

# Capítulo 4 Gestión de la información

# Sistema de Archivos

Necesitamos almacenar y recuperar información

Tres condiciones:

Almacenar gran cantidad de información.

Debe sobrevivir al procesos que la utilizan

Varios procesos pueden tener acceso concurrente a la información.

Solución:

Almacenamiento de la información en **discos y otros medios externos**.

Se almacenan en unidades llamadas **archivos**.

Un archivo sólo debe desaparecer por eliminación explícita por parte del usuario.


La **estructura, nombre, forma de acceso, uso, protección e implementación** son tarea del **Sistema Operativo**.

# Sistema de Archivos

Cada **sistema operativo** dispone de su propio **sistema de archivos** diferente

El objetivo y función de todos ellos es el mismo:

permitir al usuario un manejo fácil y lógico de sus archivos abstrayéndose de las particularidades de los dispositivos físicos empleados.



# Directorios y Archivos

En un sistema de archivos hay dos tipos fundamentales de objetos:

- los **directorios** (carpetas en Windows)

- los **archivos**.

Los **archivos** son los objetos encargados de contener los **datos**

Los **directorios** son los objetos cuya misión principal es permitir una mayor **organización** de los archivos dentro del disco.

# Directorios y Archivos

Un **directorio** es un **contenedor** que puede contener

- archivos

otros directorios dentro de él → **subdirectorios**

Se crea una **jerarquía en forma de árbol**

simplifica enormemente la tarea de organizar y estructurar los archivos dentro de un disco.



En realidad lo que un directorio contiene:

no son otros directorios ni otros archivos tal cual

sino la **información** necesaria sobre dichos archivos o directorios

generalmente la **posición del sector** del disco en el que comienzan

A los directorios también se les denomina **carpetas**

# Archivos

Los archivos son un mecanismo de abstracción.

- almacenan información en el disco

- se puede volver a leer más adelante

- el usuario no tiene que preocuparse por :

  - la forma de almacenamiento de la información

  - lugar físico

  - funcionamiento real de los discos.

# Nombre de un Archivo

Las reglas para nombrar los archivos varían de un sistema de archivo a otro

- Todos permiten cadenas de 8 caracteres

- Algunos permiten mayor longitud

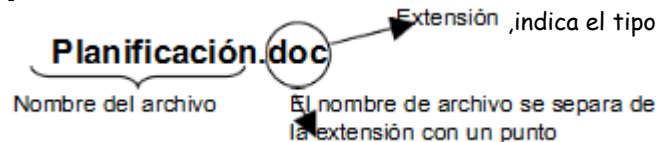
  - UNIX, últimas versiones de Windows

- Algunos diferencia entre mayúsculas y minúsculas (**case sensitive**)

  - UNIX si diferencia

  - DOS y Windows no diferencian

La mayoría tienen dos partes para el nombre (separadas por un punto ".")



Normalmente admite desde la A hasta la Z, del 0 al 9 y algunos caracteres especiales (^, \$, !, #, %, &, {, }, (, ), -, \_, ...).

# Atributos de un Archivo

Califican al archivo

Varían de un sistema a otro

## DOS

S (System): del sistema

H (Hidden) : oculto

R (Read only): solo lectura

A : Archivo

Fecha: fecha de creación o última modificación

Hora

Tamaño

Bytes	8	3	1	10	2	2	2	4
	Nombre del archivo	Extensión	Atributos	Reservado	Hora	Fecha	Nº del primer bloque	Tamaño

## UNIX

Usuario y grupo del usuario que crea el fichero

Permisos de estos y del resto de los usuarios (lectura, escritura, ejecución)

Tipo de archivo: normal, directorio, enlace

## WINDOWS

+ DOS

Comprimido o cifrado



# Los comodines

- ▶ Cualquier SA nos facilita poder nombrar un conjunto de ficheros con nombres similares, así es más fácil la ejecución de cualquier comando
- ▶ Hay dos tipos de caracteres comodines
  - "\*"• Sustituye todos los caracteres
  - "?"• Sustituye 1 sólo carácter
  - Ejemplos
    - \*data
      - Que acabe en data
      - data, adata, file.data
    - note\*
      - Que empiece por note
      - note, note1.doc, note.txt
    - \*raf\*
      - Que contenga raf
    - memo?
      - Que empiece por memo + 1 cter
      - memo1, NO memo.1
    - \*old?
      - Que acabe en old + 1 cter
      - file.old1

# Tipos de archivos

- ▶ Los tipos de archivo se diferencian en dos tipos bastante definidos:
  - **Ejecutables**
    - han sido programados bajo algún **lenguaje** específico para realizar **acciones** y rutinas por sí mismos.
    - Ejemplo: **.EXE**, **.COM**, **.BAT**, **.DLL**, etc.
  - **De datos**
    - Principalmente, contienen **datos**, y necesitan de una **aplicación específica** para ser abiertos.
    - Ejemplo: **.TXT**, **.DOC**, **.XLS**, etc...
  - Una aplicación: archivo ejecutable + ficheros necesarios

# Categorías según la extensión

## ▶ DE SISTEMA

- Necesarios para el funcionamiento interno del Sistema Operativo
  - Y de los diferentes programas que trabajan en él.
  - No está recomendado moverlos, editarlos o
  - Ejemplos:
    - .BAT, .BIN, .COM, .DLL, .SYS, INI, INF, MSI ...

# Categorías según la extensión

## ► DE AUDIO

- Los archivos de audio son todos los que contienen sonidos (no solo [música](#)).
- Extensiones: indican formato de compresión utilizado para convertir el [sonido](#) real en digital.
  - .CDA:
    - pista de audio digital de un Cd de música.
    - Haciendo clic sobre él se lanza el Reproductor de CDs de Windows.
  - .MID:
    - archivo de música MIDI (Interfase Digital de Instrumento Musical).
    - El software de instalación de la mayoría de las tarjetas de sonido dispone de un módulo para ejecutar archivos de sonido .mid.
  - .MP3:
    - archivo de audio comprimido bajo norma MPEG.
    - Se ejecuta con aplicaciones como Winamp, Xing MPEG Player o Real Player, entre otros.
  - .RA:
    - archivo de sonido Real Audio.
    - Se ejecuta con la aplicación Real Player.
  - .SND:
    - archivo de secuencia de sonido.
    - Windows lo ejecuta automáticamente con su aplicación Rundll32.
  - .WAV:
    - sonido de onda de Windows, se puede abrir con la Grabadora de Sonidos de Windows.
  - OGG:
    - Este formato es totalmente abierto y libre de patentes.
    - Tan profesional y de calidad como cualquier otro pero con todos los valores del movimiento Open Source.

# Categorías según la extensión

## ► DE VIDEO

- Los formatos de **video** no sólo contienen imágenes sino también el **sonido** que las acompaña.
  - Es bastante habitual que al intentar visualizar un vídeo no podamos ver la **imagen** aunque sí oigamos el **sonido**.
  - **Formato de compresión** → no es reconocido por nuestro ordenador
  - Codecs actualizados de cada uno de los formatos.
- **.ASF, .LSF, .ASX**
  - Archivo de secuencias de audio o video
  - Se abre con el Reproductor Multimedia de Windows.
- **.AVI:**
  - archivo de película de video de Microsoft Windows.
  - Se abre con el Reproductor Multimedia de Windows.
- **.MPA, .M1V, .MPG, .MPE, .MPEG:**
  - archivo de video comprimido bajo norma MPEG.
  - Se ejecuta con el Reproductor Multimedia de Windows, o con reproductores
- **MOV, .QT:**
  - archivo de video en formato de Quicktime.
  - Se ejecuta con la aplicación QuicktimePlayer.
- **.RM, .RAM, .RV:**
  - archivo de video en formato propietario de Real Video.
  - Se ejecuta con RealPlayer.
- **.DivX:**
  - archivo de video en formato DivX conocido como el MP3 del video ya que permite niveles muy altos de compresión.
  - Logra que una película que ocuparía un DVD entero (de hasta 7 GB), pueda grabarse en un CD-Rom común (de 700 MB) sin perder calidad.

# Categorías según la extensión

## ► COMPRIMIDOS

- De gran utilidad para almacenar información
- ocupa el menor espacio posible
- se pueden reunir muchos ficheros en uno sólo.
- .RAR:
  - archivo compactado con la aplicación WinRAR
  - uno de los dos más populares formatos de archivo compactado en la [web](#)
- .ZIP:
  - archivo compactado con la aplicación WinZip
  - uno de los dos formatos de archivo de compactado más populares.

# Categorías según la extensión

## ▶ DE IMÁGENES

- Cada formato utiliza un método de representación y que algunos ofrecen mayor calidad que otros.
- Muchos programas de edición gráfica utilizan sus propios formatos de trabajo con imágenes.
- **.BMP:**
  - Extensión que nace del nombre de este formato BitMaP o Mapa de Bits, gran calidad pero tamaño excesivo no suele ser muy utilizado en Internet por su carga lenta.
- **JPEG:**
  - También se le ve como JPE y sobre todo como JPG es uno de los más extendidos, por su compresión y calidad, en páginas webs para logotipos y cabeceras.
- **GIF:**
  - Este formato cuenta con características que lo hacen ideal para el uso en páginas web, como es la posibilidad de darle un fondo transparente o insertarle movimiento.
- **.ICO:**
  - archivo de gráfico de ícono. Es utilizado por Windows para lanzar una aplicación con un clic sobre el mismo.
- **.PCX:**
  - archivo de gráficos creado con PC Paintbrush, de Soft.
- **.PIC:**
  - archivo de gráficos de PC Paint.
- **.PSD:**
  - archivo de mapa de bits hecho con Adobe Photoshop.
- **TGA:**
  - archivo de gráficos de alta definición hecho con tarjeta de video profesional Targa.
- **.WMF:**
  - sigla de Windows Metafile.
  - Archivos que contienen información descriptiva respecto de un archivo de gráficos.
  - El formato usado por las galerías de imágenes de Microsoft Office.

# Categorías según la extensión

## ► De Texto

- Documentos de texto plano
  - formatos que sencillamente guardan las letras (txt, log...)
  - Se abre con cualquier editor de texto (Bloc de Notas)
- Documentos de texto enriquecido.
  - los que podemos asignarles un tamaño, fuente, color , etc.. (doc)
- .TXT:
  - archivo de texto plano, sin gráficos.
- .NFO:
  - archivo de texto plano que suele incluirse como descripción del contenido de un archivo compactado .zip.
- .HTML / .HTM:
  - archivo de hipertexto, utilizado para la publicación de contenidos en la Web.
  - Es un standard utilizado por los principales navegadores de Internet.
- .DIC:
  - Diccionario para procesadores de texto.
  - Se compone de texto plano .
- .DOC:
  - archivo de texto con estilo.
  - Se genera y se abre con los principales procesadores de texto para Windows.
- .WRI:
  - archivo de texto con estilo generado por el accesorio de Windows Microsoft Write.
  - Accesible con este programa o con cualquier procesador de texto.
- .RTF:
  - Formato de archivo de texto enriquecido que permite intercambiar texto entre distintos procesadores de texto y en distintos sistemas operativos.
  - Accesible con cualquier procesador de texto



# Categorías según la extensión

- ▶ Imágenes de CD/DVD
  - Para guardar en un archivo único lo incluido dentro de un CD
  - Son exactamente iguales a lo guardado en el disco
  - Con ellas se pueden hacer múltiples copias idénticas de un disco
  - ISO, IMG y CUE

# Categorías según la extensión

## ► De Programas

- La mayoría de los programas tienen formatos de archivo propios para utilizarlos en distintas funciones.
- Al ser bastante habituales algunos de ellos, detallamos los más importantes aquí.
- POWERPOINT: .PPS, .PPT
- WORD : .DOC, .DOT (Plantilla de Microsoft Word);
- EXCEL :
  - .XLS: (Hoja de cálculo)
  - .CSV: (Archivo de valores separados por comas);
  - .XLK: (Archivo de copia de seguridad);
  - .XLM: (Macro);

# Categorías según la extensión

- ▶ Existen infinidad de extensiones de archivos
  - algunas de ellas pueden pertenecer a categorías distintas y ser utilizadas por programas totalmente diferentes.
  - La mejor opción cuando se desconoce el formato de archivo es recurrir a páginas especializadas con extensas bases de datos de archivos.
    - <http://www.filext.com>
    - <http://www.icdatamaster.com>

# Permisos de los archivos

- ▶ Los permisos sobre un archivo **definen el acceso** de un usuario o de un grupo a dicho archivo.
- ▶ Solo pueden establecerlos y cambiarlos
  - el administrador
  - el propietario
  - aquel usuario con permiso para hacerlo.
- ▶ En **Windows** solo es posible establecer **permisos** para archivos de unidades formateadas con el **sistema NTFS**.

# Operaciones comunes con archivos

## ► Crear

- Con esta operación se añade un nuevo archivo
- El nombre de un archivo
  - no podrá ser igual a otro que se encuentre en la misma ubicación
  - no podrá tener determinados caracteres: \, /, :, \*, ?, ", <, >, |.

## ► Abrir

- localizar e identificar un archivo existente para que los usuarios o el propio SO pueda operar con él
  - en algunos sistemas la operación de creación no existe como tal
    - Al abrir un archivo no existente → crea un nuevo archivo.

## ► Cerrar:

- Indicar que se va a dejar de utilizar un archivo determinado.
- Finaliza la conexión entre el programa de usuario y el archivo
  - garantizando la integridad de los registros.
- Buffer → dispositivo de almacenamiento al archivo

# Operaciones comunes con archivos

## ▶ Copiar:

- se crea una copia del archivo en otra ubicación quedando el archivo en el lugar de origen.

## ▶ Mover:

- se quita el archivo del lugar de origen y se lleva a otra ubicación de destino.

## ▶ Renombrar:

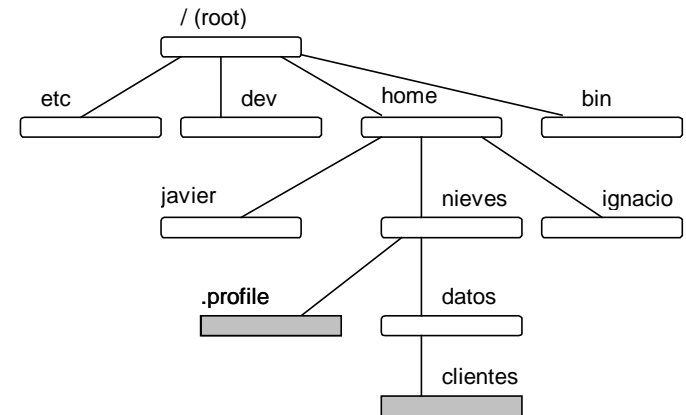
- se le cambia el nombre al archivo.

## ▶ Eliminar:

- se borra el archivo de la ubicación en la que se encuentre.

# Directorios

- ▶ Son una **división lógica de almacenamiento** de archivos u otros subdirectorios
- ▶ En Windows se llaman **carpetas**
- ▶ Constituyen una estructura jerárquica en forma de **árbol**.
- ▶ Cuando el usuario se encuentra en un directorio
  - Si no indica otra cosa
    - todos los archivos se buscan o se crean en ese directorio.



- Estructura del árbol de directorios en Linux

# Directorios

- En todo sistema de archivos hay un directorio especial llamado **raíz** (root el inglés)
- Es el directorio que contiene todos los demás directorios y archivos
- Se representa por una barra ("\" WINDOWS, "/"UNIX).
  - Mismo carácter para separar directorios
- Una **ruta** está formada por
  - una unidad y uno o varios directorios (C:\DOS)
  - En Unix no hay unidades
- Al crear un directorio automáticamente contiene otros dos subdirectorios:
  - El directorio que hace referencia al directorio en si. (.)
  - El directorio que referencia al directorio padre. (..)
- Una directorio activo
  - es la ruta en la que se está en un momento determinado
  - es el directorio actual



# Trayectoria absoluta y Trayectoria relativa

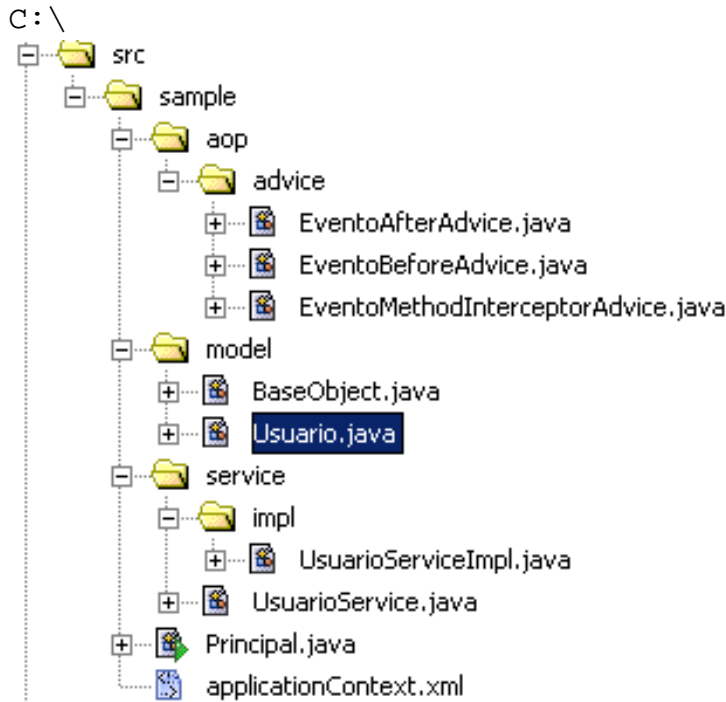
## ➤ Trayectoria absoluta

- Identifica un directorio o un fichero sin tener en cuenta el directorio activo.
- Todo fichero y directorio tienen **una sola** trayectoria absoluta.
- Se indica el **nombre de la unidad**, el **nombre del directorio** y el **nombre del archivo** o **subdirectorio** deseado. ( en UNIX no hay unidades)
  - C:\Archivos de Programas\Office\Word.exe
  - /usr/bin/calc

## ➤ Trayectoria relativa

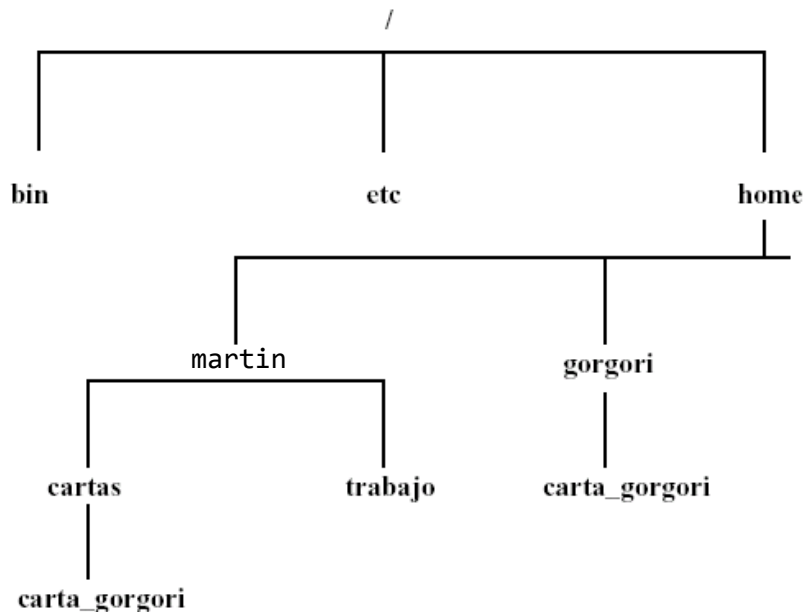
- depende de la unidad activa y del directorio activo.
- Tiene dos partes:
  - la imaginaria
    - que equivale a la unidad activa y al directorio activo
  - la real
    - Uniéndolas se obtiene la trayectoria absoluta.
- Si estoy en C:\Archivos de Programas → Office\Word.exe
- Si estoy en /usr/bin → calc

# Ejercicio Rutas Windows



- Indica la ruta relativa y absoluta de `EventoAfterAdvice.java` suponiendo que estamos en
  - `C:\`
  - `C:\src\sample`
  - `C:\src\sample\model`
  - `C:\src\sample\service\impl`

# Ejercicio Rutas UNIX



- Indica la ruta relativa y absoluta de `carta_gorgori` (en `cartas`)
- suponiendo que estamos en
  - `/`
  - `/home/gorgori`
  - `/home/martin/trabajo`
  - `/home/martin/cartas`
  - `/bin`

# Atributos de un Directorio

Califican al directorio

Varían de un sistema a otro

## DOS

H (Hidden) : oculto

R (Read only): solo lectura

A : Archivo

Fecha: fecha de creación o última modificación

Hora

Bytes	8	3	1	10	2	2	2	4
	Nombre del archivo	Extensión	Atributos	Reservado	Hora	Fecha	Nº del primer bloque	Tamaño

## UNIX

Usuario y grupo del usuario que crea el directorio

Permisos de estos y del resto de los usuarios (lectura, escritura, ejecución)

## WINDOWS

+ DOS

Comprimido o cifrado

# Permisos de los directorios

- ▶ Los permisos sobre un directorio **definen el acceso** de un usuario o de un grupo a dicho archivo.
- ▶ Solo pueden establecerlos y cambiarlos
  - el administrador
  - el propietario
  - aquel usuario con permiso para hacerlo.
  -
- ▶ En **Windows** solo es posible establecer **permisos** para archivos de unidades formateadas con el **sistema NTFS**.

# Operaciones comunes con archivos

## ▶ **Crear**

- Con esta operación se añade un nuevo directorio
- El nombre de un directorio
  - no podrá ser igual a otro que se encuentre en la misma ubicación
  - no podrá tener determinados caracteres: \, /, :, \*, ?, ", <, >, |.

## ▶ **Copiar:**

- se crea una copia del directorio en otra ubicación quedando el directorio en el lugar de origen.

## ▶ **Mover:**

- se quita el directorio del lugar de origen y se lleva a otra ubicación de destino.

## ▶ **Renombrar:**

- se le cambia el nombre al directorio.

## ▶ **Eliminar:**

- se borra el directorio de la ubicación en la que se encuentre y su contenido

# Operaciones comunes con archivos

## ▶ **Copiar:**

- se crea una copia del archivo en otra ubicación quedando el archivo en el lugar de origen.

## ▶ **Mover:**

- se quita el archivo del lugar de origen y se lleva a otra ubicación de destino.

## ▶ **Renombrar:**

- se le cambia el nombre al archivo.

## ▶ **Eliminar:**

- se borra el archivo de la ubicación en la que se encuentre.

# Implementación del SA

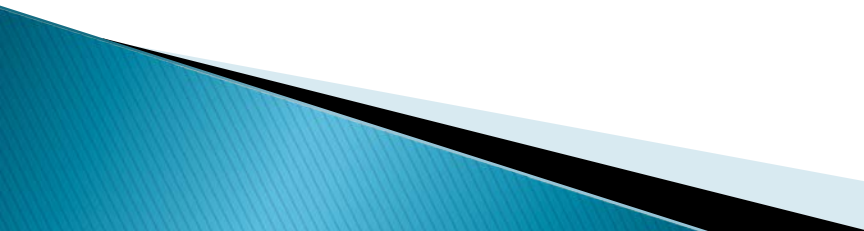
- Registro de los bloques asociados a cada archivo
  - Aspecto clave de la implementación del almacenamiento de archivos.
  - Cada SA implementa métodos distintos para solucionar este problema.
- Bloque
  - Sectores que se asocian a un único archivo.
- Un archivo
  - Se almacena en uno o más bloque de sectores.



# Tamaño del bloque

- Si el tamaño del bloque es **muy grande**
  - el archivo de un tamaño muy pequeño
    - se le asignará el bloque entero
    - se desperdicia gran parte de la capacidad del disco.
- Si el tamaño del bloque es **demasiado pequeño** para almacenar un archivo
  - harán falta muchos bloques
  - se producirá un retraso en la lectura del archivo
  - tiene que localizar en el disco todos los bloques
- Tiene que ser:
  - lo suficientemente pequeño para **no desperdiciar espacio**
  - lo suficientemente grande como **para no ralentizar la lectura**
- Son adecuados de bloque de 512 bytes, 1 KB o 2 KB.
- Tamaño de bloque de 2 KB, en un disco cuyo sector tiene 512 bytes
  - cada bloque estará compuesto por cuatro sectores.

# Bloques asociados a cada archivo

- Para manejar los bloques asociados a cada archivo se pueden utilizar varias técnicas:
    - Asignación adyacente
    - Lista enlazada
    - Lista enlazada y un índice
    - Tabla de i-nodos
- 

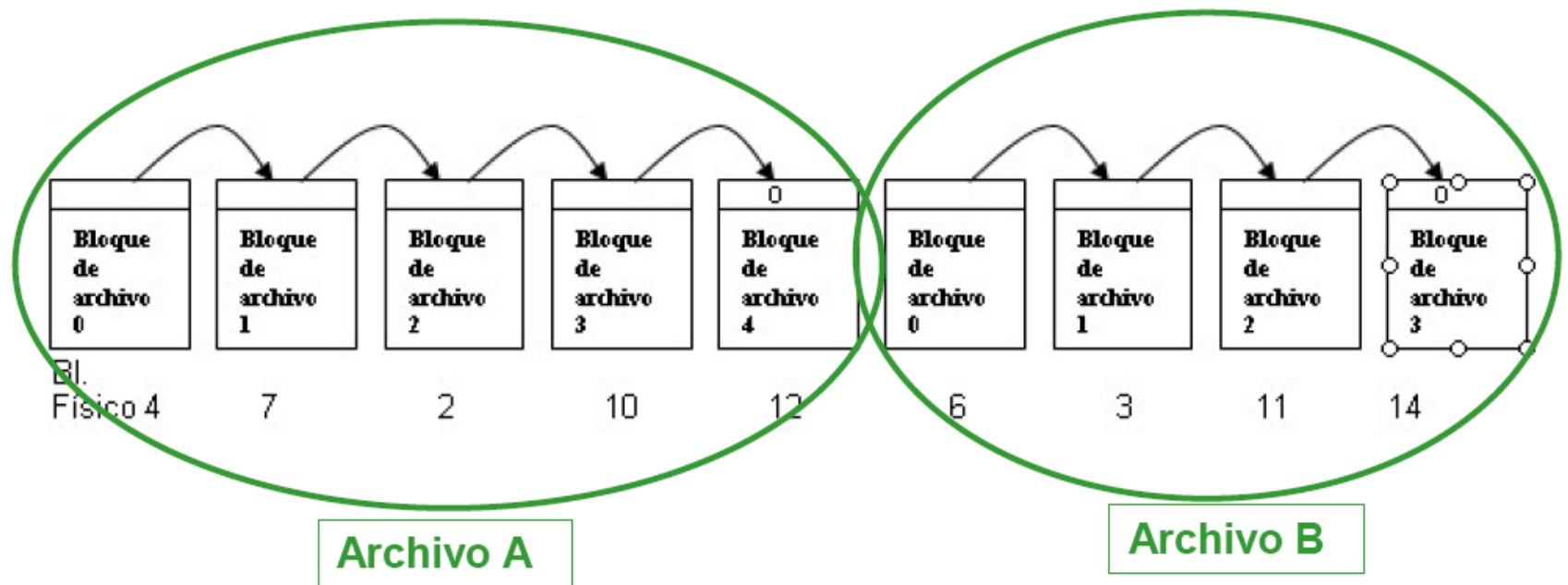
# Asignación Adyacente

- Almacena los archivos mediante bloques adyacentes en el disco
- En el directorio únicamente se tendrá que guardar la dirección en la que comienza el primer bloque,
  - los demás están a continuación
- Ventaja
  - fácil implementación
- Inconveniente
  - es necesario conocer con anterioridad el número de bloques que ocupará el fichero
  - Genera una gran fragmentación del disco que produce una pérdida de espacio.

# Lista Enlazada

- El directorio contiene la dirección del primer bloque
- Cada bloque tiene
  - la dirección del siguiente bloque
  - el valor *null* (nulo) → el último bloque del fichero.
- Se **aprovechan todos** y cada uno de los bloques del disco
  - Se evita perder capacidad por fragmentación
- Inconvenientes
  - Cada bloque **pierde parte de su capacidad**
    - Se ha de reservar un espacio para contener la dirección del siguiente bloque
    - El **retardo que se produce en la lectura del archivo** al tener que leer todos los bloques antes de llegar a uno determinado.

# Lista Enlazada



# Lista Enlazada y un Índice

- Intenta eliminar los defectos de la anterior.
- Se crea una tabla con un registro por cada uno de los bloques del disco
- En cada registro se indica:
  - si dicho bloque está libre (null)
  - cuál es la dirección del siguiente bloque
    - en el caso de que ese bloque pertenezca a un determinado archivo
- En el directorio se asocia con el nombre archivo el número de bloque en el que comienza dicho archivo
  - Con este dato y mediante la tabla se puede averiguar la dirección de todos los bloques que componen dicho archivo únicamente siguiendo la lista ligada.

# Lista Enlazada y un Índice

0		
1		
2	10	
3	11	
4	7	El Archivo A comienza aqui
5		
6	3	El Archivo B comienza aqui
7	2	
8		
9		
10	12	
11	14	
12	0	
13		
14	0	
15		

# Lista Enlazada y un Índice

- Ventajas
  - Todo el bloque está disponible para los datos.
  - El acceso aun es mucho más rápido
    - Al estar la tabla en memoria las consultas son mucho más rápidas y no es necesario acceder al disco
      - también haya que seguir la cadena de bloques como en la lista enlazada
- Inconveniente
  - Toda la tabla de registros debe estar en la memoria principal permanentemente
    - La memoria consumida para almacenar la tabla no está disponible para ser usada por otros procesos.
    - Discos con un gran número de bloques
      - La tabla de registros puede llegar a ocupar gran parte de la memoria principal del ordenador
- Es la técnica utilizada por DOS y por Windows
  - La tabla de registros se le denomina **FAT** (*File Allocation Table*)
  - Se puede encontrar en sus dos versiones
    - FAT16 → los bloques se direccionan con 16 bits
    - FAT32 → los bloques se direccionan con 32 bits



# Lista Enlazada y un Índice

- ▶ En NTFS no hay áreas de disco reservadas para datos como la *FAT*.
  - **Todo en una partición NTFS es un archivo**, incluso la *Master File Table* (MFT)
    - es una base de datos de los archivos y las carpetas de la partición
      - incluye su nombre, ubicación, tamaño y permisos, así como sus atributos y otra información.

# i-nodos

- Se asocia a cada archivo una pequeña tabla → **inodo**
- Contiene los **atributos** y **direcciones** en disco de los bloques del archivo.
- Las últimas entradas del **inodo** se reservan para cuando el archivo ocupa **más bloques** de los que el es capaz de almacenar
  - Pueden contener la dirección de otro bloque en el que se guardan las demás direcciones de los bloques del archivo.
  - Este bloque se le llama **bloque indirecto**.
- Si se necesitan se puede utilizar un **bloque doblemente indirecto** o incluso **triplemente indirecto**
- UNIX y Linux utilizan este sistema
  - Cuando abre un archivo
    - lo primero que hace cargar en memoria su inodo correspondiente
      - El proceso es lo más rápido posible

# i-nodos

Tipo de fichero y permisos
Numero de enlaces del fichero
Información del propietario y grupo
Tamaño del fichero en bytes
Fechas (acceso, modificación, etc)
Bloque directo 1
Bloque directo 2
.....
.....
Bloque directo 12
Bloque de indirección simple 13 (2048)
Bloque de indirección doble 14
Bloque de indirección tripre 15

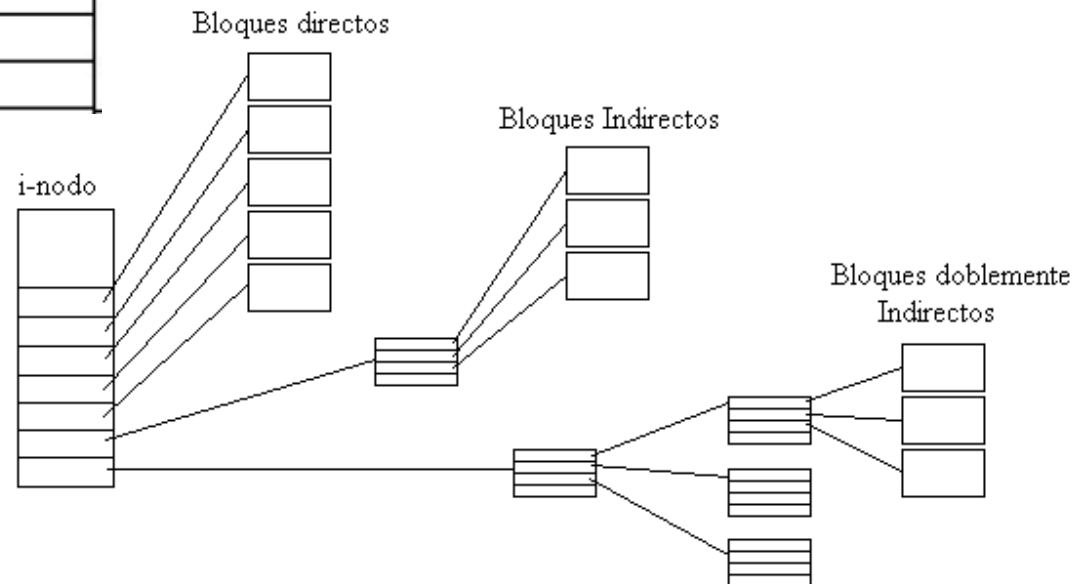
Para cada fichero se crea una entrada, cada entrada tiene 13 índices:

1 al 10 → datos

11 → un nivel de índices (**indirecto**)

12 → dos niveles de índices (**doblemente ind**)

13 → tres niveles de índices (**triplemente ind**)



# Ejercicio

- Si un archivo se encuentra grabado físicamente en disco en los bloques 5, 15, 6, 3, 11, 7 y 10 ¿Cómo se implementaría utilizando *lista enlazada y un índice*? Nuestro disco tiene 32 bloques

# Tipos de sistemas de archivos

- FAT16
  - FAT32
  - NTFS5
  - Sistema de Archivos Extendido 3 (ext3fs)
  - Sistema de Archivos Extendido 4 (ext4fs)
  - HPFS
- 

# FAT16

- ▶ También denominado FAT
- ▶ Se puede acceder a este sistema de archivos desde
  - *MS-DOS*
  - todas las versiones de Windows.
- ▶ Tamaño máximo de la partición: 2GB
- ▶ Tamaño máximo de un archivo: 2GB
- ▶ No es Case-sensitive
- ▶ No soporta dominios

# FAT32

- ▶ Se puede acceder a este sistema de archivos desde
  - *Windows 95 OSR2*
  - *Windows 98*
  - *Windows 2000*
  - *Windows XP*
  - *Windows Server 2003*
  - *Windows Vista*
  - *Windows Server 2008*
  - *Windows 7*
- ▶ Permite trabajar con particiones mayores de 2GB
- ▶ El tamaño máximo de un archivo es de 4GB
- ▶ Los volúmenes pueden llegar hasta 2 TB
  - en *Windows 2000* sólo hasta 32 GB
- ▶ No soporta dominios.

# NTFS 5

- ▶ **NT File System 5**
- ▶ Desarrollado para Windows NT 4
- ▶ Permite
  - Nombres de archivo de hasta 256 caracteres
  - Ordenación de directorios
  - Atributos de acceso a archivos
  - Reparto de unidades en varios discos duros
  - Reflexión de discos duros
  - Registro de actividades.
- ▶ En Windows 200X Server se incluyeron mejoras
  - Directorio Activo
  - Mejora de dominios
  - Cuota en disco para cada usuario
  - Cifrado y compresión de archivos
  - Almacenamiento remoto
  - Hrr. Desfragmentación
  - Enlaces a archivos ~ UNIX
- ▶ Sus volúmenes pueden llegar hasta
  - 16 TB menos 64 KB
  - Tamaño máximo de un archivo sólo está limitado por el tamaño del volumen
  - Case-sensitive



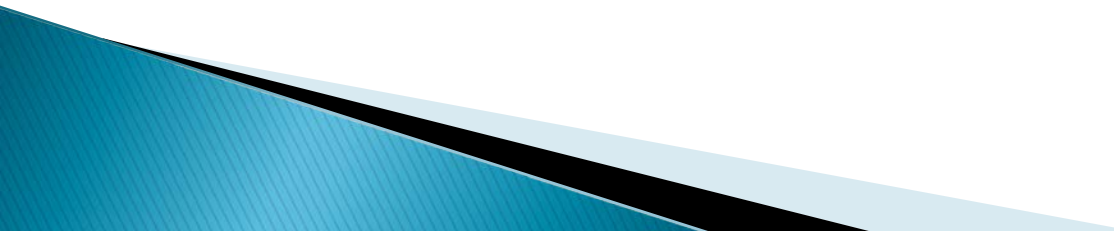
# Sistema de Archivos Extendido 3

- ▶ **Ext3fs**
- ▶ es uno de los más eficientes y flexibles sistemas de archivos.
- ▶ Se puede acceder desde *Linux*
- ▶ Permite hasta 256 caracteres en los nombres de los archivos
- ▶ Tamaño máximo de un volumen es de 32 TB
- ▶ Tamaño máximo de un archivo es de 2 TB
- ▶ Case-sensitive
- ▶ **Registro de diario**
  - permite almacenar la información necesaria para restablecer los datos afectados por una transacción en caso de que ésta falle.

# Sistema de Archivos Extendido 4

- ▶ *Ext4fs*
- ▶ Es uno de los más eficientes y flexibles sistemas de archivos.
- ▶ Se puede acceder desde *Linux*
- ▶ Es compatible con *ext3*
- ▶ Permite hasta 256 caracteres en los nombres de archivos
- ▶ Tamaño máximo de un volumen es de **1 EB**.
- ▶ Tamaño máximo de un archivo es de **16 TB**.
- ▶ Case-sensitive
- ▶ **Registro de diario**
  - permite almacenar la información necesaria para restablecer los datos afectados por una transacción en caso de que ésta falle.

# HPFS

- ▶ *High Performance File System*
  - ▶ Se creó para *OS /2*
  - ▶ Resolvía los problemas del sistema de archivos *FAT*.
  - ▶ Permite hasta 256 caracteres en los nombres de archivos
  - ▶ Tamaño máximo de un volumen es de 2 TB.
  - ▶ Tamaño máximo de un archivo es de 2 GB.
- 

# Actividad

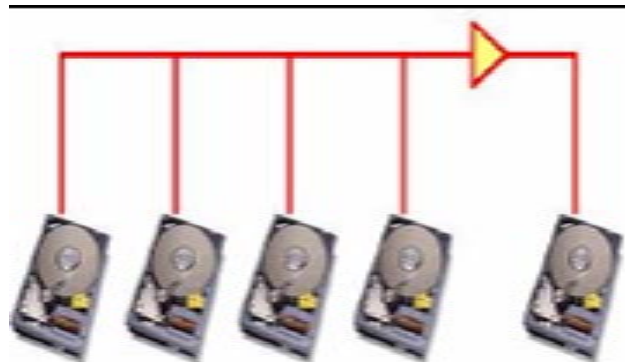
- »» Averigua cuál es el sistema de archivos que se utiliza en tu equipo

# RAID

- ▶ Redundant Array of Independent Disk
  - Conjunto redundante de discos independientes.
- ▶ Dos o más discos funcionando de forma conjunta
- ▶ Se distribuyen los datos entre los discos
- ▶ Aumenta el **rendimiento** (*striping*)
- ▶ Mejora la **fiabilidad** (bits de paridad)
- ▶ Implementación por *hw* o por *sw*
- ▶ Diferentes niveles (0-6) con diferentes características

# RAID

- ▶ RAID es un método
  - de combinación de varios discos duros
  - para formar una única unidad lógica en la que se almacenan los datos de forma redundante.
  - Ofrece mayor **tolerancia a fallos** y más altos niveles de **rendimiento** que un sólo disco duro o un grupo de discos duros independientes.
- ▶ Un RAID, para el sistema operativo, **aparenta ser un sólo disco duro lógico**



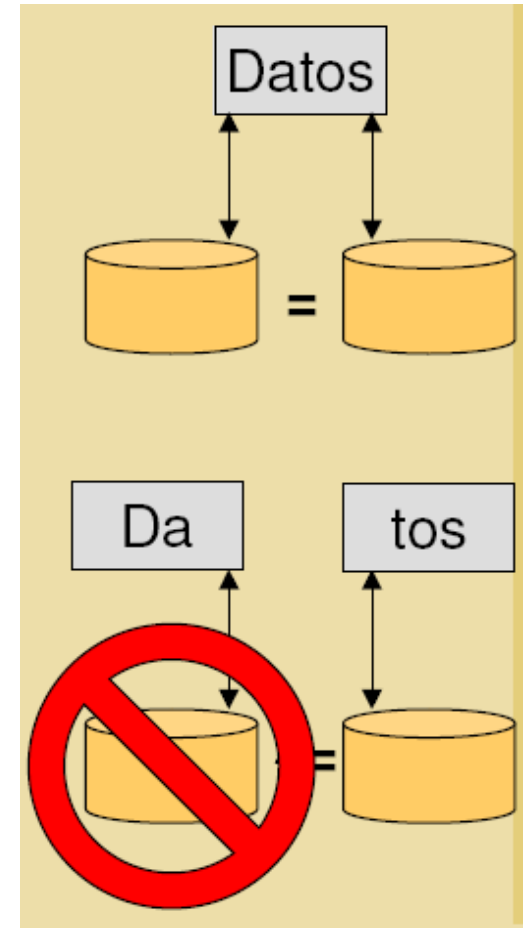
# Conceptos

## ► Redundancia

- Los datos se guardan en más de un disco
- Si un disco falla, se puede acceder a los datos de otro/s discos.
- Escritura más lenta.
- Ejemplo: discos espejos (mirroring)

## ► Fraccionamiento de datos (striping)

- Los datos se reparten entre varios discos
- Acceso paralelo a discos
- Incremento rendimiento (transferencia de datos) y búsqueda
- Si se estropea un disco, se pierden datos.



# Conceptos

- ▶ Bit de paridad: Redundancia dentro del fraccionamiento

Consideremos guardar el dato: 01110

Supongamos que cada bit lo guardamos en un disco diferente

Paridad de la cadena (01110) es igual a:

- 1 si existe un número impar de 'unos' en la cadena de bits
- 0 si el número de 'unos' es par

Se guarda la paridad de '01110' en un sexto disco

- ¿Cuál es la paridad de '01110'? -> 1

Si un disco falla, se puede recomponer la cadena

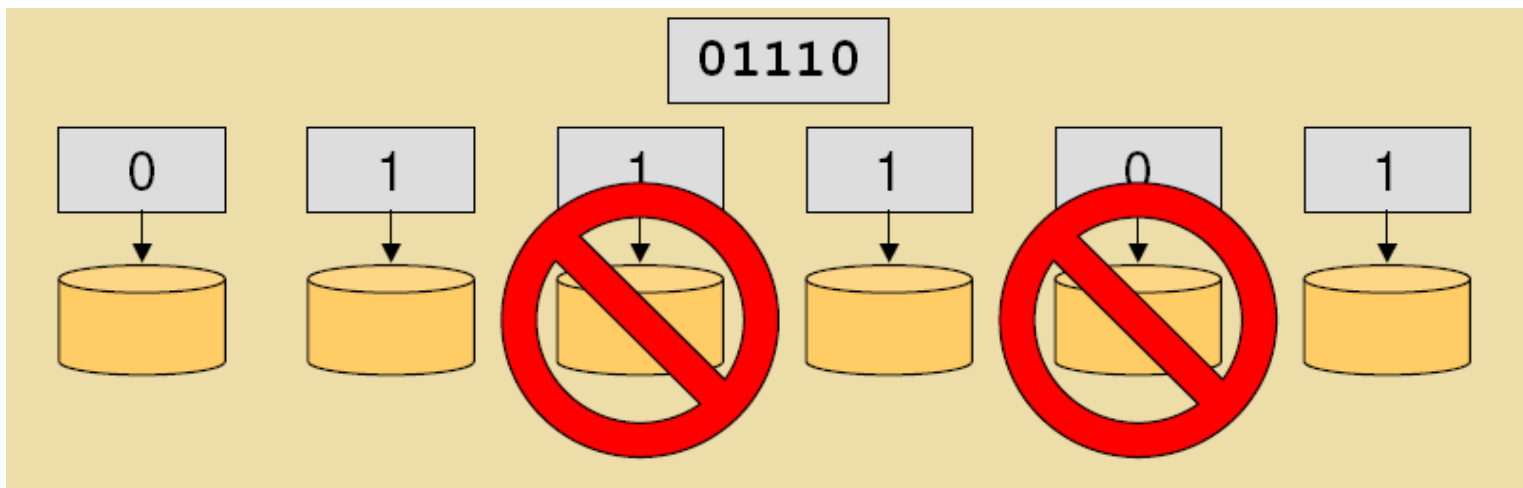
- '0X110' con paridad '1', implica que la 'X' es un '1'
- '0111X' con paridad '1', implica que la 'X' es un '0'

1	0	1	1	0		1
0	0	1	1	0		0
1	1	1	1	0		0



# Conceptos

- ▶ Bit de paridad: Redundancia dentro del fraccionamiento



¿ Qué pasa si el tercer disco falla ?

- Se reconstruye el dato a partir del bit de paridad.

¿ Qué ocurre si falla el quinto disco (con 3º bien)?

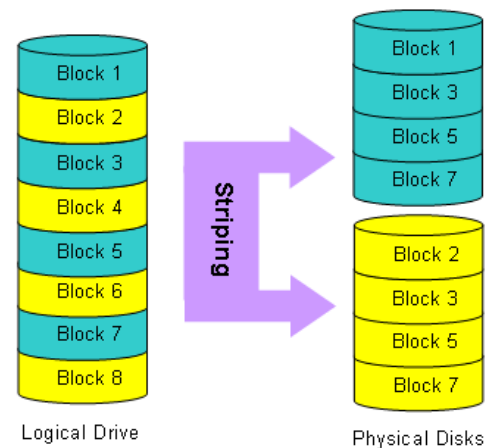
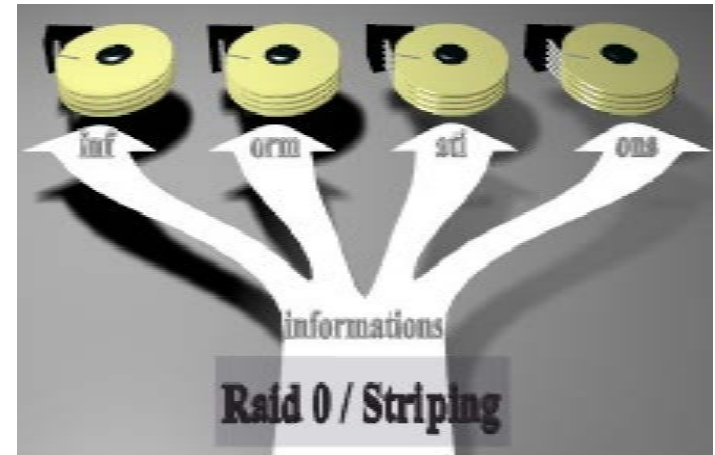
- Se reconstruye el dato a partir del bit de paridad.

¿ Y si fallan el tercero y el quinto ?

- No es posible determinar el valor de 2 bits desde la paridad.

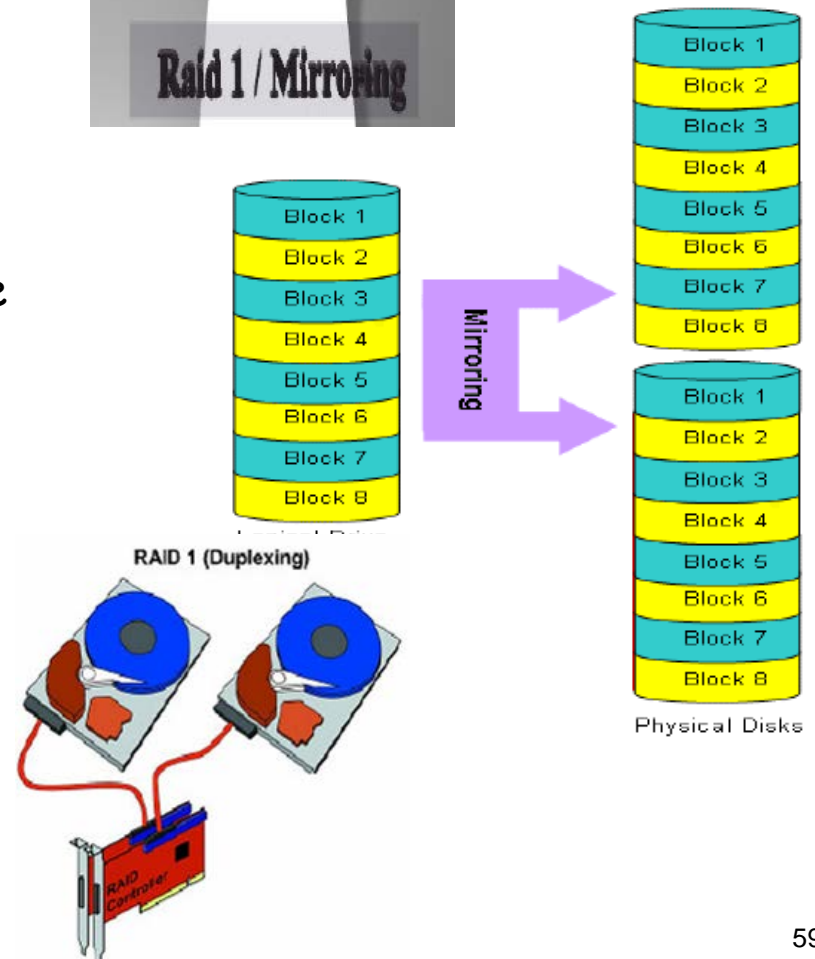
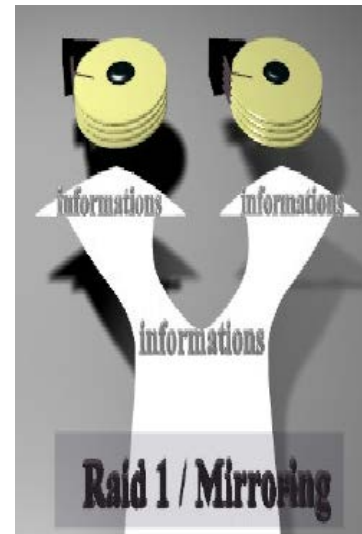
# Nivel RAID 0

- ▶ Fraccionamiento en discos
  - Se escriben los datos fraccionados en diferentes discos.
  - Requiere al menos dos discos duros.
  - No proporciona redundancia de datos.
  - Si algún disco falla, se pierde la información.
  - Mejora prestaciones en velocidades de rendimiento de datos.
    - Posibilidad de acceso en paralelo
  - El S.O. 've' un disco único (disco lógico).
  - No es verdaderamente un RAID ya que no presenta integridad de datos
    - Falla uno → pérdida total de los datos



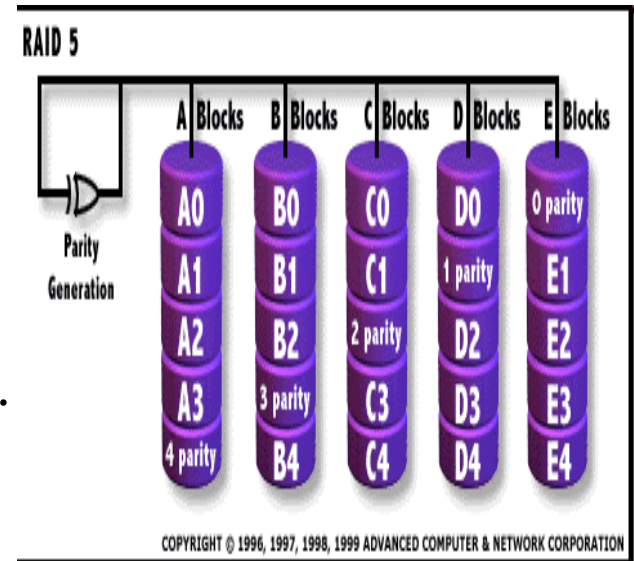
# Nivel RAID 1

- ▶ Discos 'espejos' - (mirroring)
  - Se escriben los datos en más de un disco
  - Se usan normalmente dos discos duros
  - Se produce una imagen 'espejo' idéntica al disco principal
  - Hay una penalización en velocidad de escritura y lectura
  - Si algún disco falla, NO se pierde la información
  - El S.O. 've' un disco único (disco lógico)
  - Si usamos dos controladoras se denomina 'Mirror-Duplexing'
  - Es caro:
    - Emplea el doble del espacio necesario



# Nivel RAID 5

- ▶ Fraccionamiento en discos y Fraccionamiento en discos y paridad distribuida
  - Se distribuyen los datos y la paridad de forma ordenada en los diferentes discos. No hay discos 'dedicados' a una labor.
  - Requiere al menos tres discos.
  - Aprovecha 2/3 de la capacidad total para datos.
  - La implementación RAID más extendida.
  - No se desaprovecha ningún disco exclusivamente para almacenar códigos de paridad
  - Se pueden recuperar los datos



# Nivel RAID 6

- ▶ **Acceso independiente con doble paridad**
- ▶ Similar al RAID 5, pero incluye un segundo esquema de paridad distribuido por los distintos discos.
- ▶ Tolerancia extremadamente alta a los fallos y a las caídas de disco, ofreciendo dos niveles de redundancia.
- ▶ Coste de implementación mayor.
  - Controladoras más complejas y más caras.
  - Es ineficiente cuando se usa un pequeño número de discos pero a medida que el conjunto crece la pérdida en capacidad de almacenamiento se hace menos importante, creciendo al mismo tiempo la probabilidad de que dos discos fallen simultáneamente.
- ▶ Proporciona protección contra fallos dobles de discos y contra fallos cuando se está reconstruyendo un disco.
- ▶ La capacidad de datos de un conjunto RAID 6 es  $n-2$ .
- ▶ No penaliza el rendimiento de las operaciones de lectura, pero sí el de las de escritura debido al proceso que exigen los cálculos adicionales de paridad.

