Planificación de Disco

Estudio del disco, de las políticas de planificación, la caché de disco y de las estructuras RAID

Tabla de Contenidos

Autor: Lina García Cabrera

1 Introducción

2º Grado en Ingeniería en Informática
teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco
En este módulo se estudia a fondo el dispositivo de E/S más importante, **el disco**:

- Los motivos por los que es necesario **disminuir el tiempo de acceso a disco**. Para ello, se presentan los parámetros que intervienen en una operación de disco (tiempo de búsqueda, tiempo de latencia y tiempo de transferencia), se comparan para determinar cuáles gravan o pesan más en el tiempo final de acceso a disco.
- Algoritmos de planificación de disco para reducir el tiempo de búsqueda, evaluación cualitativa de su comportamiento y logros.
- Almacenamiento redundante en discos independientes.

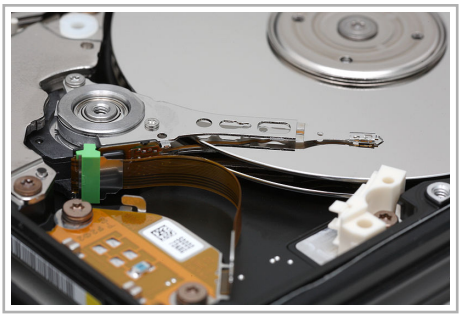


Figura 1. Interior de un disco duro; se aprecian dos platos con sus respectivos cabezales.

Un **disco duro** es un dispositivo de almacenamiento de datos no volátil que emplea un sistema de grabación magnética para almacenar datos digitales. Se compone de uno o más platos o discos rígidos, unidos por un mismo eje que gira a gran velocidad dentro de una caja metálica sellada. Sobre cada plato, y en cada una de sus caras, se sitúa un cabezal de lectura/escritura que flota sobre una delgada lámina de aire generada por la rotación de los discos.

OBJETIVOS

- Saber cómo influye la estructura física del disco en el rendimiento de la E/S.
- Conocer los distintos algoritmos de planificación: análisis y valoración de su comportamiento.
- Comprender en qué consiste la caché de disco y el almacenamiento redundante en discos independientes.

2 Motivación

2º Grado en Ingeniería en Informática
teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco
El crecimiento en velocidad de los procesadores y de la memoria principal ha dejado muy atrás al de los accesos a disco. La velocidad del procesador y de la memoria se ha incrementado en dos órdenes de magnitud con respecto al disco. Los **discos son, por lo menos, cuatro veces más lentos que la memoria principal**. Este avance se espera que continúe en el futuro inmediato. De este modo, el rendimiento de los subsistemas de almacenamiento de disco es de una importancia vital.

Para conectar la unidad del disco a la computadora se utilizan un conjunto de cables que se llama **bus de E/S**. Hay varios tipos de buses: **EIDE** (Enhanced Integrated Drive Electronics), **ATA** (Advanced Technology Attachment), **SATA** (serial SATA), **USB** (Universal Serial Bus), **FC** (Fiber Channel) y **SCSI** (Small Computer-Systems Interface.
Las transferencias de datos en un bus las realizan **los controladores** (procesadores especiales), entre **el controlador *host* y el controlador del disco**. El computador coloca una orden en el controlador *host* (puertos de E/S). El controlador *host* envía esta orden al controlador del disco y éste hace funcionar el *hardware* del disco para ejecutar la orden.
Los controladores utilizan una caché de disco, una zona en memoria, para hacer más rápida la transferencias.

3 Parámetros del Disco

2º Grado en Ingeniería en Informática
teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco

Los detalles reales de las **operaciones de E/S** con los discos dependen del ***computador, el sistema operativo, y la naturaleza del canal de E/S y el hardware controlador de disco***.

La Figura 2 es una representación esquemática de la vista lateral de un disco de cabeza móvil. Los datos se graban sobre una serie de discos magnéticos o platos. Estos discos están conectados por un eje común que gira a una velocidad muy alta (entre 60 y 200 vueltas por segundo, esto es entre 3600 rpm y 12000 rpm).

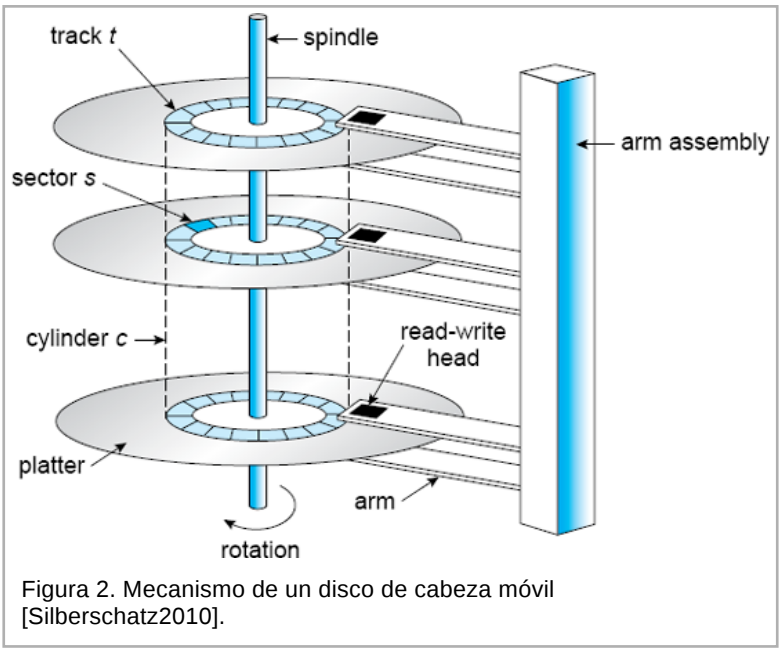


Figura 2. Mecanismo de un disco de cabeza móvil [Silberschatz2010].

Cuando la unidad de disco está operando, el disco gira a una velocidad constante. Para leer o escribir, la cabeza debe **posicionarse en la pista deseada**, **al comienzo del sector pertinente**.

Si el sistema es de cabezas móviles, hay que mover la cabeza para elegir la pista. Si el sistema es de cabezas fijas, habrá que seleccionar electrónicamente una de ellas. En un sistema de cabezas móviles, **el tiempo que se tarda en ubicar la cabeza en la pista** se llama ***tiempo de búsqueda***.
Una vez que se ha seleccionado la pista, el controlador de disco esperará hasta que el sector apropiado se posicione debajo de la cabeza en su rotación. **El tiempo que tarda el comienzo del sector en llegar hasta la cabeza** se conoce como ***tiempo de latencia***.
La **suma del tiempo de búsqueda y de latencia** es el ***tiempo de acceso***, es decir, el tiempo que se tarda en llegar a la posición de lectura o escritura.
Una vez que la cabeza está ubicada, se puede llevar a cabo la **operación de Lectura o Escritura a medida que el sector se mueve bajo la cabeza**; ésta es la parte de transferencia real de datos de la operación (***tiempo de transferencia***).

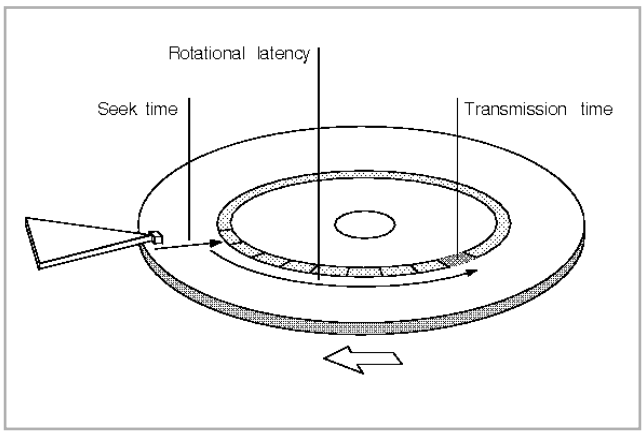


Figura 3. Componentes de un acceso a disco.
Además del tiempo de acceso y del tiempo de transferencia, en una operación de E/S intervienen algunos retardos:

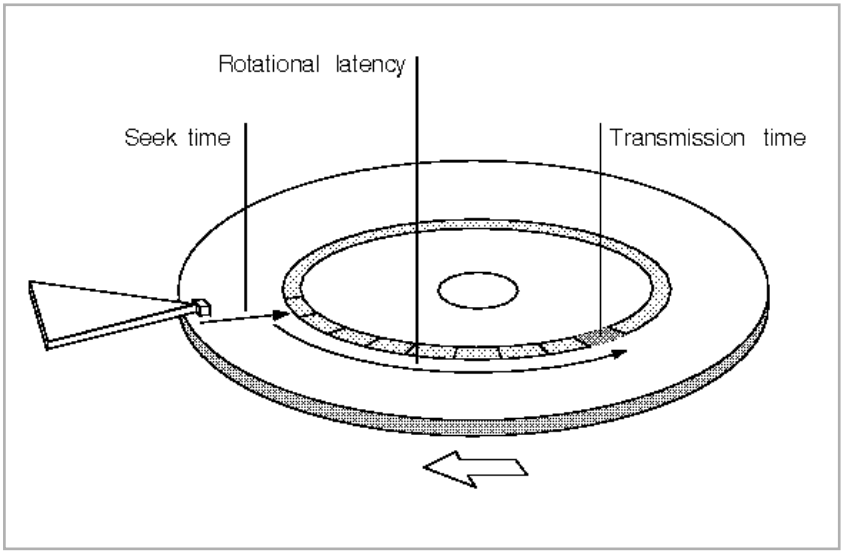
- Cuando un proceso emite una petición de E/S, primero debe **esperar en una cola a que el dispositivo** esté disponible. En ese momento, el dispositivo queda asignado al proceso.
- Si el dispositivo comparte un único canal de E/S o un conjunto de canales con otras unidades de disco, puede producirse una **espera adicional hasta que el canal esté disponible**. En ese punto se realizará la búsqueda con que comienza

el acceso al disco.

Para obtener acceso a un registro de datos en particular del disco son necesarias varias operaciones. Primero, el brazo móvil debe desplazarse hacia el cilindro (o pista) apropiado (**tiempo de búsqueda**). Después, la porción de disco en donde se encuentran los datos debe girar hasta quedar inmediatamente abajo (o arriba) de la cabeza de lectura-escritura (**tiempo de latencia**). Después, el registro debe girar para pasar por la cabeza para ser leído o escrito (**tiempo de transferencia**).
Como cada una de estas operaciones implica movimiento mecánico, el tiempo total de acceso a un registro es muy superior al tiempo de procesamiento.

3.1 Tiempo de Búsqueda

2º Grado en Ingeniería en Informática
teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco



El **tiempo de búsqueda** es el tiempo necesario para mover el brazo del disco hasta la pista (o cilindro) solicitada. Esta cantidad resulta difícil de concretar. El tiempo de búsqueda consta de dos componentes clave: el **tiempo de arranque inicial** y el **tiempo que se tarda en recorrer los cilindros o pistas**, una vez que el brazo haya cogido velocidad. Por desgracia, el tiempo de recorrido no es una función lineal del número de pistas.

Se puede aproximar el tiempo de búsqueda con la fórmula lineal:

$T_b = mxn + s$
donde:
T_b = tiempo de búsqueda estimado
n = número de pistas recorridas
m = constante que depende de la unidad de disco
s = tiempo de arranque

Por ejemplo, un disco duro económico en un computador personal podría tener, aproximadamente, m=0'3 ms y s=20 ms, mientras que uno más grande y más caro podría tener m= 0'1 ms y s = 3 ms.

3.2 Tiempo de Latencia

2º Grado en Ingeniería en Informática

teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco
El **tiempo de latencia** es el tiempo que tarda el comienzo del sector en llegar hasta la cabeza una vez situada en la pista solicitada.

> Los discos, excepto las unidades CD/DVD, giran normalmente a 7200 rpm, es decir, una revolución cada 8'3 ms. Por tanto, el retardo medio de giro será de 4'16 ms. Los discos CD/DVD giran mucho más lentamente, generalmente entre 300 y 600 rpm. Por tanto, el retardo medio estará entre 100 y 200 ms.

3.3 Tiempo de Transferencia

2º Grado en Ingeniería en Informática
teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco
El **tiempo de transferencia** es el tiempo que el registro o el bloque (un múltiplo binario de un sector) debe girar para pasar por la cabeza para ser leído o escrito.
El tiempo de transferencia de un disco depende de la velocidad de rotación de la forma siguiente:
$T_t = b/(rN)$
donde
T_t = tiempo de transferencia
b = número de bytes a transferir
N = número de bytes por pista
r = velocidad de rotación en revoluciones por segundo
Por tanto, el tiempo medio de acceso total puede expresarse como
$T_a t = T_b + T_l + T_t = (mxn + s) + (1/2r) + (b/rN)$

donde T_b es el tiempo de búsqueda, T_l el tiempo de latencia y T_t el tiempo de transferencia.

3.4 Comparativa de Tiempos

2º Grado en Ingeniería en Informática
teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco

> Considérese un disco típico con un tiempo medio de búsqueda conocido de 3 ms, velocidad de transferencia de 7200 rpm ó 1,875 MB/seg y sectores de 512 bytes, habiendo 32 sectores por pista.
>
> El **tiempo de latencia** será:
> Calculamos primero cuántas vueltas se dan en un segundo 7200/60 = 120 rps.
> Ahora calculamos el tiempo que se tarda en dar una vuelta 1/120 = 8,3 ms, luego el tiempo de latencia es 4,16 ms.
>
> Supóngase que se desea ***leer un fichero que consta de 256 sectores*** para formar un total de 128 KB. Así podría estimarse el tiempo total que dura la transferencia.
>
> En primer lugar, supóngase que el fichero se almacena en disco de la forma más compacta posible. Es decir, el **fichero ocupará todos los sectores de ocho pistas adyacentes** (8 pistas x 32 sectores/pista = 256 sectores). Esta disposición se conoce como organización secuencial. En tal caso, el tiempo que dura la lectura de la primera pista el siguiente:
>
> | | |
> |---|---|
> | Búsqueda media | 3'0 ms |

Latencia media	4’16 ms
Lectura de 32 sectores (1 vuelta)	8’3 ms
	15’46 ms

Supóngase que las pistas restantes pueden leerse ahora sin tiempo de búsqueda alguno. Es decir, la operación de E/S puede llevar el ritmo del flujo de datos que lleva el disco. En tal caso hace falta considerar, como mucho, el tiempo de latencia. Por tanto, cada pista consecutiva se puede leer en 4’16 + 8’3 = 12’46 ms. Para leer el fichero por completo:
Tiempo total = 15’46 + 7 x 12’46 = 102’68 ms = 0’1 seg.

Se calcula ahora el tiempo necesario para leer los mismos datos utilizando **acceso aleatorio** en vez de acceso secuencial; esto es, el acceso a los sectores se distribuye aleatoriamente por el disco. Para cada sector, se tiene:

Búsqueda media	3’0 ms
Latencia media	4’16 ms
Lectura de 1 sector	0’26 ms
	7’42 ms

Tiempo total = 256 x 7’42 = 1899’52 ms = 1’9 seg.

Está claro que **el orden en que se leen los sectores del disco tiene un efecto inmenso en el rendimiento de la E/S**. En el caso de los accesos a ficheros en los que se leen varios sectores, se puede ejercer algún control sobre la manera en que se distribuyen los sectores de datos (gestión de sistemas de ficheros). Sin embargo, incluso en el caso de acceso a un fichero en un entorno de multiprogramación, existirán varias solicitudes de E/S que compitan por el mismo disco. Por tanto, merece la pena investigar alguna manera más de mejorar el rendimiento de E/S a disco.

4 Políticas de Planificación

2º Grado en Ingeniería en Informática
teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco
Antes de describir las distintas políticas de planificación que se pueden adoptar, se enumeran los criterios que nos van a permitir valorarlas:

- la productividad
- el tiempo promedio de respuesta
- la varianza de los tiempos de respuesta (predecibilidad)

Está claro que una política de planificación debe tratar de lograr una **productividad máxima** (el mayor número posible de peticiones atendidas por unidad de tiempo). Para ello, se deberá reducir al máximo el tiempo desperdiciado en las búsquedas muy largas.

Pero una política de planificación también debe tratar de **reducir el tiempo promedio de respuesta** (es decir, el tiempo promedio de espera más el tiempo promedio de servicio).

Los criterios señalados tratan de **mejorar el rendimiento global**, tal vez a expensas de las peticiones individuales. La planificación mejora a menudo el rendimiento global pero

reduce el nivel de atención a ciertas peticiones. Una medida importante de este fenómeno es la varianza de los tiempos de respuesta. La **varianza es una media matemática de cuánto se desvían elementos individuales del promedio de los elementos**. Como tal, utilizaremos la varianza para indicar la predecibilidad: **a menor varianza mayor predecibilidad**. Deseamos una política que reduzca al mínimo la varianza.

4.1 Planificación FCFS (First Come First Served)

2º Grado en Ingeniería en Informática
teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco
Por supuesto, la forma más sencilla para planificar un disco es la planificación **primera solicitud en llegar primera en ser servida** (FCFS, ***F**irst **C**ome **F**irst **S**erved*). Este algoritmo es fácil de programar e intrínsecamente justo, pero es probable que **no ofrezca el mejor tiempo de servicio** (en promedio).

> Considere, por ejemplo, una cola de peticiones a disco que afecta a las pistas 98, 183, 37, 122, 14, 124, 65 y 67
> Si, en principio, la cabeza de lectura-escritura está en la pista 53, se moverá primero de la 53 a la 98, luego a la 183, 37, 122, 14, 124, 65 y por último a la 67, lo que da un movimiento total de la cabeza de 640 pistas (Figura 4).

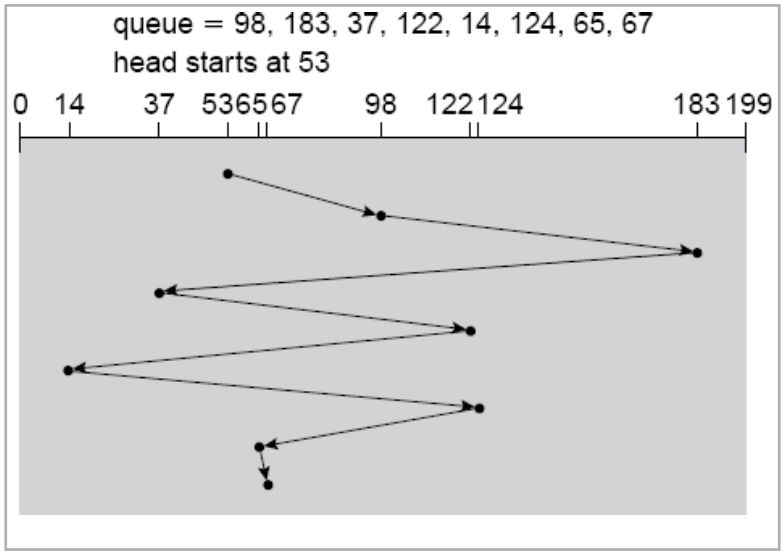


Figura 4. Planificación de disco First-Come-First-Served [Silberschatz2010].

FCFS es **justa en el sentido de que al llegar una solicitud**, su lugar en la planificación es fijo (se respeta el orden de llegada). Una petición no puede ser desplazada por la llegada de otra con mayor prioridad.
Cuando las peticiones están distribuidas uniformemente en la superficie del disco, la planificación FCFS da como resultado un patrón de búsqueda aleatorio. Hace caso omiso de la relaciones de posición entre las solicitudes pendientes. No hace ningún intento por optimizar el patrón de búsqueda.
FCFS es **aceptable cuando la carga de disco es ligera**. Pero conforme crece la carga, FCFS tiende a saturar el dispositivo, y aumenta los tiempos de respuesta.
FCFS ofrece una varianza pequeña, pero esto no es un gran consuelo para la petición que espera al final de la cola mientras el brazo móvil se desplaza de un lado a otro.

4.2 Planificación SSTF (Shortest Seek Time First)

2º Grado en Ingeniería en Informática
teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco
Parece razonable servir juntas todas las solicitudes próximas a la posición actual del disco, antes de desplazar la cabeza a un punto lejano para servir otra solicitud. Esta suposición es la base del algoritmo **primero la de menor tiempo de posicionamiento** (SSTF, *Shortest Seek Time First*) para la planificación de disco.
El **algoritmo SSTF** selecciona la solicitud con menor tiempo de posicionamiento a partir de la posición actual de la cabeza. Como el tiempo de búsqueda normalmente es proporcional a la diferencia entre las solicitudes medida en pistas, implantamos esta estrategia moviendo la cabeza a la pista más próxima de la cola de solicitudes.

> En el ejemplo de cola de solicitudes: 98, 183, 37, 122, 14, 124, 65 y 67, la más próxima a la posición inicial (53) es la pista 65. Una vez ubicados en la pista 65, la siguiente pista más cercana es la 67. En la figura aparece el orden en el que se resolverían las peticiones. Este método de planificación da como resultado un movimiento total de sólo 236 pistas, poco más de la tercera parte de la distancia recorrida con la planificación FCFS. Este algoritmo mejora considerablemente el promedio del servicio del disco.

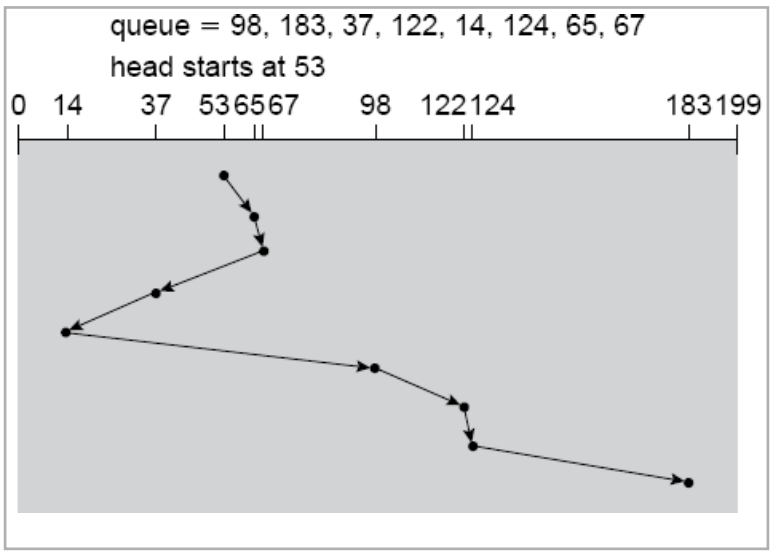


Figura 5. Planificación de disco Shortest-Seek-Time-First [Silberschatz2010].

SSTF tiende a favorecer mucho ciertas peticiones. Los patrones de búsqueda SSTF tienden a estar muy localizados y, en consecuencia, **las pistas más exteriores e interiores pueden recibir una atención deficiente** en comparación con la que reciben las pistas de la parte media.
SSTF ofrece **mejores tasas de productividad que FCFS pero no es el algoritmo óptimo**.

> Por ejemplo, si movemos la cabeza de la 53 a la 37, aunque ésta no sea la pista más próxima, y luego a la 14, antes de regresar para servir a la 65, 67, 98, 122 y 124 y 183 podemos reducir el movimiento total de la cabeza a 208 pistas.

Los **tiempos de respuesta tienden a ser más bajos** cuando la carga es moderada. Una desventaja importante es que **aumenta la varianza de los tiempos de respuesta** debido a la discriminación contra la pistas exteriores e interiores. En un caso extremo, su gestión podría sufrir un aplazamiento indefinido. Si se consideran las importantes mejoras en la productividad y en los tiempos promedio de respuesta, el aumento de la varianza puede parecer tolerable.
SSTF resulta **útil en sistemas de procesamiento por lotes**, donde la productividad es lo más importante. Pero la elevada varianza de los tiempos de respuesta (su impredecibilidad) lo hace **inaceptable en sistemas interactivos**.

4.3 Planificación SCAN

2º Grado en Ingeniería en Informática
teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco

Denning (1967) desarrolló la estrategia de planificación SCAN para evitar la discriminación y la alta variación en los tiempos de respuesta de la Planificación SSTF (Shortest Seek Time First).

SCAN opera como SSTF, excepto que **SCAN elige la solicitud que implica la menor distancia de búsqueda en una dirección predefinida**. Si la dirección predefinida es en un momento dado hacia afuera, la estrategia SCAN elige la distancia de búsqueda más corta en esa dirección. SCAN no cambia de dirección hasta que llega a la pista más exterior o hasta que ya no hay solicitudes pendientes en la dirección predefinida.

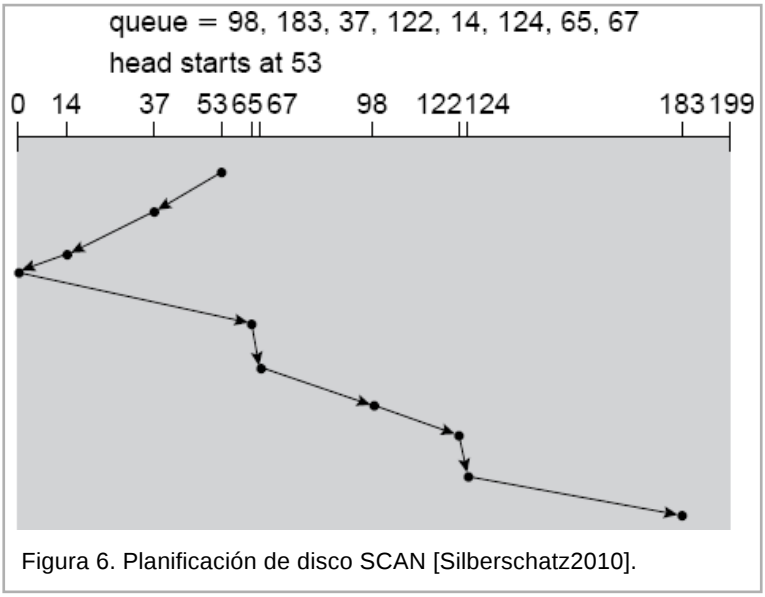


Figura 6. Planificación de disco SCAN [Silberschatz2010].

> Para el ejemplo que nos ocupa necesitamos saber la dirección de movimiento de la cabeza, así como su posición más reciente. Aquí hemos supuesto que la cabeza se desplaza hacia las pistas de menor número, y se parte de la 53.

SCAN tiene un comportamiento muy parecido al de SSTF en términos de **aumento de productividad** y **reducción de tiempos promedio de espera**, pero **elimina gran parte de la discriminación** inherente a los esquemas SSTF y ofrece una **varianza mucho menor**. A causa del movimiento oscilante de las cabezas de lectura-escritura en SCAN, las pistas más exteriores se visitan con menos frecuencia que las de la parte media, pero ello no es tan grave como la discriminación de SSTF.

4.4 Planificación N-SCAN

2º Grado en Ingeniería en Informática
teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco

SCAN de N pasos es una modificación de la estrategia SCAN básica. En esta estrategia, el brazo del disco se mueve en una y otra dirección como en SCAN, excepto que sólo atiende las solicitudes que ya estaban esperando cuando se inició un barrido específico. Las solicitudes que llegan durante una barrido se agrupan para darles una atención óptima durante el barrido de regreso. En cada barrido se atienden las primeras N peticiones.

SCAN de N pasos ofrece un buen desempeño en cuanto a productividad y tiempo promedio de respuesta. Su característica más importante es una menor variación de los tiempos de respuesta que en la planificación SSTF o en la SCAN convencional.

SCAN de N pasos elimina la posibilidad de que ocurra un aplazamiento indefinido si llega un gran número de peticiones para la pista actual. Éste guarda dichas peticiones para atenderlas en el siguiente barrido.

4.5 Planificación C-SCAN

2º Grado en Ingeniería en Informática
teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco

C-SCAN (SCAN circular) es otra modificación interesante de la estrategia SCAN básica. C-SCAN **elimina la discriminación** de SSTF contra las pistas más interiores y exteriores.

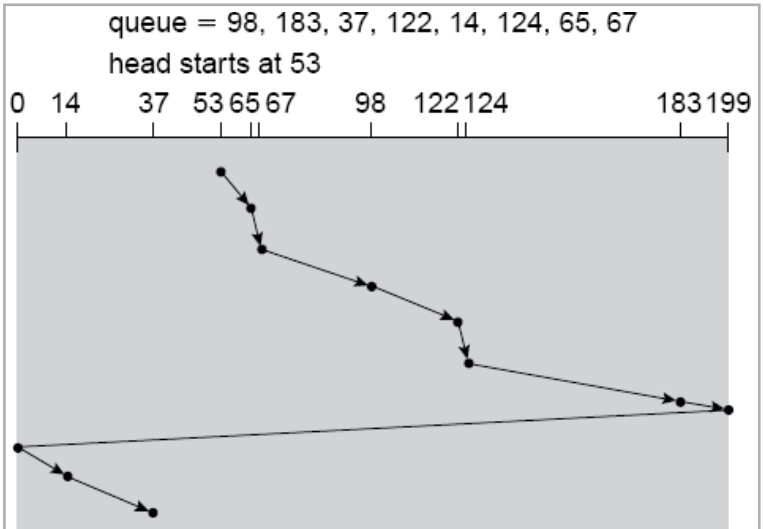


Figura 7. Planificación de disco C-SCAN [Silberschatz2010].

En la estrategia C-SCAN el brazo se mueve del cilindro exterior hacia el interior, atendiendo solicitudes según un criterio de búsqueda más corta. Cuando el brazo ha completado su barrido hacia adentro, salta (sin atender peticiones) a la solicitud más cercana al cilindro más exterior y luego reinicia su barrido hacia adentro procesando solicitudes.

C-SCAN se puede llevar a la práctica de manera tal que las solicitudes que lleguen durante un barrido sean atendidas en el siguiente barrido. Presenta una **varianza muy pequeña** con respecto a los tiempos de respuesta.

Algunos resultados de simulaciones indican que la mejor política de planificación de disco podría operar en dos etapas. Cuando la carga es ligera, la política SCAN es la mejor. Cuando la carga es mediana o pesada, C-SCAN produce los mejores resultados.

4.6 Planificación de Disco en Linux

2º Grado en Ingeniería en Informática
teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco

En Linux se puede configurar el tipo de planificación de disco que se aplicará. El archivo **`/sys/block/sda/queue/scheduler`** contiene los planificadores de disco disponibles y el activo en un instante dado (el que está entre corchetes).

```
1 % cat /sys/block/sda/queue/scheduler
2 noop deadline anticipatory [cfq]
3
4 % echo "noop" > /sys/block/sda/queue/scheduler
```

La primera línea del ejemplo es para consultar esa información y la segunda es el resultado en un Ubuntu sobre un hardware normal.

Los planificadores normales en Linux (desde la versión 2.6 del kernel) son ***noop*** (FIFO), ***deadline***, ***anticipatory*** y ***cfq***. Los tres últimos son variantes del C-SCAN, pero con algún añadido. Por ejemplo, ***deadline***, establece un quanto de tiempo y atiende a las peticiones que llevan esperando más que ese cuanto para garantizar un tiempo de espera máximo.

El planificador se puede cambiar en tiempo de ejecución y el efecto permanece hasta el siguiente arranque. La forma es: **`echo "planificador" > /sys/block/sda/queue/scheduler`**

La última línea de la transparencia permite activar el planificador FIFO (útil por ejemplo, si se tienen discos SSD, *Solid-State Drive* Discos de estado sólido), ya que no tienen partes móviles y, por tanto, no necesitan ahorrar tiempo de búsqueda.

5 Caché de Disco

2º Grado en Ingeniería en Informática
teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco
La **memoria caché** es una memoria más pequeña y rápida que la memoria principal, que se sitúa entre ésta y el procesador. Este tipo de memoria caché reduce el tiempo medio de acceso a memoria aprovechándose del principio de localidad.

El mismo principio puede aplicarse a la **caché de disco**, un buffer para sectores de disco situado en la memoria principal. La caché contiene una copia de algunos sectores del disco.
Cuando se hace una petición de E/S para un bloque, se comprueba si ese bloque está en la caché del disco. Si es así, la petición se cumple con la caché. Si no, se lee el sector pedido de disco y se coloca en la caché. Debido a la localidad de referencias o localidad, cuando se traiga una bloque de datos a la caché para satisfacer una sola petición de E/S, será probable que se produzcan referencias futuras al mismo bloque.
Como las operaciones de escritura no siempre se realizan cuando los programas creen que se realizaron, es probable que en un momento dado el contenido de los ficheros de disco difiera de lo que los programas creen que contienen. Este problema es crucial sobre todo si se cae el sistema.
Para reducir al mínimo la posibilidad de que los discos estén en desacuerdo con lo que los programas creen que contienen, el sistema operativo UNIX ejecuta periódicamente la llamada al sistema ***sync*** para grabar todos los buffers de disco.

5.1 Consideraciones de Diseño

2º Grado en Ingeniería en Informática

teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco
Son de interés varias cuestiones de diseño. En primer lugar, cuando una petición de E/S se sirve con la caché, los **datos de la misma deben ser enviados al proceso** que los solicitó:

- El envío puede hacerse por una transferencia en memoria del bloque de datos, desde la caché del disco a la memoria asignada al proceso de usuario o
- usar la posibilidad de memoria compartida y pasar un puntero a la entrada apropiada de la caché del disco. Este último método ahorra el tiempo de transferencia interna a memoria y, además, permite el acceso compartido de otros procesos que puedan seguir el modelo de los lectores/escritores descrito en el tema de concurrencia.

Una segunda cuestión de diseño tiene que ver con la **estrategia de reemplazo**. Cuando se trae un nuevo bloque a la caché del disco, uno de los bloques existentes debe ser sustituido. Este problema es idéntico al presentado en el tema de memoria virtual, donde se empleaban algoritmos de reemplazo de páginas. Se han probado un buen número de algoritmos. El algoritmo más común es el LRU, en el que se reemplaza el bloque que lleva más tiempo en la caché sin ser referenciado.
Sin considerar la estrategia de reemplazo en particular, la sustitución puede llevarse a cabo bajo demanda o puede ser planificada previamente:

- En el primer caso, los sectores se sustituyen sólo cuando se necesita su entrada de la tabla.
- En el último caso, cada vez se liberan un conjunto de entradas. La razón de ser de este método está relacionada con la necesidad de volver a escribir los bloques. Si se trae un bloque a la caché y sólo es leído, no será necesario volver a escribirlo al disco cuando sea reemplazado. Sin embargo, si el bloque es actualizado, entonces sí será necesario volver a escribirlo antes de reemplazarlo. En este último caso, tiene sentido agrupar las escrituras y ordenarlas para minimizar el tiempo de búsqueda.

5.2 Consideraciones de Rendimiento

2º Grado en Ingeniería en Informática
teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco
Se aplican las mismas consideraciones sobre el rendimiento de la administración de memoria virtual. El rendimiento de la caché será bueno siempre que sea mínima la tasa de fallos. Esto dependerá de la localidad de las referencias al disco, del algoritmo de reemplazo y de otros factores de diseño. Sin embargo, la tasa de fallos es principalmente función del tamaño de la caché de disco.

6 Estructuras RAID

2º Grado en Ingeniería en Informática
teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco

RAID (del inglés ***R****edundant* ***A****rray of* ***I****ndependent* ***D****isks*), "***conjunto redundante de discos independientes***", anteriormente conocido como *Redundant Array of Inexpensive Disks*, "conjunto redundante de discos baratos") hace referencia a un sistema de almacenamiento que usa múltiples discos duros entre los que se distribuyen o replican los datos.

Dependiendo de su configuración (**nivel**), los beneficios de un RAID respecto a un único

disco son uno o varios de los siguientes:

- mayor integridad,
- mayor tolerancia a fallos,
- mayor rendimiento y
- mayor capacidad.

Para mejorar la **fiabilidad se opta por la redundancia**. La técnica más simple es ***duplicar en espejo***. Cada disco lógico está compuesto por dos discos físicos y toda la escritura se hace en ambos discos a la vez. Si uno falla, se pueden leer los datos de otro disco.

La técnica de duplicar en espejo también **mejora las prestaciones** porque se puede enviar la solicitud a cualquiera de los dos discos.

Con múltiples discos se mejora la velocidad de transferencia dividiendo datos entre los distintos discos. La **distribución en bandas de los datos** (*data striping*) **de nivel de bit** consiste en dividir los bits de cada byte entre los distintos discos.

> Por ejemplo, con una matriz de 8 discos, se escribe el bit i de cada byte en el disco i. En cada acceso se puede leer ocho veces más datos.

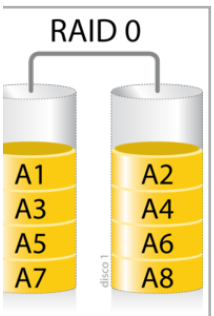
La **distribución en bandas de nivel de bloques**, los bloques de cada fichero se dividen entre n discos, el bloque i de un fichero va al disco (i mod n) + 1.

6.1 Niveles RAID

2º Grado en Ingeniería en Informática
teoría de Sistemas Operativos
Bloque Entrada Salida y Ficheros: Planificación de Disco

- **La duplicación en espejo proporciona una alta fiabilidad, pero resulta muy cara**.
- **La distribución en bandas proporciona una alta velocidad transferencia de datos, pero no mejora la fiabilidad**.

Se ha propuesto combinaciones denominados niveles en los que se establece un **compromiso entre el coste y las prestaciones**. Existen 6 niveles RAID estándar que se pueden combinar.



gura 8. RAID 0

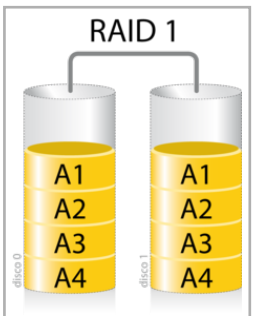


Figura 9. RAID 1

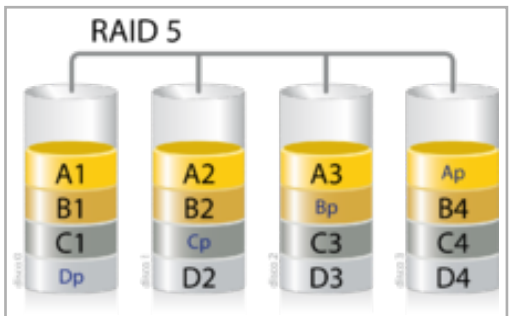


Figura 10. RAID 5

Los niveles RAID más usados son:

- **RAID 0**: ***Conjunto dividido***. Matrices de discos con distribución de bandas en el nivel de bloques, pero sin redundancia.

- **RAID 1**: ***Conjunto en espejo***. Crea una copia exacta (o espejo) de un conjunto de datos en dos o más discos.
- **RAID 5**: ***Conjunto dividido en bloques con paridad distribuida***. Necesita al menos 3 discos para su implementación. Consiste en distribuir tantos los bloques como la paridad. Los bloques se distribuyen entre n-1 discos y se utiliza el otro disco almacenar la paridad de los bloques de los otros discos. De este modo, se puede recuperar la información de cualquier bloque.

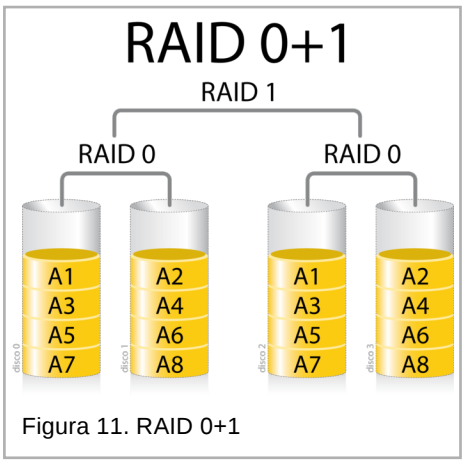


Figura 11. RAID 0+1

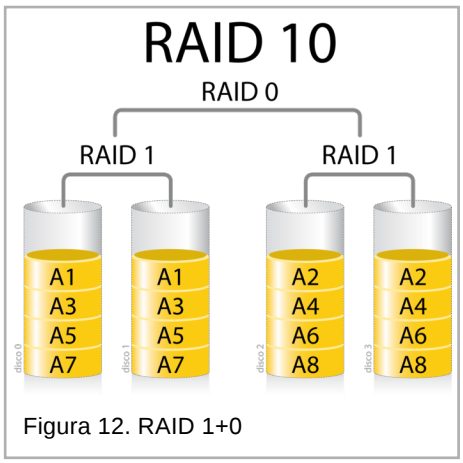


Figura 12. RAID 1+0

Normalmente se utilizan esquemas combinados. Los RAID anidados usan el nivel 0:

- **RAID 0+1**: ***Un espejo de divisiones***. Es un RAID usado para replicar y compartir datos entre varios discos. Primero se crean dos conjuntos RAID 0 (dividiendo los datos en discos) y luego, sobre los anteriores, se crea un conjunto RAID 1 (realizando un espejo de los anteriores). La ventaja de un RAID 0+1 es que cuando un disco duro falla, los datos perdidos pueden ser copiados del otro conjunto de nivel 0 para reconstruir el conjunto global.
- **RAID 1+0**: ***Una división de espejos***. Primero se hace un espejo y luego se distribuye los espejos. En cada división RAID 1 pueden fallar todos los discos salvo uno sin que se pierdan datos. Sin embargo, si los discos que han fallado no se reemplazan, el restante pasa a ser un punto único de fallo para todo el conjunto.
- **RAID 50**: ***Combina la división a nivel de bloques de un RAID 0 con la paridad distribuida de un RAID 5***, siendo pues un conjunto RAID 0 dividido de elementos RAID 5. Un disco de cada conjunto RAID 5 puede fallar sin que se pierdan datos. Sin embargo, si el disco que falla no se reemplaza, los discos restantes de dicho conjunto se convierten en un punto único de fallo para todo el conjunto. Si uno falla, todos los datos del conjunto global se pierden.

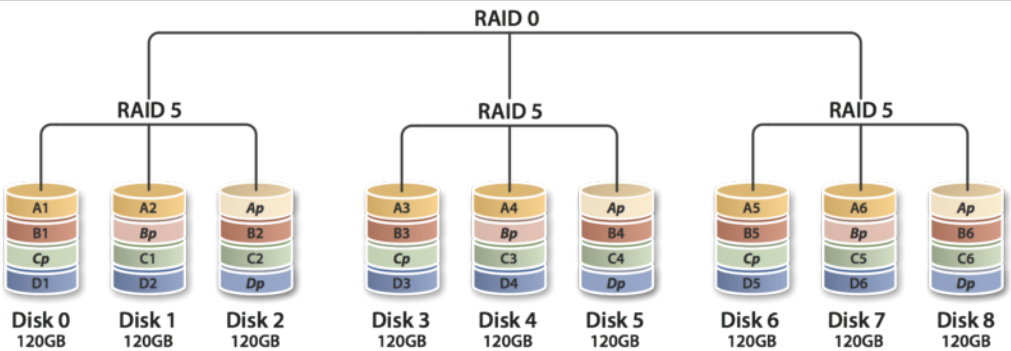


Figura 13. RAID 50