**1. Dispositivos**

**(a) Los dispositivos, según la forma de transferir los datos, se pueden clasificar en 2 tipos: Orientados a bloques , y Orientados a flujos Describa las diferencias entre ambos tipos.**

|  |  |  |
| --- | --- | --- |
|  | **DISPOSITIVOS ORIENTADOS A BLOQUES** | **DISPOSITIVOS ORIENTADOS A FLUJOS** |
| Acceso | Permiten el acceso aleatorio (no secuencial) a bloque de datos. | Permiten solo el acceso secuencial. Significa que los datos deben ser leidos o escritos secuencialmente |
| Unidad de transferencia | Los datos se transfieren en bloques de tamaño fijo | Los datos se transfueren como un flujo continuo de bytes |
| Uso | Son ideales para dispositivos de almacenamiento donde se necesita acceder a datos no secuenciales. | Adecuados para dispositivos de e/s donde el flujo es secuencial |
| Eficiencia | Mas eficientes para grandes volúmenes de datos, debiso a su transferencia por bloques. | Adecuados para datos pequeños |

**(b) Cite ejemplos de dispositivos de ambos tipos.**

**Dispositivos orientados a bloques:** HDD (Disco Duro) y SSD (Disco Solido), memorias USB, tarjetas SD

**Dispositivos orientados a flujos:** Teclados,mouses,micrófonos.

**(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 Describa como trabajan las siguientes técnicas de E/S**

**• E/S programada:** El procesador envía un mandato 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.

**• E/S dirigida por interrupciones:** El procesador emite un mandato de E/S a petición de un proceso y continúa ejecutando las instrucciones siguientes, siendo interrumpido por el módulo de E/S cuando éste ha completado su trabajo. Las siguientes instrucciones pueden ser del mismo proceso, en el caso de que ese proceso no necesite esperar hasta que se complete la E/S. En caso contrario, se suspende el proceso en espera de la interrupción y se realiza otro trabajo.

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

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

**• E/S mapeada:** Los dispositivos y memoria comparten el espacio de direcciones. I/O es como escribir/leer en la memoria por lo que no hay instrucciones especiales para I/O (ya se dispone de muchas instrucciones para la memoria)

**• E/S aislada:** Hay un espacio separado de direcciones para I/O. Se necesitan líneas de I/O. Puertos de E/S. Instrucciones especiales (conjunto limitado)

**5. Drivers (a) ¿Qué son?**

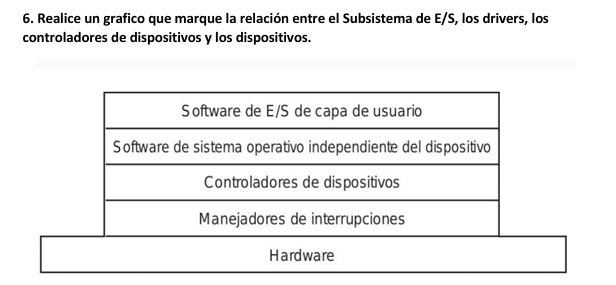
Consiste en una interfaz entre el SO y el HARDWARE cargadas como módulos en el espacio de memoria del kernel conteniendo el código dependiente del dispositivo para manejar el mismo y traducir requerimientos abstractos en los comandos para el dispositivo manejado. (b) ¿Qué funciones mínimas deben proveer?

• init\_module: Para instalarlo

• cleanup\_module: Para desinstalarlo.

**(c) ¿Quién determina cuales deben ser estas funciones?**

El fabricante



**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**

Consideremos la lectura sobre un archivo en un disco:

• Determinar el dispositivo que almacena los datos : 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 realizo el requerimiento

• Desbloquearlo: Retornar el control al proceso

**9. Enuncie que 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.

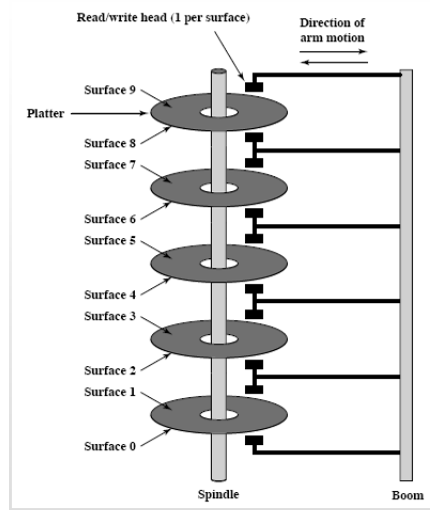
**Caching** – Mantener en memoria copia de los datos de reciente acceso para mejorar performance.

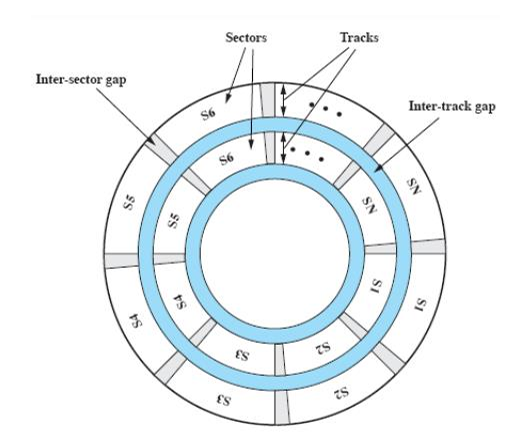
**Spooling** – Administrar la cola de requerimientos de un dispositivo.

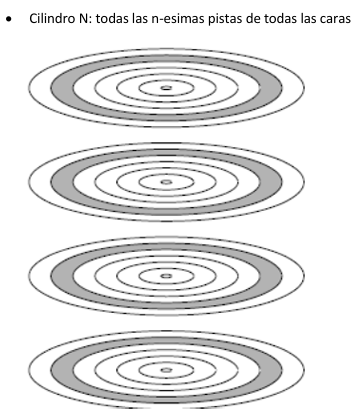
• Algunos dispositivos de acceso exclusivo, no pueden atender distintos requerimientos al mismo tiempo: Por ej. Impresora.

• Spooling es un mecanismo para coordinar el acceso concurrente al dispositivo.

**10: Describa en forma sintetica, como es la organizacion fisica de un disco.**







**11. La velocidad promedio para la obtención de datos de un disco está dada por la suma de los siguientes tiempos:**

**• Seek Time (posicionamiento):** Tiempo que tarda en posicionarse la cabeza en el cilindro

**• Latency Time (latencia):** Tiempo 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.

**12: Suponga un disco con las siguientes características:**

• 7 platos con 2 caras utilizables cada uno.

• 1100 cilindros

• 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).

1. **Calcule la capacidad**

Capacidad = 7 \* 2 \* 1100 \* 300 \* 512 bytes = 2365440000 bytes = 2.310.000KIB

= 2.255,859375MIB = 2,20gb

1. **¿Cuantos sectores ocuparía un archivo de tamaño de 3 MiB(Mebibytes)?**

Tenemos que pasar a bytes esos 3MIB = 3 \* 1024 \* 1024 = 3.145.728bytes

Para calcular los sectores hacemos 3.145.728bytes / 512 = 6144

Por ende ocuparía 6144 sectores un archivo de3MIB (3.145.728bytes)

1. **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)**

9000 vueltas = 60 segundos = 60000ms

Tiempo en dar una vuelta= 60 / 9.000 = 0.0067 segundos = 6.66 ms

O la otra es hacer

Tiempo en dar una vuelta = 60.000/9.000 = 6.66ms

Tiempo por media vuelta = 6.66ms / 2 = 3.33ms

Latencia = 0.5 \* tiempo por vuelta = 0.5 \* 6.66ms = 3.33ms

Velocidad de transferencia = 10MIB = 10 \* 1024 \*1024 = 10,485,760bytes/s.

Tamaño bloque : 512 bytes

Tiempo por bloque = TAMAÑO DEL BLOQUE / VELOCIDAD DE TRANSFERENCIA

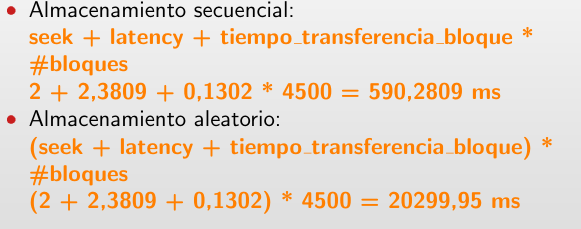
Tiempo por bloque = 512 / 10,485,760bytes = 0.0488ms

Numero de bloques = Tamaño archivo (bytes) / Tamaño de un bloque (bytes)

Tamaño del archivo = 15MIB = 15×1024×1024=15,728,640bytes

Tamaño de un bloque = 512bytes

Numero de bloques = 15,728,640bytes / 512bytes = 30.720 bloques



Almacenamiento Secuencial:

Seek = 10ms (lo da el enunciado). Latencia = 3.33ms, Tiempo por bloque: 0.0488ms

Numero de bloques = 30.720

Reemplazando en la formula de almacenamiento secuencial:

10 + 3.33 + 0.0488 \* 30.720 = 1.513,33ms = 1.5 s

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

9000 vueltas = 60 segundos = 60000ms

Tiempo en dar una vuelta= 60 / 9.000 = 0.0067 segundos = 6.66 ms

O la otra es hacer

Tiempo en dar una vuelta = 60.000/9.000 = 6.66ms

Tiempo por media vuelta = 6.66ms / 2 = 3.33ms

Latencia = 0.5 \* tiempo por vuelta = 0.5 \* 6.66ms = 3.33ms

Velocidad de transferencia = 10MIB = 10 \* 1024 \*1024 = 10,485,760bytes/s.

Tamaño bloque : 512 bytes

Tiempo por bloque = TAMAÑO DEL BLOQUE / VELOCIDAD DE TRANSFERENCIA

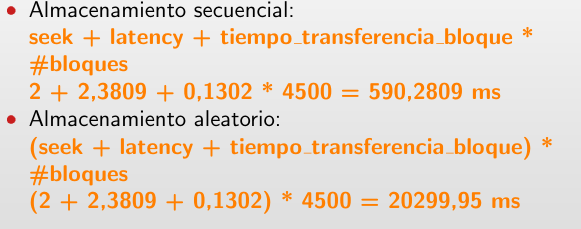
Tiempo por bloque = 512 / 10,485,760bytes = 0.0488ms

Numero de bloques = Tamaño archivo (bytes) / Tamaño de un bloque (bytes)

Tamaño del archivo = 16MIB = 16×1024×1024=16,777,216bytes

Tamaño de un bloque = 512bytes

Numero de bloques = 16,777,216bytes / 512bytes = 32.768



Almacenamiento Aleatorio:

Seek = 10ms (lo da el enunciado). Latencia = 3.33ms, Tiempo por bloque: 0.0488ms

Numero de bloques = 32.768

Reemplazando en la formula de almacenamiento secuencial:

(10 + 3.33 + 0.0488) \* 32.768 = 438,396.5184ms = 438segundos

**13) Resuelto en hoja. Deje las fotos.**

**14. ¿Alguno de los algoritmos analizados en el ejercicio anterior pueden causar inanición de requerimientos?**

El SSTF es el que mas probabilidades de inanición tiene, ya que si un requerimiento tiene el valor del limite, y empiezan a llegar cada vez mas procesos cercanos al requerimiento actual,no se podrá ejecutar hasta que se ejecuten los anteriores a el

**15) Resuelto en hoja. Deje las fotos.**

**16) hacerlo.**