

# Sistema de Archivos

E. Campo   M. Knoblauch   Ó. López   J. Clemente

*Departamento de Automática*  
Universidad de Alcalá



/gso>

# Índice

- 1 **Introducción**
  - Definiciones
  - Funciones de un sistema de archivos
- 2 **Interfaz de los sistemas de archivos**
  - Archivos y directorios
  - Sistemas de archivos remotos
- 3 **Servicios del sistema de archivos**
  - Utilización de archivos desde procesos
  - Servicios para archivos
  - Servicios para directorios
  - Proyección de archivos en memoria
- 4 **Organización de los sistemas de archivos**
  - Estructura y organización interna
  - Creación de sistemas de archivos en disco
  - Montaje de sistemas de archivos
  - Fiabilidad de sistemas de archivos
- 5 **Casos de estudio**
  - UNIX System V
  - FAT
  - Otros sistemas de archivos

# Sistema de archivos

- Algunos procesos necesitan utilizar información que no siempre debe estar toda cargada en su espacio de memoria y que necesita ser almacenada de forma persistente

## Sistema de archivos

Estructura de datos dentro de una unidad lógica que permite al sistema operativo almacenar información de forma organizada e independiente de los procesos que la utilizan

- El sistema de archivos se encarga de abstraer propiedades físicas de diferentes dispositivos
- La información, al ser independiente de los procesos, se mantiene tras su finalización y, además puede ser utilizada por varios procesos

# Archivo

## Archivo

Conjunto de **datos persistentes** con un **identificador asociado** que el usuario ve como una entidad  $\Rightarrow$  **unidad mínima de almacenamiento lógico**

- Los archivos, normalmente, residen en dispositivos de almacenamiento permanente (no volátiles): cintas, discos magnéticos u ópticos, memorias flash, etc.
- El **acceso** se realiza mediante **llamadas al sistema**
- Pueden tener un **formato específico o no**. Es el creador el que define la organización y significado de los bits que lo componen
- El objetivo es **conseguir abstracción del hardware** de los **dispositivos de almacenamiento**

# Funciones de un sistema de archivos

- El **subsistema de archivos** es el **conjunto de módulos** del SO encargados de la **interacción entre el usuario y la información almacenada**
- El interfaz con el usuario proporciona:
  - Servicios de nombrado
    - Ubicación
    - Extensión
  - Servicios de archivos
    - Seguridad, protección y cifrado
    - Compartición
    - Acceso
    - Soporte a distintos tipos de archivo
  - Servicios de directorios
    - Organización de la información

# Atributos de archivos

- Un archivo se puede caracterizar por:
  - Nombre
  - **Identificador único**
  - Tipo de archivo
  - Ubicación
  - Tamaño
  - Protección
  - Fechas
  - Identificación de propietario
  - Información de control

# Tipos de archivos

- Facilidad en reconocer y soportar distintos tipos de archivos
- Los archivos se pueden agrupar en dos grandes clases: archivos ASCII y binarios
- Técnicas:
  - Incluir el tipo como parte del nombre del archivo (extensión):
    - MS-DOS
    - UNIX: número mágico
- Linux, por ejemplo, distingue entre archivos regulares, directorios, enlaces, tuberías, etc.

## Nombrado de archivo

- Característica principal: asociación de un nombre a un archivo en su creación
- Permite el acceso al archivo de forma inequívoca
- Nombres formados por caracteres alfanuméricos y especiales
- El tipo de nombres de un archivo varía según el SO:
  - Longitud variable: MS-DOS (8), UNIX (4.096)
  - Pueden tener extensiones para indicar el tipo de archivo:
    - Formato y significado variable en cada SO
    - Independientemente de la extensión, el SO debe reconocer al menos los archivos ejecutables (número mágico)



# Métodos de acceso al contenido de un archivo

- Considerando un archivo como una secuencia de datos de longitud fija y diferente en cada archivo, ¿cómo acceder a un dato concreto?
- Los sistemas operativos pueden proporcionar uno o varios métodos de acceso
- Formas de acceso básicas:
  - Acceso secuencial
  - Acceso indexado: MVS
  - Acceso directo: UNIX

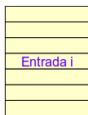
# Protección de archivos

- Puede ser necesario controlar el acceso a la información
- Los sistemas operativos modernos proporcionan mecanismos de protección y control de acceso a los archivos:
  - Esquemas de privilegios por usuarios o grupos por archivo para determinadas operaciones:
    - Lectura, escritura, ejecución...
  - Listas de control de acceso (ACLs)
  - Cifrado de sistemas de archivo

# Directorios

- Sistema de organización de los archivos que permiten al usuario acceder y localizar los archivos fácilmente
- Presentan una visión lógica simple al usuario, muy diferente del modo de almacenamiento real
- Un directorio es un objeto o estructura de datos compuesto de los elementos que agrupa, denominados entradas
- Cada entrada contiene información acerca del archivo al que referencia (MS-DOS) o un puntero a una estructura de datos que contenga dicha información (UNIX)

## Directorio



Nombre
Tipo de archivo
Tamaño
Propietario
Protección
Fecha de creación
Lista de bloques usados

## Estructuras de directorios (1/2)

- Sistema de directorios de un nivel:
  - Existe un único directorio (CP/M)
  - No hay clasificación, cada archivo debe tener un nombre diferente
- Sistema de directorios de dos niveles:
  - Existe un directorio maestro (directorio raíz) del que cuelgan otros directorios, uno por usuario
  - Las operaciones de un usuario sobre los archivos están restringidas a su directorio
  - Es posible que exista un directorio accesible a todos los usuarios que contenga ejecutables comunes:
    - Concepto de PATH

## Estructuras de directorios (2/2)

- Estructura jerárquica en árbol:
  - Complejidad de nombrado a nivel de usuario  $\Rightarrow$  estructura jerárquica más general
  - Permite a los usuarios clasificar sus archivos en subdirectorios. Ejemplos: MS-DOS, UNIX, Windows
  - Representación de directorios y subdirectorios desde el directorio raíz
  - Existe un camino único (path o ruta) desde la raíz hasta cada archivo:
    - Concepto de directorio de inicio o conexión
    - Concepto de directorio de trabajo o actual
    - Concepto de ruta absoluta y ruta relativa

# Sistemas de archivos remotos

- Existen diferentes sistemas de archivos que permiten acceder a archivos ubicados en servidores, por distintos usuarios conectados en puestos remotos
- El sistema operativo debe ser capaz de gestionar los protocolos de red e integrar las unidades remotas como parte del sistema de archivos local
- Algunos de estos sistemas son:
  - NFS
  - SMB - CIFS / SAMBA
  - CODA
  - AFS
- No confundir con sistemas de transferencia de archivos como:
  - FTP
  - SCP

## Acceso a archivos desde procesos

- Existen diferentes tablas de control de acceso para gestionar la comunicación de los procesos con los archivos
  - Tabla de descriptores de archivos
    - Existe una por proceso, el usuario sólo tiene acceso a un identificador de operación  $\Rightarrow$  descriptor
  - Tabla de archivos abiertos
    - Existe una única en el sistema, contiene tantas entradas como operaciones sobre archivos hayan sido permitidas
    - Cada entrada se vincula con un archivo mediante estructuras que dependen del tipo de sistema de archivos

# Tablas de control de acceso a archivos

## Proceso A

T. DESCR.  
DE ARCH.


0  
1  
2  
3  
4  
5

## Proceso B

T. DESCR.  
DE ARCH.


0  
1  
2  
3  
4  
5

## Kernel

T. DE ARCH.  
ABIERTOS

CUENTA: 1
CUENTA: 1
CUENTA: 1 MODO: Rd-
CUENTA: 1

ARCHIVOS

CUENTA: 3 /etc/passwd
CUENTA: 1 /usr/local/



# Servicios del sistema de archivos para archivos

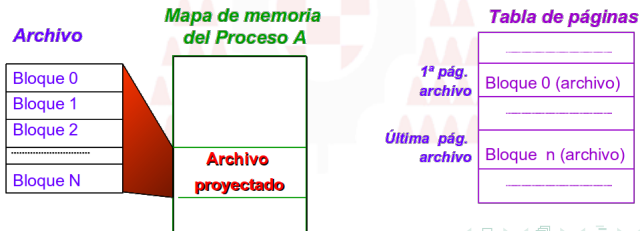
- Solicitar operación con archivo (abrir)
- Liberar operación (cerrar)
- Leer contenido de archivo
- Escribir contenido en un archivo
- Cambiar el apuntador en la operación
- Consultar atributos de un archivo
- Manipular atributos de un archivo
- Etc.

## Servicios del sistema de archivos para directorios

- Crear una nueva entrada de cualquier tipo de archivo y/o directorio
- Eliminar una entrada de un directorio
- Solicitar operación sobre un directorio (abrir, siempre para lectura)
- Liberar operación sobre un directorio (cerrar)
- Leer contenido de una entrada de un directorio
- Modificar el directorio actual
- Modificar el directorio raíz
- Obtener la ruta del directorio actual
- Etc.

## Proyección de archivos en memoria (1/2)

- Forma alternativa de acceso a los archivos mediante el uso de la técnica de memoria virtual
- Un programa puede solicitar que se haga corresponder (proyectar) un archivo (o parte) en su mapa de memoria
  - El SO se encarga de mantener la correspondencia con los bloques del archivo proyectado
- Al acceder a una dirección de memoria asociada al archivo proyectado, se está accediendo al archivo



## Proyección de archivos en memoria (2/2)

- Proyección de archivos frente a lectura/escritura habitual:
  - Menos llamadas al sistema para el acceso
  - Se evitan copias intermedias del sistema de archivos
  - Facilidad de programación de los accesos
- En la solicitud de protección se puede especificar:
  - Definición de protección
  - Privada o compartida (en algunos sistemas). Ejemplo: bibliotecas dinámicas
- Servicios principales: proyectar y eliminar proyección

## Métodos de asignación de espacio (1/2)

- Hay que maximizar el uso del espacio en disco
- Modos de almacenamiento:
  - Colocar toda la información seguida: provoca un desaprovechamiento del espacio y posible fragmentación
  - Utilizar bloques no contiguos: Necesario definir el tamaño del bloque y controlar los bloques libres
- Existen diferentes métodos para conocer los bloques asignados a cada archivo:
  - Asignación contigua o adyacente:
    - Sólo realizable si se conoce el tamaño máximo del archivo en el momento de su creación
    - Se necesita la dirección del primer bloque
    - Buen rendimiento al poder leer todo el archivo de una sola vez
    - Produce bastante fragmentación

## Métodos de asignación de espacio (2/2)

- Gestión de bloques asignados a cada archivo (cont.):
  - Asignación enlazada:
    - Todos los bloques están organizados en una lista enlazada
    - Problema: para leer o escribir en el bloque "n", hay que recorrer los "n-1" precedentes
    - MS-DOS posee una Tabla de Localización de Archivos (FAT) duplicada, si la información está en caché mejora el rendimiento
  - Asignación indexada:
    - Se utiliza un bloque que contiene un vector de índices enlazable. Las búsquedas sólo se hacen en dicho vector
    - Permite el acceso directo a cualquier posición, pero desperdicia espacio
    - Es necesario definir el tamaño del bloque inicial, ya que limita el tamaño máximo del archivo
    - UNIX utiliza una variante de este método

## Gestión de espacio libre

- Es necesario controlar los bloques disponibles para asignarlos a nuevos archivos o archivos que cambian
  - Mapa de bits (preferible si cabe en memoria):
    - Unidad con  $n$  bloques utiliza un mapa de  $n$  bits
    - Un 1 indica bloque libre, y un 0, ocupado
  - Lista enlazada de bloques:
    - El SO mantiene un apuntador al primer bloque libre, que contiene la dirección del siguiente bloque libre...
  - Lista enlazada con cuenta de bloques libres:
    - Similar al anterior, pero cada bloque almacena tanto la dirección del siguiente como el número de bloques consecutivos
  - Lista indexada:
    - El primer bloque libre se usa como índice al resto de bloques libres (normalmente también con cuenta de bloques libres)

## Particionado (1/2)

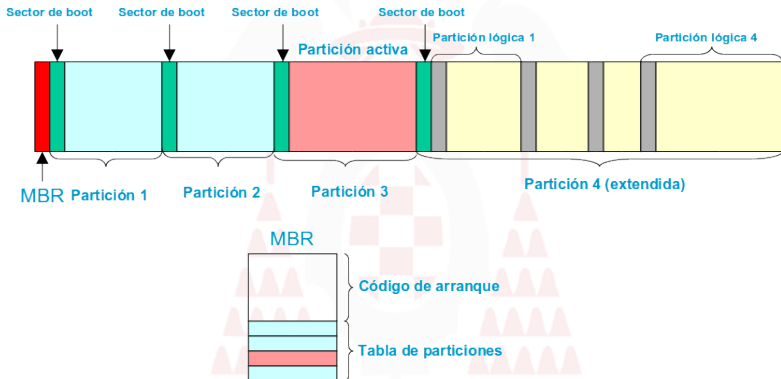
- Un sistema de archivos debe residir en una única unidad de disco
- Se pueden tener diferentes tipos de sistemas de archivos en un mismo equipo (o varios del mismo tipo) particionando los discos duros
- Particionar un disco duro es una manera de dividir el disco físico en varios discos lógicos
- Para delimitar los discos físicos, es necesario reservar una zona con cierta información  $\Rightarrow$  Master Boot Record
- El MBR ocupa el primer sector del disco, conteniendo el programa de arranque, la tabla de particiones con cuatro entradas y el número mágico (0xAA55)



## Particionado (2/2)

- En la zona del programa de arranque puede existir un gestor de arranque o directamente un enlace a la partición arrancable
- Cada entrada de la tabla de particiones contiene el tipo de partición, si es la partición activa o no y la posición de inicio y fin en el disco
- Para poder tener más de cuatro particiones en un disco, existe un tipo especial de partición, la partición extendida
- Una partición extendida permite definir un número ilimitado de particiones lógicas dentro de ella.
- El número mágico indica si el MBR es correcto o no
- El resto de sectores del disco pueden pertenecer a una partición (con un sistema de archivos o no) o estar disponibles para asignar

# Esquema de particionado de un disco



# Formateo

- No hay que confundir particionar con formatear, son procesos distintos e independientes
- En el particionado se establecen qué parte o partes van a estar disponibles para un sistema de archivos
- En el formateo se prepara una partición con el sistema de archivos elegido
- Formatear consiste en crear e inicializar todas las estructuras de datos que utilice un sistema de archivos que permiten tanto la gestión del espacio ocupado como de los bloques libres
- Por lo tanto, el formateo consume espacio de una partición y es diferente para cada tipo de sistema de archivos

# Montaje de sistemas de archivos

- Decisión de diseño: ofrecer o no un único árbol de directorios en el sistema
  - En sistemas Windows: árbol de directorios único por dispositivo lógico
  - Sistemas derivados de UNIX: árbol de directorios único
  - El sistema debe ofrecer servicios para asociar y desasociar unos sistemas de otros en un árbol de nombres único
  - Órdenes o llamadas al sistema de UNIX: mount y umount

# Fiabilidad de sistemas de archivos

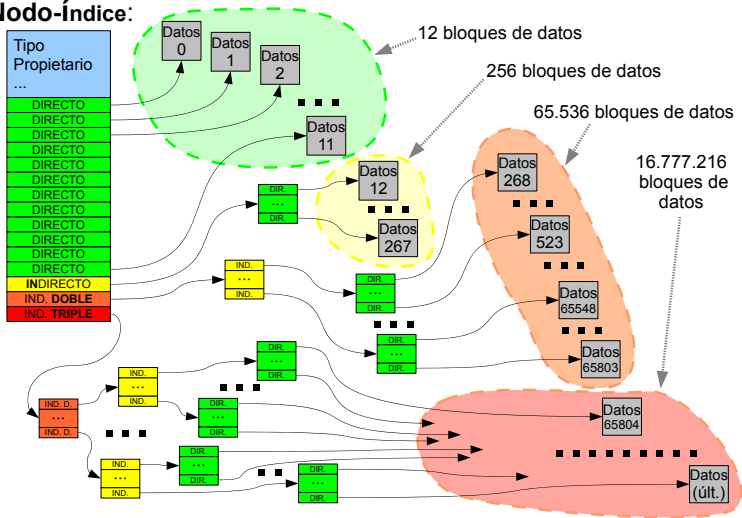
- Son necesarias soluciones ante posibles errores del medio físico
- Mecanismos de prevención de pérdida de datos:
  - Copias de seguridad
  - Control de versiones
- Mecanismos de recuperación de errores:
  - Frente a errores no catastróficos (sectores sin datos válidos):
    - Marcado de sectores defectuosos por HW o SW
  - Frente a errores catastróficos (posible pérdida de información):
    - Discos redundantes (RAID 0, 1, 10, 5, 6, 60, ...)
  - Frente a errores en el sistema de archivos (corte alimentación):
    - Sistemas transaccionales (journaling): JFS, ReiserFS, XFS, ...

## Descripción de un sistema de archivos tipo UNIX

- Posee una estructura jerárquica de árbol único
- Permite un crecimiento dinámico de los archivos
- Permite proteger los datos de los archivos
- Mantiene independencia de los dispositivos
- Todo son archivos de diferentes tipos:
  - Normales, directorios, dispositivos tipo carácter y tipo bloque, tuberías, enlaces
- Existe una tabla de nodos índice para acceder a los archivos
  - Cada nodo índice posee 15 punteros a bloques (12 punteros directos, uno indirecto simple, uno doble y uno triple)

# Estructura de bloques de un archivo

## Nodo-Índice:



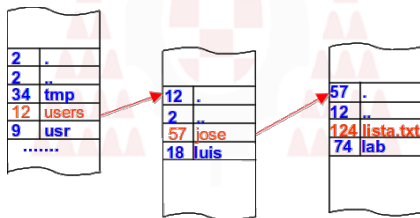
# Estructura de algunos tipos de archivos

- Archivos normales
  - Contienen la información directamente
- Directorios
  - Contienen al menos pares nodo índice-nombre
  - Al crearse, ya contienen dos entradas
- Enlaces simbólicos
  - Contienen una cadena de texto que se interpreta al leer el archivo
- Archivos especiales: de dispositivo modo bloque, modo carácter, tuberías, sockets
- Los enlaces fuertes NO son un tipo de archivos



## Paso de pathname a nodo índice

- Se recorren los nodos correspondientes a los directorios según la información que se encuentra en ellos y la ruta a resolver
- Según la ruta sea absoluta o relativa, se comienza en el nodo índice del directorio actual o en el del directorio raíz
- Ejemplo:
  - `/users/jose/lista.txt` ⇒ nodo índice 124



## Sistema de archivos FAT (1/2)

- Por cada archivo, contiene una lista enlazada de los bloques que contienen la información
- Posee una tabla de asignación de archivos (FAT) que contiene una entrada por bloque de la partición. Que puede contener:
  - Un identificador de bloque libre
  - Un valor numérico que indica el bloque que contiene la siguiente información del archivo
  - Un identificador de que el bloque es el último de un archivo
- Debido a la importancia que tiene la FAT, se almacenan dos copias de la misma
- Tanto las tablas como el directorio raíz se almacenan en una ubicación definida para que el sistema pueda arrancar





## Sistema de archivos FAT (2/2)

- La actualización de la tabla FAT es muy importante y consume mucho tiempo (las cabezas deben moverse de la posición del archivo a la posición de la tabla)
- Si no se actualiza la tabla FAT periódicamente, pueden aparecer datos no accesibles, ya que no están referenciados
- Inconvenientes:
  - Con particiones muy grandes, se desperdicia mucho espacio, ya que la FAT es muy grande
  - Fragmentación: los bloques se asignan sin ningún orden
- FAT no es útil hoy día, ya que fué diseñado para sistemas con discos pequeños y poca cantidad de memoria RAM
- FAT32 es una evolución de FAT, más robusta y flexible, con algunas mejoras como, por ejemplo, sistemas de archivo más grandes o nombres de archivo más largos

## Otros Sistemas de archivos

- NTFS (New Technology File System)
  - Sistema de archivos transaccional
  - Permite cifrado de archivos
  - Proporciona control de acceso para archivos y directorios
- HFS+ (Hierarchical File System)
  - Utilizado en Mac OS como mejora del inicial HFS
  - Mantiene la información del sistema de archivos en un archivo catálogo.
  - Se usa en disco duros o CDs y DVDs. Es utilizado también en reproductores IPod y reconocido por Linux
- ZFS
  - Creado por Sun Microsystems para Solaris
  - Transaccional permite sistemas de archivos y archivos muy grandes (16 EB)
  - Permite realizar rápidas instantáneas del sistema de archivos para copias de seguridad

## Referencias bibliográficas

-  [Sánchez, 2005] S. Sánchez Prieto.  
*Sistemas Operativos.*  
Servicio de Publicaciones de la UA, 2005.
-  [Tanenbaum, 2009] A. Tanenbaum.  
*Sistemas Operativos Modernos.*  
Ed. Pearson Education, 2009.
-  [Stallings, 1999] W. Stallings.  
*Organización y arquitectura de Computadores.*  
Ed. Prentice Hall, 1999.
-  [Silberschatz, 2006] A. Silberschatz, P. B. Galván y G. Gagne  
*Fundamentos de Sistemas Operativos.*  
McGraw Hill. 2006