Sistemas de almacenamiento en Servidores de Información multimedia

Servidores de Información Multimedia
2º Ingeniero Técnico de Telecomunicación – Imagen y Sonido

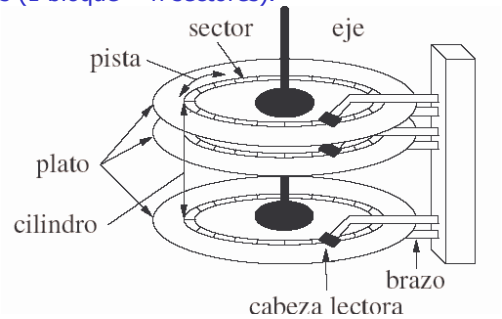
Departamento de Ingeniería Telemática
Universidad Carlos III de Madrid

2 Indice

- Estructura de un Disco Duro
- Planificación de operaciones
- Tareas adicionales de mantenimiento
- Almacenamiento en discos múltiples
- Configuraciones RAID
- Conexión de discos al sistema
- Almacenamiento Terciario

3 Estructura de un Disco Duro

- Un disco duro se estructura a nivel lógico como **una tabla de bloques lógicos**.
- El tamaño de este bloque suele ser de **512 bytes**, aunque hay discos que permiten cambiar este parámetro (por ejemplo a 1024 bytes).
- Internamente, el disco no tiene esta estructura linear. La estructura se divide en: cilindros, pistas y sectores (1 bloque = n sectores).



- El sector lógico 0 corresponde con el primer sector de la primera pista del cilindro más externo.
- El mapeo se incrementa primero por número de sector, luego por pista y finalmente por cilindro.

4 Organización Interna de Sectores

- Si todas las pistas tuviesen los mismos sectores, **el mapeo de sectores lógicos a físicos es trivial**.
- En la realidad este mapeo es mucho más complejo porque **el número de sectores por pista no tiene por qué ser constante**.
  - En discos que mantienen **velocidad lineal constante** la densidad de bits por pista cambia dependiendo de la longitud de la pista.
    - La velocidad de rotación se reduce conforme se lee una pista **más exterior**. Formato utilizado por CDs y DVDs.
  - Los discos duros suelen giran manteniendo **velocidad angular constante**, con lo que el número de sectores se mantiene constante por pista.

5 Planificación de Operaciones en Disco

- El tiempo de respuesta de un disco duro al recibir una petición consta de **dos componentes**:
  - **Tiempo de búsqueda** : Es el tiempo que tarda el brazo en situar la cabeza lectora sobre el cilindro deseado.
  - **Latencia rotacional** : Tiempo que tarda el sector requerido en alcanzar la cabeza lectora.
- Se define como **el ancho de banda** de un disco duro como el número total de bytes transferidos dividido por el tiempo transcurrido entre la primera petición de servicio y la transferencia de la última petición.
- ¿Es posible que un disco duro pueda mejorar su ancho de banda?

6 Parámetros de una Operación de Disco

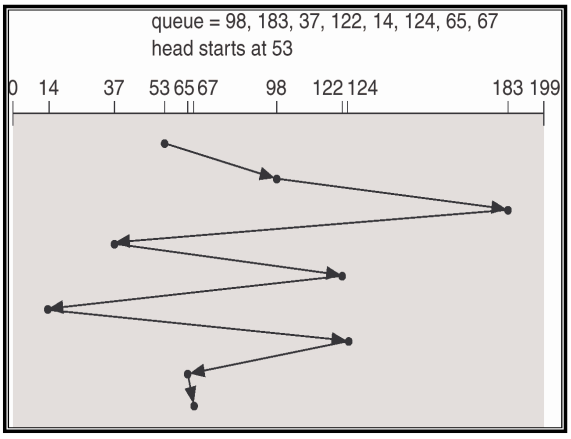
- Las operaciones sobre disco reciben siempre los siguientes parámetros:
  - Si la operación es de lectura o escritura.
  - La dirección de memoria para la transferencia de datos.
  - La dirección de la porción de disco a transferir.
  - Número de bytes.
- El sistema almacena internamente un cierto número de operaciones con sus parámetros en una **cola**.
- Es en la **gestión de esta cola** en donde es posible obtener un mayor rendimiento del dispositivo.
- A esta cola se aplican varias técnicas:
  - Orden de servicio de peticiones (el parámetro que se trata de minimizar es el **tiempo de búsqueda**).
  - Lectura en paralelo de datos de varias cabezas lectoras
  - Lectura con adelanto (al solicitar un bloque se traen también los siguientes)
  - Almacenamiento contiguo de ficheros en disco
- Veamos a continuación algunos algoritmos para mejorar el **tiempo de búsqueda**.
  - Estos algoritmos los pueden aplicar tanto el SO como el controlador (procesador integrado con el HW del disco).

7 Planificación de Operaciones FCFS

- Es la forma **más simple** de planificar internamente las operaciones sobre el dispositivo.
- Se accede a los sectores **en el orden en el que se han recibido**.
- Supongamos una secuencia de peticiones de sectores en los siguientes cilindros:
  - 98, 183, 37, 122, 14, 124, 65, 67
- Si la cabeza lectora está sobre el cilindro 53, se puede comprobar que para servir estas peticiones es preciso moverse a través de **640** cilindros.
- El número de cilindros atravesado **influye directamente** sobre el tiempo de búsqueda, y por tanto sobre el **ancho de banda**.
- **Conclusión:** Es preciso un algoritmo más inteligente que permita reducir el movimiento del brazo lector.

8 Planificación de Operaciones FCFS

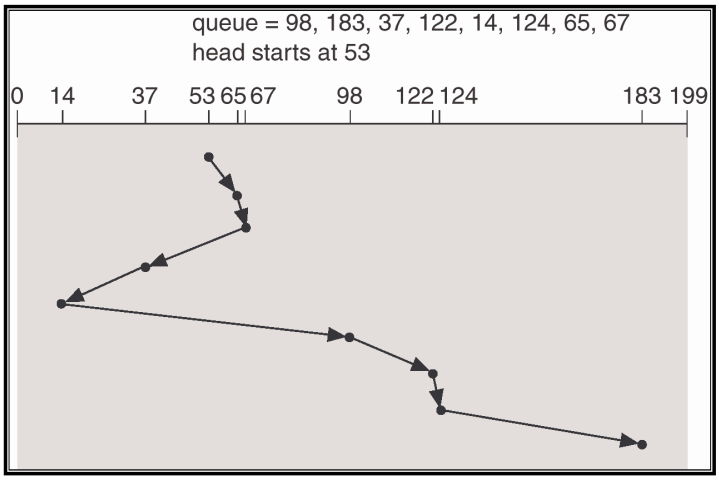
- Pros
  - Las aplicaciones obtienen acceso ordenado a disco
  - "Justo" para todas las peticiones
- Cons
  - Búsquedas largas.



9 Planificación por Menor Tiempo de Búsqueda

- Siguiendo una estrategia similar a la aplicada en planificación de procesos, si se quiere minimizar el número de desplazamientos a través de cilindros, se debe servir primero **la petición con cilindro más cercano**.
- Esta estrategia (SSTF, Shortest Seek Time First) consigue una **reducción drástica** de movimientos del brazo.
- Para la secuencia considerada anteriormente (98, 183, 37, 122, 14, 124, 65, 67), con la cabeza en posición 53, se realizan 236 movimientos.
- Al igual que su análogo en el contexto de planificación de procesos, este algoritmo padece el problema de **inanición**.
- Este algoritmo, a pesar de mejorar FCFS, **no es óptimo**.

10 Planificación por Menor Tiempo de Búsqueda



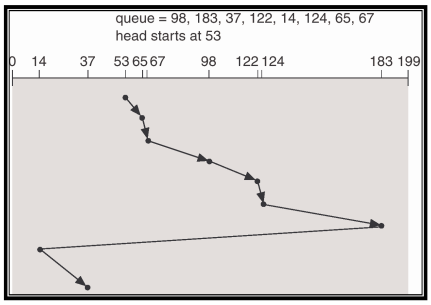
11 Planificación por Barrido

- En esta estrategia, el brazo lector comienza sobre el primer cilindro y mantiene el acceso a cilindros superiores hasta que no haya más operaciones.
- En cuanto no quedan más cilindros en la dirección de avance, cambia de dirección y sirve las peticiones con cilindros **menores** al actual.
- La cabeza, por tanto, realiza **un barrido** de los cilindros en ambos sentidos.
- A este algoritmo se le conoce también con el nombre de **algoritmo del ascensor**.
- La desventaja de este algoritmo es que los cilindros de las esquinas tienen una **frecuencia de visita** muy descompensada.
- Por ejemplo: el cilindro 1 se visita dos veces con un intervalo de tiempo **muy reducido**. Es muy probable, por tanto, que no haya peticiones la segunda vez que la cabeza lo visite.
- Para la secuencia considerada anteriormente (98, 183, 37, 122, 14, 124, 65, 67), con la cabeza en posición 53, se realizan 208 movimientos

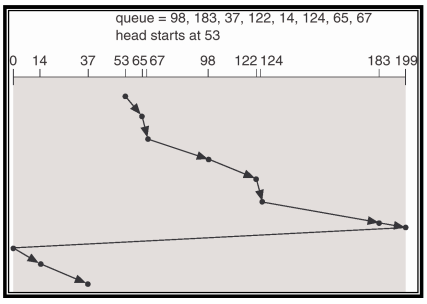
12 Planificación por Barrido Circular

- Es una variante del algoritmo de barrido pero que trata de uniformizar **la frecuencia de acceso a los cilindros**.
- Los cilindros de un disco se tratan como si estuviesen en una **lista circular**.
- Una vez que la cabeza accede al último cilindro, en lugar de cambiar de dirección se posiciona de nuevo sobre el primer cilindro.
- De esta forma, el tiempo entre accesos a un cilindro es **idéntico** para todos ellos.
- En una optimización, la cabeza lectora **no tiene por qué alcanzar los extremos del disco a no ser que haya una petición para esos cilindros**.
- Los algoritmos que implementan esta característica se les denomina **LOOK** y **C-LOOK** correspondiendo con los algoritmos de barrido y barrido circular respectivamente (que se denominan **SCAN** y **C-SCAN**).

13 Planificación por Barrido Circular



C-LOOK



C-SCAN

14 Factores Adicionales que Afectan al Rendimiento

- A pesar de que la **reorganización de las peticiones** reducen el tiempo de acceso, los discos duros actuales **no permiten saber a priori dónde están almacenados los sectores.**
- El controlador interno de un disco **puede reposicionar sectores defectuosos**.
- La tendencia actual es a incluir los algoritmos de minimización de movimiento como **parte del controlador**.
- En este caso, el sistema operativo puede reducir el tiempo de acceso **seleccionando los sectores de un mismo fichero con índices cercanos**.
- Además, como todo fichero se accede a través de su directorio, los sectores de los directorios y los de los ficheros que albergan **deben estar próximos**.
- Existen **programas compactadores** que reorganizan el contenido de un disco.

15 Tareas de Mantenimiento de Discos

- Además de la realización de operaciones de entrada/salida sobre discos, el sistema operativo debe realizar otras tareas de mantenimiento:
  - **Inicialización** : El sistema debe ser capaz de inicializar un disco.
  - **Disco de Arranque** : De los discos del sistema, uno de ellos debe estar preparado para albergar los programas de arranque.
  - **Recuperación de Errores** : Si se detectan sectores erróneos, el sistema debe poder recuperar la información.

16 Inicialización de Discos

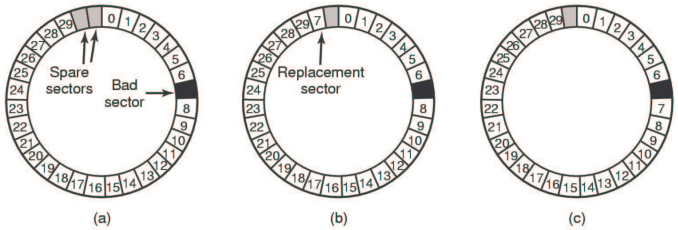
- Antes de que un disco pueda ser utilizado su superficie debe ser **dividida en sectores**.
- Esta división se denomina **formateo de bajo nivel** y se suele ser realizado **por el fabricante**.
- Algunos discos duros **poseen un controlador capaz de realizar un formateo de bajo nivel**, incluso modificando los parámetros tales como el tamaño del sector.
- Para que el sistema pueda utilizar un disco, debe **crear un conjunto de estructuras de datos** sobre sus sectores.
- Esta inicialización se estructura en **dos pasos**.
- En el primero se crean **particiones**. Una partición es **un conjunto de cilindros** y se trata como si fuese un disco independiente.
- La manipulación de particiones **implica un conjunto de cambios significativos** en la configuración del sistema operativo.
- En el segundo paso se crean las estructuras de datos necesarias para **la gestión de sectores**. A este paso se le denomina **formateo lógico**.
- Algunas aplicaciones requieren la utilización de espacio de disco sin formato lógico. Son los denominados discos "raw".

17 Disco de Arranque

- Cuando un equipo arranca se precisa un proceso que acabe **carguando el sistema operativo** en memoria y lo arranque.
- El proceso de arranque, tras una comprobación del HW, ejecuta el programa inicial que se almacena en la memoria ROM disponible en el equipo.
- El programa de la memoria ROM es **capaz de leer una posición fija del disco que se seleccione como disco de arranque**.
- El disco que contiene **el programa de arranque en un sector específico** se denomina **disco de arranque**.
- El programa almacenado en el sector de arranque es capaz de cargar el sistema operativo almacenado **en una posición arbitraria del disco**.

18 Gestión de Sectores Defectuosos

- Los discos contienen múltiples partes móviles, con lo que ocasionalmente **un sector se convierte en defectuoso**.
- La gestión de estos sectores depende **del propio dispositivo** y del **sistema operativo**.
- La forma más básica de gestión es mediante la invocación de **programas específicos para ignorar estos sectores**.
- Los dispositivos SCSI reemplazan **automáticamente** los sectores defectuosos con sectores adicionales del dispositivo.
- La información del sector defectuoso se recupera a través de un **código corrector de errores** que incluye cada sector.
- Los discos incluyen **sectores de reserva** en cada cilindro, así como algún **cilindro de reserva.**



19 Gestión de Espacio de Swap

- El espacio de swap es utilizado por los sistemas de forma diferente: para almacenar imágenes completas de procesos en ejecución, o para almacenar páginas de diferentes procesos.
- Debido a la particularidad de los datos almacenados la gestión de este espacio **requiere algoritmos especiales**.
- La velocidad de acceso a los datos almacenados en el área de swap tiene un impacto **directo** sobre el rendimiento del sistema de memoria virtual.
- Algunos sistemas operativos implementan el área de swap como **un fichero más**. La ventaja es que el código de gestión es más simple, pero la eficiencia puede ser mejorada sensiblemente.
- La mayor eficiencia se obtiene almacenando el área de swap **en su propia partición**.
- En este caso, el sistema utiliza **rutinas especiales** para su gestión que tratan de obtener la mayor velocidad de acceso para los datos.

20 Almacenamiento de Datos en Múltiples Discos

- El coste de almacenamiento de información en discos duros ha **caído sensiblemente en los últimos años**.
- Esto ha permitido que sea asequible **disponer de varios discos en un mismo sistema**.
- **Ventaja:** El tener varios discos abre la puerta a técnicas de almacenamiento redundante.
- Se pueden organizar los discos para **incrementar su resistencia a fallos**.
- El parámetro a considerar es el **tiempo medio entre fallos** (MTF).
- Si múltiples discos se utilizan sin estructura alguna, el sistema **se degrada**.
  - **Ejemplo:** El MTF de un disco es de 100.000 horas. Se dispone de un equipo con 100 discos. El MTF del sistema entero es de 1000 horas, es decir, poco más de 41 días.
- Haciendo trabajar de forma conjunta varios discos podemos obtener mejoras en:
  - Fiabilidad (mediante técnicas de redundancia)
  - Rendimiento (mediante paralelismo)

21 Mejora de la Fiabilidad a través de Redundancia

- La técnica utilizada para **incrementar la fiabilidad** de un sistema de almacenamiento masivo es mediante la **replicación de contenido**.
- La técnica más simple se denomina **espejo** (mirroring) y consiste simplemente en ejecutar las operaciones de E/S en **dos dispositivos** idénticos.
- Los dos dispositivos están ejecutando en paralelo, procesan idénticas operaciones y, por tanto, contienen **exactamente el mismo contenido**.
- Cuando un disco sufre una avería, simplemente se utiliza el **disco auxiliar** mientras el primero no se repara.
- Si al tiempo medio que se tarda en reparar un disco le denominamos MTR y es de 10 horas, se asume que los fallos en los discos son **independientes** y que el MTF de cada disco es 100.000 (11,4 años) horas, el tiempo medio entre fallos que conlleven pérdida de datos es de 57.000 años.
- **Problema:** Los fallos en los discos **distan de ser independientes**.
- La mayoría están asociados a pérdidas de tensión, o dependen de la edad del disco (los espejos suelen ser discos "gemelos").

22 Mejora del Rendimiento vía Paralelismo

- El disponer de múltiples discos no sólo permite aumentar la fiabilidad, sino **el rendimiento**.
- El tiempo de acceso de un disco a un dato es n milisegundos.
- Si se dispone de m discos, es posible acceder a m datos en n milisegundos (incremento del número de operaciones por unidad de tiempo).
- Con discos múltiples también se puede **mejorar la velocidad de transferencia de datos**.
- La técnica más simple es mediante lo que se conoce como **distribución a nivel de bits** o "bit-level striping".
- Cada bit de un byte se almacena **en un disco diferente**.
- **Ejemplo:** Si se dispone de 8 discos, se pueden considerar como un disco con sectores de tamaño 8 veces superior al normal.
- La operación de lectura tarda lo mismo que antes, pero se obtienen **ocho veces más datos**.

23 Paralelismo a nivel de Bloque

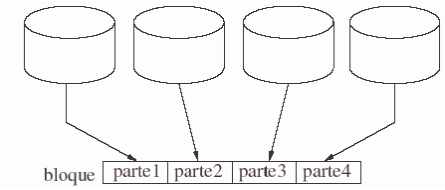
- La misma técnica que se aplica a nivel de bit **se puede aplicar a nivel de bloque** (un bloque es un conjunto de sectores – es la unidad mínima de transferencia que usa el SO).
- Los bloques de un fichero se distribuyen entre los diferentes discos.
- La lectura de un bloque se envía a todos los dispositivos en paralelo y se obtiene **toda la información** en el tiempo que tarda un único acceso.
- Los dos objetivos del paralelismo inherente en discos múltiples son:
  - 1. Incrementar el número de pequeñas operaciones servidas por unidad de tiempo.
  - 2. Reducir el tiempo de respuesta para transacciones largas.

24 Sistemas RAID

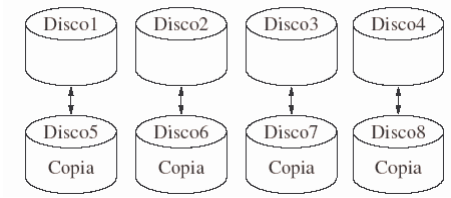
- **Técnicas de Espejo:** Mejoran la fiabilidad, pero no proveen mejora de rendimiento.
- **Técnicas de Distribución:** Mejoran las tasas de transferencia pero no ofrecen mejora de fiabilidad.
- ¿Es posible obtener **ambas ventajas a la vez**?
- Existen un conjunto de técnicas que combinan **distribución** y **redundancia** con **bits de paridad** para obtener estas ventajas.
- Todas ellas se engloban en una categoría denominada **Niveles RAID**
- **RAID:** Tabla Redundante de Discos Independientes (comenzaron siendo "baratos").

25 Configuraciones RAID 0 y RAID 1

- Son las configuraciones más sencillas.
- **RAID 0**: Se dispone de múltiples discos y se aplica la técnica de distribución a nivel de **bloques de ficheros**.
- No se contempla ningún tipo de **redundancia**.

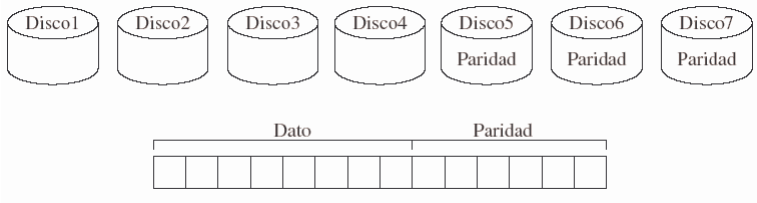


- **RAID 1**: Se aplica **únicamente** la técnica de espejo.



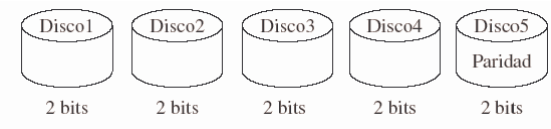
26 Configuración RAID 2

- También conocida como **organización basada en código de corrección de errores**.
- Sistema basado en el almacenamiento de **bits de paridad**.
- Con un **único bit** se detectan todos los errores de un sólo bit.
- Códigos más extensos permiten **recuperar** la información correcta a partir de la información almacenada con errores.
- **Ejemplo:** Cada byte tiene asignado un código de corrección de errores de 6 bits (14 bits en total).
- Se utilizan siete discos cada uno **almacena 2 bits**.



27 Configuración RAID 3

- Se almacena la información distribuida a nivel de bit.
- La información de paridad **se reduce a un único bit**.
- La observación clave es que **los discos son capaces de detectar cuando una operación ha fallado**.
- Si se leen varios sectores, y uno de ellos ha fallado, se sabe inmediatamente **la porción del dato incorrecta**, con lo que se precisa **menos información para recuperar su contenido**.
- La información sobre la paridad se almacena **en un único disco**.



- Se almacena la información sobre paridad necesaria para recuperar la información dañada en un disco y **correcta en el resto**.

28 Configuración RAID 4

- Aplica la técnica anterior pero en lugar de **distribución por bits** utiliza **distribución por bloques**.
- Al igual que en RAID 3, **un único dispositivo** almacena la información para detectar y corregir los datos resultantes de operaciones erróneas.
- La lectura de **un bloque** se asigna a un disco, permitiendo al resto de discos **acceder a diferentes bloques**
- Las operaciones de escritura son **lentas** debido a que se requiere modificar el bloque de paridad.
- Una operación de escritura necesita **cuatro operaciones en disco**: dos de lectura y dos de escritura.

29 Configuración RAID 5

- En lugar de almacenar el bloque de paridad en un único disco, **se almacena de forma distribuida con los discos de datos**.
- Para un determinado bloque, **un disco almacena la paridad y el resto los datos**. Para otro bloque, la paridad está almacenada en un disco diferente.
- **Ejemplo:** Con 5 discos, la paridad del bloque n se almacena en el disco $n_{mod}5) + 1$. Los otros cuatro discos almacenan la información de este bloque.
- Un bloque de paridad **no puede almacenar paridad para la información almacenada en el mismo disco**.
- Con esta distribución se consigue **distribuir la carga de operaciones** de forma homogénea entre todos los discos.

30 Configuración RAID 6

- Variación sobre la configuración RAID 5 que incluye **bits que permiten recuperar información en caso de que fallen 2 discos**.
- Se incrementa la robustez, puesto que fallos de dos dispositivos **no afectan al sistema**.
- A cambio, se pierde eficiencia puesto que **el cálculo de las paridades es más complicado**.
- Códigos como **Reed-Solomon** permiten construcción de códigos de error que permiten la recuperación de información cuando más de un dispositivo ha fallado.

31 Configuraciones RAID 0+1

- **RAID 0+1**: Combina las configuraciones 0 y 1.
- Sistema implementado como una **tabla distribuida por bloques**.
- Cada dispositivo que contiene un bloque, a su vez contiene un dispositivo redundante.
- Si un disco falla, anula la distribución, y hay que utilizar **la distribución en el espejo.**

32 Conexión de Discos al Sistema a través de Red

- La configuración más común es cuando el dispositivo **está conectado al bus del sistema**.
- Para **cantidades enormes de información** los discos están conectados de forma remota.
- Existen protocolos (por ejemplo NFS) que permiten la **gestión de discos remotos** a través del mecanismo de RPC.
- Ideal para entornos en los que un conjunto de equipos **deben compartir una gran cantidad de información.**
- Los sistemas operativos como Unix ofrecen **un espacio de nombres homogéneo** para este tipo de configuraciones.

33 Almacenamiento Terciario

- Por almacenamiento terciario se entiende **aquél que tiene un coste menor** que el almacenamiento en discos.
- En la práctica, este almacenamiento corresponde con **las plataformas extraíbles**.
- Este tipo de almacenamiento es **imprescindible** para sistemas que gestionan una cantidad **enorme** de datos.
- Las plataformas extraíbles se utilizan para poder almacenar grandes cantidades de datos con bajo costo y en lugares **separados de los propios equipos**.

34 Discos Extraíbles

- Los más comunes son los basados en métodos de almacenamiento **magnéticos**. Discos flexibles que pueden almacenar hasta **1 giga-byte** de información.
- Su principal desventaja es su **fragilidad**.
- Un campo magnético implica **una superficie delicada** y **una cabeza lectora/escritora que debe estar demasiado cerca de la superficie**.
- Como alternativa a estos discos aparecen aquellos que almacenan la información mediante técnicas **magneto-ópticas**.
- La superficie de grabación está **protegida por material resistente** (plástico o vidrio).
- La cabeza magnética está **muy separada de la superficie**. Esto, a temperatura ambiente, impide que el campo tenga algún efecto. Mediante la utilización de un láser, se calienta un punto del disco y el campo magnético surte efecto.
- La operación de lectura requiere también la utilización del láser para su lectura, pues **debido al efecto de Kerr**, la polarización de la luz se ve afectada por el campo magnético almacenado.

35 Discos Ópticos

- Estos discos no utilizan ningún tipo de **campo magnético**.
- Un ejemplo de tecnología óptica está basada en el **cambio de fase**.
- El disco contiene un material que puede **solidificarse** en estado cristalino o amorfo.
- El estado cristalino es **más transparente** a un rayo de luz que lo atraviese.
- Estos dispositivos utilizan una luz láser de tres intensidades: baja para leer, media para derretir y re-solidificar en estado cristalino, y alta para que se solidifique en estado amorfo.
- El ejemplo más común de este tipo de discos son los CD-RW y DVD-RW.

36 Cintas Magnéticas

- Permite una escala de almacenamiento **mayor que un disco duro**.
- El lector/escritor de cintas es **más caro que un disco**, en cambio, las cintas son **mucho más baratas**.
- Las cintas magnéticas requieren **acceso secuencial a su contenido**.
- Se utilizan **mayormente** para almacenar copias de seguridad de las que no tengan que extraerse datos a gran velocidad.
- Para la gestión de un número muy elevado de cintas se suelen utilizar **robots** que se encargan del almacenamiento, indexado, carga y descarga de las cintas en el lector.
- Estos robots pueden ser controlados como un dispositivo más por un sistema operativo.

37 Memorias flash

- Memorias no volátiles de lectura-escritura.
- Tipo particular de EEPROM (Electrically Erasable Programmable Read-Only Memory) que se programa y borra en bloques grandes (un chip se puede borrar por entero de forma conjunta).
- Con capacidades hasta 32 Gbytes.

Autoría del material