

Introducción a los Sistemas Operativos

Practica 6

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:

- **Modo de acceso:** Acceden a los datos en bloques de tamaño fijo.
- **Estructura de datos:** Permiten leer y escribir bloques individuales (p. ej., 512 bytes o 4 KB).
- **Acceso aleatorio:** Permiten acceder directamente a cualquier bloque.
- **Uso típico:** Son utilizados en dispositivos de almacenamiento.
- **Ejemplo de uso:** Leer/escribir sectores en un disco duro.

Orientados a flujos:

- **Modo de acceso:** Los datos se transfieren de manera secuencial (byte por byte o palabra por palabra).
- **Estructura de datos:** No tienen una estructura definida en bloques.
- **Acceso secuencial:** El acceso debe realizarse en orden, desde el principio hasta el final.
- **Uso típico:** Se emplean para la transmisión de datos en tiempo real.
- **Ejemplo de uso:** Comunicación a través de un puerto serie.

(b) Cite ejemplos de dispositivos de ambos tipos.

Orientados a bloques:

- Discos duros (HDD).
- Unidades SSD.
- Memorias USB.

Orientados a flujos:

- Teclados.
- Ratones.
- Puertos de comunicación serie (UART).

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

Modo de acceso:

- Algunos dispositivos permiten acceso aleatorio (p. ej., discos), mientras que otros solo acceso secuencial (p. ej., cintas magnéticas).

Velocidad de transferencia:

- Los dispositivos varían en velocidad; el S.O. debe gestionar la sincronización y los búferes para evitar cuellos de botella.

Mecanismo de comunicación:

- **Interrupciones:** Muchos dispositivos generan interrupciones para notificar al CPU.
- **Polling:** Algunos requieren que el S.O. los consulte periódicamente.

Semántica de datos:

- **E/S sincrónica:** El proceso espera hasta que la operación termine.
- **E/S asincrónica:** El proceso continúa mientras la operación se ejecuta en paralelo.

Formato de datos:

- Algunos dispositivos manejan datos en bruto (p. ej., discos duros), mientras que otros trabajan con datos estructurados o de alto nivel (p. ej., impresoras).

Persistencia:

- Los dispositivos de almacenamiento (p. ej., discos duros) son persistentes, mientras que otros (p. ej., pantallas) no almacenan datos.

Requerimientos de controladores:

- Cada dispositivo necesita un controlador específico que el S.O. debe gestionar para realizar operaciones de E/S.

2. Técnicas de E/S Describa cómo trabajan las siguientes técnicas de E/S

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

E/S Programada

- **Funcionamiento:**

- El **procesador controla** directamente la **transferencia de datos** entre el dispositivo de E/S y la memoria.
- El **CPU consulta continuamente al dispositivo** de E/S para **verificar si está listo** para leer o escribir datos (método de *polling*).
- Mientras el **CPU espera, no puede realizar otras tareas**, lo que genera ineficiencia.
- **Características:**
 - **Simplicidad en la implementación.**
 - Bajo rendimiento, ya que el CPU está ocupado monitoreando el dispositivo constantemente.
- **Uso típico:**
 - Dispositivos de baja velocidad donde la eficiencia no es crítica.

E/S Dirigida por Interrupciones

- **Funcionamiento:**
 - El **dispositivo de E/S genera una interrupción** para **notificar al CPU** cuando está **listo** para transferir datos.
 - El **CPU detiene temporalmente** su ejecución actual, atiende la solicitud de E/S (interrupción) y luego regresa a la tarea original.
 - No requiere que el CPU esté continuamente monitoreando al dispositivo.
- **Características:**
 - Mejora el rendimiento al **liberar al CPU para realizar otras tareas** mientras espera la interrupción.
 - Requiere un mecanismo eficiente para manejar interrupciones.
- **Uso típico:**
 - Sistemas multitarea y dispositivos de velocidad media.

DMA (Acceso Directo a Memoria)

- **Funcionamiento:**
 - El controlador **DMA maneja directamente la transferencia de datos entre la memoria y el dispositivo de E/S**, sin la intervención del CPU.
 - El **CPU configura** inicialmente el controlador **DMA**, especificando la dirección de memoria, el tamaño de los datos y el dispositivo de destino.
 - Una vez **configurado, el DMA realiza la transferencia** y **notifica al CPU** al completarla.
- **Características:**
 - **Muy eficiente**, ya que **libera al CPU de manejar las transferencias de datos**.
 - Permite transferencias de alta velocidad entre dispositivos y memoria.
 - Ideal para operaciones que involucran grandes cantidades de datos.
- **Uso típico:**
 - Transferencias de datos masivas, como las realizadas por discos duros, SSDs, y tarjetas de red.

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

- E/S mapeada
- E/S aislada

Indique cómo trabajan estas 2 técnicas

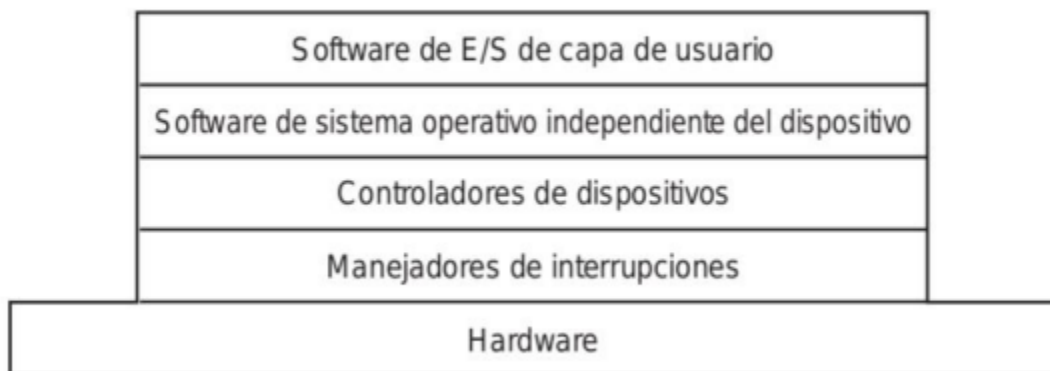
E/S Mapeada (Memory-Mapped I/O)

- **Funcionamiento:**
 - Las direcciones de los dispositivos de E/S se integran dentro del mismo espacio de direcciones de la memoria principal.
 - El procesador utiliza las mismas instrucciones (como LOAD y STORE) para acceder tanto a la memoria como a los dispositivos de E/S.
 - El controlador de E/S traduce las direcciones de memoria específicas a operaciones de E/S.
- **Características:**
 - Simplifica la programación al tratar los dispositivos de E/S como si fueran bloques de memoria.
 - Permite usar las mismas operaciones de memoria para E/S, evitando instrucciones específicas.
 - Puede ocasionar conflictos si no se gestiona adecuadamente el espacio de direcciones.
- **Ejemplo:**
 - Sistemas embebidos que interactúan con sensores o dispositivos conectados directamente.

E/S Aislada (Isolated I/O)

- **Funcionamiento:**
 - Los dispositivos de E/S tienen su propio espacio de direcciones independiente del espacio de direcciones de la memoria principal.
 - Se requieren instrucciones específicas de E/S (como IN y OUT) para interactuar con los dispositivos.
 - El procesador tiene un bus separado o controladores específicos para acceder a los dispositivos.
- **Características:**
 - Previene conflictos entre las direcciones de memoria y de E/S.
 - Puede hacer que el diseño del hardware sea más complejo, ya que requiere un bus dedicado para E/S.
 - Las instrucciones de E/S son diferentes de las operaciones de memoria, lo que puede complicar la programación.
- **Ejemplo:**
 - Sistemas que manejan dispositivos periféricos en arquitecturas como x86, donde se utiliza un espacio de direcciones de E/S separado.

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.

- Proceso → Solicitud de E/S.
- SO → Verifica y planifica.
- SO → Envío al controlador.
- Controlador → Activa dispositivo.
- Dispositivo → Transfiere datos (DMA/Interrupción).
- Dispositivo → Genera interrupción al SO.
- SO → Informa al proceso.

8. Es el 7 de nuevo?????

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

Abstracción de dispositivos: Proporciona una interfaz uniforme para acceder a diferentes dispositivos, ocultando los detalles de hardware.

Planificación de E/S: Administra el orden de las solicitudes de E/S para optimizar el uso del dispositivo.

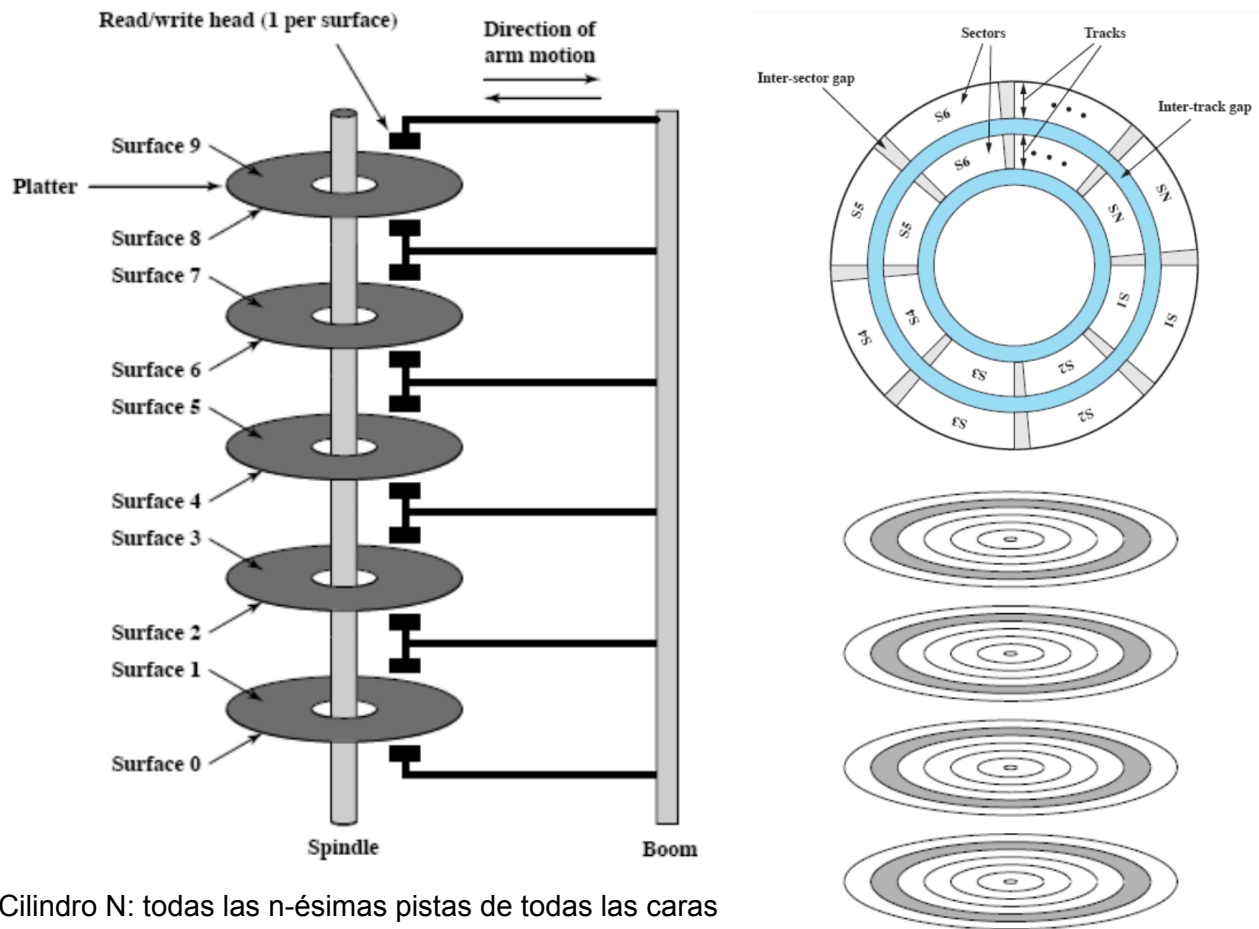
Buffering: Almacena temporalmente datos para sincronizar la velocidad entre dispositivos y procesos.

Gestión de controladores: Carga, inicializa y utiliza los controladores necesarios para interactuar con los dispositivos.

Manejo de interrupciones: Responde a señales de dispositivos para gestionar operaciones de E/S sin bloquear al procesador.

Administración de Discos

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



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

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 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.

$$\text{tamaño disco} = \# \text{caras} * \# \text{pistas cara} * \# \text{sectores pista} * \text{tamaño sector}$$

caras= 2*7

14* 1100*300*512 = tamaño disco

Tamaño disco=2365440000 bytes=2310000 kib= 2,255.859375 mib = 2.2029876708984375 gib

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

Capacidad de 1 sector:

512 bytes

• Dividimos el tamaño del archivo por la capacidad de una cara:

3 MiB = 3145728 bytes

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

9000 rpm

1 minuto = 60s= 60000ms

60000ms / 9000ms = 6.666ms

Media vuelta= 6.66ms /2= 3.33ms

latencia= 3.33ms

Transferencia

15miB(Mebibits)

1 segundo =1000ms

512 bytes

10miB*2^20= 10485760 miB

10485760miB/1000ms= 10485.76 miB/ms

Capacidad del sector/velocidad de transferencia= tiempo de transferencia bloque

512bytes/10485.76 bytes/ms= 0.048828125ms

Bloques

Cantidad de bytes/ capacidad de sector

15728640 bytes /512bytes=30720

$$\text{seek} + \text{latency} + \text{tiempo transferencia bloque} * \# \text{bloques}$$

10+ 3.33+ 0.048828125 *30720= 1,513.33

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

$$(\text{seek} + \text{latency} + \text{tiempo transferencia bloque}) * \#\text{bloques}$$

Bloques

Cantidad de bytes/ capacidad de sector = bloques

$$(2^{20} * 16) / 512 = 32768$$

$$\text{bloques} = 32768$$

$$\text{seek} = 10 \text{miB}$$

$$\text{Tiempo de transferencia Bloque} = 0.048828125 \text{ms}$$

$$\text{Latencia} = 3.33$$

$$(10 \text{ ms} + 3.33 \text{ms} + 0.048828125 \text{ms}) * 32768 = 438,397.44 \text{ms}$$

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)

Ta en papel nt

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

El SSTF porque deja demasiado tiempo esperando a los procesos

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

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, 147P F , 81, 257, 94, 133, 212, 175, 130; y después de 30 movimientos se incorporan los requerimientos de las pistas 85, 15P F , 202 y 288; y después de otros 40 movimientos más se incorporan los requerimientos de las pistas 75, 149P 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:

- (a) FCFS
- (b) SSTF
- (c) C-Scan
- (d) Scan
- (e) Look
- (f) C-Look

17. Métodos de administración de espacio

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

- **Cómo trabaja:** Los archivos se almacenan en un **bloque continuo de sectores** en el disco. Se requiere **conocer el tamaño del archivo** al momento de su creación para reservar el espacio necesario.

- **Ventajas:**
 - Acceso rápido y eficiente, especialmente para archivos secuenciales.
 - Fácil de implementar.
- **Desventajas:**
 - Problemas de fragmentación externa: espacios libres dispersos pueden impedir la asignación de nuevos archivos grandes.
 - Dificultad para extender archivos si no hay espacio continuo.

Asignación enlazada

- **Cómo trabaja:** Los bloques de un archivo están dispersos en el disco y cada bloque contiene un puntero al siguiente.
- **Ventajas:**
 - No hay fragmentación externa.
 - Fácil de extender archivos porque no requiere espacio contiguo.
- **Desventajas:**
 - Acceso lento para archivos grandes, ya que hay que seguir la cadena de punteros.
 - No es adecuada para acceso aleatorio, ya que requiere recorrer la lista desde el inicio.
 - Pérdida de espacio debido al almacenamiento de punteros.

Asignación indexada

- **Cómo trabaja:** Cada archivo tiene un bloque de índice que almacena punteros a todos los bloques de datos del archivo.
- **Ventajas:**
 - Acceso aleatorio eficiente, ya que los punteros están centralizados en el bloque de índice.
 - No hay problemas de fragmentación externa.
- **Desventajas:**
 - Requiere espacio adicional para almacenar el bloque de índice.
 - Puede haber fragmentación interna si el archivo es pequeño y los bloques no se llenan completamente.

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 cómo trabajan estos métodos.
(b) Cite ventajas y desventajas de cada uno.

Tabla de bits

- **Cómo trabaja:** Se utiliza un **mapa de bits** donde **cada bit representa un bloque del disco**. Un bit en **0** indica que el bloque está libre, y un bit en **1** indica que está ocupado.
- **Ventajas:**
 - Fácil de implementar y **compacta en términos de almacenamiento**.
 - Búsqueda eficiente de bloques libres consecutivos.
- **Desventajas:**
 - Puede ser **costosa en tiempo** si el disco es grande y el mapa de bits es extenso.

Lista ligada

- **Cómo trabaja:** Los **bloques libres del disco** se almacenan en una **lista enlazada**, donde cada bloque libre contiene un puntero al siguiente bloque libre.
- **Ventajas:**
 - No requiere espacio adicional significativo (punteros se almacenan en bloques libres).
 - **Fácil de mantener dinámicamente**.
- **Desventajas:**
 - **Acceso lento** para encontrar **múltiples bloques libres consecutivos**.
 - **Sobrecarga de punteros** en cada bloque.

Agrupamiento

- **Cómo trabaja:** Se almacenan **grupos de bloques libres** en un solo bloque de lista. El primer bloque del grupo apunta al siguiente conjunto de bloques libres, y así sucesivamente.
- **Ventajas:**
 - Reduce la cantidad de punteros necesarios en comparación con la lista ligada.
 - **Mejora el rendimiento** al manejar bloques libres en grupos.
- **Desventajas:**
 - La búsqueda de **bloques libres no consecutivos** puede ser más **lenta**.
 - **Sobrecarga de punteros** para manejar agrupamientos.

Recuento

- **Cómo trabaja:** Se almacena un **registro para cada grupo contiguo de bloques libres**. Cada registro incluye la dirección del primer bloque libre y el número de bloques libres consecutivos.

- **Ventajas:**
 - **Eficiente en términos de almacenamiento**, ya que un registro puede representar muchos bloques libres consecutivos.
 - **Rápido acceso a bloques contiguos.**
- **Desventajas:**
 - **Menos adecuado si los bloques libres están muy fragmentados.**

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.

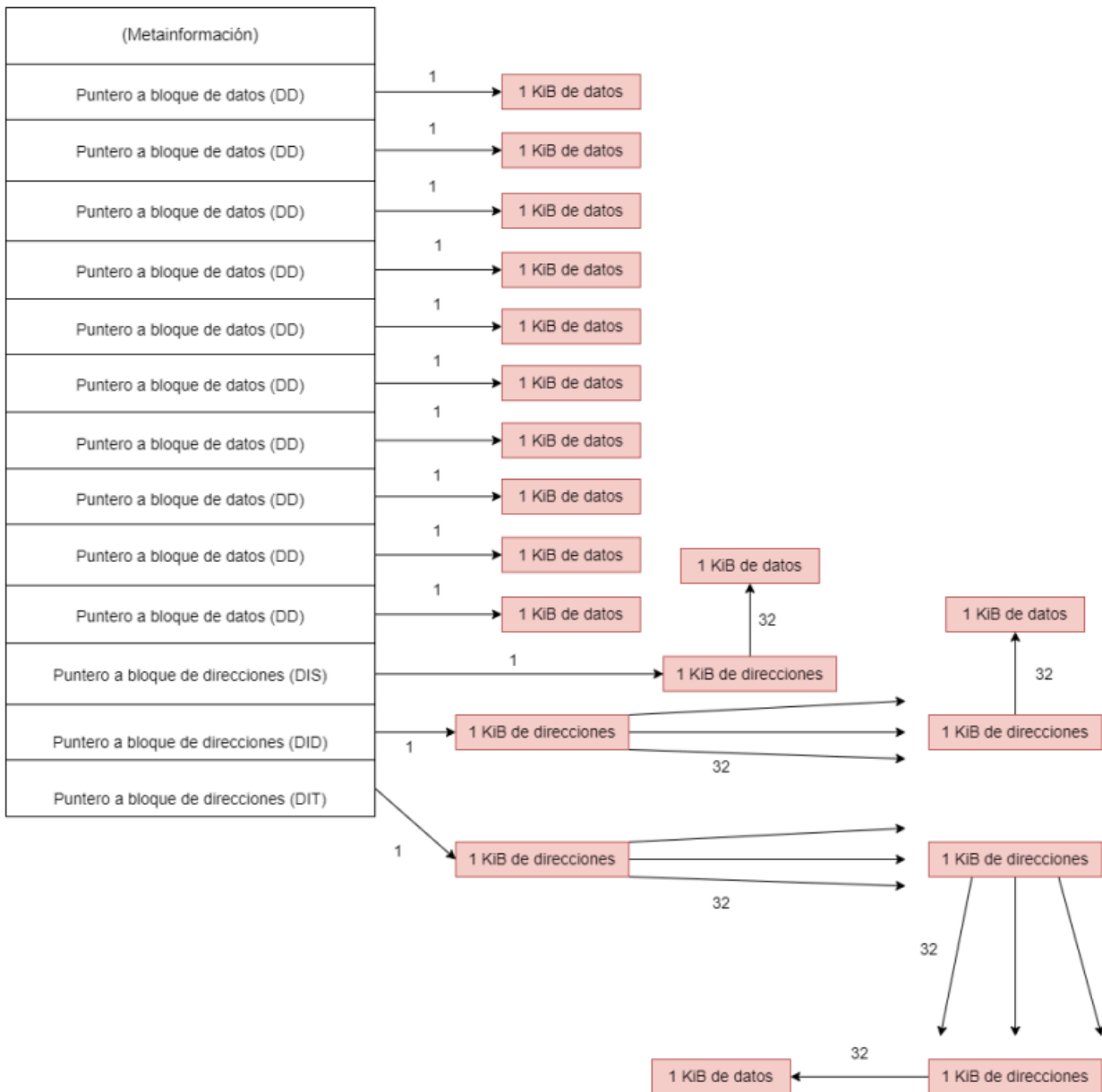
1 de direccionamiento indirecto simple.

1 de direccionamiento indirecto doble.

1 de direccionamiento indirecto triple.

(a) Realice un 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:



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

$1\text{Kib}(1024\text{ bits}) / 32\text{ bits} = 32$

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

$1\text{kib} \times 10^+$

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

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

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?

I-nodo 8

$$8 \text{ nodos} = ((8-1)/8)+2 = 2.87$$

$$16 \text{ nodos} = ((8-1)/16)+2 = 2.4375$$

I-nodo 9

$$8 \text{ nodos} = ((9-1)/8)+2 = 3$$

$$16 \text{ nodos} = ((9-1)/16)+2 = 2.5$$

(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?

I-nodo 6

$$((6-1) \% (8)) * 64 \text{ bytes} = 5 * 64 = 320$$

I-nodo 8

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