Fondamenti dei Sistemi Operativi

Device Manager

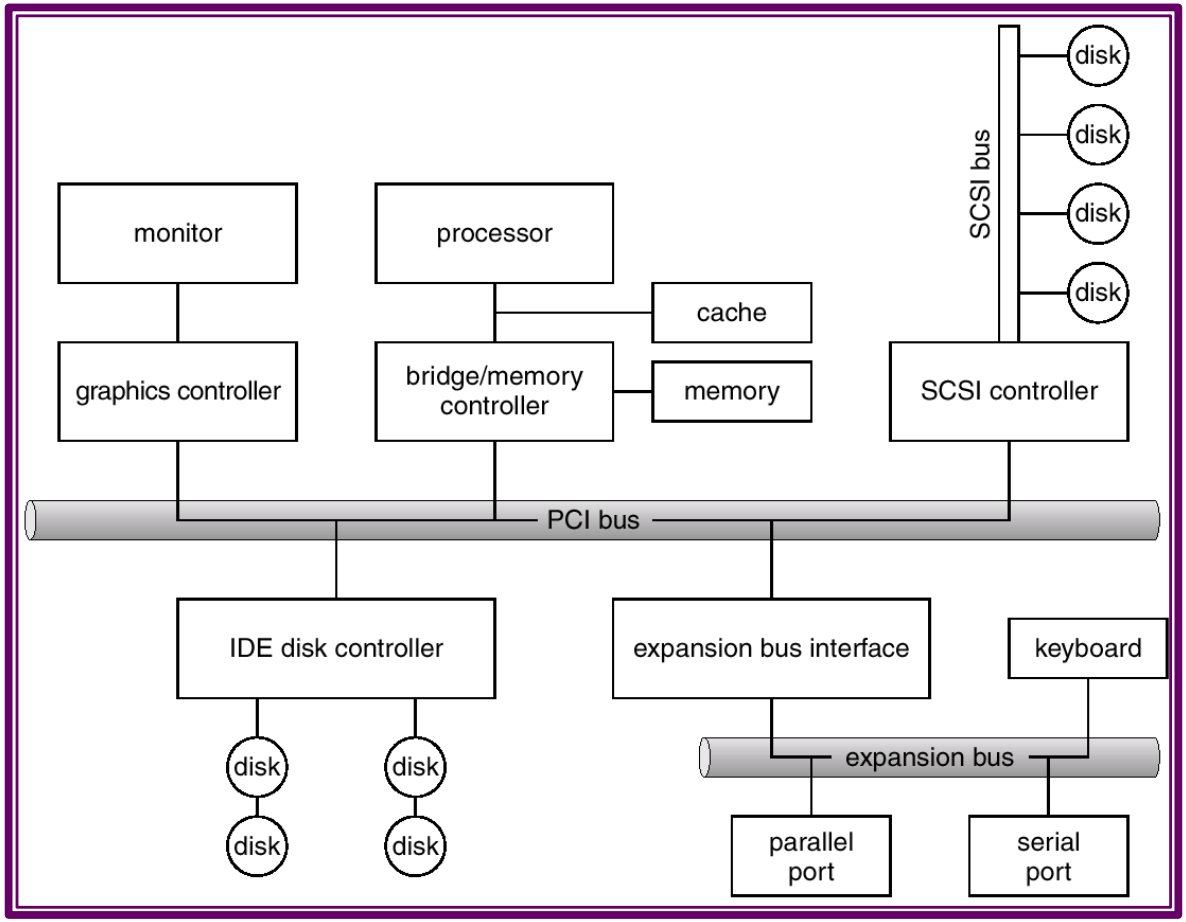
Sommario

- Dispositivi di I/O
- Interfaccia (o controller) e software di pilotaggio (driver) di un dispositivo
- Schedulazione dei dischi: i parametri
- Schedulazione dei dischi: gli algoritmi FCFS, SSTF, SCAN e LOOK
- Lo SPOOL: scopo, realizzazione e moduli

I/O devices

- Incredibile varietà di dispositivi di I/O
- Concetti comuni
  - Porta
  - Bus
  - Controller

A Typical PC Bus Structure



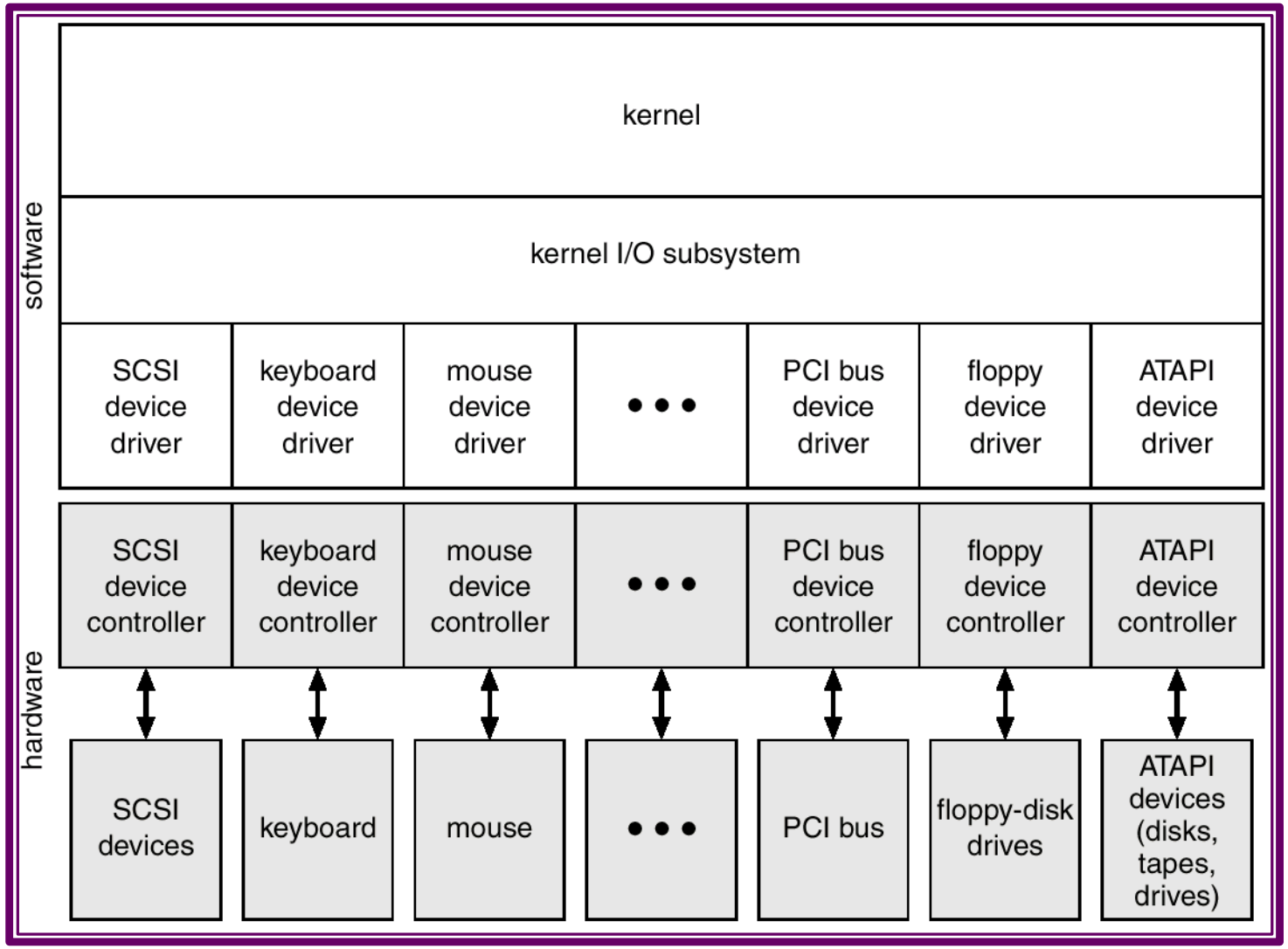
Application I/O Interface

- Le chiamate all'I/O system incapsulano i comportamenti dei dispositivi in classi generiche
- Lo strato dei device-driver layer nasconde le differenze tra I/O controller
- I device variano per diversi aspetti
  - funzionamento *character-stream* o a blocchi
  - accesso sequenzial o ad accesso diretto
  - condivisi o dedicati
  - velocità operativa
  - read-write, read only, or write only

I/O devices characteristics

aspect	variation	example
data-transfer mode	character block	terminal disk
access method	sequential random	modem CD-ROM
transfer schedule	synchronous asynchronous	tape keyboard
sharing	dedicated sharable	tape keyboard
device speed	latency seek time transfer rate delay between operations	
I/O direction	read only write only readÐwrite	CD-ROM graphics controller disk

I/O hardware (controllers) and software (drivers)



Schedulazione dei dischi

Il sistema operativo è responsabile dell'uso efficiente dell'hardware.

Per i dischi: alta banda di utilizzo e bassi tempi di accesso.

Banda di disco = il numero totale di byte trasferiti, diviso il tempo totale tra la prima richiesta di servizio e il completamento dell'ultimo trasferimento.

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.
Search time o Latenza rotazionale = il tempo aggiuntivo necessario affinché il settore richiesto passi sotto la testina.
Tenere traccia della posizione angolare dei dischi (e quindi ottimizzare il search time) è difficile, mentre si sa bene su quale cilindro si trova la testina

Obiettivo: Poichè non si può agire sul search time, occorre mimimizzare il tempo di seek. Il seek time dipende dalla distanza di seek; quindi bisogna mimimizzare la distanza di seek.

Schedulazione dei dischi (cont.)

Ci sono molti algoritmi per schedulare le richieste di I/O di disco.

Illustreremo con una coda di richieste d'esempio, su un range di cilindri

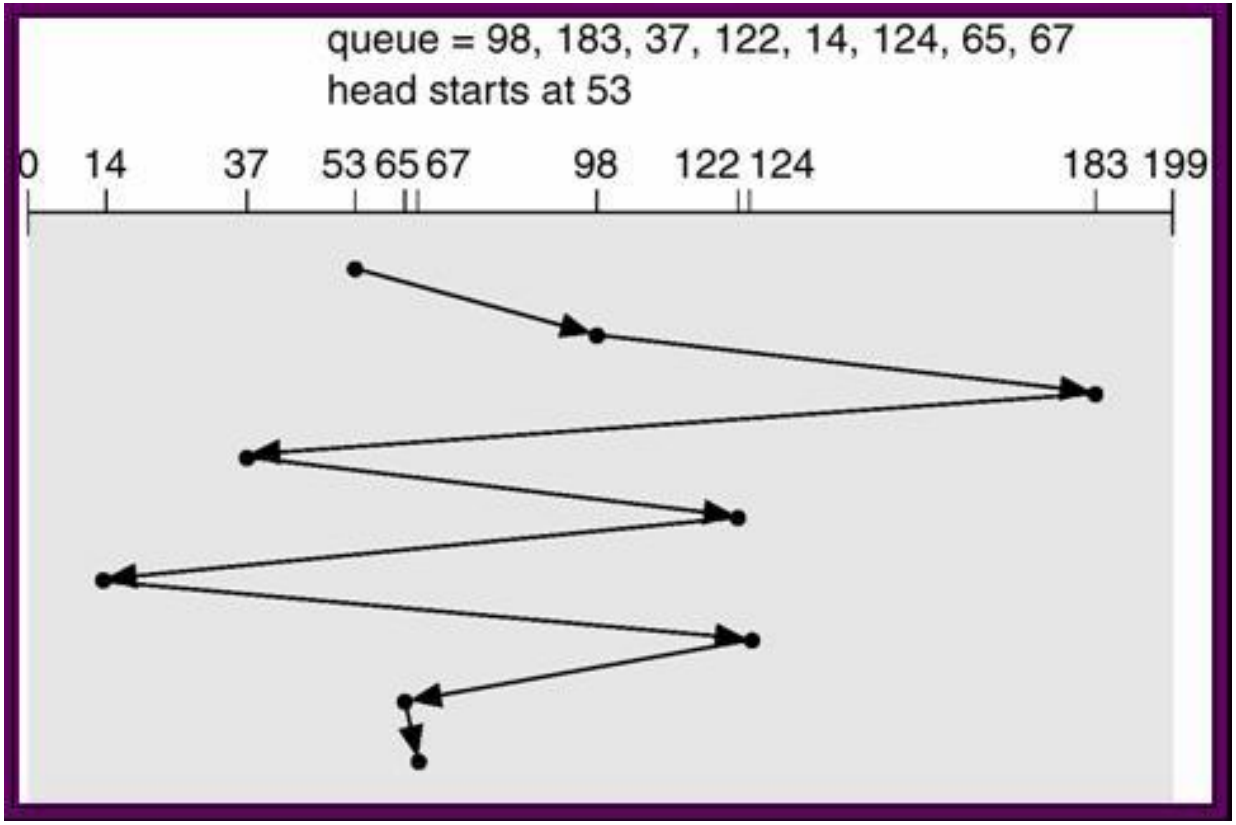
Supponiamo che la posizione attuale della testina sia 53, che il disco sia costituito da 200 cilindri, numerati da 0 a 199 e che la coda delle richieste cronologicamente pervenute sia la seguente:

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

Disk Scheduling algorithms

FCFS

total head movement: 640 cylinders

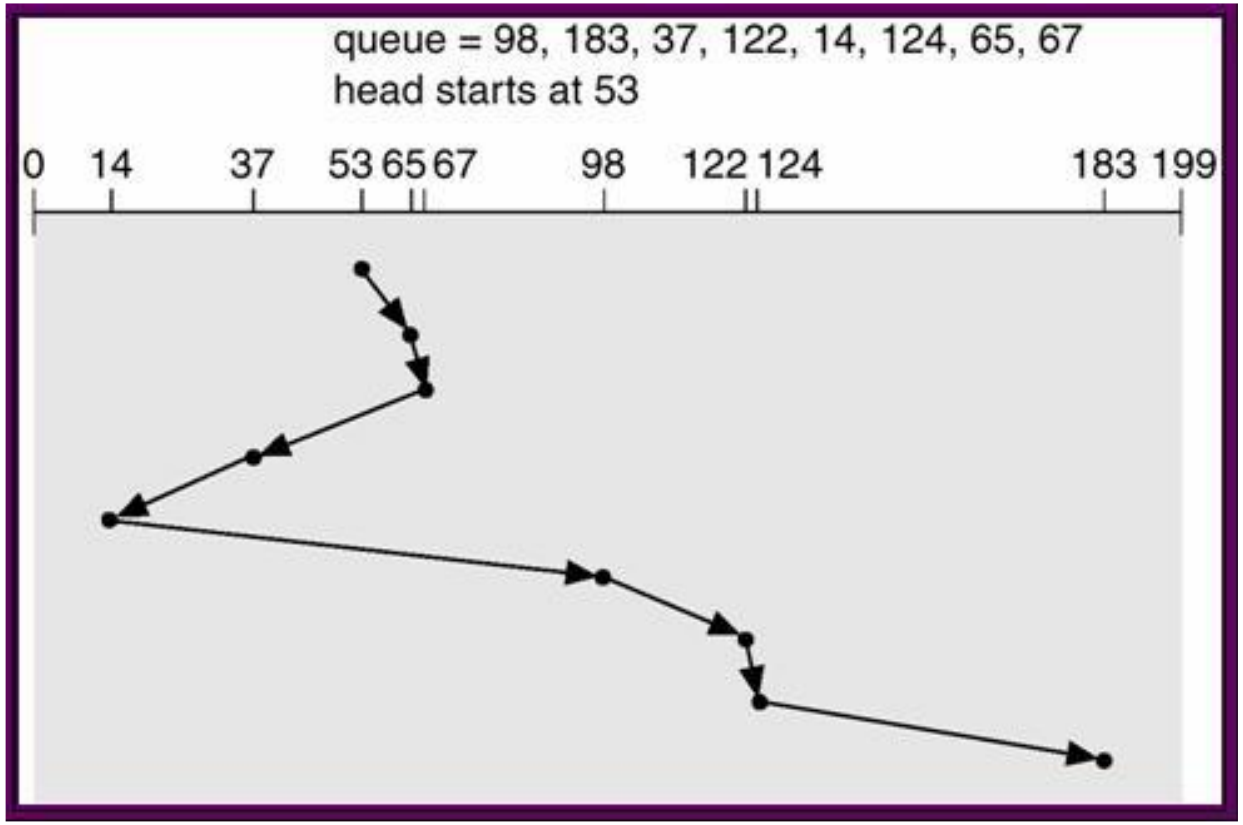


Disk Scheduling algorithms

SSTF (Shortest Seek Time First)

- sceglie la richiesta con il **tempo di seek minimo rispetto alla posizione corrente** della testina.
- è una variante dell'algoritmo Shortest Job First (SJF), che vedremo essere usato per la schedulazione dei processi; può causare *starvation* di alcune richieste.

total head movement: 236 cylinders

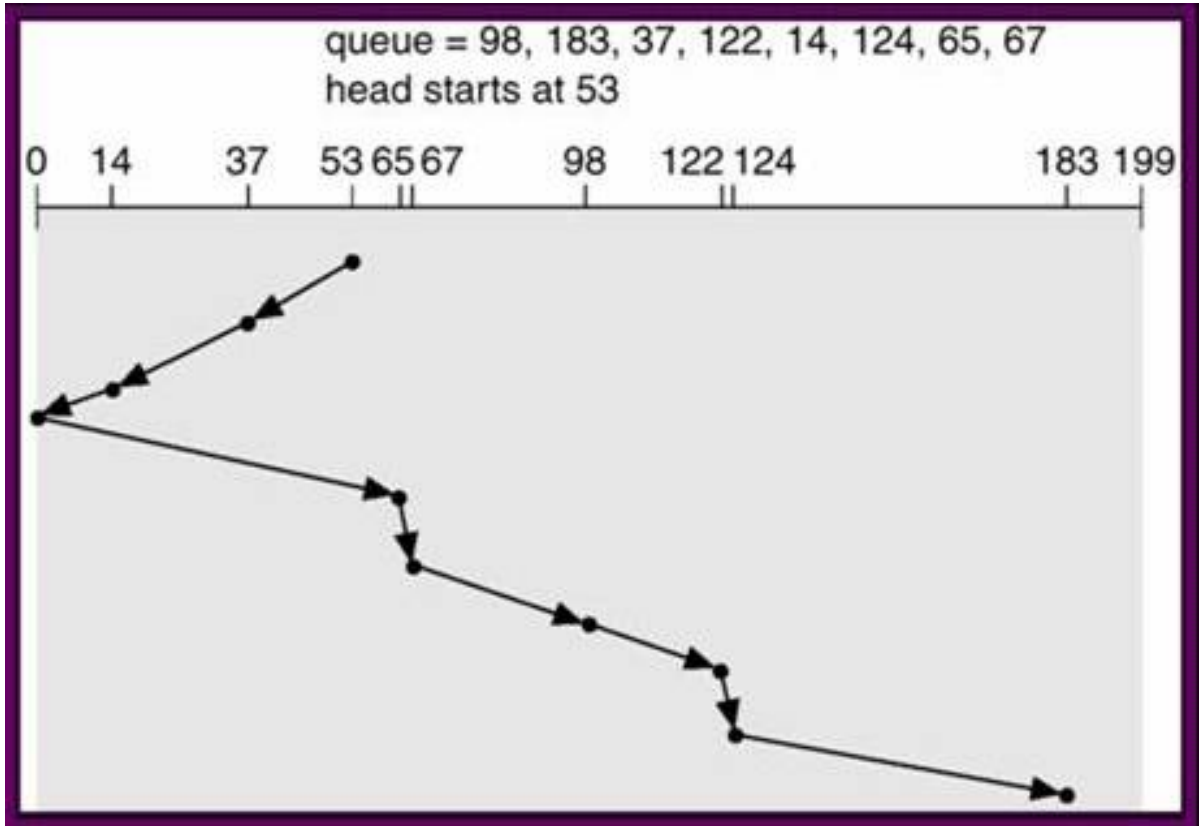


Disk Scheduling algorithms

SCAN

- La **testina si muove da un estremo all'altro del disco**, servendo le richieste. Quando raggiunge l'estremità del disco, inverte la direzione di spostamento, continuando a servire le richieste sul cammino.
- Talvolta viene chiamato algoritmo di scheduling dell'"ascensore".

total head movement: 236 cylinders.

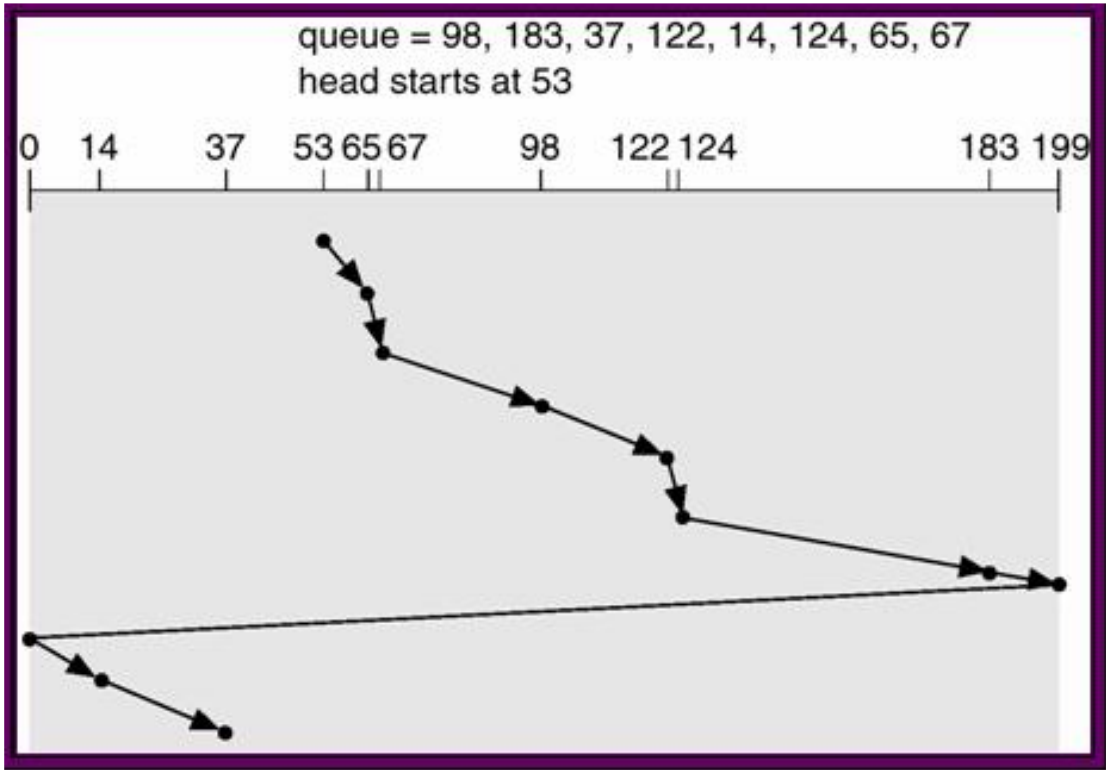


Disk Scheduling algorithms

C-SCAN (Circular Scan)

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

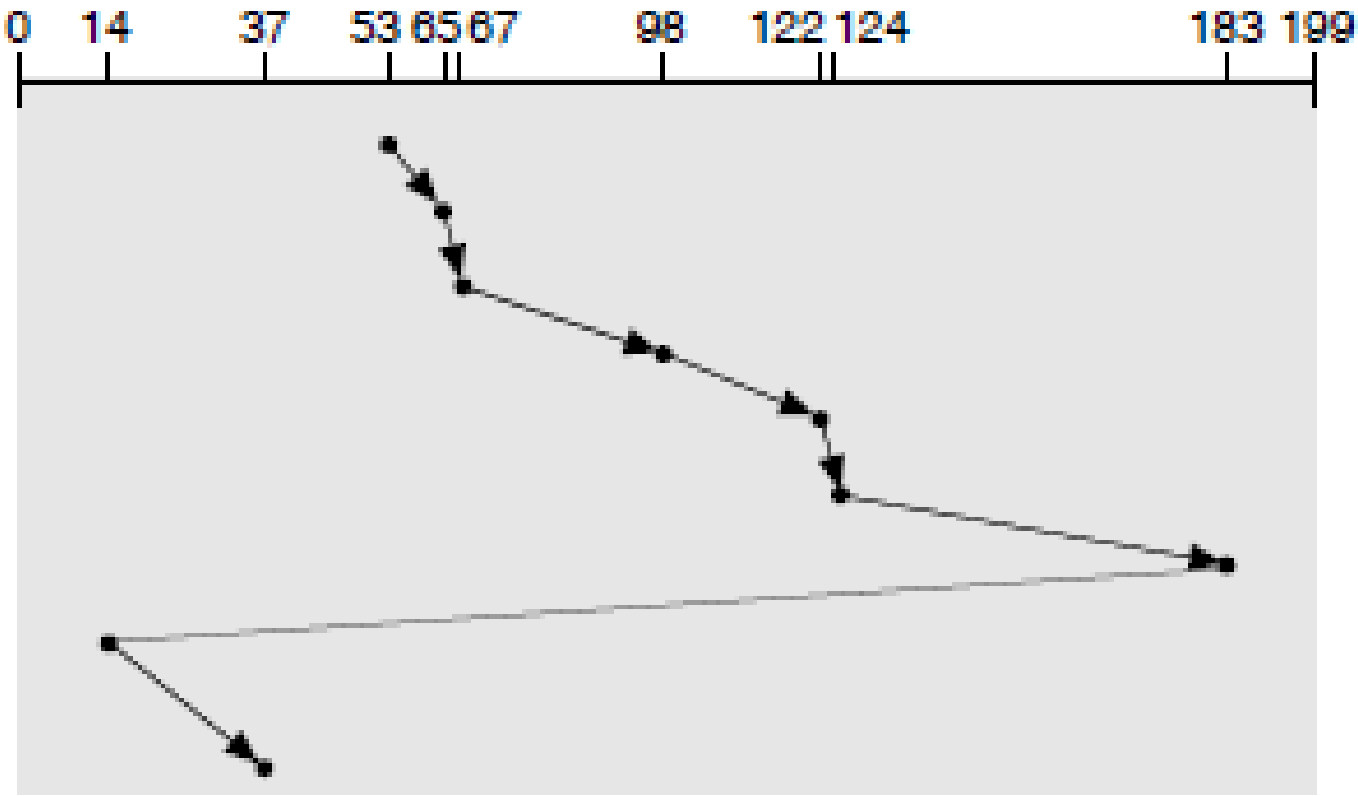
total head movement: 382 cylinders.



Disk Scheduling algorithms

C-LOOK

- Miglioramento del C-SCAN (esiste anche il semplice LOOK)
- **Il braccio si sposta solo fino alla richiesta attualmente estrema, ma non fino alla fine del disco**, e poi inverte direzione immediatamente.
  total head movement: 322 cylinders.



Disk Scheduling algorithms

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 dai 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.

SPOOL
Simultaneous Peripheral Operation On-Line

I dispositivi condivisibili sono quelli ad accesso diretto (come il disco).

I dispositivi non condivisibili sono generalmente quelli sequenziali (come la stampante). Però, anche quest'ultimi, possono essere resi condivisibili.

Condivisione di dispositivi non condividibili (virtualizzazione dei dispositivi)

Ogni operazione di scrittura sulla stampante viene trasformata in un'operazione di scrittura su un file di disco.

I programmi in esecuzione possono pertanto eseguire contemporaneamente le loro operazioni di stampa.

Le righe di stampa si accumulano sul file.

Quando un programma avrà concluso la sua esecuzione, verrà eseguito il trasferimento alla stampante dei record del file prodotti dal programma in questione.

Si ha, quindi, una virtualizzazione del dispositivo stampante, rendendo addirittura possibile associare una o più stampanti ad ogni processo.

La stampa effettiva avviene solo dopo che il programma ha completato la sua esecuzione.

Con riferimento al diagramma degli stati, ciò viene espresso introducendo dopo lo stato di COMPLETE un ulteriore stato di OUTPUT, che caratterizza i processi terminati, ma di cui non è stato ancora effettuato il trasferimento indicato in precedenza.

La realizzazione di un sistema di spool

L'operazione di **WRITE** su stampante viene trasformata in un'operazione di scrittura su un file costituito da un **pool** di record, cioè costituito da un insieme di record concatenati tra loro.

Il SO tiene traccia, per ogni stampante virtuale associata ad un processo, dell'indirizzo del primo e dell'ultimo record verso cui sono state dirottate le righe di stampa prodotte dal processo.

Attraverso il concatenamento è possibile ricostruire la successione di tutte le righe di stampa prodotte da ciascun processo. Inoltre il SO tiene traccia dell'indirizzo del primo e dell'ultimo record liberi nel pool.

Tipo record	*Indirizzo primo record*	*Indirizzo ultimo record*
processo A	5	37
processo B	8	23
. . . . .	. . . . .	. . . . .
processo I	12	43
disponibile	0	99

IL SISTEMA DI SPOOL

È costituito da due moduli:

- OUTPUT.STORE
- OUTPUT.FETCH

L'OUTPUT.STORE intercetta le richieste di scrittura su stampante e le dirotta sul pool di record.

Il modulo OUTPUT.FETCH stampa i record (in sequenza) del programma che ha terminato la sua esecuzione.

Il sistema di spool può essere realizzato tramite un monitor (cfr. il tema relativo alla "Sincronizzazione"), del quale i due moduli anzidetti rappresentano le procedure pubbliche.

I due moduli condividono l'uso del pool di record e della tabella di indirizzamento.

La mancanza di record disponibili costringerà l'Output Store a mettersi in attesa (sulla coda delle condizioni interne) del completamento delle operazioni dell'Output Fetch. Quest'ultimo infatti determinerà il liberarsi di record e quindi la riattivazione dell'Output Store.

Se nel pool non vi sono più record disponibili e nessun programma termina l'esecuzione, si determina una situazione di *deadlock*.