

## Administración de E/S

### 1) Dispositivos

#### a) Los dispositivos, según la forma de transferir los datos, se pueden clasificar en 2 tipos:

- Orientado a bloques
- Orientados a flujos

Describe las diferencias entre ambos tipos.

#### b) Cite ejemplos de dispositivos de ambos tipos.

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

Característica	Orientados a bloques	Orientados a flujos
Forma de transferencia	Los datos se transfieren en bloques de tamaño fijo.	Los datos se transfieren como un flujo continuo de bytes.
Acceso a los datos	Permiten acceso aleatorio (random access).	Permiten acceso secuencial (sequential access).
Estructura de los datos	Están organizados en bloques direccionables.	No tienen estructura, se procesan en el orden en que llegan.
Velocidad de transferencia	Más rápida para grandes cantidades de datos debido a su estructura en bloques.	Generalmente más lenta, pero adecuada para dispositivos de entrada continua.
Ejemplo de uso	Almacenamiento masivo, como discos duros.	Comunicación en tiempo real, como teclados o redes.

- Orientados a bloques:
  - Disco duro (HDD).
  - Unidad de estado sólido (SSD).
  - Pendrive USB
  - Dispositivos de almacenamiento óptico (CD/DVD).
- Orientados a flujos:
  - Teclado.
  - Mouse.
  - Micrófono.
  - Webcam.
  - Conexiones de red (como sockets).

### Diferencias que el SO debe considerar:

#### - Método de acceso:

- Acceso aleatorio: Dispositivos como discos permiten acceder directamente a cualquier posición.
- Acceso secuencial: Dispositivos como cintas magnéticas deben leerse secuencialmente.

- **Velocidad de transferencia:**
  - Los dispositivos tienen diferentes velocidades de E/S, y el SO debe sincronizarse con ellos para evitar pérdida de datos.
- **Modo de operación:**
  - Algunos dispositivos son síncronos (operan en intervalos definidos), mientras que otros son asíncronos (operan de manera irregular o por interrupciones).
- **Persistencia de datos:**
  - Dispositivos como discos duros mantienen los datos después de apagarse (almacenamiento persistente), mientras que dispositivos como la memoria RAM no lo hacen.
- **Direccionalidad:**
  - Algunos dispositivos son unidireccionales (teclado), mientras que otros son bidireccionales (discos).
- **Requerimientos de control:**
  - Los dispositivos pueden requerir operaciones especiales o comandos específicos, como en el caso de discos duros (lectura/escritura de bloques) frente a impresoras (comandos de formateo de texto).
- **Interacción del usuario:**
  - Los dispositivos interactivos (mouse, teclado) requieren un manejo en tiempo real para responder al usuario.

El sistema operativo debe manejar estas diferencias mediante controladores de dispositivo (drivers), técnicas de gestión de buffer, colas de solicitudes y políticas de planificación para la E/S.

## 2) Técnicas de E/S:

Describe cómo trabajan las siguientes técnicas de E/S:

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

**E/S programada:** El CPU controla directamente las operaciones de E/S. Esto implica que la CPU verifica constantemente si el dispositivo está listo para enviar o recibir datos, lo que se conoce como polling.

- **Funcionamiento:**
  - La CPU envía una solicitud al dispositivo.
  - La CPU espera verificando repetidamente el estado del dispositivo (polling).
  - Cuando el dispositivo está listo, la CPU transfiere los datos entre el dispositivo y la memoria principal.
  - La CPU reanuda la ejecución del programa.

### Ventajas:

- Simple de implementar.
- Adecuado para dispositivos rápidos o en sistemas donde la CPU no tiene muchas tareas paralelas.

Desventajas:

- Ineficiente, ya que la CPU permanece ocupada esperando al dispositivo.
- Consume recursos incluso cuando no hay datos disponibles.

**E/S dirigida por Interrupciones:** En esta técnica, el dispositivo genera una interrupción para notificar a la CPU cuando está listo para realizar una operación de E/S, eliminando la necesidad de un monitoreo constante por parte de la CPU.

- **Funcionamiento:**

- La CPU envía una solicitud de E/S al dispositivo.
- La CPU continúa ejecutando otras tareas mientras el dispositivo procesa la solicitud.
- Cuando el dispositivo está listo o completa la operación, genera una interrupción a la CPU.
- La CPU detiene temporalmente su trabajo actual, atiende la interrupción y transfiere los datos entre el dispositivo y la memoria.
- Una vez que la operación de E/S está completa, la CPU reanuda su trabajo.

Ventajas:

- Más eficiente que la E/S programada, ya que la CPU no se queda esperando al dispositivo.
- Libera a la CPU para realizar otras tareas mientras espera la señal de interrupción.

Desventajas:

- Puede ser más compleja de implementar.
- Si las interrupciones son frecuentes, puede introducir sobrecarga en la CPU debido al manejo de interrupciones.

**DMA:** Permite a los dispositivos de E/S transferir datos directamente hacia o desde la memoria principal, sin la intervención constante de la CPU. La CPU solo inicia la operación y recibe una notificación cuando esta ha finalizado.

- **Funcionamiento:**

- La CPU configura un controlador DMA con la dirección de memoria, el tamaño de los datos a transferir y el dispositivo de origen/destino.
- El controlador DMA toma el control del bus del sistema para gestionar la transferencia de datos entre el dispositivo y la memoria principal.
- Una vez completada la transferencia, el controlador DMA genera una interrupción para informar a la CPU que la operación ha finalizado.

Ventajas:

- Muy eficiente para transferencias de datos masivas, ya que libera a la CPU del trabajo repetitivo de mover datos.
- Reduce la sobrecarga en la CPU, permitiéndole realizar otras tareas mientras se realiza la transferencia.

**Desventajas:**

- Requiere hardware adicional (Controlador DMA).
- Es más complejo de implementar y coordinar.

Técnica	Intervención de la CPU	Eficiencia	Uso típico
E/S Programada	Alta (polling constante).	Baja.	Dispositivos simples o sistemas básicos.
E/S por Interrupciones	Media (manejo de interrupciones).	Media-Alta.	Dispositivos como teclado o mouse.
DMA	Baja (solo inicialización).	Alta.	Transferencias masivas (discos, redes).

**3)** La técnica de E/S programada puede trabajar de dos formas:

- E/S mapeada.
- E/S aislada.

Indique cómo trabajan estas 2 técnicas.

**E/S mapeada:** Las direcciones de memoria de los dispositivos de E/S se integran en el mismo espacio de direcciones que la memoria principal. Es decir, los dispositivos de E/S se acceden utilizando las mismas instrucciones que se usan para acceder a la memoria.

**Funcionamiento:**

- Cada dispositivo tiene una dirección (o rango de direcciones) en el espacio de memoria.
- Las operaciones de E/S se realizan mediante instrucciones de lectura y escritura de memoria estándar (LOAD, STORE, etc).
- La CPU no necesita instrucciones especiales para manejar dispositivos, ya que la interacción ocurre igual que con la memoria.

**Ventajas:**

- Simplicidad en la programación, ya que no se requieren instrucciones específicas para E/S.
- Los dispositivos pueden aprovechar el conjunto completo de instrucciones de la CPU para acceder a los datos.
- Posibilidad de implementar mecanismos de memoria caché para dispositivos de E/S.

**Desventajas:**

- El espacio de direcciones se comparte entre la memoria y los dispositivos, reduciendo la cantidad de memoria direccionable.
- Puede haber conflictos si no se diseñan adecuadamente las direcciones asignadas a dispositivos y memoria.

**E/S aislada:** Los dispositivos de E/S tienen un espacio de direcciones separado del espacio de direcciones de la memoria. Esto significa que las direcciones utilizadas para los dispositivos son independientes de las direcciones utilizadas para la memoria principal.

**Funcionamiento:**

- La CPU utiliza instrucciones específicas para interactuar con los dispositivos, como IN (entrada desde un dispositivo) y OUT (salida hacia un dispositivo).
- Las direcciones de los dispositivos se denominan puertos de E/S, y son independientes del espacio de memoria.
- Solo las operaciones específicas de E/S pueden acceder a los dispositivos.

**Ventajas:**

- El espacio de direcciones de la memoria permanece completamente disponible para la memoria principal, sin interferencia de los dispositivos de E/S.
- Reduce la complejidad del diseño del sistema al mantener separados los espacios de direcciones.

**Desventajas:**

- Requiere un conjunto adicional de instrucciones en la CPU para manejar la E/S.
- No se pueden utilizar las instrucciones de manipulación de memoria para interactuar con dispositivos.

Característica	E/S Mapeada	E/S Aislada
Espacio de Direcciones	Compartido entre memoria y dispositivos de E/S.	Separado entre memoria y dispositivos de E/S.
Instrucciones	Usa las mismas instrucciones que para memoria.	Requiere instrucciones específicas para E/S.
Facilidad de Implementación	Más simple en la programación.	Más simple en el diseño del hardware.
Compatibilidad	Compatible con caché de memoria.	No afecta el sistema de memoria caché.
Uso Típico	Sistemas donde la memoria y E/S están integrados (por ejemplo, microcontroladores).	Sistemas con muchos dispositivos de E/S, como computadoras personales.

4) Enuncie las metas que debe perseguir un SO para la administración de la E/S.

Meta	Descripción
Eficiencia	Optimizar el uso de dispositivos, minimizar tiempos de espera y reducir la sobrecarga en la CPU.
Independencia del Dispositivo	Abstraer el hardware para que los usuarios y aplicaciones no necesiten conocer detalles específicos.
Compartición	Permitir el acceso concurrente a dispositivos sin interferencias entre procesos.
Seguridad y Protección	Controlar el acceso a dispositivos, prevenir accesos no autorizados y garantizar la integridad de los datos.
Planificación y Priorización	Implementar políticas para gestionar solicitudes de E/S de manera justa y priorizar procesos críticos.
Velocidad	Sincronizar las velocidades de dispositivos y CPU, además de agrupar solicitudes para mayor rendimiento.
Fiabilidad	Detectar y manejar errores, y asegurar la recuperación de datos en caso de fallos durante las operaciones.
Escalabilidad	Soportar nuevos dispositivos y adaptarse a sistemas más grandes con más dispositivos de E/S.
Transparencia para el Usuario	Ofrecer interfaces simples y uniformes para interactuar con dispositivos de forma amigable.
Costo	Reducir costos operativos y prolongar la vida útil de los dispositivos de E/S.

## 5) Drivers:

- ¿Qué son?
- ¿Qué funciones mínimas deben proveer?
- ¿Quién determina cuáles deben ser estas funciones?

Un driver es un software especializado que actúa como intermediario entre el SO y un dispositivo de hardware específico.

Su función principal es traducir las solicitudes genéricas del SO o las aplicaciones a comandos específicos que el hardware pueda entender y ejecutar.

Cada dispositivo requiere un driver compatible para funcionar correctamente en un sistema.

Función	Descripción
Inicialización	Configurar el dispositivo al momento de encender el sistema o conectar el hardware.
Comunicación	Gestionar la transmisión de datos entre el dispositivo y el SO, ya sea enviando o recibiendo datos.
Manejo de interrupciones	Responder a las interrupciones generadas por el dispositivo para indicar que ha completado una tarea.
Control del dispositivo	Proveer comandos específicos para que el SO gestione configuraciones o funciones particulares del hardware.
Planificación de E/S	Coordinar las operaciones de entrada/salida con el dispositivo para optimizar el rendimiento.
Manejo de errores	Detectar, reportar y, si es posible, corregir errores en la operación del dispositivo.
Apagado/Desconexión	Desactivar el dispositivo o liberar recursos cuando deja de estar en uso o se desconecta.

- **Fabricante del hardware:**

Los fabricantes del hardware son responsables de desarrollar drivers que incluyan todas las funciones necesarias para que el dispositivo funcione correctamente y aproveche todas sus características.

- **Diseñadores del sistema operativo:**

El SO define una interfaz estándar para los drivers (como APIs y protocolos), especificando qué funciones deben implementarse para garantizar la compatibilidad y el correcto funcionamiento con el resto del sistema.

- Por ejemplo, sistemas operativos como Windows, Linux o macOS tienen especificaciones para los drivers según sus propios modelos de E/S.

- **Organismos de estandarización:**

En algunos casos, organismos internacionales (como la **USB Implementers Forum** o el **PCI-SIG**) definen estándares para dispositivos y especifican funciones que deben soportar los drivers asociados a estos estándares.

7) Describa mediante un ejemplo los pasos mínimos que suceden desde que un proceso genera un requerimiento de E/S hasta que el mismo llega al dispositivo.

- 1) El proceso solicita E/S mediante una llamada al sistema.
- 2) El SO verifica permisos y planifica la operación.
- 3) El driver traduce la solicitud a comandos entendibles para el dispositivo.
- 4) El dispositivo realiza la operación.
- 5) El dispositivo genera una interrupción para notificar la finalización.
- 6) El SO actualiza el estado del proceso y entrega los datos.

8) Describa mediante un ejemplo los pasos mínimos que suceden desde que un proceso genera un requerimiento de E/S hasta que el mismo llega al dispositivo.

- 1) Proceso solicita leer datos de un disco.
- 2) SO utiliza el driver del disco para enviar una orden de lectura.
- 3) El controlador del disco convierte la orden en comandos para el hardware.
- 4) El disco accede a los datos y los envía al SO (interrupciones o DMA).
- 5) SO entrega los datos al proceso.

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

**Abstracción:** Oculta la complejidad del hardware mediante APIs.

**Planificación:** Optimiza el orden de las operaciones de E/S.

**Manejo de interrupciones:** Responde a eventos generados por dispositivos.

**Control de dispositivos:** Provee acceso seguro y controlado a hardware.

**Manejo de errores:** Detecta y corrige fallos en dispositivos.

**Buffering y caching:** Almacena temporalmente datos para mejorar el rendimiento.

## Administración de discos

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

### Platos:

- Superficies circulares donde se almacenan los datos.
- Recubiertos de material magnético para grabar información.
- Cada disco puede tener varios platos apilados en un eje común.

### Pistas:

- Anillos concéntricos en cada lado de un plato.
- Cada pista almacena bloques de datos.

### Sectores:

- Divisiones de una pista, generalmente de tamaño fijo.
- Unidad mínima de almacenamiento direccionable.

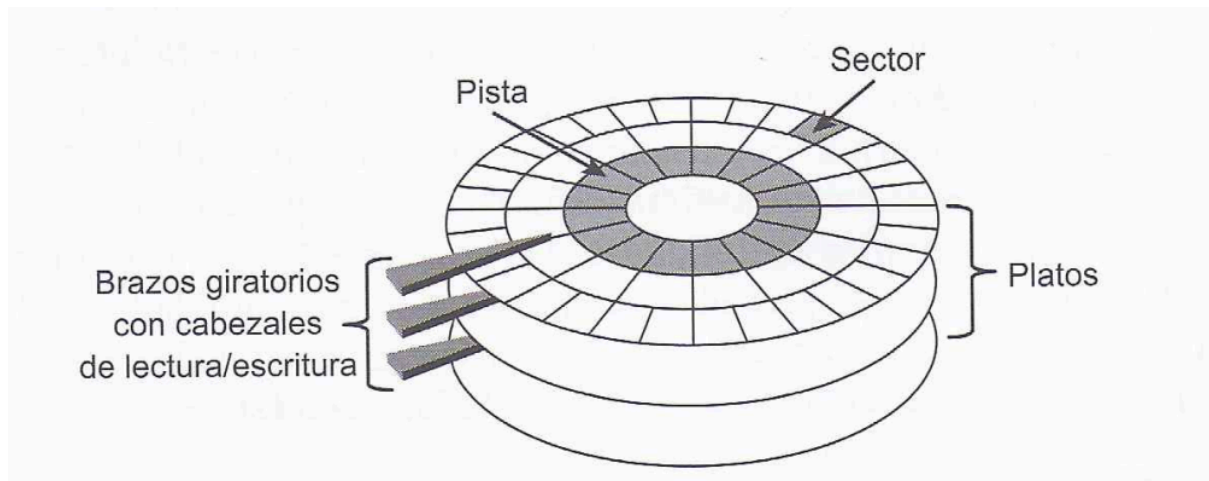


**Cilindros:**

- Conjunto de pistas alineadas verticalmente a través de los platos.
- Permite acceder a datos en diferentes platos sin mover la cabeza de lectura/escritura.

**Cabezas de lectura/escritura:**

- Mecanismos que leen/escriben datos sobre las pistas.
- Se posicionan mediante un brazo móvil que puede alcanzar diferentes pistas.



**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:** Tiempo que tarda la cabeza de lectura/escritura en posicionarse sobre la pista deseada.

Si el Seek Time no se conoce, se estima que equivale a lo que tarda el disco en dar media vuelta.

• Ejemplo. Disco de 5400 RPM →

5400 vueltas → 1' = 60" = 60000 ms

1/2 vuelta → x = 5,5 ms

**Latency Time:** Tiempo de espera para que el sector deseado gire y quede bajo la cabeza.

**Transfer Time:** Tiempo necesario para transferir los datos desde el disco a la memoria.

**12)** Suponga que tiene 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 10ms.
- 9000 RPM.
- Velocidad de transferencia de 10 MiB/s (Mebibytes por segundos).

- Calcule la capacidad total del disco.
- ¿Cuántos sectores ocuparía un archivo de tamaño de 3 MiB?
- Calcule el tiempo de transferencia real de un archivo de 15 MiB grabado en el disco de manera secuencial (todos sus bloques almacenados de manera consecutiva).
- Calcule el tiempo de transferencia real de un archivo de 16 MiB grabado en el disco de manera aleatoria.

Cálculo	Fórmula
Capacidad Total del Disco	Capacidad = número de caras × cilindros × sectores/pista × bytes/sector
Tiempo de Transferencia (Tt)	$T_t = \frac{\text{Tamaño del Archivo}}{\text{Velocidad de Transferencia}}$
Tiempo de Rotación	$T_r = \frac{60}{\text{RPM}}$
Latencia Promedio (Tl)	$T_l = \frac{T_r}{2}$
Tiempo Total	$T_{\text{total}} = T_s + T_l + T_t$

- **Almacenamiento secuencial:**  
*seek + latency + (tiempo\_transferencia\_bloque \* #bloques)*
- **Almacenamiento aleatorio:**  
*(seek + latency + tiempo\_transferencia\_bloque) \* #bloques*

Capacidad = 14 caras x 1100 cilindros x 300 sectores / pistas x 512 bytes / sector

Capacidad = 14 x 1100 x 300 x 512 = **2365440000** bytes = 2.365 GiB

Capacidad = cantidad caras x cantidad pistas x cantidad sectores por pista x tamaño sector

**Definición de la cátedra**

---

Sector = 512 bytes

1 MiB = 1024 KiB, entonces 3 MiB = 3 x 1024 KiB x 1024 bytes = 3.145.728 bytes

Sectores necesarios = Tamaño del archivo / Tamaño por sector

3.145.728/512 = 6144 sectores

---

Tamaño del archivo = 15 MiB

Tiempo de transferencia = Tamaño del Archivo / Velocidad de Transferencia

Tiempo de transferencia = 15 MiB / 10 MiB = 1.5 segundos

---

Tiempo total = Seek Time + Latencia Promedio + Tiempo Transferencia

Tiempo Rotación = 60 / RPM

Tiempo Rotación = 60 / 9000 = 0.006666667

Latencia Promedio = Tiempo Rotación / 2

Latencia Promedio = 0.006666667 / 2 = 0.003333334 s = 3.34s

Tiempo Transferencia = 16 MiB / 10 MiB = 1.6 segundos

Tiempo Total = 10 ms + 3.34 ms + 1.6 s = 1.61334 s

[Todo los cálculos de arriba los hizo Chat GPT, dudar mucho, no tengo ganas de revisarlos  
VER EL EJEMPLO SUBIDO AL GIT. Misma carpeta que este archivo.](#)

**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:

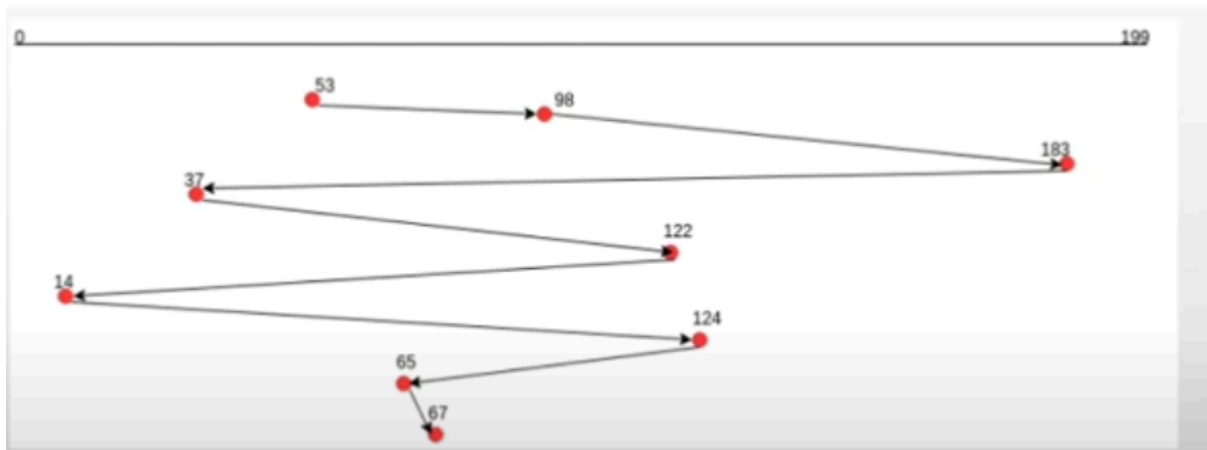
Supongamos un Head con movimiento en 200 tracks (0 a 199), que está en el track 83 atendiendo un requerimiento y anteriormente atendió un requerimiento en el track 75.

Si la cola de requerimientos es: 86, 147, 91, 177, 94, 150, 102, 175, 130, 32, 120, 58, 66, 115.

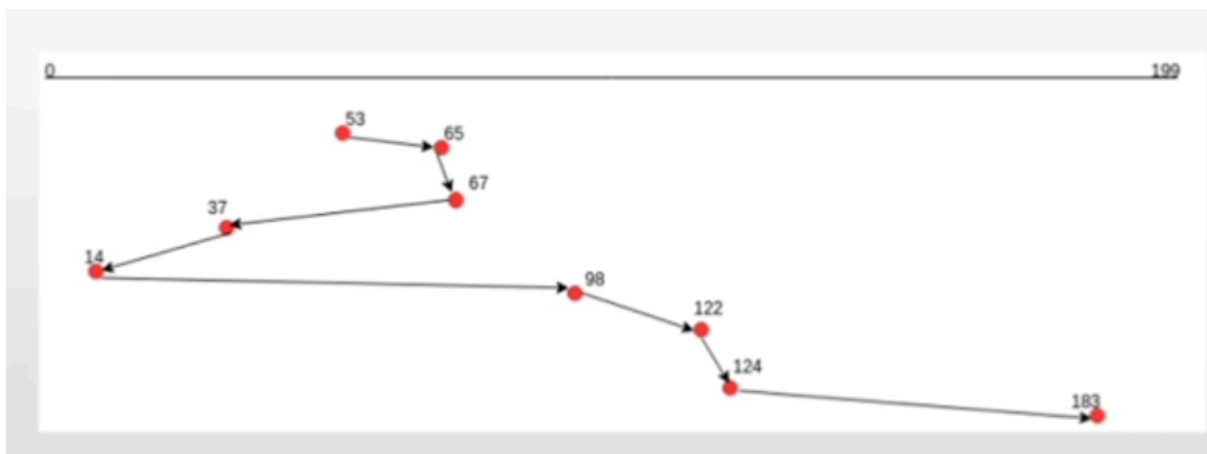
Realice los diagramas para calcular el total de movimientos de head para satisfacer estos requerimientos de acuerdo a los siguientes algoritmos de scheduling de discos:

- FCFS
- SSTF (Shortest Seek Time First)
- Scan
- Look
- C-Scan (Circular Scan)
- C-Look (Circular Look)

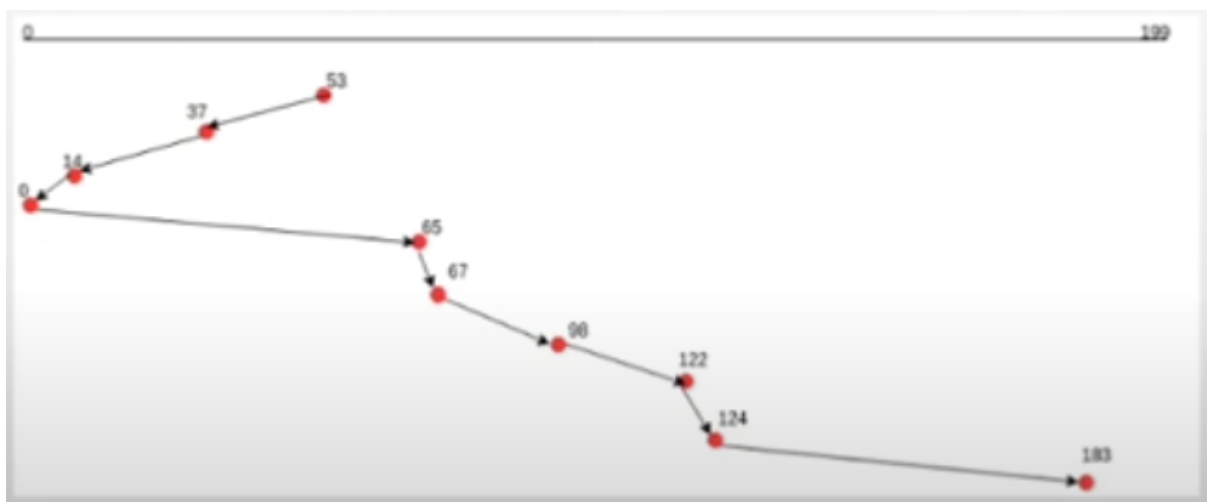
**FCFS:** Atiende los requerimientos por orden de llegada.



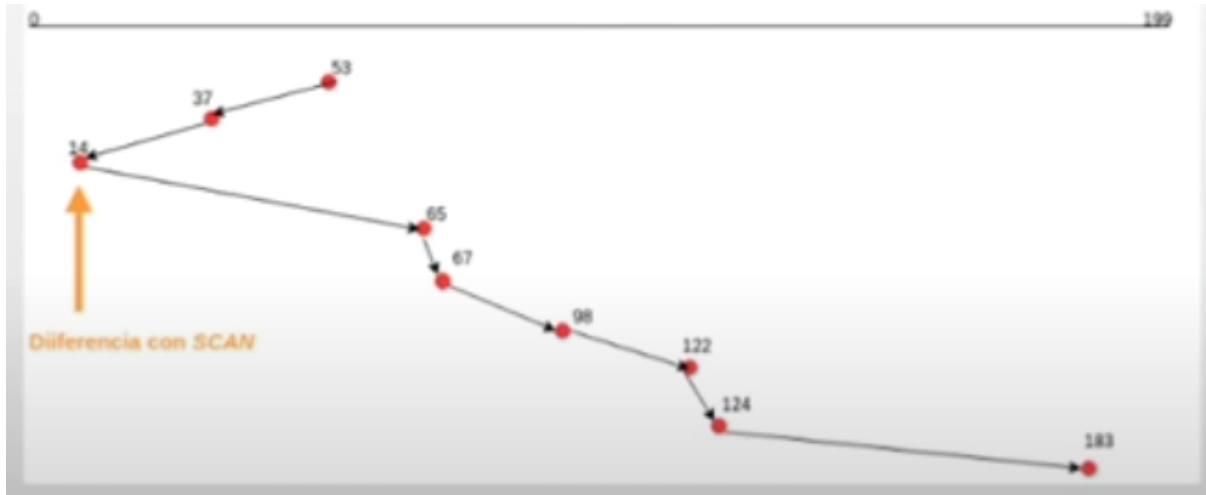
**SSTF:** Selecciona el requerimiento que requiere el menor movimiento del cabezal.



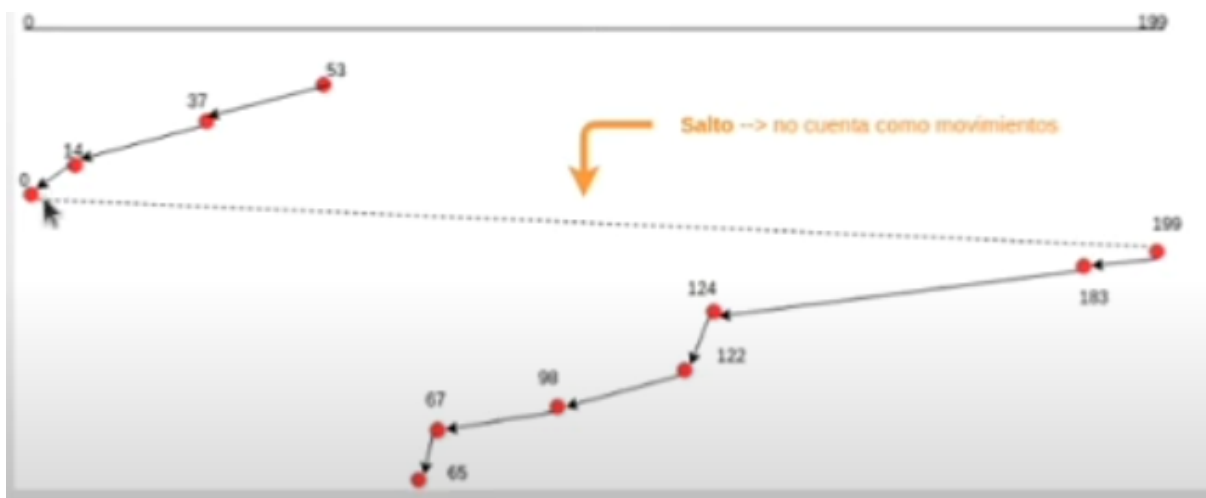
**Scan:** Barre el disco en una dirección atendiendo los requerimientos pendientes en esa ruta hasta llegar a la última pista del disco y cambia la dirección. Es **importante** saber en qué pista se está y de qué pista se viene para determinar el sentido del cabezal.



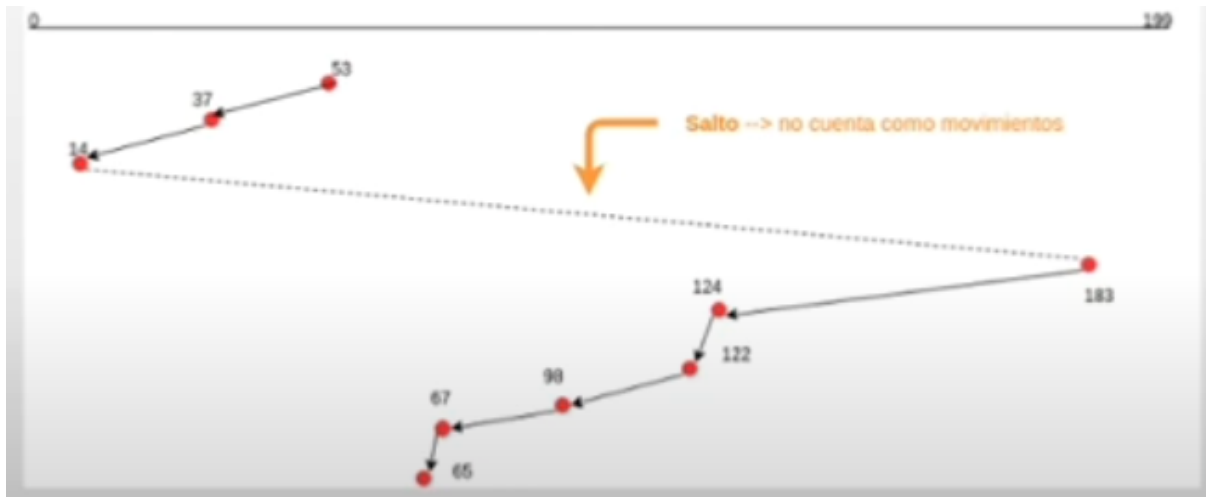
**Look:** Se importa igual que el SCAN pero no llega hasta la última pista del disco sobre la dirección actual, sino que llega hasta el último requerimiento de la dirección actual. Es **importante** saber en qué pista se está y de qué pista se viene para determinar el sentido del cabezal.



**C-Scan:** Se comporta igual que el SCAN pero restringe la atención en un solo sentido. Al llegar a la última pista del disco en el sentido actual, vuelve a la pista del otro extremo (**salto** = NO SE CUENTAN LOS MOVIMIENTOS) y sigue barriendo en el mismo sentido.



**C-Look:** Se comporta igual que el LOOK pero restringe la atención en un solo sentido. Al llegar a la última pista de los requerimientos en el sentido actual vuelve a la primera pista más lejana del otro extremo (**salto** = NO SE CUENTAN LOS MOVIMIENTOS) y sigue barriendo en el mismo sentido.



**FCFS:** Se atienden de manera separada, tantas veces como se quieran. Por ejemplo: 10, 40, 70, 10. Al 10 se le atiende 2 veces.

**SSTF, Scan, Look, C-Scan, C-Look:** Se atienden de manera consecutiva.

### (a) FCFS (First Come, First Served)

**Descripción:** Atender los requerimientos en el orden en que llegan.

**Pasos:**

El orden será el dado: 86 → 147 → 91 → 177 → 94 → 150 → 102 → 175 → 130 → 32 → 120 → 58 → 66 → 115.

**Cálculo del movimiento total:**

$$1. |83 - 86| = 3$$

$$2. |86 - 147| = 61$$

$$3. |147 - 91| = 56$$

... y así sucesivamente.

La suma total es el recorrido del cabezal.

### **(b) SSTF (Shortest Seek Time First)**

**Descripción:** Atender primero el requerimiento más cercano al cabezal.

**Pasos:**

1. El más cercano a 83 es 86.
2. Desde 86, el más cercano es 91.
3. Repetir este criterio hasta completar la cola.

**Cálculo del movimiento total:** Similar a FCFS, calculando distancias en el orden optimizado.

### **(c) Scan**

**Descripción:** El cabezal se mueve en una dirección (ascendente o descendente), atendiendo todos los requerimientos hasta llegar al extremo, luego cambia de dirección.

**Pasos:**

1. Desde 83, se atienden los tracks en dirección ascendente: 86 → 91 → 94 → ... → 177.
2. Al llegar al extremo superior (199), el cabezal regresa, atendiendo los requerimientos restantes.

### **(d) Look**

**Descripción:** Similar a **Scan**, pero el cabezal no llega a los extremos; solo atiende los requerimientos dentro de la lista.

**Pasos:**

1. Desde 83, atender tracks en dirección ascendente (máximo 177).
2. Cambiar dirección y atender los restantes (mínimo 32).

### **(e) C-Scan (Circular Scan)**

**Descripción:** El cabezal se mueve en una sola dirección (ascendente), atiende los requerimientos y, al llegar al extremo, regresa directamente al inicio sin atender solicitudes.

### **(f) C-Look (Circular Look)**

**Descripción:** Similar a C-Scan, pero solo atiende los tracks dentro de los valores requeridos (32 a 177), ignorando los extremos.