

## Práctica 6 – ISO

### 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	Orientados a flujos
<ul style="list-style-type: none"> <li>- La transferencia de datos se realiza en bloques fijos de tamaño predeterminado. Cada bloque se lee o escribe como una unidad completa.</li> <li>- Los datos se agrupan en bloques antes de ser transferidos entre la memoria y el dispositivo. Cada bloque tiene una estructura fija y es identificado por un número de bloque.</li> <li>- La transferencia en bloques tiende a ser más eficiente en términos de rendimiento.</li> </ul>	<ul style="list-style-type: none"> <li>- La transferencia de datos se realiza de manera continua, sin una estructura fija.</li> <li>- No hay una división fija en bloques, los datos se transmiten como un flujo constante de bytes.</li> <li>- Son eficientes para operaciones secuenciales donde se requiere un flujo constante de datos.</li> <li>- No poseen una unidad de transferencia predeterminada.</li> </ul>

(b) Cite ejemplos de dispositivos de ambos tipos.

Dispositivos Orientados a Bloques: Discos magnéticos y SSDs.

Dispositivos Orientados a flujos: Teclados, mouse, impresoras, terminales, interfaces de redes, etc.

(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 como trabajan las siguientes técnicas de E/S

- *E/S programada*
- *E/S dirigida por interrupciones*
- *DMA (Acceso Directo a Memoria)*

E/S Programada: La CPU controla directamente todas las operaciones de transferencia de datos entre la memoria y los periféricos, además emite comandos y supervisa el estado del dispositivo periférico y espera activamente hasta que la operación de E/S se completa.

E/S Dirigida por Interrupciones: La CPU emite un comando de E/S al periférico y luego continúa con otras tareas. Cuando el periférico completa su operación, envía una interrupción a la CPU para notificar que los datos están listos.

DMA (Acceso Directo a Memoria): Se hace uso de un controlador de DMA para transferir datos directamente entre la memoria y el periférico sin intervención constante de la CPU. Esta última se encarga de inicializar el

controlador con la dirección de inicio y la cantidad de datos a transferir y luego se libera para realizar otras tareas mientras el controlador DMA maneja la transferencia.

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

- E/S mapeada
- E/S aislada

Indique como trabajan estas 2 técnicas.

E/S mapeada: los dispositivos y memoria comparten el espacio de direcciones. Las operaciones de E/S pasan a ser 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 E/S. Se necesitan líneas de E/S. Puertos de E/S. Instrucciones especiales (conjunto limitado).

4. Enuncie las metas que debe perseguir un SO para la administración de la entrada salida.

- Eficiencia: gestionar los recursos de E/S para minimizar tiempos de espera y maximizar el uso de los dispositivos.
- Desempeño: mejorar el rendimiento con técnicas como buffering, caché y planificación eficiente.
- Abstracción de hardware: proveer una interfaz que oculte los detalles del hardware para facilitar el desarrollo y la portabilidad de aplicaciones.
- Concurrencia: permitir operaciones de E/S simultáneas y gestionar el acceso compartido entre procesos.
- Gestión de dispositivos: manejar diferentes tipos de dispositivos mediante controladores adecuados y asignarlos eficientemente.

5. Drivers

(a) ¿Qué son?

Son interfaces entre el SO y el Hardware, forman parte del espacio de memoria del Kernel y en general se cargan como módulos del mismo. Contienen el código dependiente del dispositivo para poder manejarlo y se encargan de traducir los requerimientos abstractos en los comandos para el dispositivo escribiendo sobre los registros del controlador, accediendo a la memoria mapeada y encolando requerimientos.

(b) ¿Qué funciones mínimas deben proveer?

Inicialización y configuración:

- Preparar el dispositivo para su uso (configuración de registros, memoria, etc.).
- Realizar diagnósticos iniciales para verificar que el dispositivo funciona correctamente.

Interfaz de comunicación:

- Proveer funciones para enviar comandos al dispositivo.
- Facilitar la transferencia de datos entre el dispositivo y el sistema.

Gestión de interrupciones:

- Manejar las interrupciones generadas por el dispositivo para notificar al SO sobre eventos importantes (por ejemplo, finalización de una operación).

Control de errores:

- Detectar y reportar errores en las operaciones.
- Implementar mecanismos básicos para recuperar o manejar fallos del dispositivo.

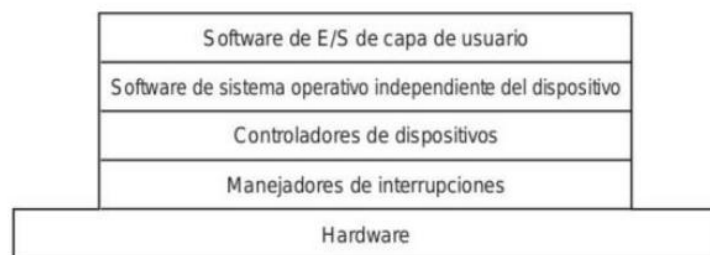
Operaciones de entrada/salida básicas:

- Leer datos del dispositivo y escribir datos hacia él.
- Asegurar que las operaciones se realicen en el orden correcto y de forma segura.

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

El proveedor es quien determina las funciones del Driver.

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.

1. Determinar el dispositivo que almacena los datos.
  - Traducir el nombre del archivo en la representación del dispositivo.
2. Traducir requerimiento abstracto en bloques de disco (Filesystem).
3. Realizar la lectura física de los datos (bloques) en la memoria.
4. Marcar los datos como disponibles al proceso que realizó el requerimiento.
  - Desbloquearlo.
5. 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 ejemplo: impresora.
- Es un mecanismo para coordinar el acceso concurrente al dispositivo

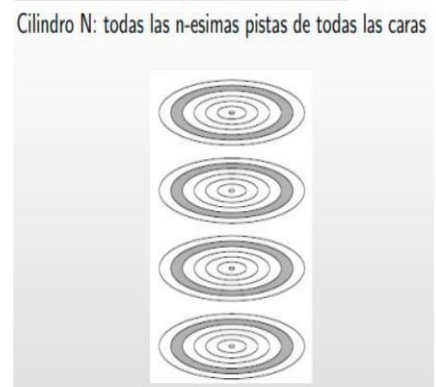
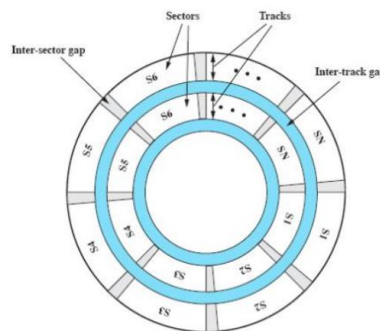
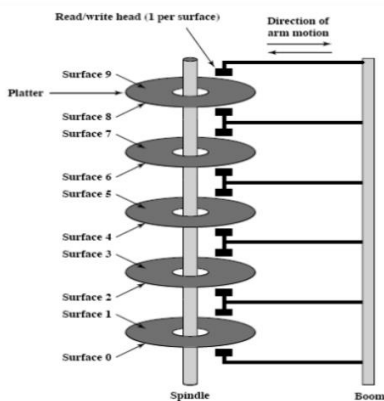
### Administración de Discos

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

La organización física de un disco se basa en una estructura jerárquica para almacenar y acceder a datos.

Incluye:

- Platos: superficies circulares recubiertas de material magnético, donde se almacena la información. Cada disco puede tener múltiples platos apilados.
- Pistas: centros circulares en la superficie de cada plato, organizados desde el borde hacia el centro.
- Sectores: divisiones de las pistas en segmentos más pequeños, que son las unidades mínimas de almacenamiento.
- Cilindros: conjunto de pistas alineadas verticalmente en los platos, formando un cilindro lógico que puede ser accedido simultáneamente.
- Cabezas de lectura/escritura: brazo mecánico con cabezas que se mueven sobre las pistas de los platos para leer o escribir datos.
- Motor: controla la rotación de los platos y el movimiento de las cabezas, permitiendo el acceso a las posiciones deseadas.



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

- *Seek Time*
- *Latency Time*
- *Transfer Time*

De una definición para estos tres 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 es de 512 bytes.
- Seek Time de 10 ms.
- 9000 RPM.
- Velocidad de Transferencia de 10 MiB/s (Mebibytes por segundos).

- Tamaño del disco = #caras \* #pistas\_por\_cara \* #sectores\_por\_pista \* tamaño\_del\_sector
- #Pistas por cara = #cilindros
- Cantidad de sectores que ocupa un archivo = tamaño\_del\_sector / tamaño\_del\_archivo
- Cantidad de bloques que ocupa un archivo = tamaño\_del\_archivo / tamaño\_del\_sector
- Almacenamiento secuencial = seek + latency + tiempo\_transferencia\_bloque \* #bloques
- Almacenamiento aleatorio = (seek + latency + tiempo\_transferencia\_bloque) \* #bloques

Unidades básicas de información (en bytes)				
Prefijos del Sistema Internacional			Prefijo binario	
Múltiplo - (Símbolo)	Estándar SI	Binario	Múltiplo - (Símbolo)	Valor
kilobyte (kB)	$10^3$	$2^{10}$	kibibyte (KiB)	$2^{10}$
megabyte (MB)	$10^6$	$2^{20}$	mebibyte (MiB)	$2^{20}$
gigabyte (GB)	$10^9$	$2^{30}$	gibibyte (GiB)	$2^{30}$
terabyte (TB)	$10^{12}$	$2^{40}$	tebibyte (TiB)	$2^{40}$

- (a) Calcule la capacidad total del disco.
- (b) ¿Cuántos sectores ocuparía un archivo de tamaño de 3 MiB(Mebibytes)?
- (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)
- (d) Calcule el tiempo de transferencia real de un archivo de 16 MiB(Mebibytes). Grabado en el disco de manera aleatoria.

<p>12.</p> <p>(a) Tamaño = <math>(7.2) \cdot 1100 \cdot 300 \cdot 512 \text{ bytes}</math></p> <p>= <math>14 \cdot 1100 \cdot 300 \cdot 512 \text{ bytes}</math></p> <p>= <math>2.365.440.000 \text{ bytes} \mid 2^{30} \approx 2,20 \text{ GiB}</math></p>	
<p>(b) ArcBytes = <math>3 \text{ MiB} \cdot 2^{20} = 3.145.728 \text{ bytes}</math></p> <p>sectores = <math>3.145.728 \text{ bytes} \mid 512 \text{ bytes} = 6144</math></p>	
<p>(c y d)</p> <p>Latencia (en 1/2 vuelta):</p> <p>9000 vueltas <math>\rightarrow 1 \text{ min} = 60 \text{ seg} \cdot 1000 = 60000 \text{ ms}</math></p> <p>0,5 vueltas <math>\rightarrow x = 3,33 \text{ ms}</math></p>	
<p>Transferencia:</p> <p>10 MiB <math>\rightarrow 1 \text{ seg}</math></p> <p>512 bytes <math>\rightarrow x</math></p> <p>10 MiB <math>\cdot 2^{20} = 10.485.760 \text{ bytes} \rightarrow 1 \text{ seg} \cdot 1000 = 1000 \text{ ms}</math></p> <p>512 bytes <math>\rightarrow x = 0,04 \text{ ms}</math></p>	
<p>Cantidad de Bloques (c):</p> <p>15 MiB <math>\cdot 2^{20} = 15.728.640 \text{ bytes}</math></p> <p>15.728.640 bytes <math>\mid 512 \text{ bytes} = 30720</math></p>	<p>Cantidad de Bloques (d):</p> <p>16 MiB <math>\cdot 2^{20} = 16.777.216 \text{ bytes}</math></p> <p>16.777.216 bytes <math>\mid 512 \text{ bytes} = 32768</math></p>
<p>Almacenamiento secuencial:</p> <p><math>10 \text{ ms} + 3,33 \text{ ms} + 0,04 \text{ ms} \cdot 30720 = 124,43 \text{ ms}</math></p>	<p>Almacenamiento aleatorio:</p> <p><math>(10 \text{ ms} + 3,33 \text{ ms} + 0,04 \text{ ms}) \cdot 32768 = 438.108,16 \text{ ms}</math></p>

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)

Solución en la carpeta de diagramas de la Práctica-6.

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

Sí, el algoritmo SSTF podría generar inanición.

15. Supongamos un Head con movimiento en 300 pistas (numerados de 0 a 299), que está en la pista 143 atendiendo un requerimiento y anteriormente atendió un requerimiento en la pista 125.

Si la cola de requerimientos es: 126, 147, 81, 277, 94, 150, 212, 175, 140, 225, 280, 50, 99, 118, 22, 55; y después de 30 movimientos se incorporan los requerimientos de las pistas 75, 115, 220 y 266. 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

**(b)** SSTF

**(c)** Scan

**(d)** Look

**(e)** C-Scan

**(f)** C-Look

Solución en la carpeta de diagramas de la Práctica-6.

16. Supongamos un Head con movimiento en 300 pistas (numerados de 0 a 299), que está en la pista 140 atendiendo un requerimiento y anteriormente atendió un requerimiento en la pista 135.

Si la cola de requerimientos es: 99, 110, 42, 25, 186, 270, 50, 99, 147<sup>PF</sup>, 81, 257, 94, 133, 212, 175, 130; y después de 30 movimientos se incorporan los requerimientos de las pistas 85, 15<sup>PF</sup>, 202 y 288; y después de otros 40 movimientos más se incorporan los requerimientos de las pistas 75, 149<sup>PF</sup>, 285, 201 y 59. 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
- (b) SSTF
- (c) C-Scan
- (d) Scan
- (e) Look
- (f) C-Look

Solución en la carpeta de diagramas de la Práctica-6.

### Administración de Archivos

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

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

(a) Describa como trabaja cada uno.

(b) Cite ventajas y desventajas de cada uno.

Asignación Contigua: con esta asignación, un archivo se almacena en bloques de manera contigua en el disco, es decir, todos los bloques del archivo se ubican uno al lado del otro en el espacio de almacenamiento. Este es un método que facilita el acceso secuencial al archivo ya que los bloques están contiguos en memoria. Sin embargo, puede haber problemas de fragmentación y asignación de espacio que pueden resultar en uso ineficiente del espacio en disco.

Asignación Enlazada: un archivo se divide en bloques que pueden estar dispersos en todo el disco. Cada bloque contiene un puntero al siguiente bloque en secuencia, formando una cadena enlazada. El último bloque de la cadena tiene un puntero especial que indica el final del archivo. Este método evita problemas de fragmentación, pero puede tener un impacto en el tiempo de acceso ya que la lectura o escritura secuencial puede requerir múltiples búsquedas de punteros.

Asignación Indexada: en la asignación indexada, se utiliza un bloque especial llamado bloque de índice. Este bloque contiene un conjunto de punteros que apuntan a bloques de datos del archivo. Cada entrada en el bloque de índice corresponde a un fragmento específico del archivo. Este enfoque combina ventajas de la asignación contigua y enlazada, ya que permite acceso rápido a bloques específicos y también facilita el crecimiento y la contracción del archivo sin fragmentación excesiva.

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

- *Tabla de bits*
- *Lista Ligada*
- *Agrupamiento*
- *Recuento*

(a) Describa como trabajan estos métodos.

(b) Cite ventajas y desventajas de cada uno.

**Tabla de Bits:** en este método, se utiliza una tabla de bits para representar cada bloque en el disco. Cada bit en la tabla corresponde a un bloque y su valor indica si el bloque está ocupado (1) o libre (0). Por lo tanto, la tabla de bits proporciona una representación eficiente del estado de ocupación del espacio en el disco. El tamaño de la tabla de bits depende del número total de bloques en el disco.

**Lista Ligada:** se utiliza una lista ligada para llevar un registro de los bloques libres. Cada entrada en la lista ligada apunta al siguiente bloque libre. Cuando se necesita asignar espacio, se toma el primer bloque de la lista. Cuando un bloque se libera, se agrega a la lista y normalmente se reorganiza la misma para mantener un orden coherente.

**Agrupamiento:** en la gestión de espacio libre mediante agrupamiento, los bloques contiguos libres se agrupan en conjuntos o grupos. La información sobre la disponibilidad de espacio se mantiene mediante una estructura de control que indica cuántos bloques libres hay en cada grupo. Esto ayuda a reducir la fragmentación y mejora la eficiencia en comparación con algunos otros métodos.

**Recuento:** se mantiene un contador que registra el número total de bloques libres en el disco. No se utiliza una estructura adicional como una tabla de bits o una lista ligada. Cada vez que se asigna o libera un bloque, se actualiza el contador en consecuencia. Aunque es simple, puede tener un rendimiento subóptimo en términos de acceso aleatorio y puede ser más adecuado para sistemas con una gestión de espacio de disco más simple.

#### 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.

Solución en la carpeta de diagramas de la Práctica-6.

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?

Si cada bloque es de 1 KiB  $\rightarrow 1 \cdot 2^{10} = 1024$  bits y las direcciones son de 32 bits  $\rightarrow 1024 / 32 = 32$  direcciones por bloque.

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

$$\begin{aligned}
 & 1 \text{ KiB} \cdot 10 + 32 \cdot 1 \text{ KiB} + 32^2 \cdot 1 \text{ KiB} + 32^3 \cdot 1 \text{ KiB} = \\
 & = 10 \text{ KiB} + 32 \text{ KiB} + 1024 \text{ KiB} + 32.768 \text{ KiB} = \\
 & = 33.824 \text{ KiB}
 \end{aligned}$$



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

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

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

**(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?

Inodo 8 con 8 inodos por bloque:

$$\begin{aligned} nroBloque &= \left( \frac{(8 - 1)}{8} \right) + 2 \\ nroBloque &= \frac{7}{8} + 2 \\ nroBloque &= 2,87 \end{aligned}$$

Inodo 8 con 16 inodos por bloque:

$$\begin{aligned} nroBloque &= \left( \frac{(8 - 1)}{16} \right) + 2 \\ nroBloque &= \frac{7}{16} + 2 \\ nroBloque &= 2,43 \end{aligned}$$

Inodo 9 con 8 inodos por bloque:

$$\begin{aligned} nroBloque &= \left( \frac{(9 - 1)}{8} \right) + 2 \\ nroBloque &= \frac{8}{8} + 2 \\ nroBloque &= 3 \end{aligned}$$

Inodo 9 con 16 inodos por bloque:

$$\begin{aligned} nroBloque &= \left( \frac{(9 - 1)}{16} \right) + 2 \\ nroBloque &= \frac{8}{16} + 2 \\ nroBloque &= 2,5 \end{aligned}$$

**(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?

Inodo 6 (Desplazamiento del mismo en el bloque):

$$\begin{aligned} \text{desplazamientoEnElBloque} &= ((6 - 1) \text{ MOD } 8) \cdot 64 \\ \text{desplazamientoEnElBloque} &= (5 \text{ MOD } 8) \cdot 64 \\ \text{desplazamientoEnElBloque} &= 5 \cdot 64 \\ \text{desplazamientoEnElBloque} &= 320 \end{aligned}$$

Inodo 8 (Desplazamiento del mismo en el bloque):

$$\begin{aligned} \text{desplazamientoEnElBloque} &= ((8 - 1) \text{ MOD } 24) \cdot 128 \\ \text{desplazamientoEnElBloque} &= (7 \text{ MOD } 8) \cdot 128 \\ \text{desplazamientoEnElBloque} &= 7 \cdot 128 \\ \text{desplazamientoEnElBloque} &= 896 \end{aligned}$$

### Laboratorio de Entrada-Salida

21. Instale e investigue para que sirve el siguiente programa `hdparm`

```
sudo apt-get install hdparm
```

Es una herramienta que permite configurar y analizar discos duros ATA/IDE y SATA. Proporciona información sobre el hardware y permite realizar pruebas de rendimiento.

22. Ahora ejecute el siguiente comando:

```
sudo hdparm -l /dev/sda
```

¿Cuántos cilindros, cabezas y sectores tiene su disco?

¿Qué pasa si ejecuta esto en un disco de estado sólido?

Estos datos corresponden a la geometría lógica del disco.

En discos mecánicos, representan cómo están organizados los datos físicamente. En discos de estado sólido (SSD), este concepto no aplica de la misma manera ya que los SSD no tienen partes móviles; el comando puede devolver valores genéricos.

23. Ahora ejecute el siguiente comando varias veces (al menos 5), de manera tal de poder calcular el promedio de resultados obtenidos

```
sudo hdparm /dev/sda
```

Prueba 1: 120.3 MB/s

Prueba 2: 119.8 MB/s

Prueba 3: 121.0 MB/s

Prueba 4: 120.5 MB/s

Prueba 5: 120.1 MB/s

$$\text{Promedio} = \frac{120.3 + 119.8 + 121.0 + 120.5 + 120.1}{5} = 120.34 \text{ MB/s}$$

24. Ahora ejecútelo de la siguiente manera:

```
hdparm -t --direct --offset 500 /dev/sda
```

25. *¿Para qué sirven los parámetros?:*

- *direct*
- *offset*

--direct: Ejecuta la prueba evitando el caché del sistema, usando acceso directo al disco.

--offset: Define el punto de inicio para la prueba (en bloques). En este caso, empieza 500 bloques después del inicio del disco.

26. *Compare los tiempos promedios obtenidos con los parámetros direct y offset y sin ellos (Recuerde comparar tiempos promedio y no ejecuciones aisladas).*

Sin --direct, los resultados pueden ser más rápidos debido al uso del caché del sistema.

Con --direct, los resultados son más reales, pero pueden ser más lentos porque se accede directamente al disco.

27. *¿Con que concepto de la teoría asocia el parámetro direct?*

Este parámetro se asocia con el acceso directo al disco, evitando cualquier tipo de optimización del sistema operativo (caché de disco o buffers). Este concepto se relaciona con la gestión de entrada/salida directa (Direct I/O) en sistemas operativos.

28. *Ejecute el siguiente comando:*

```
ls /sys/block/
```

29. *A que le parece que corresponde cada entrada del directorio anterior?*

Las entradas corresponden a los dispositivos de bloques (discos y particiones) detectados por el sistema, como discos duros (sda, sdb) y unidades virtuales.

30. *Ejecute el siguiente comando (ajuste el dispositivo de disco según su equipo):*

```
cat /sys/block/sda/queue/scheduler
```

31. *Investigue el resultado del comando anterior. ¿Qué quiere decir cada item del resultado?, investigue cada uno de ellos y asocio con conceptos de la teoría y de esta práctica. ¿Cuál es la diferencia entre los siguientes conceptos?*

- *noop*
- *deadline*
- *cfq*
- *Anticipatory IO*

Significado de los resultados:

- *noop*: Un planificador simple que ejecuta las solicitudes en el orden en que llegan. Ideal para SSD.
- *deadline*: Prioriza las solicitudes según su vencimiento para evitar el bloqueo de I/O.
- *cfq* (Completely Fair Queueing): Intenta dividir el tiempo de I/O equitativamente entre los procesos.
- *Anticipatory I/O* (teórico): Intenta predecir las solicitudes futuras para optimizar el acceso al disco (ya no se usa ampliamente).

Impacto en el rendimiento: Los SSD suelen beneficiarse de noop debido a su acceso directo, mientras que los discos mecánicos pueden aprovechar deadline o cfq.

32. Como root, ejecute el siguiente comando

```
sudo echo "noop" > /sys/block/sda/queue/scheduler
```

¿Cuál es el efecto de esto? Ayuda: vuelva a ejecutar

```
cat /sys/block/sda/queue/scheduler
```

Cambia el planificador a noop

33. Ahora ejecute el siguiente programa al menos 5 veces, de manera tal de poder calcular el promedio del resultado obtenido

```
hdparm -t --direct --offset 500 /dev/sda
```

34. Ahora del mismo modo repita el paso con las demás opciones obtenidas en el ejercicio 29 y compare los resultados

¿Cual le parece que debería ser más óptimo? ¿Por qué?

Óptimo: Para discos SSD, generalmente noop es mejor porque no requiere ordenar operaciones de I/O, mientras que para discos mecánicos deadline o cfq pueden mejorar el rendimiento.