

Para calcular la capacidad de un HDD: se multiplica la cantidad de caras, la cantidad de pistas, la cantidad de sectores por pista y la cantidad de sectores.

Para calcular la transferencia de un archivo almacenado de manera contigua:

>**Seek**

>**Latencia:** Si no se conoce, es igual a lo que tarda el disco en dar media vuelta: $((60000 * 0,5) / \text{RPM})$.

>**Tiempo de transferencia,** pasar a bytes la velocidad de transferencia y hacer tamaño del sector / velocidad de transferencia en bytes.

Almacenamiento secuencial:

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

Almacenamiento aleatorio:

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

Prefijos:

Kibibyte (KiB): 2^{10}

Mebibyte (MiB): 2^{20}

Gibibyte (GiB): $2^{30} = 1,073,741,824$

Tebibyte (TiB): 2^{40}

Si quiero pasar una cantidad X de bytes a KiB, MiB, etc, simplemente dividir la cantidad de bytes X por el resultado de $2^?$ dependiendo del prefijo que quiera.

Calcular cuántas caras ocupará un archivo de X Kib, Mib, etc:

Calcular cuántos bytes ocupa el archivo: ej 512 MiB = $(2^{20}) * 512$

Calcular de qué tamaño son las caras: pistas por cara * sectores * pistas (convertir a bytes en caso sea necesario)

Dividir: archivo / tamaño de caras = cantidad de pistas (prestar atención a los decimales, si tiene decimales entonces necesitará 1 cara más)

Algoritmos de planificación en HDD:

Objetivo: minimizar el movimiento de la cabeza

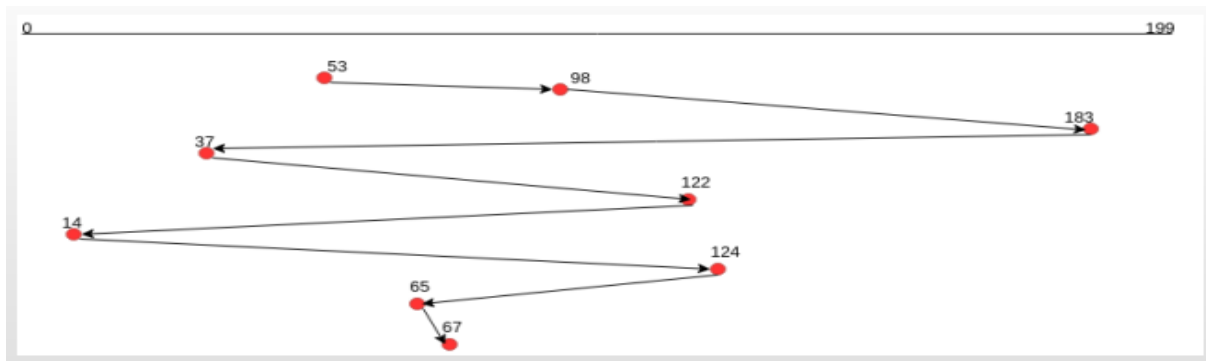
Cómo: ordenado lógicamente los requerimientos pendientes (que están en la cola) al disco, considerando el número de cilindro de cada requerimiento. En cualquier momento se pueden encolar nuevos movimientos.

Para calcular el movimiento del seek debo sumar la cantidad que se mueve por cada requerimiento, ya sea que sume o reste, se suma.

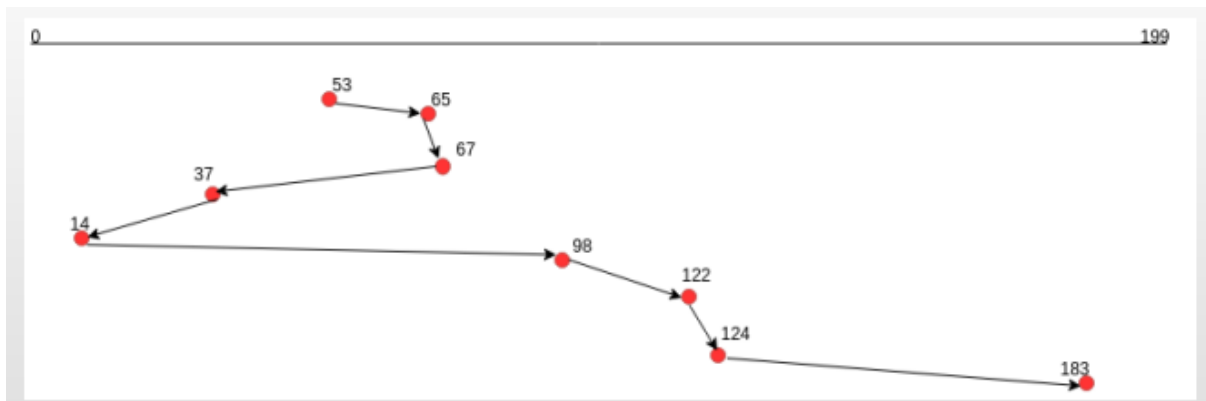
Pistas: 200 (0..199)

Para el lote de requerimientos: {98 , 183 , 37, 122, 14, 124, 65, 67}

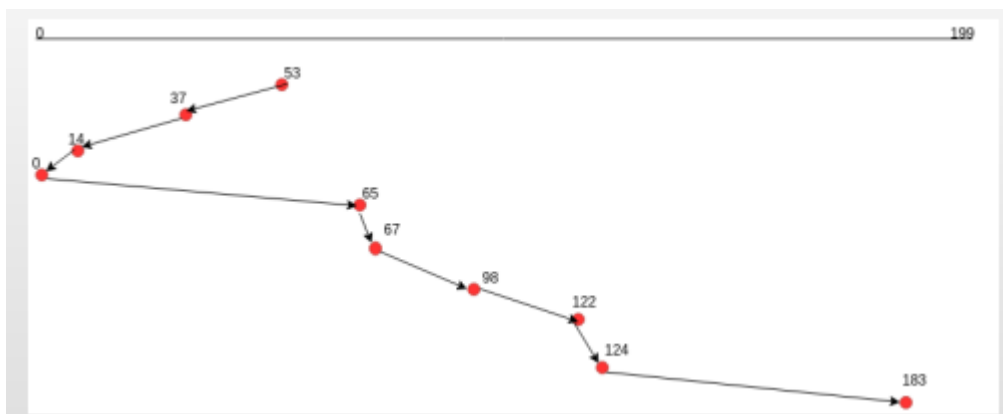
FCFS: Se atienden los requerimientos por orden de llegada.



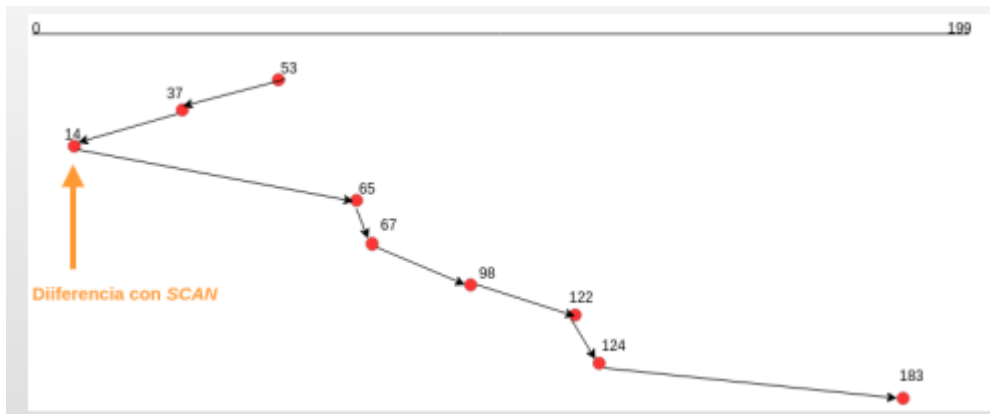
SSTF: (Sortest Seek Time First) se atiende el requerimiento con menor movimiento de cabezal.



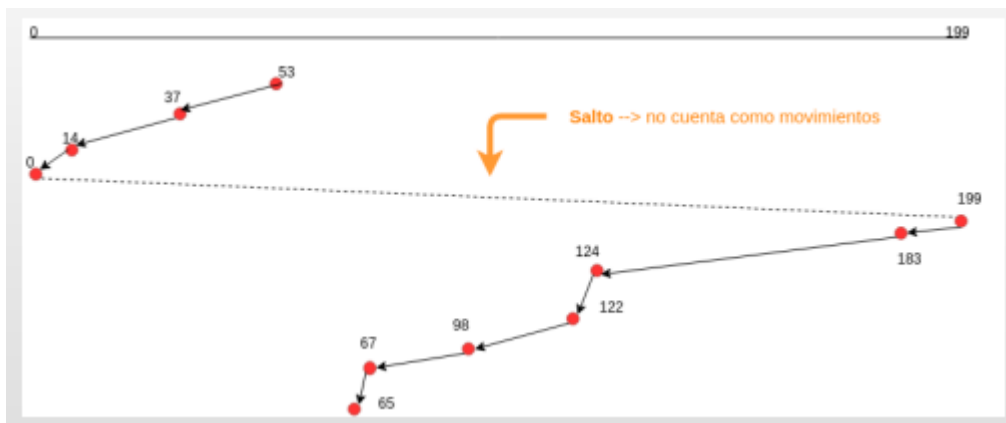
SCAN: barre el disco en una dirección atendiendo los requerimientos pendientes en esa ruta **hasta llegar a la última pista del disco (llega hasta 0 aunque no sea necesario)** y cambia la dirección. Es importante saber en qué pista se está y de qué pista se viene para determinar el sentido del cabezal.



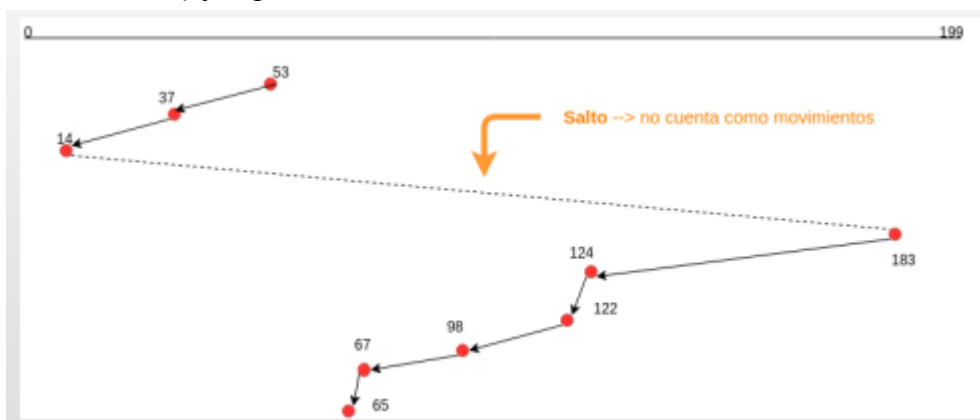
LOOK: se comporta igual que el SCAN pero no llega hasta la última pista del disco sino que llega hasta el último requerimiento de la dirección actual.



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.



Si tengo requerimientos con PF entonces se atienden primero.

Una vez que no existan más requerimientos por page faults en la cola, se procede:

FCFS: en orden FCFS

SSTF: en orden SSTF

SCAN: con el sentido que determina la atención de los últimos dos requerimientos
→ puede cambiar de sentido

C-SCAN: con el sentido original → el sentido no cambia

LOOK: del mismo modo en que lo hace el SCAN

C-LOOK: del mismo modo en que lo hace el C-SCAN