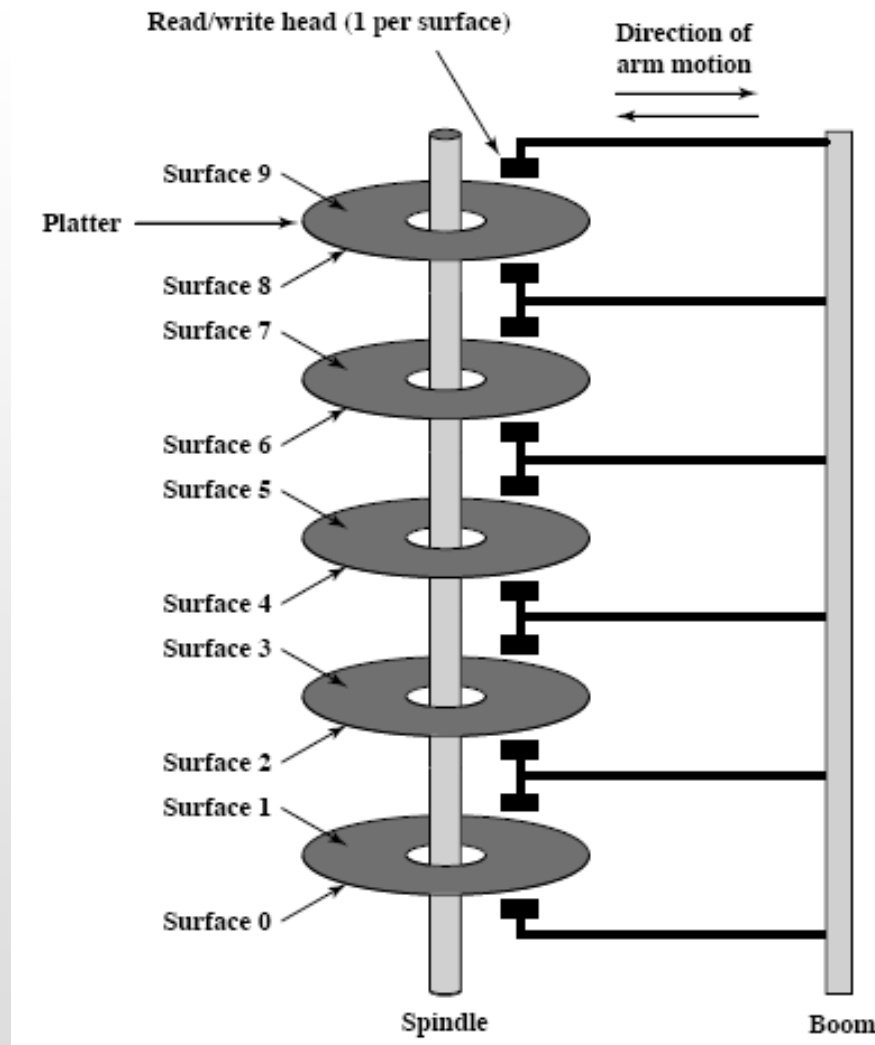


Introducción a los Sistemas Operativos

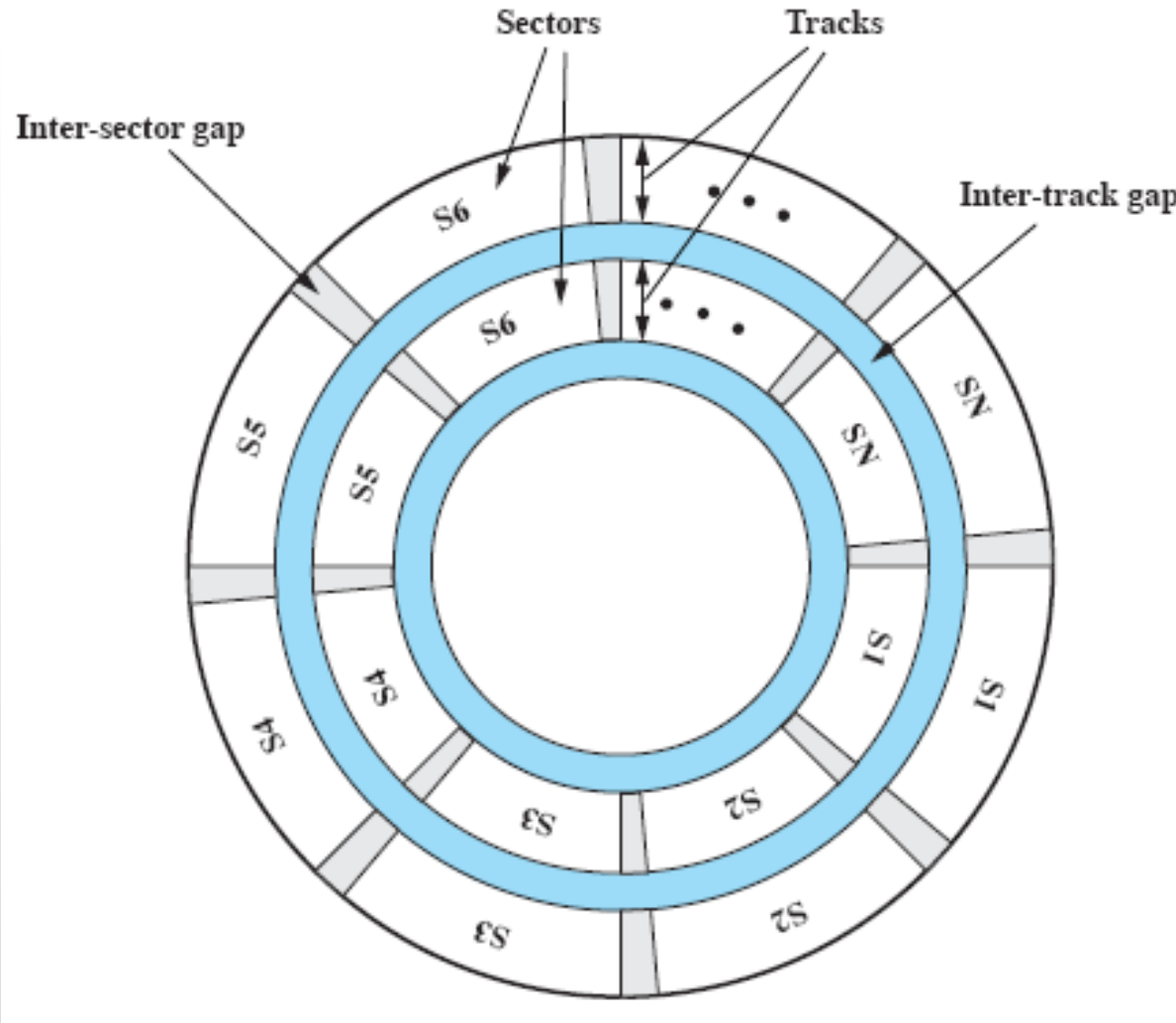
*Administración de
E/S - Discos
Práctica 6*



Organización Física de Discos



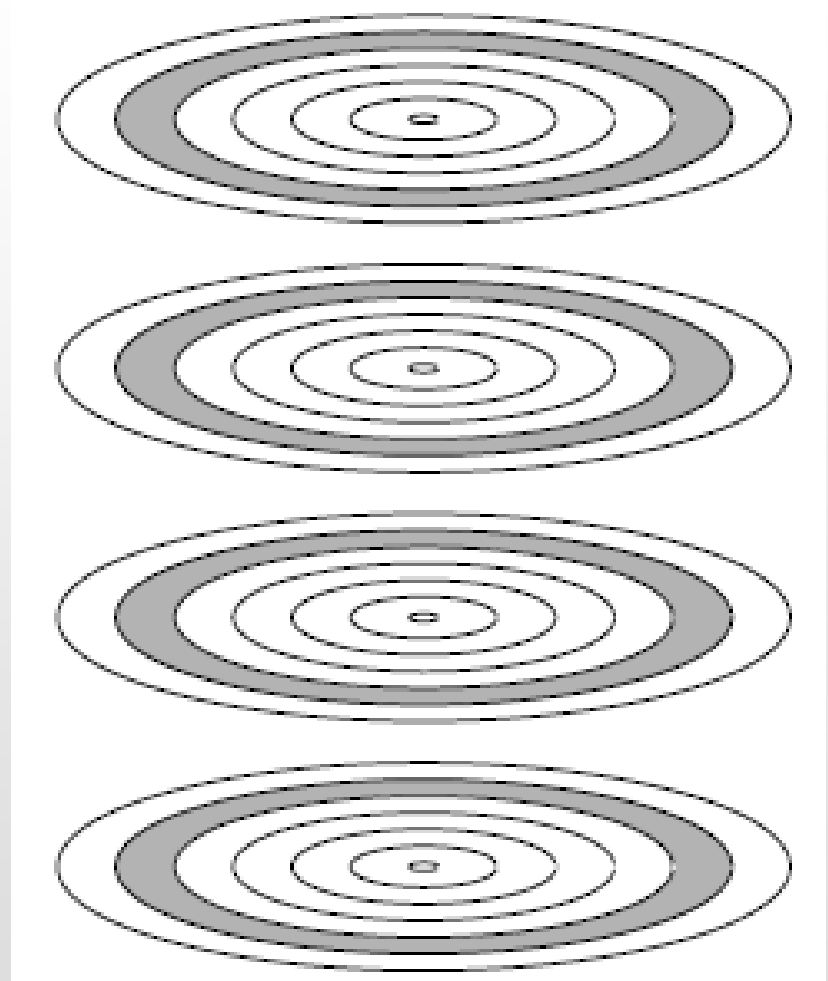
Organización Física de Discos (Cont.)



Organización Física de Discos (Cont.)

✓ Cilindros

- ✓ Cilindro N: todas las n-esimas pistas de todas las caras



Capacidad de un disco

☑ La capacidad de un disco esta dada por:

- ✓ Cantidad de Caras: W
- ✓ Cantidad de Pistas: X
- ✓ Cantidad de Sectores por Pista: Y
- ✓ Tamaño del Sector: Z

$$\text{Capacidad} = W * X * Y * Z$$



Acceso a un Disco

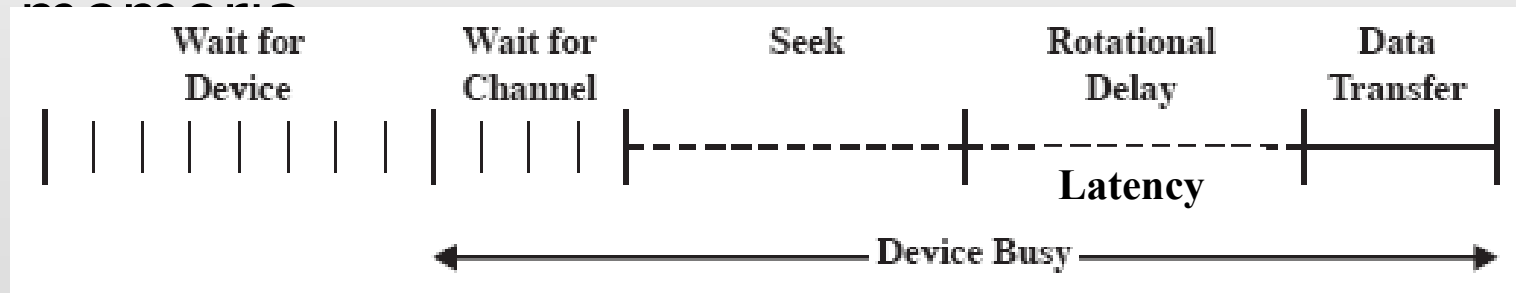
- ✓ Para realizar una Entrada/Salida, por ejemplo un acceso a disco, se requiere de una Llamada al Sistema. En la misma se especifica:
 - ✓ Tipo de Operación (E o S)
 - ✓ Dirección en disco para la transferencia (file descriptor). El file descriptor se obtuvo al hacer la apertura del archivo.
 - ✓ Dirección en memoria para la transferencia (de donde se lee o escribe).
 - ✓ Numero de bytes a transmitir.
- ✓ Este requerimiento es pasado por el kernel al sub-sistema de E/S quien lo traduce en:

(# Cara , # Cilindro, # Sector)



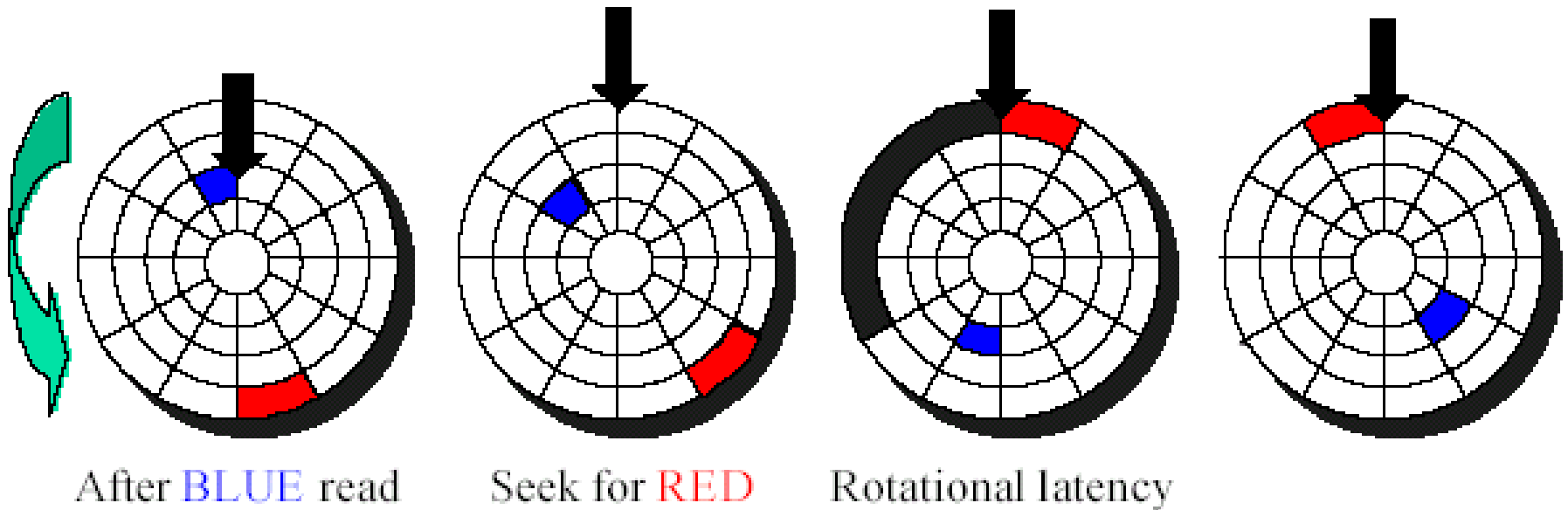
Tiempo de Acceso a un Disco

- ✓ El tiempo de acceso esta dado por:
 - ✓ Seek Time (Posicionamiento): Tiempo que tarda en posicionarse la cabeza en el cilindro
 - ✓ Latency Time (Latencia): Tiempo que se sucede desde que la cabeza se posiciona en el cilindro hasta que el sector pasa por debajo de la misma.
 - ✓ Transfer Time (Transferencia): Tiempo de transferencia del sector (bloque) del disco a la memoria



Tiempo de Acceso a un Disco (cont.)

✓ La Latencia



Tiempo de Acceso a un Disco (cont.)

- ✓ Latency: Si este tiempo no se conoce se considera que es igual a lo que el disco tarda en dar media vuelta.
- ✓ Ej:

Disco de 5400 RPM (5400 Vueltas en 1 minuto) →

$$5400 \rightarrow 1' = 60'' = 60.000 \text{ ms.}$$

$$\frac{1}{2} \rightarrow X? = 5,5 \text{ ms.}$$



Tiempo de Acceso a un Disco (cont.)

- ✓ Archivo almacenado de manera secuencial

$\text{Seek} + \text{Latency} + (\text{Tiempo_Trans_1_bloque} * \text{\#Bloques})$

- ✓ Archivo almacenado de manera aleatoria

$(\text{Seek} + \text{Latency} + \text{Tiempo_Trans_1_bloque}) * \text{\#Bloques}$



Prefijos Binarios

- ✓ Nos permiten crear múltiplos binarios (basados en potencias de 2)
- ✓ Son similares en concepto, aunque difieren en valor a los prefijos del Sistema Internacional (SI) basados en potencias de 10 (kilo, mega, giga)

Unidades básicas de información (en bytes)

Prefijos del Sistema Internacional			Prefijo binario	
Múltiplo - (Símbolo)	Estándar SI	Binario	Múltiplo - (Símbolo)	Valor
kilobyte (kB)	10^3	2^{10}	kibibyte (KiB)	2^{10}
megabyte (MB)	10^6	2^{20}	mebibyte (MiB)	2^{20}
gigabyte (GB)	10^9	2^{30}	gibibyte (GiB)	2^{30}
terabyte (TB)	10^{12}	2^{40}	tebibyte (TiB)	2^{40}



Tiempo de Acceso a un Disco - Ejemplos

- ✓ Supongamos un disco con 6 platos, con 2 caras útiles, 1500 pistas por cara y 700 sectores por pista de 256 bytes cada uno
- ✓ Si queremos calcular la capacidad total del disco, hacemos:

#Caras Total * #Pistas por cara * #Sectores por pista * Tamaño del sector = Tamaño del Disco

$(6 * 2) * 1500 * 700 * 256 \text{ bytes} =$

$225600000 \text{ bytes} = 3,00407 \text{ GiB (Gibibytes)}$



Tiempo de Acceso a un Disco - Ejemplos

- ✓ Supongamos un disco con 6 platos, con 2 caras útiles, 1500 pistas por cara y 700 sectores por pista de 256 bytes cada uno
- ✓ Si queremos saber **cuántas caras** ocupará un archivo de 513 Mebibytes almacenado de manera contigua a partir del primer sector de la primera pista de una cara determinada :
 - ✓ Lo primero que tenemos que hacer es ver cuánta información podemos almacenar en 1 cara
 $1500 * 700 * 256 \text{ bytes} = 268800000 \text{ bytes}$
 - Una vez que tenemos este dato, lo dividimos por el tamaño del archivo que queremos almacenar:

513 MiB = 537919488 bytes

$537919488 / 268800000 = 2,00118 \rightarrow 3 \text{ Caras}$



Tiempo de Acceso a un Disco - Ejemplos

- ✓ Supongamos un disco con 6 platos, con 2 caras útiles, 1500 pistas por cara y 700 sectores por pista de 256 bytes cada uno
- ✓ El disco gira a 12600 RPM, tiene un tiempo de posicionamiento (seek) de 2 milisegundos y una velocidad de transferencia de 15 Mib/seg (Mebibits por Segundo)
- ✓ Si queremos saber Cuantos milisegundos se tardarían en transferir un archivo **almacenado de manera contigua** de 4500 sectores :

✓ Recordemos las formulas:

■ De estos datos tenemos: **Seek: 2 ms**
Seek + Latency + (Tiempo_Trans_1_bloque * #Bloques)

■ Latencia:

12600 vueltas → 1 Minuto = 60 Segundos = 60000 ms

0,5 vueltas → **x = 2,3809**



Tiempo de Acceso a un Disco - Ejemplos (cont.)

- ✓ Supongamos un disco con 6 platos, con 2 caras útiles, 1500 pistas por cara y 700 sectores por pista de 256 bytes cada uno
- ✓ El disco gira a 12600 RPM, tiene un tiempo de posicionamiento (seek) de 2 milisegundos y una velocidad de transferencia de 15 Mib/seg (Mebibits por Segundo)
- ✓ Si queremos saber Cuantos milisegundos se tardarían en transferir un archivo **almacenado de manera contigua** de 4500 sectores :

✓ Recordemos las formulas:

Seek + Latency + (Tiempo_Trans_1_bloque * #Bloques)

■ Tiempo de transferencia de 1 bloque:

15 Mebibits → 1 Segundo = 1000 ms

256 bytes → X

15728640 Bits → 1 Segundo = 1000 ms

2048 Bits → X = 0,1302

■ Si tengo 4500 bloques:

$0,1302 * 4500 = 585,9 \text{ ms.} \leftarrow \text{Tiempo_Trans_1_bloque} * \# \text{Bloques}$



Tiempo de Acceso a un Disco - Ejemplos (cont.)

- ✓ Supongamos un disco con 6 platos, con 2 caras útiles, 1500 pistas por cara y 700 sectores por pista de 256 bytes cada uno
- ✓ El disco gira a 12600 RPM, tiene un tiempo de posicionamiento (seek) de 2 milisegundos y una velocidad de transferencia de 15 Mib/seg (Mebibits por Segundo)
- ✓ Si queremos saber Cuantos milisegundos se tardarían en transferir un archivo **almacenado de manera aleatoria** de 4500 sectores :

✓ Recordemos las formulas:

$$(\text{Seek} + \text{Latency} + \text{Tiempo_Trans_1_bloque}) * \# \text{Bloques}$$

■ Juntando los datos:

- Seek: 2 ms
- Latencia: 2,3809
- Tiempo de Transferencia de 1 Bloque: 0,1302 milisegundos
- Cantidad de Bloques: 4500

$$(2 + 2,3809 + 0,1302) * 4500 = 20299,95 \text{ ms.}$$



Planificación de Requerimientos

- ☑ Seek Time → Parámetro que mas influye en el tiempo de acceso al disco
- ☑ El SO:
 - ✓ Es responsable de utilizar el hardware en forma eficiente. Para los discos, esto significa obtener el menor tiempo de atención del requerimiento.
 - ✓ Debe por lo tanto minimizar el Seek Time → implica menor distancia recorrida por el brazo.



- ❑ Objetivo:
 - ❑ Minimizar el movimiento de la cabeza
- ❑ Como:
 - ❑ Ordenando lógicamente los requerimientos pendientes a disco, considerando el número de cilindro de cada requerimiento



Algoritmos – Ejemplo de Enunciado

✓ Cantidad de Pistas:

200 (0..199).

✓ Requerimientos en la cola:

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

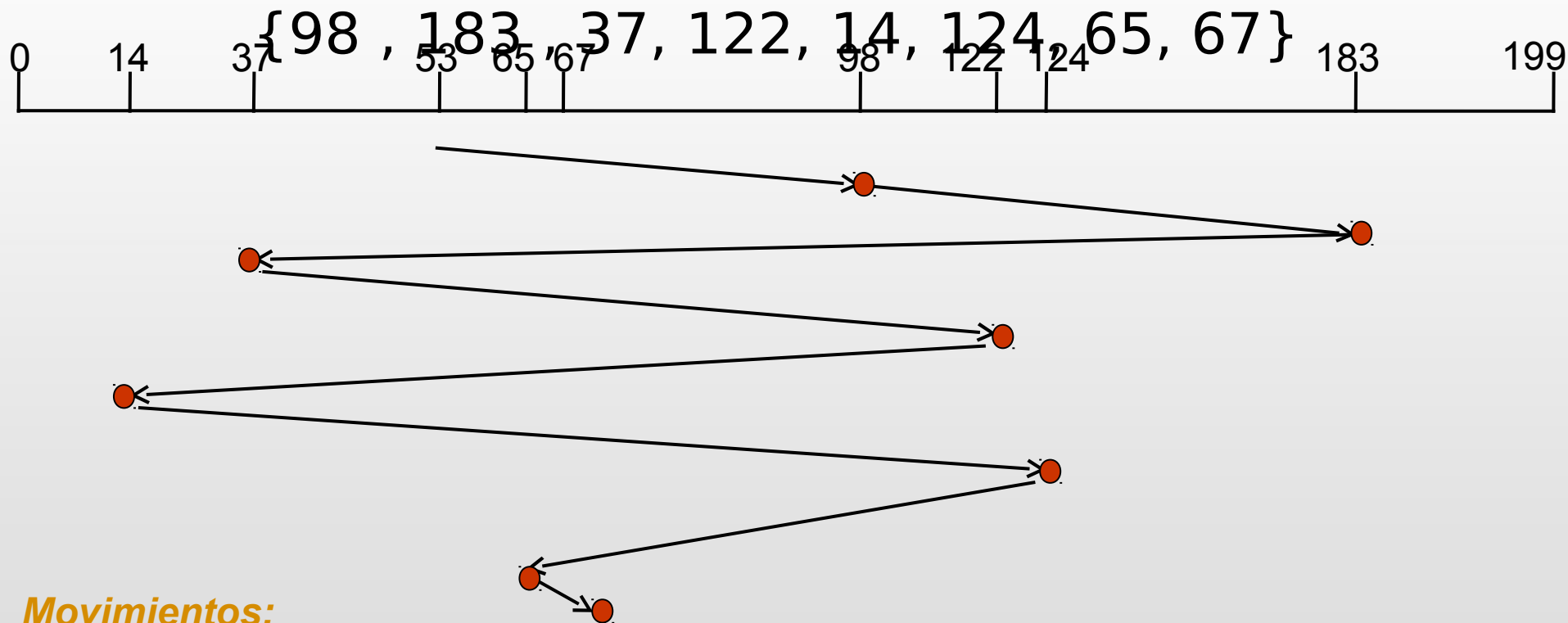
✓ Ubicación actual de la cabeza

Pista 53



Algoritmos - Ejemplo - FCFS

✓ FCFS: Atiende los requerimientos por orden de llegada.



Movimientos:

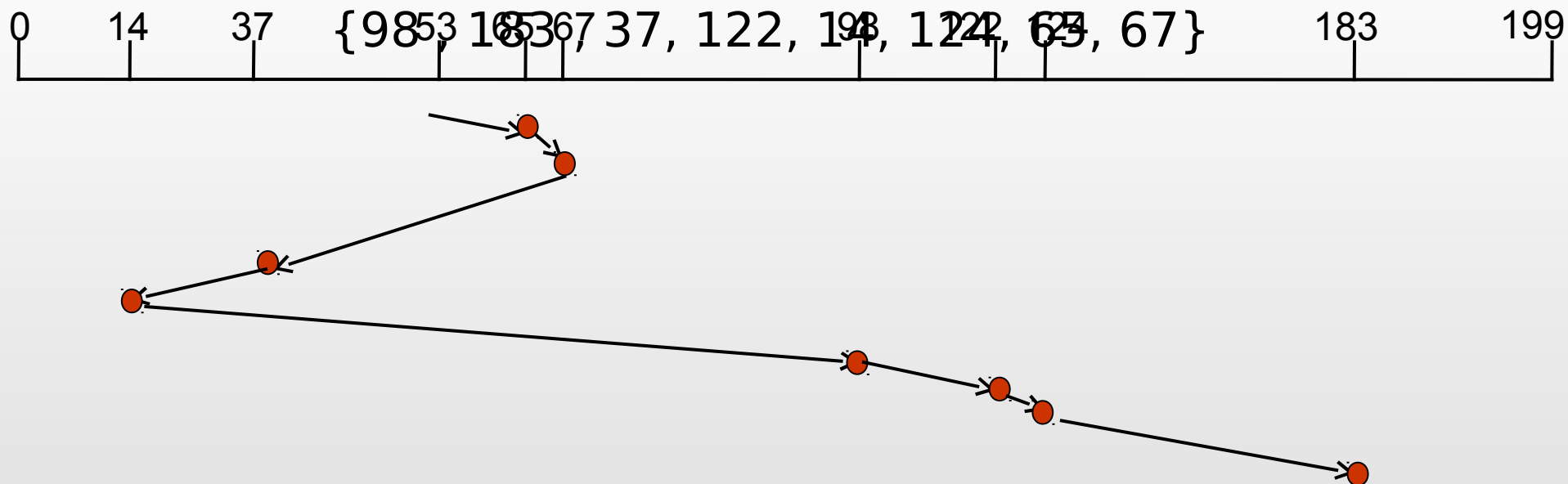
$$(183 - 53) + (183 - 37) + (122 - 37) + (122 - 14) + (124 - 14) + (124 - 65) + (67 - 65) = 640$$



Algoritmos - Ejemplo - SSTF



SSTF (Shortest Seek Time First): Selecciona el requerimiento que requiere el menor movimiento de la cabeza.



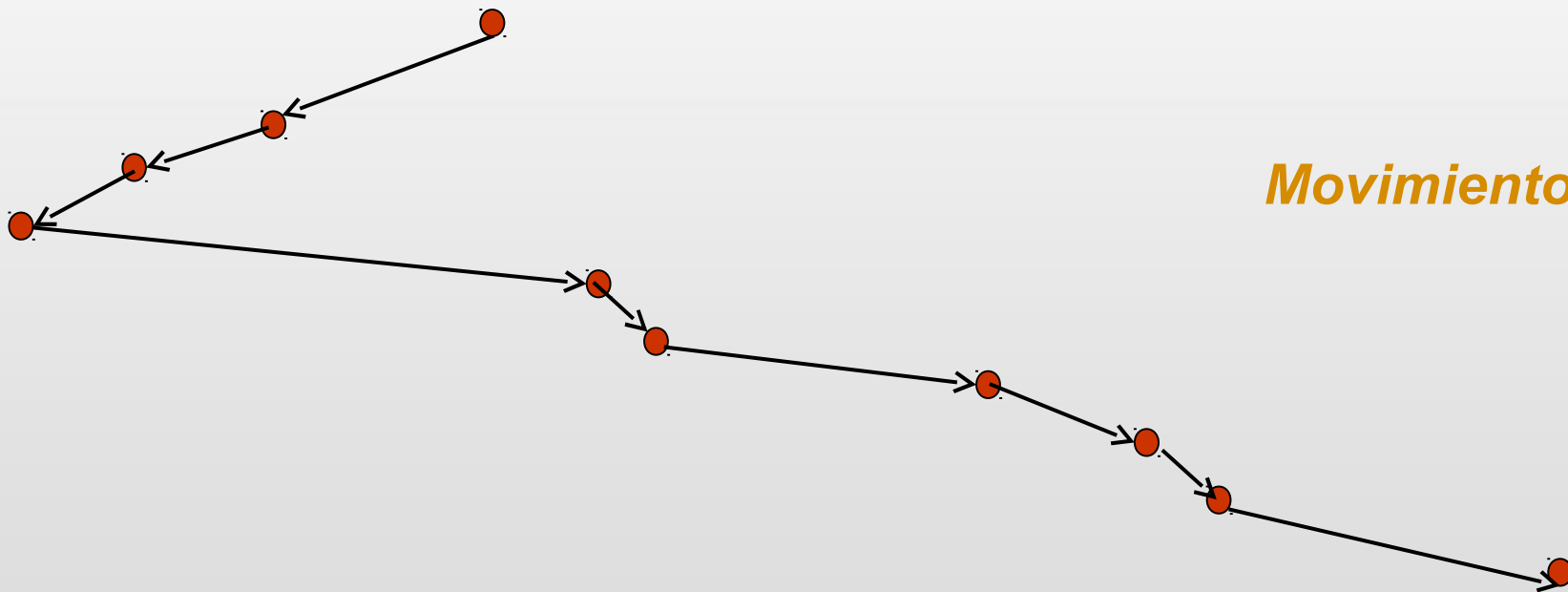
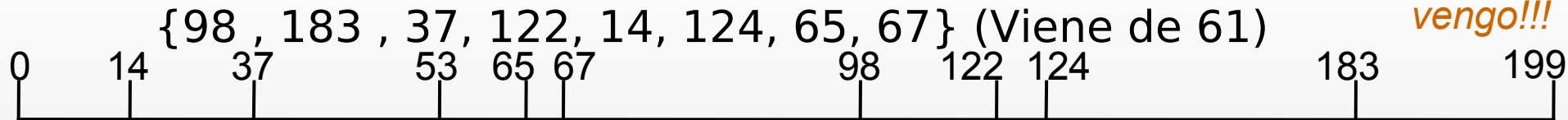
Algoritmos - Ejemplo - SCAN



SCAN: Barre el disco en una dirección atendiendo los req. pendientes en esa ruta hasta llegar a la ultima pista y luego cambia de dirección.

Importante: De que pista vengo!!!

{98, 183, 37, 122, 14, 124, 65, 67} (Viene de 61)



Movimientos: 236

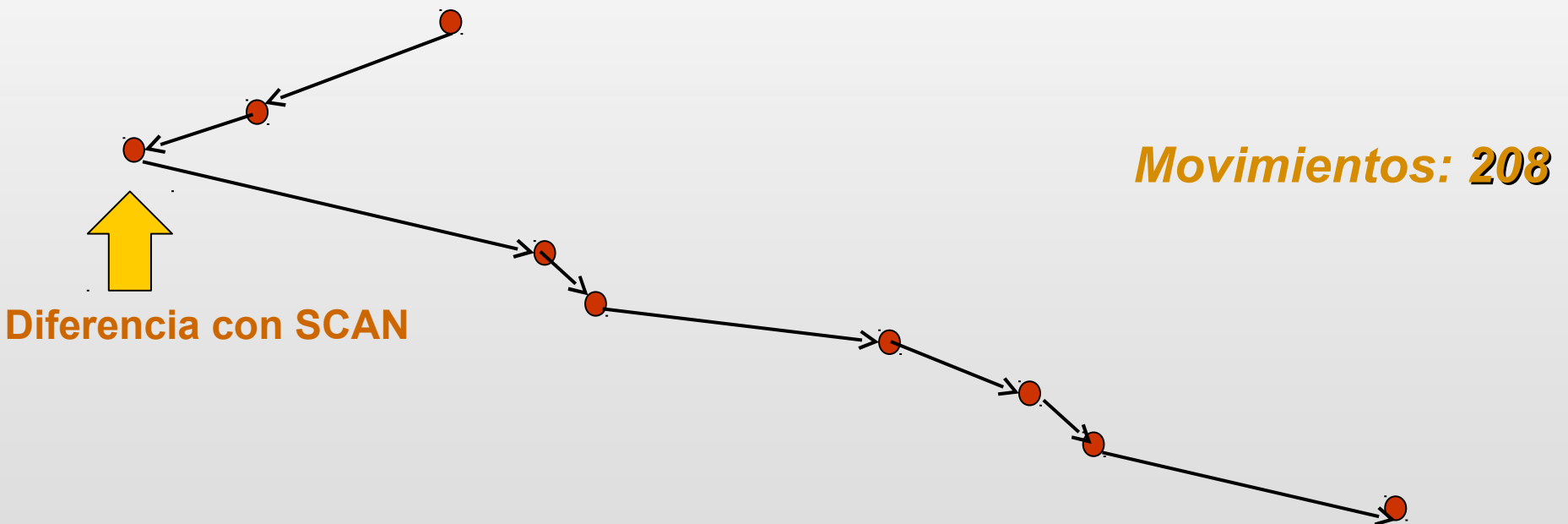
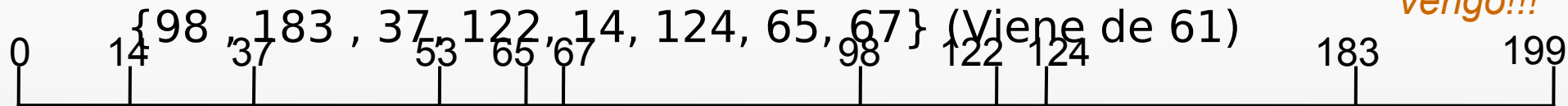


Algoritmos - Ejemplo - LOOK



LOOK: Idem SCAN pero no llega hasta la ultima pista sino hasta el ultimo requerimiento en la dirección que se encuentra atendiendo.

Importante: De que pista vengo!!!

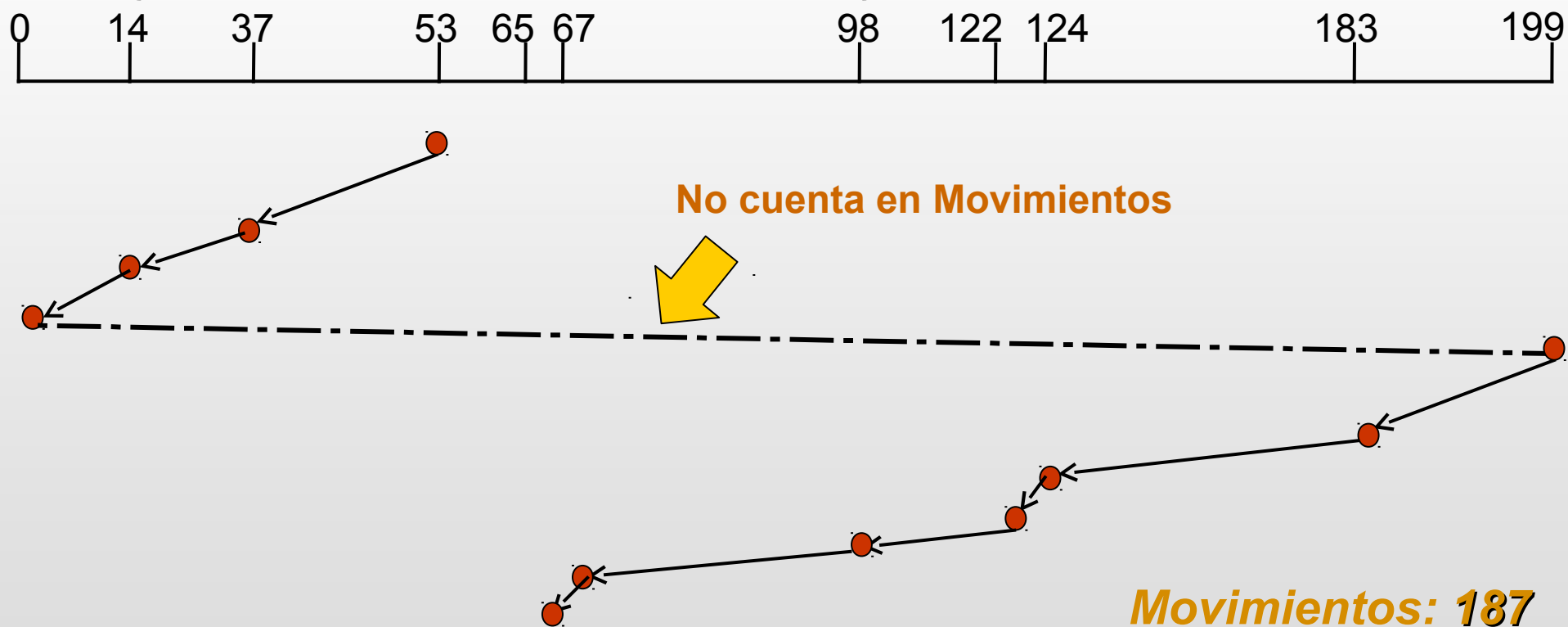


Algoritmos - Ejemplo - C-SCAN



C-SCAN: Idem SCAN, pero restringe la atención en un solo sentido. Al llegar a la ultima pista (o ultimo requerimiento) vuelve a la primer pista y comienza a barrer nuevamente. **Importante:** De que pista vengo!!!

{98 , 183 , 37, 122, 14, 124, 65, 67} (Viene de 61)



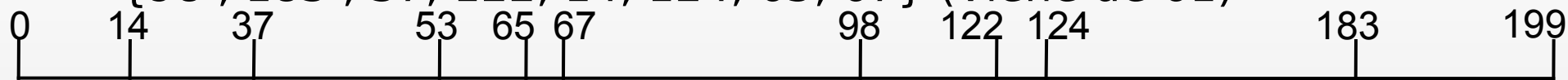
Algoritmos - Ejemplo - C-LOOK



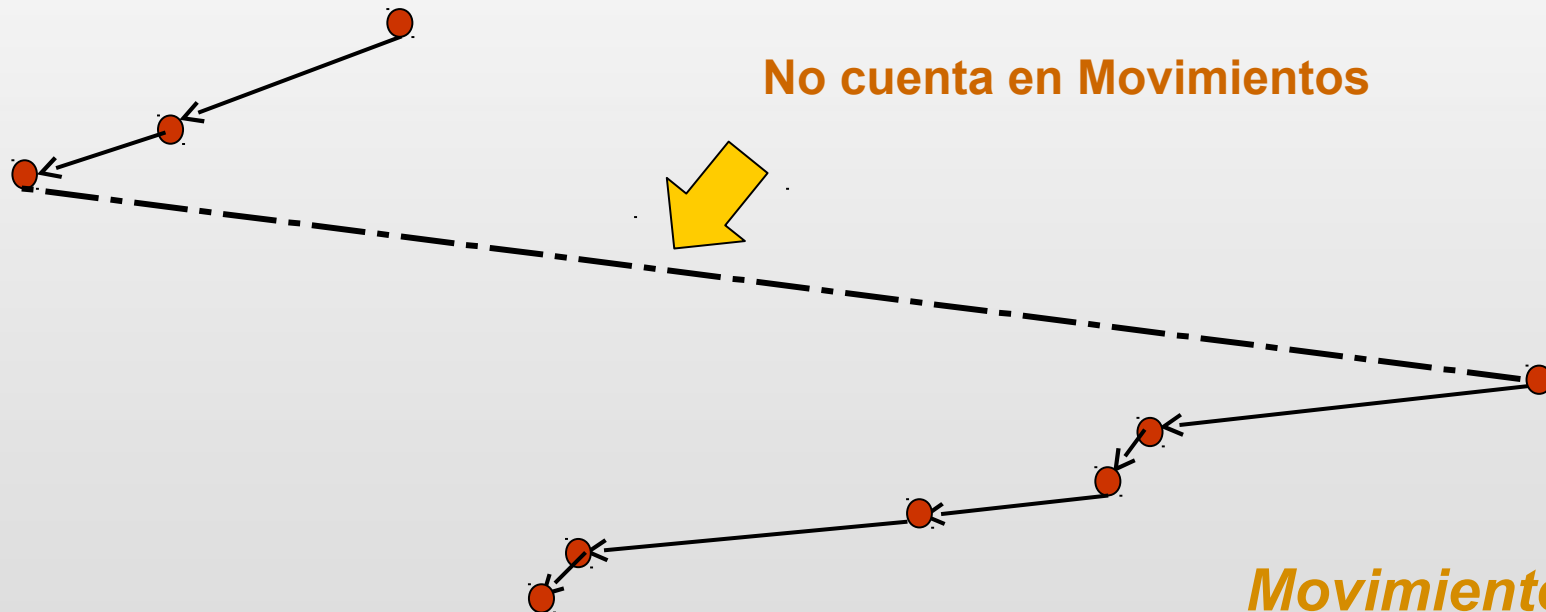
C-LOOK: Idem LOOK, pero restringe la atención en un solo sentido. Al llegar al ultimo requerimiento vuelve al menor requerimiento y comienza a barrer nuevamente.

Importante: De que pista vengo!!!

{98, 183, 37, 122, 14, 124, 65, 67} (Viene de 61)



No cuenta en Movimientos

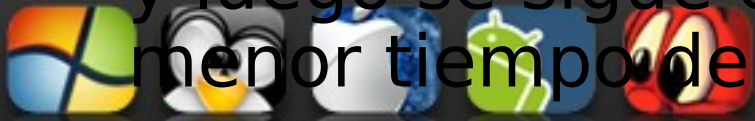


Movimientos: 157



Algoritmos – Atención de Fallos de Página

- ✓ Existen requerimientos especiales que deben atenderse con urgencia (PF o Fallos de Página)
- ✓ Los movimientos utilizados para atender los PF se cuentan
- ✓ Los fallos de página indican mayor prioridad respecto a requerimientos convencionales:
 - ✓ En FCFS: Se atiende el PF instantáneamente y luego se sigue en orden FCFS
 - ✓ En SSTF: Se atiende el PF instantáneamente y luego se sigue con el requerimiento que menor tiempo de seek genere a partir del PF



Algoritmos – Atención de Fallos de Página (cont.)



Existen requerimientos especiales que deben atenderse con urgencia (PF o Fallos de Página):

- ✓ En SCAN: Se atiende el PF instantáneamente inclusive si esto implica cambiar el sentido de giro:
 - ♦ Si hubo que cambiar el sentido de giro, una vez atendido el PF se sigue barriendo los requerimientos con el nuevo sentido de giro (cambia el sentido)
- ✓ En C-SCAN: Se atiende el PF instantáneamente inclusive si esto implica cambiar el sentido de giro:
 - ♦ Si hubo que cambiar el sentido de giro, una vez atendido el PF se vuelve al sentido de giro original (no cambia el sentido)
- ✓ En LOOK: Idem a SCAN
- ✓ En C-Look: Idem a C-SCAN



Algoritmos – Atención de Fallos de Página (cont.)

- ✓ Suponga un disco rígido con 100 pistas (0..99), donde la cabeza se encuentra en la pista 20 y viene de la 18. Sea la siguiente la secuencia de atención a requerimientos:
{ 55, 75, 25^{PF}, 45, 10 }. Luego de 30 movimientos entra
{ 52^{PF}, 60 } y luego de 10 movimientos mas entra
{ 90, 10 }
- ✓ Realice los diagramas de planificación de disco teniendo en cuenta los siguientes algoritmos. Indique para cada uno la cantidad total de movimientos:
- ✓ FCFS
- ✓ SSFT
- ✓ C-LOOK
- ✓ SCAN



Algoritmos – Atención de Fallos de Página (cont.)

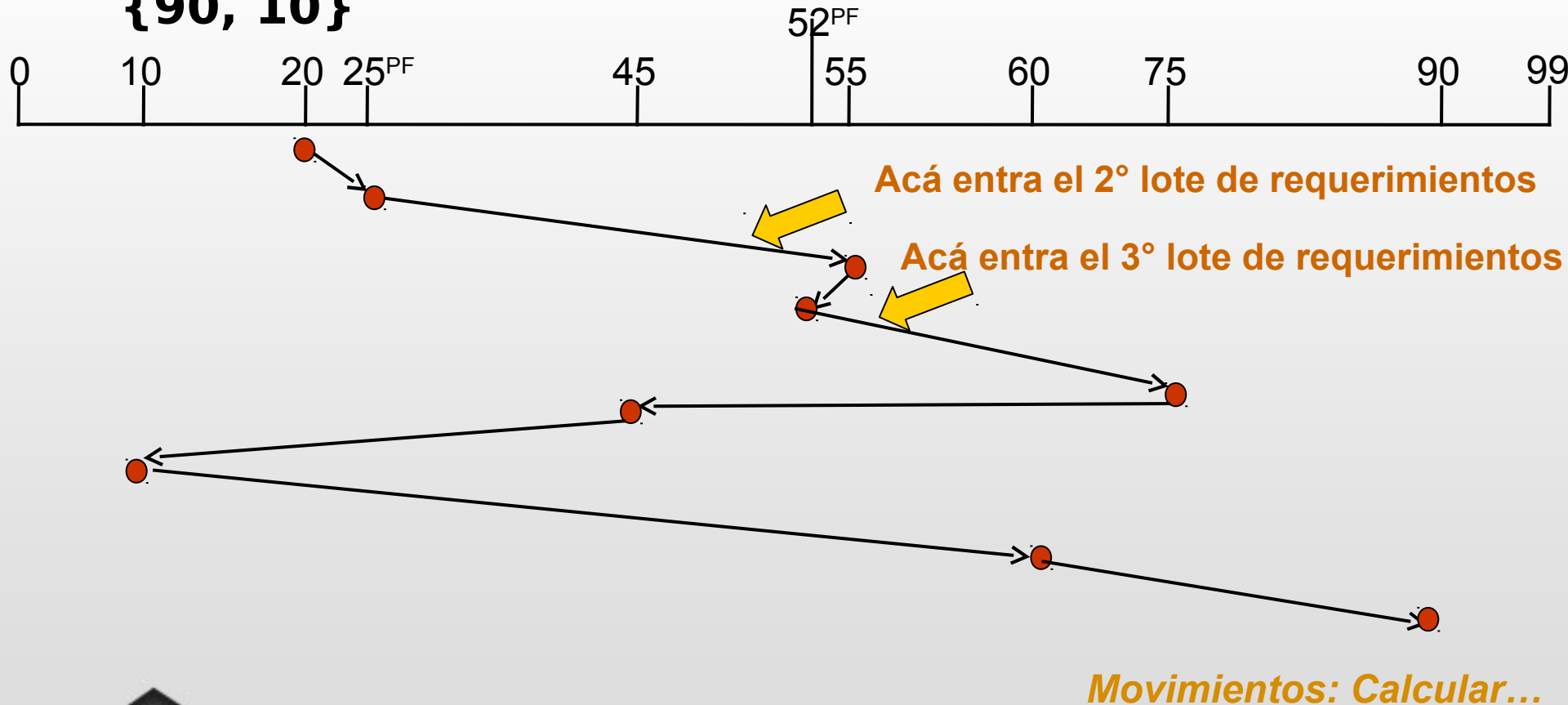
- ✓ La atención de requerimientos a pistas duplicadas se resuelven según el algoritmo de planificación
- ✓ FCFS: Se atienden de manera separada (tantas veces como se requieran)
- ✓ SSFT/SCAN/LOOK/C-SCAN/C-LOOK: se atienden una única vez, ya que los requerimientos se ordenan
 - ✓ Tener en cuenta que si hay más de un lote de requerimientos, podría ocurrir que haya que atender más de una vez un mismo requerimiento (en caso de no contar con el segundo lote en la primera atención)



Algoritmos - Ejemplo - FCFS



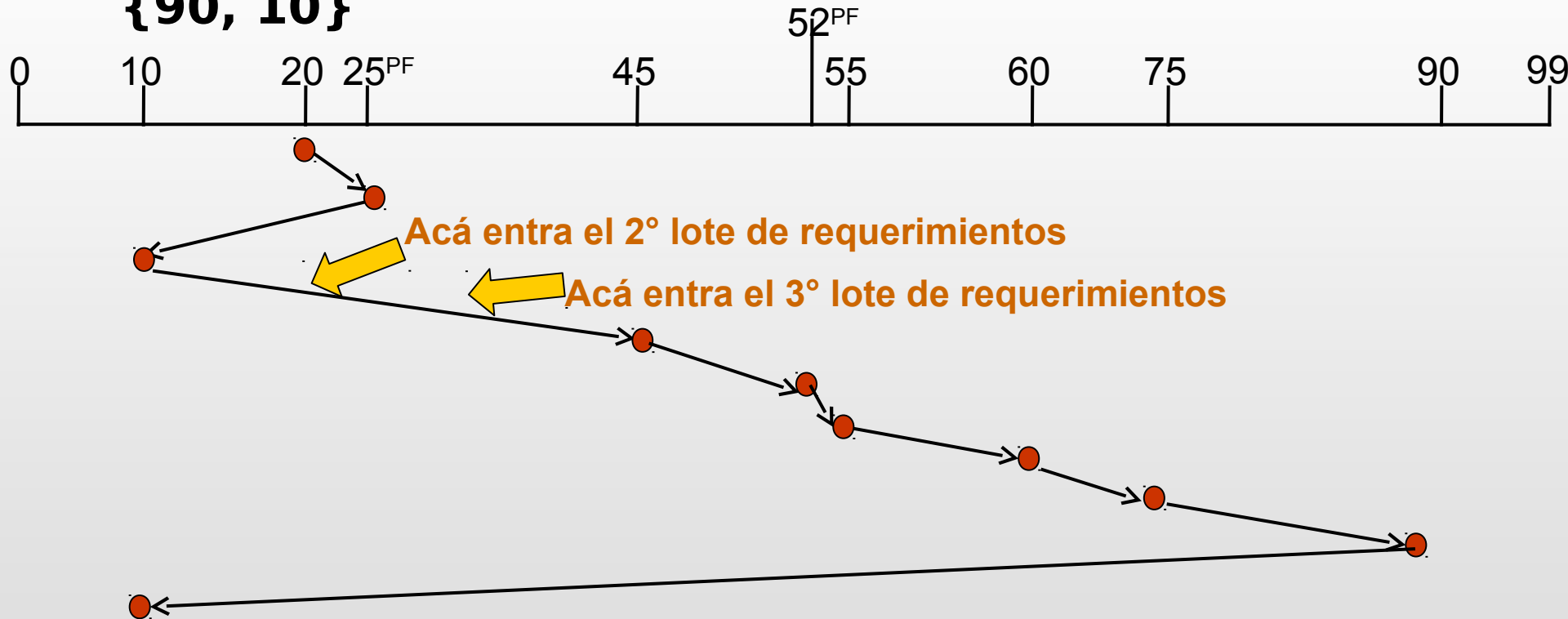
{ 55, 75, 25^{PF}, 45, 10 }. Luego de 30 movimientos entra **{ 52^{PF}, 60 }** y luego de 10 movimientos mas entra **{ 90, 10 }**



Algoritmos - Ejemplo - SSTF



{ 55, 75, 25^{PF}, 45, 10 }. Luego de 30 movimientos entra **{ 52^{PF}, 60 }** y luego de 10 movimientos mas entra **{ 90, 10 }**



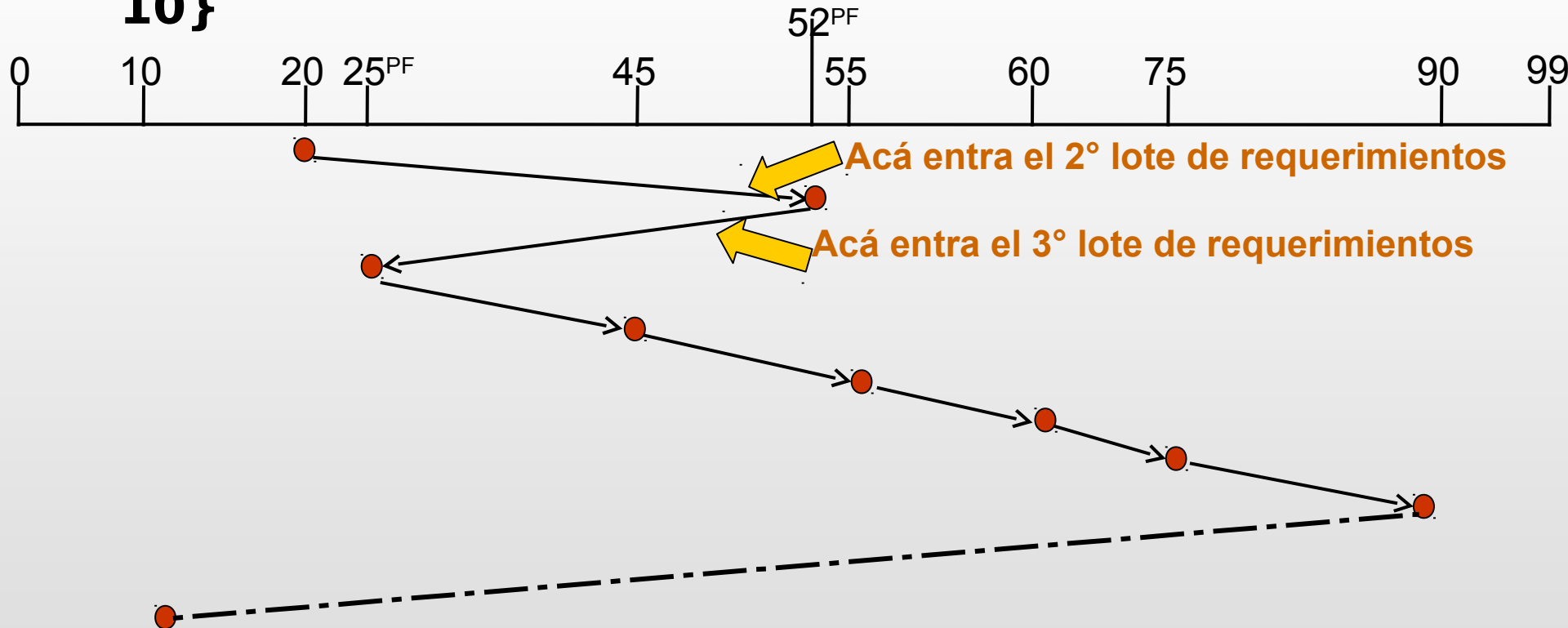
Movimientos: Calcular...



Algoritmos - Ejemplo - C-LOOK



{ 55, 75, **52^{PF}**, 45, 10 }. Luego de 30 movimientos entra {**25^{PF}**, 60 } y luego de 10 movimientos mas entra {90, 10}



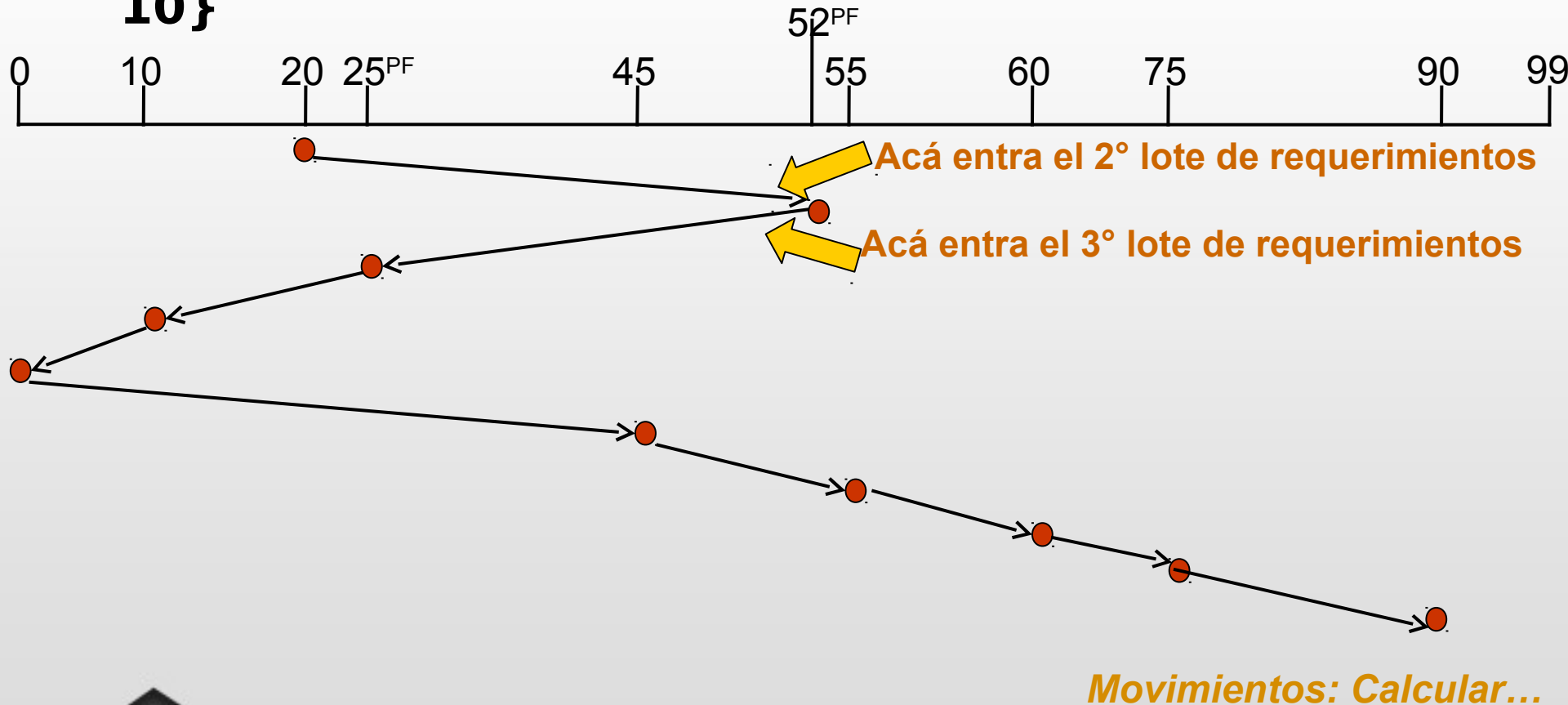
Movimientos: Calcular...



Algoritmos - Ejemplo - SCAN

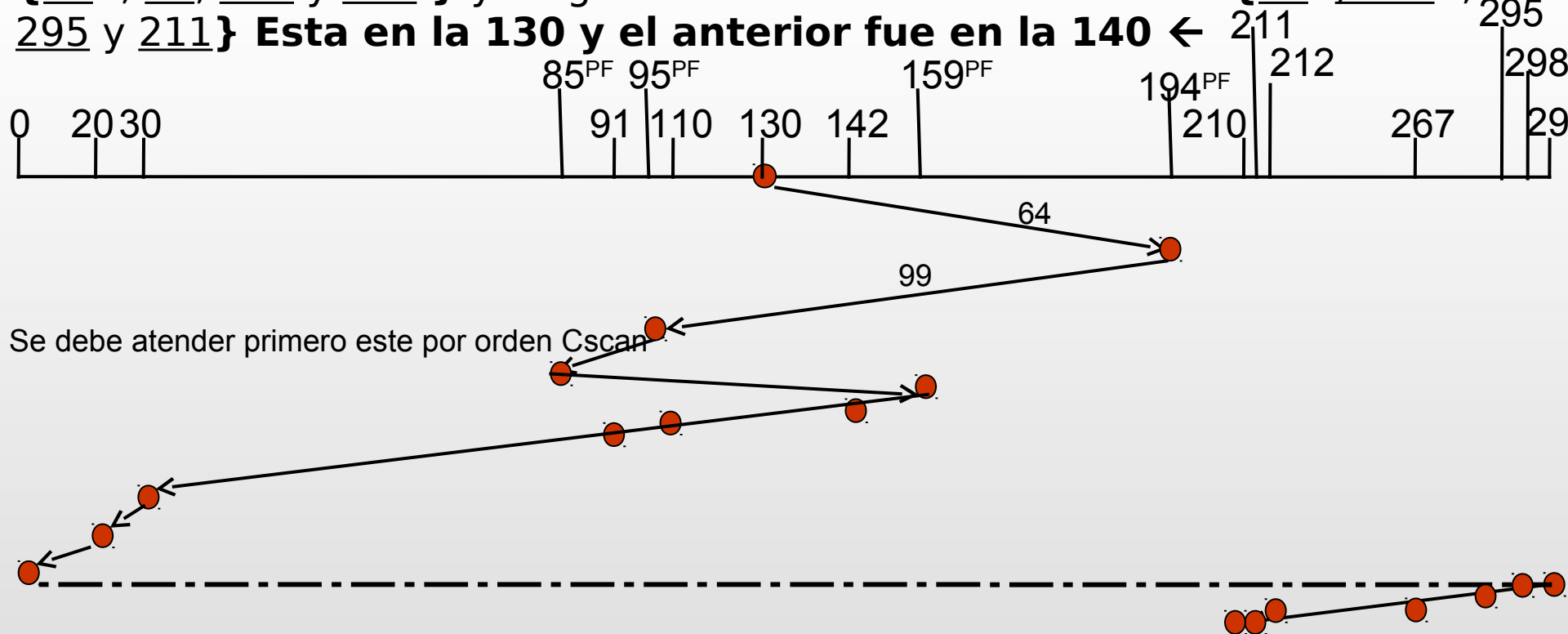


{ 55, 75, **52^{PF}**, 45, 10 }. Luego de 30 movimientos entra {**25^{PF}**, 60 } y luego de 10 movimientos mas entra {**90**, 10}



C - SCAN

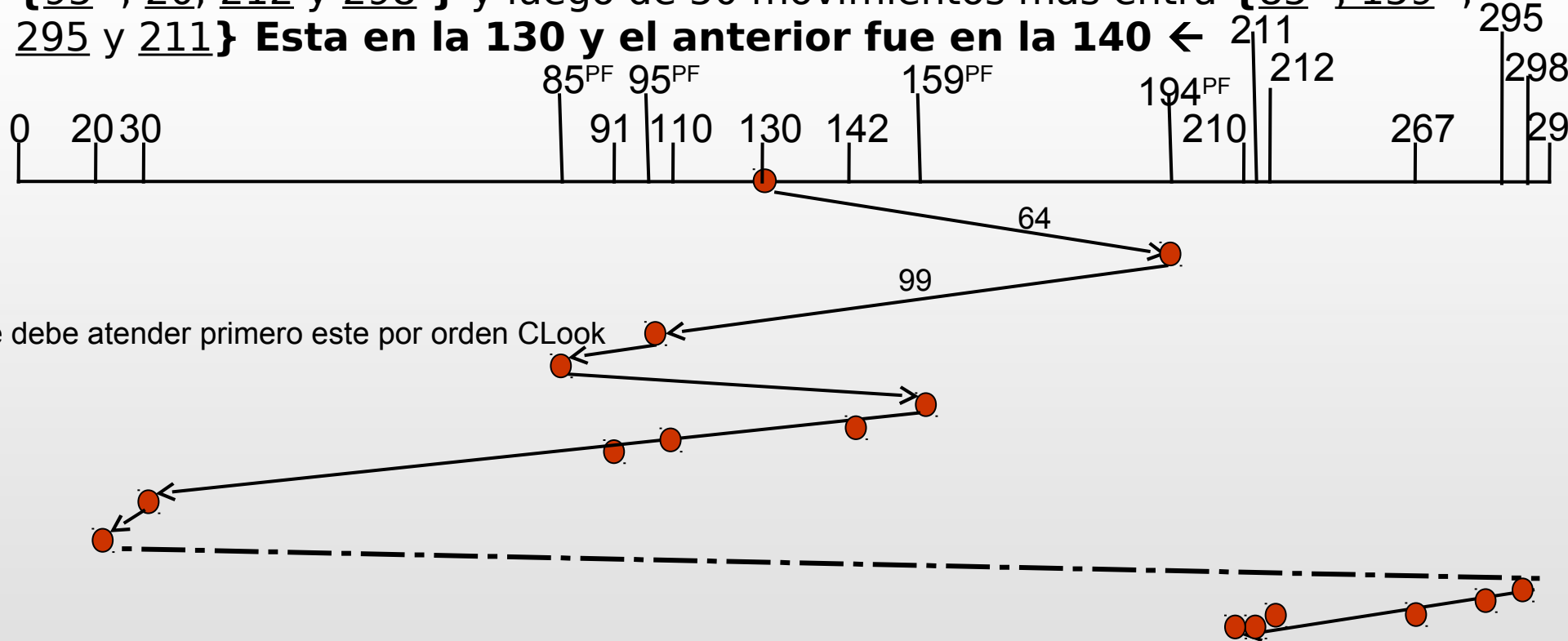
{30, 110, 142, 267, 91, 210, 194^{PF}, 20 }. Luego de 20 movimientos entra {95^{PF}, 20, 212 y 298 } y luego de 50 movimientos mas entra {85^{PF}, 159^{PF}, 295 y 211} **Esta en la 130 y el anterior fue en la 140** ←



Movimientos: 495



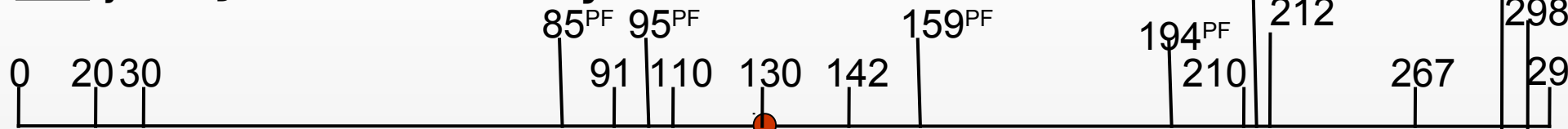
{ 30, 110, 142, 267, 91, 210, 194^{PF}, 20 }. Luego de 20 movimientos entra { 95^{PF}, 20, 212 y 298 } y luego de 50 movimientos mas entra { 85^{PF}, 159^{PF}, 295 y 211 } **Esta en la 130 y el anterior fue en la 140** ←



Movimientos: 474



{ 30, 110, 142, 267, 91, 210, 194^{PF}, 20 }. Luego de 20 movimientos entra { 95^{PF}, 20, 212 y 298 } y luego de 50 movimientos mas entra { 85^{PF}, 159^{PF}, 295 y 211 } **Esta en la 130 y el anterior fue en la 140** ← 211 295

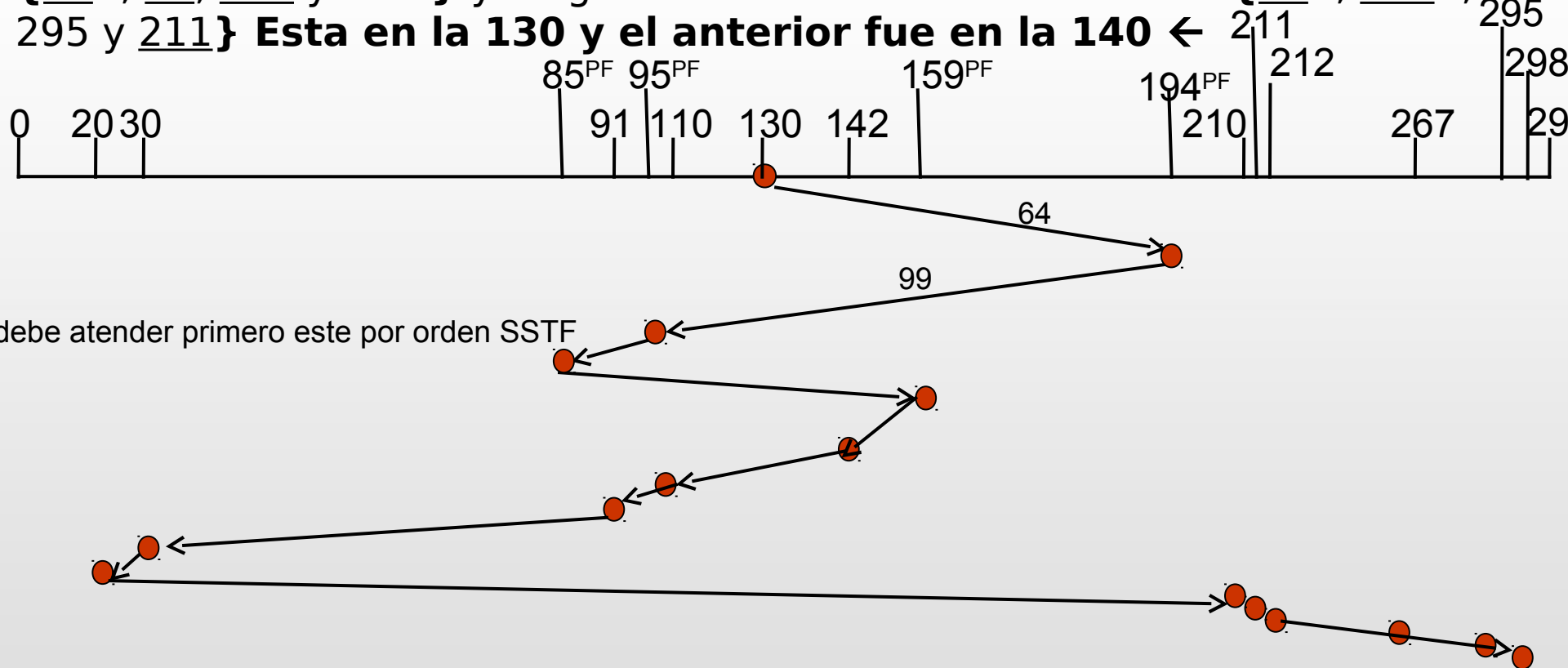


debe atender primero este por orden FCFS

20 una sola vez porque estan pegados



{30, 110, 142, 267, 91, 210, 194^{PF}, 20 }. Luego de 20 movimientos entra {95^{PF}, 20, 212 y 298 } y luego de 50 movimientos mas entra {85^{PF}, 159^{PF}, 295 y 211} **Esta en la 130 y el anterior fue en la 140** ←



Movimientos: 664



