Tema 4 - Gestión de archivos

▼ Archivos

- Información relacionada almacenada en un dispositivo de almacenamiento secundario bajo un mismo nombre
- Necesario almacenar en dicho dispositivo toda la secuencia de bytes (normalmente por bloques)

▼ Necesario almacenar metadatos asociados

- Nombre
- Fecha de creación/modificación
- Permisos de acceso
- Estructuras de datos necesarias para poder gestionar bloques de datos

- Pueden existir archivos especiales → dispositivos o interfaz con núcleo del SO

▼ Directorios

- Permite agrupar distintos archivos bajo nuevo nombre → agrupaciones con arreglo a algún criterio
- Pueden contener directorios
- Se implementan como archivos especiales con información relativa a archivos/directorios que contienen

▼ Para cada archivo almacenan atributos

- Nombre
- Referencia a mecanismo para acceder al resto de información

- Gestión eficiente → estructuras complejas (árbol)

▼ Directorio raíz

- Nodo inicial de estructura tipo árbol
- Cada directorio puede contener archivos y otros directorios

▼ Gestión del espacio

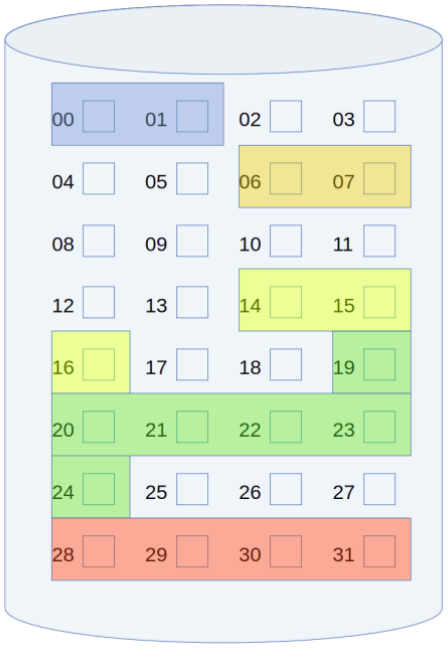
▼ Almacenamiento contiguo

▼ Ventajas

- Acceso eficiente a bloques tanto de forma secuencial como directo
- Asociación entre direcciones lógicas y físicas resulta trivial

▼ Desventajas

- Reserva de espacio → si no se conoce tamaño previo es problemática
- Fragmentación externa → procesos de compactación costosa



Directorio

Fichero	Inicio	Longitud
lista.odt	0	2
tmp.txt	14	3
a.out	19	6
core	28	4
video.mp4	6	2

▼ Almacenamiento enlazado

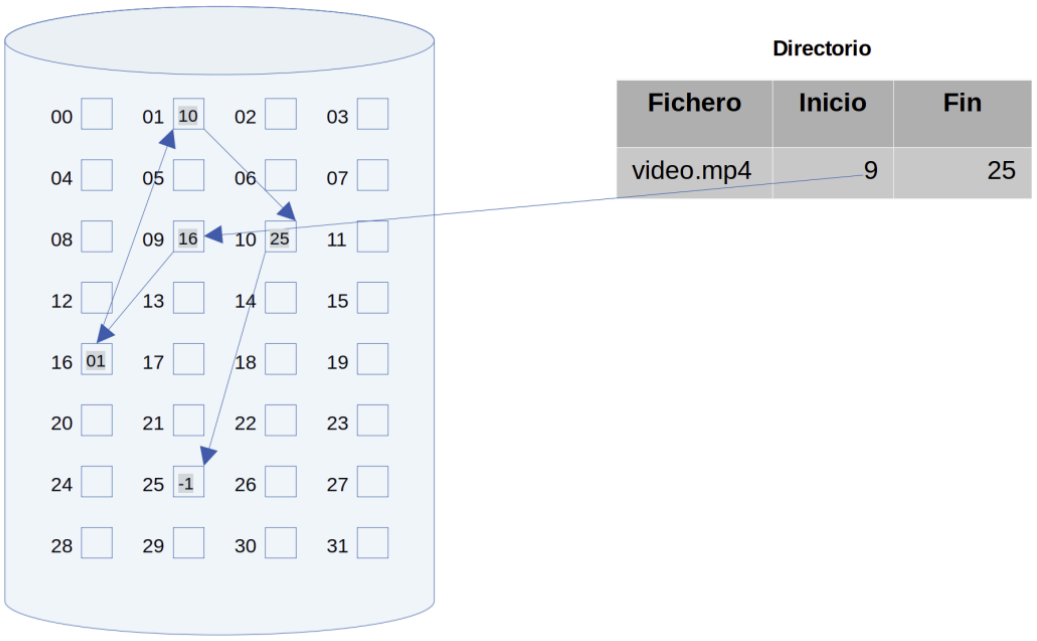
- Lista enlazada de bloques → sin correspondencia con bloques físicos contiguos

▼ Ventajas

- Se evita fragmentación externa
- Archivos pueden crecer dinámicamente sin necesidasd de realizar operaciones de compactación
- Acceso a todos los datos a partir de un puntero al primer bloque

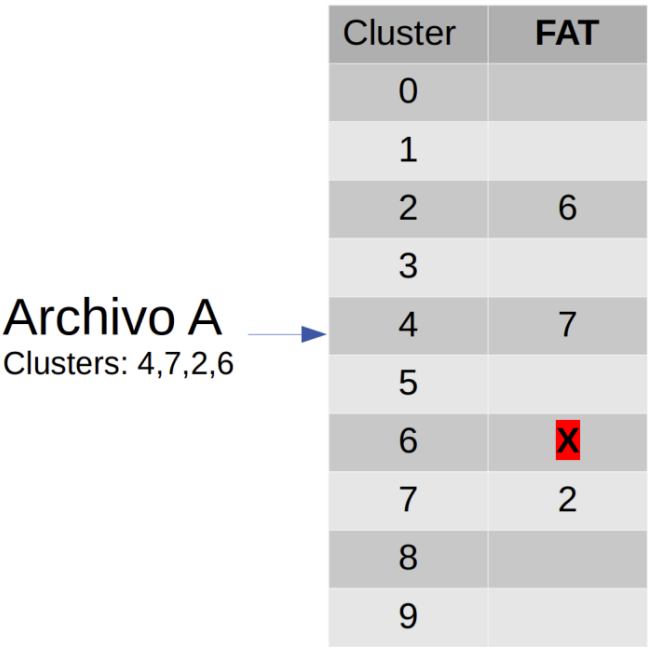
▼ Desventajas

- Acceso directo ineficiente
- ▼ Espacio requerido por punteros de enlace no es despreciable
  - Puede paliarse haciendo que la lista contenga grupos de bloques (clusters)



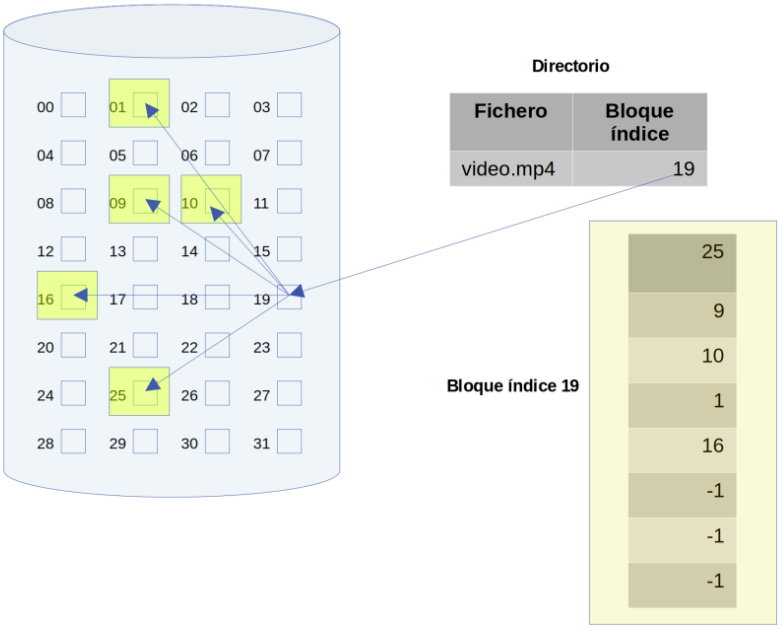
▼ Ejemplo → FAT

- Variación de almacenamiento enlazado
- Unidad mínima de almacenamiento → cluster
- ▼ Tabla (FAT)
  - No hay entrada por cada cluster
  - Índice de cada entrada → cluster (su contenido es siguiente cluster)
  - Para cada archivo se guarda el número del primer cluster
  - Se mantiene copia en el mismo sistema de archivos



▼ Almacenamiento indexado

- Bloque índice que guarda punteros a bloques de datos
- Entrada i-ésima contiene punto a i-eśimo bloque de datos



▼ Ventajas

- Acceso directo eficiente
- No produce fragmentación externa

▼ Desventajas

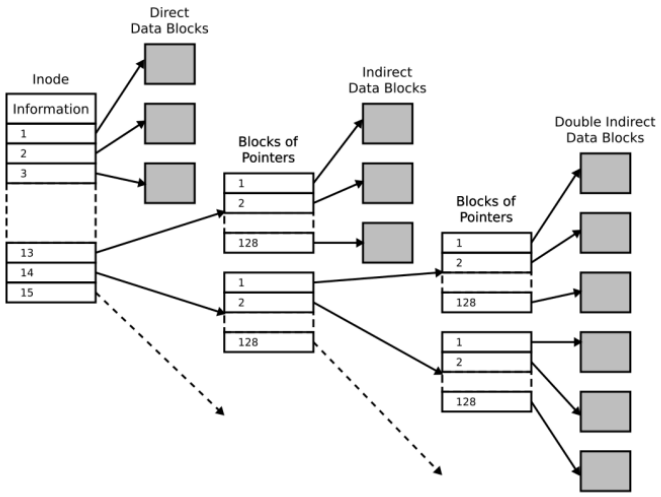
- Bloques índice grandes respecto a tamaño de archivo → no aprovechan espacio de punteros a bloques no utilizados
- Bloques índice pequeños → limitación importante en tamaño máximo de archivo

▼ Para paliar desventajas

- Bloques índice enlazados

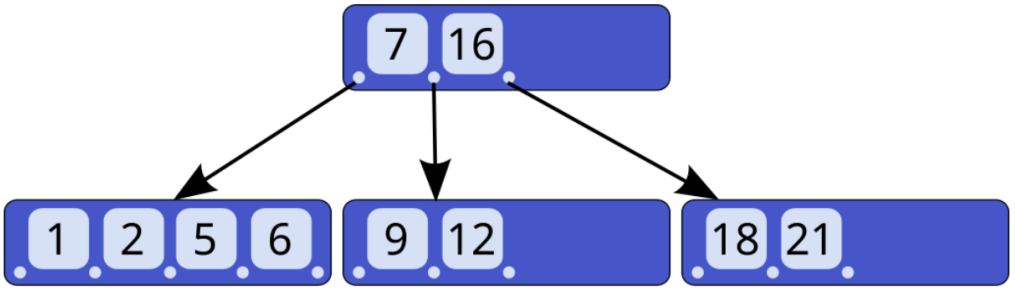
▼ Bloques índice multinivel

- Bloques índice apuntan a otros bloques índice
- Son los que referencian datos
- Se puede elegir nivel máximo de indirección



▼ Árboles balanceados

- Agrupaciones de bloques contiguos → extents
- Acceso eficiente a extents → árbol balanceado



▼ Espacio libre

- Gestión más simple → normalmente no es necesario buscar un bloque libre concreto

▼ Bitmap

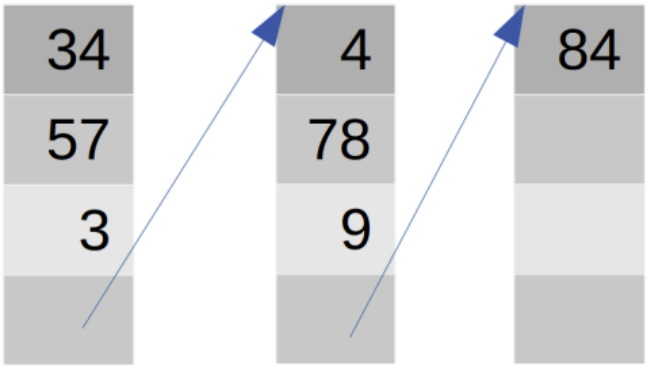
- Cada bloque se representa con un bit
- Se puede encontrar eficientemente (secuencias de) bloques libres analizando en bloque los bits de una palabra
- Se suele mantener en memoria principal por razones de eficiencia

▼ Lista enlazada

- Por cada bloque libre hay que almacenar un puntero al siguiente

▼ Lista enlazada con agrupación

- Lista enlazada guarda conjuntos de bloques libres
- Reduce cantidad de accesos y punteros para representar espacio libre



▼ Lista de bloques contiguos

▼ Lista con pares

- Número de bloque
- Contador de bloques libres adyacentes

▼ Sistema de archivos

▼ Características

- Parte del SO encargada de gestión de almacenamiento permanente de información (SO + usuarios)
- Abstracción de archivo/directorio/otros elementos relacionados
- Soporte para varios tipos de sistemas de archivos → cada uno implementa abstracciones de forma distinta
- Formateo → creación de nuevo sistema de archivos sobre un dispositivo

▼ Elementos

- Estructura y características de metadatos que definen el sistema de archivos
- Metadatos relativos al mantenimiento de la consistencia
- Mecanismos que permiten establecer restricciones de acceso
- Programas para creación/reparación/generación de metadatos

▼ Tipos

▼ Basados en disco

- Orientados al almacenamiento de datos

▼ Virtuales

- Generador por kernel del SO → interfaz para interactuar y obtener información
- No requieren espacio de almacenamiento secundario

▼ De red

- Permiten compartir datos en red como si los datos estuvieran almacenados localmente

▼ Particiones

▼ Características

- Divisiones a nivel lógico de una unidad de almacenamiento
- Cada partición será tratada por el SO como disco lógico
- Permiten tener instalados varios sistemas operativos a la vez

- Cada partición puede tener un sistema de archivos completamente distinto
- Información almacenada en zona específica al comienzo de cada disco

▼ Estándar UEFI

- Interfaz entre firmware y SO reemplazando interfaz BIOS

▼ Tabla de particiones GPT

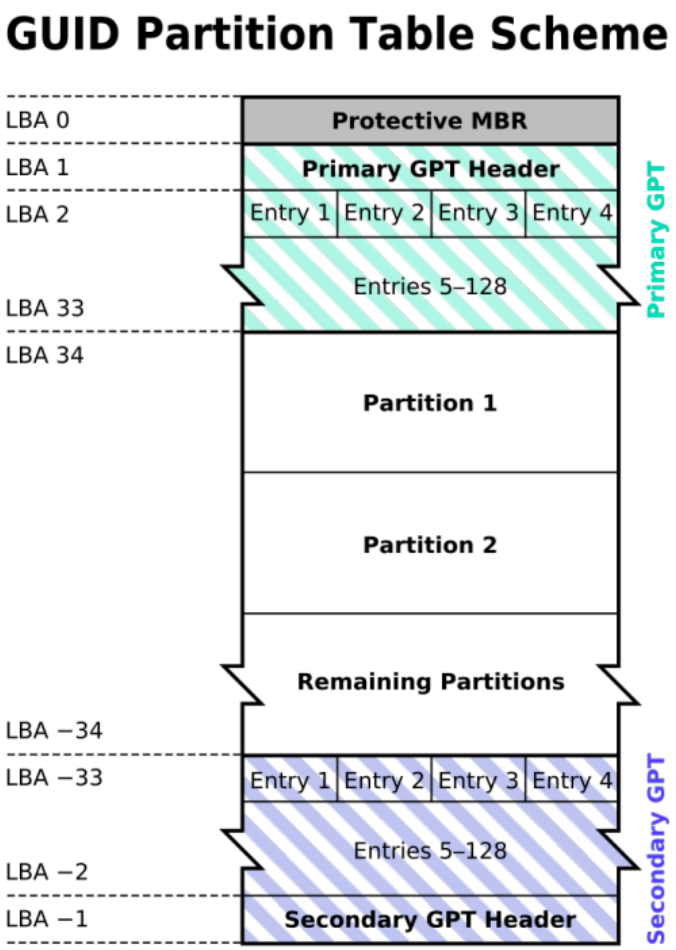
- Esquema para superar limitaciones anteriores
- Permiten superar centena de particiones
- Proporcionan mucho más espacio para programas cargadores de distintos SO
- Cada partición tiene un único identificador UUID

▼ Partición especial ESP

- Contiene cargadores de los SO presentes
- Contiene sistema de archivos basado en FAT
- Facilita tener varios SO instalados en mismo disco sin interferir cargadores de los mismos

▼ Restringe ejecución de núcleos de SO que no tengan firma válida

- Evita que se pueda alterar de forma maliciosa algún elemento de la cadena de arranque



▼ Ejemplos

▼ ext2

▼ Características

- Sistema de archivos por defecto de Linux
- Información global guardada en superbloque
- Se divide en grupos de bloques → cada grupo gestiona cantidad de bloques de almacenamiento contiguos

▼ Se intenta almacenar en el mismo grupo elementos relacionados

- Directorio y sus archivos
- Directorio y su directorio padre → maximizar localidad

- En algunos grupos se almacenan copias del superbloque en previsión de que este pueda corromperse

▼ Contenido de cada grupo de bloques

- Tabla de descripción del grupo
- Bitmap de bloques libres/ocupados
- Bitmap de i-nodos libres/ocupados
- Tabla de i-nodos del grupo
- Bloques contiguos de almacenamiento
- Copia de superbloque (opcional)

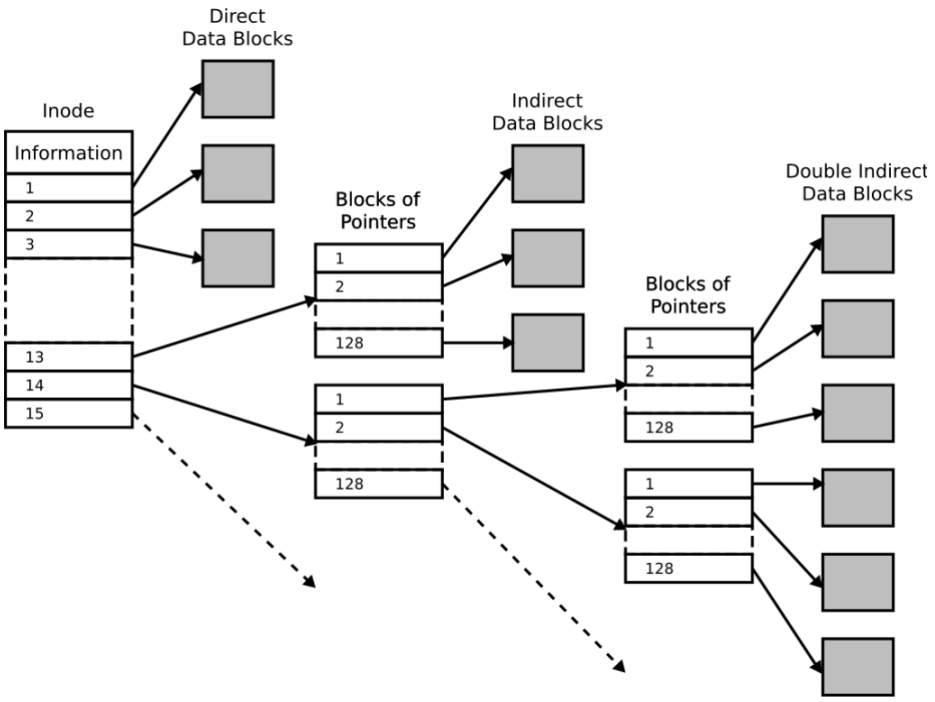
▼ i-nodo

- Estructura de datos utilizada para representar archivos
- Información sobre propietario, tipo, permisos, tiempos de acceso/modificación, contador de enlaces

▼ Contiene enlaces a bloques de datos

▼ Tipos de enlaces

- Directos
- Indirectos
- Bidireccionales



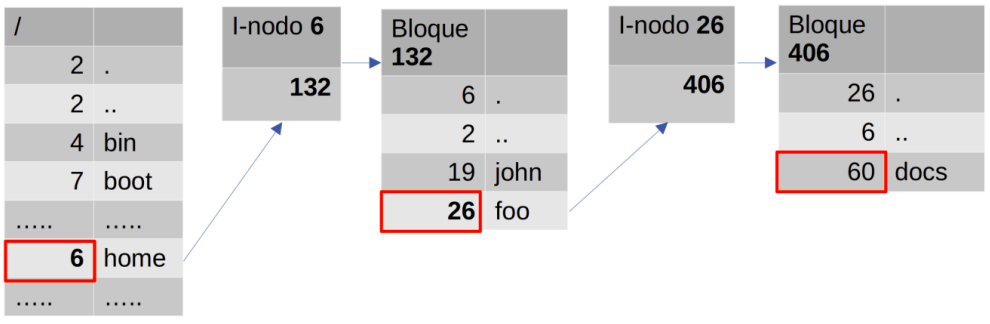
▼ Directorios

- Almacenan número de i-nodo que representa cada archivo/directorio contenido

▼ Información en bloque de datos

▼ Archivos especiales → datos almacenados

- Información de nombre
- i-nodo de otros archivos o directorios



▼ ext3

▼ Principal cambio frente a ext2

- Transacción al sistema de archivos → evitar cambios parciales del sistema de archivos por bloqueos o apagados

no controlados

▼ Registro circular de todas las operaciones en una estructura → journal

▼ Objetivos

- Permite realizar recuperación ordenada de transacciones realizadas solo parcialmente
- Pretende mantener en todo momento integridad del sistema
- Pretende reducir tiempo de comprobación del estado del sistema de archivos ante un apagado no realizado limpiamente

▼ Modos de journaling

▼ Journal

- Se registran metadatos y datos
- Mayor nivel de integridad
- Mayor coste

▼ Ordered

- Se registran sólo metadatos

▼ Funcionamiento

- Se escriben los datos en sistema de archivos
- Se registran metadatos en diario
- Se transfieren metadatos al sistema de archivo

▼ No hay problemas ante interrupciones al crear o añadir información a un archivo

- Pueden haberse escritos datos y no quedar registrados en sistema de archivos
- En caso de sobre-escritura → archivo parcialmente sobre-escrito si se interrumpe la operación

▼ Writeback

- Similar a ordered
- Los dos primeros pasos (escribir y registrar) pueden ocurrir en cualquier orden
- Modo más rápido
- Modo menos seguro
- Puede registrarse un aumento de tamaño de archivo sin escribir datos → contenido basura

▼ ext4

▼ Optimizaciones añadidas sobre ext3

▼ Bloques flexibles

- Grupos de grupos de bloques

▼ Funciones de cada grupo flexible

- Guarda estructuras de todos los grupos que contiene en el primer bloque
- Deja el resto de espacio para bloques de datos
- Aumenta espacio de almacenamiento contiguo para datos
- Permite acceso eficiente a metadatos de varios grupos

▼ Extents

- Conjunto de bloques contiguos representados en una única estructura
- Sustituyen a los mapas de bloques indirectos
- Se almacenan en estructura de árbol → acceso eficiente a partir de tamaño de archivo determinado

▼ NTFS

- Sistema de archivos nativo de los sistemas operativos de Windows

▼ Unidad básica de almacenamiento → cluster

- Independencia de tamaño de bloque físico de dispositivo

- Toda la información se guarda en archivos (incluyendo metadatos)
- ▼ MFT
  - Corazón del sistema de archivos NTFS
  - Conjunto de entradas → cada entrada se corresponde a un archivo
  - Se guardan atributos y valores de esos atributos para cada archivo
  - Cada archivo puede tener cuantas entradas sean necesarias
- ▼ Extents
  - Datos asociados a otro archivo → otro atributo más
  - ▼ Atributos que no caben en MFT → extents enlazados
    - Intervalos de clusters que no residen en MFT
    - Se definen por un cluster de inicio y su longitud

▼ Otras estructuras de almacenamiento

- ▼ Volúmenes lógicos
  - ▼ Características
    - Redimensionado sencillo de particiones lógicas generadas → adaptación a necesidades futuras
    - ▼ Almacenamiento lógico total contiguo
      - Aparentemente contiguo → en realidad puede estar repartido entre varios dispositivos
    - ▼ Instantáneas
      - Copia exacta del estado de una partición lógica en un momento dado
      - Permite volver a un estado definido deshaciendo los cambios producidos después de la creación de esa instantánea
  - ▼ LVM en Linux
    - ▼ Elementos

▼ Volumen físico (PV)

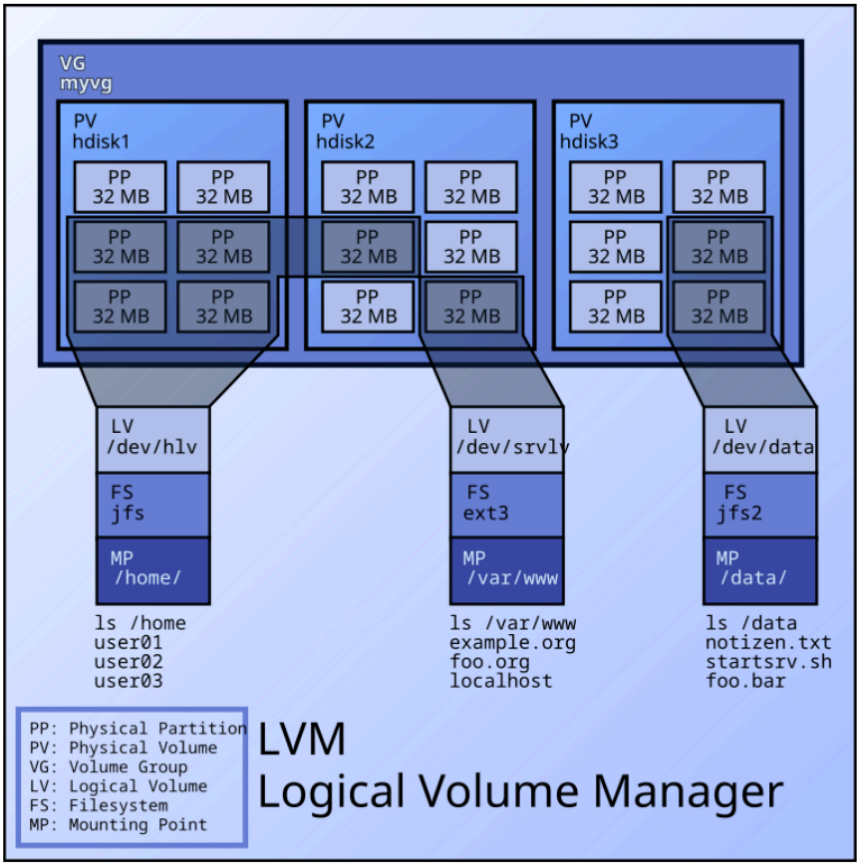
- Particiones/dispositivos completos
- Internamente se dividen en extents físicos (PE)

▼ Grupo de volúmenes (VG)

- Agrupa PV
- Permite creación de volúmenes lógicos
- Proporciona espacio de almacenamiento único → sustentado por una serie de PV

▼ Volumen lógico (LV)

- Particiones lógicas que utilizan espacio proporcionado por VG
- Pueden ser redimensionadas
- Pueden estar soportados por varios PV



▼ RAID

▼ Características

- Grupo de discos que se integran para conseguir tolerancia a fallos o mayor rendimiento
- Ningún nivel sustituye a una política de copias de seguridad → sólo permiten evitar interrupción del servicio prestado

▼ Se intenta que no se pierda información → información de paridad

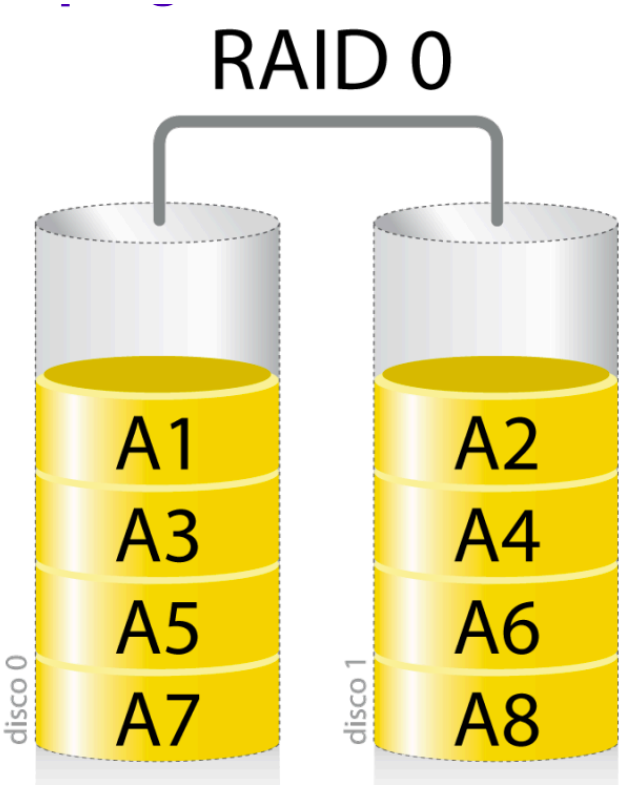
- Si perdemos datos de paridad → se mantienen los datos
- Se se pierden datos → se pueden reconstruir con los que quedan y la información de paridad

▼ Data striping

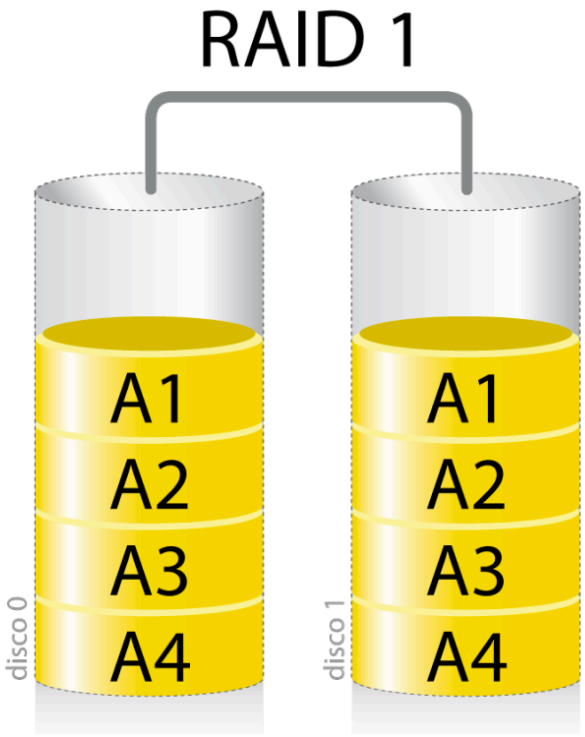
- Usado por todos los niveles menos RAID 1
- Segmentación del espacio lógicamente contiguo y secuencial en stripes de un determinado tamaño
- Stripes repartidos cíclicamente en distintos dispositivos → acceso a stripes consecutivos en paralelo
- Bloques de datos de un fichero acaban repartidos entre varios stripes en varios discos

▼ Niveles

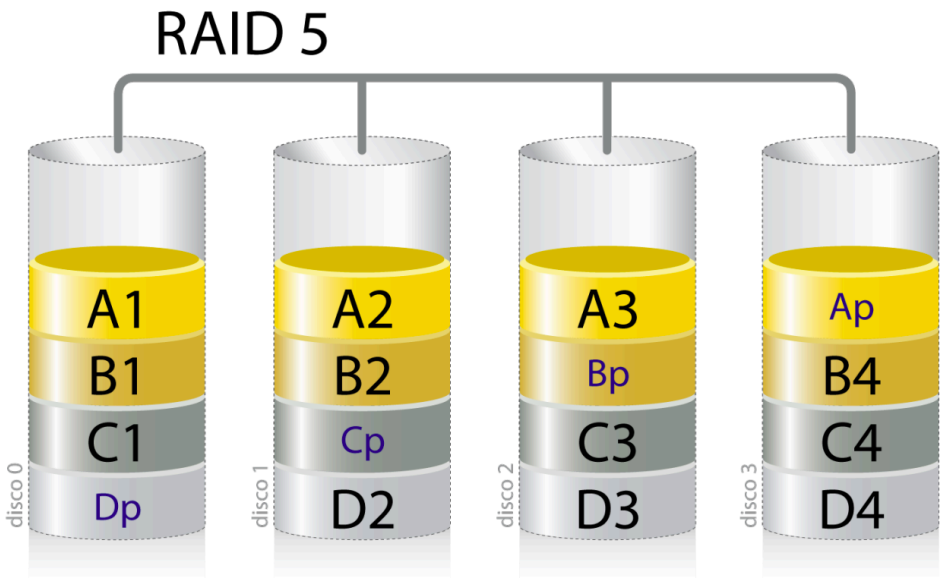
▼ RAID 0 → división de datos



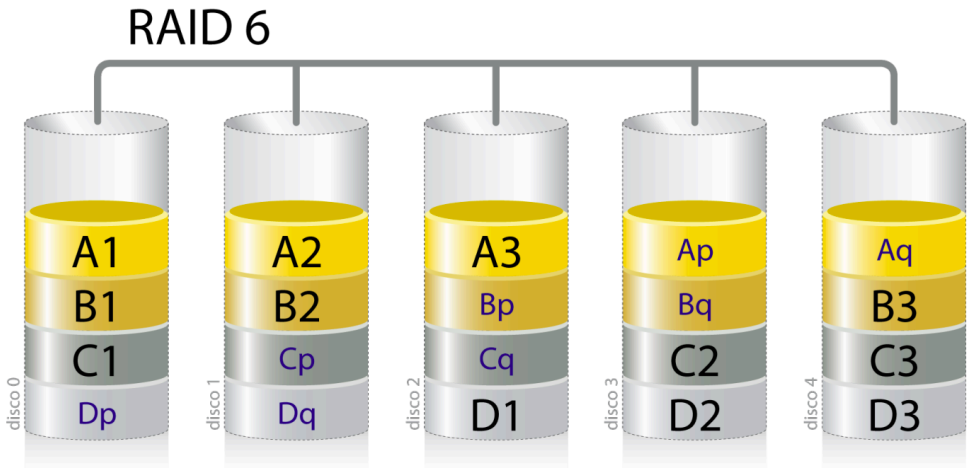
▼ RAID 1 → copia de datos



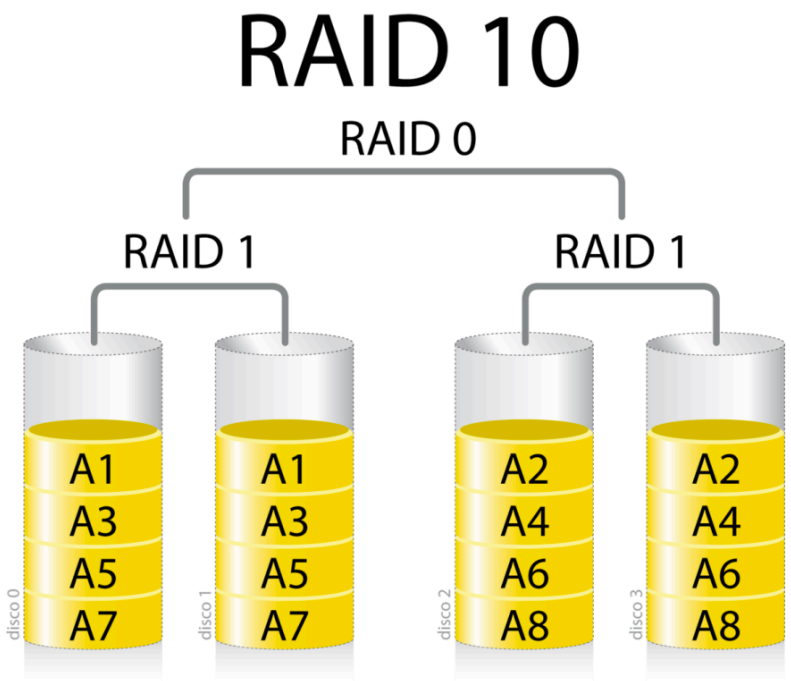
▼ RAID 5 → división con paridad



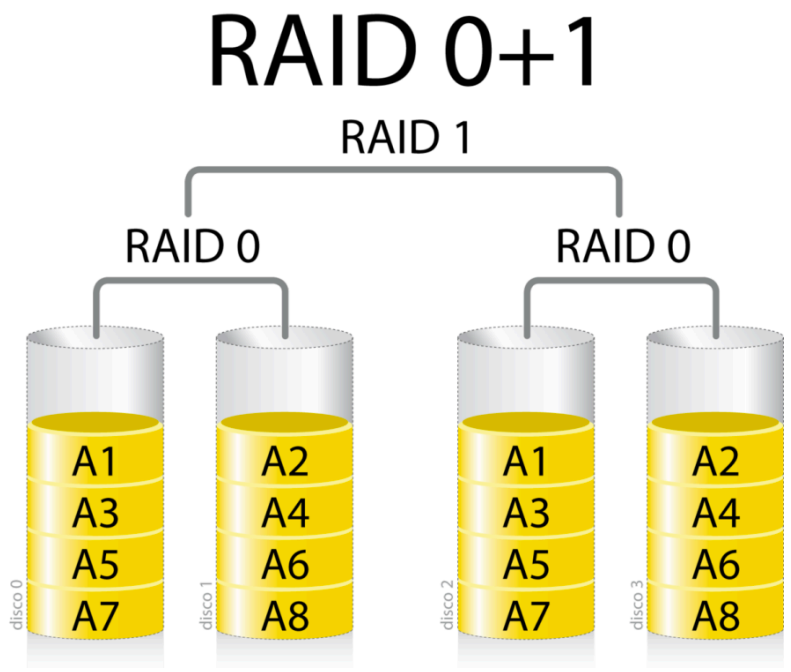
▼ RAID 6 → división con doble paridad



▼ RAID 10 → RAID 0 sobre dos RAID 1



▼ RAID 0+1 → RAID 1 sobre dos RAID 0



▼ Cifrado

▼ SO ofrecen mecanismos para cifrar datos almacenados

- Impedir accesos no autorizados a información sensible
- Sin este sistema, si el SO no está ejecutando → acceso directo al hardware supone acceso sin restricciones a información

- ▼ Posibles implementación
  - En el propio sistema de archivos
  - ▼ En un módulo independiente
    - Se ejecuta por debajo del sistema de archivos
    - Ofrece directamente un dispositivo virtual → cifrado transparente a sistema de archivos
- ▼ LUKS
  - Utilizado para realizar cifrado completo de dispositivos
  - Proporciona dispositivo lógico cifrado → permite crear cualquier sistema de archivos
- ▼ BitLocker
  - Cifrado completo de dispositivos en SO de familia Windows

▼ Device mapper

- Utilizado por LVM, RAID y LUKS
- Permite establecer correspondencias entre dispositivos de bloques físicos y dispositivos de bloques lógicos

▼ Volúmenes lógicos/espacios de almacenamiento RAID/dispositivo cifrado → dispositivos de bloques

- Permiten crear sistema de archivos sobre ellos como si fuesen dispositivos de bloques simples

▼ Tendencia a incluir en sistemas de archivos

- Funcionalidades RAID
- Gestión de volúmenes lógicos
- Cifrado

▼ Integración de distintos sistemas de archivos → VFS en Linux

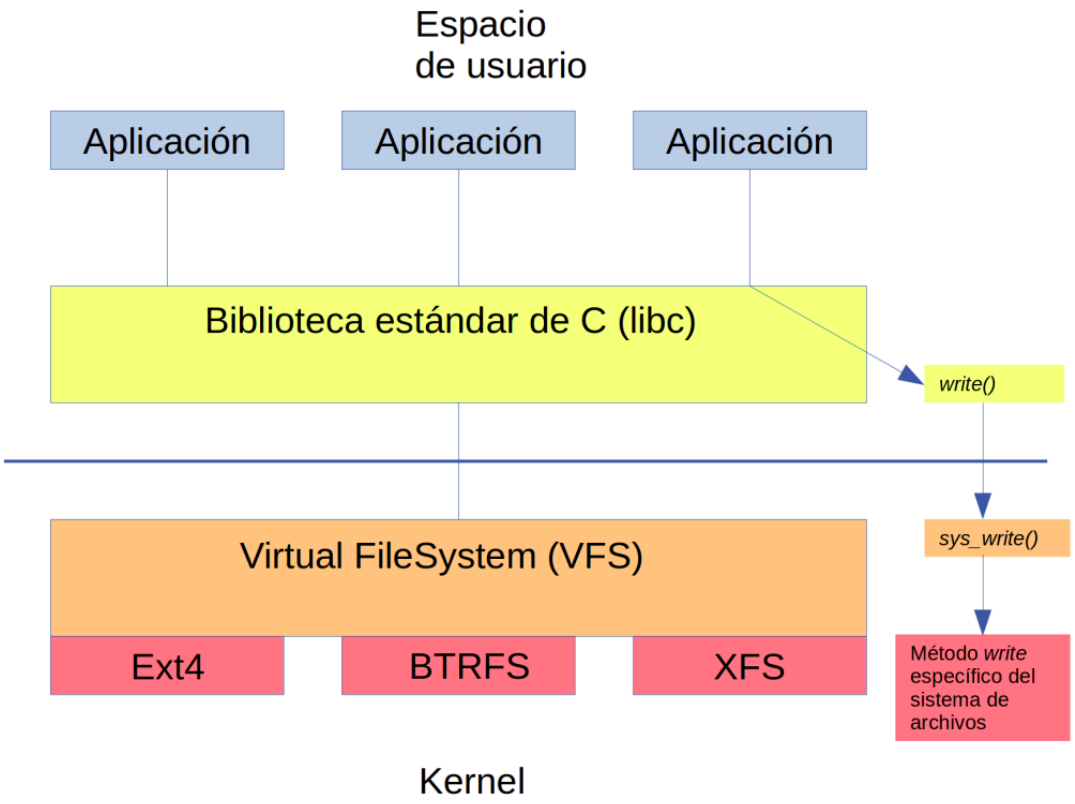
- Capa de abstracción al espacio de usuario
- Define comportamiento que debe implementar cualquier sistema de archivos

- Programas de usuario, a través de biblioteca estándar, pueden usar uniformemente distintos sistemas de archivos

▼ Todos los sistemas de archivos deben proporcionar implementación de funciones de VFS

  - Interfaz común con distintas implementaciones en cada sistemas de archivos

- Enfoque utilizado en diseños orientados a objetos



▼ Ejercicios

▼ En la siguiente figura se representa una tabla FAT. Al borde de sus entradas se ha escrito, como ayuda de referencia, el número correspondiente al bloque en cuestión. También se ha representado la entrada de cierto directorio. Como simplificación del ejemplo, suponemos que en cada entrada del directorio se almacena: Nombre de archivo/directorio, el tipo (F=archivo, D=directorio), la fecha de creación y el número del bloque inicial.

Nombre	Tipo	Fecha	Bloque

DATOS	F	8-2-90	3
DATOS1	F	1-3-90	1
DATOS2	F	2-3-90	2
D	D	3-3-90	8
CARTAS	F	13-3-90	9

Bloque	Siguiente bloque
1	*
2	4
3	15
4	5
5	*
6	7
7	*
8	*
9	10
10	11
11	12
12	*
13	
14	
15	6
16	
17	
18	

▼ Si usamos un Mapa de Bits para la gestión del espacio libre, especifique la sucesión de bits que contendría respecto a los 18 bloques del ejercicio anterior.

Bloque	Estado	Bit
1	Ocupado	1
2	Ocupado	1
3	Ocupado	1

4	Ocupado	1
5	Ocupado	1
6	Ocupado	1
7	Ocupado	1
8	Ocupado	1
9	Ocupado	1
10	Ocupado	1
11	Ocupado	1
12	Ocupado	1
13	Libre	0
14	Libre	0
15	Ocupado	1
16	Libre	0
17	Libre	0
18	Libre	0

- 1111 1111 1111 0010 00

▼ El espacio libre en disco puede ser implementado usando una lista encadenada con agrupación o un mapa de bits. La dirección en disco requiere D bits. Sea un disco con B bloques, en que F están libres. ¿En qué condición la lista usa menos espacio que el mapa de bits?

▼ Cuando hay pocos bloques libres

▼ Mapa bits

- Necesita un bit por cada bloque
- Tamaño total → B

▼ Lista enlazada

- Cada bloque libre requiere una dirección para apuntar al siguiente bloque libre
- Tamaño total→ F * D