

Capítulo 1

Almacenando Información: Discos y Archivos

Tema 1 . *Almacenamiento de Datos* . 2008/2009

Irene Martínez
Depto. Lenguajes y Computación. UAL.

1.1

Objetivos

Motivación

- Objetivo: Almacenar y acceder a colecciones de datos muy extensas
- Discos tienen gran capacidad pero son muy lentos

Objetivos

1. Conocer dispositivos de almacenamiento de grandes cantidades de datos
2. Jerarquías de almacenamiento entre estos dispositivos
3. Modo de funcionamiento de los discos. Entrada/salida
4. Costo de las operaciones de acceso a disco. Mejora de la eficiencia

1.2

Contents

1	Jerarquía de Memoria	1
2	Discos	3
2.1	Características Físicas de los Discos	3
2.2	Acceso a Disco	4
3	Eficiencia del Almacenamiento Secundario	5
3.1	Buffers de E/S	5
3.2	Otras Soluciones	5
4	Conclusiones	6

1.3

1 Jerarquía de Memoria

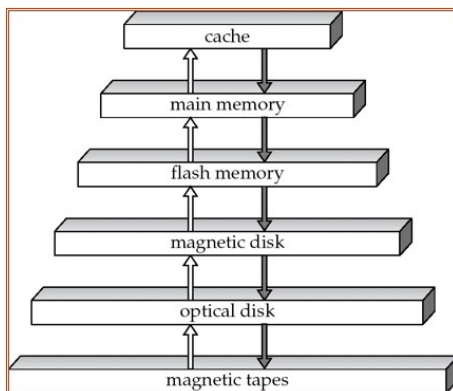
Clasificación de los Dispositivos de Almacenamiento

	Tiempo medio de acceso típico
--	-------------------------------

- Almacenamiento primario (ns).
 - Registros de la CPU
 - Memoria caché: de primer, segundo y tercer nivel
 - RAM
 - Discos RAM
 - Caché de disco
- Almacenamiento secundario
 - Discos magnéticos (ms).
 - Cintas (s).
- Almacenamiento *offline*.
 - Discos magnéticos removibles (ms).
 - Discos ópticos (CD-ROM, DVD-ROM) (ms).
 - Cintas (s).

1.4

Almacenamiento Secundario



Copyright by Silberschatz, Korth, Stallman, 2005

↑Mayor *Velocidad*

Mayor *Capacidad*↓

- Almacenamiento secundario es muy lento en comparación con el primario
- Capacidad "ilimitada"
- La información permanece una vez desconectados

1.5

Almacenamiento Secundario Ejemplos

Tiempos de acceso

- Tiempo medio de acceso de RAM: 7,5 ns ($7,5 \times 10^{-9}$ sg)
- Discos SCSI más rápidos superan los 5 ms (5×10^{-3} sg)
- diferencia de acceso de orden 1:500.000 (mínimo)

Esto significa que un proceso que ordena una lista:

- Si cabe completa en *memoria RAM*: si tarda 1 segundo

- Si se utilizara el *disco* para la ordenación: tardaría 500.000 sg, es decir, 139 horas (casi seis días)

Coste económico

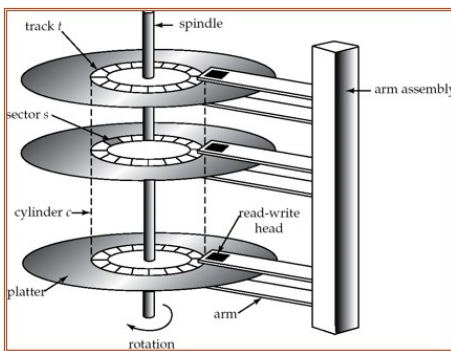
- Coste de 1GB de RAM es aproximadamente 300 euros
 - Coste Disco SCSI de 70GB , igual
 - Esto significa que el coste económico por Mbyte sería:
 - 29.3 céntimos de euro para la *memoria RAM*
 - 0.4 céntimos de euro en *disco*
- ➔ proporción del orden 1:73

1.6

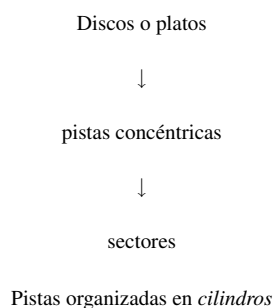
2 Discos

2.1 Características Físicas de los Discos

Organización



Copyright by Silberschatz, Korth, Sudarshan, 2005



Sector

Es la unidad de lectura/escritura en disco

Cilindro

Las pistas de un cilindro se pueden leer sin mover el **cabezal**, que es lo que consume más tiempo

Capacidad del disco

- Capacidad Pista = n° sectores por pista \times bytes por sector
- Capacidad Cilindro = n° pistas por cilindro \times capac-Pista
- Capacidad Disco = n° cilindros \times capac-Cilindro

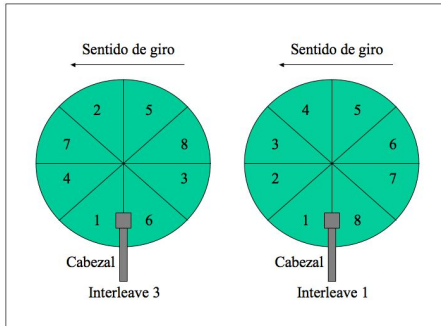
1.7

Organización de Pistas por Sectores

A *nivel lógico* un fichero puede considerarse formado por una sucesión de sectores adyacentes dentro de una misma pista. Sin embargo, a *nivel físico* no es así.

- **Disposición Física de los Sectores**

- En general, no es posible leer sectores adyacentes de una misma pista → El controlador debe procesar la información leída antes de aceptar la siguiente



Interleaving

situar varios sectores físicos entre dos sectores lógicos adyacentes

1.8

Organización de Pistas por Sectores

- **Clusters: Disposición lógica del S.O.**

- Agrupa los sectores en *clusters* → un fichero es una sucesión de clusters

Cluster

Es un número fijo de sectores consecutivos (lógicos). Todos los sectores de un cluster se leen con un sólo posicionamiento

- FAT: contiene la lista enlazada de todos los clusters de un fichero
- Entrada FAT: para cada cluster, su posición física
- En muchos S.O. puede decidirse cuántos sectores asignar a cada cluster
- Para minimizar el desplazamiento del brazo del disco, si hay suficiente espacio, el fichero podría almacenarse en una extensión:

Extensión

Un fichero que puede almacenarse completo en *clusters* contiguos del disco se dice que ocupa una sola **extensión**: todos sus sectores, pistas, incluso cilindros, forman un todo contiguo.

1.9

Organización de Pistas por Bloques

Las pistas de los discos no están divididas en sectores, sino en números enteros de bloques que el usuario define y cuyos tamaños pueden variar

- Los bloques pueden tener longitud fija o variable
- La cantidad de información transmitida en un acceso depende del tamaño de los bloques
- El tamaño puede elegirse de forma que almacene un número entero de datos lógicos

Factor de Bloqueo

Número de registros que se almacenan en cada bloque

- Información adicional para gestión de bloques:
 - cada *bloque de datos* se acompaña de uno o más **sub-bloques con información complementaria**:
 - *count subblock* : nºbytes del bloque
 - *key subblock* : clave del último registro del bloque
 - *data subblock* : datos

1.10

2.2 Acceso a Disco

Factores que Intervienen en el Acceso a Disco

Tiempo de una Operación de Acceso a Disco

||

Tiempo de Posicionamiento (Seek Time)

Tiempo que tarda el brazo del cabezal en posicionarse en el **cilindro** correspondiente

+

Retardo de Rotación (Rotational Delay)

Tiempo que tarda el **sector** buscado en colocarse debajo del cabezal de R/W

+

Tiempo de Transferencia (Transfer Time)

Viene dado por la siguiente fórmula:

$$\text{TiempoTransferencia} = \text{Tiempo Rotacion} * \frac{\text{num bytes transferidos}}{\text{num bytes por pista}}$$

1.11

Coste de una operación de Entrada/Salida

$$\boxed{\text{Tiempo Acceso Disco}} =$$

$$\boxed{\text{Tiempo Posicion.} + \text{Tiempo Rotación} + \text{Tiempo Transferencia}}$$

- Los tiempos de *posicionamiento* y de *rotación* son los más costosos
- Por ello es más eficiente transferir bloques consecutivos de la misma pista o cilindro:
 - Eliminamos los dos primeros tiempos en todos los bloques excepto en el primero
 - **Ejemplo:** Leemos dos bloques B1 y B2 consecutivos
 - Tiempo B1 = T.Posicion. + T.Rotación + T.Transferencia
 - Tiempo B2 = T.Transferencia

1.12

3 Eficiencia del Almacenamiento Secundario

3.1 Buffers de E/S

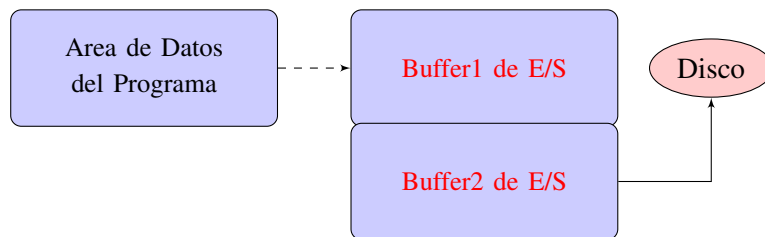
Uso de Buffers para Mejorar la Eficiencia

- Un buffer se utiliza para trabajar con grandes cantidades de datos en memoria RAM de modo que se *reduzcan el número de accesos a disco*
- Los sistemas de E/S suelen usar dos bufferes: uno de E y otro de S

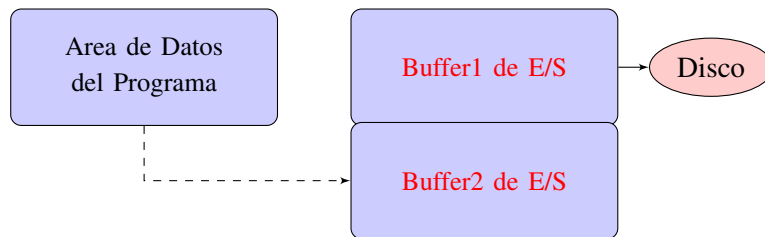
- Otra solución es utilizar varios buffers que puedan usarse como E o S:

Estrategia de Doble Buffer

Mientras uno de los buffers se llena con los datos de la aplicación, el otro vuelca su contenido al disco, alternat.



Ahora, es el Buffer2 el que recibe los datos mientras que el Buffer1 se escribe en disco:



1.13

3.2 Otras Soluciones

El Disco como Cuello de Botella

- **Técnicas para transmitir datos a mayor velocidad de la que permiten los discos:**

Stripping

- Sistemas de almacenamiento con múltiples discos en los que la información se almacena a trozos entre todos
- Se suma el ancho de banda generado por cada uno de ellos
- Se reduce proporcionalmente el tiempo de transferencia

1.14

El Disco como Cuello de Botella

- **Técnicas de Buffering:**

RAM disk

- Utilizar parte de la RAM como unidad disco
- Simula un disco en memoria: eliminamos el retardo de rotación y el de posicionamiento se reduce enormemente
- El mayor problema: *volatilidad*

Caché de Disco

- Utilizar parte de la RAM como caché de disco.
- Contiene páginas de datos del disco: el sistema busca primero en la caché para evitar acceder a disco
- Alto rendimiento con datos ubicados contiguamente

1.15

4 Conclusiones

Conclusiones

1. Aplicaciones trabajan con grandes volúmenes de datos
2. Necesitamos dispositivos con gran capacidad de almacenamiento
3. Problema: acceso a disco es muy costoso
4. Buscamos **minimizar el número de accesos**
5. Técnicas:
 - Bloques de datos, buffering, etc..
 - *Estructuras de datos adecuadas* para almacenar y recuperar la información

1.16

References

- [1] M.J. Folk y B. Zoellick *Estructuras de Archivos*. Addison-Wesley, 1992.
- [2] Silberschatz, Korth, Sudarshan *Database System Concepts*. McGraw-Hill, 2005.
<http://www.db-book.com/>
- [3] Garcia-Molina, Ullman, Widom *Database System Implementation*. Prentice-Hall, 2000. <http://infolab.stanford.edu/~ullman/dbsi.html>
- [4] William Stallings *Operating Systems: Internals and Design Principles, 5/E*. Prentice-Hall, 2004. <http://williamstallings.com/OS/OS5e.html>

1.17