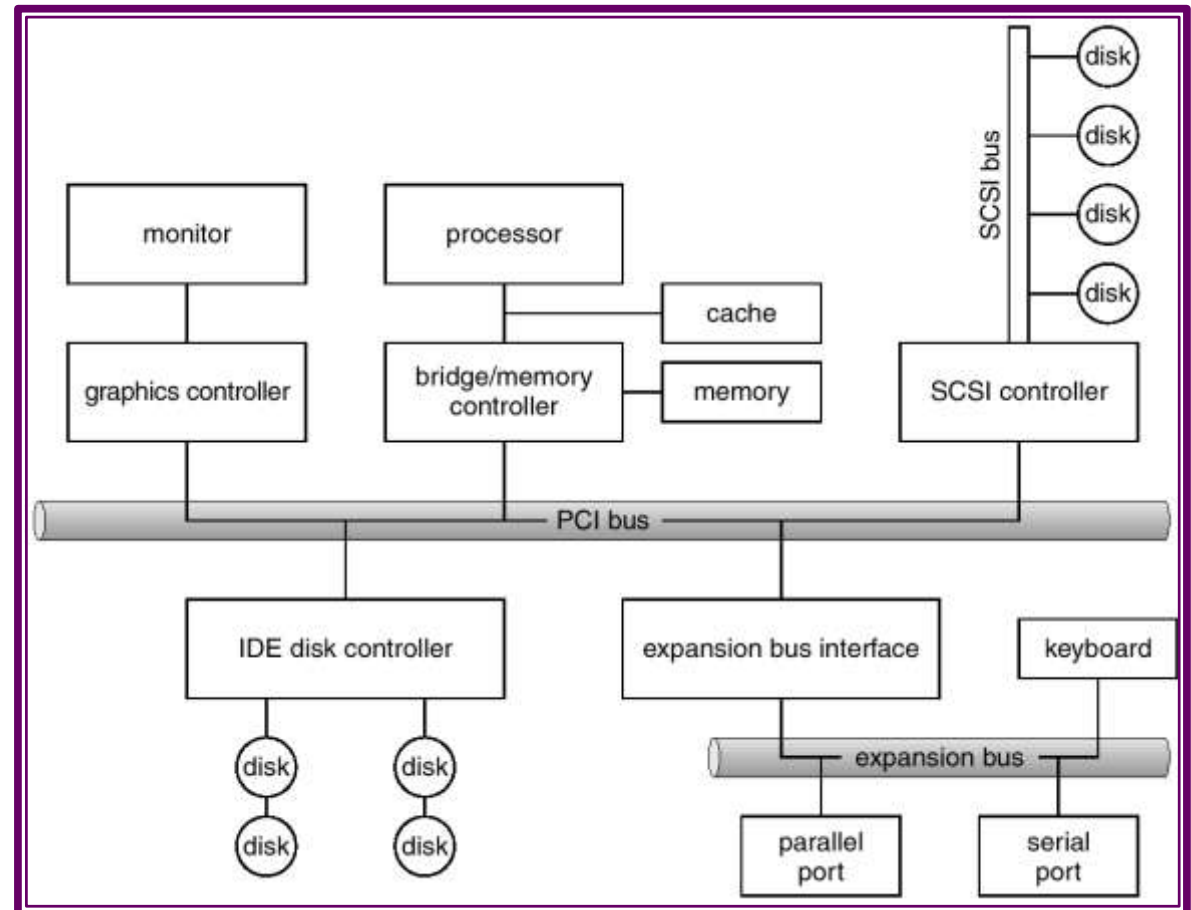


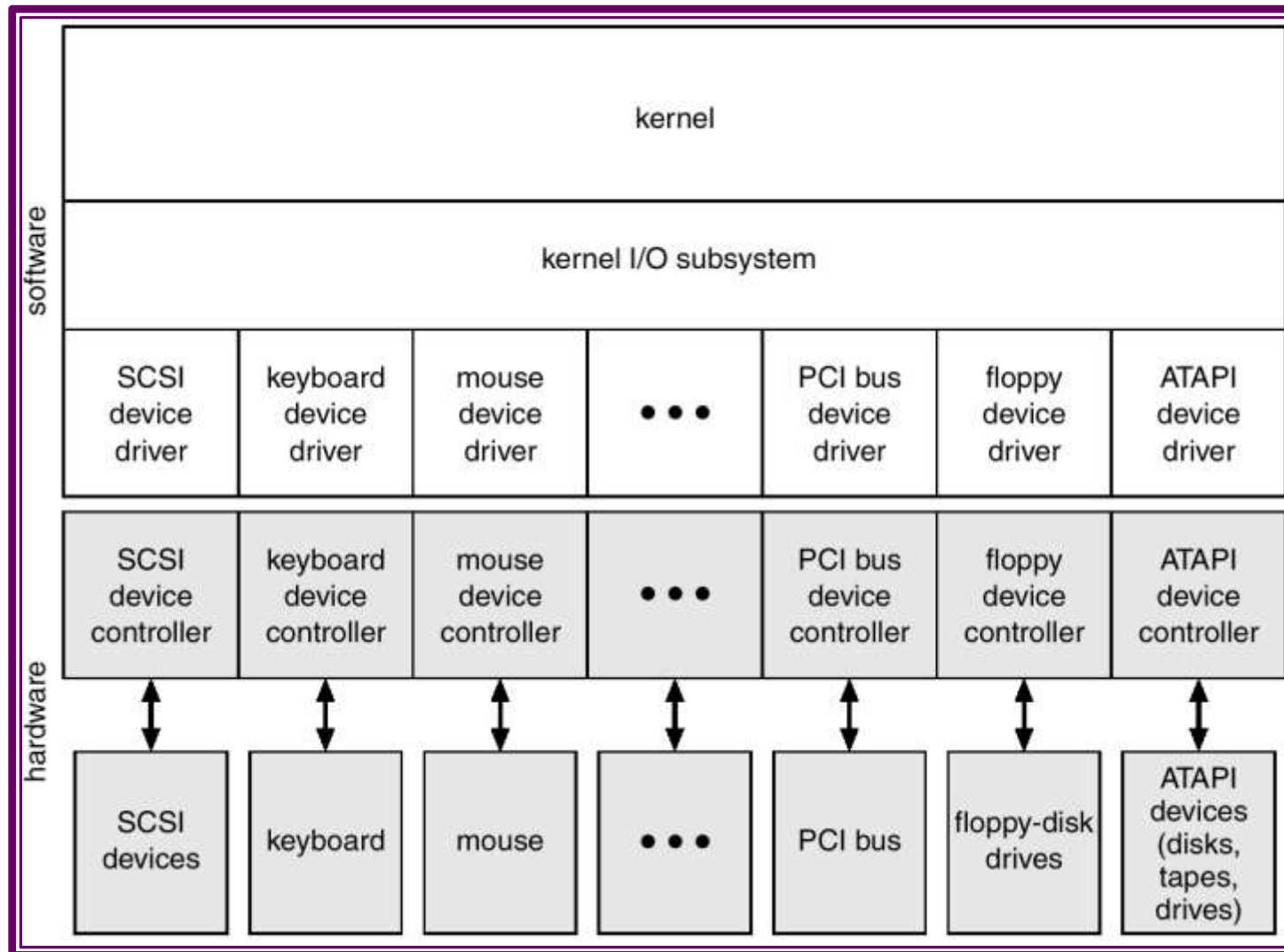
## I/O DEVICES

- Incredible variety of I/O devices
- Common concepts
  - Φ Port
  - Φ Bus
  - Φ Controller (host adapter)
- I/O instructions control devices
- Devices have addresses, used by
  - Φ Direct I/O instructions
  - Φ Memory-mapped I/O

### *A Typical PC Bus Structure*



## I/O HARDWARE AND SOFTWARE (DRIVERS)



## DISK STRUCTURE

- ↳ Disk drives are addressed as large 1-dimensional arrays of *logical blocks*, where the logical block is the smallest unit of transfer.
- ↳ The 1-dimensional array of logical blocks is mapped into the sectors of the disk sequentially.
  - Φ Sector 0 is the first sector of the first track on the outermost cylinder.
  - Φ Mapping proceeds in order through that track, then the rest of the tracks in that cylinder, and then through the rest of the cylinders from outermost to innermost.

## DISK SCHEDULING

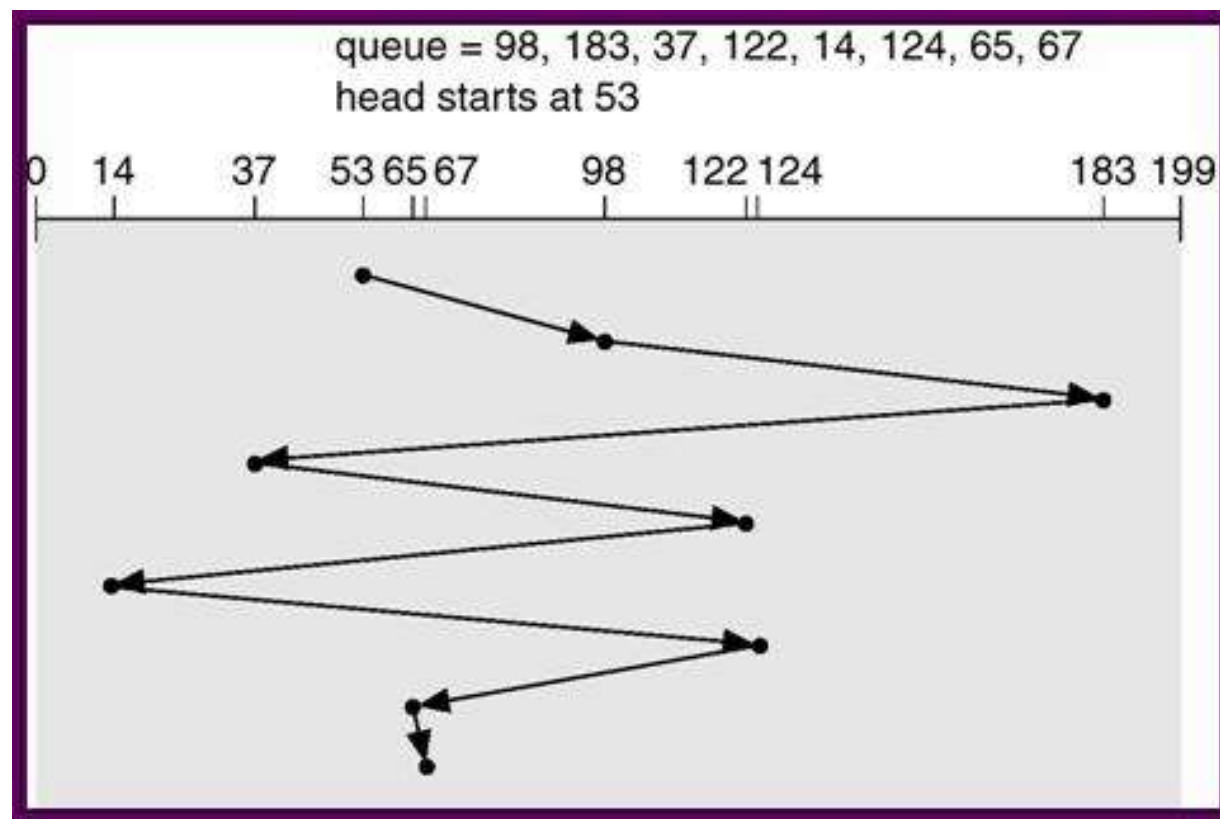
- ↳ The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth.
- ↳ Access time has two major components
  - Φ *Seek time* is the time for the disk are to move the heads to the cylinder containing the desired sector.
  - Φ *Rotational latency* is the additional time waiting for the disk to rotate the desired sector to the disk head.
- ↳ Minimize seek time
- ↳ Seek time  $\approx$  seek distance
- ↳ Disk bandwidth is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer.

## DISK SCHEDULING ALGORITHMS

Several algorithms exist to schedule the servicing of disk I/O requests.  
We illustrate them with a request queue (0-199).

### FCFS

total head movement: 640 cylinders

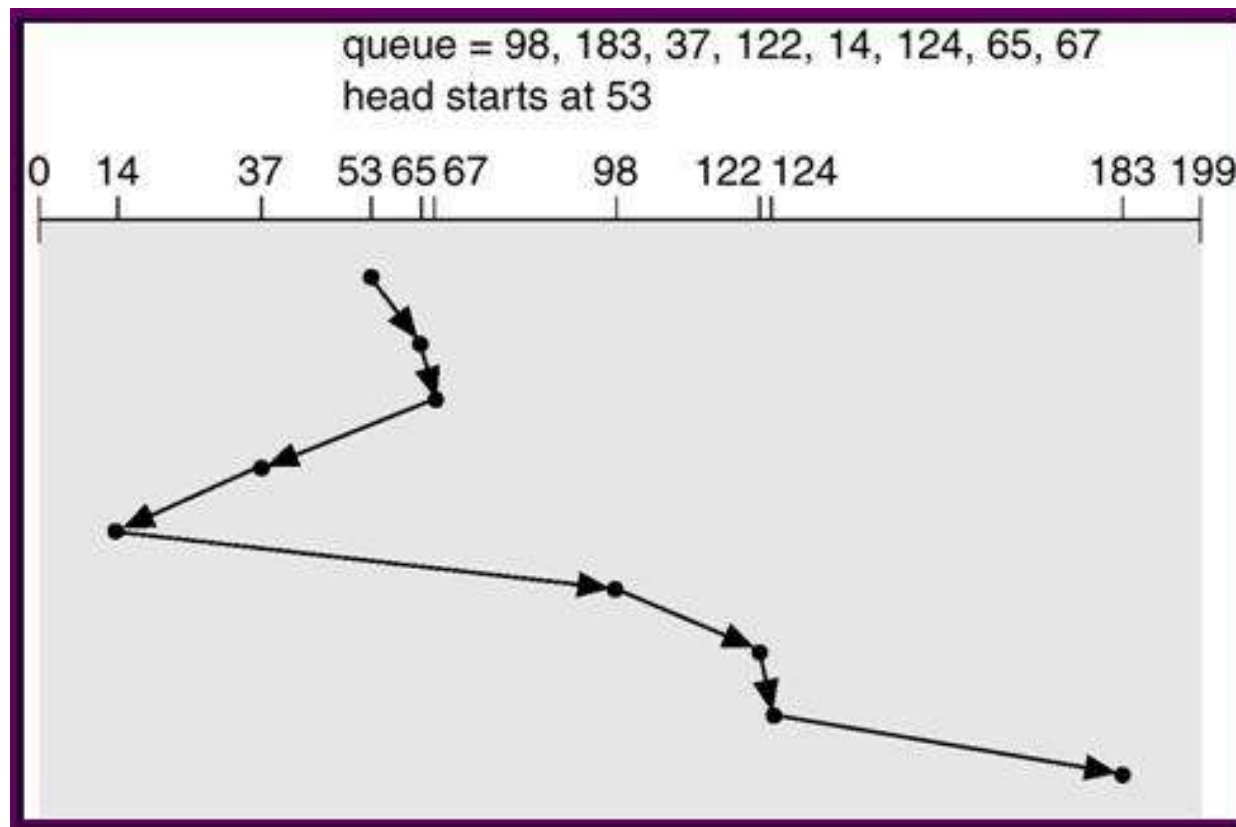


## DISK SCHEDULING ALGORITHMS

### SSTF

- Φ Selects the request with the minimum seek time from the current head position.
- Φ SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests.

total head movement: 236 cylinders

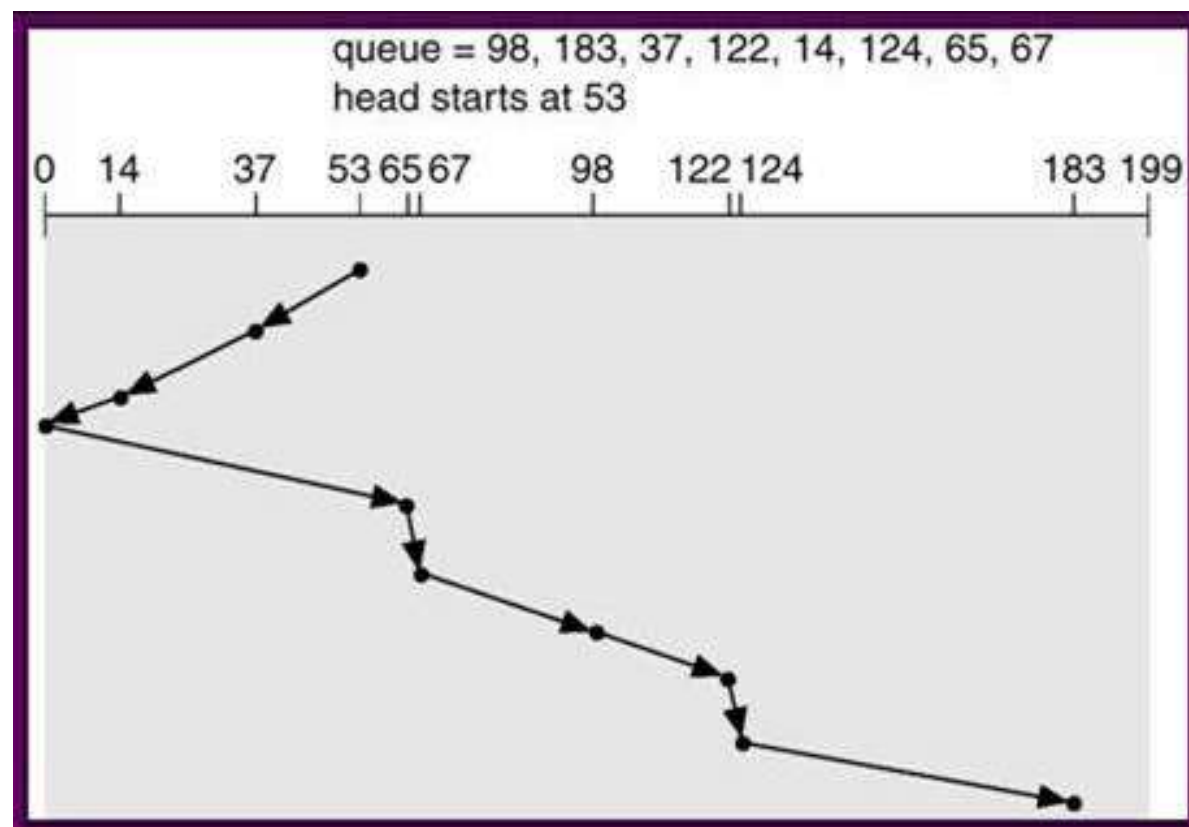


## DISK SCHEDULING ALGORITHMS

### SCAN

- Φ The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues.
- Φ Sometimes called the *elevator algorithm*.

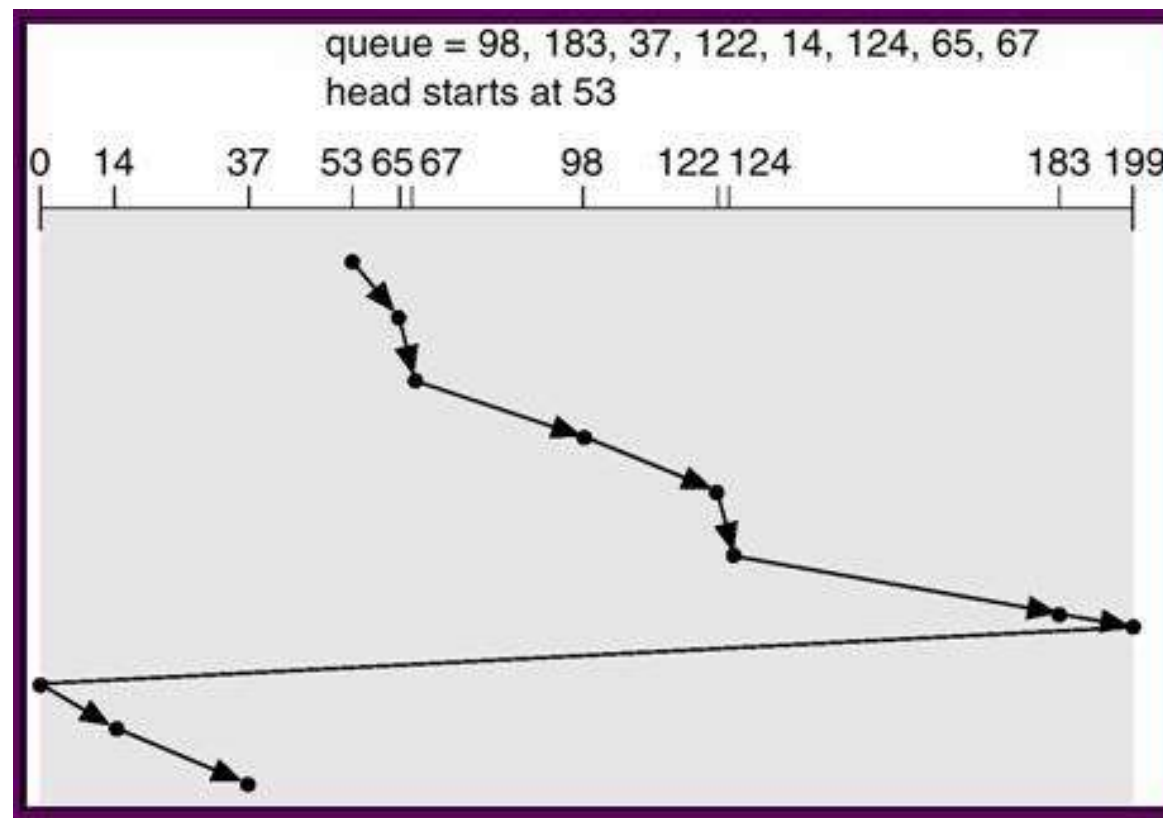
total head movement: 208 cylinders.



## DISK SCHEDULING ALGORITHMS

### C-SCAN

- Φ Provides a more uniform wait time than SCAN.
- Φ The head moves from one end of the disk to the other, servicing requests as it goes. When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip.
- Φ Treats the cylinders as a circular list that wraps around from the last cylinder to the first one.



## **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.**

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.

<i>Tipo record</i>	<i>Indirizzo primo record</i>	<i>Indirizzo ultimo record</i>
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, 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.