Università di Udine — Facoltà di Scienze MM.FF.NN.

Laurea in Informatica — A.A. 2005/06

Trasparenze del Corso di *Sistemi Operativi*

Ivan Scagnetto

Università di Udine

Struttura dei dischi

- La gestione dei dischi riveste particolare importanza.
- I dischi sono indirizzati come dei grandi array monodimensionali di *blocchi logici*, dove il blocco logico è la più piccola unità di trasferimento con il controller.
- L'array monodimensionale è mappato sui settori del disco in modo sequenziale.
  - Settore 0 = primo settore della prima traccia del cilindro più esterno
  - la mappatura procede in ordine sulla traccia, poi sulle rimanenti tracce dello stesso cilindro, poi attraverso i rimanenti cilindri dal più esterno verso il più interno.

Schedulazione dei dischi

- Il sistema operativo è responsabile dell'uso efficiente dell'hardware. Per i dischi: bassi *tempi di accesso* e alta *banda di utilizzo*.
- Il tempo di accesso ha 2 componenti principali, dati dall'hardware:
  - *Seek time* = il tempo (medio) per spostare le testine sul cilindro contenente il settore richiesto.
  - *Latenza rotazionale* = il tempo aggiuntivo necessario affinché il settore richiesto passi sotto la testina.
- Tenere traccia della posizione angolare dei dischi è difficile, mentre si sa bene su quale cilindro si trova la testina
- Obiettivo: mimimizzare il tempo speso in seek
- Tempo di seek $\approx$ distanza di seek; quindi: mimimizzare la distanza di seek
- Banda di disco = il numero totale di byte trasferiti, diviso il tempo totale dalla prima richiesta di servizio e il completamento dell'ultimo trasferimento.

Schedulazione dei dischi (Cont.)

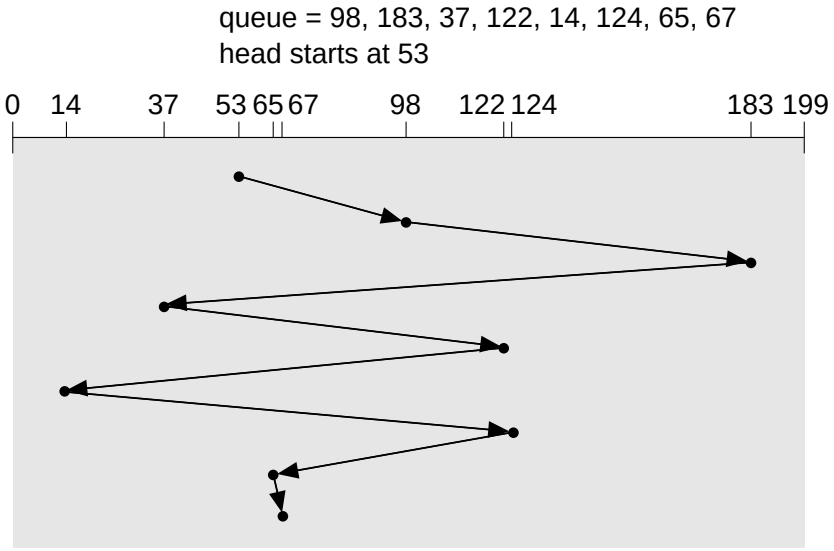
- Ci sono molti algoritmi per schedulare le richieste di I/O di disco.
- Al solito, una trattazione formale esula dal corso
- Illustreremo con una coda di richieste d'esempio, su un range di cilindri 0–199:

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

- Supponiamo che la posizione attuale della testina sia 53

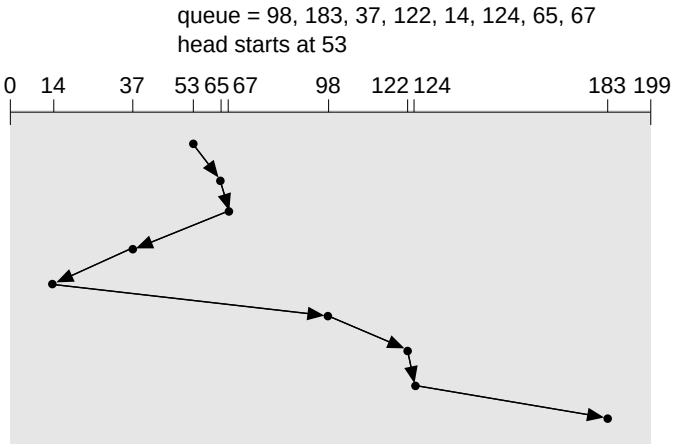
FCFS

Sull'esempio: distanza totale di 640 cilindri



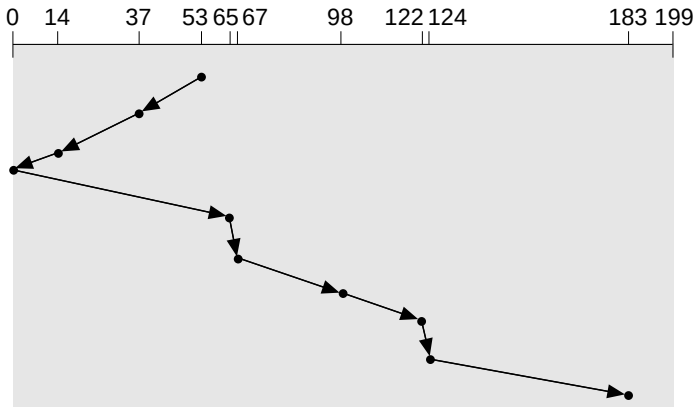
Shortest Seek Time First (SSTF)

- Si seleziona la richiesta con il minor tempo di seek dalla posizione corrente
- SSTF è una forma di scheduling SJF; può causare *starvation*.
- Sulla nostra coda di esempio: distanza totale di 236 cilindri (36% di FCFS).



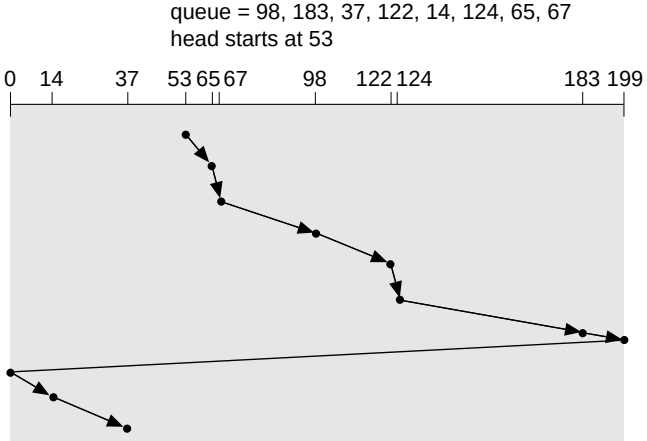
SCAN (o "dell'ascensore")

- Il braccio scandisce l'intera superficie del disco, da un estremo all'altro, servendo le richieste man mano. Agli estremi si inverte la direzione.
- Sulla nostra coda di esempio: distanza totale di 208 cilindri



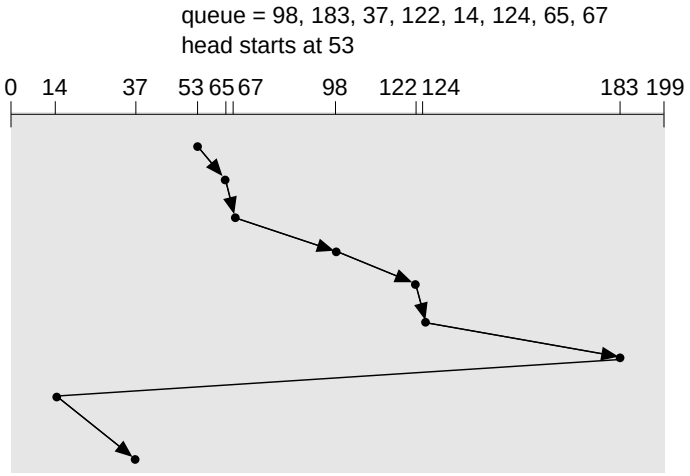
C-SCAN

- Garantisce un tempo di attesa più uniforme e equo di SCAN
- Tratta i cilindri come in lista circolare, scandita in rotazione dalla testina si muove da un estremo all'altro del disco. Quando arriva alla fine, ritorna immediatamente all'inizio del disco senza servire niente durante il rientro.



C-LOOK

- Miglioramento del C-SCAN (esiste anche il semplice LOOK)
- Il braccio si sposta solo fino alla richiesta attualmente più estrema, non fino alla fine del disco, e poi inverte direzione immediatamente.



Quale algoritmo per lo scheduling dei dischi?

- SSTF è molto comune e semplice da implementare, e abbastanza efficiente
- SCAN e C-SCAN sono migliori per i sistemi con un grande carico di I/O con i dischi (si evita starvation)
- Le performance dipendono dal numero e tipi di richieste
- Le richieste ai dischi dipendono molto da come vengono allocati i file, ossia da come è implementato il file system.
- L'algoritmo di scheduling dei dischi dovrebbe essere un modulo separato dal resto del kernel, facilmente rimpiazzabile se necessario.
- Sia SSTF che LOOK (e varianti circolari) sono scelte ragionevoli come algoritmi di default.

Gestione dell'area di swap

- L'area di swap è parte di disco usata dal gestore della memoria come estensione della memoria principale.
- Può essere ricavata dal file system normale o (meglio) in una partizione separata.
- Gestione dell'area di swap
  - 4.3BSD: alloca lo spazio appena parte il processo per i segmenti text e data. Lo stack, man mano che cresce.
  - Solaris 2: si alloca una pagina sullo stack solo quando si deve fare un page-out, non alla creazione della pagina virtuale.
  - Windows 2000: Viene allocato spazio sul file di swap per ogni pagina virtuale non corrispondente a nessun file sul file system (es: DLL).

Affidabilità e performance dei dischi

- aumenta la differenza di velocità tra applicazioni e dischi
- le cache non sempre sono efficaci (es. transazioni, dati da esperimenti)
- Suddividere il carico tra più dischi che cooperano per offrire l'immagine di un disco unitario virtuale
- Problema di affidabilità:

$$MTBF_{array} = \frac{MTBF_{disco}}{\#dischi}$$

RAID

- RAID = Redundant Array of Inexpensive/Independent Disks: implementa affidabilità del sistema memorizzando informazione ridondante.
- La ridondanza viene gestita dal controller (RAID *hardware*) — o molto spesso dal driver (RAID *software*).
- Diversi *livelli* (organizzazioni), a seconda del tipo di ridondanza

  0: *striping*: i dati vengono "affettati" e parallelizzati. Altissima performance, non c'è ridondanza.

  1: *Mirroring* o *shadowing*: duplicato di interi dischi. Eccellente resistenza ai crash, basse performance in scrittura

  5: *Block interleaved parity:* come lo striping, ma un disco a turno per ogni stripe viene dedicato a contenere l'informazione di parità del resto della stripe. Alta resistenza, discrete performance (in caso di aggiornamento di un settore, bisogna ricalcolare la parità)