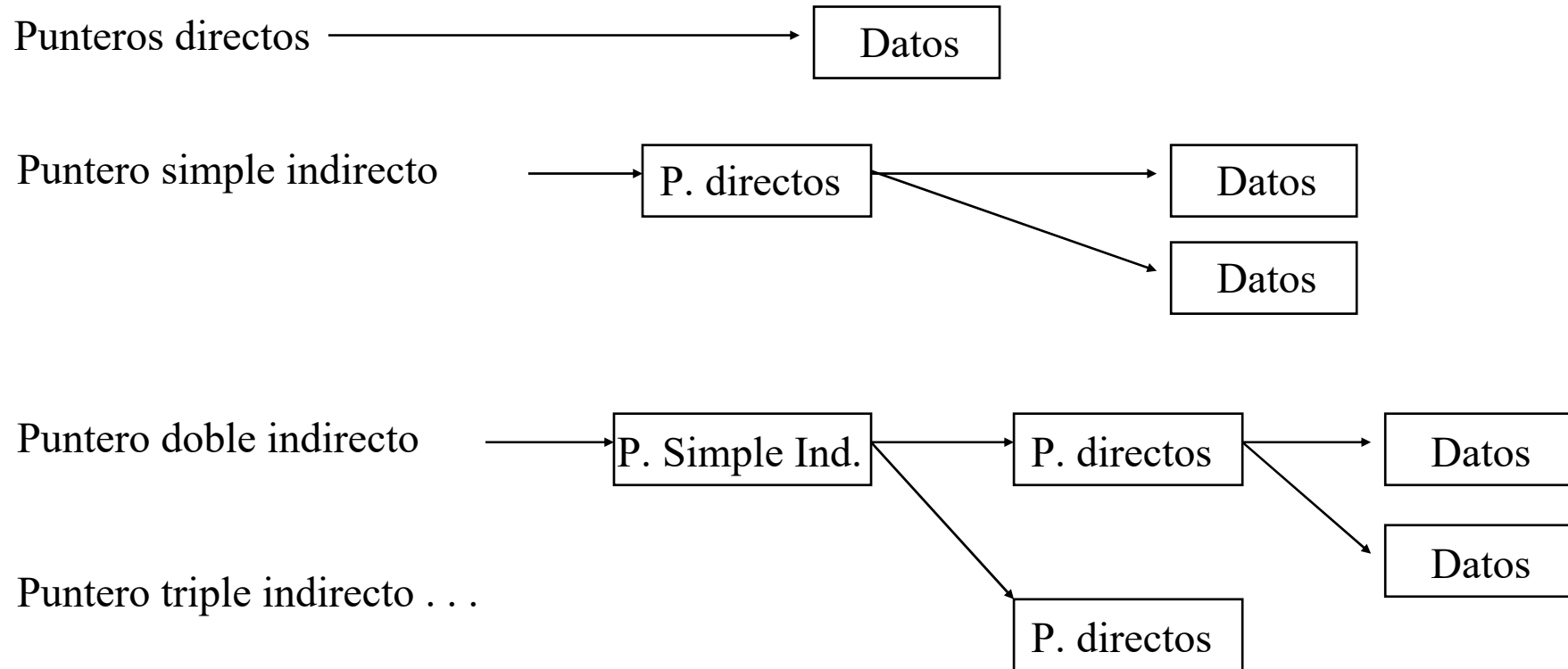
# UNIX: I-node

– Estructura de datos que almacena toda la información relativa a un fichero

- Número de I-node
- Tipo fichero
- Protección
- UID, GID
- Tamaño
- Fecha último cambio
- Número de links
- Información sobre localización de los bloques del fichero

# UNIX: I-node, localización

- Asignación indexada con varios niveles



# UNIX: I-node, localización

- Tipo de fichero Dispositivo

número **Mayor** / número **menor**

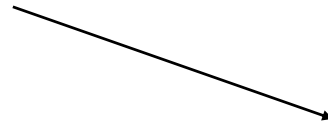
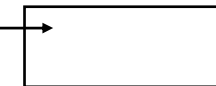
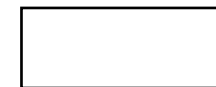
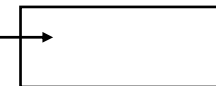


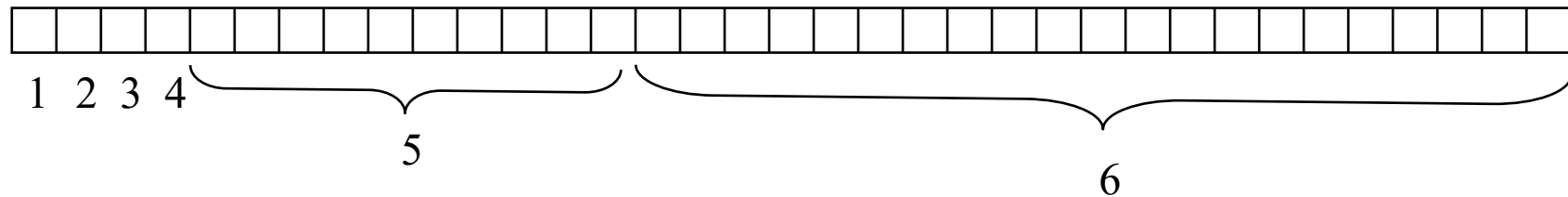

Tabla de localización



*drivers para la e/s*

# UNIX: Estructura de un disco

Ejemplo: MINIX



- 1. Bloque de arranque
- 2. Superbloque
- 3. Mapa de bits de nodos-l
- 4. Mapa de bits de bloques (o zonas)
- 5. Nodos-l
- 6. Bloques de datos

# Ejercicio: asignación en UNIX

- Recuerda:
  - Estructura de directorio: árbol
    - Directorio: fichero con parejas <nombre, i-nodo>
  - Localización de la información de un fichero
    - I-nodo: índice con varios niveles
  - Estructura de un disco
    - Tabla de I-nodos y bloques de datos
- Ejercicio: ¿ Cuantos bloques de disco se han de leer para encontrar un byte de un fichero? Por ej.
  - Byte: 450
  - Fichero: /users3/sistemas/error.h

# Tablas en memoria del Sistema de Ficheros

- Guardan información de los ficheros en uso
    - Razones: eficiencia, implementación del cursor, tipos de acceso
  - El usuario debe avisar al sistema que desea trabajar con un fichero: llamada open()
  - El usuario debe avisar cuando deja de trabajar con un fichero: llamada close()
  - Tablas en memoria: 2 tablas globales
    - I-nodos de todos los ficheros en uso (los abiertos) o los recientemente usados ...
- TI: tabla de I-nodos**
- Para el acceso secuencial se necesita un cursor. Si queremos que varios procesos trabajen al mismo tiempo con un fichero:
    - varios cursores, distintos permisos de acceso
- TFA: tabla de ficheros abiertos**
- Tablas en memoria: 1 tabla por proceso que contiene punteros a las entradas de TFA que está usando

**TDF: Tabla de Descriptores de Fichero**





# Tablas en memoria del Sistema de Ficheros (2)

