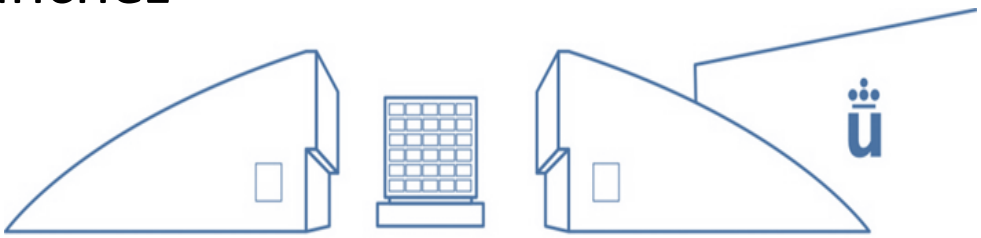
Tema 6

Gestión de ficheros

Alberto Sánchez



Bibliografía

- **“Fundamentos de sistemas operativos”**

  Silberschatz, Galvin, Gagne. Editorial Mc Graw Hill.

  - Capítulo 10: 10.1, 10.2, 10.3, 10.4, 10.6
  - Capítulo 11: 11.2.1, 11.3, 11.4, 11.5

Introducción

- Hardware de E/S
  - Controladores de dispositivos
    - E/S programada: No concurrencia E/S-procesador
    - Interrupciones: Concurrencia E/S-procesador
    - DMA: Máxima concurrencia E/S-procesador
  - Dispositivos de E/S
    - De bloques (discos)
    - De caracteres (teclado, ratón)
- Objetivos del SO:
  - Controlar el funcionamiento de los dispositivos de E/S
  - Facilitar el manejo de los dispositivos de E/S a través de interfaces
  - Proporcionar mecanismos de protección

Índice

- **Interfaz del sistema de ficheros**
  - Sistema de ficheros
  - Ficheros
  - Directorios
- Implementación del sistema de ficheros
  - Estructuras
  - Asignación de bloques
  - Gestión del espacio libre

Sistema de ficheros

- El acceso a los dispositivos es:
  - Complejo
    - Detalles físicos de los dispositivos
    - Dependiente de las direcciones físicas
  - Sin protección
    - Si el usuario accede a nivel físico no tiene restricciones. El controlador del dispositivo no limita.
- Sistema de ficheros: permite organizar la información dentro de los dispositivos de almacenamiento secundario
  - El S.O. contempla los archivos como conjuntos de datos estructurados según sus necesidades de almacenamiento y representación.
- Objetivos:
  - Suministrar una visión lógica de los dispositivos, presentándolos como ficheros
  - Ofrecer primitivas de acceso cómodas e independientes de los detalles físicos
  - Incorporar mecanismos de protección

Fichero

- Un fichero es la unidad lógica de almacenamiento secundario
- Los ficheros tienen una estructura definida que dependerá de su tipo:
  - Secuencia de Bytes
  - Caracteres ASCII separados en líneas
  - Registros de longitud fija o variable
  - Documentos formateados
  - Binarios: almacenan archivos ejecutables, objetos, y datos no textuales.
- En Linux/Unix existen archivos especiales, que permiten modelar cualquier dispositivo de E/S como una archivo más del sistema (/dev)

Tipos de ficheros

- En ocasiones es útil conocer la naturaleza de la información contenida en un fichero:
  - ¿Es una imagen? ¿Un ejecutable? ¿Un PDF?
  - Se evitan confusiones

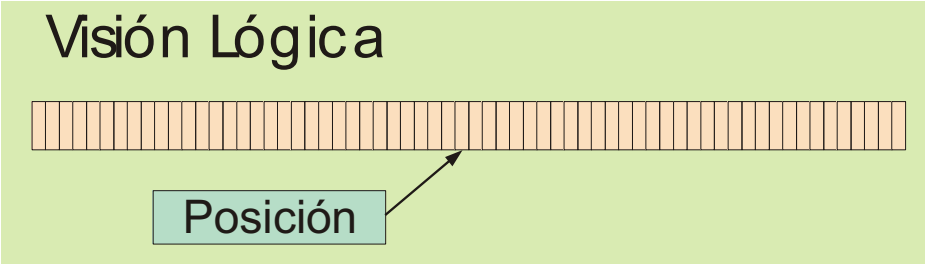
- Se puede tener información del tipo de fichero:
  - Microsoft Windows: mediante la extensión. El nombre del fichero se complementa con un código que indica el tipo:
    - .exe, .bat, .com, .jpg, .c
  - Linux/Unix: mediante números mágicos. Al comienzo del fichero se incluye una secuencia de Bytes que indica el tipo de fichero
    - Mandato `file`

Estructura

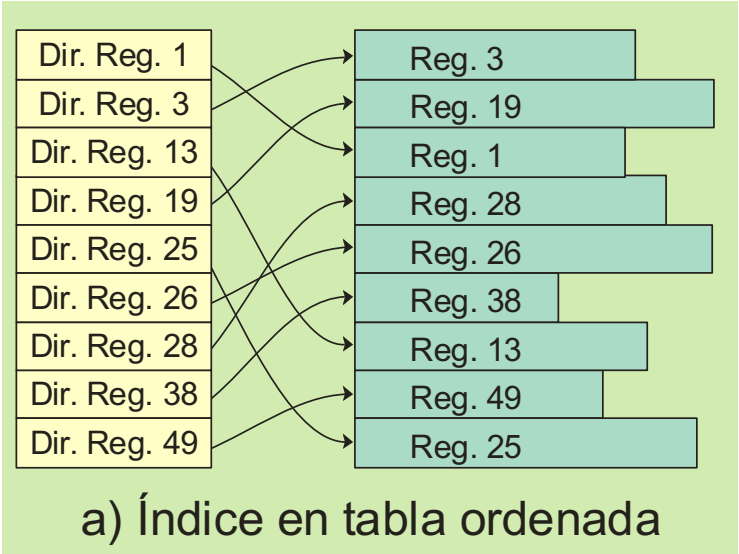
- El S.O operativo debe proporcionar una estructura de archivo genérico que:
  - De soporte a todos los tipos de archivos
  - Proporcione un mecanismo de nombrado
  - Proporcione facilidades para proteger los archivos
  - Permita explotar el almacenamiento secundario de forma sencilla y eficiente.
- Estructura genérica
  - I-nodo en UNIX
  - FAT en MS-DOS
  - Registro MFT en NTFS (Windows NT en adelante)

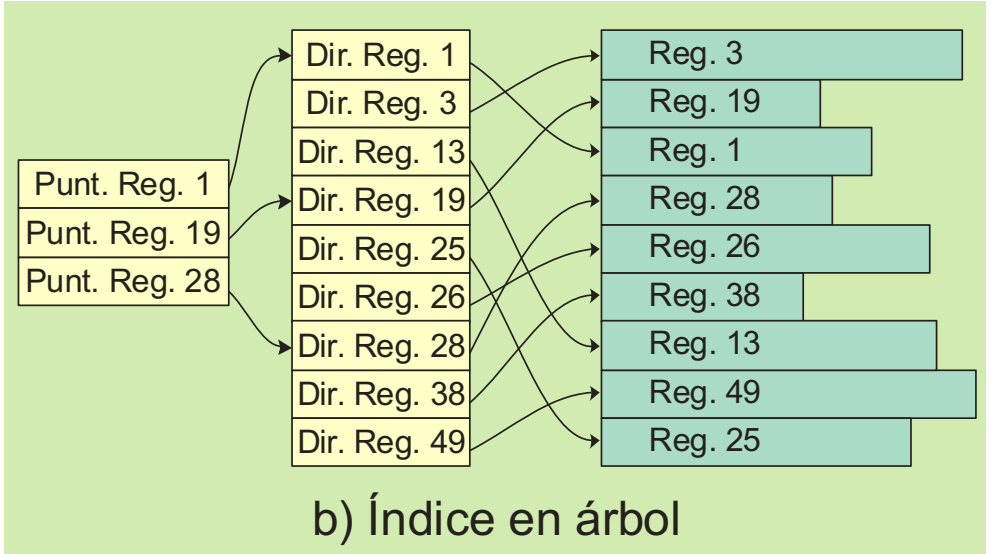
Visión lógica del fichero

- Existen varias alternativas. La más utilizada es la primera
  - Secuencia o tira de bytes



  - Ficheros indexados: se suelen construir como una capa software adicional sobre la visión de tira de bytes





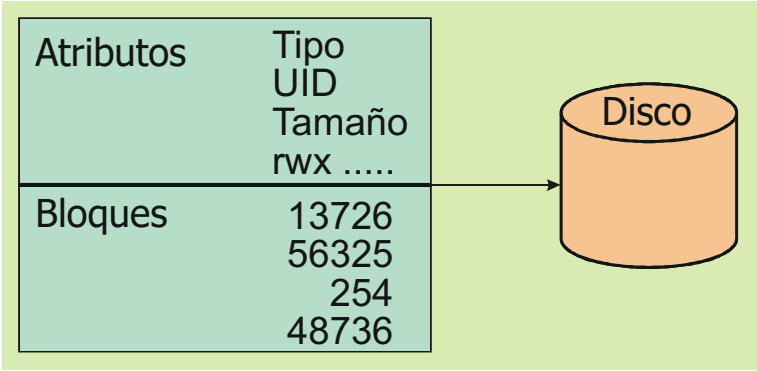
  - Visión compleja: Bases de datos

Visión física del fichero

- En que bloques de disco se encuentra realmente almacenado . Depende del Sistema de Ficheros
  - Fragmentación interna: el último bloque puede no estar completamente ocupado

Archivo A
Bloques: 13
20
1
8
3
16
19

- En general, cada fichero tiene atributos y una lista ordenada de bloques:



Atributos de un fichero

- **Atributos**
  - **Nombre**: identificador legible por los usuarios
  - **Identificador**: etiqueta unívoca, normalmente un entero, que utiliza el sistema de ficheros
  - **Tipo**: indica el tipo de información que contiene
  - **Dueño y grupo**
  - **Tamaño**: cuánto ocupa (en Bytes, palabras o bloques)
  - **Protección**: información de quién y cómo puede acceder al fichero
  - **Fecha, hora**
  - **Número de enlaces**
- **Operaciones sobre ficheros:**
  - Crear, abrir y cerrar
  - Renombrar y borrar
  - Leer y escribir
  - Truncar
  - Reposicionamiento
  - Obtener y modificar atributos

Semántica de coutilización

- En situaciones de coutilización de un archivo, las acciones de un proceso pueden afectar a la visión que los otros tienen de los datos almacenados.
- La semántica de coutilización especifica que ocurre cuando varios procesos acceden de forma simultánea al mismo archivo, y especifica en que momento las modificaciones que un proceso realiza sobre un archivo son visibles al resto de procesos.
- Ej: semántica de UNIX
  - Cualquier lectura de archivo ve los efectos de todas las escrituras previas sobre ese archivo.
  - En caso de accesos concurrentes de escritura, se escriben los datos en el orden de petición al S.O.
  - Si los usuarios quieren estar seguros de que los datos se escriben en el orden correcto, deben usar cerrojos para bloquear los accesos al archivo.

Directorios

- Los sistemas de archivos pueden llegar a ser muy grandes.
- Para facilitar el acceso a los archivos, todos los S.O. proporcionan formas de organizar los nombres de archivos mediante directorios
- Operaciones sobre directorios:
  - Buscar fichero
  - Crear fichero
  - Borrar fichero
  - Listar el directorio
  - Crear (insertar) y borrar (eliminar) directorios.
  - Leer entradas de un directorio.
  - Montaje

Árbol de directorios

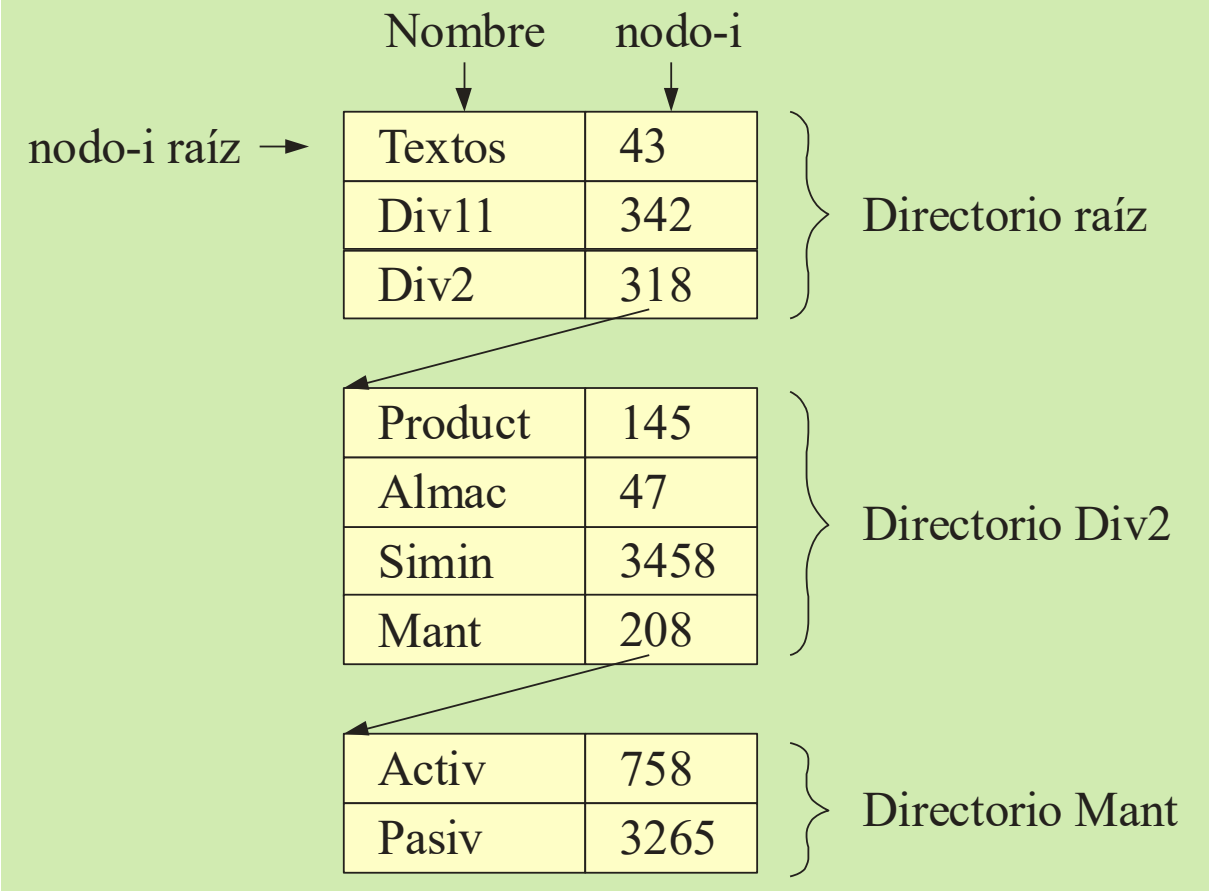
- Los directorios pueden contener ficheros o subdirectorios
- Un subdirectorio es un *fichero especial* que contiene entradas de directorio
- Nombre de ruta: lugar dentro del árbol donde se encuentra un fichero
- Dos tipos de nombres de ruta:
  - Absoluto: desde el directorio raíz
  - Relativo: desde el directorio actual

Visión lógica de directorios

- Un directorio es una tabla de entradas que asocia nombres simbólicos a ficheros.
- Esquema jerárquico.
- Cuando se abre un fichero el SO busca el nombre en la estructura de directorios.
- La organización jerárquica de directorios
  - Simplifica el nombrado de ficheros (nombres únicos)
  - Proporciona una gestión de la distribución => agrupar ficheros de forma lógica (mismo usuario, misma aplicación)

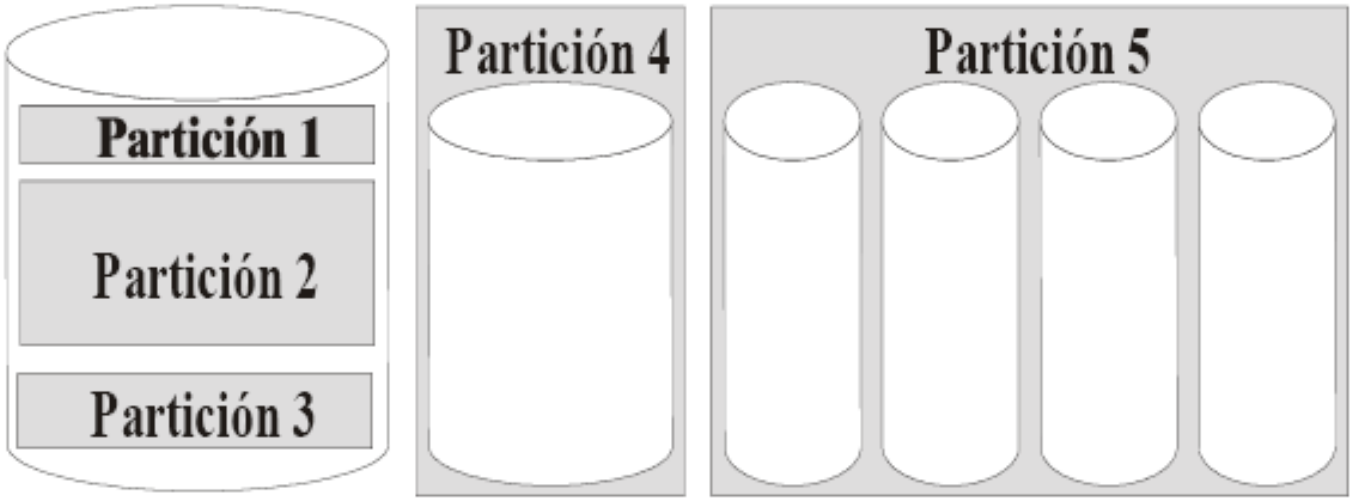
Visión física de directorios

- Directorios en UNIX



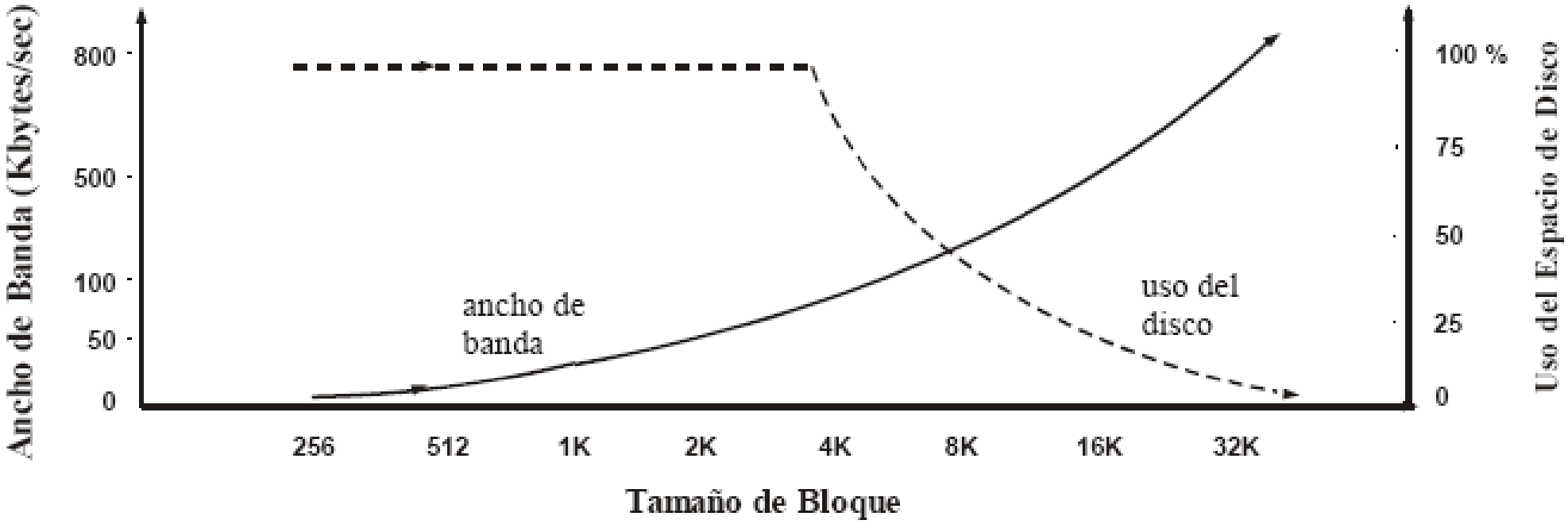
Particiones

- Una partición o volumen es una porción de disco a la que se dota de una identidad propia y que puede ser manipulada por el sistema operativo como una entidad lógica independiente.
- Una partición también puede ser un conjunto de discos agrupados como una sola entidad lógica independiente.



Tamaño de bloque

- Bloque: agrupación lógica de sectores de disco. Es la unidad de transferencia mínima que usa el sistema de archivos.



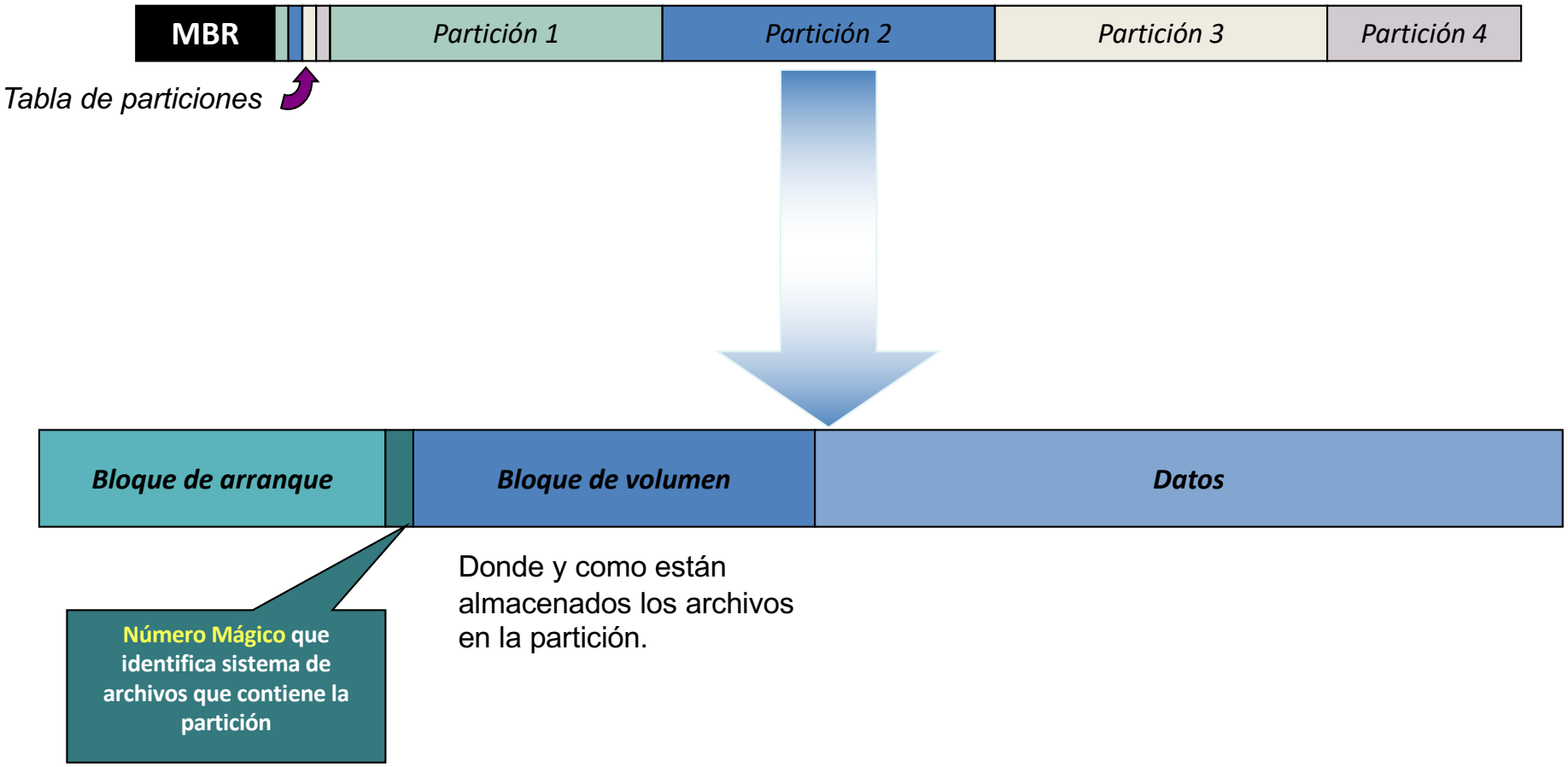
Índice

- Interfaz del sistema de ficheros
  - Sistema de ficheros
  - Ficheros
  - Directorios
- **Implementación del sistema de ficheros**
  - Estructuras
  - Asignación de bloques
  - Gestión del espacio libre

Estructuras

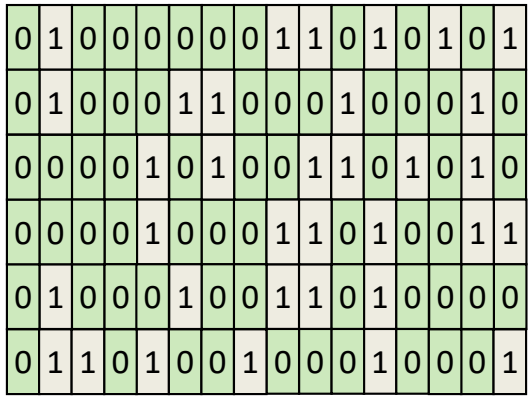
- Varias estructuras en disco para implementar el sistema de ficheros:
  - **Bloque de control de arranque**: primer bloque de la partición, contiene el cargador del SO
  - **Tabla de particiones** información sobre número, tipo y tamaño de particiones
  - **Superbloque/Bloque de control de volumen/partición**: primera estructura que se lee y que indica las ubicaciones de todas las demás. Su ubicación es fija y contiene información sobre número y tamaño de los bloques, relación de bloques libres, **estructura de directorios** (relación de filename-FCBs)
  - **Bloque de control de fichero (FCB)**: contiene los atributos del fichero (nombre, identificador, permisos,…) y la relación de bloques en disco
    - En sistemas de tipo Unix se llama *i-nodo*
    - En sistemas de tipo Windows se llama *FAT, Registro MFT*

Organización sistema de archivos

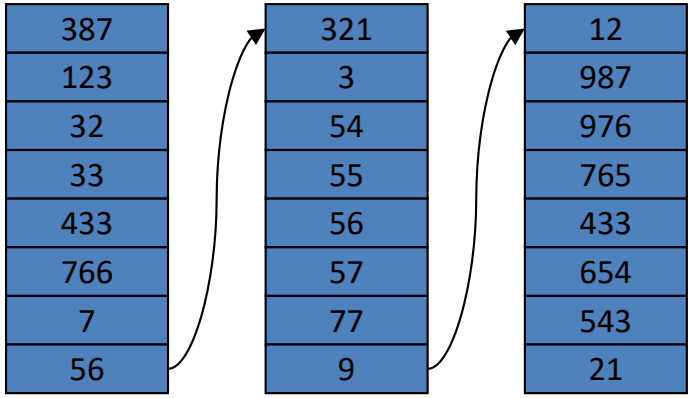


- El sistema de ficheros tiene que tener algún mecanismo que le permita gestionar el espacio libre

- **Mapa de bits:**
  - Un mapa de bits indica si un bloque está libre o no
  - Problema:
    - El mapa de bits ocupa mucho espacio si el dispositivo es muy grande
      - Usarlo directamente desde el disco => puede ser muy ineficiente
      - Guardarlo en memoria RAM => puede ser muy grande



- **Lista enlazada de recursos libres:**
  - Se utiliza una lista enlazada de bloques de disco que contienen números de bloques libres
  - Se almacenan las direcciones de N bloques libres en el primer bloque libre
    - Las N-1 primeras direcciones son realmente bloques libres
    - La última apunta a un bloque con direcciones de bloques libres
  - Para agilizar el proceso de búsqueda se mantiene uno o más bloques en memoria, dejando el resto en disco



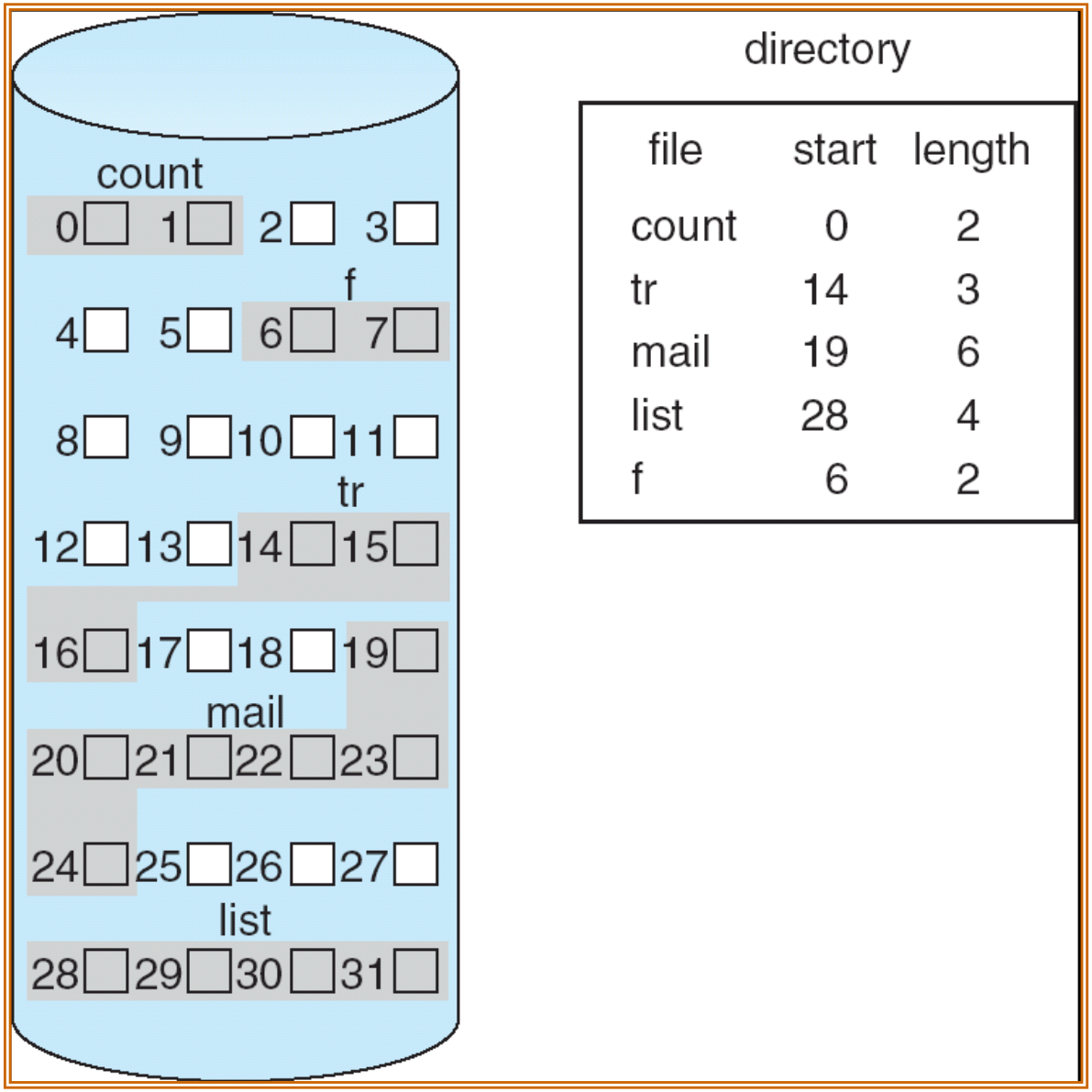
Asignación de bloques

- Los ficheros se almacenan en disco, y ocuparán uno o varios bloques del disco
- Se debe decidir cómo asignar los bloques a los ficheros:
  - Asignación contigua
  - Asignación enlazada (DOS/Windows)
  - Asignación indexada (Unix)

Asignación contigua

- Cada fichero ocupa un conjunto de bloques contiguos en el disco
- El FCB indica el primer bloque del fichero y el número de bloques
- Ventajas:
  - Óptimo en acceso directo
- Problemas:
  - Fragmentación externa
  - Crecimiento de ficheros

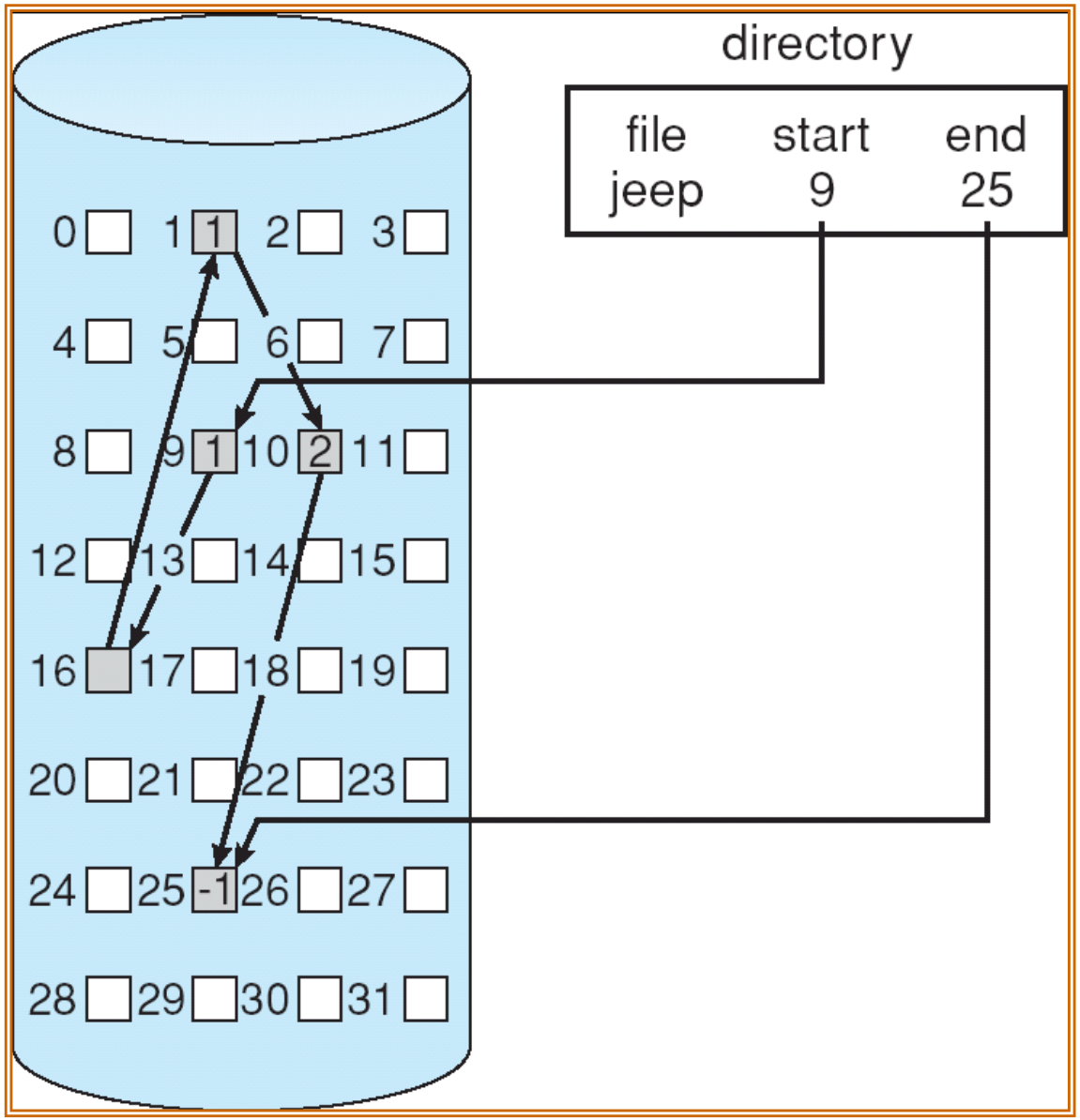
Asignación contigua



Asignación enlazada

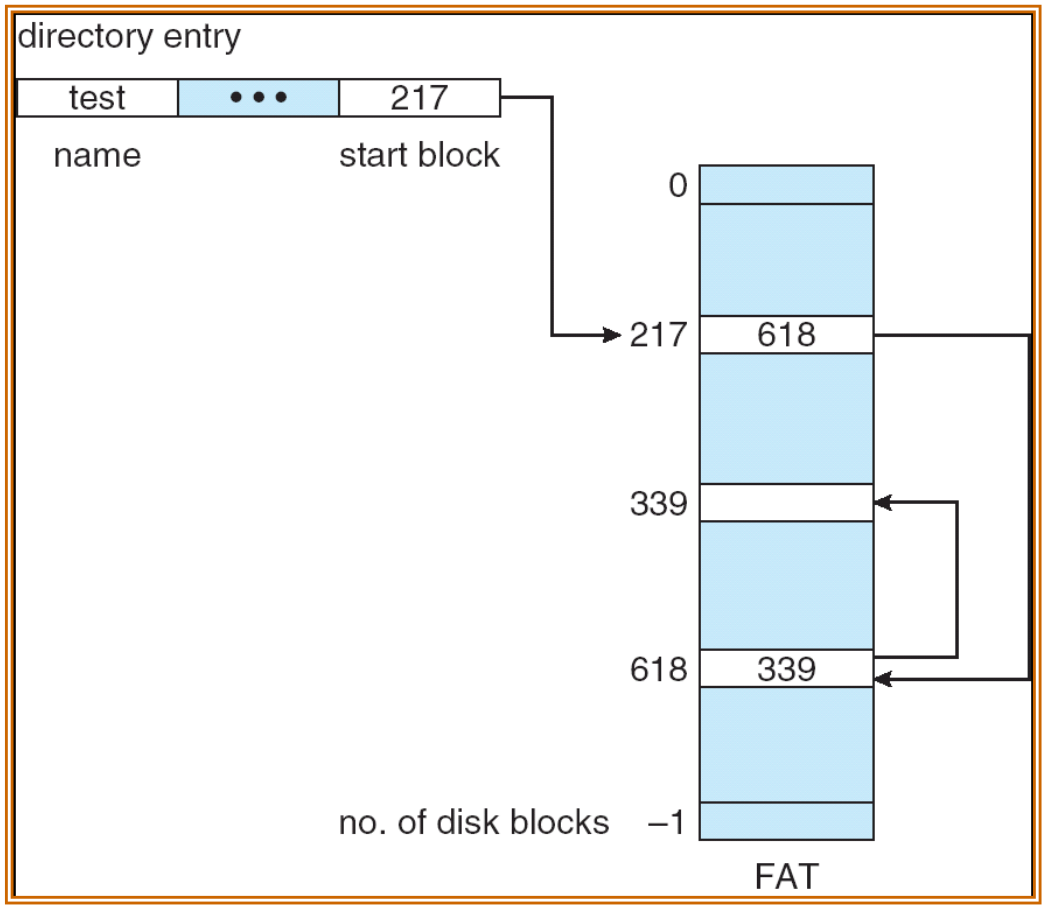
- Los bloques contienen punteros al próximo bloque dentro del fichero
- El FCB contiene el índice del primer bloque del fichero (opcionalmente también el del último)
- Ventajas:
  - Sin fragmentación externa
- Problemas:
  - Poco óptimo en accesos directos (a una posición específica del fichero)
  - Los punteros desperdician espacio

Asignación enlazada



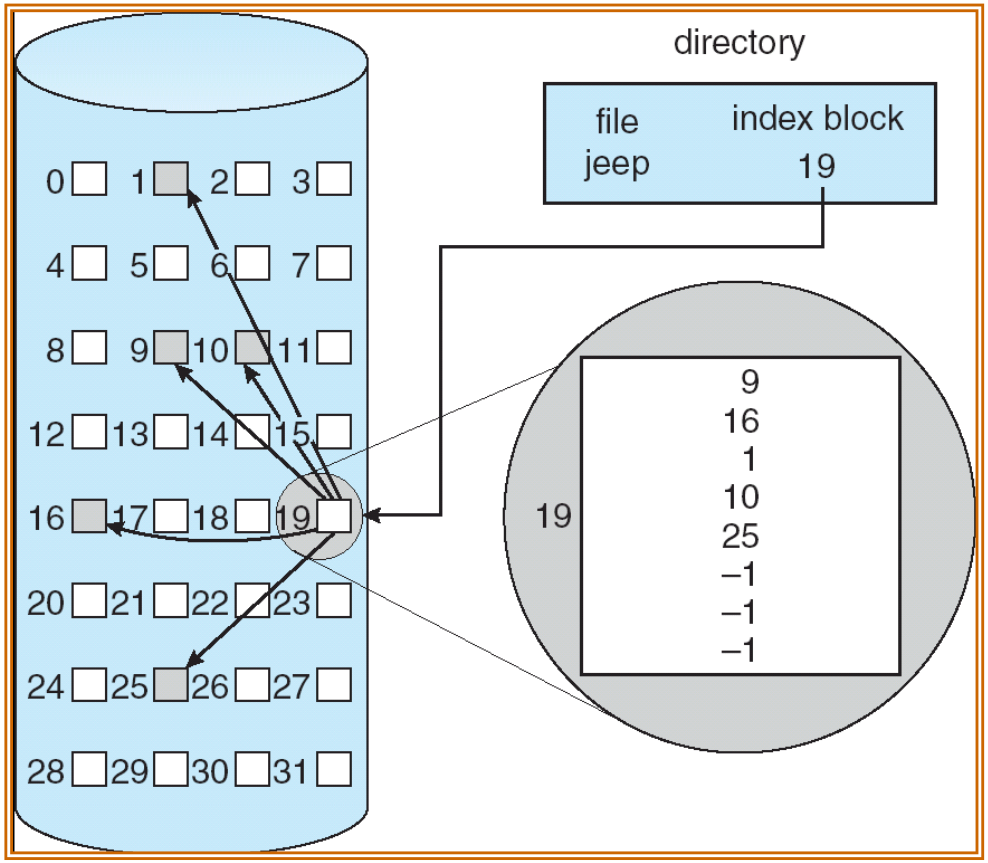
FAT, File Allocation Table

- Variante de asignación enlazada en DOS/Windows 9x
- Emplea una tabla que enlaza los bloques de cada fichero
- El FCB contiene el índice del primer bloque del fichero



Asignación indexada

- Se usa un bloque como índice por fichero
- El FCB contiene el número del bloque índice
- Ventajas:
  - Acceso directo óptimo
- Problemas:
  - Tamaño del bloque índice
  - Los ficheros pequeños ocupan al menos 2 bloques



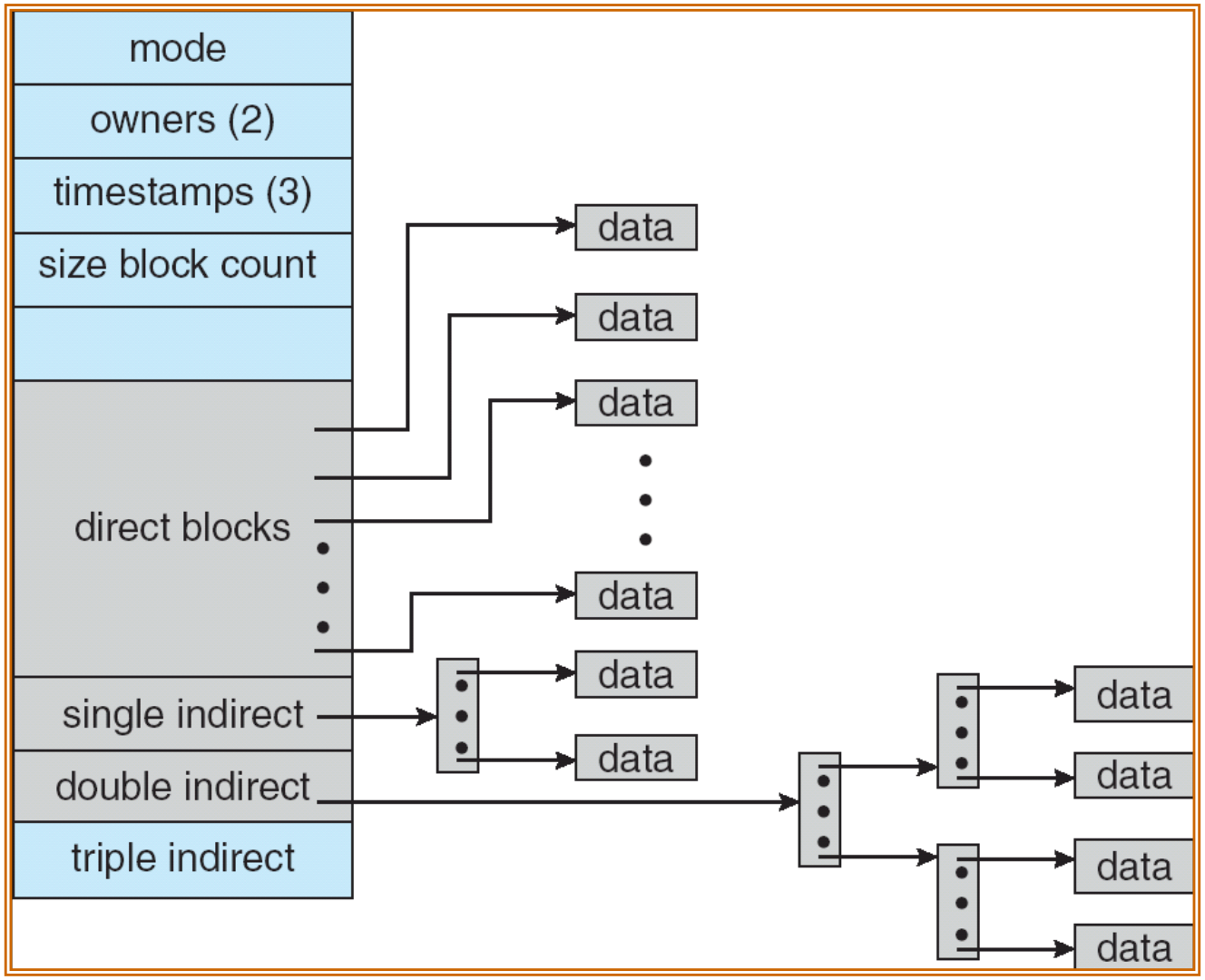
Asignación indexada - Esquemas

- Esquema enlazado:
  - Cada bloque índice ocupa un bloque de disco
  - La última entrada apunta al próximo bloque índice (si existe)
  - Problema: poco óptimo en acceso directo (accediendo directamente a una posición específica)
- Esquema multinivel:
  - El bloque índice de primer nivel contiene direcciones de bloques índice de segundo nivel
  - Los bloques de segundo nivel contienen las direcciones de los bloques del fichero
  - Problema: los ficheros pequeños ocupan mínimo 3 bloques: 2 bloques índice y 1 bloque de datos

Asignación indexada – Esquema

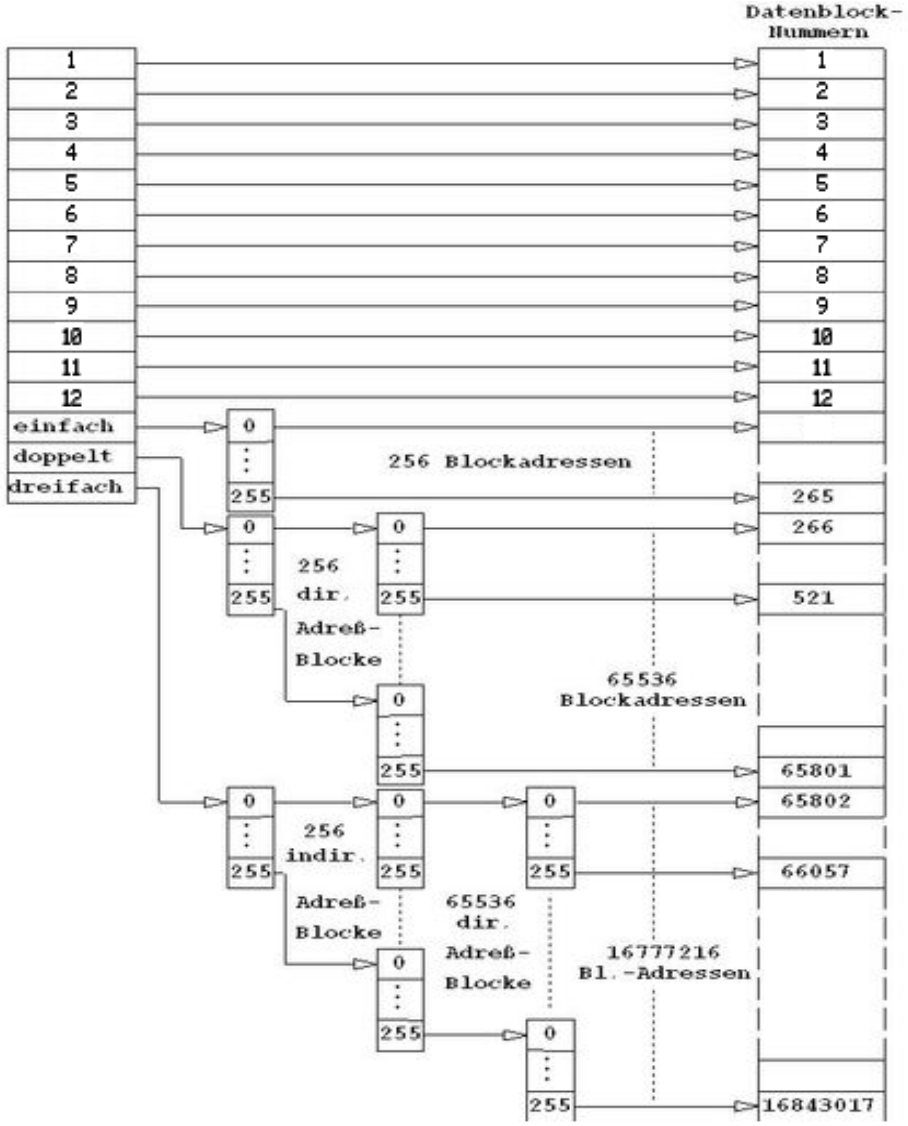
- Esquema combinado:
  - Sistema usado en Unix
  - El FCB contiene N direcciones de bloques
  - Las N-3 primeras direcciones indican los N-3 primeros bloques del fichero
  - La entrada N-2 apunta a un bloque índice indirecto de un nivel
  - La entrada N-1 apunta a un bloque índice indirecto de dos niveles
  - La entrada N apunta a un bloque índice indirecto de tres niveles

Asignación indexada – Esquema



Extended File System Ext2/Ext3

- El i-nodo mismo contiene 15 entradas de dirección de bloques
- Las primeras 12 entradas contienen referencias directas a bloques de datos
- Las 3 últimas contienen referencias indirectas:
  - 1 puntero para indirección simple.
  - 1 puntero para indirección doble.
  - 1 puntero para indirección triple.
- Un archivo puede contener 16 millones de bloques
  - El tamaño máximo posible de archivo depende del tamaño del bloque (p. ej. 4KB por bloque)



Fuente: http://pics.computerbase.de/lexikon/20358/500px-I-Node-Struktur.jpg

Sistemas actuales: Ext4

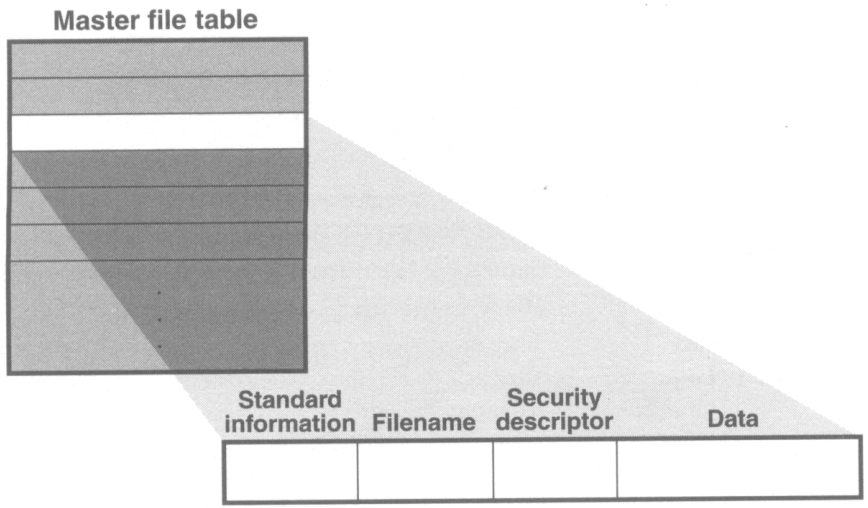
- Sistema de archivos transaccional (journaling).
  - Mejora compatible de ext3 en 2006.
  - En 2008 se elimina la etiqueta de "experimental" de ext4 en el kernel.
- Principales mejoras:
  - Permite la reserva de espacio en disco para un fichero
  - Soporte de volúmenes de hasta 1024 PiB (PebiByte).
  - Soporte añadido de *extents* (conjunto de bloques físicos contiguos), mejorando el rendimiento al trabajar con ficheros grandes y reduciendo la fragmentación.
    - Un *extent* simple en ext4 es capaz de mapear hasta 128 MiB de espacio contiguo con un tamaño de bloque igual a 4 KiB.
  - Mejoras en la velocidad de lectura y escritura.

Sistemas actuales: NTFS

- ***NTFS*** (New Technology File System): sistema de archivos de los sistema MS Windows a partir de Windows 2000.
- Pensado para reemplazar a la familia de FAT, eliminando todas sus limitaciones, y poder extenderlo fácilmente.
- Conceptualmente NTFS ve todo en el sistema como un archivo, incluyendo a los metadatos.
  - **MFT** (Master File Table): contiene la información de donde están los archivos y sus atributos.
  - Si se daña la MFT, los datos de todo el volumen se pierden.
- Utiliza mapa de bits para determinar los bloques libres en el disco.
- Indexa los directorios a través de árboles B+.

Master File Table NTFS

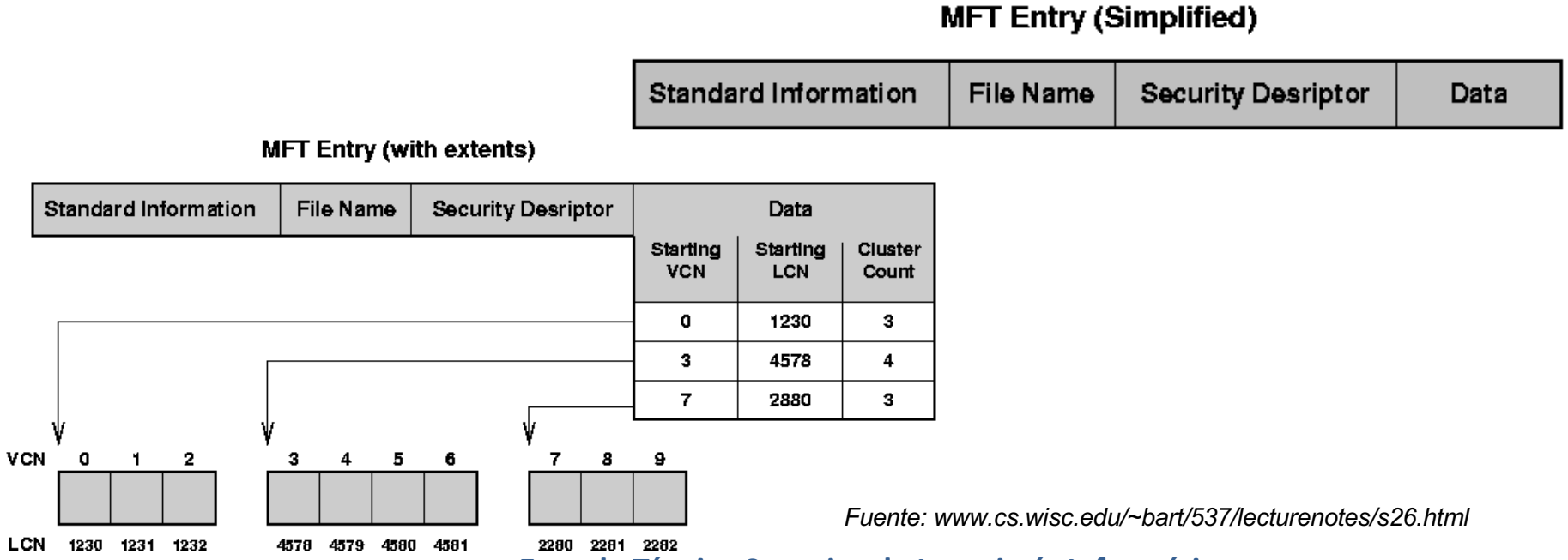
- Al formatear un volumen, se reserva espacio para que la MFT pueda crecer (alrededor del 12% del disco por defecto).
- Este espacio sólo se usa si no queda más espacio disponible en el disco.
- Si la MFT crece mucho, puede particionarse y colocar una porción en otro lugar del disco.
- Tamaño de una entrada de la MFT: entre 1.024 y 4.096 bytes
- Cada registro de la MFT contiene un encabezamiento o ***header*** para indicar:
  - Tipo de entrada.
  - Cantidad de bytes usados.
  - Contador de referencias.
  - Etc.



Fuente: http://dbserver.kaist.ac.kr/~yjlee/Courses/CS230/ntfs/NTFS-3.html

Archivos NTFS

- Los punteros a datos son referencias directas a *extents*.
- Un **extent** se identifica con 3 partes:
  - **VCN** (*Virtual Cluster Number*): número de *cluster* en el archivo en el que comienza el extent.
  - **LCN** (*Logical Cluster Number*): número del *cluster* del disco (físico) en el que comienza el *extent*.
  - **Largo:** cantidad de *clusters* que mide este *extent*.



Fuente: www.cs.wisc.edu/~bart/537/lecturenotes/s26.html

Sistemas actuales: XFS

- Creado por SGI (antiguamente Silicon Graphics Inc.) para Irix (Unix) en 1994.
- Actualmente, sistema de archivos journaling de 64 bits:
  - Los cambios primero son escritos a un diario o *journal* antes de que se actualicen los datos del disco.
- Diseñado para tratar grandes ficheros y para escalar grandes volúmenes:
  - Adecuado para servidores de video, ficheros multimedia, bases de datos, …
  - Se ha extendido en entornos de BigData y Cloud.
  - CentOS y RedHat proponen el uso de XFS para los puntos de montaje "/home" y "/“.
- Problemas cuando trata con muchos archivos pequeños (*small file problem*).

Sistemas actuales: BTRFS

- Sistemas de archivos copy-on-write anunciado por Oracle Corporation (con la participación de Red Hat, SUSE, Intel, entre otras) para Linux.
- Objetivo: sustituir a los sistemas de archivos ext3 y ext4, eliminando alguno de los fallos que tiene, como puede ser el tamaño máximo de ficheros.
- Su nombre deriva del nombre de las estructuras de datos Árbol B.
- Se centra en tolerancia a fallos y fácil administración.
- Características:
  - Almacenamiento de archivos basado en *extents*.
  - Tamaño máximo de archivo: 16 EiB.
  - Asignación dinámica de i-nodos.
    - No se fija un número máximo de i-nodos al crear el sistema de archivos.
  - Se realizan sumas de verificación sobre datos y metadatos.
  - Se puede comprobar el sistema de archivos sin desmontar.
  - Comprobación rápida del sistema de archivos si está desmontado.

Sistemas actuales: APFS

- APFS (Apple File System): desarrollado por Apple Inc.
  - Reemplaza a HFS+ (HFS Plus), también conocido como HFS Extended y Mac OS Extended.
  - Permite nombres de fichero de hasta 255 caracteres de longitud UTF-16.
  - Optimizado para discos SSD y memorias flash.
  - Escala su funcionamiento para ser usado tanto en Apple Watch como en Mac Pro.
  - Proporciona *checksums* para asegurar la integridad de los metadatos.
  - Números de i-nodo de 64 bits.
    - Número máximo de ficheros por volumen: 2^{63}.
  - Direccionamiento de bloque de 64 bits.
    - Tamaño máximo archivo: 2^{63} bytes = 8Eib.

Comparación

FS	Tamaño máximo archivo	Número máximo archivos	Tamaño máximo volumen	POSIX Permisos de archivo	Journaling	Extents	Sistema
XFS	8Eib	2^{64}	8Eib	Sí	Si	Si	Linux
ext4	16 GiB -16 Tib	2^{32}	1Eib	Sí	Sí	Si	Linux
NTFS	16 EiB	2^{32}	16 EiB	Sí	No	Si	Windows, Linux, macOs con ntfs-3g
BTRFS	16 EiB	2^{64}	16 EiB	Si	Si	Si	Linux
APFS	8Eib	2^{63}	8Eib	¿ ACL ?	Sí	Si	macOs Sierra