

Administración disco

aspectos fundamentales

Sistema archivos vs disco

- Sistema archivos es la abstracción usada por el kernel para representar y organizar los recursos de almacenamiento
- Recursos almacenamiento pueden ser:
 - El disco duro
 - El disquette
 - El CD-ROM
 - La cinta magnética
 - Memoria Flash
 - Memoria de estado sólido (SSD: Solid State Drive, p.e. MAC Air)

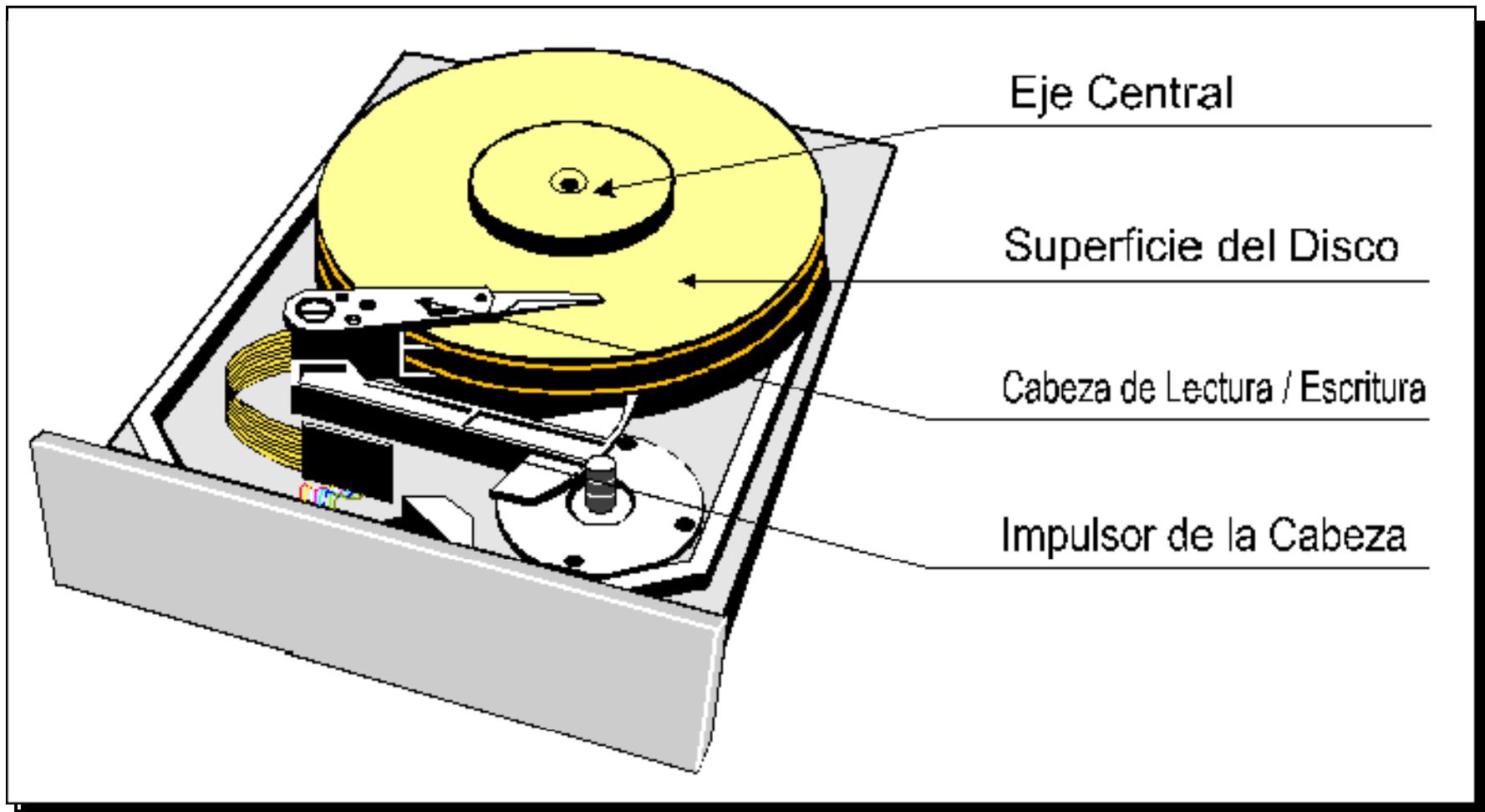
Tareas Admon. Archivos

- Asegurar disponibilidad
- Vigilar, supervisar y administrar recursos de almacenamiento
- Protección información
- Asegurar confiabilidad
- Detectar y reparar alteraciones en el sistema de archivos
- Instalar y configurar los nuevos dispositivos periféricos

El disco

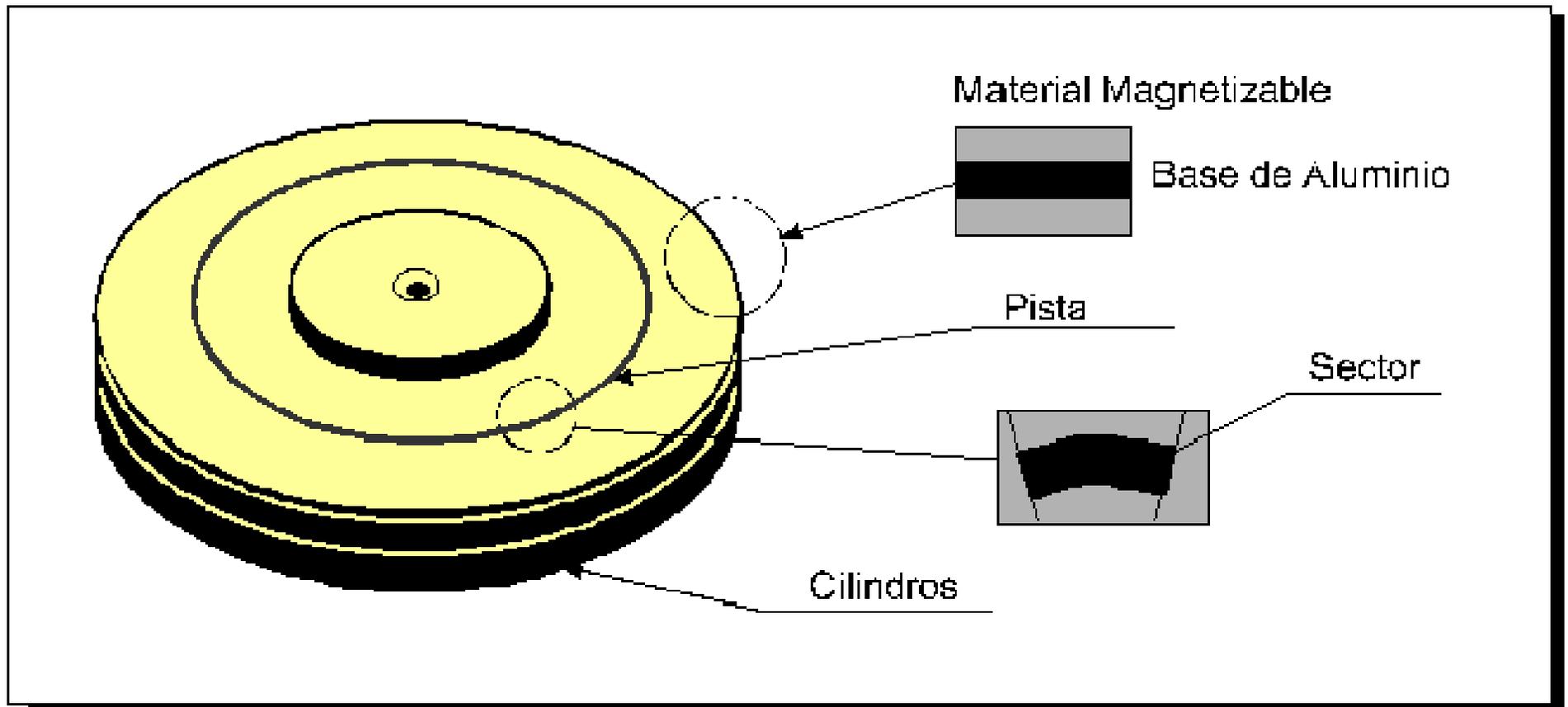
- Los discos constituyen el grueso del almacenamiento secundario
- Unidad de discos, compuesta de varios platos cada uno con dos caras
- Cada cara esta compuesta de pistas concéntricas de superficie diferente
 - densidad información varía entre centro y orilla
- Pistas divididas en sectores compuestos de bytes
- Sector es la unidad más chica

Esquema general del disco

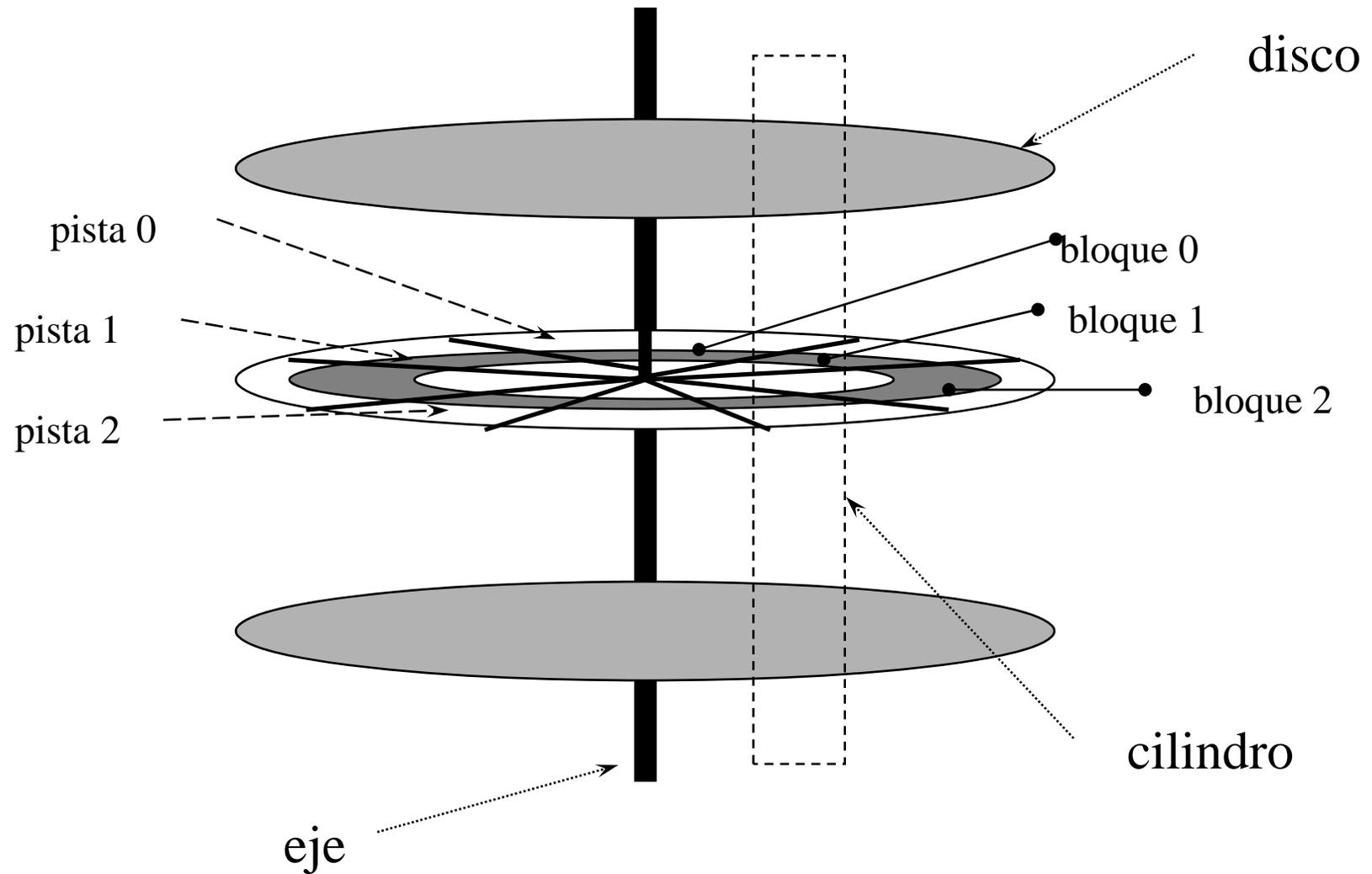


- Es posible manejar sectores pero no fragmentos de sectores
- Un conjunto de sectores se conoce como eje
- Las pistas son agrupadas en cilindros
- A cada cara le corresponde solo una de las cabezas de lectura
- Diámetro de un disco es el de sus caras y es medido en pulgadas
- Capacidad discos de 5 1/4pgs es del orden de los 4 Go y de 2 Go para los de 3 1/2 pgs

Pista, sector y cilindros



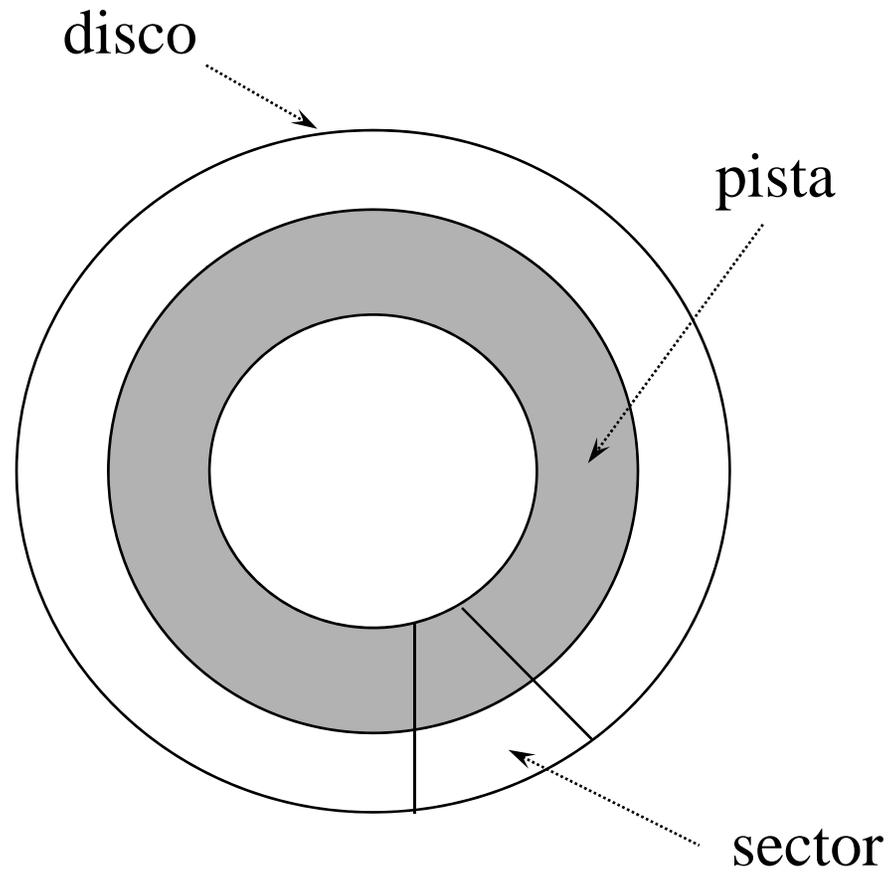
Las partes de un disco



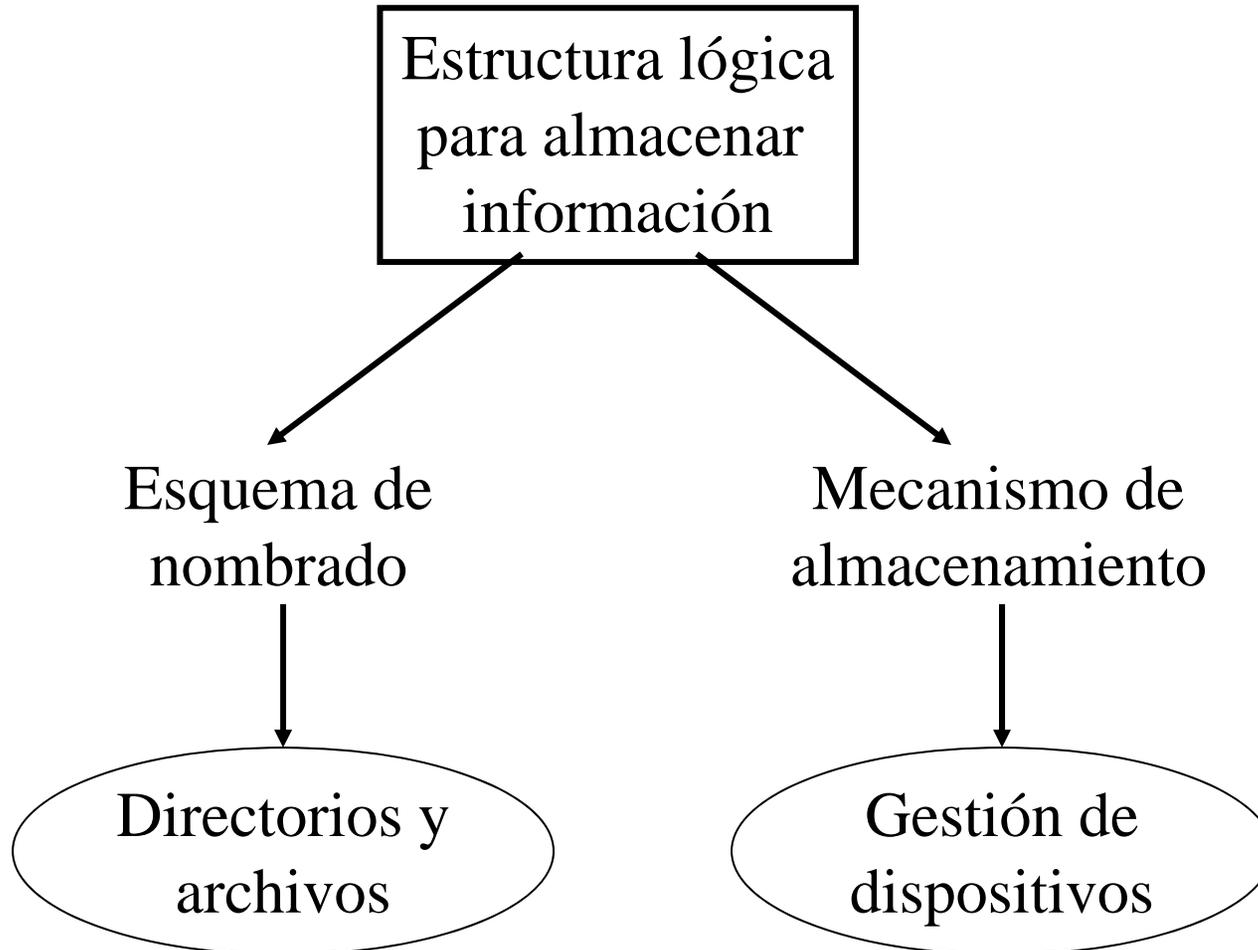
Geometría de un disco

- Descripción del conjunto de componentes de un disco y sus características:
 - número sectores por pista
 - número de pistas por cilindro
 - número de cilindro por disco
 - la velocidad de rotación (en general 3600 vueltas/minuto, pero puede llegar hasta 5400)

Ejemplo disco, pista sector

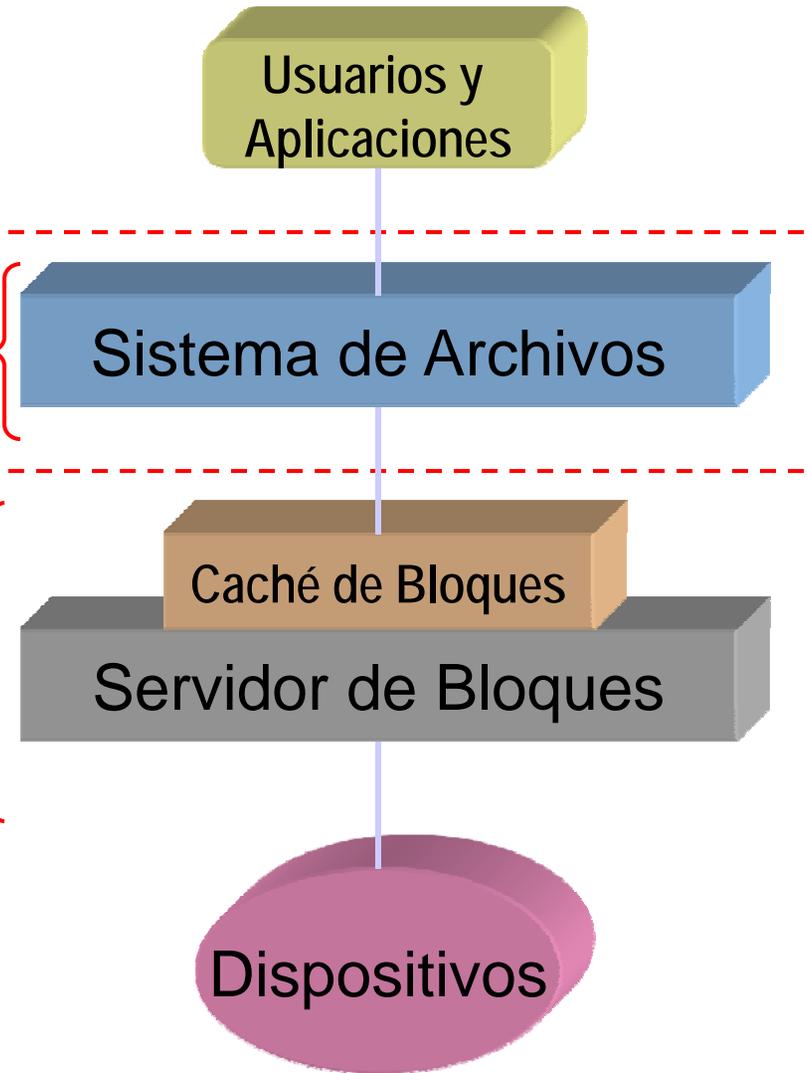


Estructura del sistema de archivos



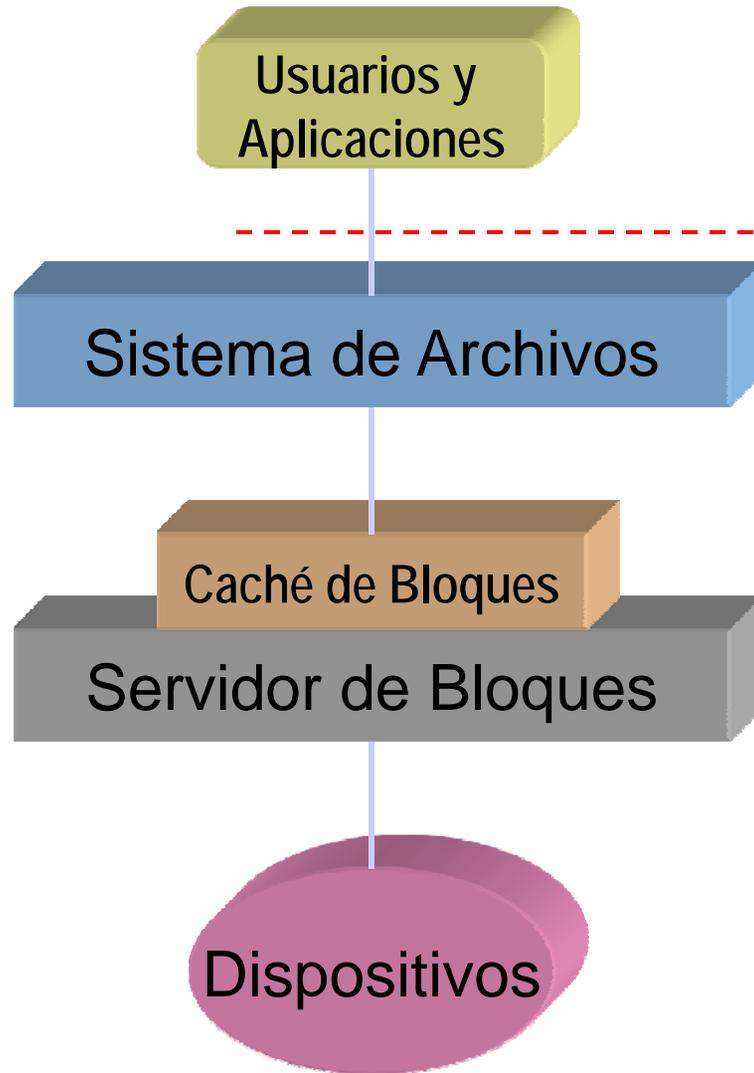
El sistema de archivos y el resto del sistema operativo

- **Visión del usuario:**
 - Archivos
 - Directorios
 - Utilidades
 - Protección
- **Sistema de Archivos:**
 - Estructura lógica
 - Eficiencia
 - Consistencia
- **Servidor de Bloques:**
 - Gestión de dispositivos
 - *Caché y buffering*



Visión externa del sistema de archivos

1



**Servicios lógicos
(archivos y directorios)**

El servidor de archivos

- Parte del sistema operativo.
- Proporciona un acceso eficiente y sencillo a los dispositivos de almacenamiento
- Permite almacenar, buscar y leer datos fácilmente.
- Aborda dos tipos de problemas
 - Definir la visión de usuario del sistema de entrada/salida, incluyendo servicios, archivos, directorios, sistemas de archivos, etc.
 - Definir los algoritmos y estructuras de datos a utilizar para hacer corresponder la visión del usuario con el sistema físico de almacenamiento secundario.

Visión lógica de los archivos

- Conjunto de información relacionada. Definida por su creador.
- La estructura lógica (para las aplicaciones) de un archivo puede ser muy diferente: árboles, registros, índice, etc
 - Archivos ejecutables o de bibliotecas dinámicas.
 - Archivos en otros formatos
- Estructura del archivo para el sistema operativo:
 - Secuencia o tira de bytes (UNIX, POSIX).
- La información relacionada con el archivo se mantiene en el descriptor del archivo, al que se apunta desde los directorios.
- Es distinto en cada sistema operativo: i-nodo, registro Windows, etc.

Atributos de un archivo

- **Nombre:** la única información en formato legible por una persona.
- **Identificación única del archivo y del usuario:** descriptor interno del archivo, dueño y grupo del archivo
- **Tipo de archivo:** necesario en sistemas que proporciona distintos formatos de archivos.
- **Tamaño del archivo:** número de bytes en el archivo, máximo tamaño posible, etc.
- **Protección:** control de accesos y de las operaciones sobre archivos
- **Información estampillas de tiempo:** de creación, de acceso, de modificación, etc.
- **Información de control:** archivo oculto, de sistema, normal o directorio, etc.

El directorio

- Objeto que relaciona de forma unívoca un nombre de usuario de archivo con su descriptor interno
- Organizan y proporcionan información sobre la estructuración de los sistemas de archivos
- Una colección de nodos que contienen información acerca de los archivos

Visión lógica del directorio

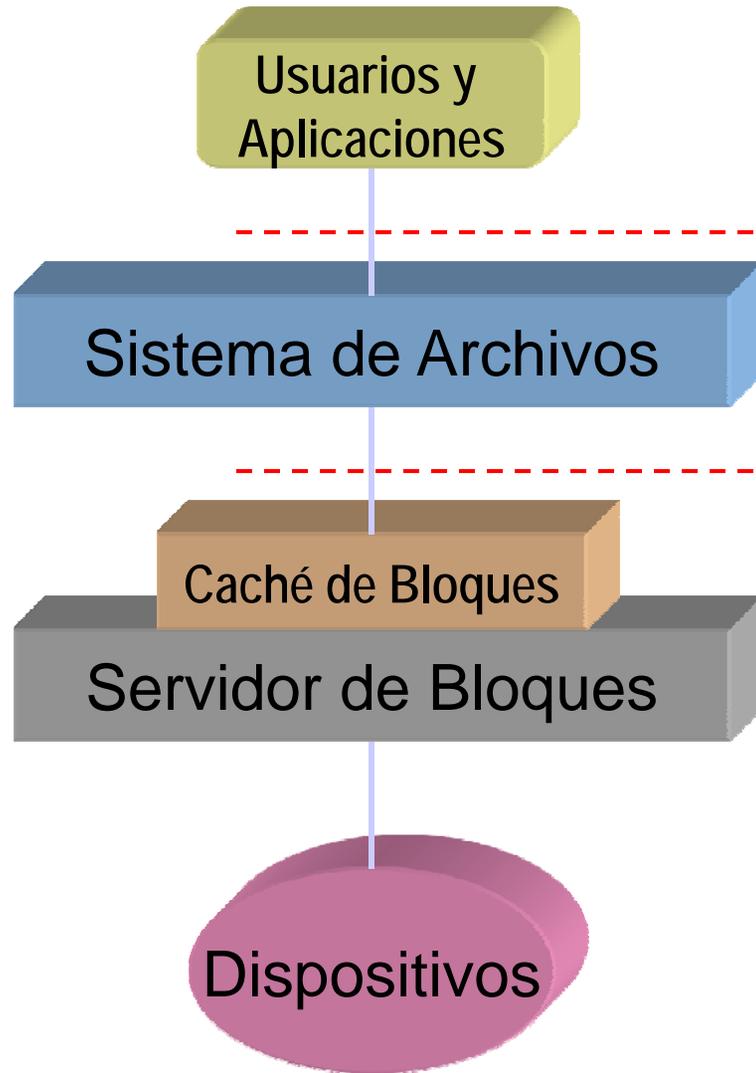
- Esquema jerárquico.
- Cuando se abre un archivo el SO busca el nombre en la estructura de directorios.
- Operaciones sobre un directorio:
 - Crear (insertar) y borrar (eliminar) directorios.
 - Abrir y cerrar directorios.
 - Renombrar directorios.
 - Leer entradas de un directorio.
 - Montar (combinar)
- La organización jerárquica de un directorio
 - Simplifica el nombrado de archivos (nombres únicos)
 - Proporciona una gestión de la distribución => agrupar archivos de forma lógica (mismo usuario, misma aplicación)

Estructura del directorio

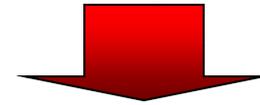
- Eficiencia: localizar un archivo rápidamente
- Nombrado: conveniente y sencillo para los usuarios
 - Dos usuarios pueden tener el mismo nombre para archivos distintos
 - Los mismos archivos pueden tener nombres distintos
 - Nombres de longitud variable
- Agrupación: agrupación lógica de los archivos según sus propiedades (por ejemplo: programas Pascal, juegos, etc.)
- Estructurado: operaciones claramente definidas y ocultación
- Sencillez: la entrada de directorio debe ser lo más sencilla posible.

Visión lógica y organización física

2



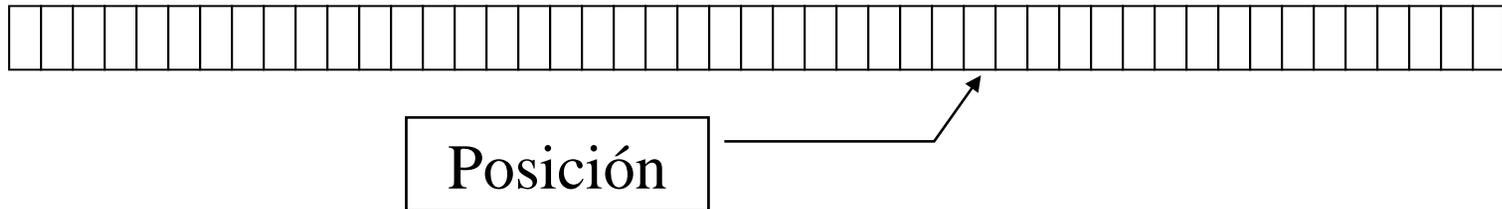
**Servicios lógicos
(archivos y directorios)**



Bloques de disco

Estructura lógica vs estructura física

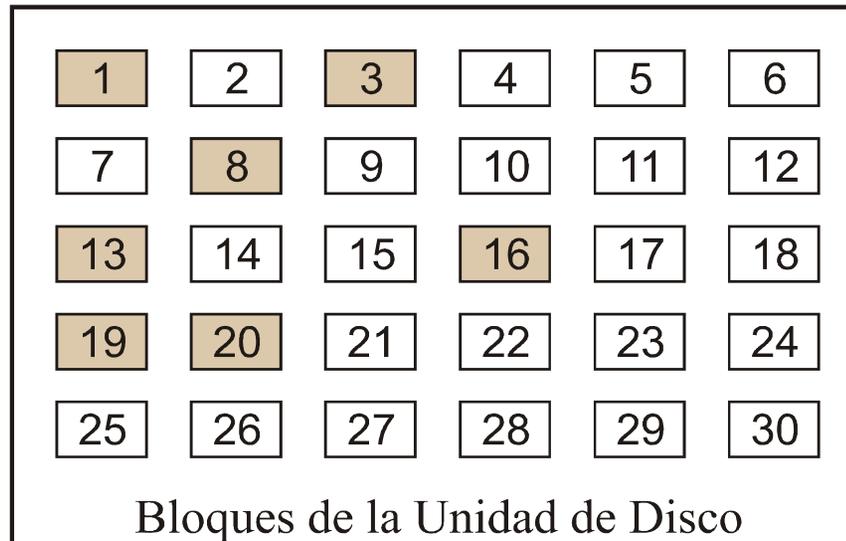
- Usuario: Visión lógica.



- Sistema operativo: visión física ligada a dispositivos.
Conjunto de bloques.

Archivo A

Bloques: 13
20
1
8
3
16
19



Visión física archivos

- Bloque
 - Unidad de transferencia
 - 2ⁿ sectores
 - Parámetro fijo por sistema de archivos
- Agrupación
 - Unidad de asignación
 - 2^p bloques
 - Aumenta la secuencialidad del archivo
- Descripción de la estructura física:
 - Bloques utilizados

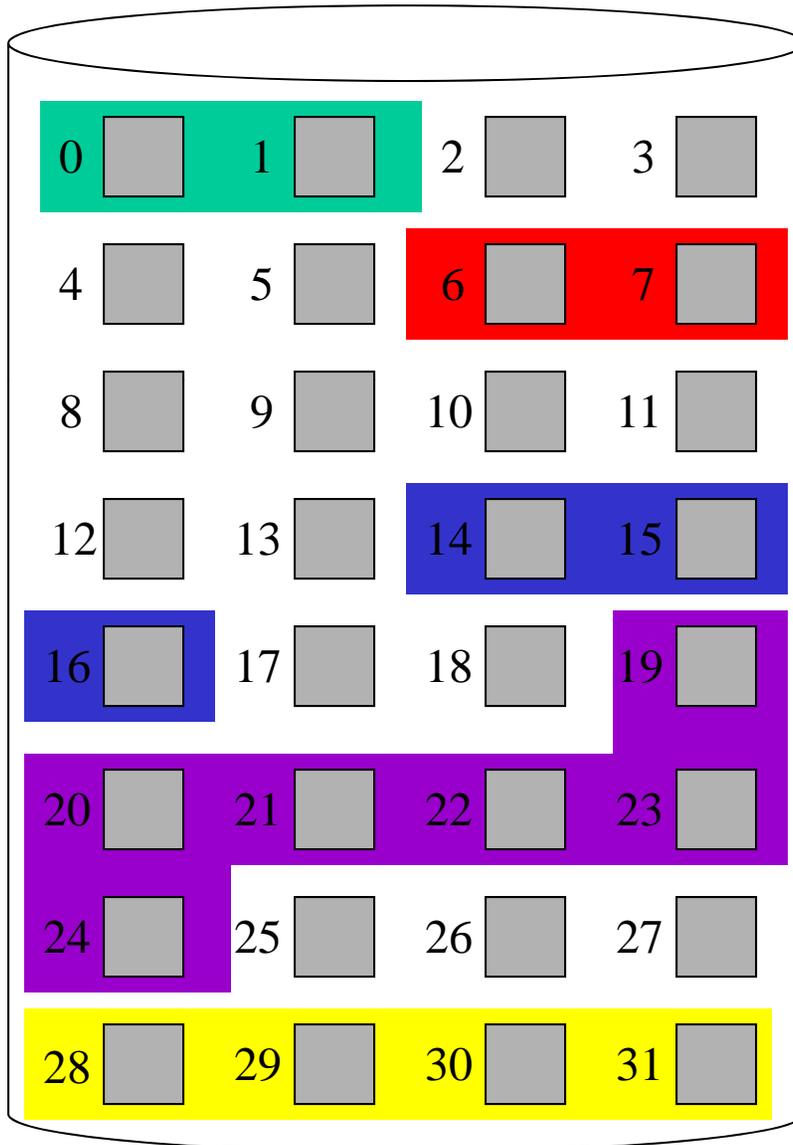
Métodos asignación

- Asignación continua
- Asignación enlazada
 - tabla de asignación de archivos (FAT)
- Asignación indizada

Asignación contigua

- Requiere que cada archivo ocupe un conjunto de bloques contiguos en el disco.
- Las direcciones de disco definen un ordenamiento lineal en él.
- Con este ordenamiento (suponiendo que sólo un trabajo está accediendo al disco) el acceso al bloque $b+1$ después del bloque b normalmente no requiere movimiento de cabeza
- Sistema operativo VM/CMS de IBM lo utiliza

Ejemplo asignación continua



archivo	inicio	longitud
cuenta	0	2
tr	14	3
correo	19	6
lista	28	4
f	6	2

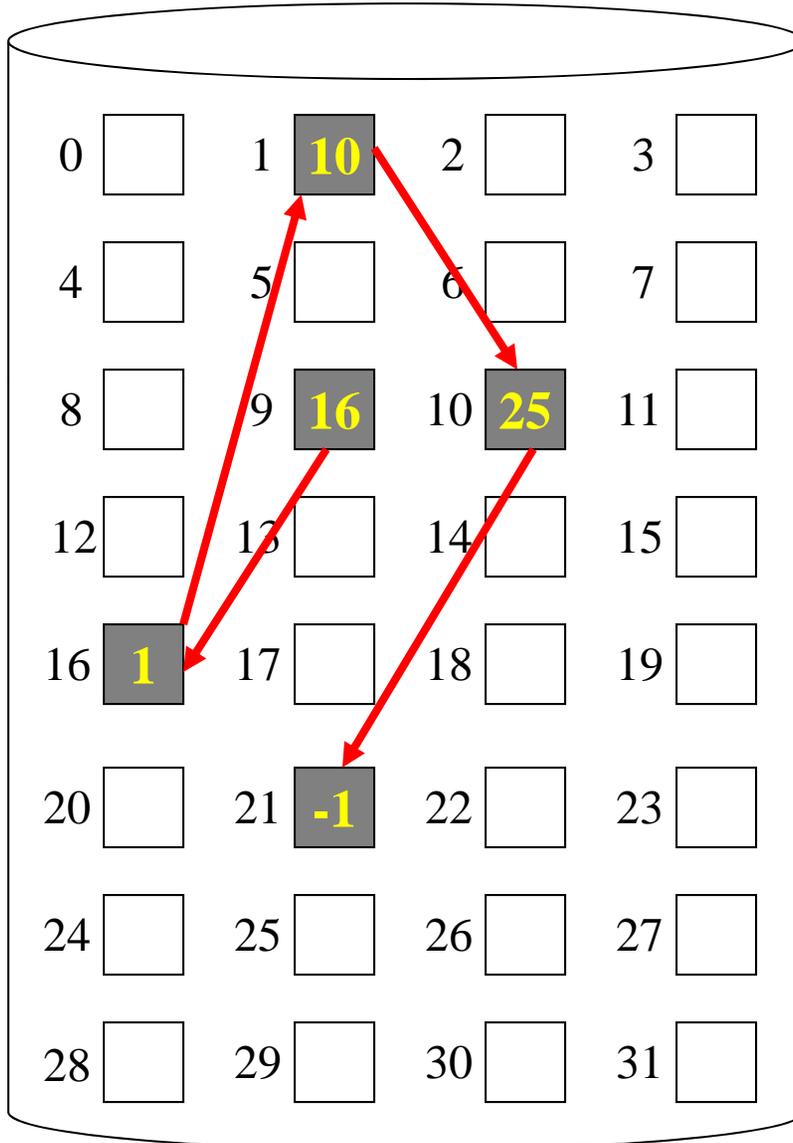
Ventajas/desventajas

- El acceso a un archivo que se asignó de forma contigua es fácil
- Una dificultad de la asignación contigua es encontrar espacio para un archivo nuevo
 - cómo satisfacer una solicitud de tamaño n a partir de una lista de huecos libres
 - estrategias de primer ajuste y mejor ajuste son muy usadas para encontrar un hueco libre

Asignación enlazada

- Resuelve todos los problemas de la asignación contigua
- Cada archivo es una lista enlazada de bloques de disco:
 - el directorio contiene un puntero al primer y último bloques del archivo
 - cada bloque contiene un puntero al siguiente bloque
 - estos punteros no se proporcionan al usuario
 - cada bloque tiene 512 bytes y una dirección de disco (el puntero) requiere cuatro bytes, el usuario verá bloques de 508 bytes

Ejemplo asignación enlazada



archivo	inicio	final
jeep	9	25

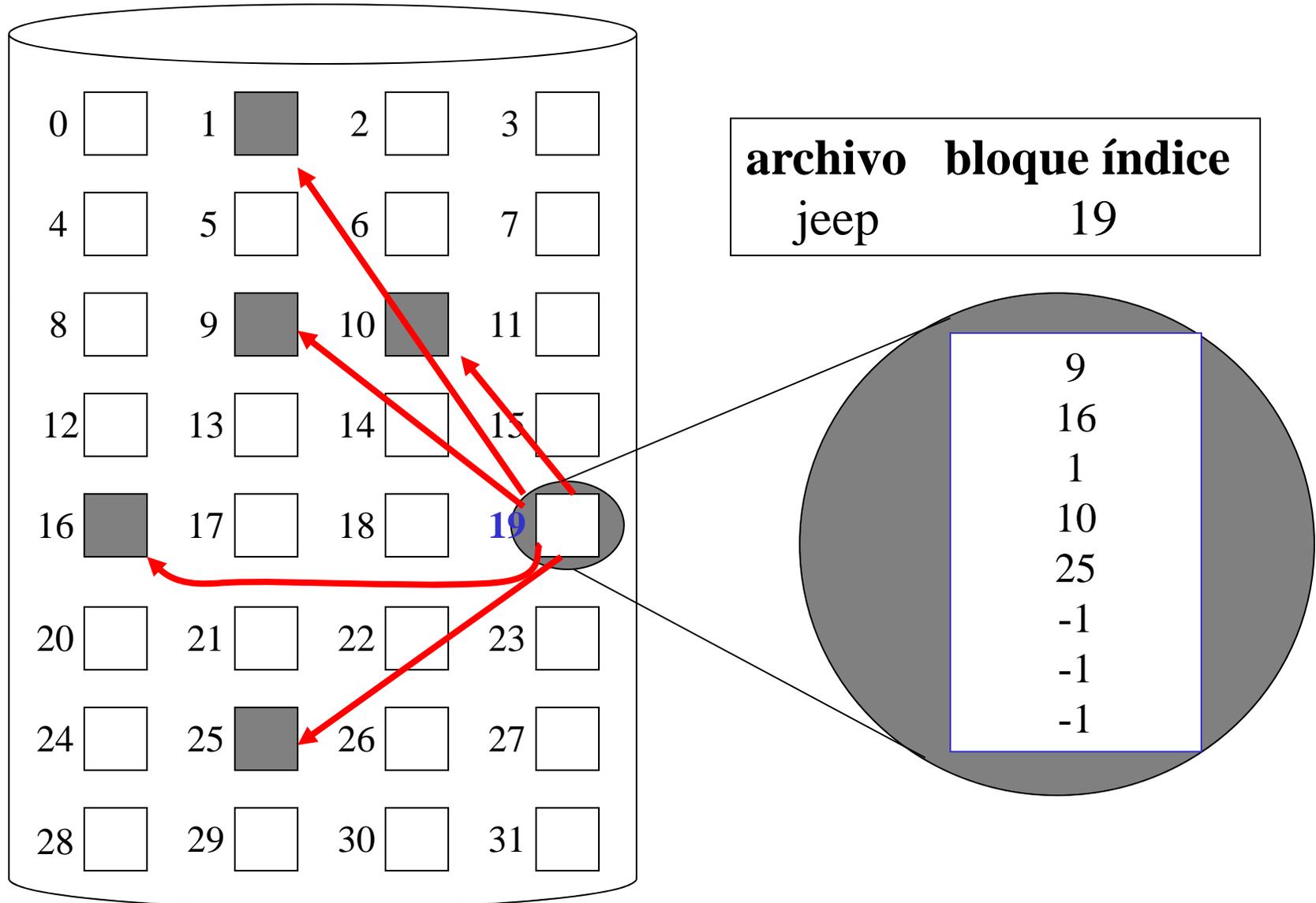
Creando un archivo nuevo

- Para crear un archivo nuevo se crea una nueva entrada en el directorio
- Cada entrada de directorio tiene un puntero al primer bloque de disco del archivo
 - inicialmente el puntero tiene un valor de nil para indicar un archivo vacío y se asigna un valor de cero al campo de tamaño

Asignación indizada

- Todos los punteros se reúnen en un solo lugar conocido con el nombre de bloque índice
- Cada archivo tiene su propio bloque índice, que es una matriz de direcciones de bloques de disco
 - la i -ésima entrada del bloque índice apunta al i -ésimo bloque del archivo
- El directorio contiene la dirección del bloque índice
- Cuando se crea el archivo, se asigna nul a todos los apuntadores del bloque índice

Ejemplo asignación indizada



Sistema Archivos Windows

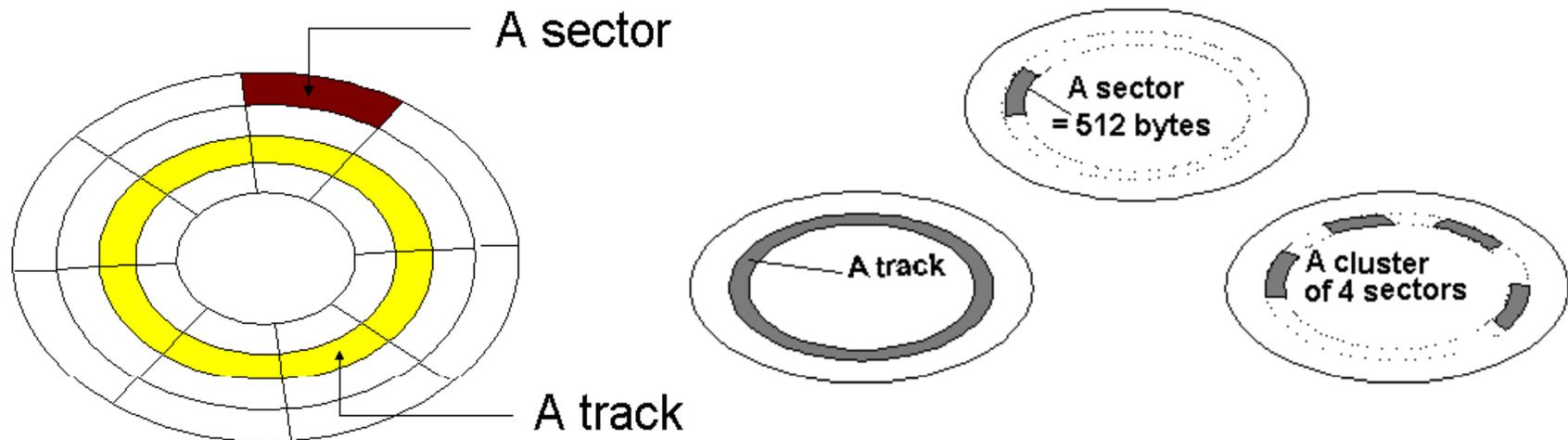
FAT y NTFS

El cluster

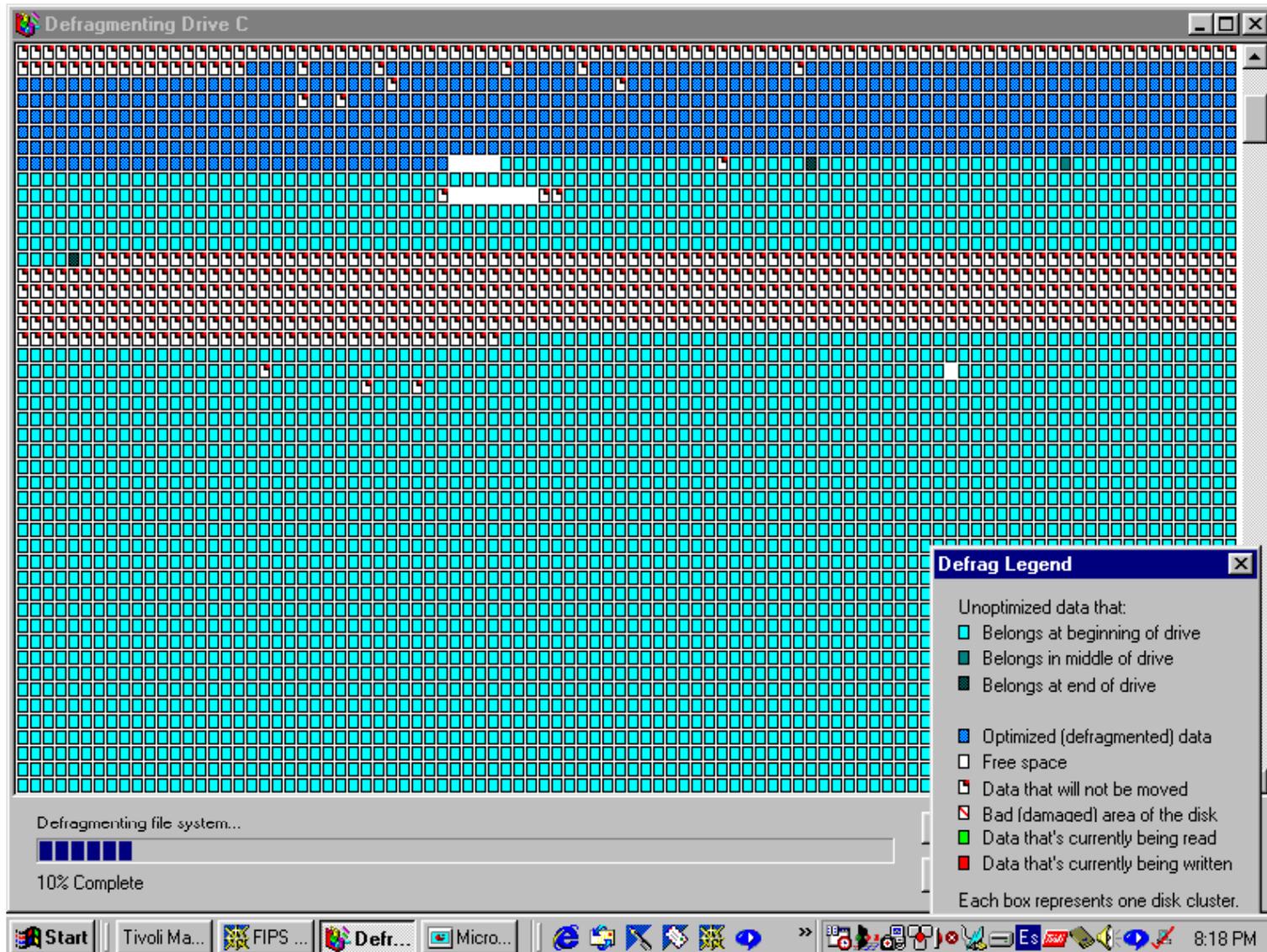
- La unidad más pequeña que cualquier software puede acceder es el sector (bloque en sistemas Unix).
- Por desempeño los sectores son agrupados en clusters o unidades de asignación.
- El tamaño del cluster es proporcional al tamaño del volumen del disco:
 - mayor es el volumen, mayor es el tamaño del cluster
 - discos duros: clusters desde 4 sectores a 64 sectores (en algunos casos hasta 128 sectores)
 - discos flexibles: cluster más chicos (en algunos casos un cluster ocupa un sector)

Clusters y sectores

- Sectores en un cluster están contiguos.
- El tamaño del cluster es determinado cuando el volumen del disco es particionado.
- A un archivo se le asigna un número entero de clusters



Ejemplo clusters en Windows



The screenshot shows the 'Defragmenting Drive C' window. The main area is a grid of small boxes representing disk clusters. A legend in the bottom right corner explains the colors and symbols used in the grid:

- Unoptimized data that:
 - Light blue: Belongs at beginning of drive
 - Dark blue: Belongs in middle of drive
 - Black: Belongs at end of drive
- Optimized (defragmented) data: Dark blue
- Free space: White
- Data that will not be moved: White box with a red 'X'
- Bad (damaged) area of the disk: White box with a red 'X' and a red border
- Data that's currently being read: Green
- Data that's currently being written: Red

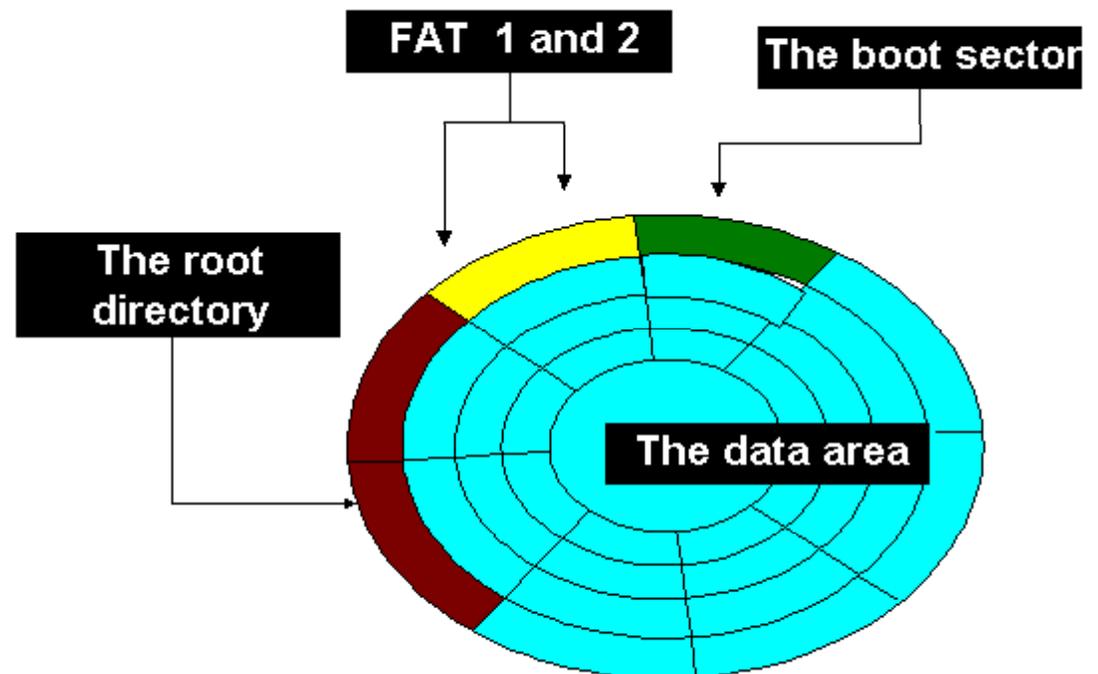
Each box represents one disk cluster.

At the bottom of the window, a progress bar shows 'Defragmenting file system...' and '10% Complete'.

The Windows taskbar at the bottom shows the Start button, several open applications (Tivoli Ma..., FIPS..., Defr..., Micro...), and the system tray with the time 8:18 PM.

Estructura de un volumen

- El registro de booteo (MBR), el cual siempre es el primer sector.
- Las áreas donde se ubica el FAT: usualmente son dos idénticas.
- El directorio raíz
- El área de datos

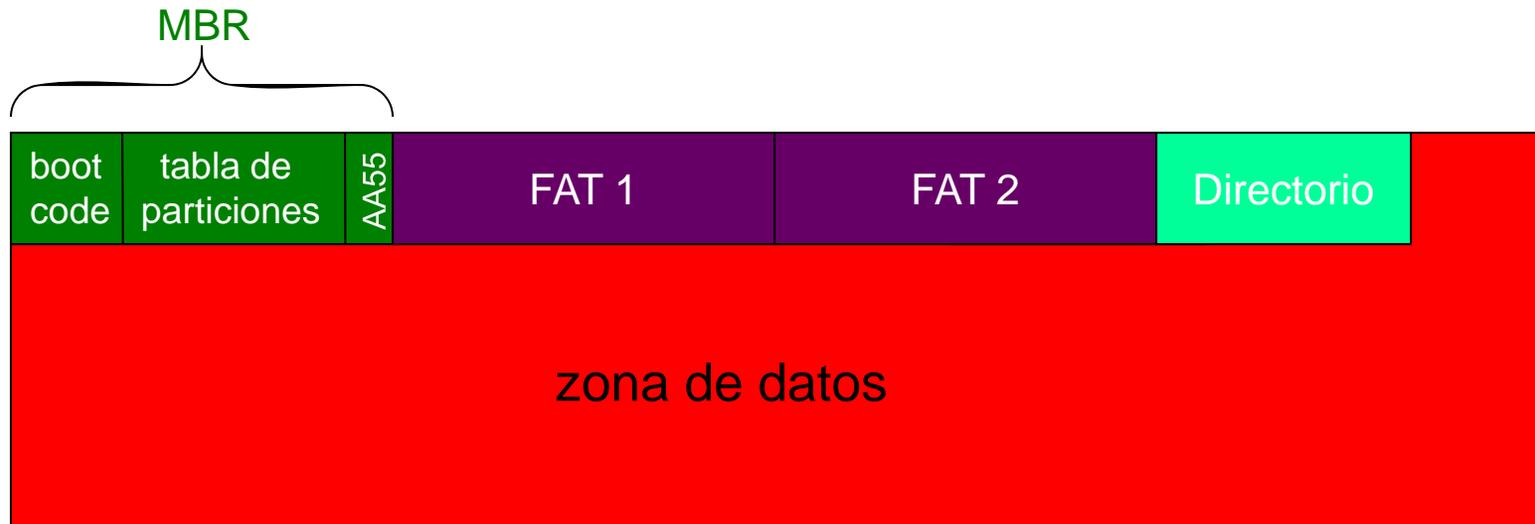


El MBR: Master Boot Record

- Primer sector del disco o sector de arranque principal
- Consta de tres partes:
 - El código de booteo:
 - del byte 1 al 446 (es lo que ejecuta el BIOS)
 - La tabla de particiones del disco:
 - mini lista de las particiones del disco
 - El número mágico AA55:
 - byte 511: 55
 - byte 512: AA
 - identifica a este sector como un sector de arranque

La Tabla de Archivos (FAT)

- Después del MBR sigue la tabla de archivos (FAT)
- Generalmente hay dos tablas, una de respaldo
- Consiste de una tabla de números
 - tiene 65,536 entradas
 - cada entrada contiene información acerca de un cluster en forma de un número.



Funcionamiento FAT

- Se usa de forma similar a una lista enlazada
- La entrada de directorio contiene el número de bloque del primer bloque del archivo.
- La entrada de la tabla indizada por un ese de bloque contiene el número del siguiente bloque del archivo.
- La cadena continúa hasta el último bloque
 - tiene un valor especial de fin de archivo como entrada de la tabla

Ejemplo tabla asignación archivos

entrada directorio

prueba	...	217
--------	-----	-----



0	
217	618
339	fin archivo
618	339
-1	

núm bloques en disco

FAT

El tamaño de FAT

- Cada cluster tiene una entrada en la FAT, el tamaño del área de FAT depende del tamaño del disco.
 - cada entrada en el FAT ocupa 16 bits
- Consideremos un disco de 160 MB
- El tamaño máximo del FAT es de 128KB,
 - 16 archivos, 2 bytes c/u: $65,536 \times 2 = 131,072$ bytes (128 KB)
 - Hay 40,400 clusters, ya que la partición es de 160 MB
- Se tienen dos FATs:
 - 40,400 x 2 bytes
 - esto nos da un total de 161,600 y eso ocupará 316 sectors

¿Y el directorio?

- Es la última área administrativa en el disco.
- Siempre hay 512 entradas de archivos en el directorio
 - es del mismo tamaño para todos los discos duros.
- La estructura del directorio consiste de un número de entradas de directorio.
 - cada entrada ocupa 32 bytes
 - las entradas son idénticas ya sea que estén en el directorio raíz o en algún subdirectorio
 - contienen información como:
 - el nombre del archivo (en el formato 8.3)
 - tamaño del archivo en bytes
 - fecha y hora de la última revisión

Estructura del directorio

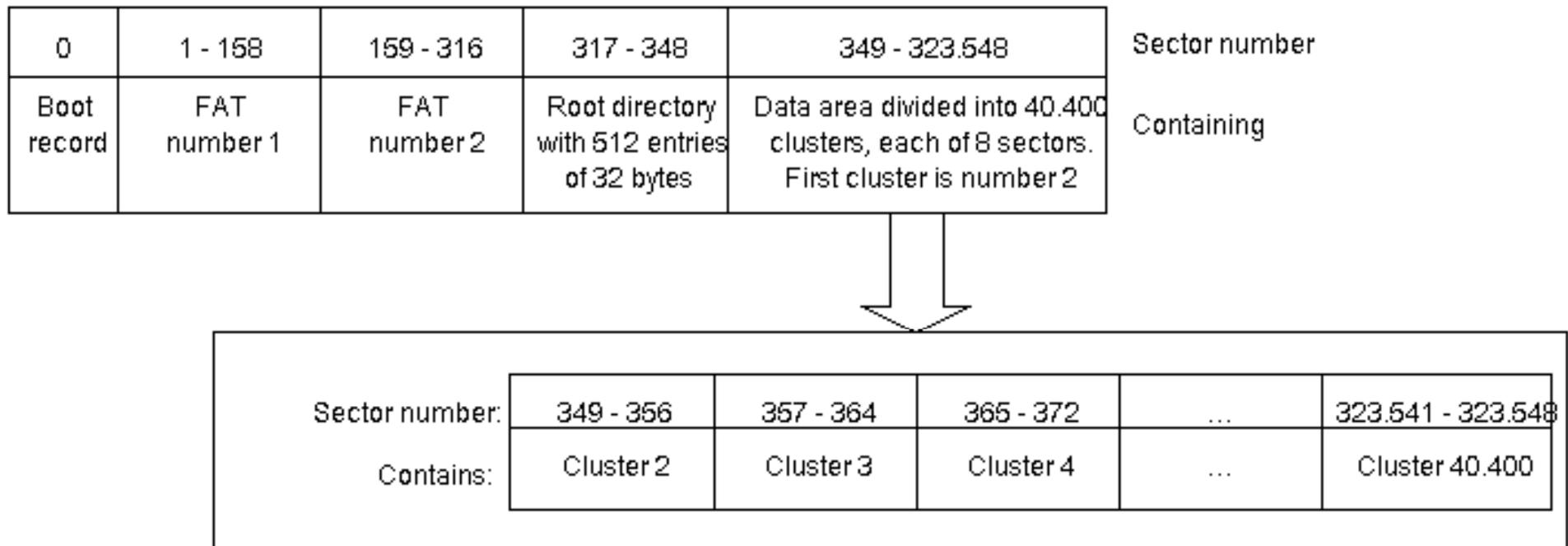
File name 8 bytes
Extension 3 bytes
Attribute 1 byte
Reserved 10 bytes (FAT32 uses two of them)
Time 2 bytes
Date 2 bytes
First cluster 2 bytes
File size 2 bytes

- Los 32 bytes estan agrupados en secciones
 - válido para todas las secciones, ya sea que se trate de archivos o directorios (directorio raíz y subdirectorios)
- Se cuenta con el número del primer cluster
 - importante ya que a partir de eso empieza a buscar al archivo
 - el primer cluster es leído de la entrada del directorio, los siguientes números de clusters son leídos del FAT
- En discos duros formateados como FAT16 el directorio raíz ocupa 512 entradas, las cuales son de 32 bytes cada una.
 - entonces ocupa 16 KB

El área de datos

- El resto del disco alberga la parte más importante, el área de datos, donde todos los archivos y subdirectorios son almacenados.
- El área de datos es la parte más grande del disco
- Los sectores del área de datos están conjuntados en clusters.
- Como se dijo antes, el máximo número de clusters para datos es $2^{16} = 65,535$
- Si el disco duro es de 160 Mb:
 - se tienen 40,400 clusters de 8 sectores cada uno

Un ejemplo de relación tabla particiones y FAT



Tipos de FAT

- VFAT
 - para versiones anteriores de Windows 95
- FAT 12
 - sectores de 512 bytes
 - sistema MS-DOS determina tamaño del FAT, basado en el número de clusters
 - si hay 4085 clustes o menos sistema usa tabla FAT-12
 - si hay 4086 o más clusters se utiliza FAT de 16 bits
- FAT 16
 - versiones Microsoft MS-DOS y posteriores permiten a FDISK particionar discos duros de hasta 4 gigabytes
 - sin embargo la tabla solo soporta 2GB por partición

El sistema de archivos NTFS

NT File System

Pasando de FAT a NTFS

- De FAT a NTFS

CONVERT [letra_unidad]: /FS:NTFS

NTFS

- Sistema de archivos estándar de Windows NT y de sus descendientes
 - 2000, 2003 y XP
- Versiones 9x (MS-DOS, W95, W98 y WME) no pueden leer este sistema de archivos de manera predeterminada
 - existen utilidades para salvar esta carencia.
- Tres versiones de NTFS:
 - v1.2 en NT 3.51 y NT 4 (v4.0)
 - v3.0 en Windows 2000 (v5.0)
 - v3.1 en Windows XP y Windows 2003 Server (v5.1)

Mejoras con respecto a FAT

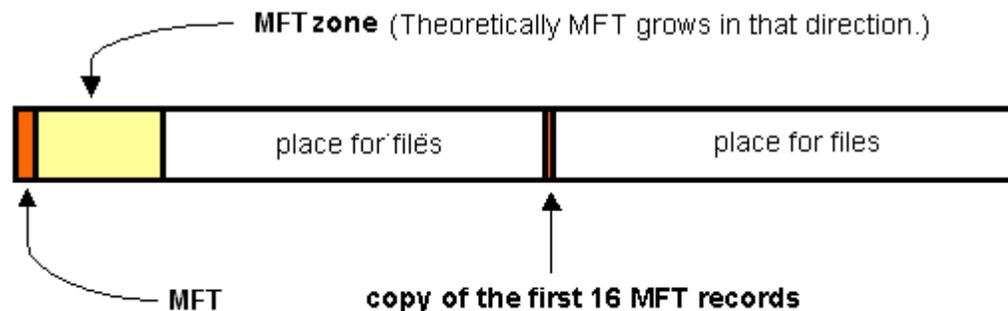
- Compatibilidad mejorada con metadatos
- Uso de estructura de datos avanzadas (árboles-B) para optimizar el rendimiento, estabilidad, y el aprovechamiento del espacio en disco,
- Listas de control de acceso
- Registro de transacciones (journaling).
- Seguridad a nivel de archivo y carpeta
- Compresión de disco
- Encriptación de datos

Tamaño partición y cluster

Rango tamaño partición (GiB)	Número de sectores por cluster por default	Tamaño por default del cluster
≤ 0.5	1	0.5
>0.5 a 1.0	2	1
>1.0 a 2.0	4	2
>2.0 a 4.0	8	4
> 4.0 a 8.0	16	8
> 8.0 a 16.0	32	16
> 16.0 a 32.0	64	32
> 32.0	128	64

Estructura de una partición

- Espacio dividido en clusters
- Disco NTFS está simbólicamente dividido en dos partes
 - Primera parte (12%) asignado al área MFT
 - Master File Table
 - no es posible grabar información en este espacio
 - Segunda parte (88%) espacio para archivos
 - todo tipo de información, inclusive info de MFT



Meta-archivos

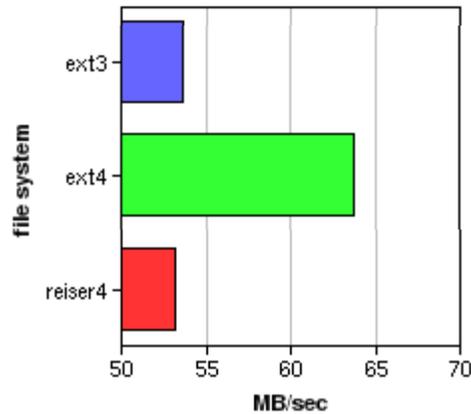
Archivo	Función
\$MFT	el MFT en sí
\$MFTmirr	copia de los primeros 16 registros del MFT
\$LogFile	archivo de soporte de bitácoras
\$Volume	información del volumen, nombre, versión sistema archivos, etc
\$AttrDef	lista de los atributos estandar de los archivos
\$.	directorio raíz
\$Bitmap	bitmap de espacio libre
\$Boot	sector de booteo (partición booteable)
\$Quota	archivo donde los derechos de los usuarios sobre el uso del espacio en disco se almacenan (empieza a funcionar en la v5)
\$Upcase	tabla de correspondencia entre las letras minúsculas y mayúsculas

Los sistemas de archivos de Linux

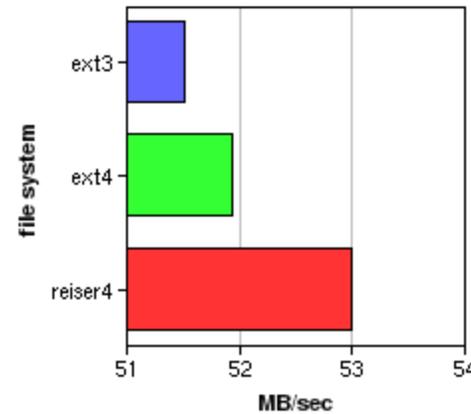
Nombre	Creador	Año introducción	Log. Máxima nombre archivo	Caracteres permitidos en entradas directorio	Long. máxima pathname	Long. máxima archivo	Long. máxima volumen
ext2	Remy Card	1993	255 bytes	cualquiera excepto NUL	No limite definido	16 Gb a 2 Tb	2 Tb a 32 Tb
ext3	Stephen Tweedie	1999	255 bytes	cualquiera excepto NUL	No limite definido	16 Gb a 2 Tb	2 Tb a 32 Tb
ext4	Andrew Morton	2006	255 bytes	cualquiera excepto NUL	No limite definido	16 Gb a 2 Tb	1024 Pb
reiser FS	Namesys	2001	4032 bytes/255 c	cualquiera excepto NUL	No limite definido	4Gb a 8Tb	16 Tb
reiser4	Namesys	2004	3976 bytes	cualquiera excepto NUL	No limite definido	8Tb en x86	?
GFS	Sistina (Red Hat)	2000	255 bytes	cualquiera excepto NUL	No limite definido	2Tb a 8Eb	2Tb a 8Eb
OCFS	Oracle Corporation	2002	255 bytes	cualquiera excepto NUL	No limite definido	8Tb	8Tb
OCFS2	Oracle Corporation	2005	255 bytes	cualquiera excepto NUL	No limite definido	4Pb	4Pb
GFS*	Google	2003					
NILFS	NTT	2005					

Benchmark ext4

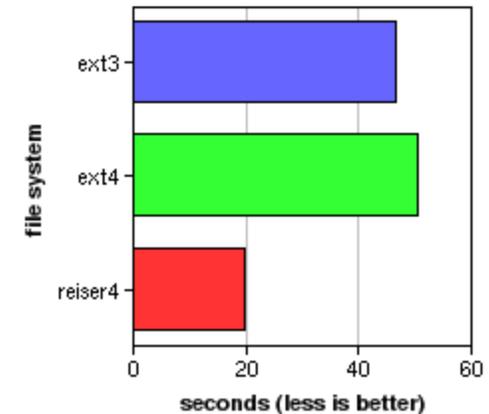
Sequential writing



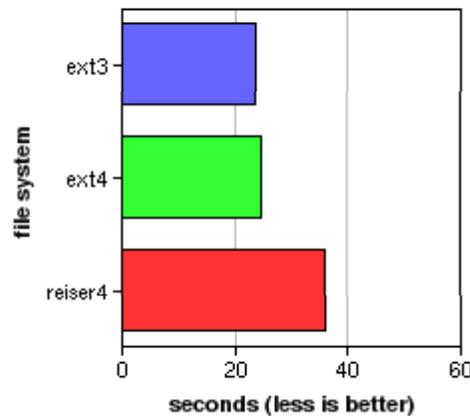
Sequential reading



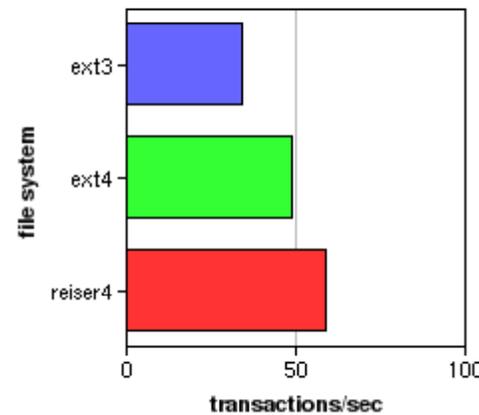
Creating many small files



Deleting many small files



Postmark



set read 4096
set write 4096
set transactions 10000
set size 500 500000
set number 5000

Sistema archivos ext2

- Sistema de archivos estándar en Linux por varios años y continúa siendo ampliamente utilizado.
 - diseñado originalmente por Rémy Card.
- La principal desventaja de EXT2 es que no posee una bitácora
 - muchos usuarios emigran a ReiserFS y su sucesor EXT3.
- Aunque no es leído por Windows, hay varias utilidades para acceder al EXT2 desde Windows
 - Ext2 IFS For Windows NT4.0 a XP (<http://www.fs-driver.org/>)
 - Explore2fs
(<http://uranus.it.swin.edu.au/~jn/linux/explore2fs.htm>)

Sistema archivos ext2

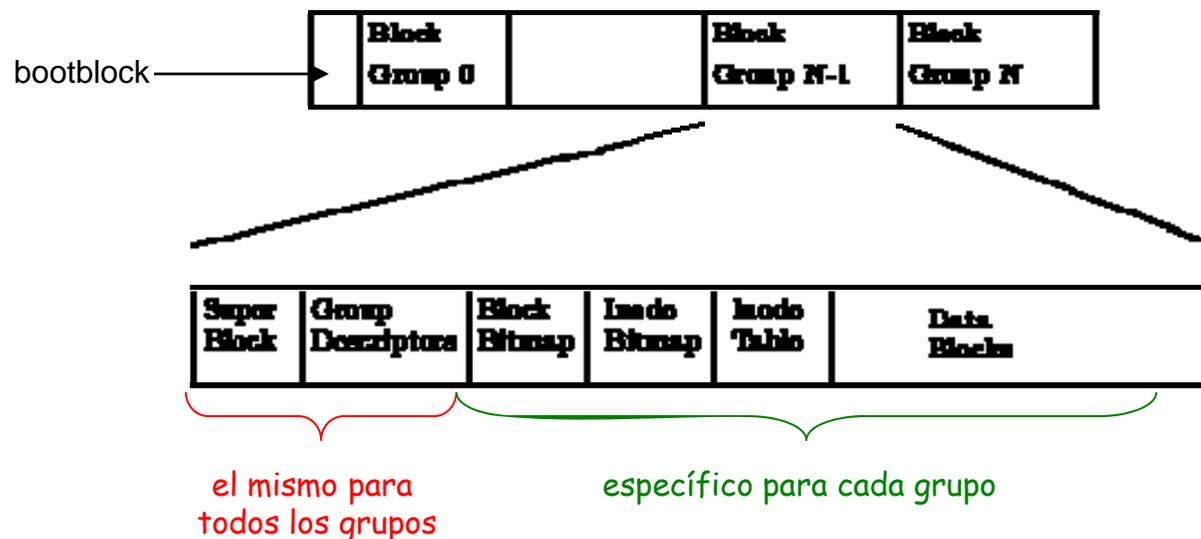
- El ext2 tiene un tamaño de i-nodo fijo entre 1 y 4K, independientemente del tamaño de la partición.
- El tamaño del i-nodo se selecciona al crear el sistema de archivos y es seleccionable por el usuario.
- El ext2 tiene una unidad similar al cluster, llamada bloque, y que es, por lo general de 1K, especificable por el usuario e independiente del tamaño de la partición,
 - asegura un buen aprovechamiento del espacio libre con archivos pequeños.

Sistema archivos ext2

- El ext2 no usa una FAT, sino una tabla de i-nodos distribuidos en un número determinable de grupos a través de la superficie,
 - permite balancear la distribución de los bloques de archivos en la superficie a través de dichos grupos para asegurar la mínima fragmentación.
- El ext2 tiene un límite máximo de 4GB de archivo, pero no limita el tamaño

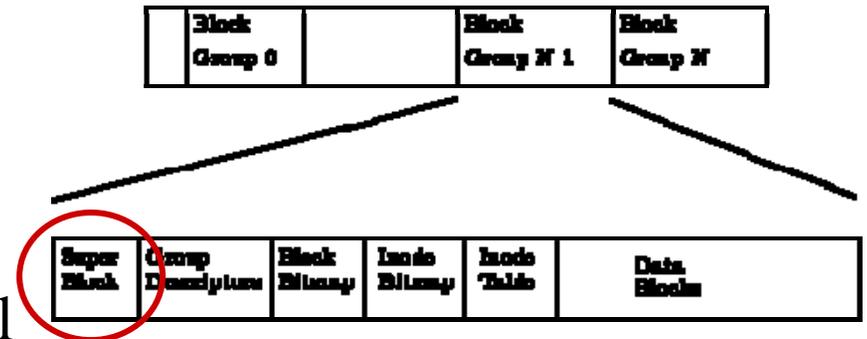
Los superbloques

- Sistema divide la partición lógica que ocupa en grupos de bloques
- Cada bloque contiene una copia de la información crítica para la integridad del sistema archivos
 - copia del superbloque, y el descriptor del sistema de archivos



El superbloque ext2

- Contiene una descripción del tamaño básico y alcance del sistema de archivos
- Información contenida permite al sistema de archivos para usar y mantener el sistema de archivos.
- Usualmente el superbloque en el grupo de bloques 0, se lee cuando el sistema de archivos se monta
 - pero cada grupo de bloques contiene una copia de duplicado en el caso de una corrupción del archivo



Campos superbloque

- Numero mágico
 - permite al software de montaje verificar que el superbloque es un sistema archivos EXT2
 - para EXT2 actual este es 0xEF53
- Nivel revisión
 - permite verificar si sistema archivos soporta características que solo se encuentran disponibles en revisiones del sistema archivos
- Mount Count y Maximum Mount Count
 - permiten determinar si el sistema de archivos debe ser verificado por completo
 - mount count es incrementado cada vez que el sistema es montado y cuando iguala a maximum count:

maximal mount count reached, running e2fsck is recommended

Campos superbloque (cont.)

- **Block Group Number**
 - El numero de grupo de bloque que almacena la copia de este superbloque
- **Block size**
 - tamaño del bloque en este sistema archivos
- **Bloques por grupo**
 - número de bloques en un grupo, al igual que el tamaño del bloque se asigna cuando el sistema de archivos se crea
- **Free blocks**
 - numero de bloques libres en el sistema de archivos
- **Free Inodes**
 - numero de inodes libres en el sistema de archivos
- **First inode**
 - número de inode en el primer inode en el sistema de archivos
 - el primer inode en un sistema archivos raíz EXT2 es la entrada del directorio raíz (/)

El descriptor de Grupo de EXT2

- Estructura de datos que describe al grupo
- Se encuentra duplicado en cada grupo de bloques
- Cada descriptor contiene la información siguiente

– Blocks Bitmap

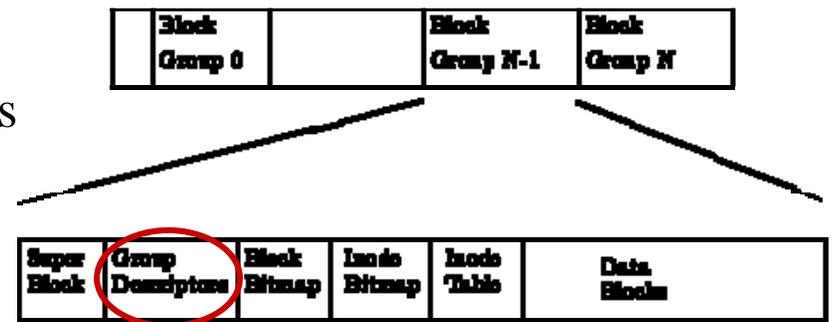
- numero de bloque que contiene el bitmap para este grupo de bloques
- usado durante la asignación y desasignación de bloques

– Inode Bitmap

- bitmap de los inodes
- usado durante asignación y desasignación de inodes

– Inode Table

- numero de bloque del bloque donde inicia la tabla de inodes para el grupo de bloques



La estructura del descriptor de grupo

```
struct EXT2_GROUP_DESC
```

```
{
```

```
    DWORD    bg_block_bitmap;
```

The block which contains the block bitmap for the group.

```
    DWORD    bg_inode_bitmap;
```

The block contains the inode bitmap for the group

```
    DWORD    bg_inode_table;
```

The block contains the inode table first block (the starting block of the inode table.).

```
    WORD     bg_free_blocks_count;
```

Number of free blocks in the group.

```
    WORD     bg_free_inodes_count;
```

Number of free inodes in the group.

```
    WORD     bg_used_dirs_count;
```

Number of inodes allocated to the directories.

```
    WORD     bg_pad;
```

Padding (reserved).

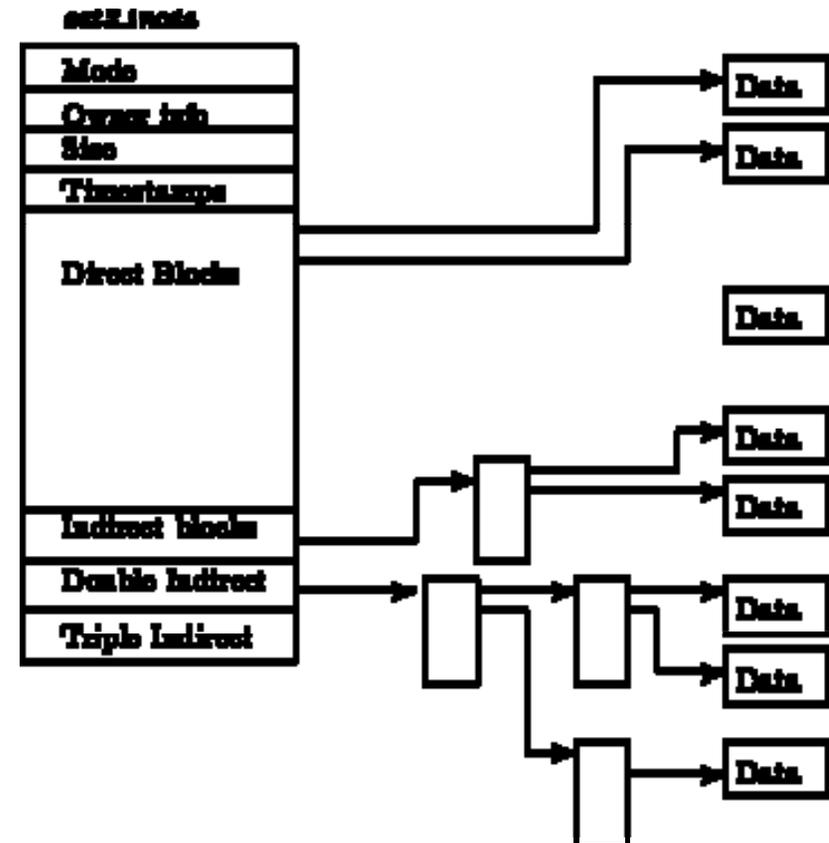
```
    DWORD    bg_reserved[3];
```

Reserved.

```
};
```

El inode en ext2

- mode
 - tipo archivo y permisos
- owner information
 - identificadores usuario y grupo
- size
 - tamaño archivo en bytes
- timestamps
 - tiempo creación y ultima modificación
- datablocks
 - apuntadores a bloques que contienen los datos que el inode esta describiendo
 - último tres son más niveles de inderección



Conociendo el nodo i de un archivo

- Opción `-i` del comando `ls`

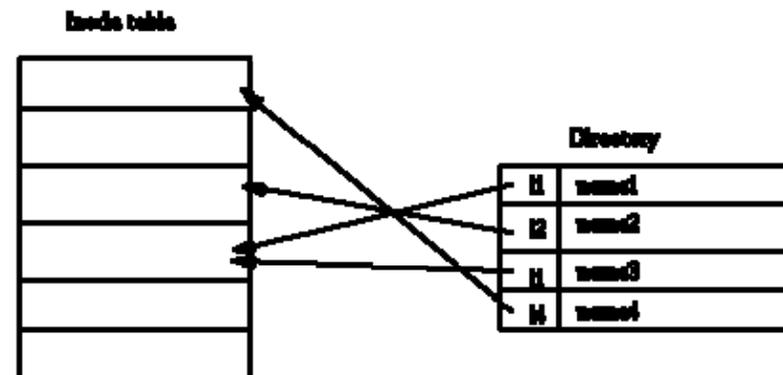
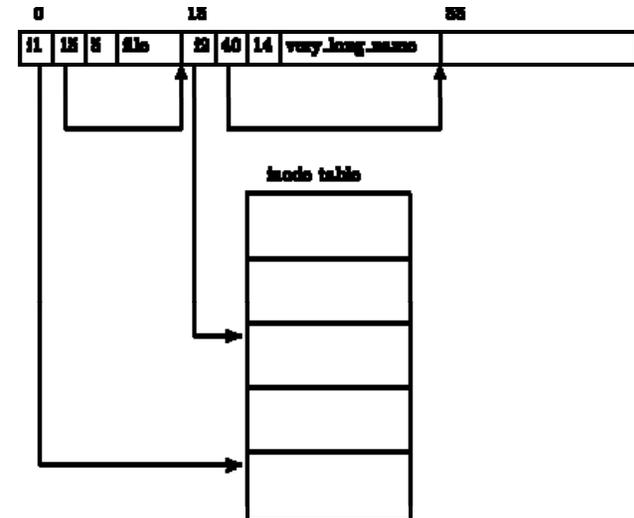
```
$ ls -i /etc/passwd  
  
32820 /etc/passwd
```

- Comando `stat`

```
$ stat /etc/passwd  
  
File: `/etc/passwd'  
Size: 1988          Blocks: 8          IO Block: 4096   regular file  
Device: 341h/833d   Inode: 32820       Links: 1  
Access: (0644/-rw-r--r--)  Uid: (  0/   root)   Gid: (  0/   root)  
Access: 2005-11-10 01:26:01.000000000 +0530  
Modify: 2005-10-27 13:26:56.000000000 +0530  
Change: 2005-10-27 13:26:56.000000000 +0530
```

Directorios EXT2

- Archivos especiales usados para crear y contener paths de acceso a los archivos del sistema
- Información
 - inode
 - inode para el directorio
 - name length
 - longitud del directorio en bytes
 - name
 - nombre del directorio
- Dos primeras entradas de cada directorio son: “.” y “..”



Ejemplo entrada directorio

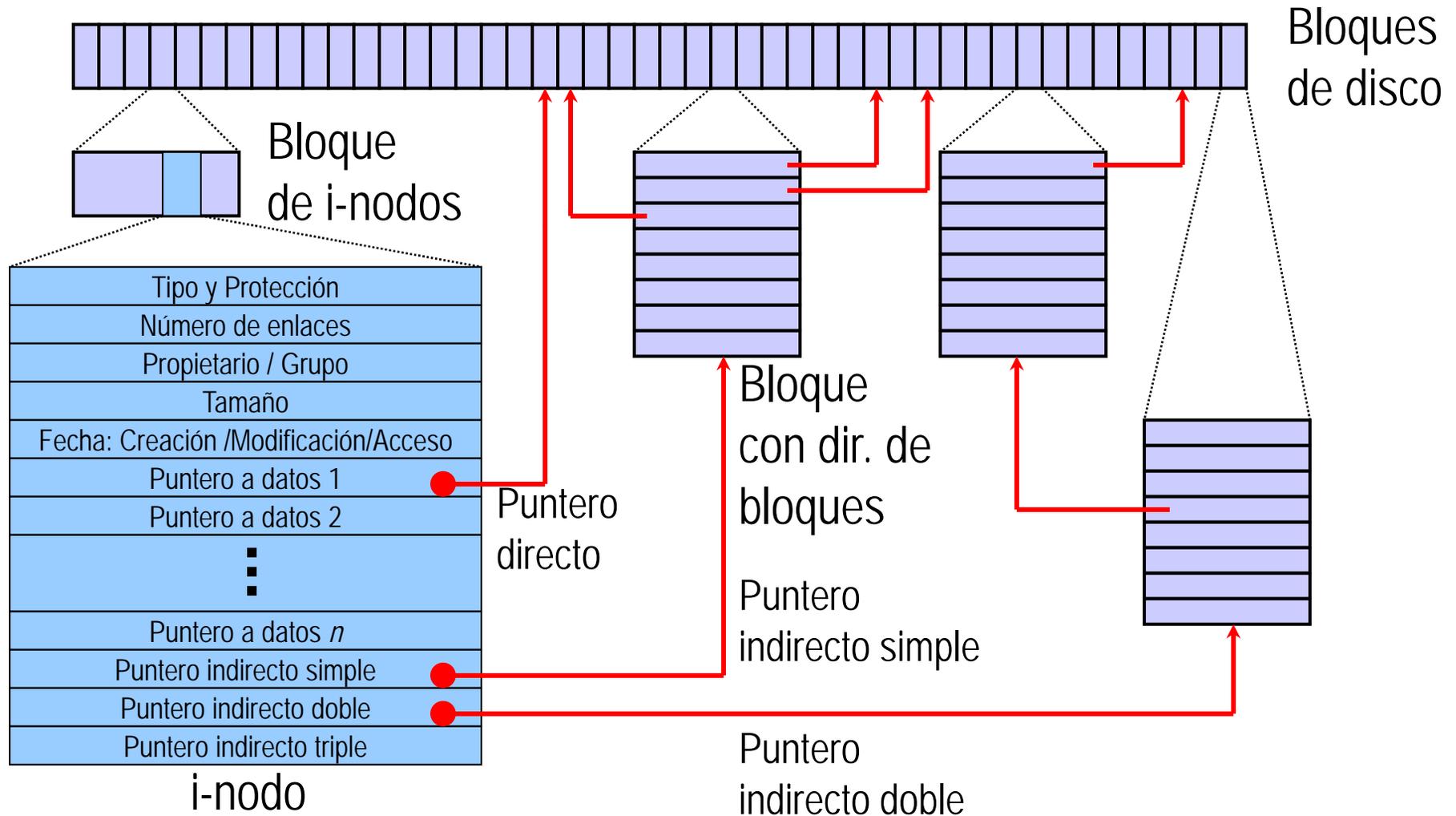
- Formato de la entrada del directorio



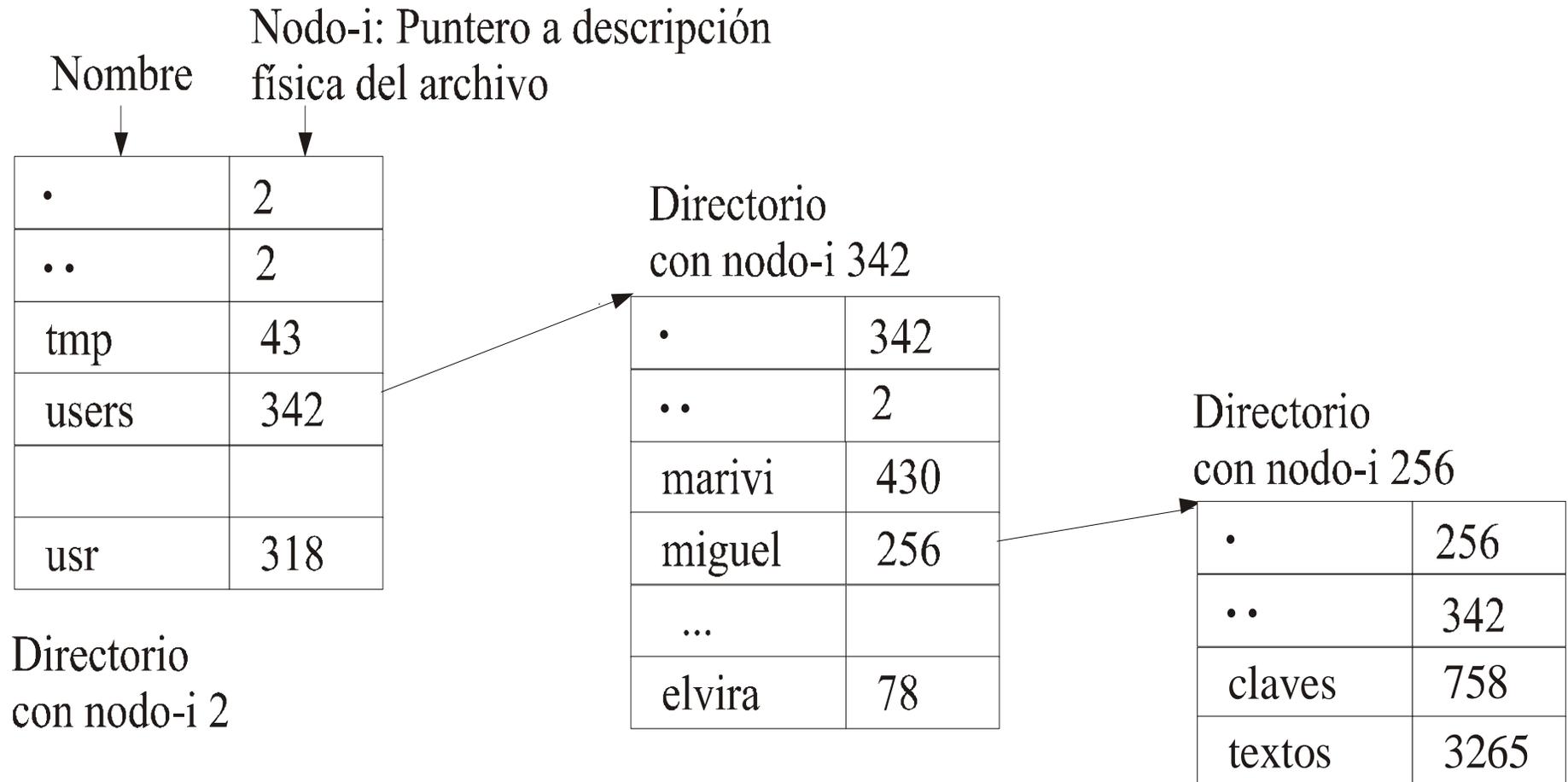
- Ejemplo de un directorio que cuenta con tres archivos:
file1, long_file_name y f2

i1	16	05	file1
i2	40	14	long_file_name
i3	12	02	f2

Descripción física en UNIX (i-nodo)



Interpretación de nombres en Linux



Ejemplo búsqueda archivo

- Considerando: `/home/toto/.cshrc`
 - Primer inode: el de la raíz del sistema archivos
 - Se encuentra en el superbloque del sistema archivos
 - para encontrar el inode se debe leer en la tabla de inodes del grupo de bloques apropiado
 - p.e. número inode es 41, es necesario el 42avo. inode de la tabla de inodes del Grupo de Bloques 0
 - inode raíz es un directorio que contiene entradas de directorio
 - dentro de las entradas se encuentra home
 - se lee las entradas de home para encontrar toto
 - se lee las entradas de toto para encontrar `.cshrc`
 - de esta última se obtiene los bloques que contienen la información del archivo

Interpretación de nombres

