CSO TP6

Administración de E/S

1)

a) Un dispositivo orientado a bloque almacena información bloques que son usualmente de tamaño fijo realizándose la transferencia de bloques en bloques, las cintas son un ejemplo de dispositivo orientado a bloque. Un dispositivo orientado a flujo de caracteres transfiere los datos tanto de entrada como de salida, como un flujo de bytes sin estructuras de bloques.

b) Los terminales, las impresoras, puertos de comunicación, el mouse y otros dispositivos apuntadores están orientados a flujos de caracteres. Mientras que los discos duros, las lectoras CD-ROM, y las memorias flash están orientadas a bloques.

c)

1 – Naturaleza de acceso:

* Bloques: permiten acceso aleatorio (lectura y escritura en cualquier parte del dispositivo). El SO debe gestionar las operaciones de búsqueda, transferencia de bloques y buffering.
* Flujos: requieren manejo secuencial, con operaciones síncronas o asincronías.

2- Velocidad de transferencia : Los dispositivos tienen diferentes velocidades de transferencia que el SO debe sincronizar con buffers en memoria.

3- Capacidad de almacenamiento: los dispositivos orientados a bloques tienden a tener grandes capacidades, lo que requiere sistemas de archivos para su administración.

4- Sincronización:

* Para dispositivos lentos como impresoras, el SO puede necesitar implementar técnicas de spooling.
* Para dispositivos rápidos, el SO debe optimizar la interrupción y buffering.

5- Error Handling:

* Dispositivos orientados a bloques pueden requerir manejo avanzado de errores como detección y corrección de sectores defectuosos.
* Dispositivos orientados a flujos suelen necesitar menos corrección de errores a nivel físico, pero más control lógico (como en la secuencia de teclas).

2) a) En la E/S programada, la CPU se encarga de iniciar de supervisar las operaciones de entrada/salida mediante instrucciones específicas. La CPU envía comandos al dispositivo E/S, espera a que complete la operación y luego transfiere los datos entre el dispositivo y la memoria principal. Sus características mas importantes son: es sencilla de implementar, la CPU controla todos los aspectos del proceso, pero es ineficiente porque la CPU pierde tiempo esperado.

b) En la E/S dirigida por interrupciones la CPU envía un comando al dispositivo de E/S y sigue ejecutando otras tareas mientras espera la operación. Cuando el dispositivo termina, genera una interrupción para notificar a la CPU. La CPU responde a la interrupción mediante una rutina de servicio que procesa los datos transferidos. Sus características: Mejora el rendimiento al permitir que la CPY ejecuto otras tareas en lugar de esperar, la CPU aun gestiona las transferencias entre la memoria principal y el dispositivo. Requiere soporte de HW para interrupciones.

c) El DMA se encarga de transferir datos directamente entre el dispositivo de E/S y la memoria principal. La CPU solo inicializa la transferencia indicando la dirección de memoria, el tamaño de los datos y el dispositivo involucrado. Una vez configurado, el controlador DMA gestiona la transferencia.

Cuando la operación finaliza, el controlador DMA genera una interrupción para notificar a la CPU.

Sus características: Reduce significativamente la carga de trabajo de la CPU, Ideal para transferencias grandes y rápidas, como en discos duros o tarjetas de red, Requiere hardware especializado (el controlador DMA).

3)

a) La E/S mapeada usa el mismo bus de direcciones para memoria y dispositivos de E/S, y las instrucciones de la CPU usadas para acceder a la memoria son también usadas para acceder los dispositivos. Para tener espacio para los dispositivos de E/S, las áreas del espacio direccionable por la CPU deben ser reservadas para E/S más que para memoria. Esta reserva puede ser temporal o permanente. Cada dispositivo de E/S monitoriza el bus de direcciones de la CPU y responde a cualquier acceso de esta al espacio de direcciones del dispositivo, conectando el bus de datos con la localización en memoria física del dispositivo deseado.

b) La E/S aislada contempla el acceso ala entrada/salida y se caracteriza por tener dos mapas de memoria separados, uno para la memoria y otro para la E/S. además cuenta con señales e instrucciones específicas. En general, los dispositivos de E/S permiten que el sistema recopile, almacene y transmita datos. Se encuentran en la unidad de servidor o en las unidades de expansión que están conectadas a ella. Para comunicarse con los dispositivos de E/S, la CPU utiliza instrucciones de referencia a memoria. El procesador ve el módulo de E/S como un espacio de memoria, pero las direcciones corresponden a los registros del módulo de E/S y se denominan puertos de E/S.

5) a) Los drivers o controladores son programas que permiten al SO interactuar con un periférico, haciendo una abstracción del HW y proporcionando una interfaz para utilizar el dispositivo.

b) Las funciones básicas de un driver son la inicialización, administración del dispositivo, el manejo de errores, la gestión de interrupciones, la liberación de recursos.

c) El fabricante

6) Tabla

Descripción generada automáticamente

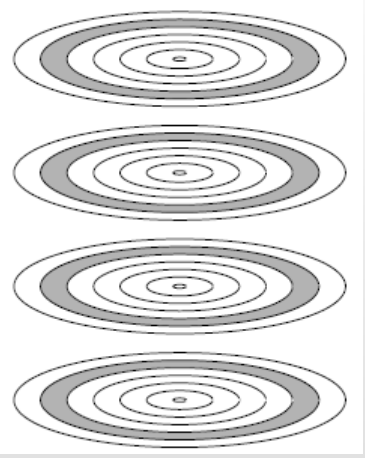
Administración de Discos

Diagrama

Descripción generada automáticamente1)

Gráfico, Gráfico de proyección solar

Descripción generada automáticamente

Cilindro N: todas las n-esimas pistas de todas las caras

2)

Seek Time (posicionamiento): tiempo que tarda en posicionarse la cabeza en el cilindro

Latency time (latencia): tiempos que sucede desde que la cabeza se posiciona en el cilindro hasta que el sector en cuestión pasa por debajo de la misma

Transfer time (transferencia): tiempo de transferencia del sector (bloque) del disco a la memoria.

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja

3) a) tamaño\_disco = 7 \* 2 \*1100 \*300 \*512 bytes = 2365440000 bytes = 2310000 KiB = 2255.859375 MiB = 2.2029076709 GiB

b) Si la capacidad de un sector es de 512 bytes: dividimos el tamaño del archivo por la capacidad de una cara

3 MiB = 3145728 bytes

3145728 / 512 = 6144 sectores

c) Latencia:

9000 🡪 60000 ms

0.5 🡪 3.333 ms

Transferencia:

10 MiB 🡪 1000 ms

512 bytes 🡪 x

Unificar unidades

10485760 bytes 🡪 1000ms

512 bytes 🡪 0,048828125 ms

Bloques: 15728640 bytes / 512 = 30720

Datos obtenidos:

Seek time: 10 ms

Latency time: 3.333 ms

Tiempo de transferencia bloque: 0,048828125 ms

Cantidad de bloques: 30720

Almacenamiento secuencial: 10ms + 3.333ms + (0,048828125 ms \* 30720) = 4843 ms

d) **Latencia:**

9000 vueltas 🡪 60000 ms

0.5 🡪 3.333 ms

**Transferencia:**

10 MiB 🡪 1000 ms

512 bytes 🡪 Z

Unificar unidades:

10485760 bytes 🡪 1000 ms

512 bytes 🡪 0.048828125 ms

**Bloques:** 16777212 bytes / 512 bytes = 32768

**Datos obtenidos:**

Seek time: 10 ms

Latency time: 3.333 ms

Tiempo transferencia bloque: 0.048828125 ms

Cantidad de bloques: 32768

**Almacenamiento aleatorio:** (10ms + 3.333ms + 0.048828125ms) \* 32768 = 438397,44 ms

4) Si, El SSTF

Administración de Archivos

1) **Asignación Contigua:** los bloques de disco asignados a un archivo están almacenados en secuencia continua. Es decir, los bloques ocupan posiciones consecutivas en el disco.

Ventajas:

* Excelente rendimiento en lectura/escritura secuencial, ya que no hay necesidad de buscar bloques separados.
* Fácil de implementar con acceso directo, ya que la ubicación de cualquier bloque se calcula fácilmente como dirección\_base + (número\_bloque × tamaño\_bloque).

Desventajas:

* Problemas de fragmentación externa: con el tiempo, puede ser difícil encontrar un espacio contiguo lo suficientemente grande para archivos grandes.
* Dificultad para expandir archivos: si un archivo necesita crecer, es posible que no haya espacio contiguo disponible.

**Asignación enlazada:** Cada archivo se almacena como una lista de bloques enlazados. Cada bloque contiene un puntero al siguiente bloque, y no necesitan ser contiguos en el disco.

Ventajas:

* No hay fragmentación externa, ya que los bloques no necesitan ser contiguos.
* Flexible para crecer o reducir archivos dinámicamente.

Desventajas:

* Bajo rendimiento en lectura/escritura, especialmente para accesos directos, ya que se deben seguir punteros de un bloque a otro.
* Vulnerabilidad a fallos: si un bloque enlazado se daña, se puede perder el acceso a parte o todo el archivo.
* Sobrecarga adicional: cada bloque necesita almacenamiento para el puntero al siguiente bloque.

**Asignación indexada:** Se utiliza un bloque índice que contiene una lista de punteros a los bloques que almacenan los datos del archivo. Los bloques de datos no necesitan ser contiguos.

Ventajas:

* Soporte para acceso directo eficiente: se puede acceder a cualquier bloque directamente utilizando el índice.
* No hay fragmentación externa y el archivo puede crecer dinámicamente.
* Eficiente para manejar archivos grandes, ya que el índice organiza todos los bloques relacionados.

Desventajas:

* Sobrecarga de espacio: se necesita un bloque adicional para almacenar los índices.
* Para archivos muy grandes, el bloque índice puede no ser suficiente, requiriendo índices multinivel que complican la estructura.
* Complejidad de implementación mayor en comparación con los otros métodos.

2) **Tabla de Bits (Bitmap):** El disco se divide en bloques y se usa una tabla de bits donde cada bit representa un bloque del disco. Un bit en 1 indica que el bloque está ocupado, y un bit en 0 indica que está libre.

Ventajas:

* Permite búsquedas rápidas de bloques libres mediante operaciones de bits, lo que es útil para encontrar múltiples bloques contiguos.
* Fácil de implementar y mantener

Desventajas:

* Requiere memoria adicional para almacenar el mapa de bits, especialmente en discos grandes.
* Buscar espacio libre puede ser lento si no se optimiza, ya que requiere escanear los bits.

**Lista ligada:** Los bloques libres se organizan como una lista enlazada. Cada bloque libre contiene un puntero al siguiente bloque libre. Solo se necesita almacenar el puntero al primer bloque libre de la lista.

Ventajas:

* No se requiere memoria adicional significativa; los punteros se almacenan en los bloques libres.
* Fácil de implementar y mantener en sistemas simples.

Desventajas:

* Ineficiente para búsquedas, ya que requiere seguir punteros, especialmente si se buscan múltiples bloques libres contiguos.
* Vulnerable a la pérdida de punteros: si se daña un bloque libre, puede romperse la lista y dificultar la recuperación.

**Agrupamiento (Grouping):** Similar a la lista ligada, pero en lugar de almacenar solo un puntero en cada bloque libre, se agrupan direcciones de bloques libres en un solo bloque. El primer bloque libre contiene punteros a varios bloques libres adicionales, reduciendo el número de accesos necesarios para buscar espacio libre.

Ventajas:

* Reduce el número de accesos al disco en comparación con la lista ligada, ya que un solo bloque puede almacenar referencias a varios bloques libres.
* Más eficiente para administrar grandes cantidades de bloques libres.

Desventajas:

* Requiere almacenamiento adicional en los bloques para almacenar múltiples punteros.
* Puede ser más complejo de implementar en comparación con la lista ligada.

**Recuento (Counting):** Este método utiliza pares de valores que indican el inicio de un grupo de bloques libres y la cantidad consecutiva de bloques libres.

Ventajas:

* Muy eficiente para discos con grandes áreas contiguas de bloques libres, ya que se almacena un rango en lugar de direcciones individuales.
* Ocupa menos espacio en memoria en comparación con otros métodos como la lista ligada o el agrupamiento.

Desventajas:

* No es adecuado para discos altamente fragmentados, ya que las áreas contiguas son menos comunes.
* Más complicado de implementar si las áreas libres cambian frecuentemente.