

## Considerazioni conclusive

Da quanto fin qui detto, appare evidente che le soluzioni presentate sono tra loro complementari e possono quindi essere usate contemporaneamente sullo stesso sistema, in modo da sfruttare tutti i vantaggi che ciascuna di esse presenta.

Di conseguenza, la scelta delle periferiche e delle tecniche da usare per il loro collegamento al sistema deve essere operata in base alle esigenze di efficienza, economicità e velocità presenti all'interno del centro di calcolo.

↳ Se, per esempio, in un centro di calcolo si devono gestire molti lavori che richiedono spesso la stampa di lunghi tabulati, sarà sufficiente adottare una gestione a spool e collegare al sistema, attraverso un canale multiplexato, una o più stampanti veloci dotate di un buffer piuttosto ampio.

Se invece il centro è collegato con utenti che lavorano quasi esclusivamente in modo interattivo attraverso terminali video, sarà meglio predisporre un adeguato numero di canali e di linee di collegamento veloci, in modo tale da ridurre al minimo la possibilità che una richiesta di I/O rimanga a lungo in attesa.

## LE TECNICHE DI SCHEDULAZIONE DEL DISCO

Le periferiche che condizionano maggiormente l'efficienza di un sistema di elaborazione sono indubbiamente le unità a disco, che rappresentano le memorie di massa più utilizzate.

Si capisce quindi come l'ottimizzazione dei tempi medi di accesso al disco, pianificando le richieste dei processi (*politiche di schedulazione*) possa incidere in modo non trascurabile sui tempi di risposta dei singoli programmi.

In particolare, le politiche di schedulazione si basano su principi molto simili a quelli esposti per la schedulazione dei processi; vengono però applicate ai processi in coda di attesa della periferica allo scopo di determinare il modo più veloce per servire le richieste di accesso.

Nella realizzazione degli algoritmi di schedulazione del disco, comunque, bisogna tener conto dei seguenti fattori, che influenzano il tempo di risposta della periferica:

- *tempo di seek*, che rappresenta il tempo necessario alle testine per raggiungere il cilindro indirizzato dall'operazione di I/O;
- *tempo di latenza*, che rappresenta il tempo di rotazione necessario per portare il settore indirizzato sotto la testina;
- *tempo di trasferimento*, che rappresenta il tempo necessario per il trasferimento fisico dei dati dal blocco selezionato al buffer in memoria o viceversa.

In particolare, la maggior parte delle politiche di schedulazione tende ad ottimizzare soprattutto i tempi di seek, molto più gravosi di quelli di latenza e di trasferimento.

### Schedulazione FCFS (First Come First Served)

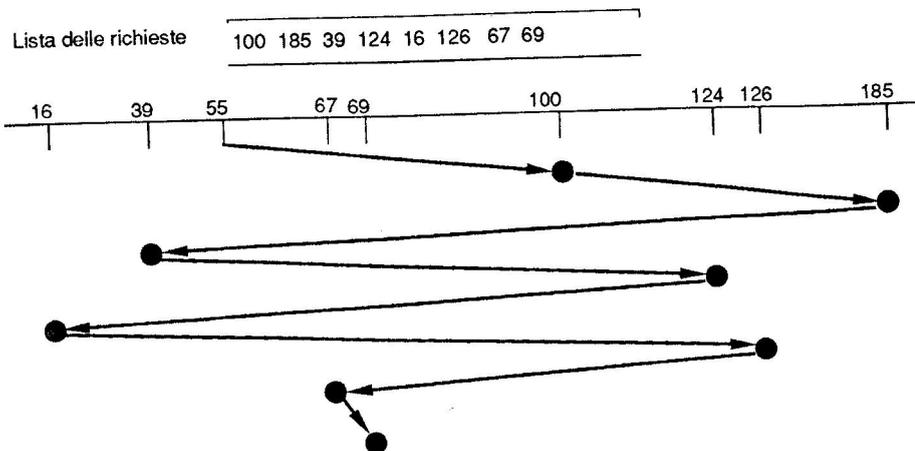
Con questa strategia non viene effettuato alcun particolare ordinamento tra i processi in attesa, e ogni volta viene servito quello che si trova in testa alla coda (il primo arrivato).

Si tratta di un algoritmo molto semplice da implementare ed è intrinsecamente imparziale, ma per contro non offre un buon tempo medio di risposta, dato che le testine sono costrette a muoversi casualmente sulla superficie del disco in accordo con le tracce coinvolte nelle singole richieste dei processi.

Per esempio, se le richieste di accesso pervenute coinvolgono nell'ordine le seguenti tracce:

100 185 39 124 16 126 67 69

e la testina, all'inizio, è posizionata sulla traccia 55, lo spostamento totale risulta essere pari a 640:



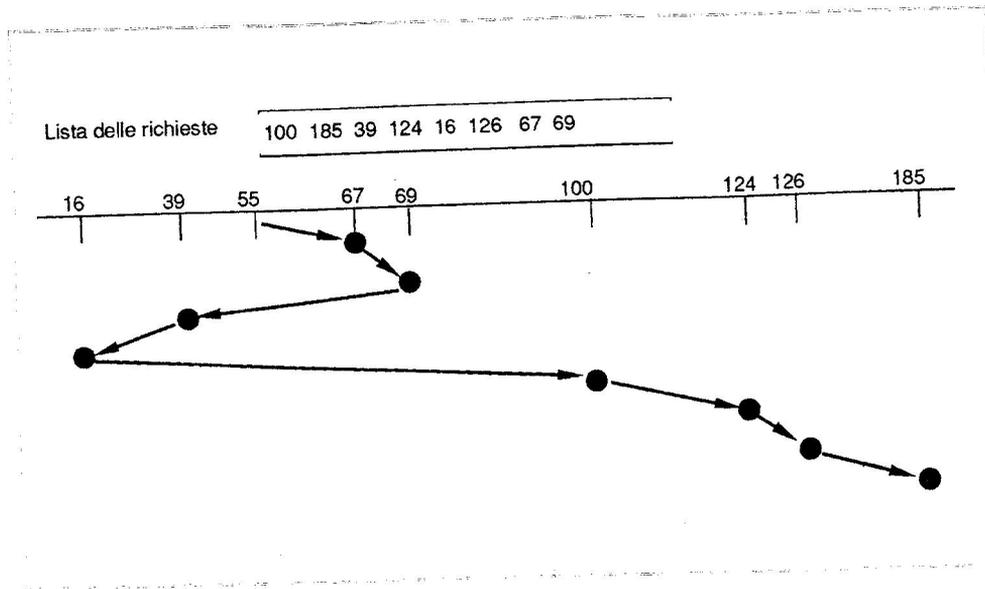
Come si può vedere, la testina è costretta ad effettuare diverse oscillazioni in avanti e indietro e i tempi di servizio risultano notevolmente appesantiti da questi movimenti.

### Schedulazione SSTF (Shortest Seek Time First)

Con questa strategia viene servita per prima la richiesta che indirizza la traccia più vicina a quella su cui è già posizionata la testina.

Si tratta di un algoritmo che privilegia fortemente le richieste localizzate e penalizza invece quelle molto distribuite sulla superficie del disco e, pur fornendo un tempo medio di risposta migliore del precedente, può essere ulteriormente migliorato.

Facendo riferimento all'esempio precedente, il movimento delle testine può essere così rappresentato:



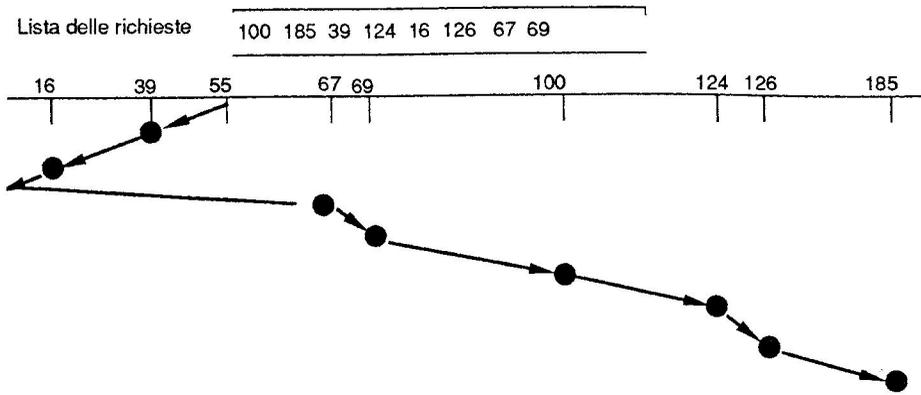
### Schedulazione SCAN (politica dell'ascensore)

Questa strategia è un'evoluzione della SSTF e la selezione di una richiesta viene fatta basandosi sul tempo di seek minore, mantenendosi però nella direzione corrente del movimento della testina, che non viene cambiata finché non si raggiunge l'estremità della superficie del disco.

La testina, quindi, effettua una scansione continua del disco da un'estremità all'altra, prima in una direzione e poi nell'altra.

Anche in questo caso, possiamo descrivere il percorso delle testine sulla superficie del disco, ricordando che le richieste di accesso pervenute coinvolgono nell'ordine le seguenti tracce:

100    185    39    124    16    126    67    69



### Schedulazione C-SCAN (Circular SCAN)

Questa strategia rappresenta una variante della precedente ed è stata progettata per fornire un tempo di attesa più uniforme.

Come nella schedulazione SCAN, la testina si sposta da un'estremità all'altra servendo le richieste man mano che le incontra, ma una volta raggiunta un'estremità, torna immediatamente all'inizio del disco senza servire le richieste che si trovano sul percorso di ritorno.

Nel nostro esempio si avrebbe il seguente percorso:

