

SISTEMAS OPERATIVOS: SISTEMAS DE FICHEROS

Ficheros

Objetivos

2

- Conocer los conceptos de fichero y directorio así como sus características.
- Utilizar los servicios de gestión de Ficheros y directorios ofrecidos por el sistema operativo.
- Comprender la estructura de un sistema de ficheros.
- Comprender los mecanismos en los que se apoya un servidor de ficheros y aplicarlos a ejercicios sencillos.

Contenido

3

- Fichero.
- Atributos y operaciones.
- Vista lógica.
- Semántica de compartición.
- Representación.

Almacenamiento

4

- Memoria principal.

- Memoria volátil → datos no persistentes.

- Datos accedidos por el procesador.

↳ Desaparecen al apagar el sistema

- Memoria secundaria.

- Memoria no volátil → datos persistentes.

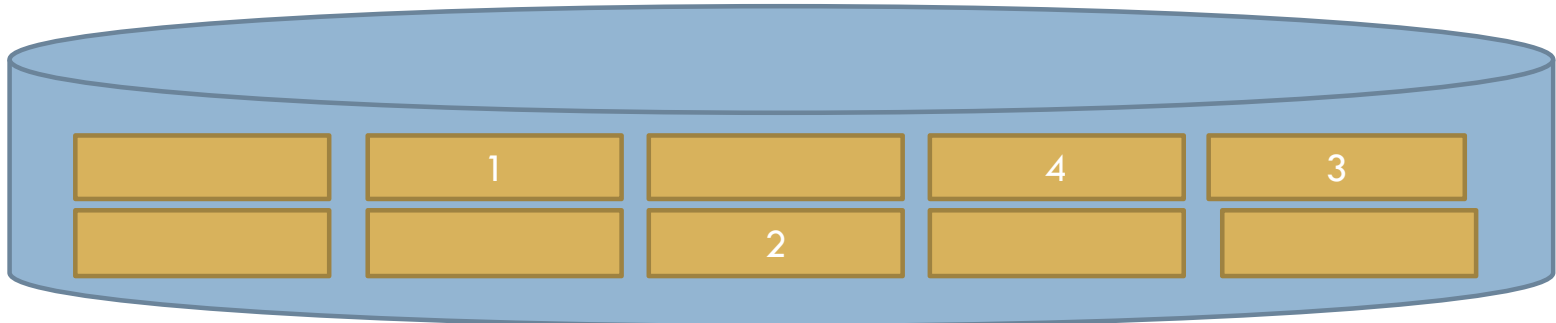
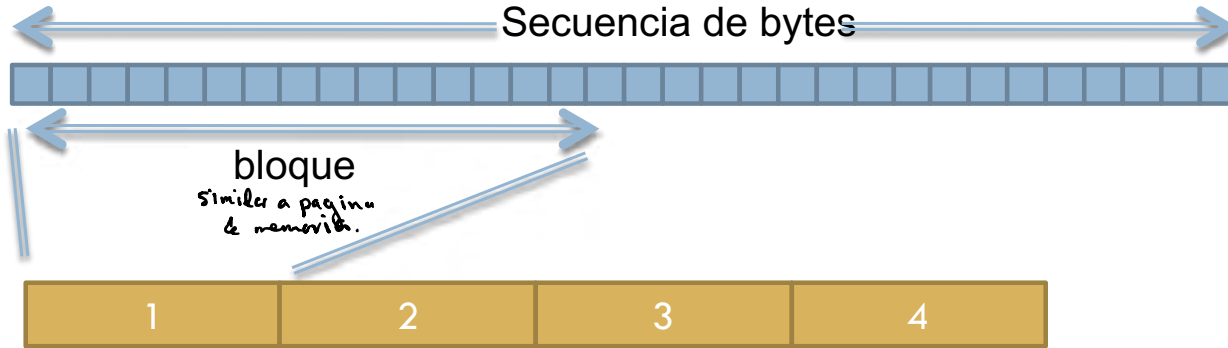
- Organizada en bloques de datos.

- Se necesita una abstracción para simplificar las aplicaciones: Fichero.

Fichero

Fid = lo importante para el sistema no con el que crea el i-node
Debe conocer tamaño fichero y de bloque para saber el n° de bloques
Nombre
Identificación (nombre)
leer aunque no lepa wales.
Dispositivo
Tamaño bloque
Donde está cada bloque { ... : 13 2 -> blo 35
Permisos
Owner
update time
3 2 -> blo 36
Cuando se lee o escribe se opera a nivel de bloque, no de byte. Se coge uno completo.

5



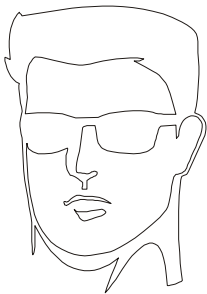
El sistema de ficheros

- Ofrece al usuario una visión lógica simplificada del manejo de los dispositivos periféricos en forma de ficheros.
- Proporciona un mecanismo de abstracción que oculta los detalles relacionados con el almacenamiento y distribución de la información en los periféricos.
- Constituye la parte del S.O. que gestiona los ficheros.
- Funciones:
 - Organización
 - Almacenamiento
 - Recuperación
 - Gestión de nombres
 - Implementación de la semántica de Coutilización
 - Protección

Función principal

7

- El SF establece una correspondencia entre los ficheros y los dispositivos lógicos.

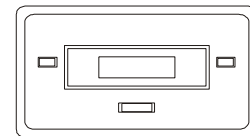
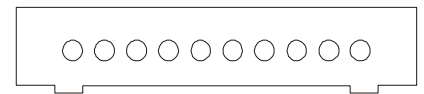
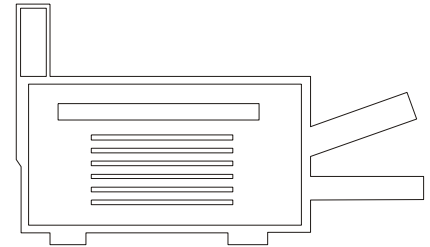


Interfaz de acceso

SISTEMA
DE
FICHEROS

Protección

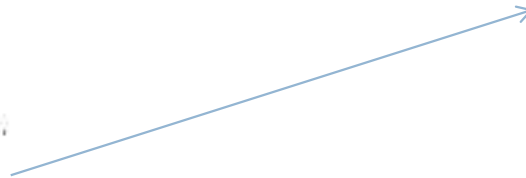
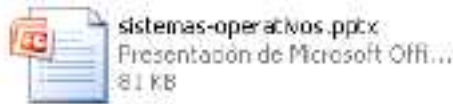
Sistemas Operativos - Ficheros



Sistema de ficheros: Visión del usuario

- Visión lógica:
 - Ficheros
 - Directorios
 - Sistemas de Ficheros y particiones
- Visión física:
 - Bloques o bytes ubicados en dispositivos

→ Simula discos a partir de uno.
 → La tabla de particiones almacena los nombres, bloque ini y fin de cada partición.



Características para el usuario

9

- Almacenamiento permanentes de información.
 - No desaparece aunque se apague el computador.
- Conjunto de información estructurada de forma lógica según criterios de aplicación.
- Nombres lógicos y estructurados.
- No están ligados al ciclo de vida de una aplicación particular.
- Abstraen los dispositivos de almacenamiento físico.
- Se acceden a través de llamadas al sistema operativo o de bibliotecas de utilidades.

Contenido

10

- Fichero.
- **Atributos y operaciones.**
- Vista lógica.
- Semántica de compartición.
- Representación.

Atributos de un fichero

11

- **Nombre:** Identificador en formato legible por una persona.
- **Identificador:** Etiquetan unívoca del archivo
 - Suele ser numérico.
- **Tipo de fichero:** necesario en sistemas que proporcionan distintos formatos de Ficheros. Como mínimo se suele diferenciar el atributo de ejecutable.
- **Ubicación:** Identificación del dispositivo de almacenamiento y la posición dentro del dispositivo.
- **Tamaño del fichero:** número de bytes en el fichero, máximo tamaño posible, etc.
- **Protección:** control de accesos y de las operaciones sobre el fichero.
- **Información temporal:** de creación, de acceso, de modificación, etc.

Nombres de fichero y extensión

12

- Muy importante para los usuarios. Es característico de cada sistema de Ficheros.
- Problema: usar nombre lógicos basados en tiras de caracteres.
- Motivo: los usuarios no recuerdan el nombre 001223407654
- Tipo y longitud cambian de un sistema a otro:
 - Longitud: fija en MS-DOS o variable en UNIX, Windows.
 - Extensión: obligatoria o no, más de una o no, fija para cada tipo de Ficheros, etc.
- Sensibles a tipografía. Ejemplo: CATALINA y catalina son el mismo fichero en Windows pero distintos en LINUX.
- El sistema de ficheros trabaja con descriptores internos, sólo distingue algunos formatos (ejecutables, texto, ...). Ejemplo: número mágico UNIX.

Nombres de fichero y extensión

13

- Los directorios relacionan nombres lógicos y descriptores internos de ficheros
- Las extensiones son significativas para las aplicaciones (html, c, cpp, etc.)

Name	Size	Type	Modified
My Pictures		File Folder	07/09/2000 11:36
My Webs		File Folder	06/09/2000 11:57
pstr-inf_files		File Folder	14/09/2000 16:21
.fvwmrc	13 KB	FvWMRC File	06/05/1999 18:00
cartacas.tex	1 KB	COREL Texture	06/05/1999 17:55
cata99.ps	193 KB	PS File	06/05/1999 17:55
control.bib	16 KB	BIB File	06/05/1999 17:55
faxing.log	4 KB	Text Document	06/05/1999 17:55
fig3-1.tif	734 KB	Corel PHOTO-PAIN...	22/08/2000 11:59
fig3-7.cdr	27 KB	CDR File	03/05/2000 18:27
pstr-inf.doc	53 KB	Microsoft Word Doc...	14/09/2000 16:21
pstr-inf.htm	1 KB	Microsoft HTML Doc...	14/09/2000 9:51
remain.zip	0 KB	WinZip File	30/05/2000 12:34
Sample.jpg	10 KB	Corel PHOTO-PAIN...	05/09/2000 17:08
winamp265.exe	2.112 KB	Application	07/09/2000 13:10
cmutex.cpp	3 KB	CPP File	11/07/2000 15:30
secobject.c	2 KB	C File	14/07/2000 12:52
adasm pkg.adb	13 KB	ADB File	24/02/2000 9:49
Demo.ppt	345 KB	Microsoft PowerPoi...	24/07/1998 8:15
vol3tc04.html	10 KB	Microsoft HTML Doc...	22/12/1999 11:28
remain.pdf	4.110 KB	Adobe Acrobat Doc...	07/04/1999 11:11

Sistemas Operativos - Ficheros

Operaciones sobre ficheros

14

- Creación: Asignación de espacio inicial y metadatos.
- Borrado: Liberación de recursos asociados.
- Escritura: Almacena información en el fichero.
- Lectura: Recupera información del fichero.

Operaciones adicionales dependiendo de la semántica concreta de acceso a ficheros

Sistema de Ficheros

15

- El acceso a los dispositivos es:
 - Incómodo
 - Detalles físicos de los dispositivos
 - Dependiente de las direcciones físicas
 - No seguro
 - Si el usuario accede a nivel físico no tiene restricciones
- El sistema de Ficheros es la capa de software entre dispositivos y usuarios.
- Objetivos:
 - Suministrar una visión lógica de los dispositivos.
 - Ofrecer primitivas de acceso cómodas e independientes de los detalles físicos.
 - Mecanismos de protección.

Contenido

16

- Fichero.
- Atributos y operaciones.
- **Vista lógica.**
- Semántica de compartición.
- Representación.

Estructura del fichero

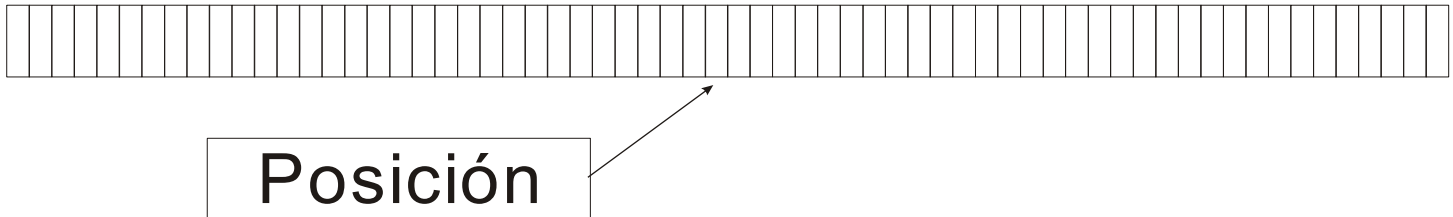
17

- Ninguna - secuencia de palabras o bytes (UNIX)
- Estructura sencilla de registros
 - Líneas
 - Longitud fija o'
 - Longitud variable
- Estructuras complejas
 - Documentos con formato (HTML, postscript)
 - Fichero de carga reubicable (módulo de carga)
- Se puede simular estructuras de registro y complejas con una estructura plana y secuencias de control
- ¿Quién decide la estructura?
 - Interna: El sistema operativo
 - Externa: Las aplicaciones

Ficheros: visión lógica

18

- Conjunto de información relacionada que ha sido definida por su creador
- Estructura de un fichero:
 - Secuencia o tira de bytes (UNIX, POSIX)

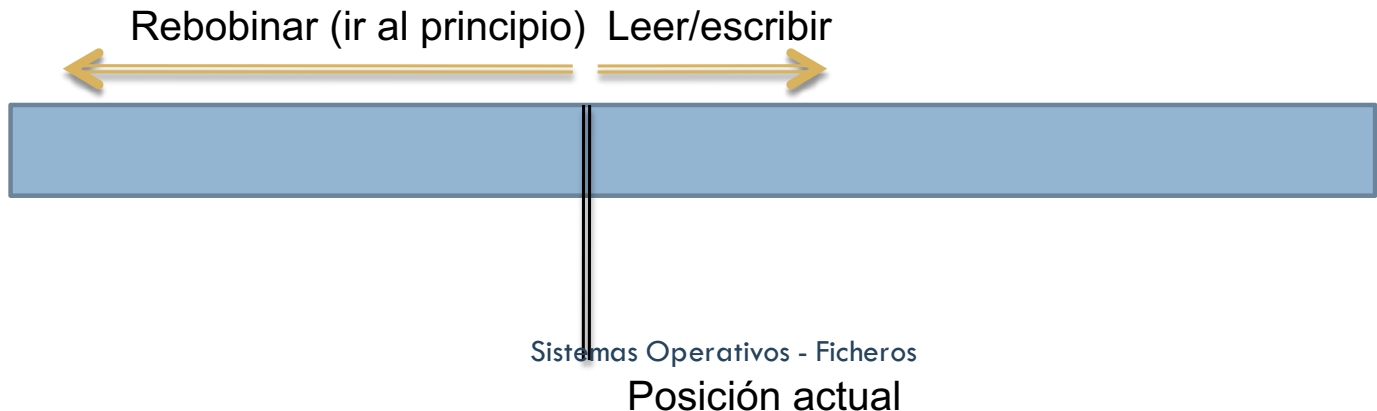


Métodos de acceso

19

□ Acceso secuencial

- Basado en el modelo de acceso a datos en una cinta magnética.
- Utilizable en dispositivos de acceso secuencial o directo.
- Operaciones orientadas a bytes o a registros.



Métodos de acceso

20

□ Acceso directo

- ▣ Basado en el modelo de acceso a dispositivo de disco.
- ▣ Fichero dividido en registros de longitud fija.
- ▣ Se puede especificar el número de registro para las operaciones de lectura y escritura.
- ▣ Se puede utilizar un puntero de posición para evitar tener que especificar la posición en todas las operaciones.
- ▣ Permite construir sobre él otros métodos de acceso más complejos (ejemplo: secuencial indexado).

Contenido

21

- Fichero.
- Atributos y operaciones.
- Vista lógica.
- **Semántica de compartición.**
- Representación.

Compartición de ficheros

22

- Varios procesos pueden acceder simultáneamente a un fichero

Ambos pueden leer a la vez, pero no escribir simultáneamente.

- Es necesario definir una semántica de coherencia.

Señales

- ¿Cuándo son observables por otros procesos las modificaciones a un fichero?

- Opciones:

- Semántica UNIX.

- Semántica de sesión.

- Semántica de archivos inmutables, *una vez que se crea y se cierra no se puede modificar*

Semántica UNIX

23

- Las escrituras en un archivo son inmediatamente visibles a todos los procesos.
- Un archivo abierto tiene asociado un puntero de posición.
- Alternativas en cuanto al puntero.
 - ▣ Cada proceso mantiene su propio puntero de posición.
 - ▣ Posibilidad de que dos procesos puedan compartir el puntero de posición.
- Implicación:
 - ▣ El sistema operativo debe mantener una imagen única del fichero.
 - ▣ Problemas de contención por acceso exclusivo a la imagen.

Semántica de sesión

24

- Las escrituras sobre un archivo abierto no son visibles por otros procesos con el archivo abierto.
- Cuando se cierra un fichero los cambios son visibles por otros procesos que abran el fichero posteriormente.
- Un fichero puede estar asociado con varias imágenes distintas.
- No hay contención.
- Caso de utilización: AFS (*Andrew File System*).

Semántica inmutable

25

- Un archivo puede ser declarado como compartido.
 - A partir de ese momento no se puede modificar.

- Un archivo inmutable no admite modificación de
 - Nombre.
 - Contenido.

Semántica de versiones

26

- Las actualizaciones se hacen sobre copias con n° versión.
- Sólo son visibles cuando se consolidan versiones.
- Sincronización explícita si se requiere actualización inmediata.

Control de acceso

27

- Listas de control de acceso.
 - ▣ Definen la lista de usuarios que pueden acceder a un fichero.
 - ▣ Si hay diferentes tipos de acceso una lista por tipo de control de acceso.
- Permisos.
 - ▣ Versión condensada.
 - ▣ Tres tipos de acceso (rwx).
 - ▣ Permisos para tres categorías (usuario, grupo, otros).

Contenido

28

- Fichero.
- Atributos y operaciones.
- Vista lógica.
- Semántica de compartición.
- Representación.

`chmod 766 fichero` Cambiar permisos.
 $\begin{matrix} 7 & 6 & 6 \\ \swarrow & \downarrow & \searrow \\ rwx & rwx & rwx \end{matrix}$
 user group otros
 uid gid i -?

`creat(, -)` crea un archivo para write
 7 te puedo leer

Mirar video

Ojo apuntar lo del nodo i,
 que almacena toda la información
 del fichero (nombre, tiempos, owner,
 punteros...)

hardlink → Se crea un enlace al nodo i del
 fichero indicado, si se ha creado nodo borrar
 hasta eliminar los todos
 link simbólico
 Lo tiene su propio nodo → Su contenido es el nombre al que apunta
 pero no aumenta el contador de enlaces y se podría borrar
 unlink borra el fichero, si solo hay un hardlink (no + de 1)
 (decrementa el cont) → cont. enlaces = 0
 Pero si no se ha cerrado no se Sistemas Operativos - Ficheros
 borra en el momento, de estar abierto se puede acceder.

Ojo el dir raíz o raíz vacío
 tiene su propio nodo i.

Todo fich. o dir
 tiene asociado
 un nodo i

Representación del fichero

29

- El sistema operativo debe mantener información sobre el fichero: metadatos.
- Los metadatos son dependientes del sistema de ficheros.
- Importante: Un sistema operativo puede admitir varios sistemas de ficheros.
 - ▣ Ejemplo: en Linux se pueden montar particiones Ext2, NTFS, ...

Asignación de espacio en disco

30

- Gestión de espacio libre y ocupado del disco.
- Asignación de espacio a cada fichero.

- Aspectos:
 - ▣ Ficheros nuevos: ¿Se asigna el espacio máximo en creación?
 - ▣ ¿Qué unidad de asignación se utiliza?
 - ▣ ¿Qué estructura de datos representa la asignación del fichero?

Preasignación versus asignación dinámica

31

- Preasignación: Asignación en creación del tamaño máximo posible del fichero.
 - ▣ Se reserva todo el espacio que podría necesitar el fichero.
- Asignación dinámica: Asignación de espacio según se va necesitando.
 - ▣ División del fichero en unidades de asignación que se van tomando según haga falta.

Tamaño de asignación

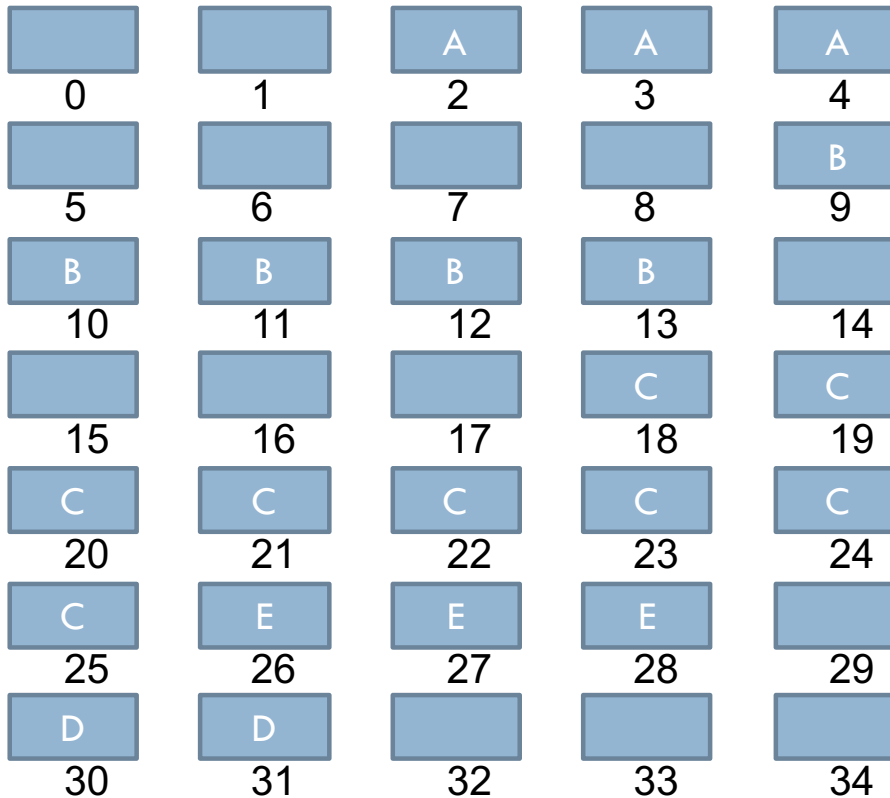
32

□ Cuestiones a considerar:

- ▣ Tamaño de asignación grande → información contigua en disco.
 - Mayor rendimiento.
- ▣ Tamaño de asignación pequeño → aumenta el tamaño de los metadatos.
- ▣ Tamaño de asignación fijo → reasignación de espacio simple.
- ▣ Tamaño de asignación fijo y grande → incrementa el malgasto de espacio (fragmentación interna).
- ▣ Tamaño de asignación variable y grande → incrementa el rendimiento, pero aumenta la fragmentación externa.

Asignación contigua

33



Fichero	Inicio	Long
A	2	3
B	9	5
C	18	8
D	30	2
E	26	3

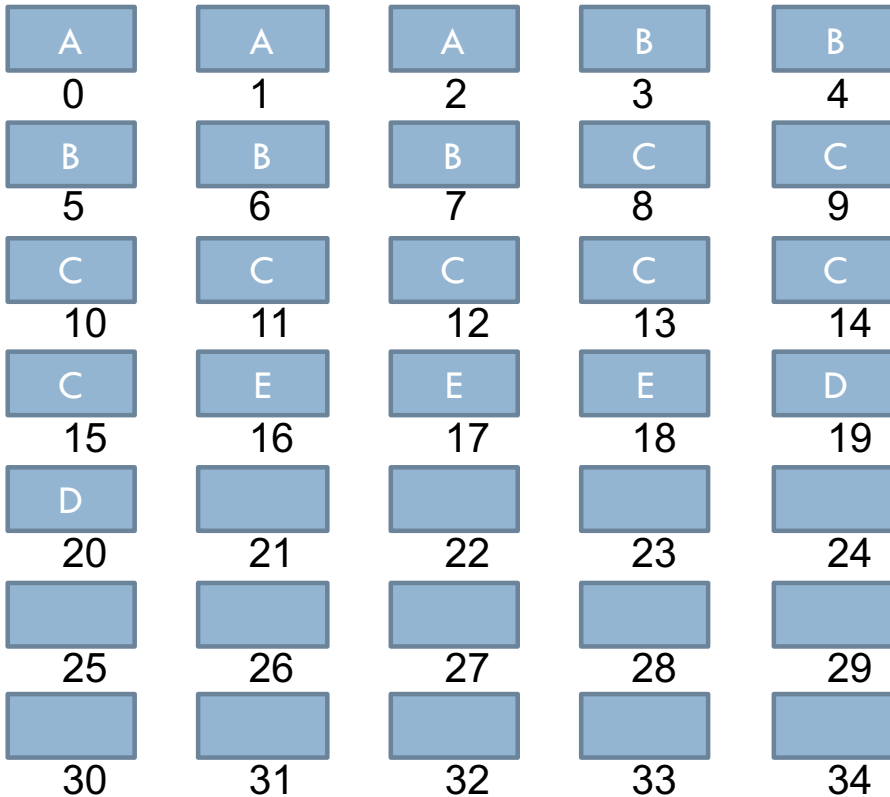
Fragmentación interna: Cuando queda espacio libre en el último bloque de la secuencia

Fragmentación externa: Cuando hay bloques que no podemos usar el no haber suficientes contiguos, pero están vacíos.

Necesidad de compactación

Asignación contigua (compactación)

34



Fichero	Inicio	Long
A	0	3
B	3	5
C	8	8
D	19	2
E	16	3

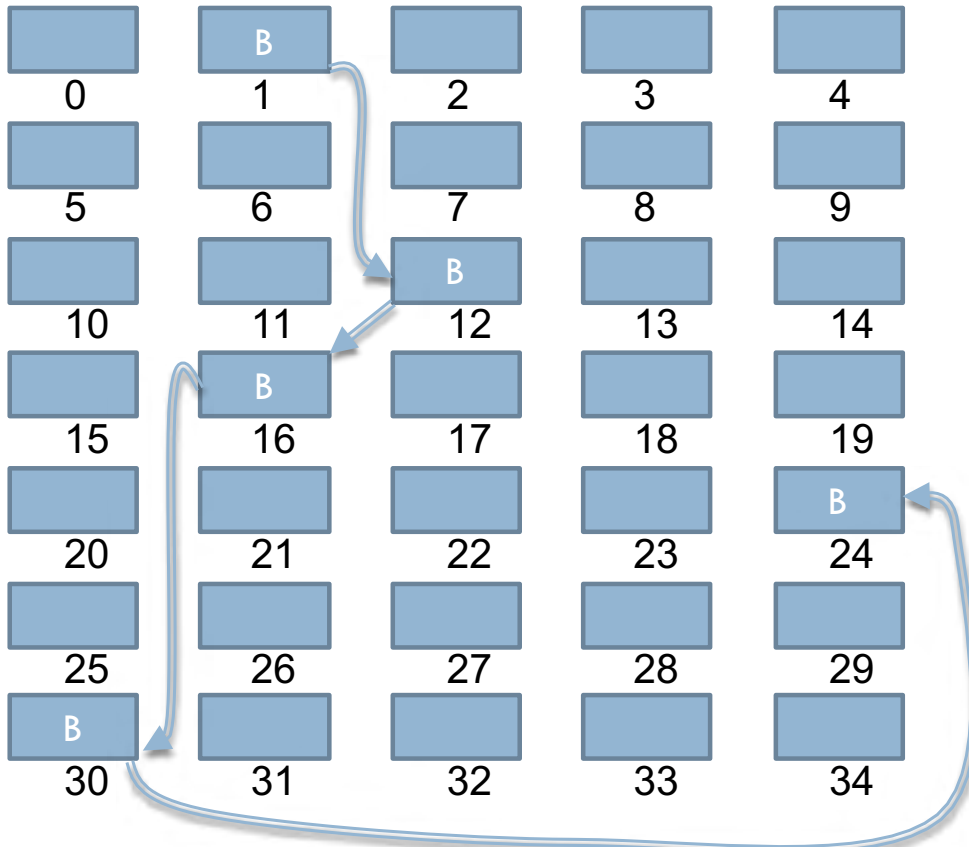
Asignación encadenada

35

- Cada bloque contiene un puntero al bloque siguiente.
- Asignación de bloques de uno en uno.
- No hay fragmentación externa.
- Bloques distribuidos por todo el disco.
- Consolidación del sistema para mejorar las prestaciones de procesamiento de archivos secuenciales.

Asignación encadenada

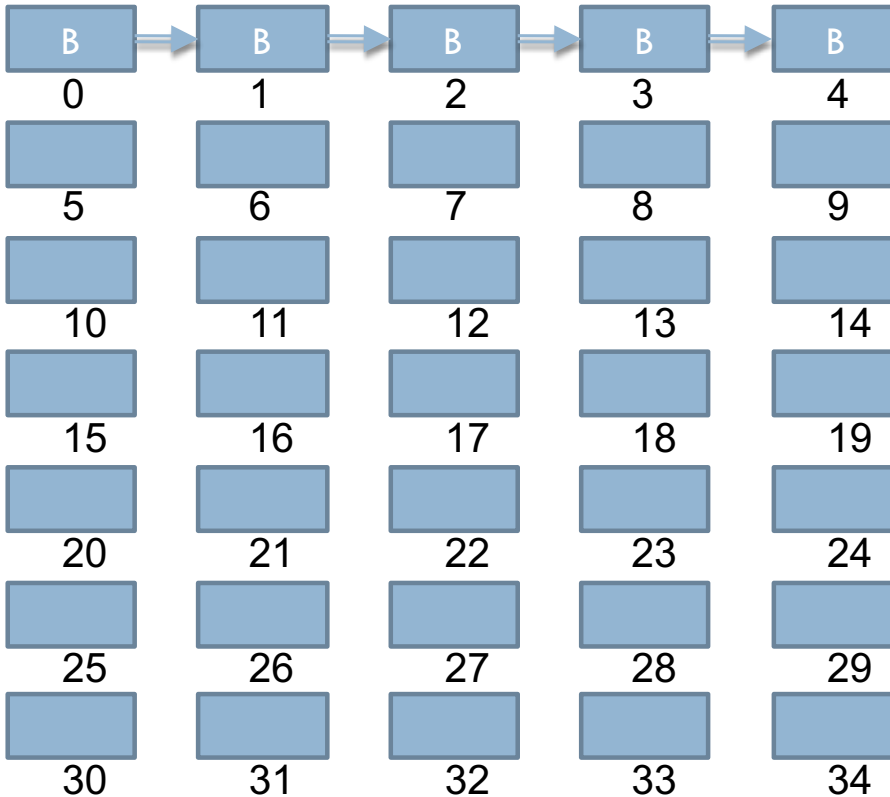
36



Fichero	Inicio	Long
B	1	5

Asignación encadenada (consolidación)

37



Fichero	Inicio	Long
B	0	5

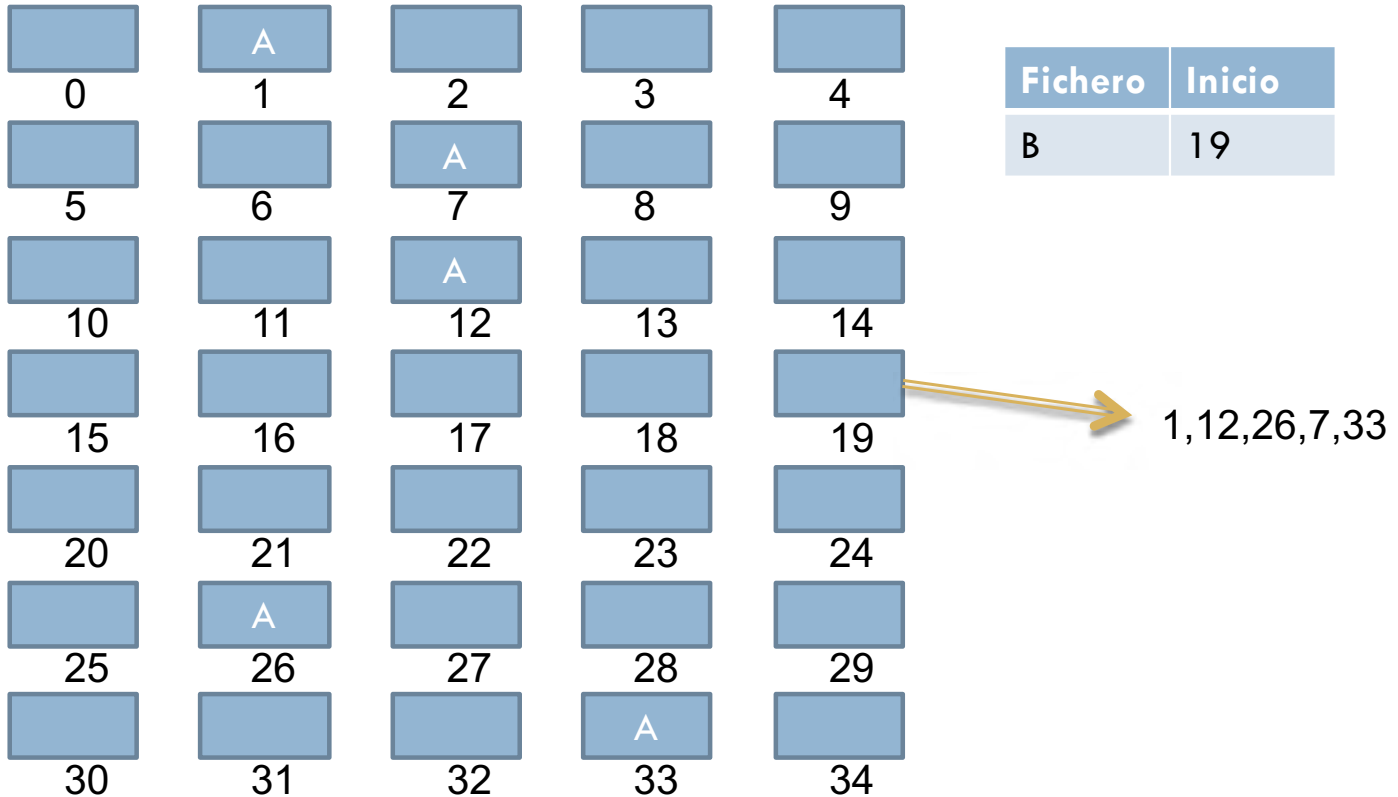
Asignación indexada

38

- Se mantiene una tabla con los identificadores de las unidades de asignación que forman el fichero.
- Alternativas:
 - ▣ Asignación por bloques.
 - ▣ Asignación por porciones (extents).

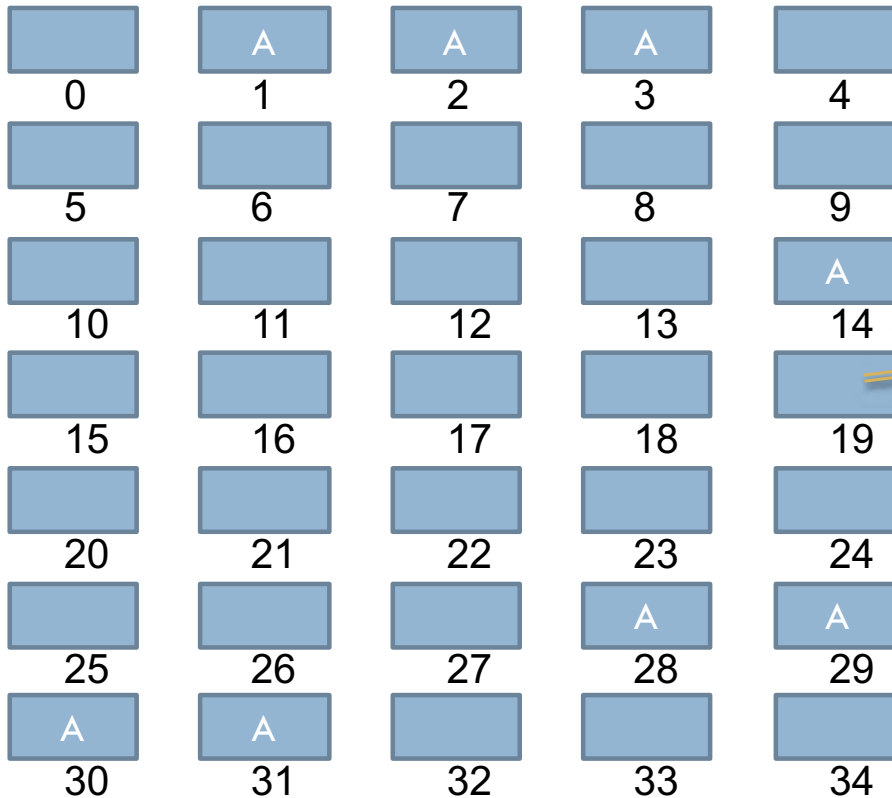
Asignación indexada por bloques

39



Asignación por porciones

40



Fichero	Inicio
B	19

Inicio	Long
1	3
28	4
14	1

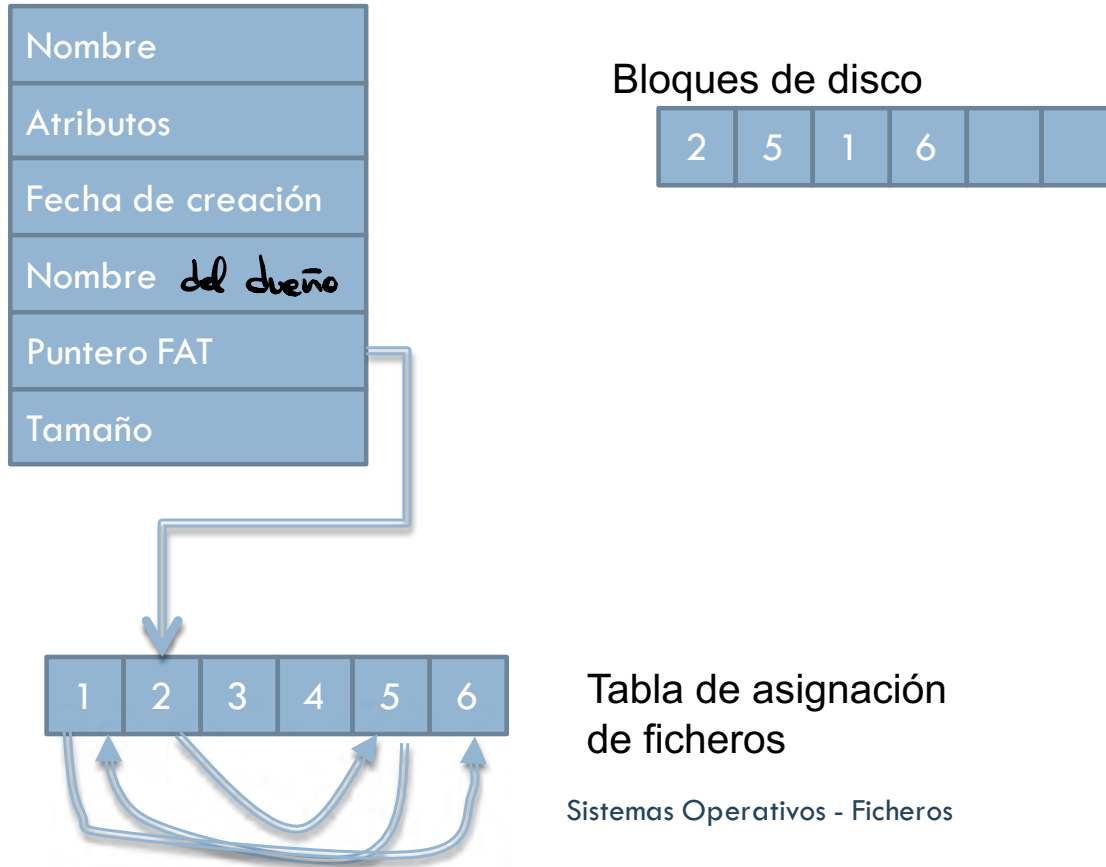
Gestión del espacio de disco

41

- El sistema operativo debe saber que bloques están libres.
- Alternativas:
 - ▣ Mapas de bits: Vector con un bit por bloque.
 - Tabla resumen por rangos de direcciones: número de bloques libres en el rango.
 - ▣ Lista encadenada de porciones libres.
 - ▣ Indexación: Tabla índice de porciones libres.

Representación: FAT ^{File Allocation Table.}

42



Representación: UNIX

43

NO almacena el nombre, ni el puntero de la pos. de lect/escrit.

- Tipo de fichero y protección.
- Usuario propietario del fichero.
- Grupo propietario del fichero.
- Tamaño del fichero.
- Hora y fecha de creación.
- Hora y fecha del último acceso.
- Hora y fecha de la última modificación.
- Número de enlaces.
- Punteros directos a bloques (10).
- Puntero indirecto simple.
- Puntero indirecto doble.
- Puntero indirecto triple.

→ No tiene fragmentación interna.
Info. guardada en nodo i.

tabla fich. intermedios

→ Esta se almacena en la tabla de fich. abiertos y puede ser apuntada por varios, si se copia el puntero.
Pero si no se copia, al abrir de entrada en esta tabla para el mismo fich. uno para cada proceso.

→ Guarda los 10 primeros punteros a bloques, de esta manera puede acceder rápido a esos y se usan los otros para ampliar lo almacenado.

→ 10 punteros

Si bloque de 4 kB y 4 bytes de puntero

10 · 4 kB = 40 kB los 10 punteros

Puntero indirecto simple. $\frac{4 \text{ kB}}{4 \text{ B}} \Rightarrow 1024$ punteros a bloque · 4 kB de bloque = 4 MB

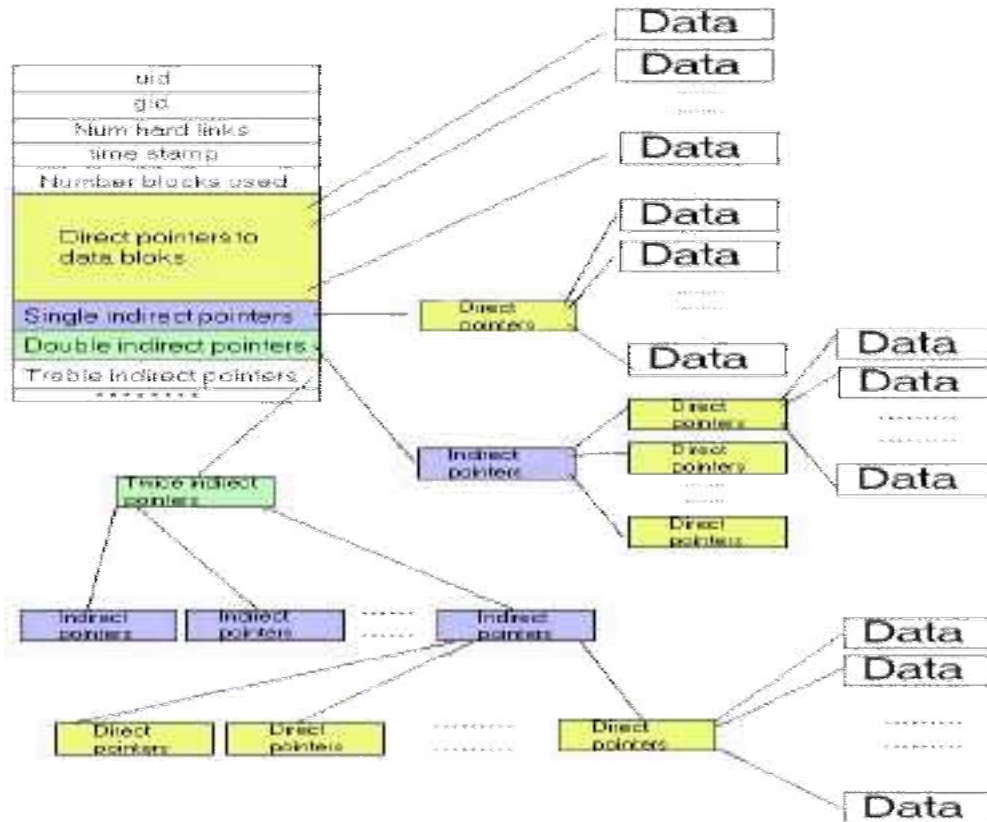
Puntero indirecto doble. $\frac{4 \text{ kB}}{4 \text{ B}} \Rightarrow 1024$ punteros simples · 1024 punteros a bloque simple · 4 kB = 4 GB

$\frac{4 \text{ kB}}{4 \text{ B}} \Rightarrow 1024$ dobles · 1024 simples · 1024 simples · 4 kB = 4 TB

Estructura de punteros

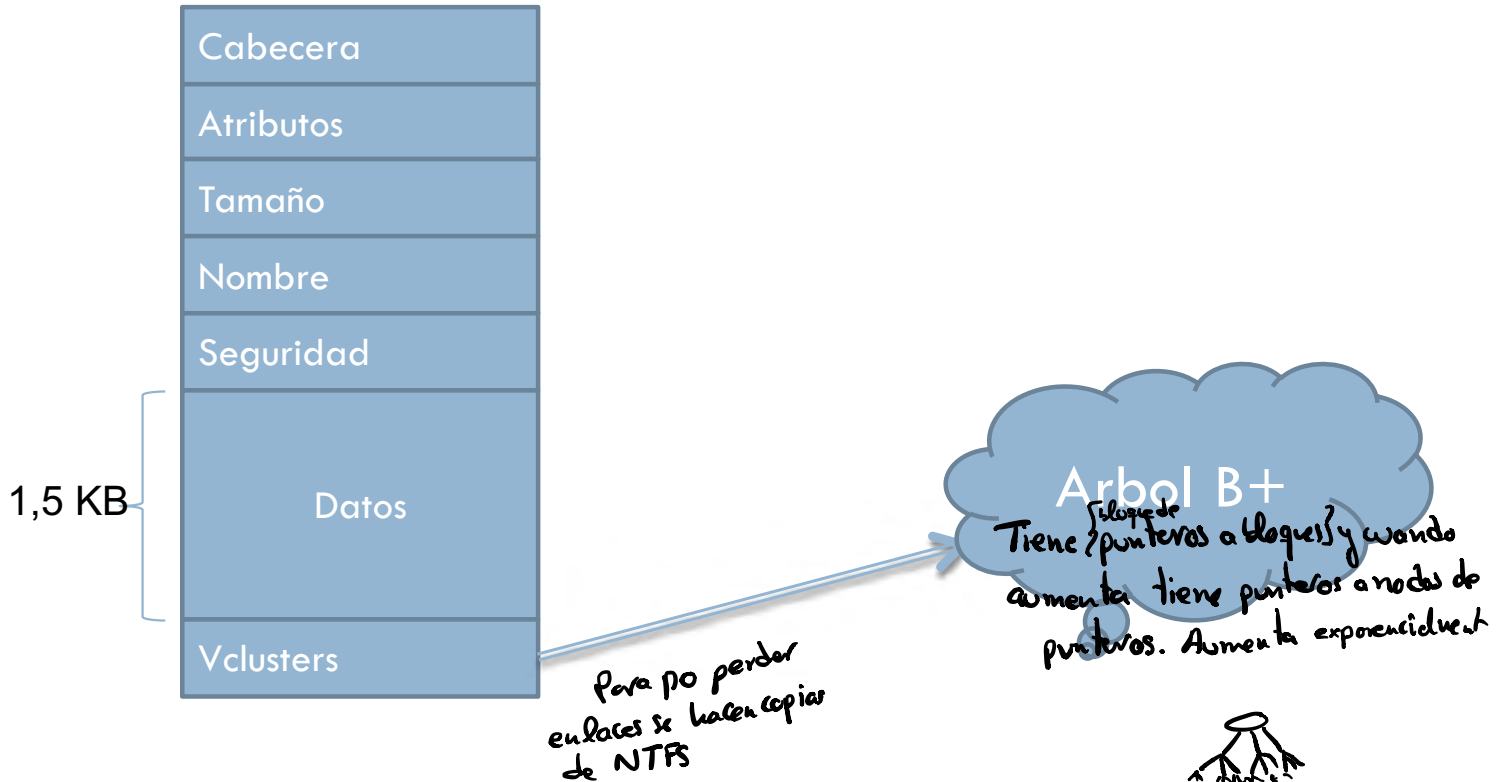
UNIX: Punteros a bloques

44



Representación: NTFS

Node Tree File System



Lecturas recomendadas

46

Básica

- Carretero 2007:
 - ▣ 9.1. Visión de usuario del sistema de ficheros.
 - ▣ 9.2. Ficheros.
 - ▣ 9.5. Ficheros compartidos.
 - ▣ 9.8. Estructura y almacenamiento del fichero.

Complementaria

- Stallings 2005:
 - ▣ 12.1. Descripción básica.
 - ▣ 12.2. Organización y acceso a los ficheros.
 - ▣ 12.4. Compartición de ficheros.
 - ▣ 12.6. Gestión de almacenamiento secundario.
- Silberschatz 2006:
 - ▣ 13. Sistemas de entrada/salida.