

## 12 Memoria secondaria

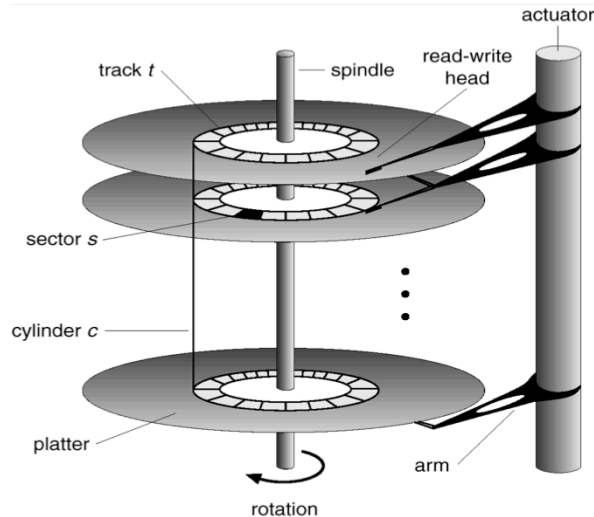
- Struttura del disco
- Scheduling del disco
- Gestione del disco
- Gestione dell'area di swap
- Memorie a stato solido

### 12.1 Struttura del disco

- Un HD è composto da una serie di “dischi” o **piatti** (platter) sovrapposti.
- ogni platter è suddiviso in una serie di **traccie** (track) circolari concentriche
- ogni traccia è suddivisa in una serie di **settori** (sector)
- L'insieme delle traccie nella stessa posizione sui diversi piatti prende il nome di **cilindro** (cylinder)
- Un braccio mobile supporta una testina di lettura e scrittura per ogni piatto

## 12.1 Struttura del disco

3



l'intera struttura è in continua rotazione (fig. 12.1)

## 12.1 Struttura del disco

4

- Logicamente, un HD può essere visto come un array unidimensionale di **blocchi logici** (solitamente) di 512 byte: la più piccola unità di trasferimento dati.
- Ogni settore contiene un blocco logico
- L'array unidimensionale di blocchi logici è tradotto in settori del disco sequenzialmente:
  - Il settore 0 è il primo settore della traccia più esterna del primo piatto (di solito il più in alto della pila)
  - Poi si numerano consecutivamente gli altri settori della traccia, i settori delle tracce più interne, e si procede con la numerazione allo stesso modo nei piatti inferiori.

## 12.1 Struttura del disco

5

- In questo modo, è possibile tradurre un numero di blocco logico in un punto preciso del disco specificato da:
  - un numero di cilindro
  - un numero di piatto all'interno del cilindro (n. di cilindro e n. di piatto identificano insieme una ben precisa traccia)
  - un numero di settore nella traccia
- In pratica, questa traduzione è piuttosto complicata a causa dei difetti di fabbricazione e delle diverse lunghezze delle tracce

## 12.1 Struttura del disco

6

- Infatti, i dischi possono avere settori difettosi che vengono nascosti dal meccanismo di mappatura tra il numero dei blocchi logici e i settori fisici sul disco
- Inoltre, il numero di settori per traccia non è costante: più una traccia è lontana dal centro del disco, più è lunga, e quindi maggiore è il numero di settori che può contenere (fino al 40% in più di una traccia interna)

## 12.4 Scheduling del disco

7

- Il sistema operativo è responsabile dell'utilizzo efficiente dell'hardware
  - per i dischi, questo significa minimizzare tempo di accesso, e massimizzare la quantità di dati trasferiti.
- Il tempo di accesso ad un settore (e quindi blocco) del disco dipende da due componenti principali
  - **Seek time (tempo di posizionamento)**: il tempo impiegato per muovere le testine sul cilindro desiderato.
  - **Rotational latency (latenza rotazionale)**: il tempo richiesto perché il disco ruoti portando il settore interessato sotto la testina magnetica.

## 12.4 Scheduling del disco

8

- Se l'HD è idle, una richiesta di lettura/scrittura può essere soddisfatta immediatamente
- Se l'HD è impegnato (a servire una richiesta) ogni nuova richiesta viene inserita in una coda di richieste pendenti per quell'HD
- Quando una richiesta è stata servita, il sistema operativo deve scegliere quella successiva da servire, in modo da cercare di massimizzare le prestazioni

## 12.4 Scheduling del disco

9

- **Il SO non può influenzare la latenza rotazionale**, che in media sarà pari a  $\frac{1}{2}$  del tempo necessario a compiere una rotazione completa.
- **Però il SO può cercare di minimizzare il seek time medio complessivo**, riordinando le richieste in attesa di essere servite in modo che le testine si debbano muovere il meno possibile.

## 12.4 Scheduling del disco

10

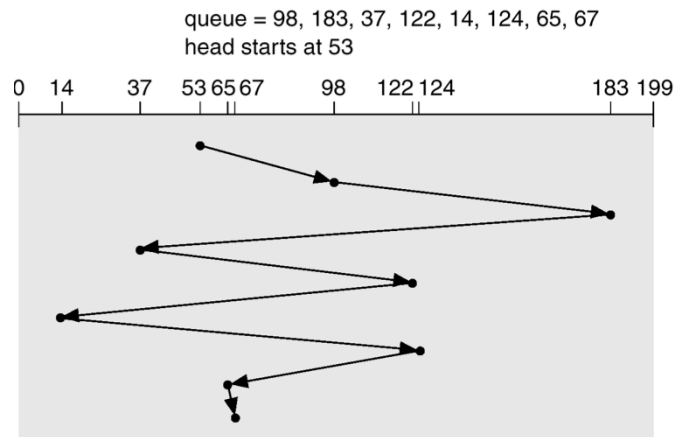
- Esistono diversi algoritmi di scheduling delle richieste di I/O del disco.
- Ne vediamo alcuni, utilizzando come esempio la seguente coda di richieste (numeri di cilindro) (0-199).

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

- Supponiamo inoltre che la testina sia posizionata inizialmente sul cilindro 53

### 12.4.1 Scheduling FCFS (fig. 12.4)

11



Lo spostamento totale della testina è di 640 cilindri. Invece che “122 – 14 – 124” non era meglio fare “122 – 124 – 14” ?

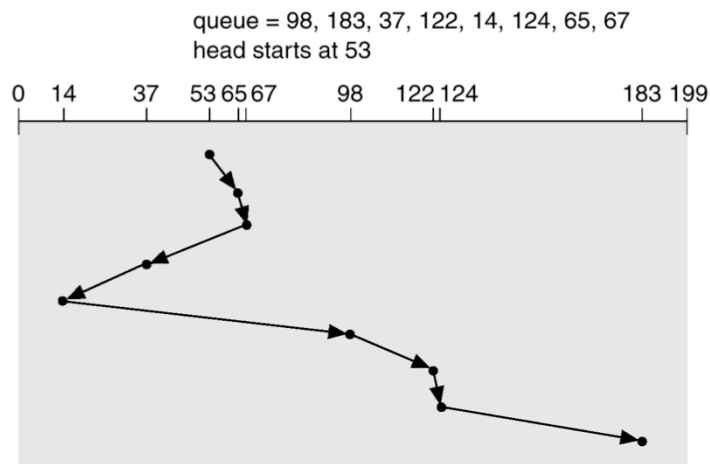
### 12.4.2 Shortest Seek Time First (SSTF)

12

- Sceglie la richiesta con il tempo di seek minimo rispetto alla posizione corrente della testina.
- SSTF è chiaramente una variante di SJF; quindi può causare starvation di alcune richieste.
- In generale dà prestazioni migliori di FCFS, ma non necessariamente ottimali
- Notate che qui conosciamo le richieste in coda, per cui non c'è nulla da “indovinare”

## 12.4.2 SSTF (fig. 12.5)

13



Lo spostamento tot. della testina è di 236 cilindri. La sequenza

“... 53, 37, 14, 65, 67, 98, ...” da uno spostamento di 208

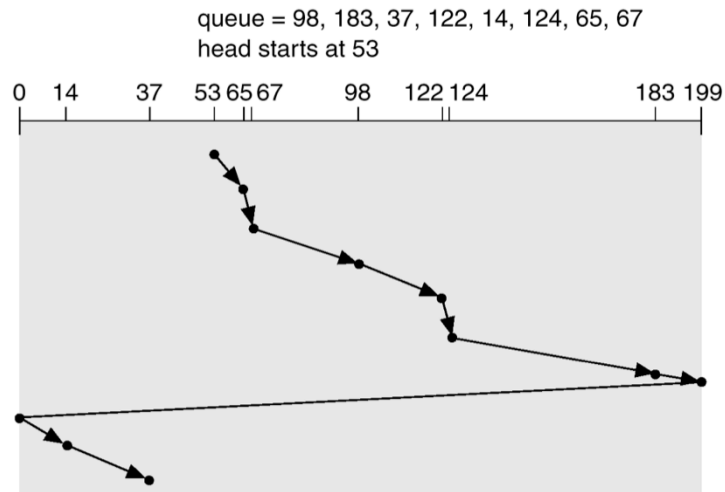
## 12.4.4 C-SCAN (Circular SCAN)

14

- Fornisce un tempo di attesa per le varie richieste più uniforme di altri algoritmi, anche se non riesce a garantire un tempo medio di attesa minimo.
- La testina si muove da un estremo all'altro del disco, servendo le richieste.
- Quando raggiunge l'estremità del disco, torna immediatamente all'inizio senza servire richieste.
- In pratica, tratta i cilindri come una lista circolare.

### 12.4.4 C-SCAN (fig. 12.7)

15



Lo spostamento totale della testina è di 183 (+ 200) cilindri.

### 12.4.5 C-LOOK

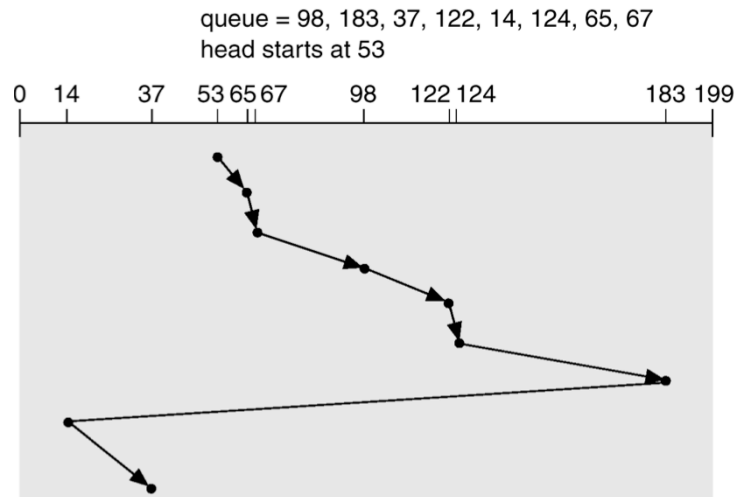
16

- Sapreste proporre un ovvio miglioramento dell'algoritmo C-SCAN?



### 12.4.5 C-LOOK (fig. 12.8)

17



Lo spostamento totale della testina è di 153 (+ 169) cilindri.

### 12.5.1 Formattazione dell'HD

18

- Per poter essere usato, un disco ha sempre bisogno di subire un processo di **formattazione a basso livello** (o **formattazione fisica**)
- Questa formattazione (fatta di solito dal costruttore dell'HD) permette di associare ad ogni settore il suo numero, e di prevedere uno spazio per inserire un codice di correzione degli errori, usato in ogni operazione di I/O su quel settore
- In fase di formattazione fisica, è anche possibile scegliere la dimensione dei blocchi fisici (il valore standard è di 512 byte per settore)

## 12.5.1 Formattazione dell'HD

19

- Il SO sottopone poi l'HD ad una formattazione logica, necessaria per creare e gestire sull'HD il File System del SO.
- Il SO crea quindi la lista dei blocchi liberi (secondo lo schema adottato), e una directory iniziale, da cui si dipartiranno tutte le altre
- Sull'HD vengono poi riservate le aree che dovranno essere gestite direttamente dal SO:
  - il boot block (che può anche essere vuoto)
  - l'area che contiene gli attributi dei file (ad esempio, gli i-node Unix o la MFT di Windows XP)

## 12.5.2 Boot Block

20

- Il Boot Block contiene il codice necessario per far partire il SO.
- All'accensione, un piccolo programma contenuto in ROM istruisce il disk controller in modo da trasferire il contenuto del Boot Block in RAM.
- Il controllo è trasferito al codice del Boot Block, che si occupa di far partire l'intero SO prelevando in codice dal disco stesso.

## 12.6 Gestione dello spazio di Swap

21

- Durante la formattazione logica dell'HD, il SO riserva a se stesso uno spazio da usare come area di swap.
- Nel caso più semplice, l'area di Swap può essere un file molto grande all'interno del FS.
- Windows usa appunto questa soluzione, e lo swap file si chiama **pagefile.sys**. Potete cambiargli dimensione, ad esempio riducendola, se avete molta RAM a disposizione. In questo modo recuperate spazio sull'hard disk.

## 12.6 Gestione dell'area di Swap

22

- Alternativamente, una partizione specifica dell'HD può essere riservata per l'area di Swap.
- Questa partizione non viene trattata allo stesso modo del FS, ma vengono usate strategie di allocazione diverse per migliorare al massimo la velocità di accesso.
- Ad esempio, l'allocazione dei blocchi alle pagine swappate può essere contigua, in modo da non dover gestire un meccanismo di ricerca dei blocchi liberi.

## 12.6 Gestione dell'area di Swap

23

- Perché tutto funzioni adeguatamente, è altamente preferibile sovradimensionare l'area di Swap, in modo che il SO trovi sempre in fretta un'area libera per lo swap di pagine e segmenti.
  - Ad esempio, in Solaris si consiglia di usare un area di Swap pari alla differenza tra lo spazio di indirizzamento logico e quello fisico
  - Linux suggerisce di usare un'area di swap di dimensione doppia a quella della RAM disponibile.
- Nei sistemi con più dischi, si può prevedere un'area di swap per ciascun disco, in modo da poterle usare contemporaneamente, bilanciando il carico di lavoro

## Memorie a Stato Solido

24

- Negli ultimi anni sono andate sempre più diffondendosi le **memorie a stato solido** o **flash memory** come supporto di memoria secondaria.
- Questo tipo di memoria è usato per implementare la memoria di massa nei dispositivi mobili, come gli smart phone e i tablet.
- Più di recente, le memorie flash vengono anche usate nei computer, ad integrazione (più raramente in sostituzione) del disco rigido.

## Memorie a Stato Solido

25

- Tecnicamente, sono particolari tipi di EEPROM (Electrically Erasable Programmable Read-Only Memory): possono non solo essere lette, ma anche riscritte. Tuttavia, la riscrittura avviene a livello di interi blocchi di dati, ossia anche i dati non modificati del blocco vanno riscritti.
- Sono memorie permanenti ma con un limite di circa 100.000 riscritture per blocco
- Prima di essere riscritto, il blocco deve essere cancellato (appunto con un processo di “flashing”, da cui il nome delle memorie). Questo rende le memorie flash più lente in scrittura che in lettura.

## Memorie a Stato Solido

26

- Costi:
  - **Hard disk**: circa 0,09\$ al GigaByte
  - **Flash memory**: circa 2\$ al GigaByte,
  - **RAM memory**: circa 20\$÷40\$ al GigaByte
- Velocità in lettura: (sia T il tempo di lettura in RAM)
  - **Hard disk**: circa  $1000 \times T$
  - **Flash memory**: circa  $4 \times T$
  - **RAM memory**: T
- Velocità in scrittura: (sia T il tempo di scrittura in RAM)
  - **Hard disk**: circa  $1000 \times T$
  - **Flash memory**: da  $10 \times T$  fino a  $100 \times T$
  - **RAM memory**: T

## Memorie a Stato Solido

27

- Attualmente le memorie flash vengono usate, oltre che come supporto di memoria rimovibile anche per complementare il funzionamento dell'hard disk, sostanzialmente in due modi (anche combinabili):
- Come livello di memoria cache intermedio tra l'hard disk e la memoria RAM
- Come un secondo hard disk, più piccolo (32 ÷ 128 GB) ma molto più veloce, su cui tenere memorizzato (completamente o parzialmente) il sistema operativo, gli applicativi, i file utente più usati, l'area di swap. Tutti gli altri file verranno memorizzati sull'hard disk.