

Introducción a los

Sistemas

Operativos

**Objetivo**

Práctica 6

El objetivo de esta práctica es que el alumno comprenda los aspectos principales acerca de la planificación de E/S y de discos.

# Administración de E/S

1. Dispositivos
   1. Los dispositivos, según la forma de transferir los datos, se pueden clasificar en 2 tipos:  Orientados a bloques

 Orientados a flujos

Describa las diferencias entre ambos tipos.

* 1. Cite ejemplos de dispositivos de ambos tipos.
  2. Enuncie las diferencias que existen entre los dispositivos de E/S y que el SO debe considerar.

**(a) Diferencias:**

**Transferencia de Datos:**

* + **Orientados a Bloques:** Transfieren datos en bloques fijos de tamaño predeterminado. Los datos se leen o escriben en unidades de bloques completos.
  + **Orientados a Flujos:** Transfieren datos de manera continua, sin la necesidad de dividirlos en bloques fijos. La lectura y escritura son más flexibles y pueden ocurrir byte por byte

**Acceso Secuencial vs. Aleatorio:**

* + **Orientados a Bloques:** Suelen permitir el acceso aleatorio, ya que los bloques pueden ser direccionados individualmente.
  + **Orientados a Flujos:** Generalmente, el acceso es secuencial, y la lectura o escritura ocurre de manera progresiva.

**Latencia y Eficiencia:**

* + **Orientados a Bloques:** Pueden ser más eficientes para operaciones que requieren acceso aleatorio y manipulación de datos en bloque.
  + **Orientados a Flujos:** Son más eficientes para operaciones secuenciales y transmisiones de datos continuas.

**(b) Ejemplos:**

**Dispositivos Orientados a Bloques:**

* + Disco duro
  + Memoria USB
  + Tarjeta de memoria SD

**Dispositivos Orientados a Flujos:**

* + Teclado
  + Ratón
  + Impresora

**(c) Diferencias entre Dispositivos de E/S:**

**Velocidad de Acceso:**

* + Algunos dispositivos son más rápidos que otros en términos de acceso y transferencia de datos.

**Método de Transferencia:**

* + Algunos dispositivos utilizan interrupciones para notificar al sistema operativo sobre eventos, mientras que otros pueden utilizar el acceso directo a memoria (DMA) para transferencias eficientes sin intervención de la CPU.

**Unidades de Transferencia:**

* + Diferencias en la unidad básica de transferencia, como bytes, palabras o bloques.

**Dispositivos de Carácter vs. Dispositivos de Bloque:**

* + Dispositivos de carácter transmiten datos uno a la vez (por ejemplo, teclado), mientras que dispositivos de bloque manejan datos en bloques más grandes (por ejemplo, disco duro).

**Naturaleza Temporal:**

* + Algunos dispositivos son temporales y volátiles (por ejemplo, RAM), mientras que otros almacenan datos de forma más permanente (por ejemplo, disco duro).

**Capacidad de Almacenamiento:**

* + La capacidad de almacenamiento varía significativamente entre diferentes dispositivos, desde pequeñas unidades como teclados hasta grandes discos duros o servidores de almacenamiento.

**Sincronización y Control:**

* + Algunos dispositivos requieren sincronización y control preciso (por ejemplo, impresoras), mientras que otros pueden operar de manera más asincrónica.

1. Técnicas de E/S Describa 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 la transferencia de datos entre dispositivos de E/S y memoria.
* Simple pero puede ser intensiva en tiempo de CPU.
* Adecuada para dispositivos lentos o transferencias de datos pequeñas.

**E/S Dirigida por Interrupciones:**

* La CPU se ocupa de otras tareas y es interrumpida cuando el dispositivo de E/S necesita atención.
* Mayor eficiencia de la CPU al no estar constantemente involucrada en operaciones de E/S.

**DMA (Acceso Directo a Memoria):**

* Un controlador DMA maneja la transferencia directa de datos entre dispositivos de E/S y memoria sin intervención continua de la CPU.
* Mejora significativamente la eficiencia de la CPU, especialmente en operaciones con grandes cantidades de datos.

1. La tecnica de E/S programa puede trabajar de dos formas:  E/S mapeada

 E/S aislada

Indique como trabajan estas 2 técnicas.

**E/S Mapeada:** En la técnica de E/S mapeada, los registros de control de los dispositivos de E/S se mapean directamente en el espacio de direcciones del procesador, como si fueran direcciones de memoria. Los programas de E/S acceden a estos registros de control utilizando instrucciones de carga y almacenamiento normales. Esto permite que la CPU interactúe directamente con los dispositivos de E/S mediante operaciones de lectura y escritura en las direcciones de memoria específicas asignadas a los registros de control de E/S.

**E/S Aislada:** En la técnica de E/S aislada, se utiliza un conjunto de instrucciones especializado para realizar operaciones de E/S. En lugar de mapear los registros de control de E/S directamente en el espacio de direcciones del procesador, se asigna un conjunto de instrucciones dedicadas para operaciones de E/S. Estas instrucciones permiten que el programa de E/S transfiera datos entre la memoria y los dispositivos de E/S sin interferir directamente con las direcciones de memoria normales utilizadas por el programa principal.

1. Enuncie las metas que debe perseguir un SO para la administración de la entrada salida.
2. **Eficiencia:** Optimizar el uso de recursos para garantizar una administración eficiente de operaciones de entrada/salida.
3. **Transparencia:** Ocultar la complejidad de los dispositivos de entrada/salida para que los programas no tengan que preocuparse por detalles específicos del hardware.
4. **Equidad:** Asignar de manera justa y equitativa los recursos de entrada/salida entre los diferentes procesos para evitar monopolización.
5. **Manejo de Dispositivos Diversos:** Ofrecer soporte para una variedad de dispositivos de entrada/salida, independientemente de su naturaleza y velocidad.
6. **Confiabilidad:** Garantizar la confiabilidad en las operaciones de entrada/salida, minimizando errores y pérdida de datos.
7. **Sincronización:** Coordinar eficientemente las operaciones de entrada/salida con el resto del sistema para mantener la coherencia de los datos.
8. **Priorización:** Permitir la priorización de operaciones de entrada/salida según la importancia y los requisitos de tiempo de los procesos.
9. **Seguridad:** Implementar mecanismos de seguridad para proteger los datos de entrada/salida y prevenir accesos no autorizados.
10. Drivers
    1. ¿Qué son?
    2. ¿Qué funciones mínimas deben proveer?
    3. ¿Quién determina cuales deben ser estas funciones?

**(a)** Los drivers, o controladores, son programas de software que actúan como interfaces entre el sistema operativo y los dispositivos de hardware. Su función principal es permitir la comunicación y la interacción eficiente entre el software del sistema y el hardware del dispositivo. Los drivers son esenciales para que el sistema operativo y las aplicaciones puedan utilizar y controlar hardware específico.

**(b) ¿Qué funciones mínimas deben proveer?:** Las funciones mínimas que deben proveer los drivers incluyen:

1. **Inicialización y Configuración:** Preparar el hardware para su uso, configurando los registros y estableciendo el estado inicial necesario.
2. **Transferencia de Datos:** Facilitar la transferencia de datos entre la memoria del sistema y el dispositivo de hardware. Esto implica la gestión de operaciones de lectura y escritura.
3. **Manejo de Interrupciones:** Si el hardware utiliza interrupciones para notificar eventos al sistema, el driver debe manejar estas interrupciones de manera adecuada.
4. **Control de Estado:** Proporcionar funciones para consultar y cambiar el estado del dispositivo, como encender o apagar, cambiar configuraciones, etc.
5. **Manejo de Errores:** Detectar y gestionar errores que puedan ocurrir durante la operación del dispositivo, informando al sistema operativo o aplicaciones sobre posibles problemas.
6. **Compatibilidad con Estándares:** Cumplir con estándares de la industria para garantizar la interoperabilidad y la compatibilidad con diferentes sistemas y aplicaciones.

**(c) ¿Determinación de funciones?:**  
Las funciones son determinadas por estándares de la industria, especificaciones de hardware y requisitos del sistema operativo, en colaboración con fabricantes de hardware y desarrolladores de software.

1. Realice un grafico que marque la relación entre el Subsistema de E/S, los drivers, los con- troladores de dispositivos y los dispositivos.

Imagen que contiene Texto

Descripción generada automáticamente

1. 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.

**Generación del Requerimiento de E/S:** Word necesita imprimir un documento.

**Solicitud al Sistema Operativo:** El proceso envía una solicitud al sistema operativo para imprimir.

**Intervención del Sistema Operativo:** Verifica la disponibilidad de la impresora y asigna el recurso.

**Acceso al Controlador de Dispositivos:** El sistema operativo utiliza el controlador de la impresora.

**Transferencia de Datos al Controlador:** Se transfieren los datos del documento al controlador.

**Envío de Datos al Dispositivo:** Los datos se envían a la impresora a través del controlador.

**Procesamiento en el Dispositivo:** La impresora procesa los datos y realiza la tarea de impresión.

**Finalización de la Operación:** El controlador notifica al sistema operativo y libera el recurso.

1. Enuncie que servicios provee el SO para la administración de E/S.

Los servicios que ofrece el S.O. son:

1. **Control de Dispositivos:**
   * Gestión de controladores de dispositivos para facilitar la comunicación entre el sistema operativo y el hardware.
2. **Buffering:**
   * Almacenamiento temporal de datos en búferes para mejorar la eficiencia en la transferencia entre dispositivos y memoria.
3. **Caché:**
   * Implementación de cachés para reducir los tiempos de acceso a dispositivos y mejorar el rendimiento general del sistema.
4. **Spooling:**
   * Utilización de spoolers para encolar tareas de impresión, permitiendo que múltiples procesos envíen trabajos de impresión sin interferir entre sí.
5. **Planificación de E/S:**
   * Planificación y coordinación de operaciones de entrada/salida para maximizar la eficiencia y minimizar los tiempos de espera.
6. **Manejo de Interrupciones:**
   * Atención a interrupciones generadas por dispositivos para gestionar eventos de E/S de manera asincrónica.
7. **Control de Acceso a Dispositivos:**
   * Establecimiento de mecanismos para garantizar la exclusividad de acceso a ciertos dispositivos, evitando conflictos.
8. **Establecimiento de Conexiones:**
   * Gestión de conexiones para dispositivos que requieren configuraciones específicas antes de la operación, como la inicialización de puertos de comunicación.
9. **Manejo de Errores:**
   * Detección y manejo de errores durante las operaciones de E/S para garantizar la integridad de los datos y la estabilidad del sistema.
10. **Sincronización:**
    * Coordinación de operaciones de E/S y sincronización con otras partes del sistema para mantener la coherencia de los datos.
11. **Virtualización de Dispositivos:**
    * Ofrecimiento de interfaces de dispositivos virtuales para permitir una mayor flexibilidad y portabilidad de aplicaciones.

En resumen, El sistema operativo provee servicios esenciales para la administración eficiente de entrada/salida (E/S), incluyendo control de dispositivos, buffering, caché, spooling, planificación de E/S, manejo de interrupciones, control de acceso, establecimiento de conexiones, manejo de errores, sincronización y virtualización de dispositivos. Estos servicios mejoran la eficiencia, confiabilidad y coordinación de las operaciones E/S en el sistema.

# Administración de Discos

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

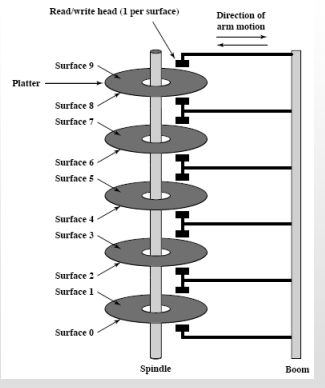
La organización física de un disco se compone de varios platos (discos) circulares que giran alrededor de un eje central en una unidad llamada huso. Cada plato tiene dos caras (No necesariamente se usan ambas), y las caras de todos los platos se dividen en pistas concéntricas. Cada pista, a su vez, se divide en sectores. Los datos se almacenan en estos sectores, que son la unidad básica de lectura o escritura.

Un brazo de lectura/escritura está asociado con cada superficie del plato y se mueve hacia el centro o hacia el borde del plato para acceder a diferentes pistas. A todas las pistas que están en la misma posición vertical en cada plato del disco duro se les llama cilindro y un cluster es la suma de uno o mas sectores (Es la unidad mínima de almacenamiento en el disco) La posición exacta del cabezal de lectura/escritura se controla mediante un actuador.

El disco gira a alta velocidad, y el acceso a los datos se realiza mediante la lectura o escritura en los sectores mientras el brazo se posiciona en la pista adecuada. La separación entre pistas, llamada espacio inter-pista, y la separación entre sectores, llamada espacio inter-sector, son aspectos críticos para garantizar la integridad y eficiencia del acceso a los datos almacenados en el disco.

Diagrama

Descripción generada automáticamente

Gráfico, Gráfico de proyección solar

Descripción generada automáticamente

1. La velocidad promedio para la obtención de datos de un disco esta 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

1. 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 total del disco.
  2. ¿Cuantos sectores ocuparía un archivo de tamaño de 3 MiB(Mebibytes)?
  3. 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)
  4. Calcule el tiempo de transferencia real de un archivo de 16 MiB(Mebibytes). grabado en el disco de manera aleatoria.

Texto, Carta

Descripción generada automáticamente

Caras: 14 ; Pistas: 1100 ;Sectores por pista: 300 ; Tamaño sector: 512bytes

Capacidad Total=7×2×1100×300×512 = 2365440000 bytes (2.2 GiB)

* 1. Sectores: Tamaño archivo/ Tamaño sector

3 MiB = 3\*2^20bytes

Sectores: 3\*2^20bytes /512 bytes = 6144 s.

* 1. Almacenamiento secuencial: seek + latency + (T. transf. bloque \* #bloques)

Latency = Tiempo de media vuelta = 0.5/RPM = 55x10-6 m =3.3x10-3 s

Seek: 10ms = 0.01 s

TtransfBloque= 10 MiB/s

Cant Bloques= 2

Almacenamiento secuencial: 0.01 s + 3.3x10-3 s+ (10 \* 2) = 20.01 s.

* 1. Almacenamiento aleatorio: (seek + latency + t. transfbloque) \* #bloques

Latency = Tiempo de media vuelta = 0.5/RPM = 55x10-6 m =3.3x10-3 s

Seek: 10ms = 0.01 s

TtransfBloque= 10 MiB/s

Cant Bloques= 2

Almacenamiento secuencial: (0.01 s + 3.3x10-3 s+ 10 )\* 2) = 20.02s.

1. 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:

* 1. FCFS (*First Come, First Served*)
  2. SSTF (*Shortest Seek Time First*)
  3. Scan
  4. Look
  5. C-Scan (*Circular Scan*)
  6. C-Look (*Circular Look*)

Nota: Es 199 el fin, no 200

Diagrama

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente

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

SSTF: Puede causar inanición para los requerimientos que están lejos del cabezal si constantemente llegan requerimientos cercanos.

1. Supongamos un Head con movimiento en 300 pistas (numerados de 0 a 299), que esta 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:

* 1. FCFS
  2. SSTF
  3. Scan
  4. Look
  5. C-Scan
  6. C-Look

Diagrama

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente con confianza media

Diagrama

Descripción generada automáticamente con confianza media

Imagen en blanco y negro

Descripción generada automáticamente con confianza baja

Un mapa de color blanco

Descripción generada automáticamente con confianza baja

Gráfico

Descripción generada automáticamente

1. Supongamos un Head con movimiento en 300 pistas (numerados de 0 a 299), que esta 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*P F* , 81, 257,

Diagrama

Descripción generada automáticamente94, 133, 212, 175, 130; y después de 30 movimientos se incorporan los requerimientos de las pistas 85, 15*P F* , 202 y 288; y después de otros 40 movimientos más se incorporan los requerimientos de las pistas 75, 149*P F* , 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:

* 1. FCFS
  2. SSTF
  3. C-Scan
  4. Scan
  5. Look
  6. C-Look

Diagrama

Descripción generada automáticamente con confianza media

Diagrama

Descripción generada automáticamente

Imagen que contiene Diagrama

Descripción generada automáticamente

Imagen que contiene texto, pizarrón, tabla, parado

Descripción generada automáticamenteDiagrama

Descripción generada automáticamente

# Administración de Archivos

1. Dados los siguientes métodos de administración de espació de un archivo:  Asignación contigua

 Asignación enlazada  Asignación indexada

* 1. Describa como trabaja cada uno.
  2. Cite ventajas y desventajas de cada uno.

a)

**Asignación Contigua:**

* + **Descripción:** El archivo se almacena de manera contigua en el disco.
  + **Funcionamiento:** Un solo bloque de espacio continuo se asigna al archivo. La ubicación y tamaño se registran en una tabla.

**Asignación Enlazada:**

* + **Descripción:** El archivo se divide en bloques con punteros al siguiente.
  + **Funcionamiento:** Los bloques se asignan dinámicamente. Cada bloque tiene un puntero al siguiente, con el último siendo nulo.

**Asignación Indexada:**

* + **Descripción:** Se utiliza un bloque de índice con punteros a bloques de datos.
  + **Funcionamiento:** Los datos se dividen en bloques y un bloque de índice mantiene punteros a estos bloques, permitiendo el acceso directo.

**(b) Ventajas y Desventajas:**

**Asignación Contigua:**

* + **Ventajas:** Rápido acceso secuencial, implementación simple.
  + **Desventajas:** Problemas de fragmentación, ineficiente para inserción/eliminación.

**Asignación Enlazada:**

* + **Ventajas:** Eficiente para archivos de tamaño variable, sin fragmentación interna.
  + **Desventajas:** Mayor tiempo de acceso, uso adicional de espacio por punteros.

**Asignación Indexada:**

* + **Ventajas:** Acceso aleatorio eficiente, sin fragmentación interna.
  + **Desventajas:** Requiere más espacio por el bloque de índice, implementación más compleja.

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

 Tabla de bits

 Lista Ligada

 Agrupamiento  Recuento

* 1. Describa como trabajan estos métodos.
  2. Cite ventajas y desventajas de cada uno.

a)

1. **Tabla de Bits:**
   * **Descripción:** En este método, se utiliza una tabla de bits para mantener un registro del estado de cada bloque en el disco (ocupado o libre).
   * **Funcionamiento:** Cada bloque en el disco tiene un bit correspondiente en la tabla de bits. Un bit "1" indica que el bloque está ocupado, y un bit "0" indica que está libre.
2. **Lista Ligada:**
   * **Descripción:** En este método, se mantiene una lista de bloques libres mediante punteros.
   * **Funcionamiento:** Cada bloque libre tiene un puntero al siguiente bloque libre, formando así una lista ligada de bloques libres. El último bloque de la lista tiene un puntero nulo.
3. **Agrupamiento (o Buddy System):**
   * **Descripción:** Este método agrupa bloques contiguos de manera que los bloques libres tienen tamaños predefinidos (potencias de 2) y se combinan para formar bloques más grandes si están contiguos.
   * **Funcionamiento:** Cuando un bloque se libera, se combina con bloques adyacentes de tamaño similar para formar bloques más grandes. El sistema utiliza una estructura de árbol binario para organizar y buscar bloques libres.
4. **Recuento:**
   * **Descripción:** En este método, se mantiene un contador global que indica el número de bloques libres en el disco.
   * **Funcionamiento:** Cuando se libera un bloque, el contador se incrementa, y cuando se asigna un bloque, se decrementa. El sistema realiza un seguimiento del espacio libre en términos de bloques, no en términos de bloques individuales.

**(b) Ventajas y Desventajas:**

1. **Tabla de Bits:**
   * **Ventajas:** Simple de implementar y entender. Acceso rápido a la información sobre el estado del bloque.
   * **Desventajas:** Ineficiente en términos de espacio para discos grandes debido al tamaño de la tabla de bits.
2. **Lista Ligada:**
   * **Ventajas:** Eficiente para discos grandes, ya que solo se requiere un puntero para el primer bloque libre.
   * **Desventajas:** Acceso más lento en comparación con la tabla de bits. Puede haber fragmentación de la lista.
3. **Agrupamiento (Buddy System):**
   * **Ventajas:** Minimiza la fragmentación y permite una asignación eficiente de bloques contiguos.
   * **Desventajas:** Complejidad en la implementación. Puede haber fragmentación interna.
4. **Recuento:**
   * **Ventajas:** Simplicidad en la implementación. Ofrece un conteo rápido del espacio libre total.
   * **Desventajas:** Menos eficiente para el seguimiento de bloques individuales. Puede haber fragmentación.
5. Gestión de archivos en *UNIX*.

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

Cada archivo o directorio esta representado por una estructura que mantiene, entre otra información, las direcciones de lo 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.

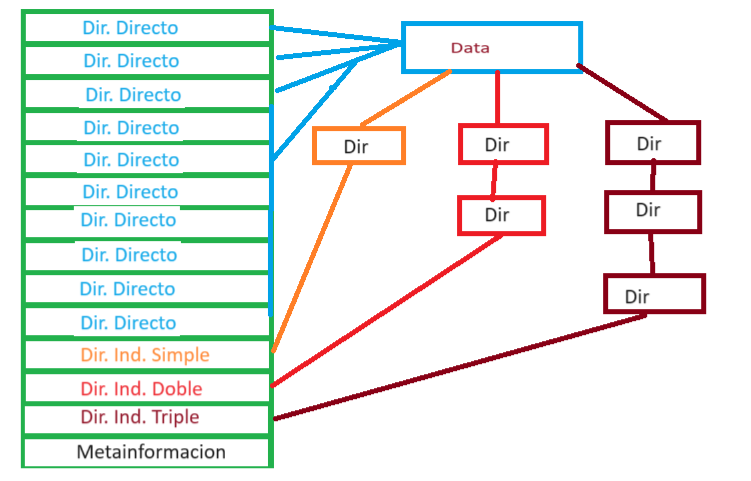
* 1. Realice un grafico que describa la estructura del *I-NODO* y de los bloques de datos. Cada bloque es de 1 Kib(Kibibits). Si cada dirección para referenciar un bloque es de 32 bits:
     1. ¿Cuántas referencias (direcciones) a bloque pueden contener un bloque de disco?
     2. ¿Cuál sería el tamaño máximo de un archivo?

a)

Diagrama

Descripción generada automáticamente

En mi estructura particular quedaría así:



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

1 Kib ÷ 32 bits = 32 direcciones por bloque.

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

10 x 1 Kib (DD) + 1 x 32 x 1 Kib (DIS) + 1 x 32 x 32 x 1 Kib (DID) + 1 x 32 x 32 x 32 x 1 Kib (TID)

33834 Kib

1. 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.

* 1. Según la primer 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?

* 1. De acuerdo a la segunda fórmula, si cada inodo del disco ocupa 64 bytesy 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 bytesy 24 inodos por bloque: ¿dónde empezaría el inodo 8?

a)Para encontrar el inodo 8: Inodo=(8−1)/8+2=2.8 (Nro bloque =2)

El inodo 8 estaría en el bloque 2.

Para encontrar el inodo 9: Inodo=(9−1)/8+2=3 (Nro bloque =3)

El inodo 9 estaría en el bloque 3.

Si ahora consideramos bloques de disco con 16 inodos:

Para encontrar el inodo 8: Inodo=(8−1)/16+2=2.43 (Nro bloque =2)

El inodo 8 estaría en el bloque 2.

Para encontrar el inodo 9: Inodo=(9−1)/16+9= 2.5 (Nro bloque =3)

El inodo 9 estaría en el bloque 2 también.

**b)** Dada esta fórmula y asumiendo que cada inodo ocupa 64 bytes y hay 8 inodos por bloque:

Para encontrar el desplazamiento del inodo 8:

Desplazamiento=((8−1)mod8)×64=7×64=448

El inodo 8 comienza en el desplazamiento 448 del bloque de disco.

Para encontrar el inicio del inodo 6: Desplazamiento=((6−1)mod8)×64=320

El inodo 6 comienza en el desplazamiento 320 del bloque de disco.

Si ahora los inodos son de 128 bytes y hay 24 inodos por bloque:

Para encontrar el desplazamiento del inodo 8:

Desplazamiento=((8−1)mod24)×128=7×128=896

El inodo 8 comienza en el desplazamiento 896 del bloque de disco.

# Laboratorio de Entrada-Salida

Se recomienda resolver este laboratorio con el Sistema en modo consola y con la menor cantidad de programas en ejecución posible.

1. Instale e investigue para que sirve el siguiente programa hdparm

sudo apt-get install hdparm

**hdparm** es una utilidad de línea de comandos en sistemas operativos basados en Unix, como Linux y algunas versiones de Unix, que se utiliza para interactuar con y controlar características de discos duros (HDD) y unidades de estado sólido (SSD). Aquí hay algunas de las funciones comunes que **hdparm** puede realizar:

1. **Obtención de Información del Disco:**
   * **hdparm -I /dev/sdX**: Muestra información detallada sobre el disco, incluyendo la capacidad, el modelo, la velocidad de transferencia y características específicas del disco.
2. **Medición y Optimización de la Velocidad de Transferencia:**
   * **hdparm -t /dev/sdX**: Realiza una prueba de lectura secuencial para medir la velocidad de transferencia del disco.
   * **hdparm -T /dev/sdX**: Realiza una prueba de caché para medir la velocidad de transferencia desde la caché del sistema.
3. **Configuración de Parámetros del Disco:**
   * **hdparm -A1 /dev/sdX**: Activa o desactiva la característica de ajuste automático de energía.
   * **hdparm -S <valor> /dev/sdX**: Establece el tiempo de inactividad antes de que el disco entre en modo de bajo consumo (spindown).
4. **Configuración de Modo de UDMA:**
   * **hdparm -XudmaX /dev/sdX**: Configura el modo UDMA (Ultra DMA) del disco.
5. **Desactivación de la Caché de Escritura:**
   * **hdparm -W0 /dev/sdX**: Desactiva la caché de escritura para mejorar la seguridad contra pérdida de datos en caso de falla de energía.
6. Ahora ejecute el siguiente comando:

sudo hdparm -I /dev/sda

 ¿Cuando cilindros, cabezas y sectores tiene su disco?

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

Imagen que contiene Tabla

Descripción generada automáticamenteLos discos de estado sólido no tienen cilindros, cabezas ni sectores físicos en el mismo sentido que los discos duros tradicionales. Estos valores son más relevantes para discos magnéticos. En un SSD, el comando puede devolver información diferente, como la capacidad, la interfaz, la versión del firmware y otras características específicas de los SSD.

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

sudo hdparm -t /dev/sda

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media

1. Ahora ejecutelo de la siguiente manera:

hdparm -t --direct --offset 50 /dev/sda (El 500 me dice error)











Texto

Descripción generada automáticamente con confianza media

**direct:**

* **Descripción:** Este parámetro controla si las operaciones de lectura y escritura deben realizarse directamente sin pasar por la caché del sistema.
* **Valores:** Puede tener los valores **on** o **off**.
* **Uso:**
  + **direct=on**: Indica que las operaciones deben realizarse sin la caché del sistema, lo que puede ser útil para mediciones de rendimiento precisas.
  + **direct=off**: Permite que las operaciones utilicen la caché del sistema.
* **Ejemplo:** hdparm --direct=on /dev/sdX

**offset:**

* **Descripción:** Este parámetro permite especificar un desplazamiento (offset) para la operación de lectura o escritura, que indica dónde comenzar en el dispositivo.
* **Uso:**
  + **offset=N**: Donde **N** es el número de bloques desde el principio del dispositivo.
* **Ejemplo:** hdparm --offset=1024 /dev/sdX

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

T promedio 22. = 84,75 Mbi/seg

T promedio 23. = 985.24 Mbi/seg

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

El parámetro **direct** en el comando **hdparm** está asociado al concepto de **caché**. El uso de este parámetro controla si las operaciones de lectura y escritura deben realizarse directamente sin pasar por la caché del sistema.

1. Ejecute el siguiente comando:

ls /sys/block/



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

En el resultado del comando **ls /sys/block/**, las entradas corresponden a diferentes dispositivos de bloque presentes en el sistema. Aquí está una interpretación general basada en la salida que proporcionaste:

1. **loop0, loop1, ..., loop7:**
   * Estas entradas generalmente representan dispositivos de bucle (loop devices). Los dispositivos de bucle son utilizados para montar archivos como si fueran sistemas de archivos, y cada número indica un dispositivo de bucle diferente.
2. **sda:**
   * Esta entrada probablemente corresponde al primer disco SATA en el sistema. Los nombres de dispositivos de bloques para discos duros suelen comenzar con "sd" seguido de una letra que indica el orden del dispositivo.
3. **sr0:**
   * Esta entrada podría corresponder a un dispositivo de CD-ROM o DVD-ROM. Los dispositivos ópticos suelen ser denominados "sr" seguido de un número.
4. Ejecute el siguiente comando(ajuste el dispositivo de disco según su equipo):

cat /sys/block/sda/queue/scheduler



1. Investigue el resultado del comando anterior. ¿Que quiere decir cada item del resultado?, investigue cada uno de ellos y asocielo con conceptos de la teoría y de esta práctica. ¿Cual es la diferencia entre los siguientes conceptos?

**noop:**

**Descripción:** Algoritmo simple de planificación que ejecuta operaciones de entrada/salida en el orden en que se reciben, sin reordenamiento.

**Uso Común:** En entornos donde el reordenamiento de operaciones no es crítico. (FIFO)

**deadline:**

**Descripción:** Asigna plazos a las operaciones de entrada/salida, priorizando según el tiempo límite para minimizar el tiempo promedio de respuesta.

**Uso Común:** En sistemas que requieren equilibrio entre rendimiento y garantía de tiempo de respuesta. (Tipo SRTF)

**cfq (Completely Fair Queuing):**

**Descripción:** Asigna prioridades a procesos generadores de operaciones de entrada/salida, busca un reparto justo del tiempo de acceso al disco entre procesos.

**Uso Común:** En entornos donde la justicia en el acceso al disco es esencial. (Tipo R.R)

**Anticipatory IO:**

**Descripción:** Algoritmo antiguo que anticipaba la dirección de la cabeza de lectura/escritura del disco para realizar operaciones de E/S. (Quedo obsoleto)

**none:**

**Descripción:** noop actualized version (FIFO)Principio del formulario

Lo que quiere decir lo expuesto arriba es cual algoritmo scheduler esta seleccionado (En mi caso mq-deadline)

1. Como root, ejecute el siguiente comando

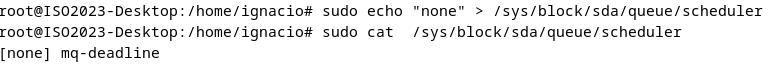
sudo echo "noop" > /sys/block/sda/queue/scheduler



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

cat /sys/block/sda/queue/scheduler

El efecto es cambiar el scheduler de I.O a noop (No lo tengo instalado, por eso el error)

 (Hecho con none)

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

hdparm -t --direct --offset 50 /dev/sda

Interfaz de usuario gráfica, Texto, Aplicación, Carta

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Carta

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Carta

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Carta

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Carta

Descripción generada automáticamente

Promedio = 904.49 Mbi/seg

1. Ahora del mismo modo repita el paso con las demas opciones obtenidas en el ejercicio 25 y compare los resultados

 ¿Cual le parece que debería ser mas óptimo?  ¿Porque?

Debería ser mas optimo none, ya que es un FIFO (No se pierde tiempo planificando que I.O se usa)