



Sistemas Operativos 2

Unidad 3: Dispositivos de almacenamiento

Planificación

René Ornelis
Primer semestre de 2025

Contenido

1	Introducción	4
1	Objetivos de la planificación	5
2	PEPS/FIFO.....	5
3	SSTF: Menor tiempo de búsqueda primero	6
4	SCAN	7
5	C-SCAN	7
5.1.1	Parking	8
6	N-SCAN/SCAN-N: SCAN de N pasos	8

Índice de figuras

Figura 1: Arquitectura del subsistema de almacenamiento	5
Figura 2: Planificación PEPS.....	6
Figura 3: Planificación SSTF.....	6
Figura 4: Planificación SCAN	7
Figura 5: Planificación C-SCAN	8

Planificación de discos

1 Introducción

""

Las políticas de planificación de discos duros en un sistema operativo son esenciales para gestionar cómo se accede y se organiza la lectura y escritura de datos en los discos. Estas políticas afectan directamente el rendimiento del sistema, la eficiencia en el uso del hardware y la experiencia del usuario. La importancia de estas políticas consiste en la influencia en el funcionamiento general del sistema en los siguientes aspectos:

2 Optimización del rendimiento

- **Minimización de la latencia:** Las políticas de planificación ayudan a reducir el tiempo que toma acceder a los datos en el disco. Esto es crucial para minimizar la latencia, especialmente en sistemas con altas cargas de trabajo o en aplicaciones que requieren acceso rápido a datos.
- **Maximización del rendimiento:** La planificación eficaz de las solicitudes de disco puede aumentar el throughput (rendimiento) general del sistema. Al agrupar las solicitudes y organizar las operaciones de manera eficiente, se puede mejorar la velocidad de transferencia de datos.

3 Gestión de la carga de trabajo

- **Equilibrio de carga:** En sistemas con múltiples procesos o usuarios, las políticas de planificación aseguran que la carga de trabajo se distribuya de manera equilibrada. Esto evita que un único proceso o usuario monopolice el acceso al disco, lo que podría llevar a una degradación del rendimiento para otros usuarios o procesos.
- **Priorización de solicitudes:** Las políticas de planificación permiten priorizar ciertas solicitudes de disco sobre otras. Por ejemplo, un sistema puede dar preferencia a las operaciones de lectura de archivos críticos del sistema operativo sobre las solicitudes de escritura de archivos temporales.

4 Reducción del desgaste del hardware

- **Eficiencia en el movimiento de cabezas:** En discos duros mecánicos, la planificación adecuada puede reducir el movimiento innecesario de las cabezas de lectura/escritura. Menos movimiento significa menos desgaste mecánico y una vida útil más prolongada del disco.
- **Optimización del uso de recursos:** Al minimizar el número de movimientos del disco y agrupar las solicitudes de manera eficiente, se reduce el consumo de energía y el estrés en el hardware.

5 Mejora en la experiencia del usuario

- **Respuesta rápida del sistema:** Una planificación eficiente mejora el tiempo de respuesta del sistema, lo que se traduce en una experiencia de usuario más fluida y menos demoras durante las operaciones que requieren acceso a disco.

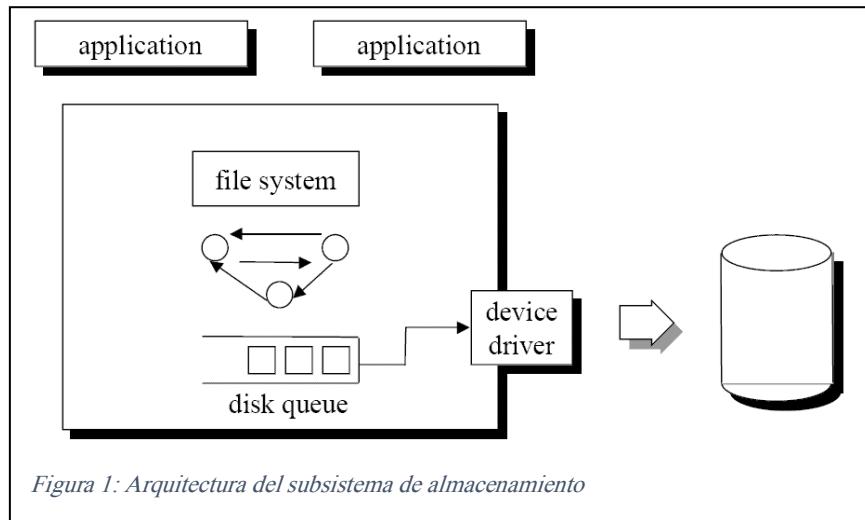
- **Consistencia en el rendimiento:** Las políticas de planificación ayudan a mantener un rendimiento consistente, evitando picos y caídas abruptas en la velocidad de acceso a disco que podrían afectar negativamente a la experiencia del usuario.

1 Objetivos de la planificación

El objetivo es minimizar los movimientos mecánicos, ya que éstos son mucho más lentos que los componentes electrónicos

Tal como se aprecia en la Figura 1, el proceso de acceso a un disco pasa por las siguientes capas:

1. Las aplicaciones invocan el API del sistema operativo para el sistema de archivos para acceder a un archivo, definido por una ruta de directorios y un nombre de archivo
2. El sistema de archivos (siguiente unidad) tiene la función de convertir el nombre de archivo y directorio en una secuencia de sectores para el disco.
3. Todos los requerimientos de disco (sectores) de todos los procesos se ingresan en la cola
4. Esta cola es la entrada para el proceso de planificación que debe reordenar los requerimientos con una de las políticas para optimizar el tiempo de acceso.
5. Cada requerimiento se envía al driver para acceder al disco.

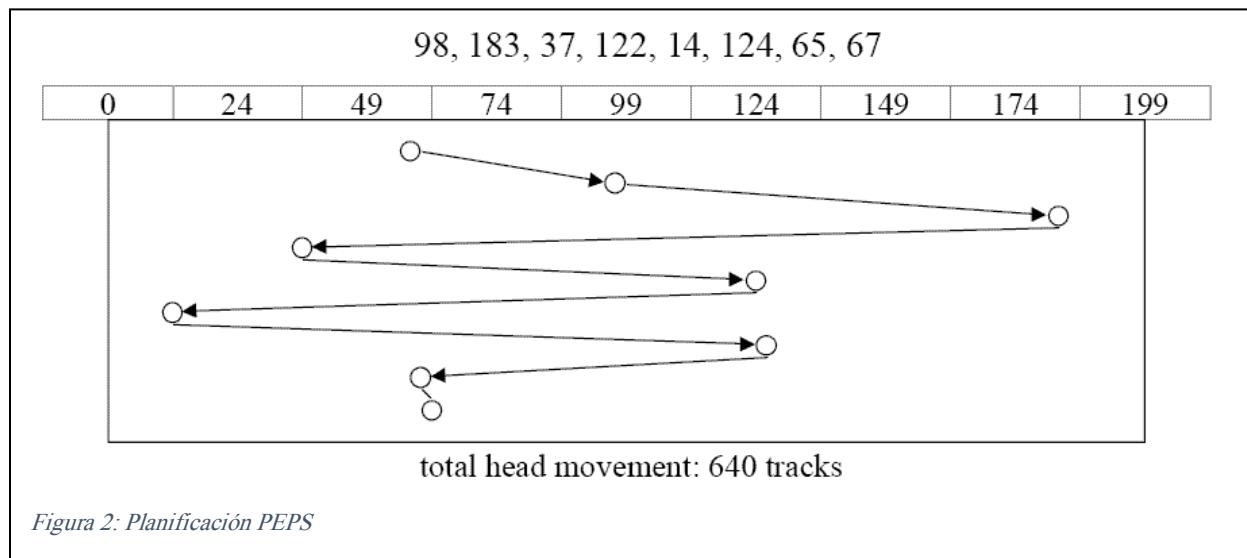


2 PEPS/FIFO

Los pedidos se atienden en el orden en que llegan. Es sencillo, pero puede no ser el más eficiente en términos de rendimiento y usualmente se utiliza solo para comparar con otras metodologías.

En la Figura 2 se ve el efecto de atender la siguiente cola de requerimientos de cilindros, a partir de la posición actual en el cilindro 53:

98, 183, 37, 122, 14, 124, 65, 67

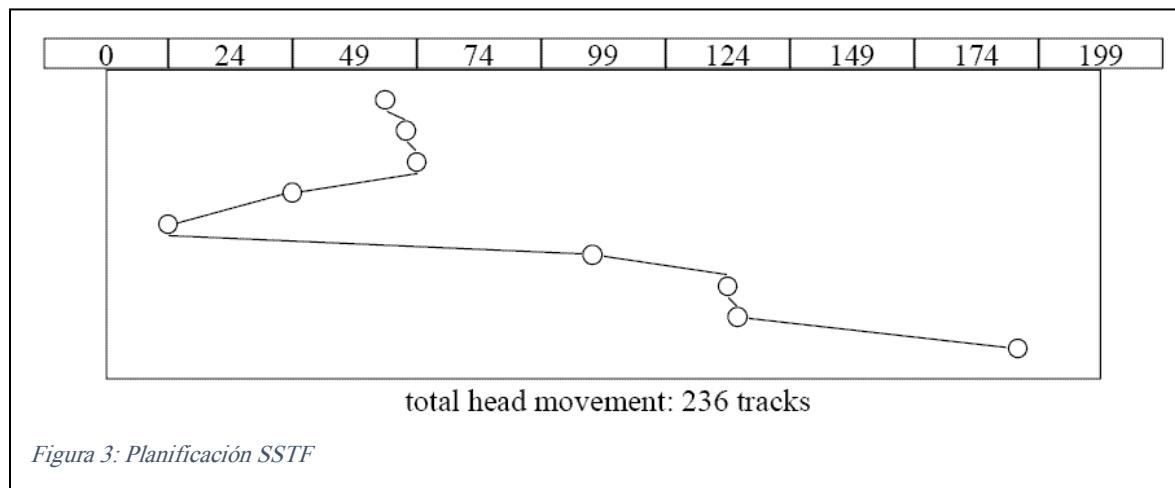


Como se ve, se tiene un recorrido total de 640 cilindros.

3 SSTF: Menor tiempo de búsqueda primero

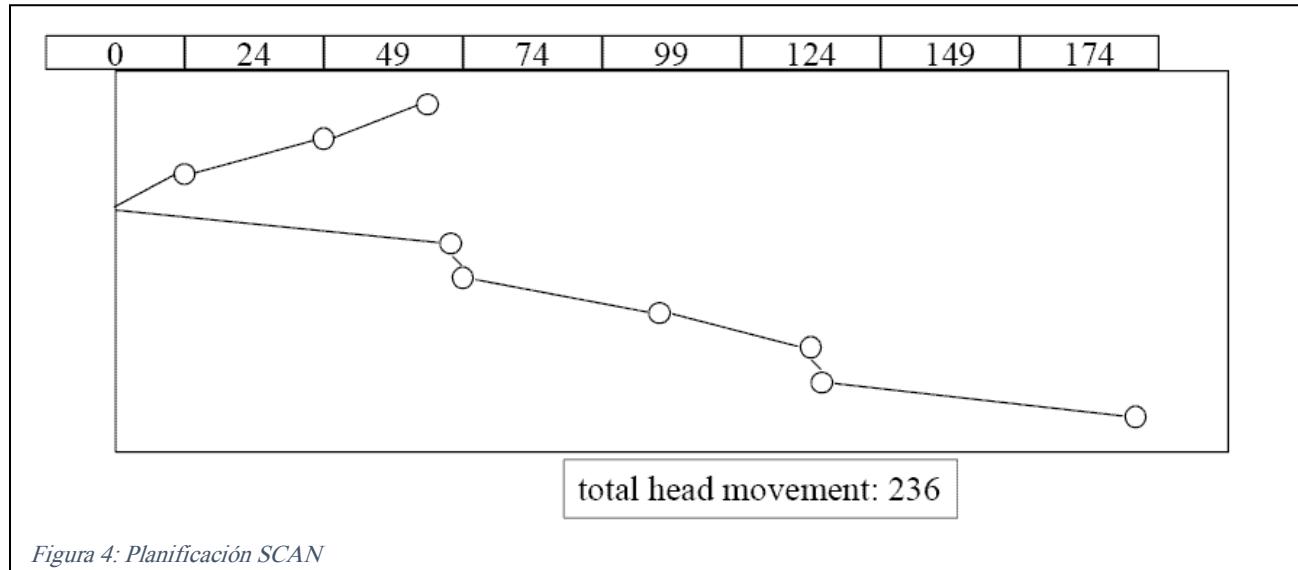
Da preferencia a la solicitud que está más cerca de la posición actual de la cabeza de lectura/escritura. Reduce el tiempo de búsqueda, pero puede llevar a la inanición de solicitudes más lejanas, debido a la **localización** de la cabeza, es decir, una vez entra en una región no sale de esa región hasta que atiende todos los requerimientos cercanos, lo que aumenta varianza del tiempo de espera, lo que provoca **postergación indefinida**.

En la Figura 3 se ve el efecto de esta política con la misma cola del ejemplo anterior.



4 SCAN

La cabeza de lectura/escritura se mueve en una dirección hasta el final del disco y luego vuelve en la dirección opuesta, atendiendo las solicitudes en su camino. También conocido como algoritmo del elevador, tal como se observa en la Figura 4.



La varianza de las peticiones de los procesos mejora porque los procesos son atendidos más más justamente ya que, en el peor de los casos, serán resueltos en una vuelta y contrario a SSTF los requerimientos ya no se acumulaban en una zona.

Sin embargo, se tiene el efecto de que los cilindros en la región del centro obtendrán más atención que los cilindros en las regiones de los extremos. **Variante Look:** Tal como se aprecia en el ejemplo, el algoritmo original de SCAN recorría todos los cilindros, pero se optimizó para llegar hasta el cilindro del último requerimiento e iniciar en el cilindro del primer requerimiento. Es decir: en el ejemplo anterior, en vez de llegar al cilindro 0, se cambiará de dirección en el cilindro 14.

5 C-SCAN

Similar al SCAN, pero una vez llega al final del disco, la cabeza de lectura/escritura vuelve al principio del disco para continuar el procesamiento de las solicitudes. Esto asegura un tiempo de espera más uniforme.

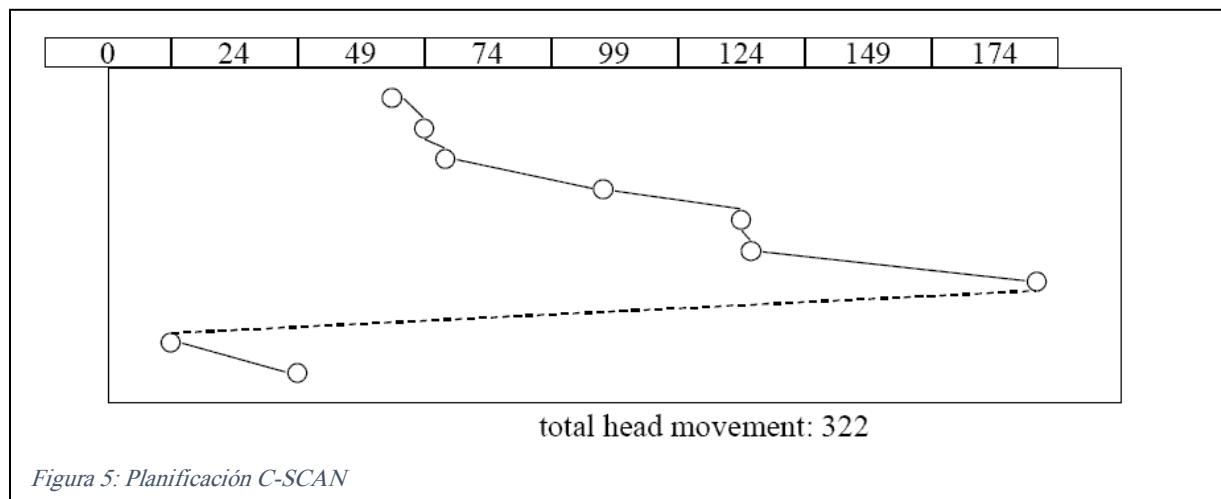


Figura 5: Planificación C-SCAN

Resuelve el problema del **SCAN** de favorecer a los cilindros del medio.

El movimiento mecánico para regresar sin atender es más rápido que el del SCAN (ver la sección de *parking*).

5.1.1 Parking

El **movimiento de parking** en un **HDD** (disco duro electromecánico) se refiere a un proceso específico que ocurre cuando el disco está apagado o fuera de funcionamiento, y las **cabezas de lectura/escritura** se colocan en una posición segura para evitar daños. Cuando el disco duro deja de estar en uso o se apaga, las cabezas se mueven hacia una **zona de estacionamiento** (el "parking"). Esta área generalmente se encuentra en un lugar seguro al borde del disco, donde las cabezas no están en contacto con los discos magnéticos. Esto protege las cabezas de posibles daños y evita el riesgo de que se rayen o deterioren los platos.

En general es más rápido realizar el movimiento de parking que mover la cabeza hasta el cilindro 0, ya que el firmware y el hardware están optimizados para que el aparcamiento se realice de manera rápida y segura.

6 N-SCAN/SCAN-N: SCAN de N pasos

- C-SCAN con listas de prioridad.
- **Caso básico de dos listas:** Se tiene una lista de los requerimientos actuales pendiente de atender. Al iniciar el recorrido de atención, los nuevos requerimientos se almacenan en otra lista: la lista de siguientes. Al finalizar el recorrido la lista de siguientes se convierte en la actual y se repite el proceso.
- Se trata con más justicia a los requerimientos (menor varianza del tiempo de espera).
- Puede haber listas de prioridad (postergación indefinida de los procesos de baja prioridad).

- Linux aplica tres listas: Memoria virtual, requerimientos del kernel y requerimientos de los procesos.