

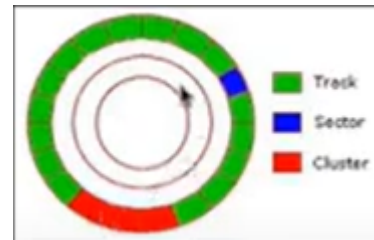
PRÁCTICA 6 - CSO

Administración de E/S

1. Dispositivos

a) Los dispositivos, según la forma de transferir los datos, se pueden clasificar en 2 tipos:

- Orientados a bloques
- Orientados a flujos



Describe las diferencias entre ambos tipos.

Orientados a bloques: Transfieren datos en bloques fijos de tamaño predeterminado. Los datos se leen o escriben en unidades de bloques, y cada bloque es tratado como una entidad independiente.

Cada bloque tiene dirección propia, permite leer, escribir o buscar un bloque sin depender de los demás. Utiliza buffer de datos del SO

Orientados a flujos: transfieren datos de manera continua, sin una estructura de bloques fijos. Los datos se transfieren en forma de un flujo continua

No utilizan buffer, transmiten bit o un byte a la vez. No permiten operaciones de búsqueda

b) Cite ejemplos de dispositivos de ambos tipos.

Orientado a bloques → disco magnético

Orientado a flujos → impresora, terminales, interfaz de redes

c) Enuncie las diferencias que existen entre los dispositivos de E/S y que el SO debe considerar.

- Heterogeneidad de dispositivos
- Características de los dispositivos
- Velocidad
- Nuevos tipos de dispositivos
- Diferentes formas de realizar E/S

2. Técnicas de E/S

Describe cómo trabajan las siguientes técnicas de E/S

- **E/S programada:** la CPU tiene control directo sobre la E/S. Controla el estado, comandos para leer y escribir, transfiere datos.
El procesador envía un mensaje de E/S, a petición de un proceso, a un módulo de E/S. A continuación ese proceso realiza una espera activa hasta que se complete la operación antes de continuar.
Esta técnica tiene como desventaja que hace que se desperdicien ciclos de CPU
- **E/S dirigida por interrupciones:** el dispositivo de E/S envía una solicitud de interrupción al procesador el cual solicita el mandato de la CPU.

La CPU deja su tarea actual y atiende la interrupción. El proceso en ejecución puede seguir con sus instrucciones únicamente si no necesita esperar a que termine la interrupción. Caso contrario, el proceso quedará en espera hasta que la interrupción finalice.

- **DMA (Acceso Directo a Memoria):** un módulo DMA controla el intercambio de datos entre memoria principal y el módulo de E/S. El procesador envía una petición de transferencia de un bloque de datos al módulo DMA y resulta interrumpido sólo cuando se haya transferido el bloque completo.

3. La técnica de E/S programa puede trabajar de dos formas:

- **E/S mapeada:** los dispositivos y memoria comparten el bus de direcciones.
- **E/S aislada:** se tiene un espacio de direcciones separadas de la memoria para E/S. Un bus dedicado a E/S.

Indique cómo trabajan estas 2 técnicas.

4. Enuncie las metas que debe perseguir un SO para la administración de la E/S.

- **Eficiencia:** maximizar el uso de los recursos de E/S para garantizar una operación eficiente del sistema. Se busca minimizar el tiempo de inactividad de la CPU mientras espera la finalización de operaciones de E/S
- **Distribución de recursos:** se busca distribuir equitativamente los recursos de E/S entre los distintos procesos y usuarios.
- **Prioridades:** priorizar las operaciones de E/S según la importancia de tareas y las necesidades del sistema.
- **Concurrencia:** gestionar de manera efectiva distintas operaciones de E/S concurrentes para evitar conflictos y asegurar coherencia de datos.
- **Uso eficiente de memoria:** minimizar la cantidad de memoria utilizada para buffers y otros dispositivos relaciones con la E/S, sin comprometer el rendimiento del sistema

5. Drivers

a) ¿Qué son?

Son programas de software que permiten que el SO interactúe con un HW específico (interfaz entre el SO y HW), como periféricos, dispositivos de almacenamiento, tarjetas de red, etc.

Los drivers contienen el código dependiente del dispositivo al cual manejan (siempre manejan un tipo de dispositivo). Traducen los requerimientos abstractos en los comandos para el dispositivo.

Las interrupciones generadas por los dispositivos son atendidas por funciones provistas por el driver.

b) ¿Qué funciones mínimas deben proveer?

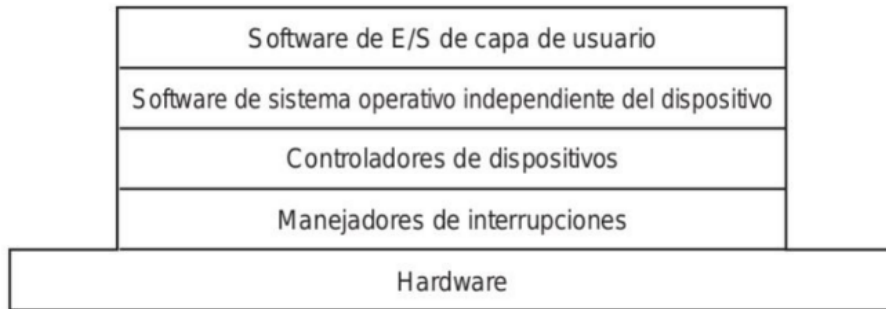
init_module: para instalarlo

cleanup_module: para desinstalarlo

c) ¿Quién determina cuáles deben ser estas funciones?

El fabricante

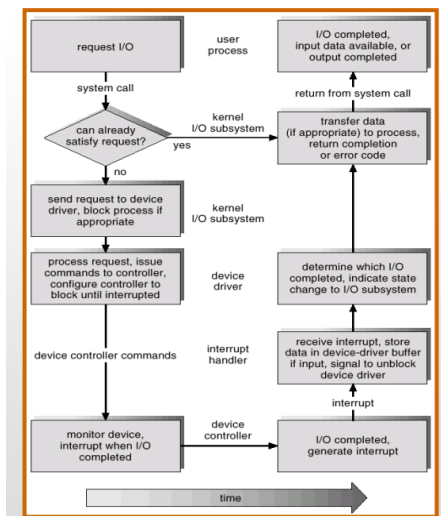
6. Realice un gráfico que marque la relación entre el Subsistema de E/S, los drivers, los controladores de dispositivos y los dispositivos.



7. Describa mediante un ejemplo los pasos mínimos que se suceden desde que un proceso genera un requerimiento de E/S hasta que el mismo llega al dispositivo.

- Determinar el dispositivo que almacena los datos
 - o Traducir el nombre del archivo en la representación del dispositivo.
- Traducir requerimiento abstracto en bloques de disco (Filesystem)
- Realizar la lectura física de los datos (bloques) en la memoria
- Marcar los datos como disponibles al proceso que realizó el requerimiento
- Desbloquearlo

8. Describa mediante un ejemplo los pasos mínimos que se suceden desde que un proceso genera un requerimiento de E/S hasta que el mismo llega al dispositivo.

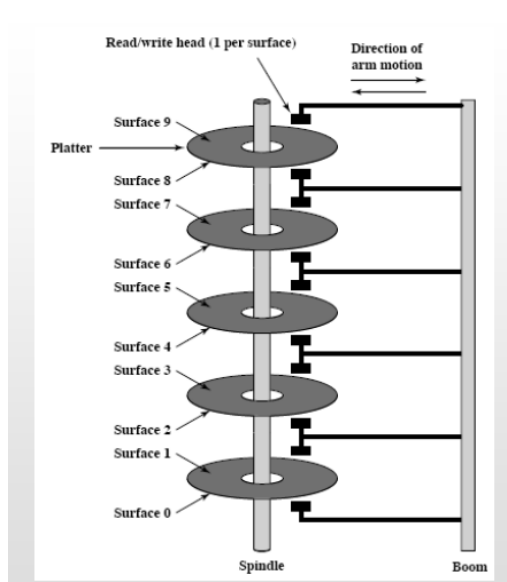


9. Enuncie qué servicios provee el SO para la administración de E/S

- Buffering → almacenamiento de los datos en memoria mientras se transfieren
 - Solucionar problemas de velocidad entre los dispositivos
 - Solucionar problemas de tamaño y/o forma de los datos entre los dispositivos
- Caching → mantener en memoria copia de los datos de reciente acceso para mejorar performance
- Spooling → administrar la cola de requerimientos de un dispositivo

Administración de E/S

10. Describa en forma sintética, cómo es la organización física de un disco, puede utilizar gráficos para mayor claridad.



Organización física de un disco duro

Posee:

- Platos, los cuales tienen caras útiles (pueden ser una o dos por plato)
- Brazo con un cabezal de lectura y escritura por superficie.
- Los discos giran y la unidad es RPM.
- La cara de un disco está formada por pistas (tracks)
 - porción de pista → sectores
 - cluster → conjunto de sectores

11. La velocidad promedio para la obtención de datos de un disco está dada por la suma de Seek Time, Latency Time y Transfer Time. De una definición para estos tres tiempos:

- **Seek Time** → tiempo que tarda en posicionarse la cabeza en el cilindro. *Tiempo que toma mover el brazo del disco al cilindro que se desea*
- **Latency Time** → tiempo que sucede desde que la cabeza se posiciona en el cilindro hasta que el sector en cuestión pasa por debajo de la misma. *Tiempo que el sector deseado tarda en girar y quedar debajo de la cabeza de disco*
- **Transfer Time** → tiempo de transferencia del sector del disco a la memoria. *Rapidez con la que los datos fluyen en la unidad de disco y la computadora.*

12. Suponga un disco con las siguientes características:

- 7 platos con 2 caras utilizables cada uno.
- 1100 cilindros → *pistas*
- 300 sectores por pista, donde cada sector de es 512 bytes.
- Seek Time de 10 ms
- 9000 RPM .

Velocidad de Transferencia de 10 MiB/s (Mebibytes por segundos).

(a) Calcule la capacidad total del disco.

$\text{tamaño_disco} = \#caras * \#pistas_cara * \#sectores_pista * \text{tamaño_sector}$

tamaño = $(7 * 2) * 1100 * 300 * 512$ bytes

tamaño = 2365440000 bytes = 2310000 Kib = 2255,859375 Gib

(divide 2 veces x 1024)

(b) ¿Cuántos sectores ocuparía un archivo de tamaño de 3 MiB(Mebibytes)?

512 bytes → capacidad de un sector

3 Mib x 1024 x 1024 = 3145728 bytes → pasamos los 3 Mib a bytes para tener las mismas unidades

$3145728 / 512 = 6144$ sectores

(c) Calcule el tiempo de transferencia real de un archivo de 15 MiB(Mebibytes). grabado en el disco de manera secuencial (todos sus bloques almacenados de manera consecutiva)

Latencia:

9 RPM → 9000 vueltas → 60 segs → 60000 ms

0.5 vueltas → $x = 3,333$ ms

Transferencia:

10 Mib → $10 \times 2^{20} = 10485760$ bytes

10485760 bytes → 1 s = 1000 ms

512 bytes → $x = 0,048828$ ms

Bloques:

15 Mib = 15728640 bytes

15728640 bytes / 512 bytes = 30720 bloques

Almacenamiento secuencial:

seek + latency + tiempo transferencia bloque * #bloques

$10 \text{ ms} + 3,333 \text{ ms} + (0,048828 \text{ ms} * 30720) = 1513,32916 \text{ ms}$

(d) Calcule el tiempo de transferencia real de un archivo de 16 MiB (Mebibytes). grabado en el disco de manera aleatoria

Latencia:

9 RPM \rightarrow 9000 vueltas \rightarrow 60 segs \rightarrow 60000 ms

0.5 vueltas $\rightarrow x = 3,333$ ms

Transferencia:

10Mib = 10485760 bytes $\rightarrow 1$ s = 1000 ms

512 bytes $\rightarrow x = 0,048828$ ms

Bloques:

16 MiB = 16777216 bytes

16777216 bytes / 512 bytes = 32768 bloques

Almacenamiento aleatorio:

(seek + latency + tiempo transferencia bloque) * #bloques

(10 ms + 3,333 ms + 0,048828 ms) * 32768 = 438495,7399 ms

13. El Seek Time es el parámetro que posee mayor influencia en el tiempo real necesario para transferir datos desde o hacia un disco. Es importante que el SO planifique los diferentes requerimientos que al disco para minimizar el movimiento de la cabeza lecto-grabadora.

Analicemos las diferentes políticas de planificación de requerimientos a disco con un ejemplo:

Supongamos un Head con movimiento en 200 tracks (numerados de 0 a 199), que está en el track 83 atendiendo un requerimiento y anteriormente atendió un requerimiento en el track 75.

Si la cola de requerimientos es: 86, 147, 91, 177, 94, 150, 102, 175, 130, 32, 120, 58, 66, 115. Realice los diagramas para calcular el total de movimientos de head para satisfacer estos requerimientos de acuerdo a los siguientes algoritmos de scheduling de discos:

(a) FCFS (First Come, First Served)

(b) SSTF (Shortest Seek Time First)

(c) Scan

(d) Look

(e) C-Scan (Circular Scan)

(f) C-Look (Circular Look)

En hoja

Administración de Archivos

17. Dados los siguientes métodos de administración de espacio de un archivo:

- **Asignación contigua**
- **Asignación enlazada**
- **Asignación indexada**

a) Describa cómo trabaja cada uno.

b) Cite ventajas y desventajas de cada uno.

Asignación contigua: se utilizan conjuntos de bloques, se requiere una pre-asignación debido a que se debe conocer el tamaño del archivo durante su creación.

File Allocation table (FAT) es simple, solo una entrada que incluye bloque de inicio y longitud. El archivo puede ser leído con una única operación. Puede existir fragmentación externa.

Desventajas → encontrar bloques libres continuos en el disco y el incremento del tamaño de un archivo

Asignación enlazada: la asignación es en base a bloques individuales, cada bloque tiene un puntero al próximo bloque del archivo.

File allocation table tiene una única entrada por archivo que incluye bloque de inicio y tamaño de archivo. No hay fragmentación externa es útil para acceso secuencial, los archivos pueden crecer bajo demanda y no se requieren bloques contiguos.

Se pueden consolidar los bloques de un mismo archivo para garantizar cercanía de los bloques de un mismo archivo.

Asignación indexada: la FAT contiene un puntero al bloque índice, el cual no contiene datos propios del archivo, sino que contiene un índice a los bloques que lo componen.

La asignación es en base a bloques individuales, no se produce fragmentación externa y el acceso random a un archivo es eficiente.

FAT posee una única entrada con la dirección del bloque de índices.

18. Gestión de espacio libre. Dados los siguientes métodos de gestión de espacio libre en un disco:

Gestión de espacio libre es el control sobre cuáles de los bloques de disco están disponibles

- **Tabla de bits**
- **Lista Ligada**

- **Agrupamiento Recuento**

a) Describa cómo trabajan estos métodos.

b) Cite ventajas y desventajas de cada uno.

Tabla de bits: es una tabla o vector con un bit por cada bloque de disco

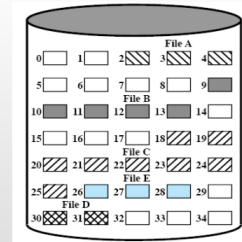
Cada entrada puede contener un 0 o 1, en donde 0 corresponde a un bloque libre y 1 a un bloque en uso

La ventaja es que es fácil de encontrar un bloque o grupo de bloques libres

La desventaja es el tamaño del vector en la memoria

✓ Ejemplo

00111
00001
11110
00011
11111
11110
11110
11000

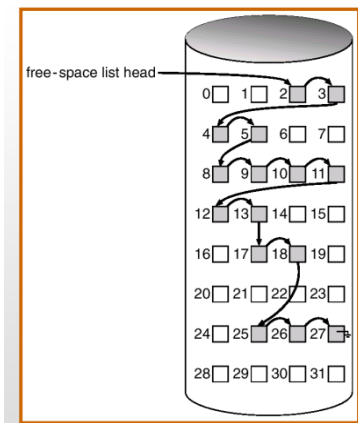


Lista ligada: se tiene un puntero al primer bloque libre, cada bloque libre tiene un puntero al siguiente bloque libre.

La desventaja es que es ineficiente para la búsqueda de bloques libres debido que hay que realizar varias operaciones de E/S para obtener un grupo libre.

Otra desventaja es que si se pierde un enlace se pierde la lista de bloques libres

Y por último es difícil encontrar bloques libres consecutivos.



Agrupamiento recuento: es una variante de bloques libres encadenados. El primer bloque libre contiene las direcciones de N bloques libres. Las N-1 primeras direcciones son bloques libres, la N-ésima dirección referencia a otro bloque con N direcciones de bloques libres

19. Gestión de archivos en UNIX.

El sistema de archivos de UNIX utiliza una versión modificada del esquema de Asignación Indexada para la administración de espacio de los archivos.

Cada archivo o directorio está representado por una estructura que mantiene, entre otra información, las direcciones de los bloques que contienen los datos del archivo: el I-NODO.

Cada I-NODO contiene 13 direcciones a los bloques de datos, organizadas de la siguiente

manera:

10 de direccionamiento directo.

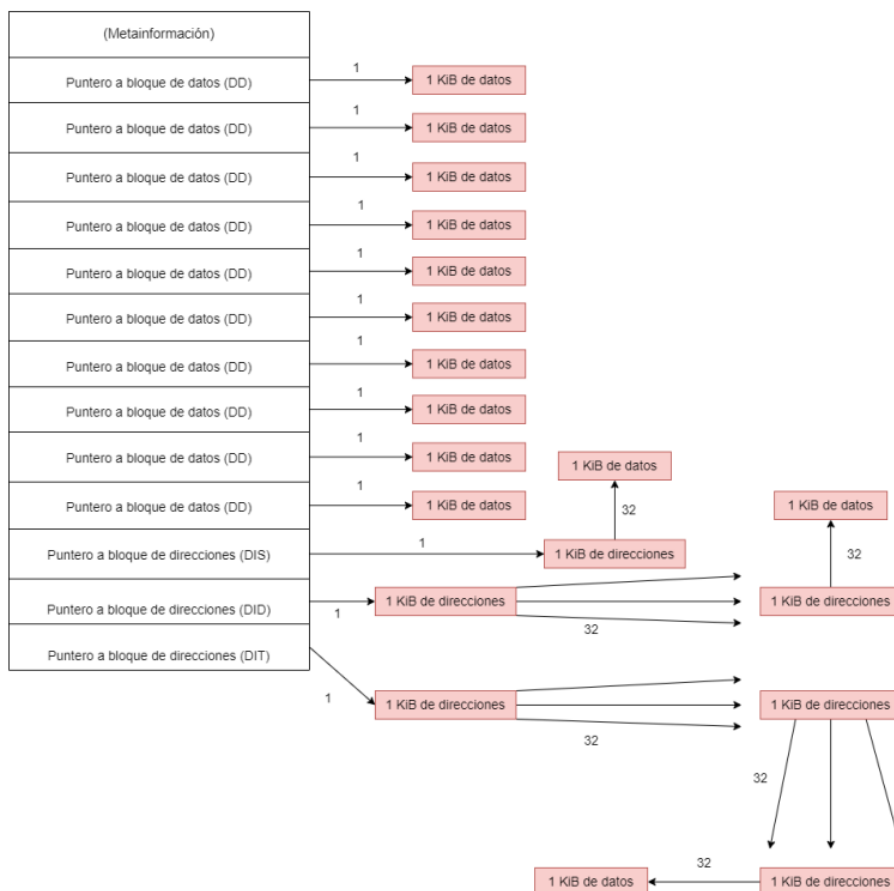
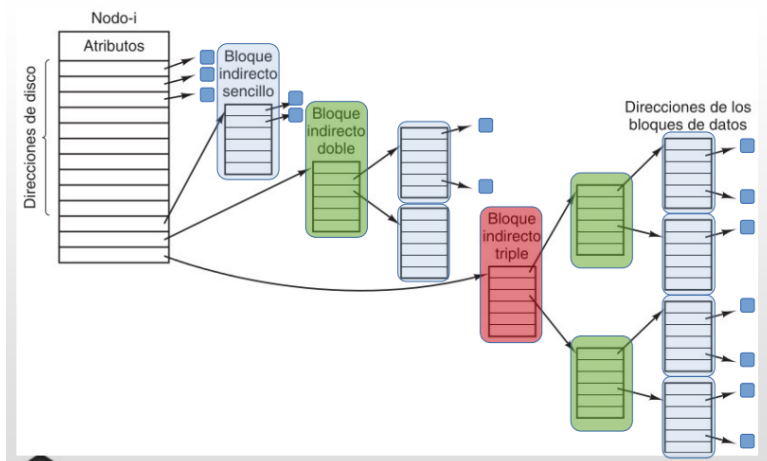
1 de direccionamiento indirecto simple.

1 de direccionamiento indirecto doble.

1 de direccionamiento indirecto triple.

a) Realice un gráfico que describa la estructura del I-NODO y de los bloques de datos.

Cada bloque es de 1 Kib(Kibibits). Si cada dirección para referenciar un bloque es de 32 bits:



i. ¿Cuántas referencias (direcciones) a bloque pueden contener un bloque de disco?

1 Kib \rightarrow 1024 bits

Cada dirección para referenciar un bloque es de 32 bits

$1024 \text{ bits} / 32 \text{ bits} = 32 \text{ direcciones por bloque}$

tamaño de INODO $\rightarrow 13 \times 32 \text{ bits} = 416 \text{ bits}$

ii. ¿Cuál sería el tamaño máximo de un archivo?

1 Kib * 10 de direccionamiento directo

1 Kib * 32 bits

10 punteros a bloque de datos (DD) + 1 puntero a bloque de direcciones que apunta a bloques de datos (DIS) + 1 puntero a bloque de direcciones que apunta a bloque de direcciones, que apuntan a bloques de datos (DID) + 1 puntero a bloque de direcciones que apunta a bloque de direcciones, que apunta a bloque de direcciones que apunta a bloque de direcciones

$(10 * 1 \text{ Kib}) + (1 \text{ Kib} * 32) + (1 \text{ Kib} * 32 * 32) + (1 \text{ Kib} * 32 * 32 * 32) = 33834 \text{ Kib} = 4229.25 \text{ KiB}$

1 Kib $- 0,125 \text{ KiB}$

$33834 \text{ Kib} - x = 4229.25 \text{ KiB}$

20. Analice las siguientes fórmulas necesarias para localizar un I-NODO en la lista de inodos

nro bloque = ((nro de inodo -1)/nro. de inodos por bloque) + bloque de comienzo de la lista de inodos.

Desplazamiento del inodo en el bloque = ((nro de inodo - 1) módulo (número de inodos por bloque)) * medida de inodo del disco.

a) Según la primera fórmula, asumiendo que en el bloque 2 está en el comienzo de la lista de inodos y que hay 8 inodos por bloque: calcule donde se encuentra el inodo 8 y el 9. ¿Dónde estarían para bloque de disco de 16 inodos?

- Si hay 8 inodos por bloque
 - Inodo 8
 $\text{nro bloque} = ((8-1) / 8) + 2 = 2,875$

- Inodo 9

$$\text{nro bloque} = ((9-1) / 8) + 2 = 3$$
- Si hay 16 inodos por bloque
 - Inodo 8

$$\text{nro bloque} = ((8-1) / 16) + 2 = 2,4375$$
 - Inodo 9

$$\text{nro bloque} = ((9-1) / 16) + 2 = 2,5$$

b) De acuerdo a la segunda fórmula, si cada inodo del disco ocupa 64 bytes y hay 8 inodos por bloque de disco, el inodo 8 comienza en el desplazamiento 448 del bloque de disco. ¿Dónde empieza el 6? Si fueran inodos de 128 bytes y 24 inodos por bloque: ¿dónde empezaría el inodo 8?

- Si hay 8 inodos por bloque de tamaño 64 bytes
 - Inodo 6

$$\text{Desplazamiento} = ((6-1) \% 8) * 64 \text{ bytes} = 5 * 64 \text{ bytes} = 320$$
- Si hay 24 inodos por bloque de tamaño 128 bytes
 - Inodo 8

$$\text{Desplazamiento} = ((8-1) \% 24) * 128 \text{ bytes} = 7 * 128 \text{ bytes} = 896$$