Sistemas Operativos

Estructuras de dispositivos masivos de datos

Curso 2024

Facultad de Ingeniería, UDELAR

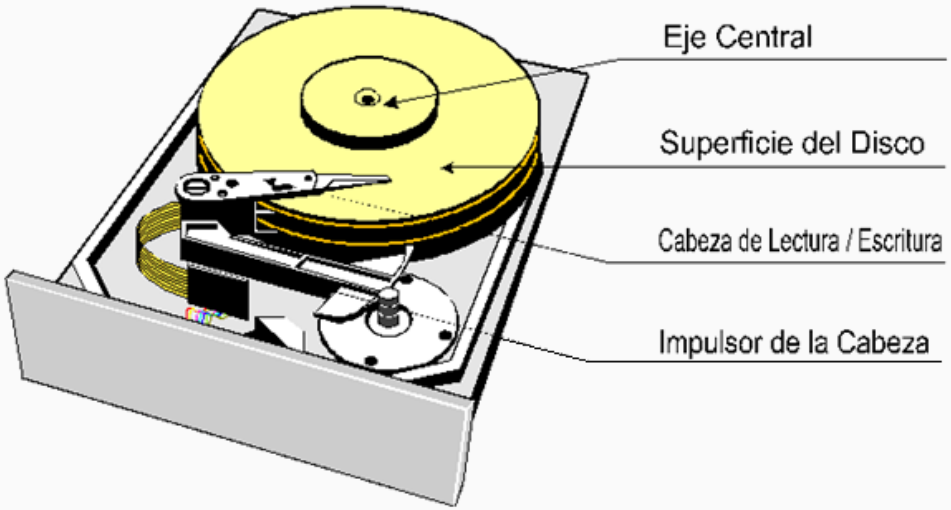
Agenda

1. Estructura de almacenamiento masivo
2. Acceso al almacenamiento
3. Planificación de disco
4. Manejo de discos
5. Estructuras RAID

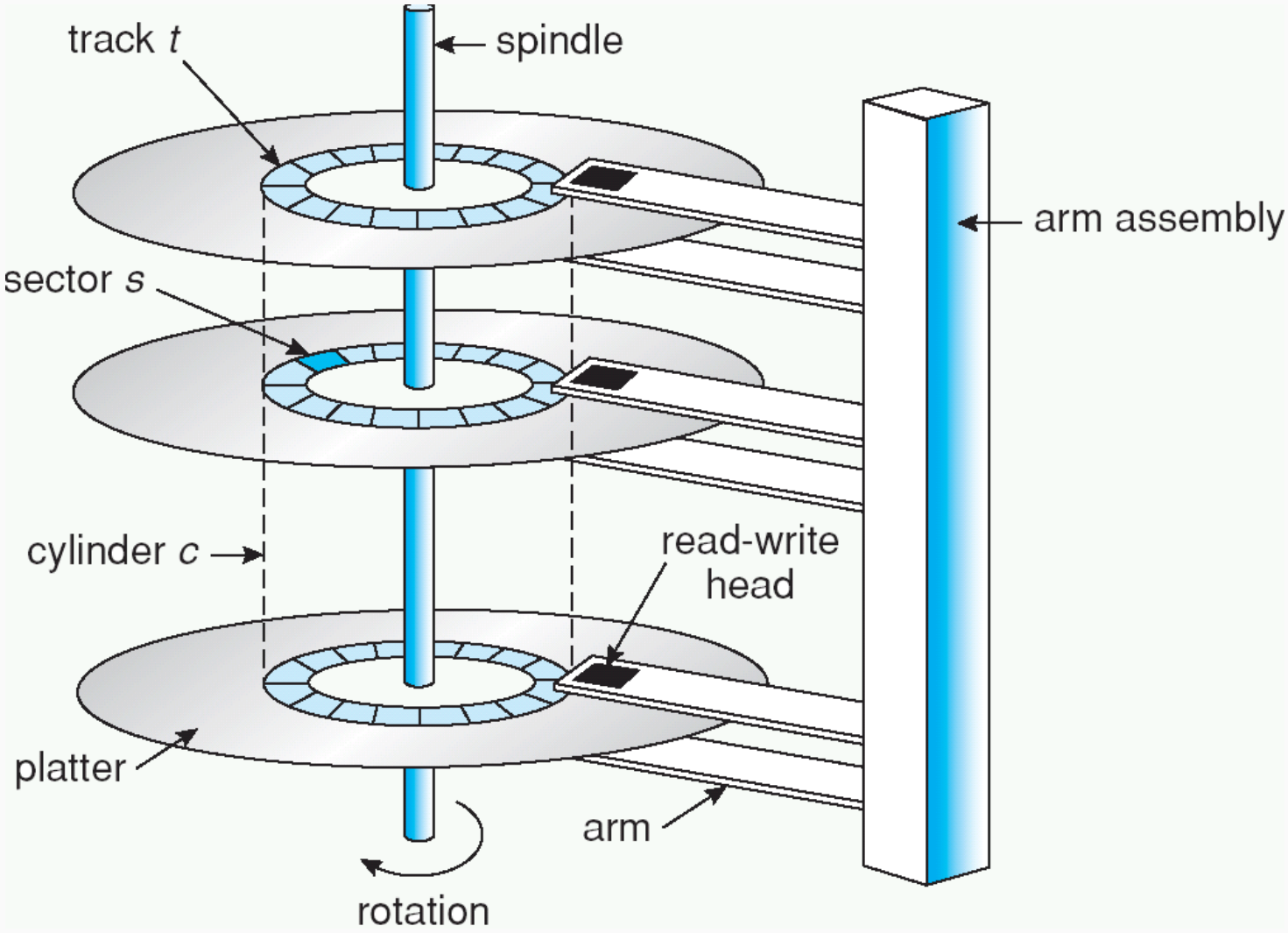
Estructura de almacenamiento masivo

Discos

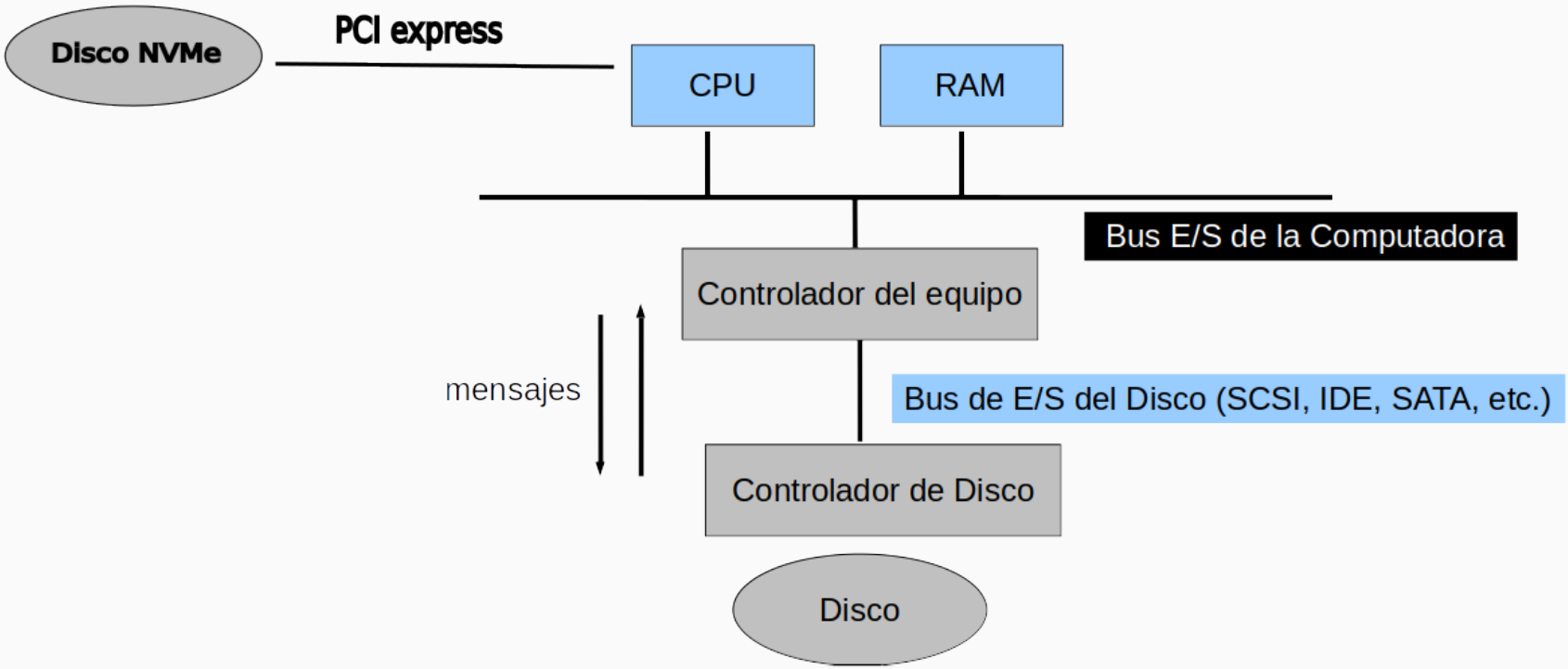
- Dispositivos para almacenamiento no volátil
- Son dispositivos electromecánicos (HDD), optomecánicos (CD-ROM y DVD) o de estado sólido (SSD). Se acceden a nivel de bloques por el sistema de archivos.



Esquema HDD


track t
spindle
arm assembly
sector s
cylinder c
read-write
head
platter
arm
rotation

Conexión de discos



Cintas

- Relativamente permanente y almacena una gran cantidad de datos
- Tiempo de acceso lento
- Acceso aleatorio 1000 veces más lentos que discos
- Se usa para backup
- Tamaños de varios terabytes en modo comprimido
- Tecnología LTO-5 / SDLT

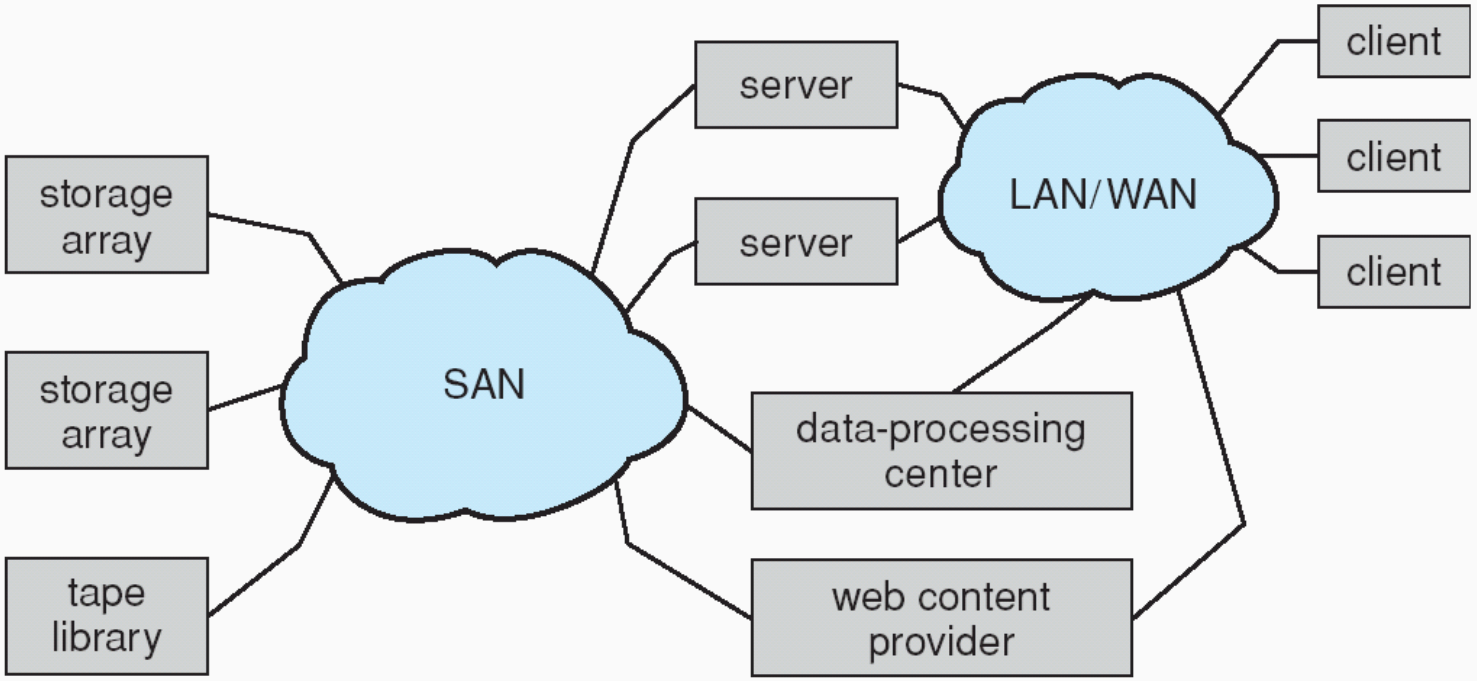
Acceso al almacenamiento

Conexión directa

- Se accede al almacenamiento a través del bus de E/S
- Internos al PC
  - SCSI
  - SATA
- Externos
  - USB
  - Thunderbolt

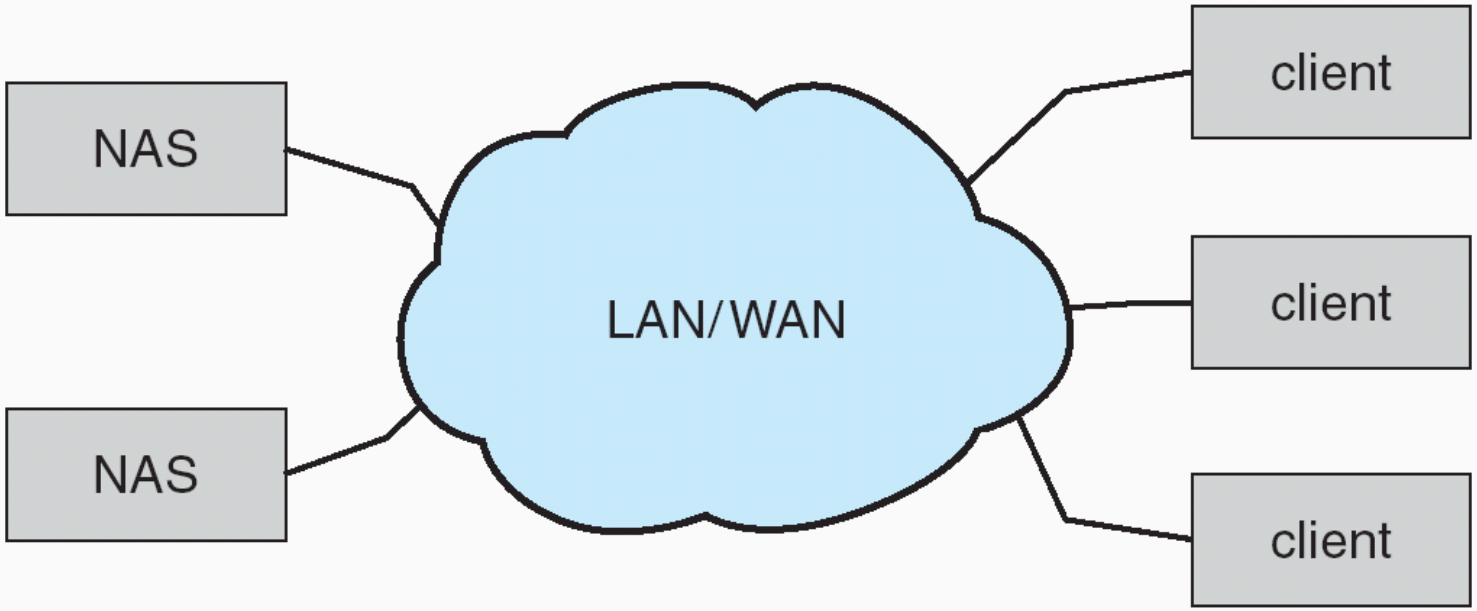
Storage Area Network (SAN)

- Muy común en ambientes grandes



Network Attached Storage (NAS)

- Se accede al almacenamiento a través de la red y no a través de una conexión local (BUS)
- Implementado a través de RPC entre host y storage
- Protocolo iSCSI usa el protocolo TCP o UDP sobre una red IP para transportar el protocolo SCSI



Planificación de disco

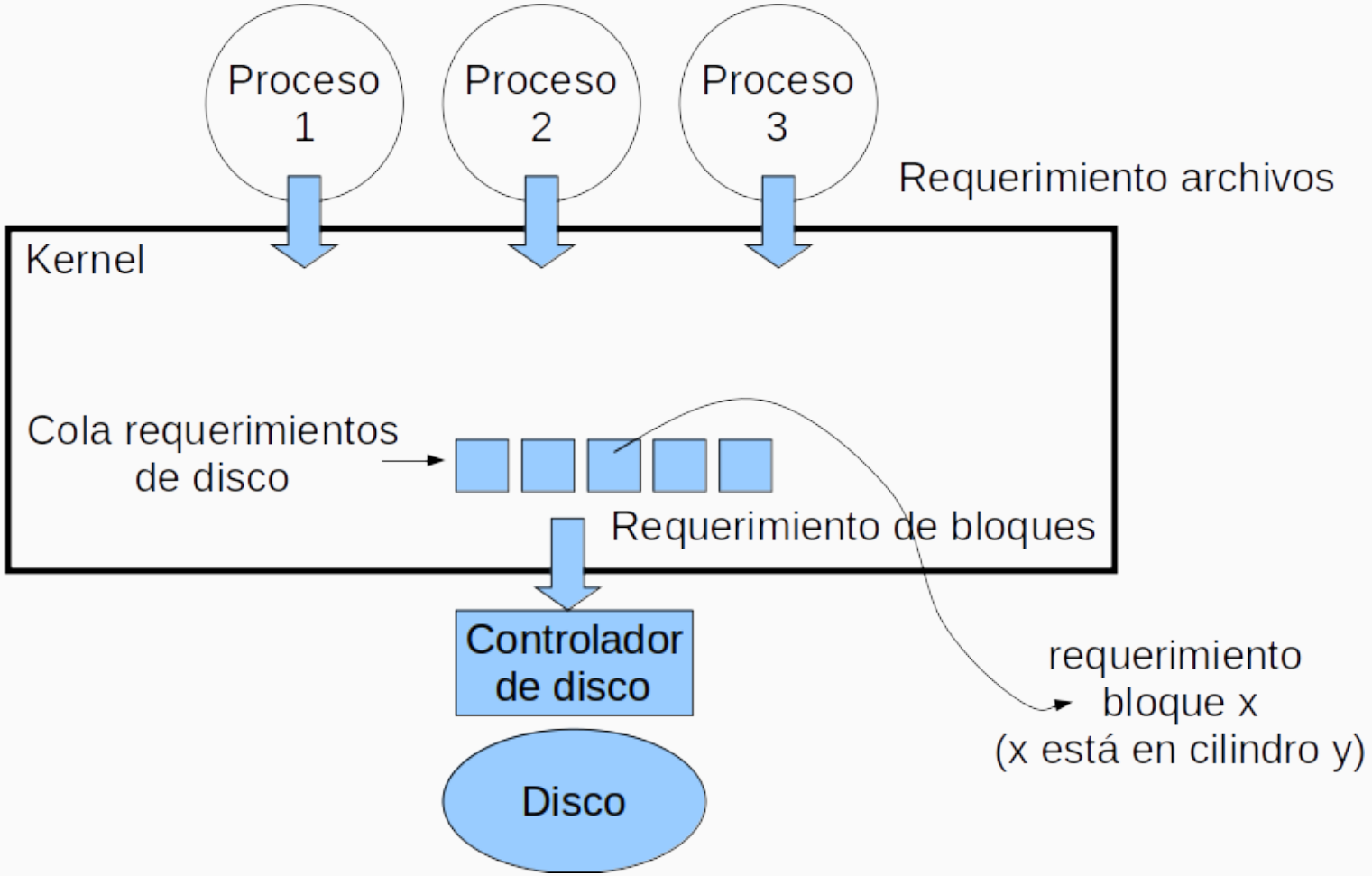
Planificación de disco

- El sistema operativo es responsable de usar el hardware de forma eficiente. Desde la perspectiva del disco, esto significa obtener un rápido acceso a los datos y aprovechar al máximo el ancho de banda al disco.
- Es por eso que el planificador de disco es uno de los más importantes.
- El acceso a disco tiene dos grandes componentes:
  - Tiempo de posicionamiento (*seek time*): Es el tiempo que el brazo del disco necesita para posicionar la cabeza en el cilindro que contiene el sector.
  - Latencia de rotación (*rotational latency*): Es el tiempo que demora rotar el plato al sector correcto.
- Tiempo de acceso = Tiempo de posicionamiento + Tiempo de rotación + Tiempo de transferencia

Planificación de disco

- El ancho de banda (*bandwidth*) de un disco es la cantidad de bytes transferidos, dividido por tiempo total de la transferencia (desde el comienzo del pedido hasta la última transferencia).
- Ejemplo
  - tiempos de disco: 7200 rpm = 1 revolución cada 60/7200 segundos $\Rightarrow$ 1 revolución cada 8,333... milisegundos
  - Latencia promedio $\frac{1}{2}$ revolución $\Rightarrow$ 4,16666... milisegundos

Planificación de disco



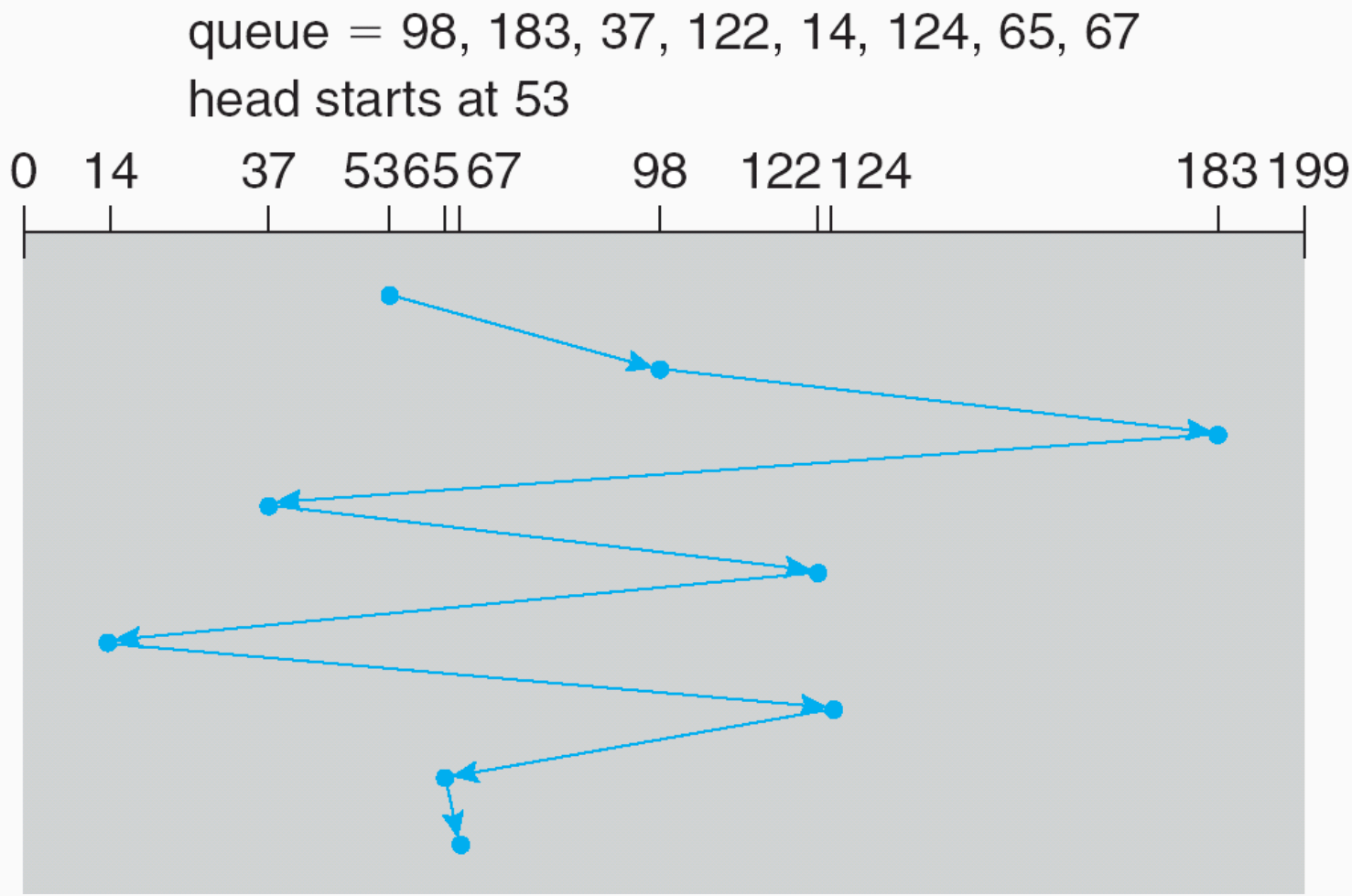
Planificación de disco

- Para mejorar el acceso a los datos se debe minimizar el tiempo de búsqueda. De esa forma, surgen varios métodos de planificación de disco:
  - FCFS.
  - SSTF.
  - SCAN.
  - C-SCAN.
  - LOOK.
  - C-LOOK.

FCFS - First-Come, First-Served

- En este caso la planificación es realizar los pedidos como vayan llegando.
- En un disco con 200 cilindros (numerados del 0 al 199), la cabeza posicionada en el cilindro 53 y una lista de pedidos:
  - 98, 183, 37, 122, 14, 124, 65, 67
- Genera un movimiento de la cabeza sobre 640 cilindros.

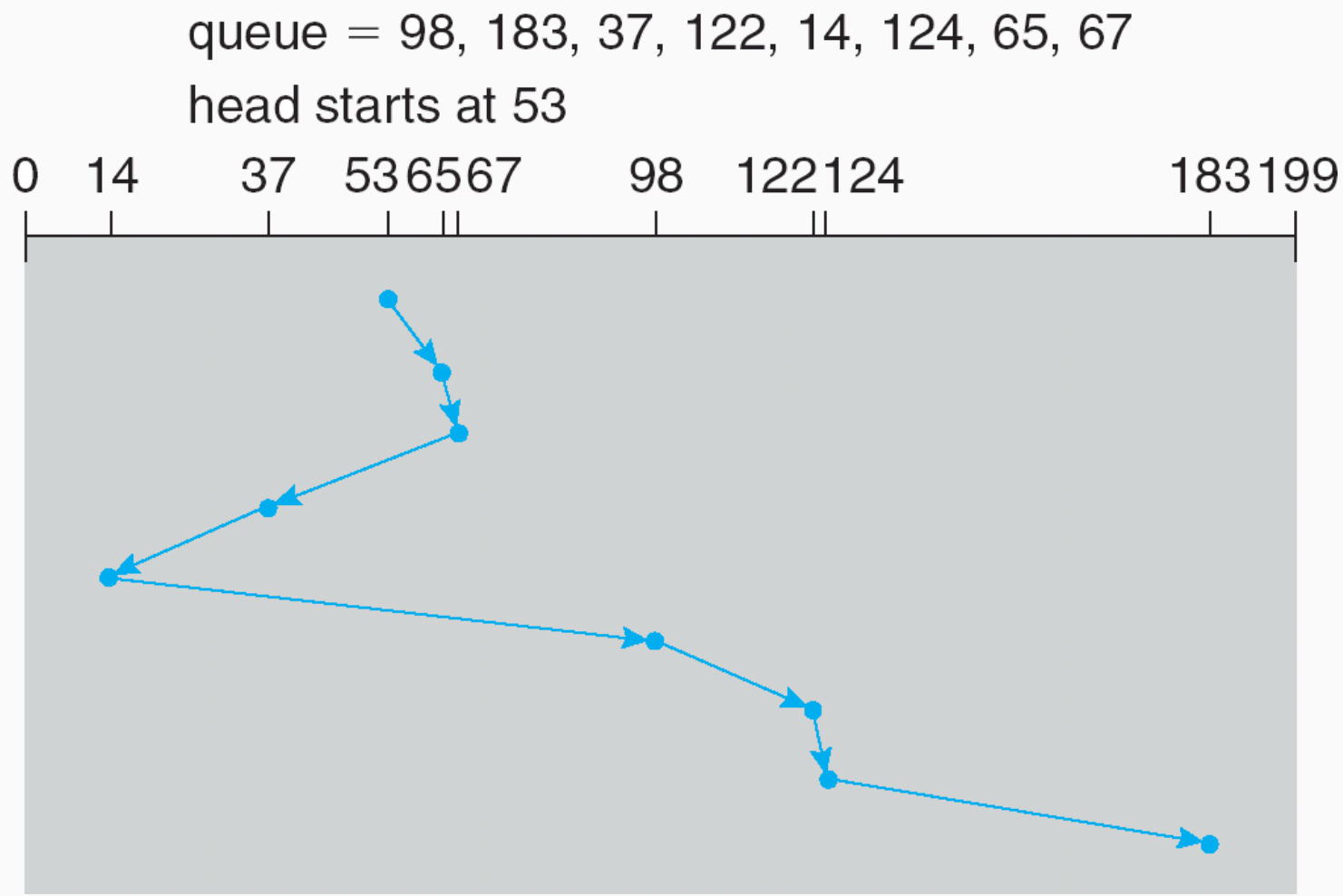
FCFS - First-Come, First-Served



SSTF – Shortest Seek Time First

- Se elige como próximo pedido a realizar el que genere menos tiempo de búsqueda (*seek time*).
- Esto puede generar que pedidos nunca sean ejecutados o demorados mucho tiempo.
- Para el ejemplo anterior tenemos que la cabeza recorrerá un total de 236 cilindros.

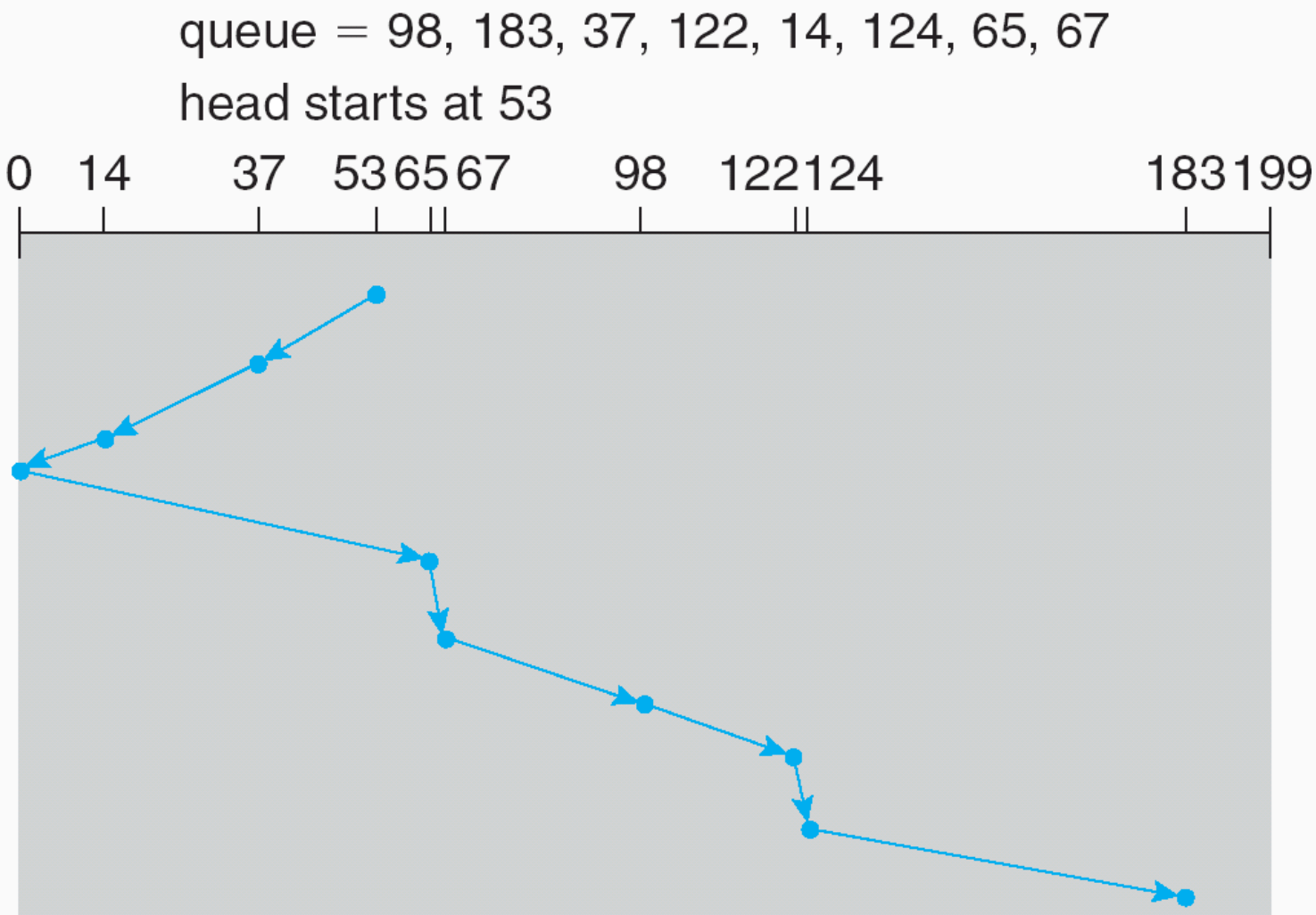
SSTF – Shortest Seek Time First



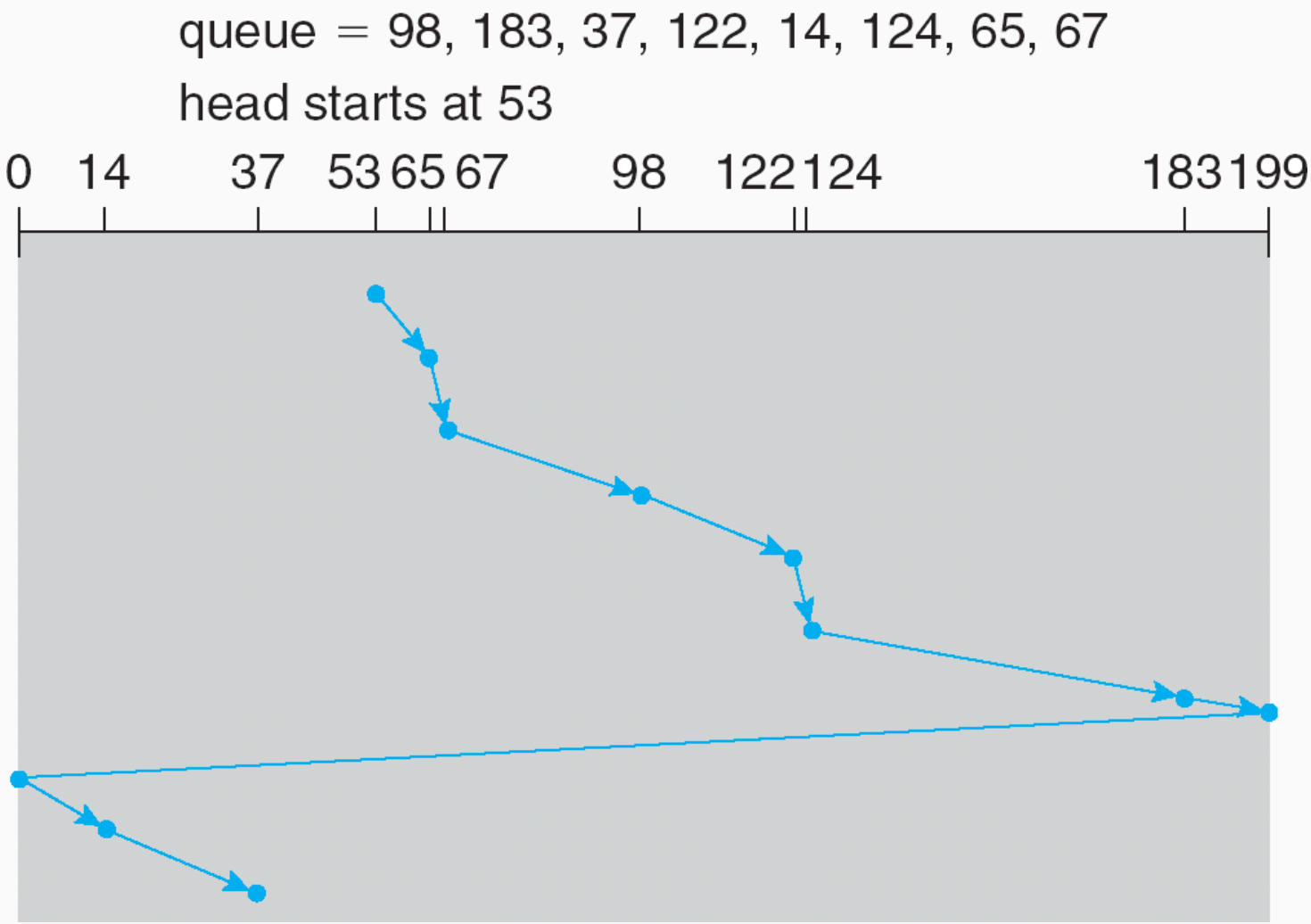
SCAN y C-SCAN

- SCAN
  - El brazo posiciona la cabeza al comienzo del disco y la mueve hacia el otro extremo, resolviendo los pedidos mientras pasa por los cilindros. Al llegar al final, hace el camino inverso resolviendo las solicitudes.
  - Es también llamado el algoritmo del elevador.
  - En este caso también se recorre un total de 236 cilindros.
- C-SCAN
  - Es igual que el SCAN, pero al llegar al final del disco en la recorrida, vuelve al principio inmediatamente sin resolver ningún pedido.

SCAN

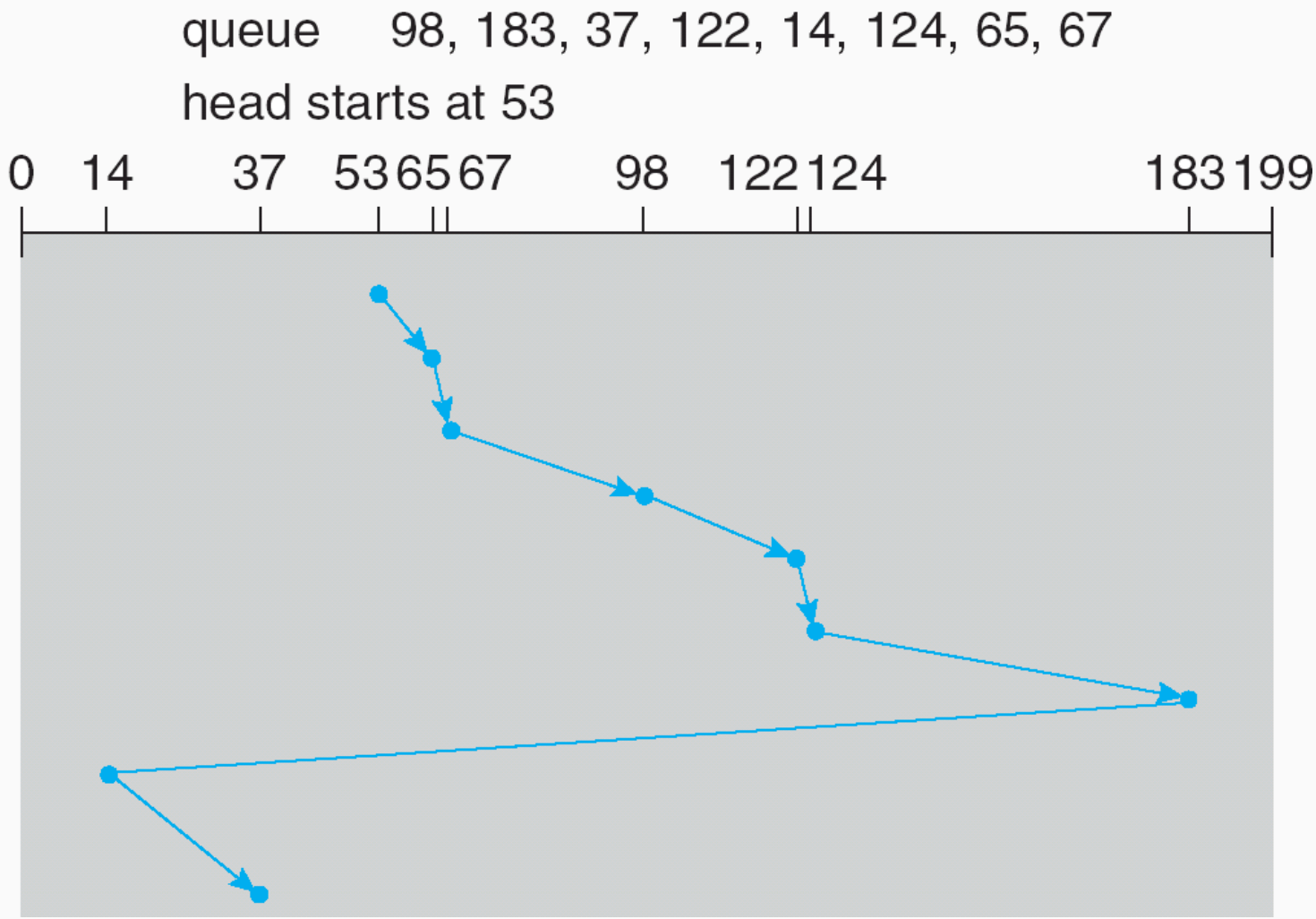

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

C-SCAN


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

C-LOOK

- Es parecido al C-SCAN, pero el brazo va hasta donde haya pedidos y retorna hacia el otro sentido.



Conclusiones

- SSTF y LOOK son los más utilizados.
- SCAN y C-SCAN han demostrado ser mejores ante sistemas que tienen una alta carga en disco.
- El servicio de disco es muy dependiente del método de asignación de archivos.
- El algoritmo de planificación del disco debe estar escrito como un módulo del sistema operativo, permitiendo cambiarlo si es necesario

Planificación de discos SSD

- Los algoritmos anteriores no aplican para discos SSD dado que no tienen partes móviles
- En general un algoritmo FCFS es suficiente para discos SSD (uniendo las escrituras contiguas en el mismo sector)

TRIM

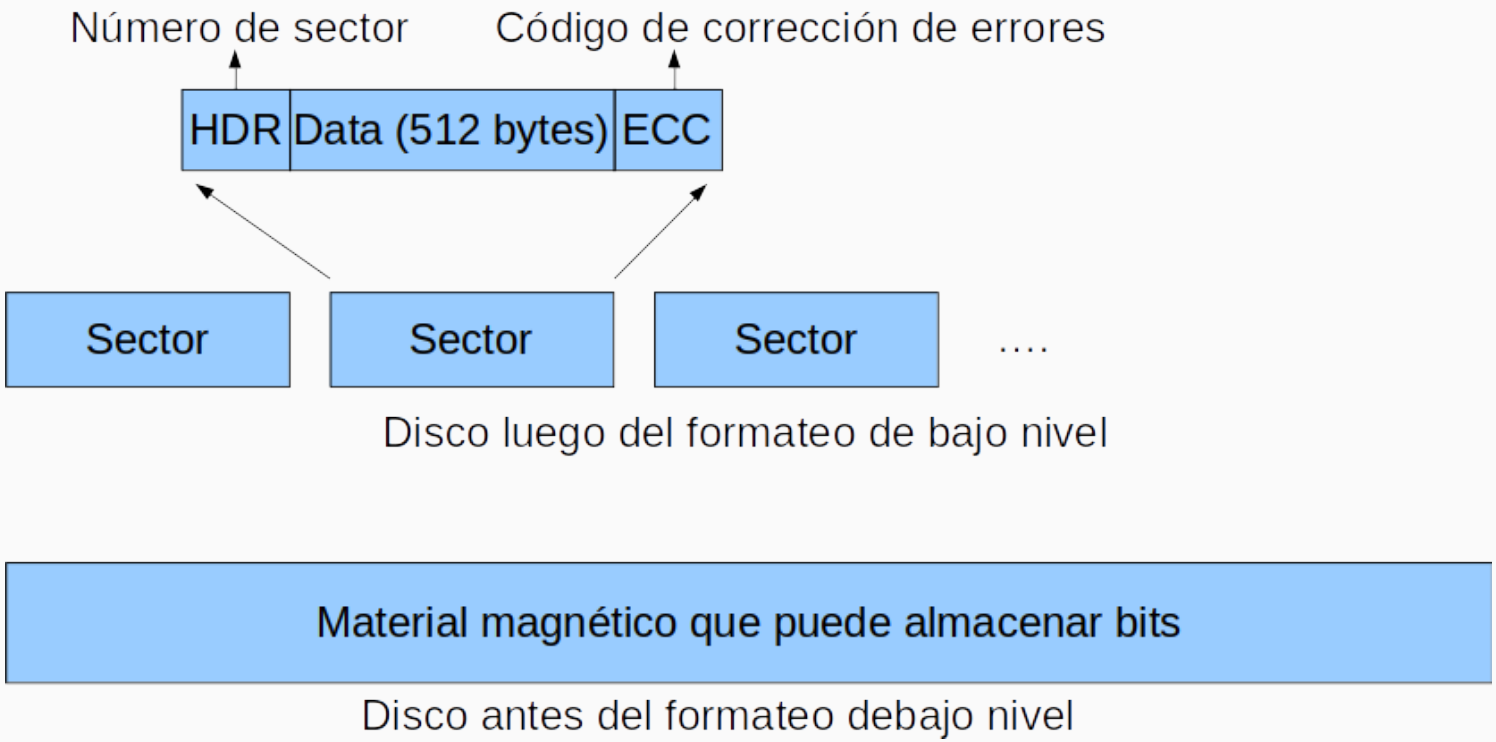
- Los sectores de los discos SSD no se pueden sobreescribir, hay que borrarlos primero y el borrado es la operación más lenta
- Al sobreescribir un bloque muchas veces se marca al original como inválido y se escribe uno nuevo
- Cuando no hay espacio libre hay que hacer garbage collection y borrar los bloques con datos inválidos
- Para optimizar esto es útil que el filesystem le indique al disco cuando hay sectores que se borraron (solo se marcaron como libres en el sistema pero no se borraron)
- Esta operación es conocida como TRIM y permite que el disco libere más bloques en el garbage collection.
- Esta operación se realiza en general cuando el disco no tiene actividad para que no impacte el rendimiento.

Manejo de discos

Manejo de discos

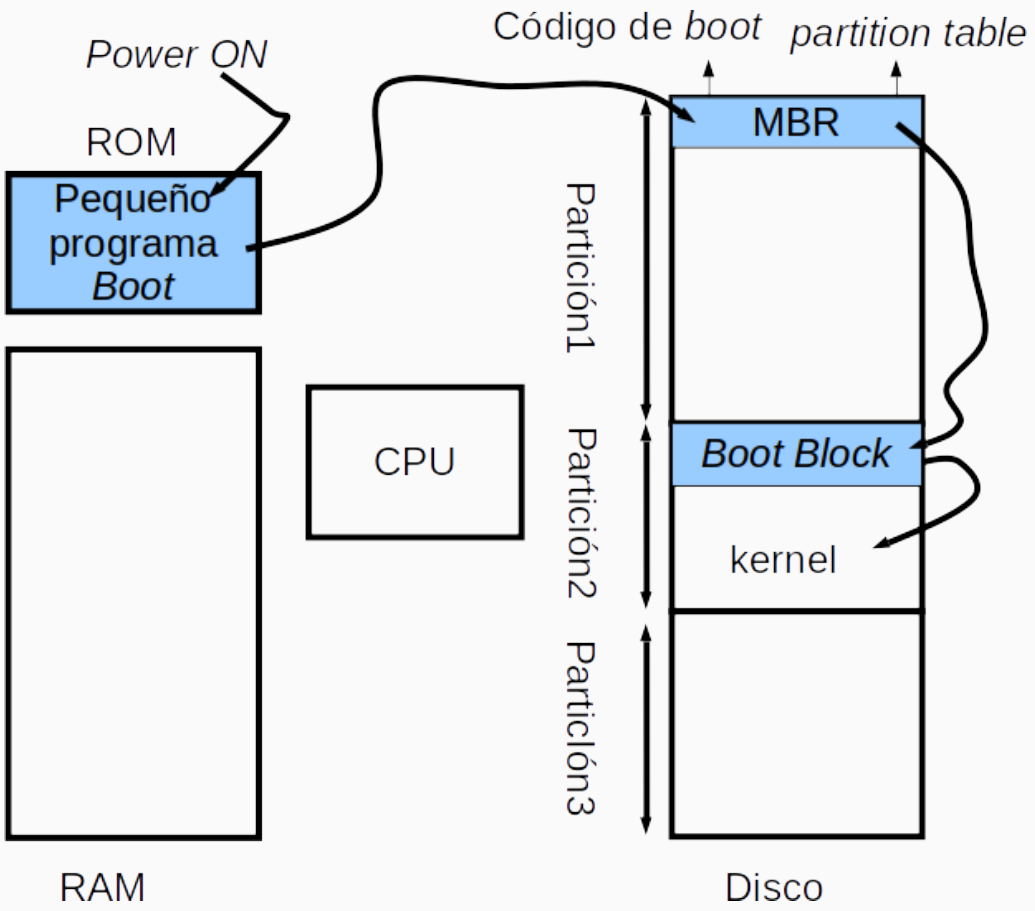
- Formateo del disco: formateo de bajo nivel o formateo físico.
  - Dividiendo el disco en sectores que el controlador de discos pueda leer o escribir.
- Para usar el disco para almacenar archivos el sistema operativo necesita guardar su propia estructura en el disco
  - Particionamiento del disco en un o más grupos de cilindros
  - Formateo lógico o “armado de un sistema de archivos”

Manejo de discos



Sector de Boot

- El código de boot es ejecutado en ROM, este trae el MBR a memoria y empieza a ejecutarlo
- Ejecuta el MBR, mira la tabla de particionamiento, aprende la partición boot, trae y ejecuta la partición de boot
- El código de boot en el bloque de boot carga el kernel que está en esa partición.



Estructuras RAID

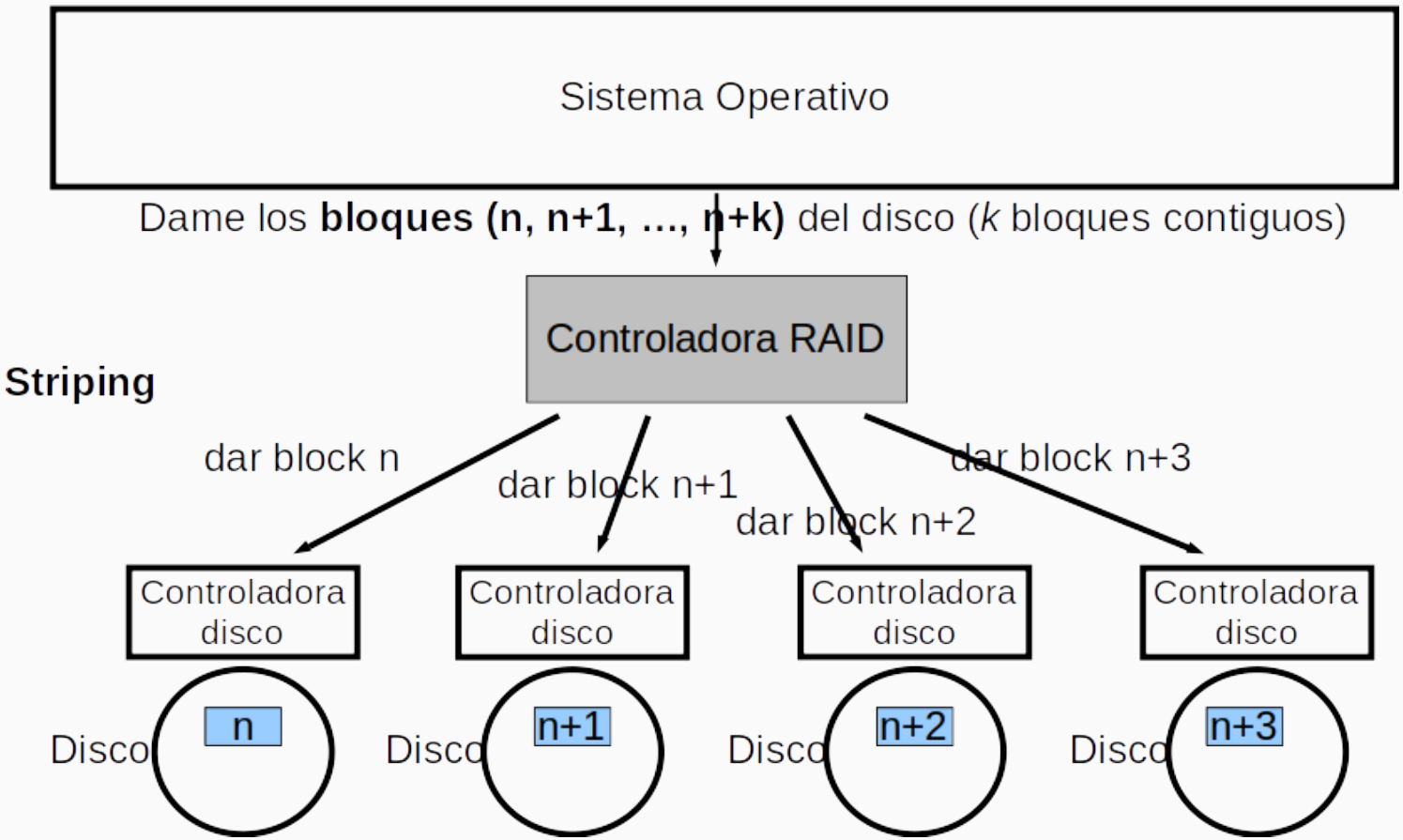
Estructuras RAID

- Los sistemas de almacenamiento de disco (*storage*) presentan tecnologías orientadas a mejorar el servicio.
- Las mejoras implementadas se basan en dar:
  - Confiabilidad: Casos de fallo de discos.
  - Performance: Lograr mejorar los tiempos de transferencia.
- Las técnicas denominadas RAID (Redundant Array of ~~Inexpensive~~ Independent Disks) tienen una amplia aceptación.
- RAID no un backup (`https://www.raidisnotabackup.com`)

Estructuras RAID

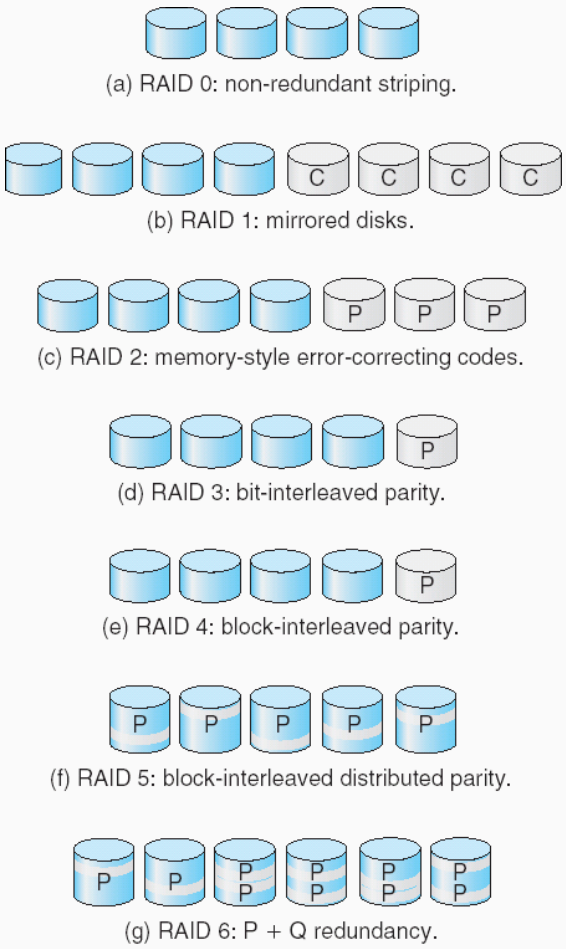
- La confiabilidad es lograda a través de redundancia de la información.
- La redundancia puede darse duplicando discos (mirror), o a través de bits de control.
- La mejora de los tiempo de respuesta (*performance*) se logra a través de la disposición de la información en los diferentes discos.
- Técnicas de striping son utilizadas a nivel de bit, byte, sectores, bloque de sectores y bloque.

Estructuras RAID



Diferentes niveles de RAID

- RAID 0.
- RAID 1.
- RAID 2.
- RAID 3.
- RAID 4.
- RAID 5.
- RAID 6.



(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 2: memory-style error-correcting codes.

(d) RAID 3: bit-interleaved parity.

(e) RAID 4: block-interleaved parity.

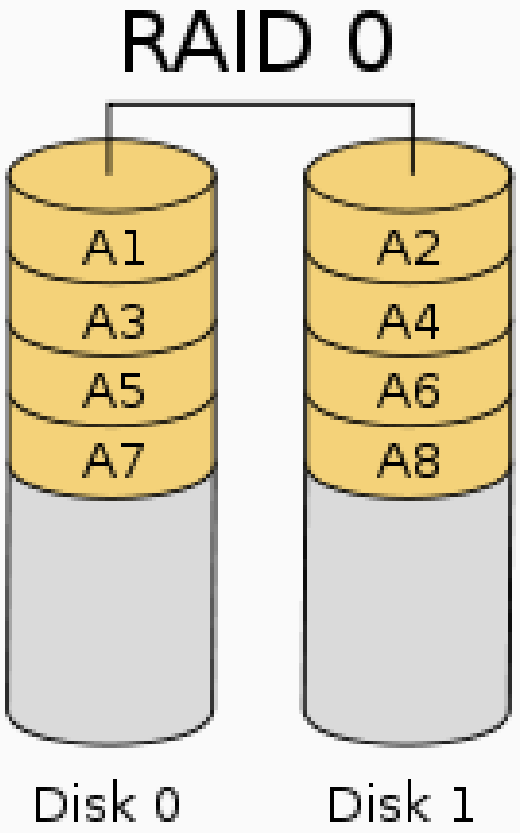
(f) RAID 5: block-interleaved distributed parity.

(g) RAID 6: P + Q redundancy.

RAID 0

- Un RAID 0 (*stripe set* o *striped volume*) divide los datos de forma homogénea en dos o más discos, sin información de paridad para la redundancia.
- El RAID 0 no era uno de los niveles RAID originales y no proporciona redundancia de datos.
- RAID 0 se utiliza normalmente para aumentar el rendimiento, aunque también se puede utilizar como una forma de crear grandes discos virtuales de un gran número de pequeñas unidades físicas.

RAID 0 - Diagrama



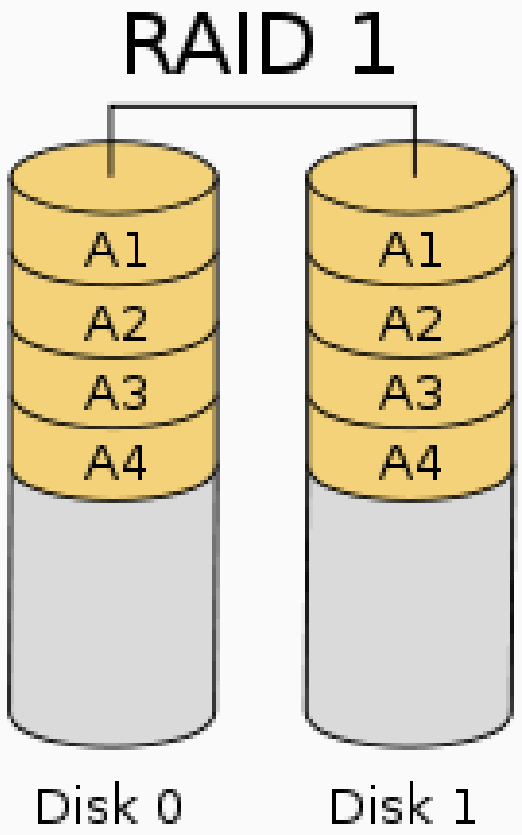
RAID 0 - Rendimiento

- Si bien el tamaño del bloque técnico, pueden ser tan pequeño como un byte, es casi siempre un múltiplo del tamaño del disco duro de sector de 512 bytes.
- Si los sectores accedidos se distribuyan de forma equilibrada entre las N unidades, el acceso aleatorio será N veces mas rápido.
- La velocidad de transferencia del sistema completo será la velocidad de transferencia de todos los discos sumados, limitado sólo por la velocidad de la controladora RAID (no se puede aumentar indefinidamente la cantidad de discos)

RAID 1

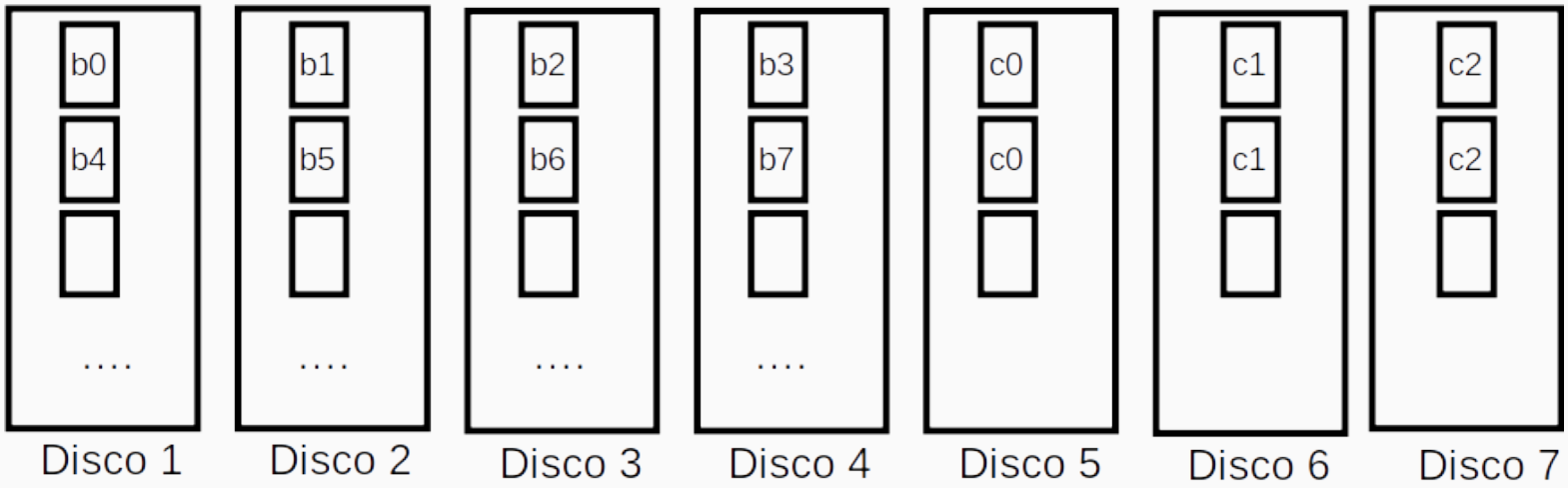
- Un RAID 1 crea una copia exacta (o espejo) de un conjunto de datos en dos o más discos.
- Esto es útil cuando el rendimiento de lectura o la confiabilidad son más importantes que la capacidad de almacenamiento de datos.
- Este tipo de arreglo sólo puede ser tan grande como el disco más pequeño. Un clásico RAID 1 contiene dos discos duplicados pero podrían ser más.
- Puesto que cada miembro contiene una copia completa de los datos, que pueden tratarse de forma independiente, la fiabilidad es incrementada por la potencia del número de copias.
- Protección eficaz contra la falla de disco físico, no contra la corrupción de datos debido a virus, cambios o eliminaciones accidentales de archivos, etc.

RAID 1 - Diagrama



RAID 2

- Organización de corrección de errores similar al de memoria
- ECC (*Error Correcting Code*):
  - Cada byte en memoria puede tener un bit de paridad
  - si es par $\Rightarrow$ paridad 0 si impar $\Rightarrow$ paridad 1
  - Para corregir errores necesitamos más bits
- Idea de ECC en array de discos:

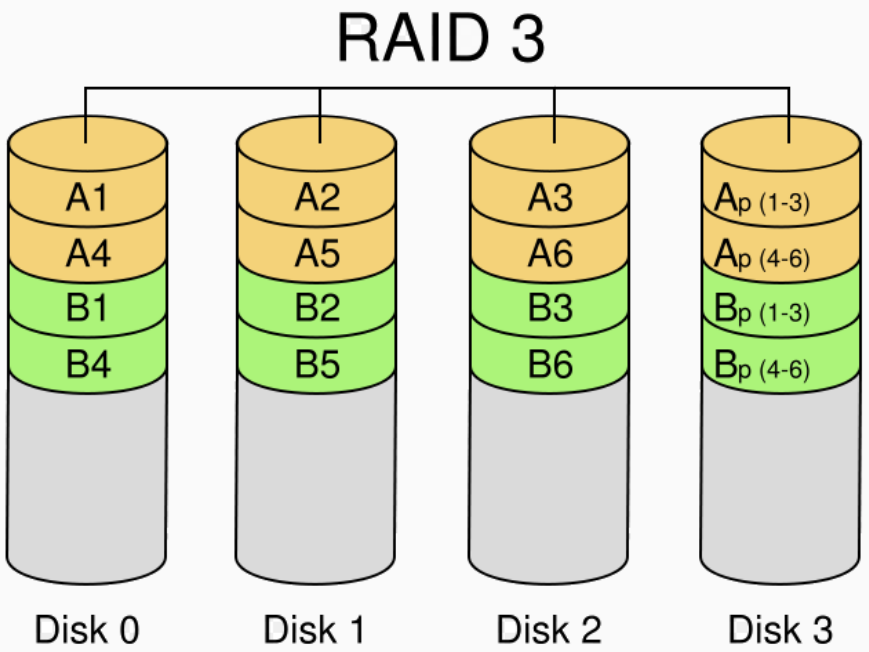


RAID 2

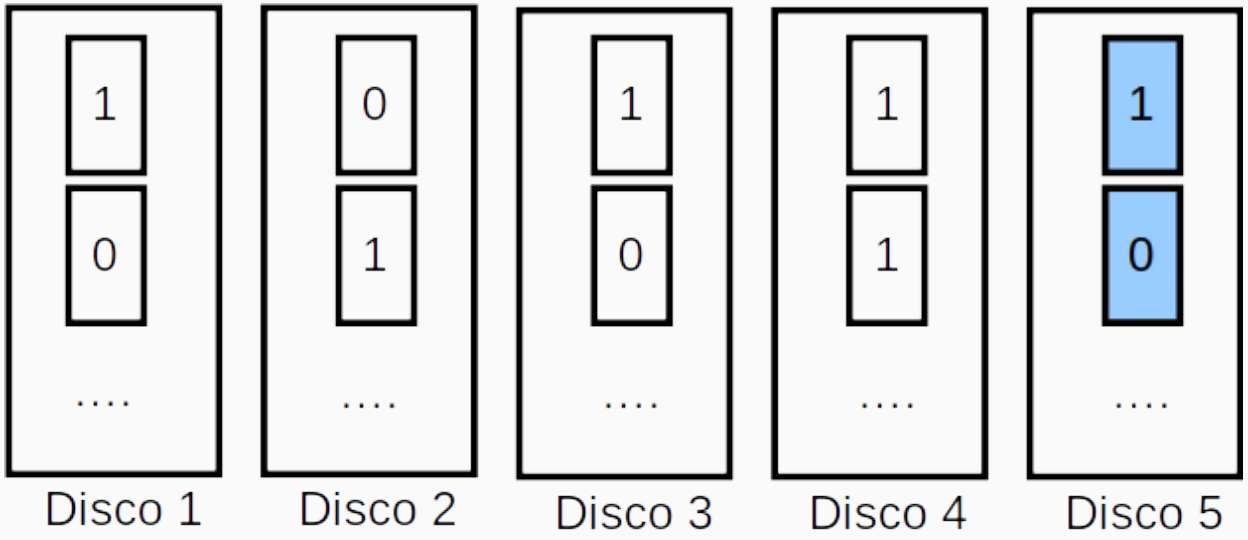
- División a nivel de bits
- Usa código de Hamming para corrección de errores
- Los discos son sincronizados por la controladora para funcionar a la vez.
- No es usado por ser inaplicable en la práctica

RAID 3

- División a nivel de bits
- Usa un disco de paridad dedicado. Usa XOR en lugar de Hamming
- Una operación de lectura o escritura necesita que todos los discos funcionen a la vez
- Se necesita un mínimo de 3 discos

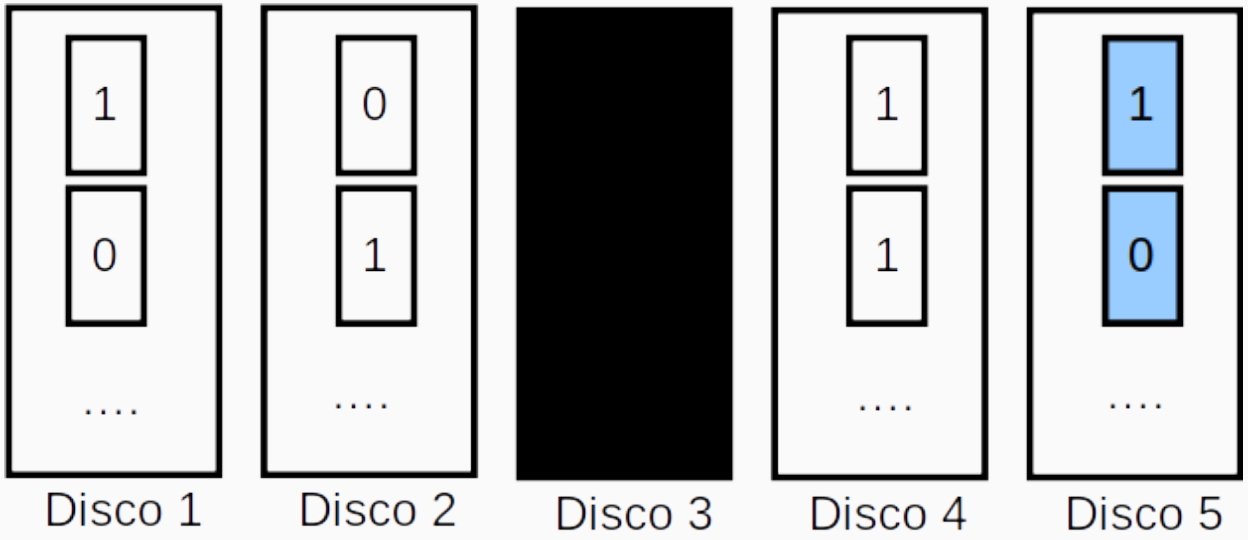


RAID 3



Paridad par

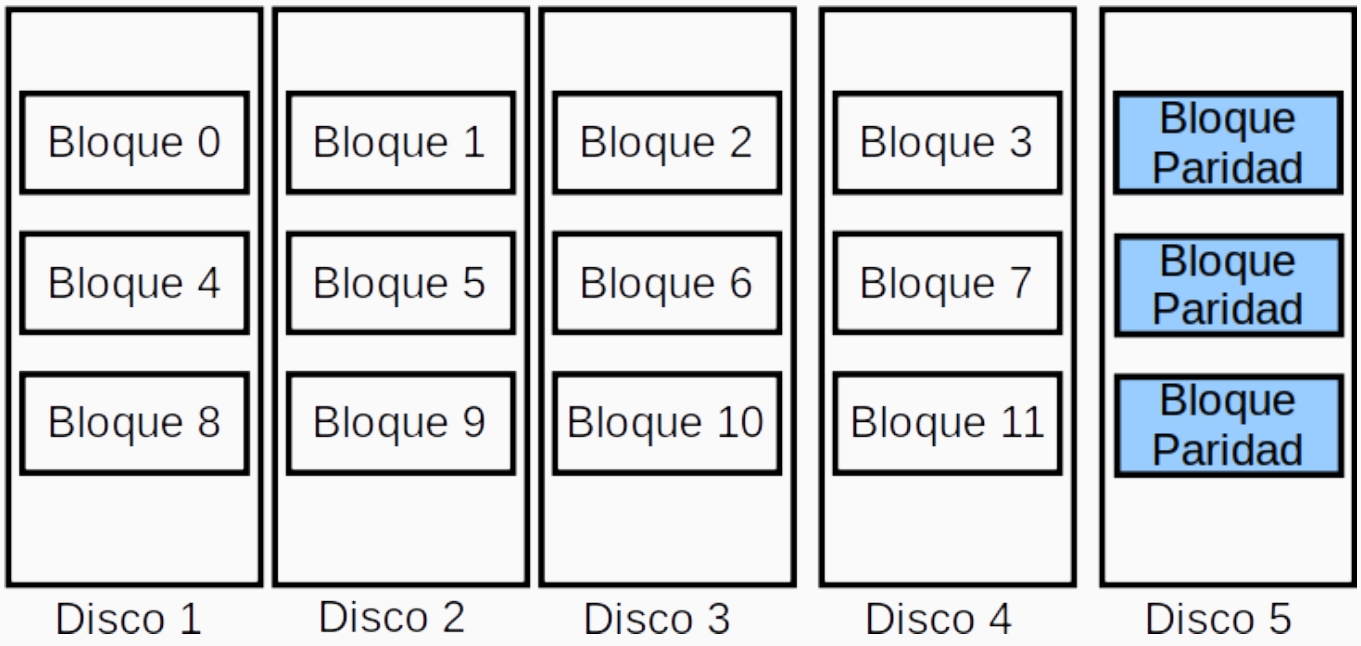
RAID 3



Con los discos 1,2,4 y 5 podemos recalcular el 3

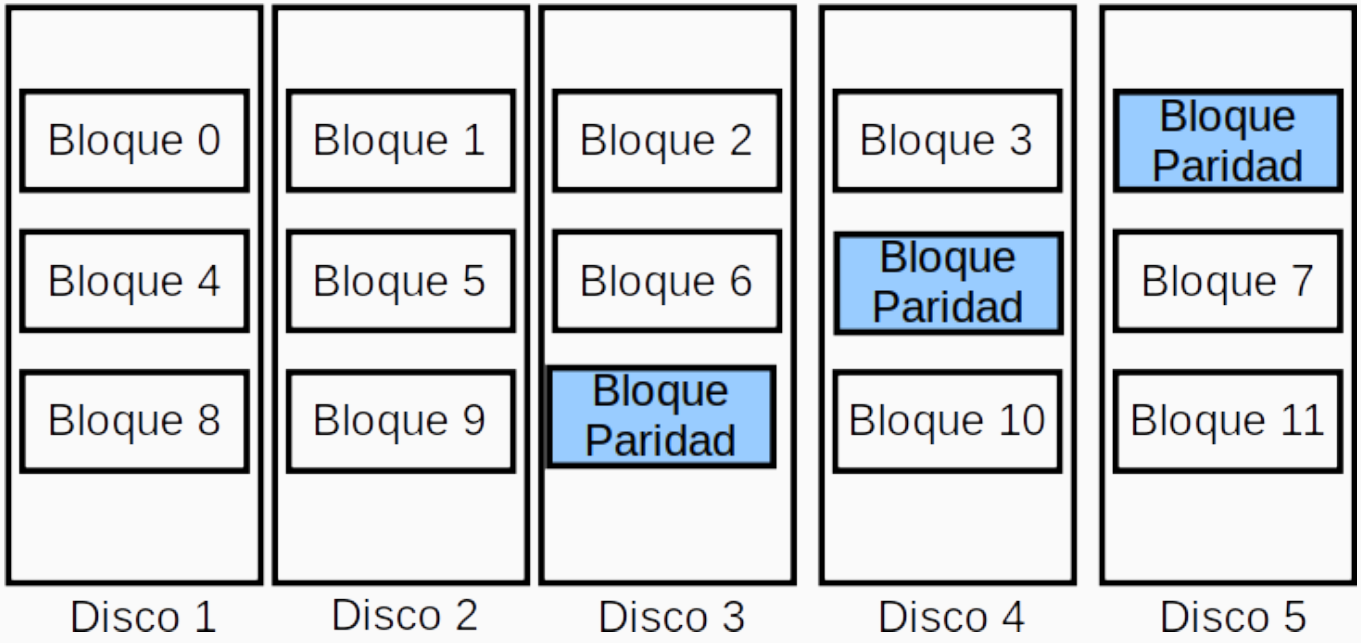
RAID 4

- Similar a RAID 3. Usa división a nivel de bloques
- Usa un disco de paridad
- Permite lecturas en paralelo pero no puede hacer pequeñas escrituras en paralelo.
- Una pequeña escritura implica 2 lecturas y 2 escrituras



RAID 5

- Similar a RAID 4 pero los datos de paridad son distribuidos entre todos los discos miembros.
- RAID 5 es muy popular debido a su bajo costo de redundancia.
- 4 unidades de 1TiB pueden construir un arreglo de 3 TiB en RAID 5.



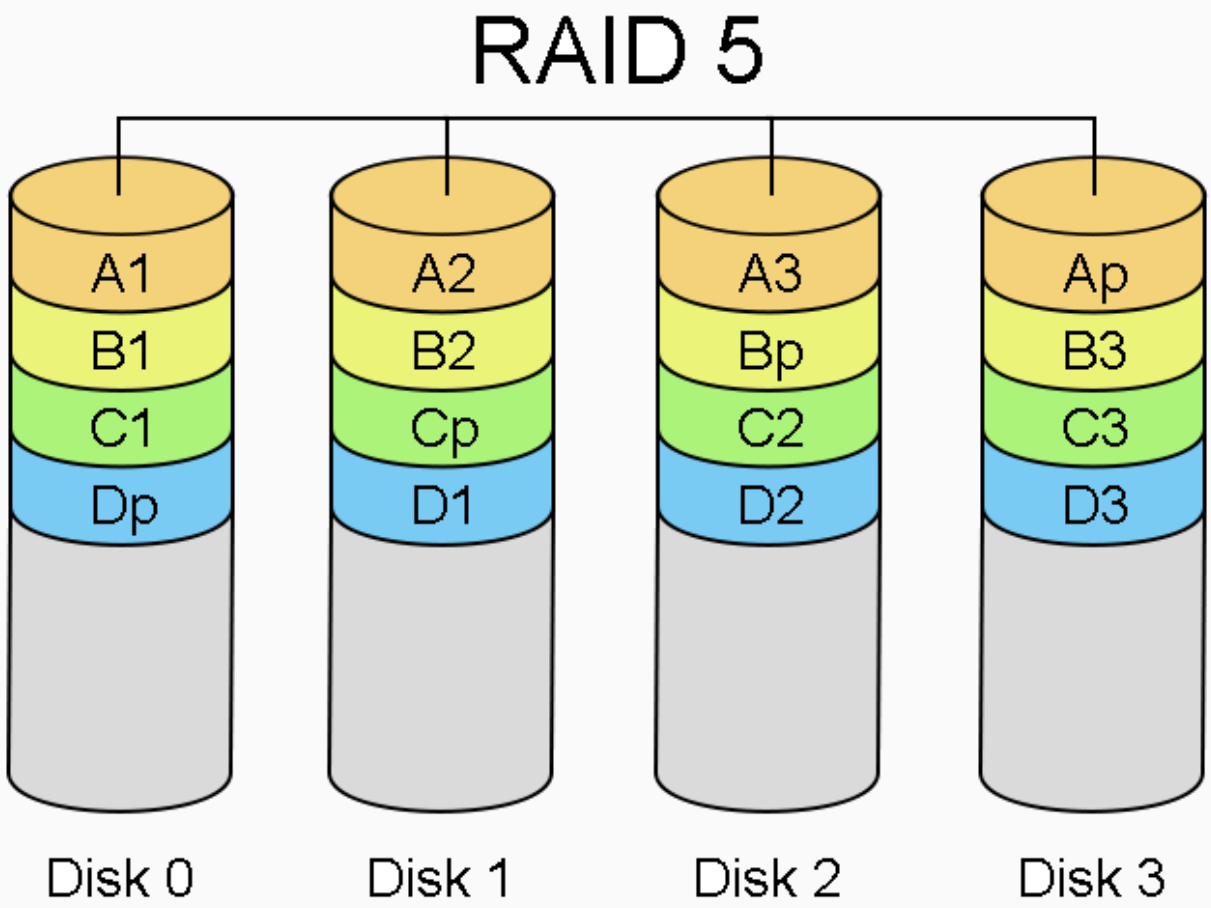
RAID 5

- En el ejemplo, una solicitud de lectura para el bloque 0 sería servida por el disco 0.
- Una solicitud de lectura simultánea para el bloque 4 tendría que esperar, pero una solicitud de lectura del bloque 1 podría ser resuelta al mismo tiempo por el disco 1 logrando mejor tiempo de respuesta.
- Si falla un disco puede seguir funcionando

RAID 5 - Manejo de paridad (parity handling)

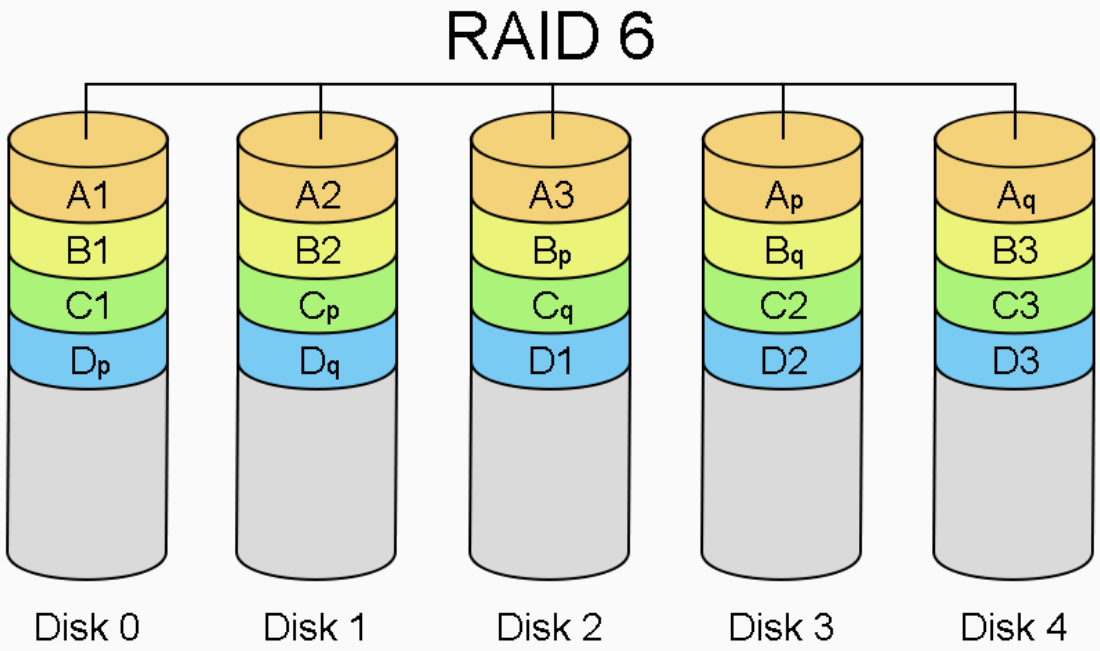
- Una serie de bloques simultáneos (uno en cada uno de los discos de una matriz) se le llama colectivamente una raya. Si otro bloque, o alguna parte del mismo, está escrito en esa misma franja, el bloque de paridad, o alguna parte del mismo, se vuelve a calcular y reescrito.
- Para las pequeñas escrituras requiere:
  - Leer el bloque de datos antiguos.
  - Leer el bloque de paridad antiguo.
  - Comparar el bloque de datos antiguo con la solicitud de escritura. Por cada bit que se ha invertido (cambia de 0 a 1, o de 1 a 0) en el bloque de datos, dar la vuelta el bit correspondiente en el bloque de paridad.
  - Escribe el bloque de datos nuevos.
  - Escribe el nuevo bloque de paridad.

RAID 5


RAID 5
A1
B1
C1
Dp
A2
B2
Cp
D1
A3
Bp
C2
D2
Ap
B3
C3
D3
Disk 0
Disk 1
Disk 2
Disk 3

RAID 6

- Agrega un segundo esquema de paridad al RAID 5
- Ineficiente para pequeño número de discos
- Protección ante fallos de 2 discos
- Puede ser más adecuado que un RAID 5 + 1 disco de reserva



Escritura de un solo sector en RAID 6

4 bloques de datos y dos de paridad. Se quiere escribir el sector **Datos 0**

Datos 0	Datos 1	Datos 2	Datos 3	Paridad 1	Paridad 2

Leer bloques 1 a 3 para calcular paridad

Datos 0	Datos 1	Datos 2	Datos 3	Paridad 1	Paridad 2

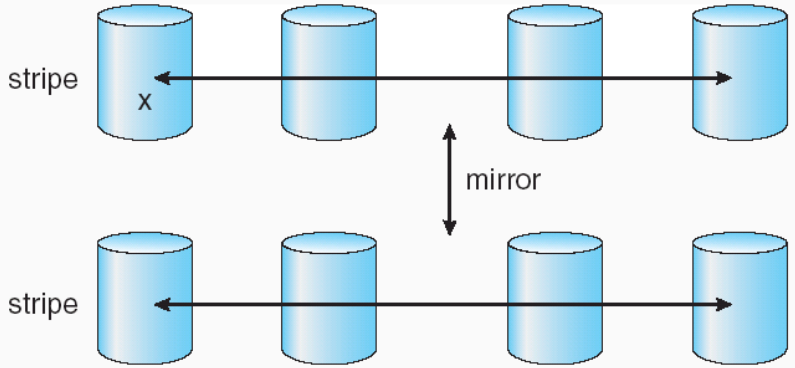
Escribir los datos y la paridad

Datos 0	Datos 1	Datos 2	Datos 3	Paridad 1	Paridad 2

Se usan todos los discos en dos operaciones secuenciales

RAID 0+1 y 1+0

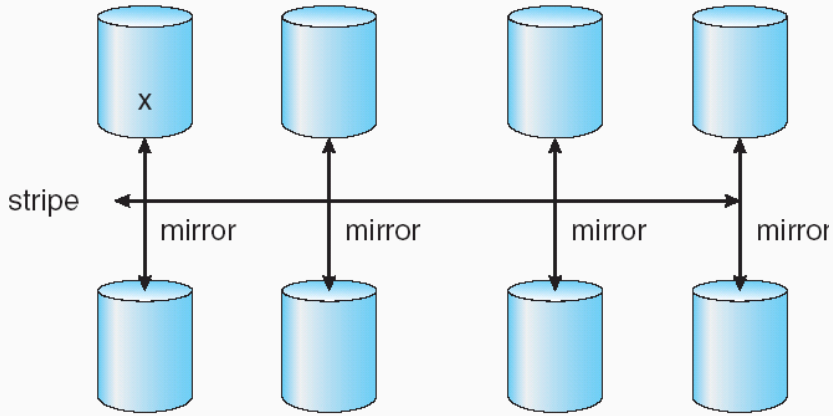
RAID 0+1



a) RAID 0 + 1 with a single disk failure.

Primero se divide, luego espejado. Es un espejo de divisiones

RAID 1+0



b) RAID 1 + 0 with a single disk failure.

Primero espejado, luego división Es una división de espejos

Escritura de un solo sector en RAID 10

3 pares de espejos. Se quiere escribir el sector **Datos 0**

Datos 0	Espejo 0	Datos 1	Espejo 1	Datos 2	Espejo 2

No hay cálculo de paridad, se escriben los datos directo

Datos 0	Espejo 0	Datos 1	Espejo 1	Datos 2	Espejo 2

Se usan dos discos en una sola operación