

Università di Ferrara
Laurea Triennale in Informatica
A.A. 2021-2022
Sistemi Operativi e Laboratorio

11. Gestione dell'Input/Output

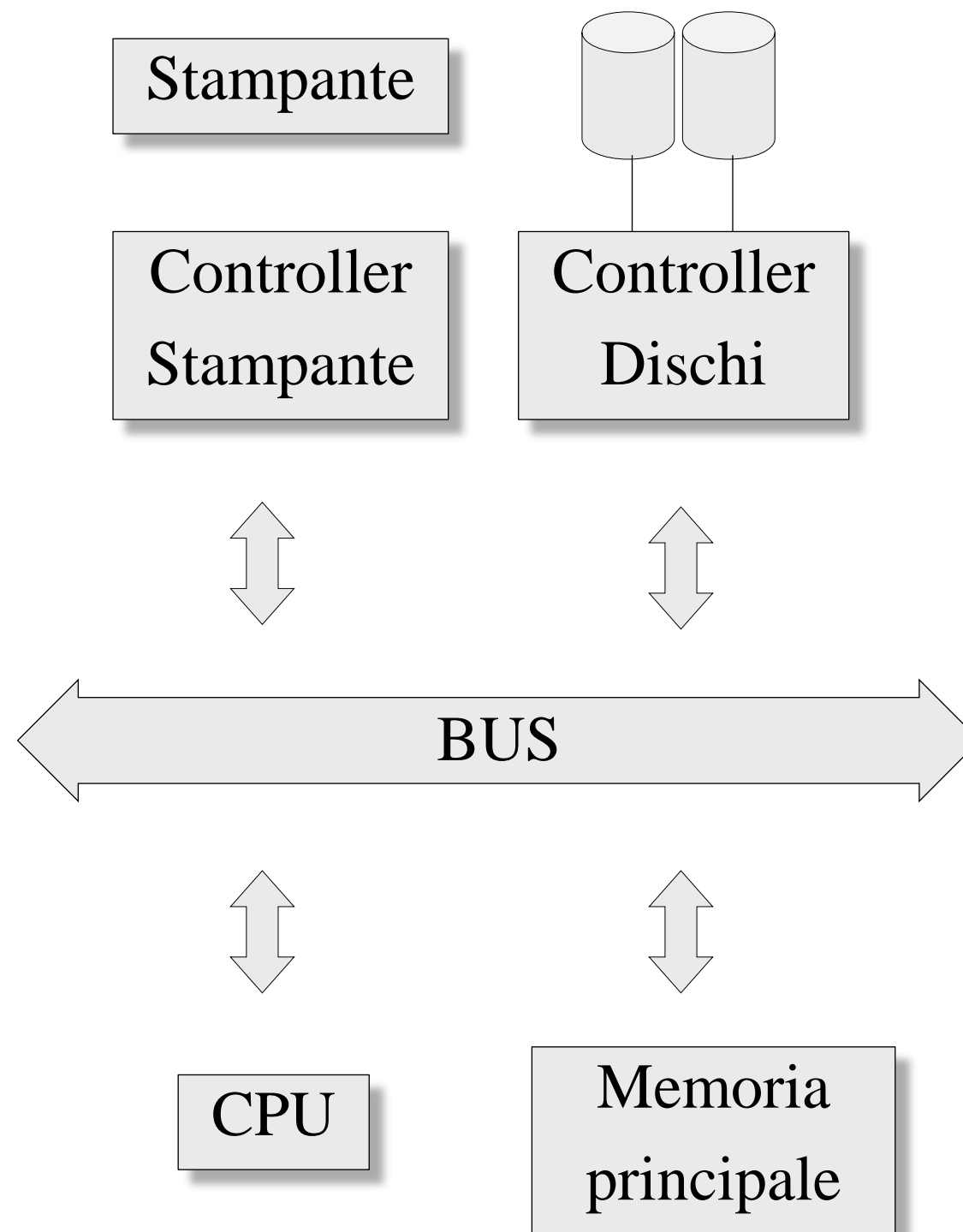
Prof. Carlo Giannelli

`http://www.unife.it/scienze/informatica/insegnamenti/
sistemi-operativi-laboratorio`

`http://docente.unife.it/carlo.giannelli`

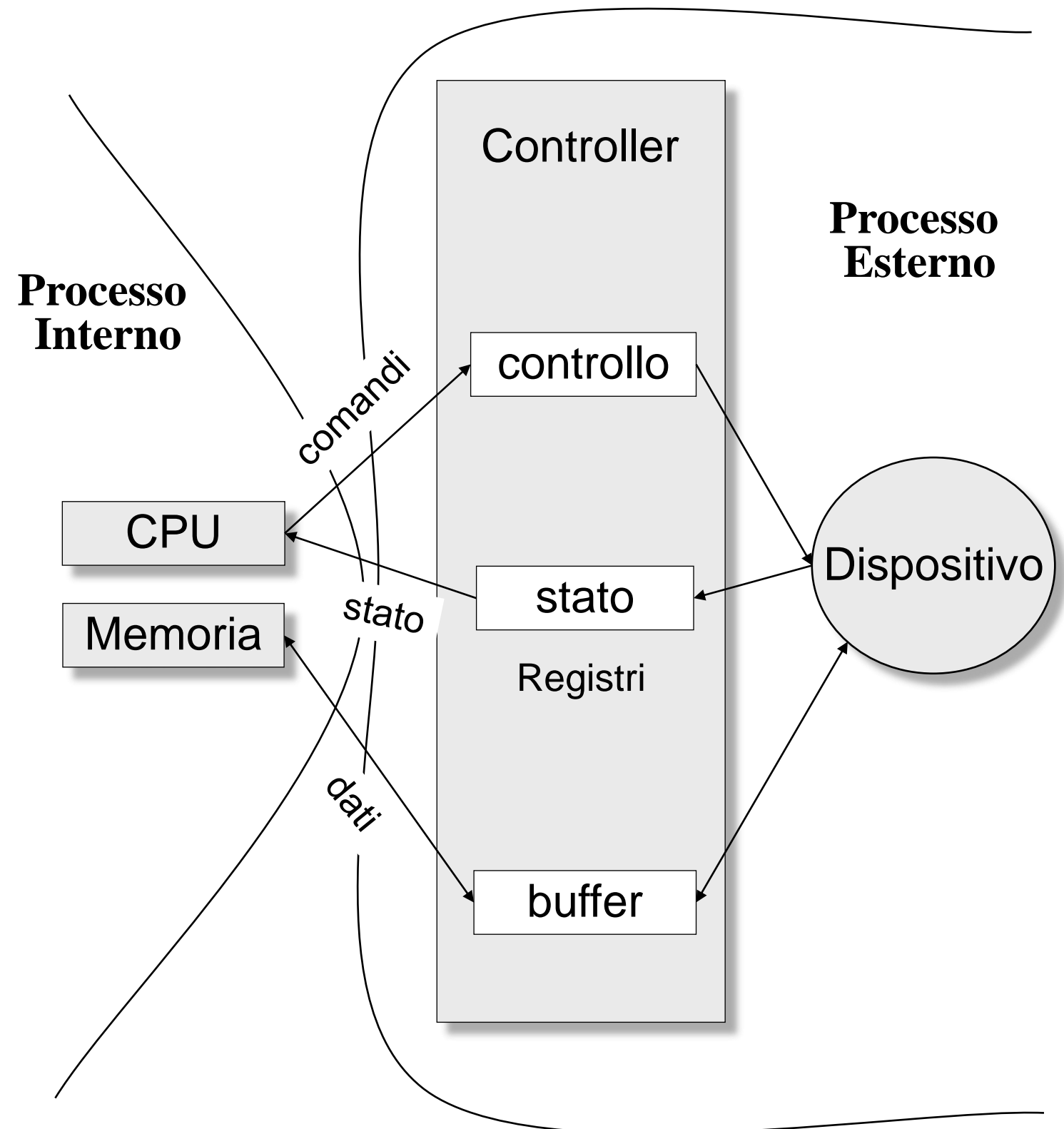
`https://ds.unife.it/people/carlo.giannelli`

Architettura hardware di un sistema di calcolo



Architettura hardware del sottosistema di I/O

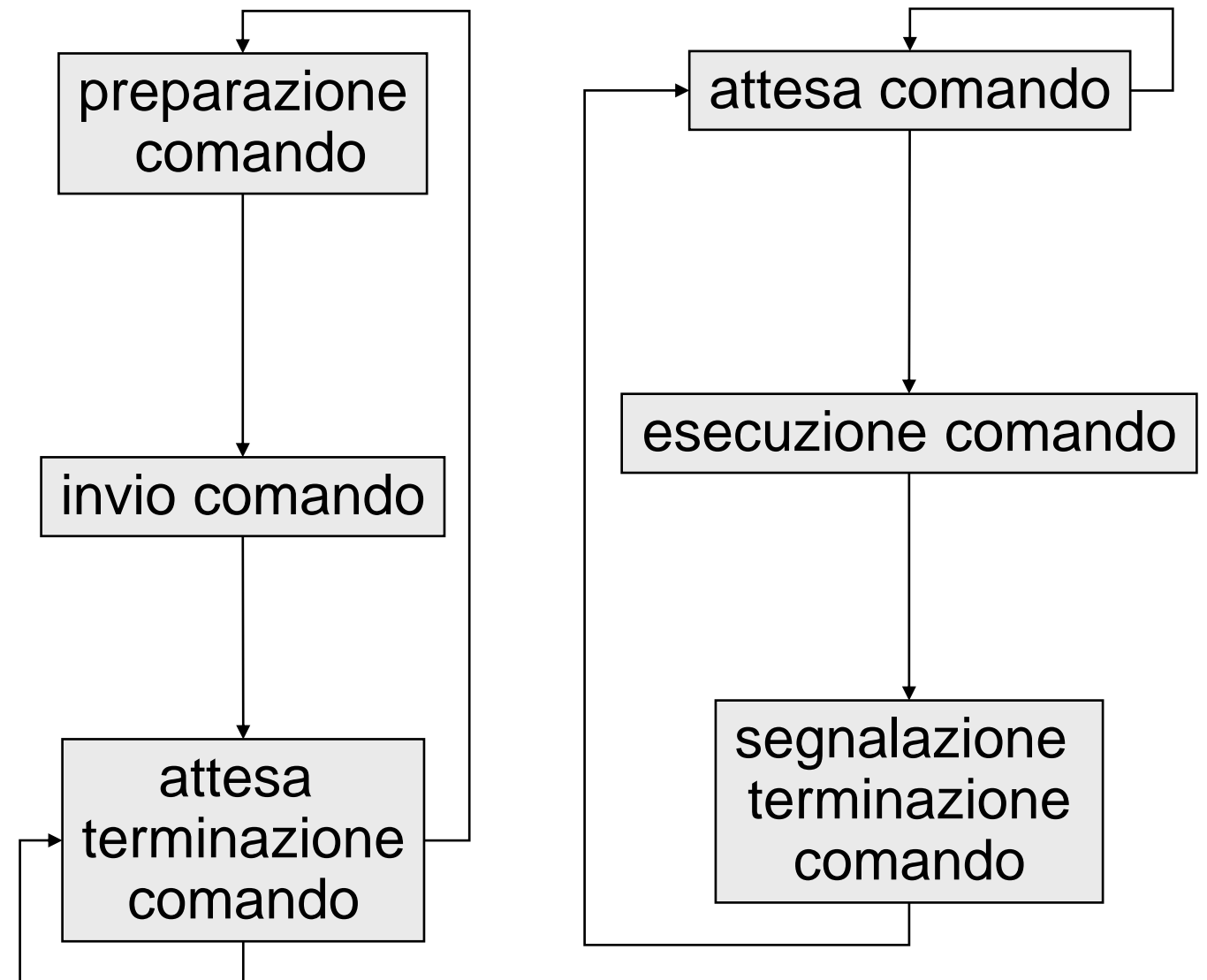
- La **CPU** legge e scrive i registri del controller mediante apposite istruzioni
- Il **dispositivo** invia e riceve le informazioni e i dati tramite i registri e il buffer del controller



Architettura software del sottosistema di I/O

Possibili modelli di funzionamento:

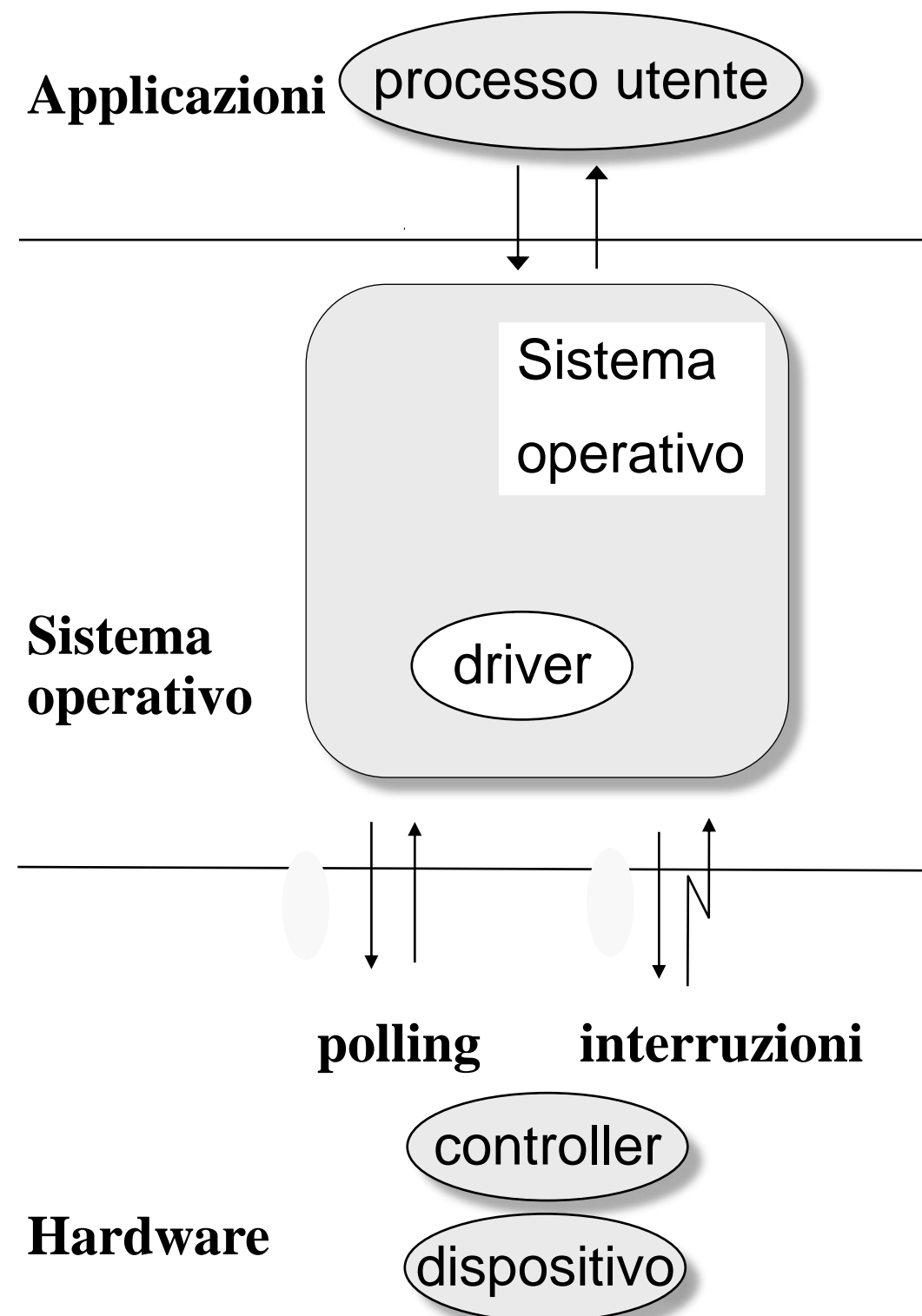
- funzionamento a **controllo di programma** (polling) (problema attesa attiva)
- gestione a **interruzioni**



**Processo
Interno**
(device driver del SO)

**Processo
Esterno**
(controller e
dispositivo)

Organizzazione logica per la gestione dei dispositivi

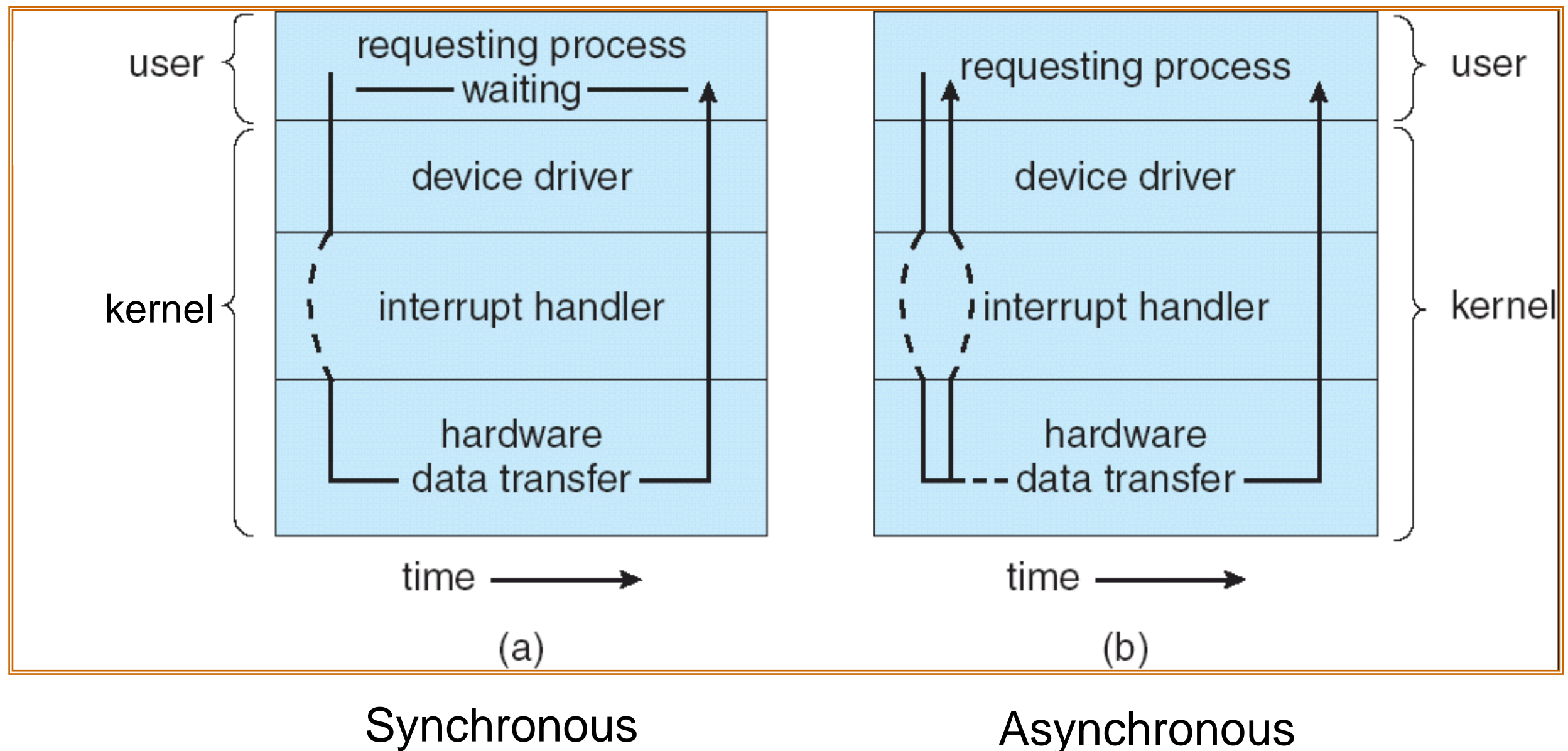


Input/Output guidato dalle interruzioni 1/2

Il punto di vista del **processo richiedente**.

- **Gestione sincrona:** ogni processo che inizia un'operazione di I/O viene **bloccato** in attesa che il sistema operativo porti a termine l'operazione di I/O richiesta.
- **Gestione asincrona:** al termine dell'operazione di I/O (per esempio lettura di un blocco di file da un disco) il controller del dispositivo lancia una **interruzione hardware** al sistema operativo che può quindi informare il processo richiedente.

Due metodi di I/O: sincrono vs. asincrono



Input/Output guidato dalle interruzioni 2/2

Il punto di vista del **sistema operativo**.

- **Polling**: la gestione a interruzione evita l'inefficienza delle attese attive presente del SO nella gestione dell'I/O eseguita a controllo di programma (polling).
- **Driver**: i driver sono la parte del sistema operativo che gestiscono i dispositivi. Compito del driver è di **inviare i comandi** appropriati ai dispositivi (al controller) e **gestire le interruzioni**. È la sola parte del sistema operativo che conosce i comandi dei controller, il numero dei registri, etc.

Polling vs. Interrupts

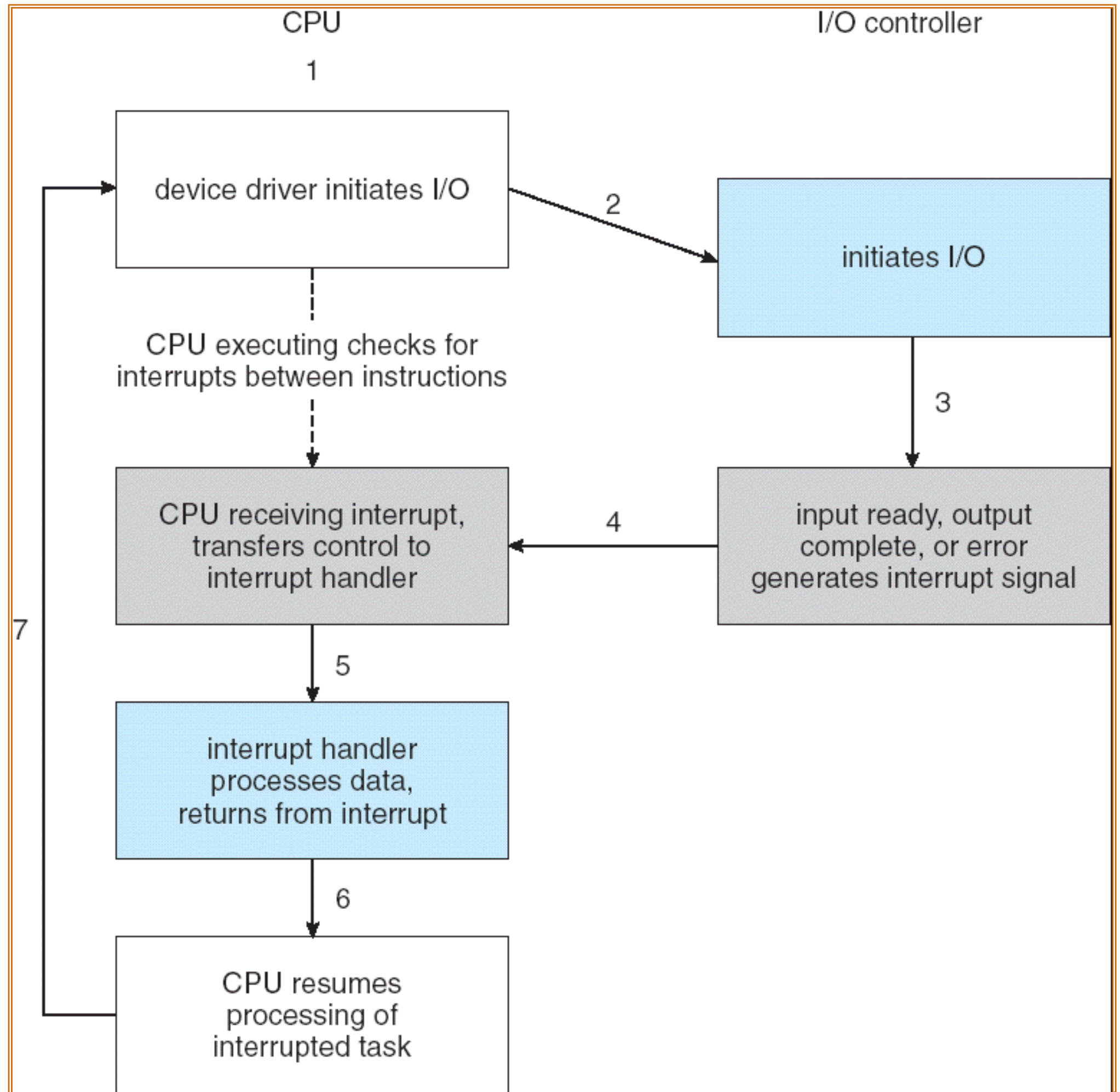
Polling

- Determinazione dello stato del device
 - command-ready
 - busy
 - error
- "Busy-wait cycle" per attendere I/O dal device

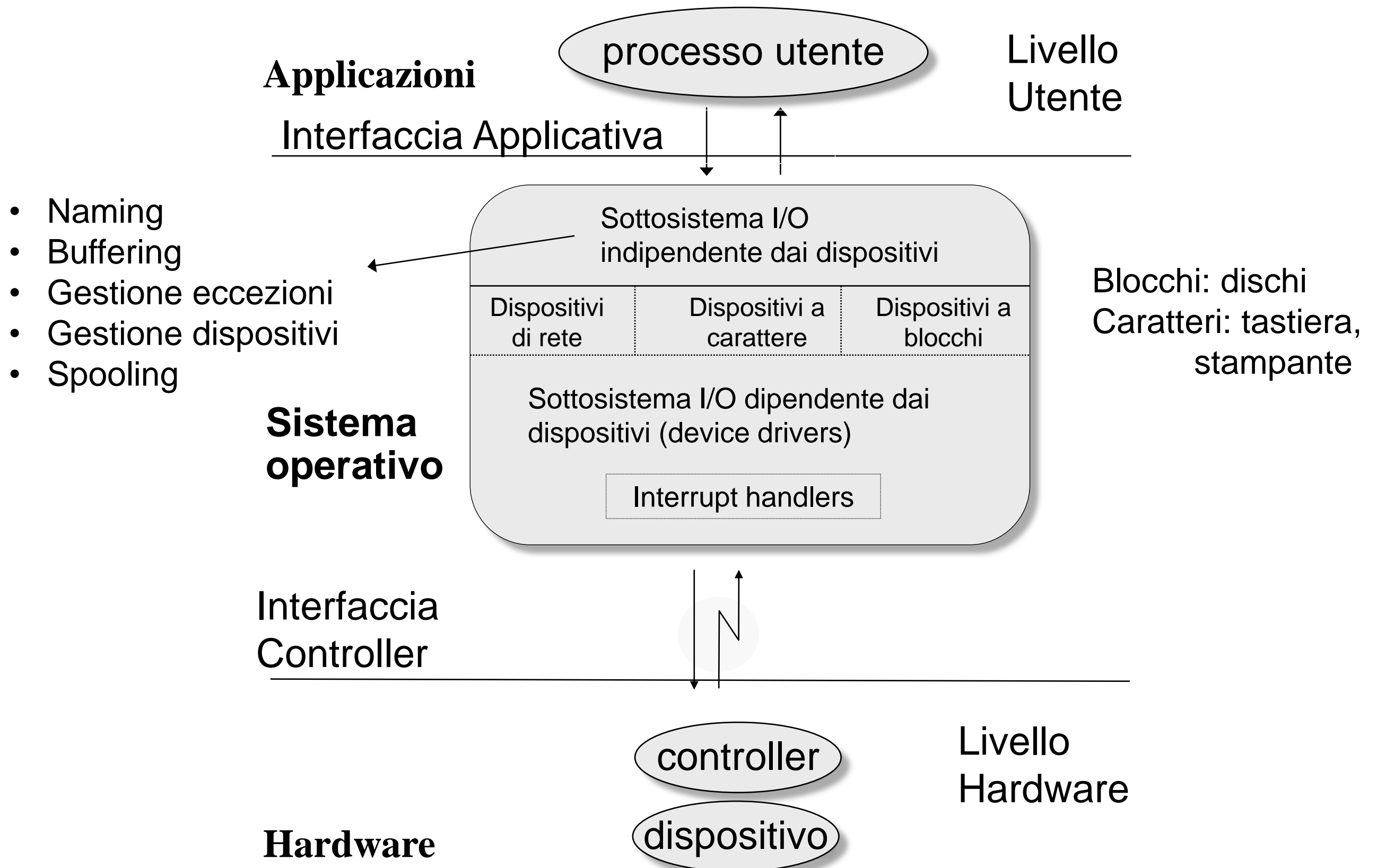
Interrupts

- CPU Interrupt-request generato da dispositivo I/O
- Interrupt handler riceve gli interrupt
- "Maskable" per ignorare o rimandare gli interrupt
- Interrupt vector per distribuire gli interrupt al giusto handler
 - basato su priorità
 - alcuni interrupt "nonmaskable"
- Meccanismo di interrupt usato anche per le eccezioni

Interrupt-Driven I/O Cycle



Organizzazione logica per la gestione dei dispositivi



Organizzazione logica per la gestione dei dispositivi

Naming. Ogni dispositivo è identificato univocamente. In Unix ogni dispositivo ha un nome simbolico all'interno dello spazio dei nomi del file system (si veda la directory /dev).

Buffering. Aree buffer che ospitano i dati nel trasferimento tra i dispositivi e le aree di memoria dei processi applicativi. Servono per:

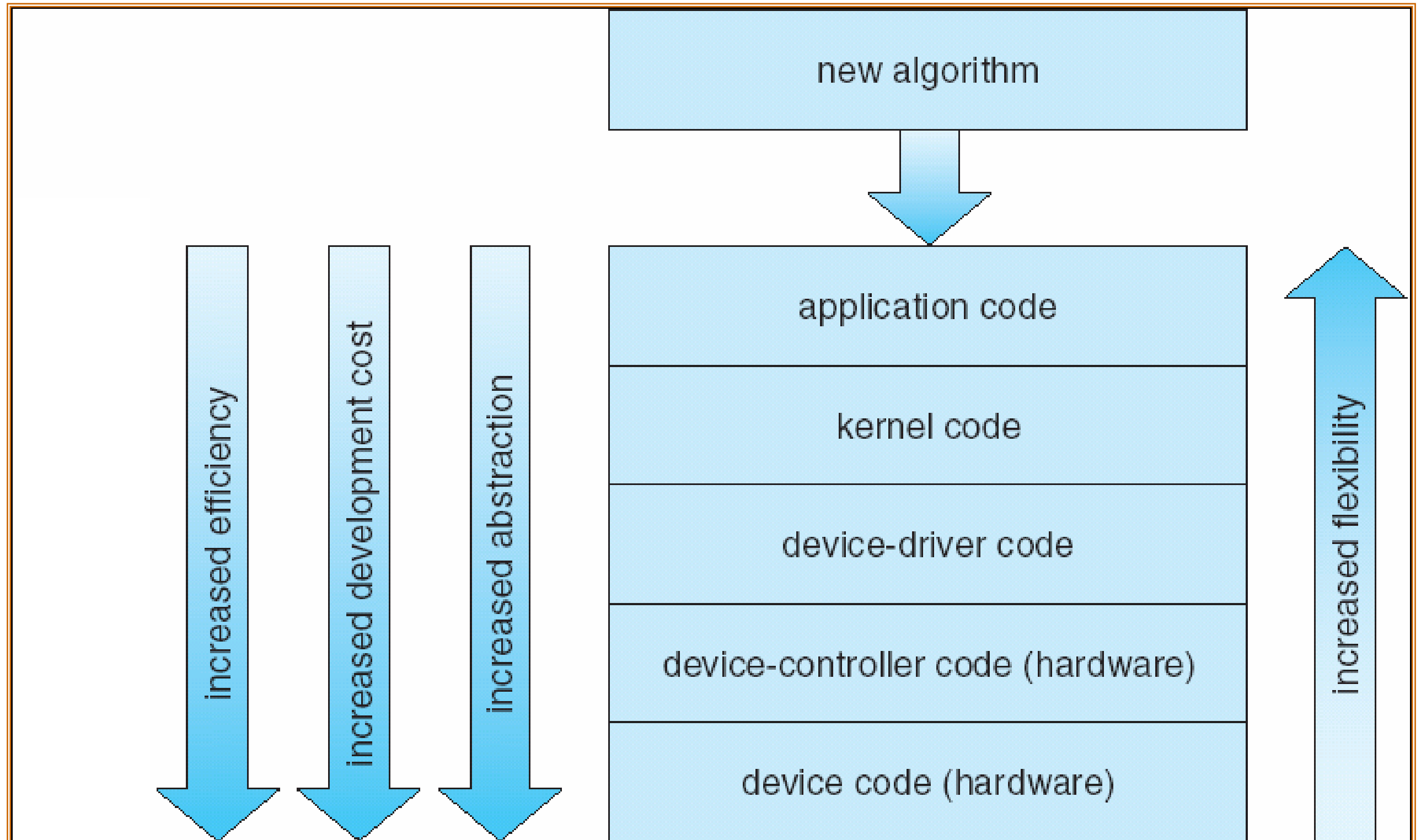
- 1) mediare tra diverse velocità di produzione/consumo tra processi e dispositivi,
- 2) trasferire efficacemente dei blocchi dati,
- 3) parallelizzare le operazioni di accesso a I/O.

Gestione eccezioni. Nelle operazioni di I/O si possono verificare molti eventi anomali, che possono essere:

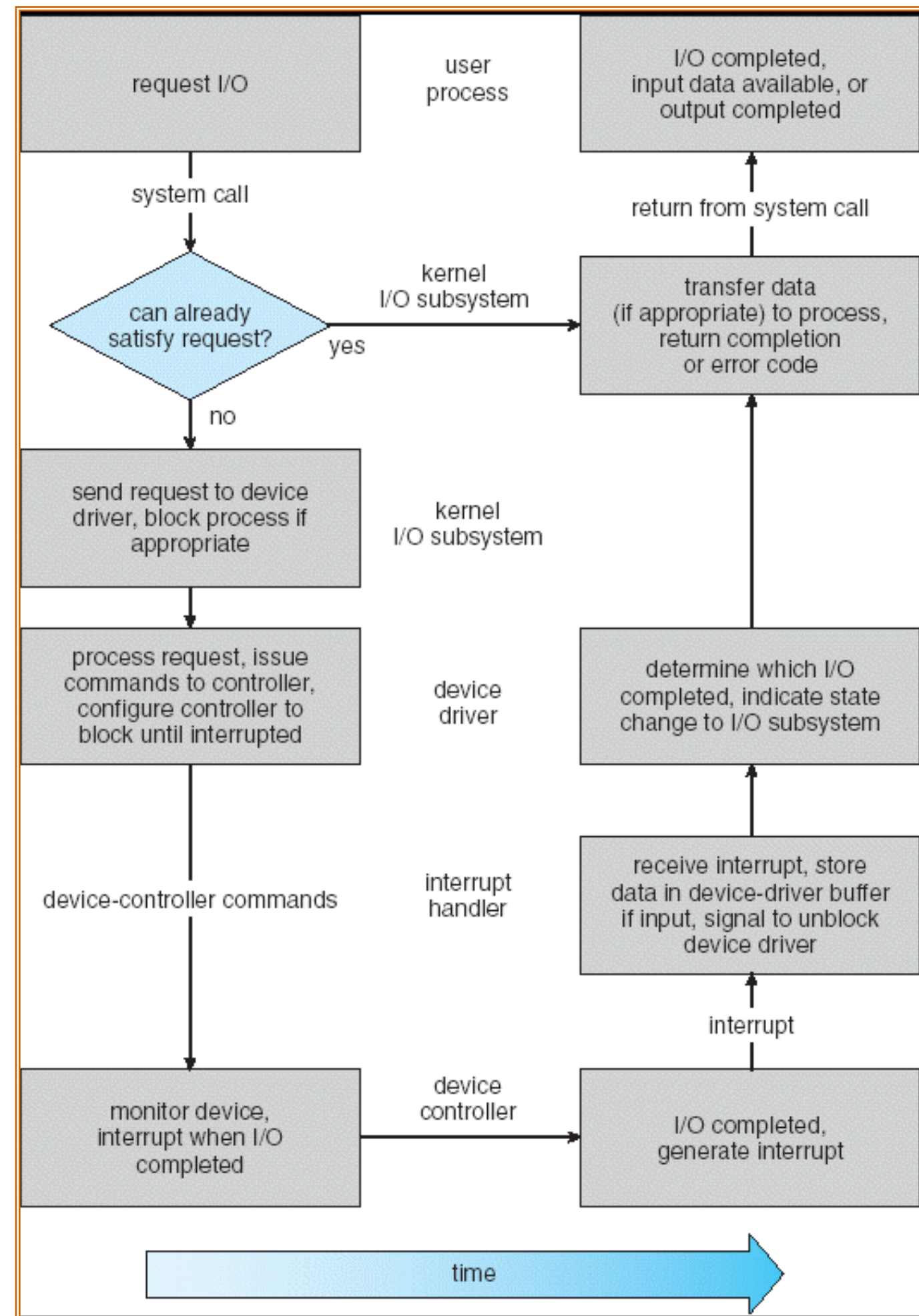
- mascherati e nascosti agli utenti (il sistema prova a completare le operazioni fallite)
- comunicati e propagati a processi e utenti.

Spooling. Tecnica di gestione per risorse condivise (un processo gestore per ogni risorsa).

Device-Functionality Progression

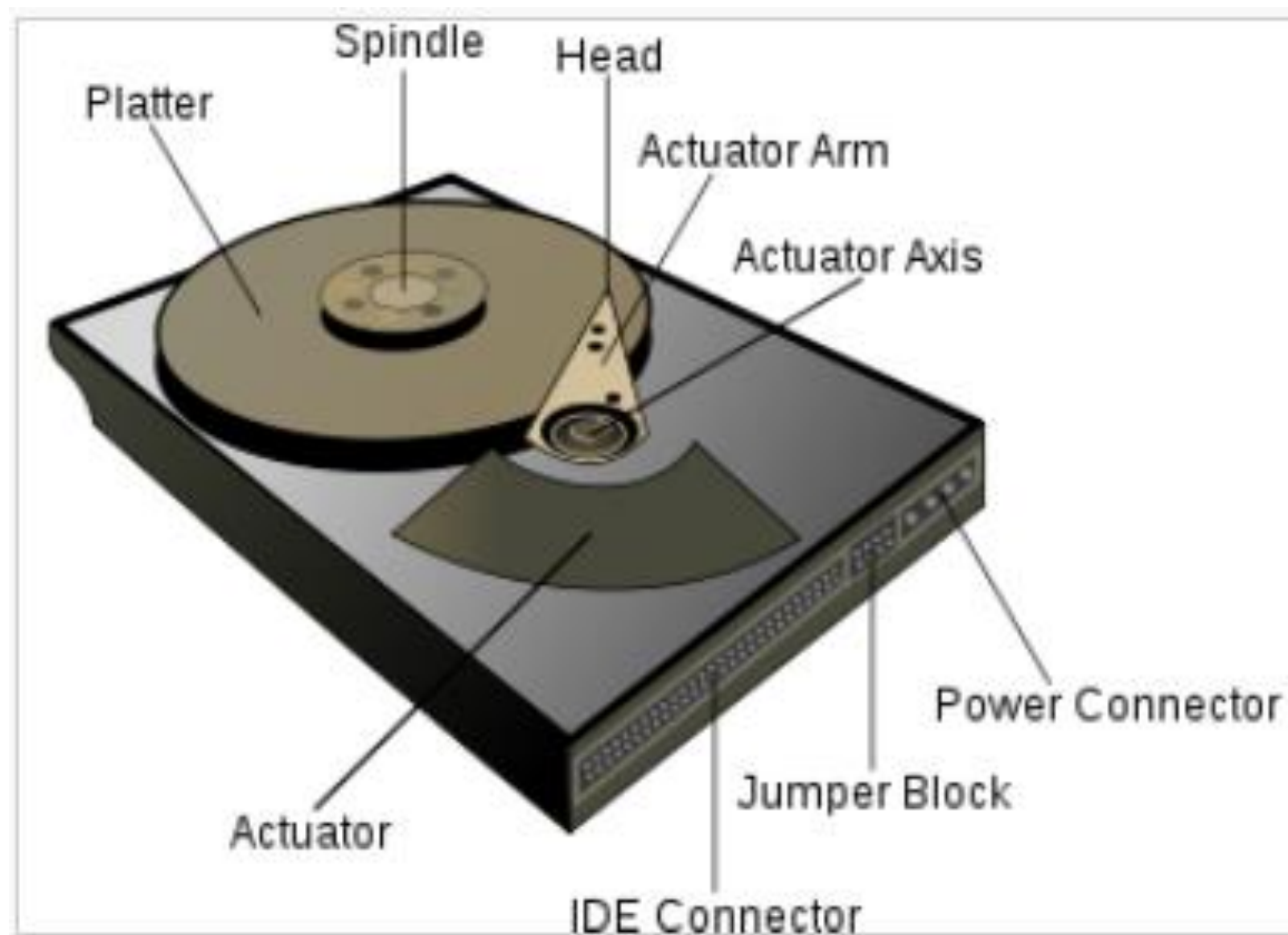


Life Cycle of an I/O Request

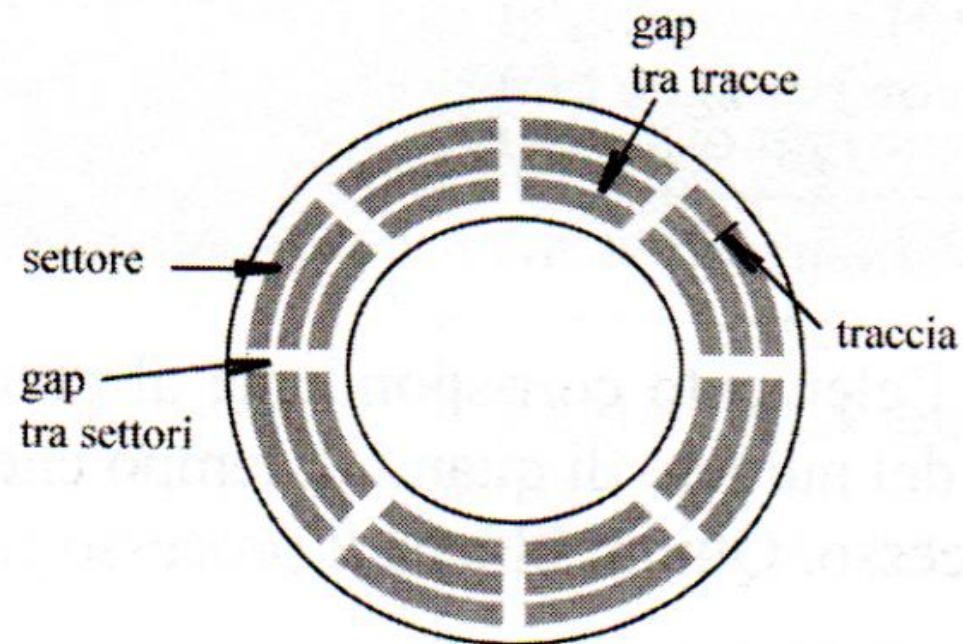


Gestione degli Hard Disk

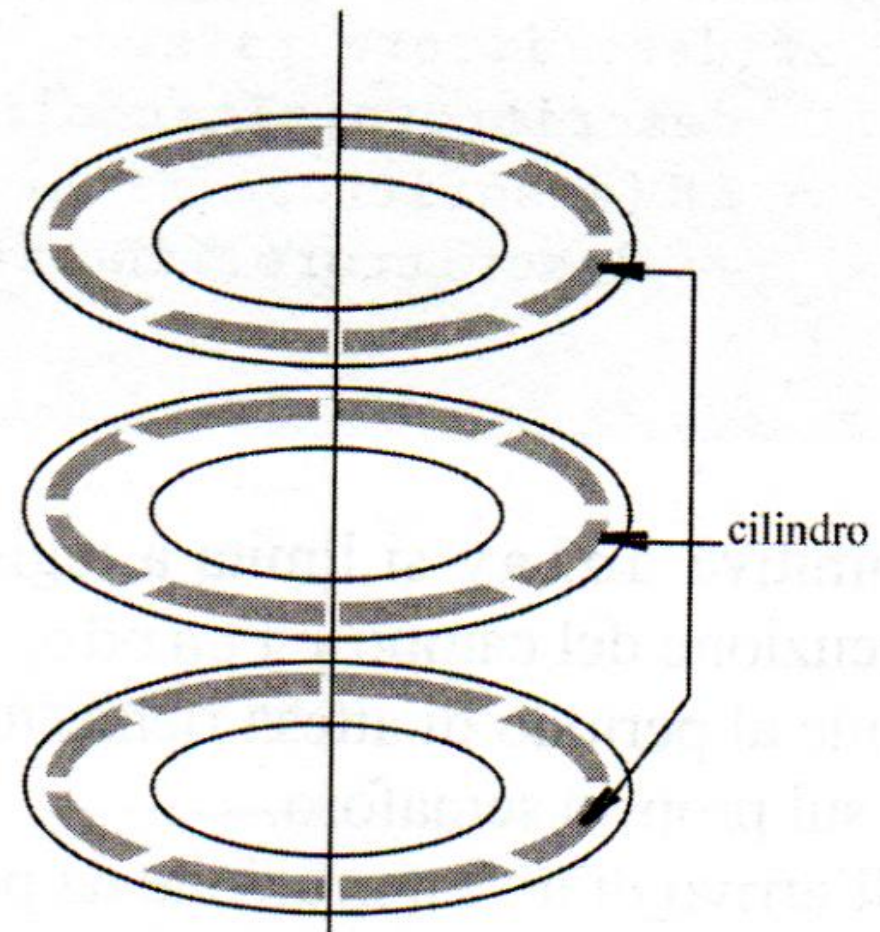
Gli Hard Disk sono dispositivi particolarmente importanti perché offrono uno spazio di memoria di massa, utilizzato per il file system ma **anche per la memoria virtuale**.



Organizzazione fisica dei dischi



a) disco singolo



b) disk pack

Il settore è l'unità minima di allocazione e di trasferimento (ordine di grandezza dei KB)

Un settore è identificato da:

- N. della faccia del disco
- N. della traccia (o cilindro)
- N. del settore dentro la traccia

Prestazioni Hard Disk

Le **prestazioni** di un Hard Disk sono valutate in termini di **tempo medio di trasferimento**:

$$TF = TA + TT$$

TF: Tempo medio di trasferimento

TA: Tempo medio di accesso (per posizionare testina)

TT: Tempo medio di trasferimento dati (per trasferire dati)

$$TA = ST + RL$$

ST: Seek Time, tempo per spostare longitudinalmente la testina del disco sulla traccia richiesta

RL: Rotational Time, tempo necessario per ruotare il disco in modo da leggere il settore richiesto. Prestazioni dischi espresse in giri al minuto, tra 5.400 e 15.000.

TT ordine microsecondi, ST e RL ordine millisecondi.

Per ridurre tempi di accesso ai dati, progettare strategie, **politiche**, per:

- allocazione dei file (in settori se possibile contigui)
- schedulare le richieste di accesso ai dischi (per minimizzare tempi spostamento testina)

Politiche scheduling accesso Hard Disk

In un sistema concorrente, molti processi accedono al file system, che si trova quindi a gestire molte richieste, che devono essere schedulate (adottando specifiche **politiche**) opportunamente per ridurre i tempi di attesa dei processi.

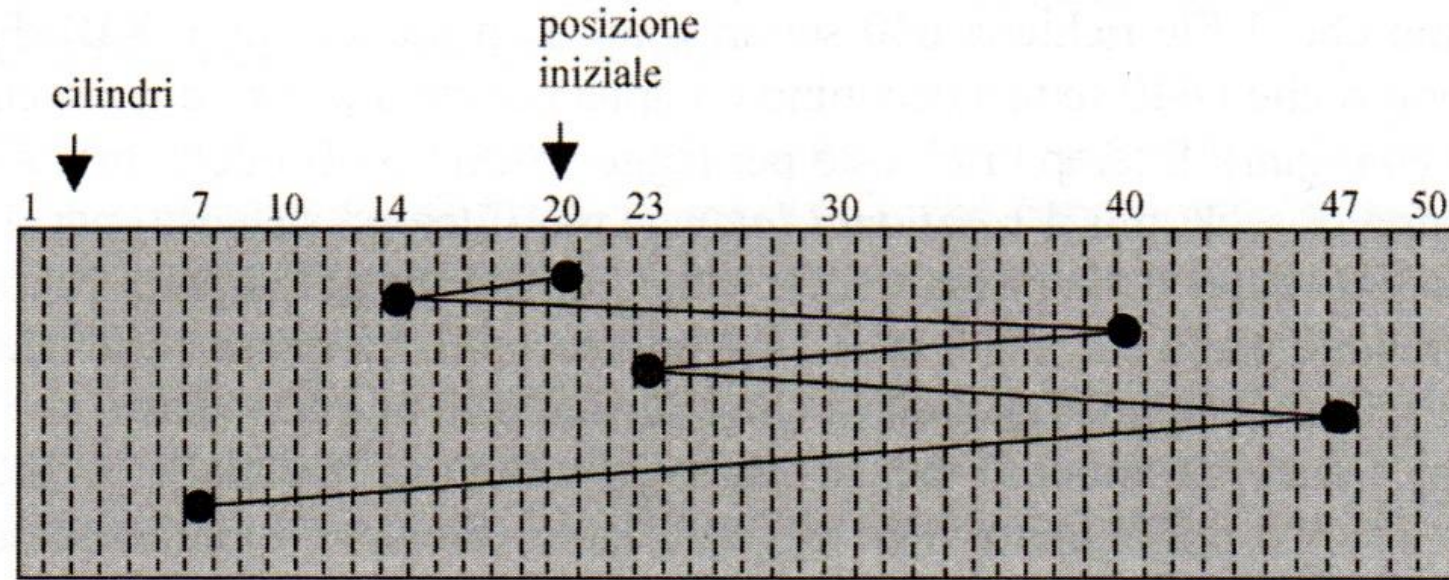
Esempio: ipotizziamo che la testina sia sulla traccia 20 e che siano in coda le richieste di operare sul disco sulle tracce 14, 40, 23, 47, 7.

Politiche scheduling accesso Hard Disk

20, 14, 40, 23, 47, 7

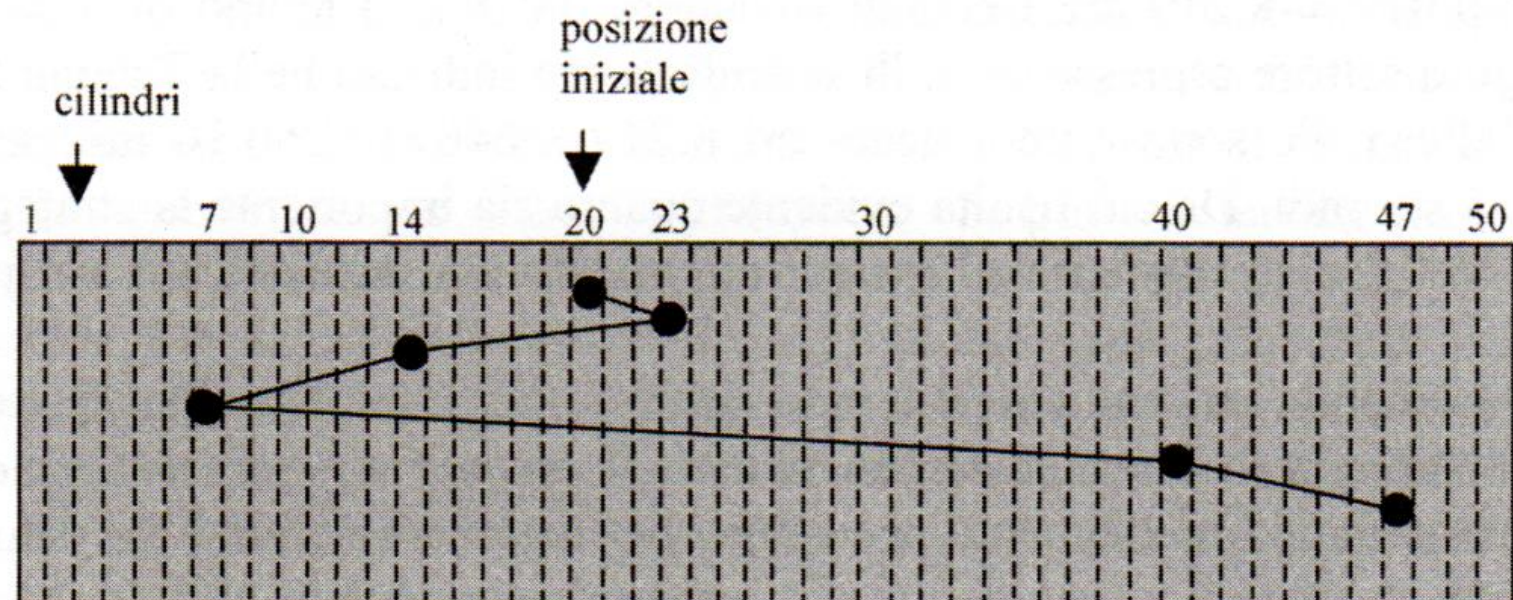
FCFS

First Come First Served



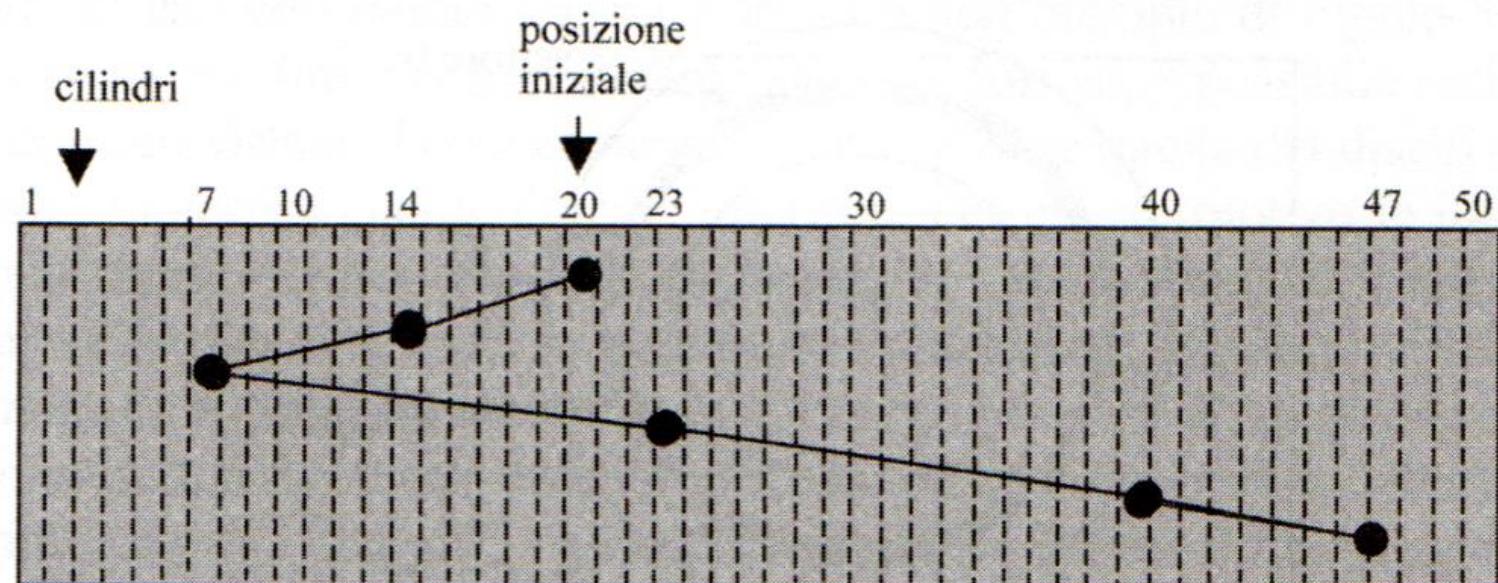
SSTF

Shortest Seek Time First
(possibile starvation)



SCAN

Si sposta dal primo cilindro all'ultimo e viceversa



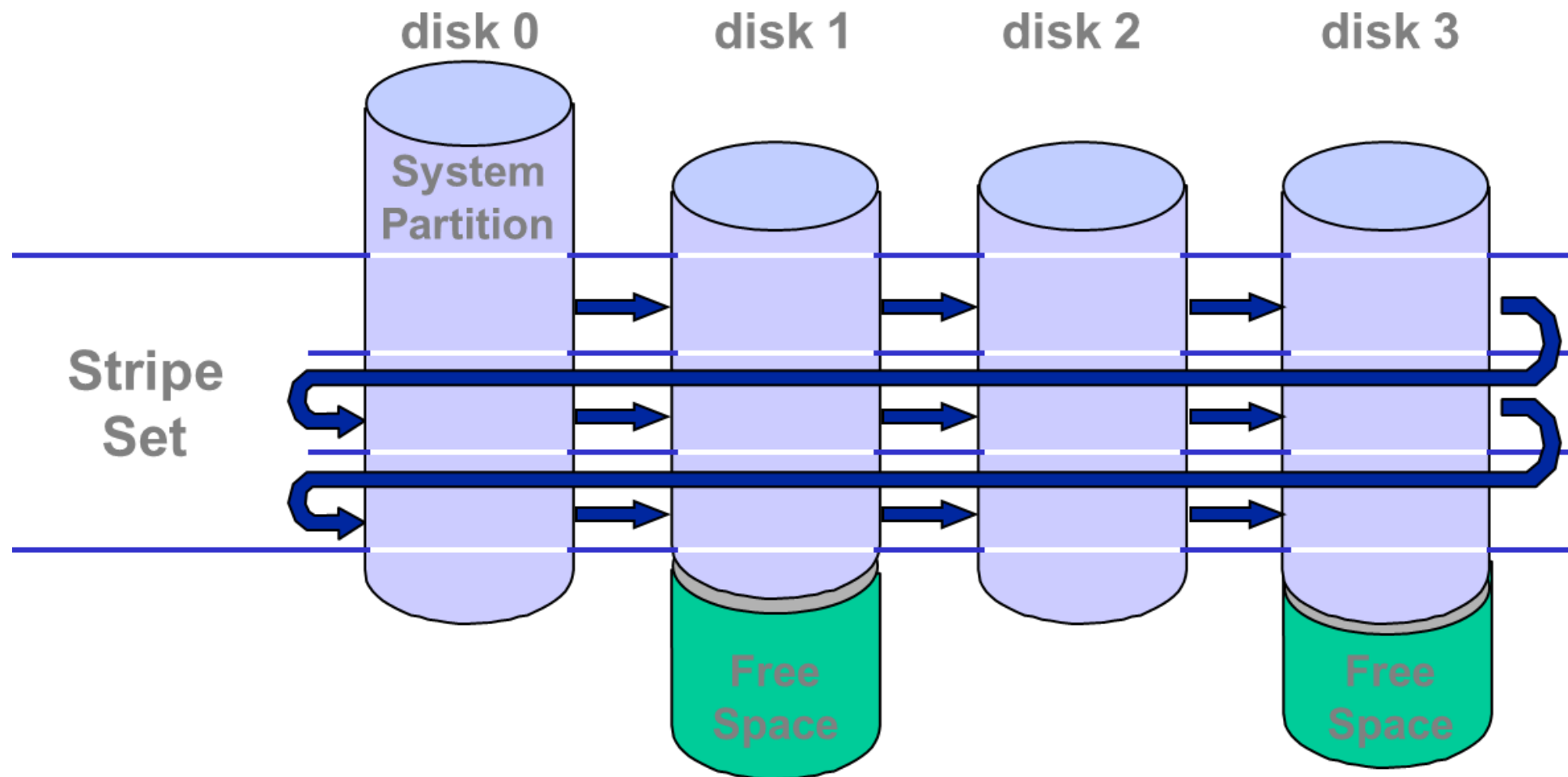
Dischi RAID

Per migliorare ulteriormente le **prestazioni**, si possono utilizzare in parallelo più dischi fissi. Questo può permettere anche di migliorare l'**affidabilità** e la **tolleranza ai guasti** (tramite ridondanza dei dati).

Sistemi **RAID** (Redundant Array of Independent Disks).

RAID 0	Striping
RAID 1	mirroring
RAID 2	Disk striping with error-correction code (ECC)
RAID 3	Disk striping with ECC stored as parity
RAID 4	Disk striping large blocks; parity stored on one drive
RAID 5	Disk striping with parity across multiple drives

RAID livello 0 (striping)



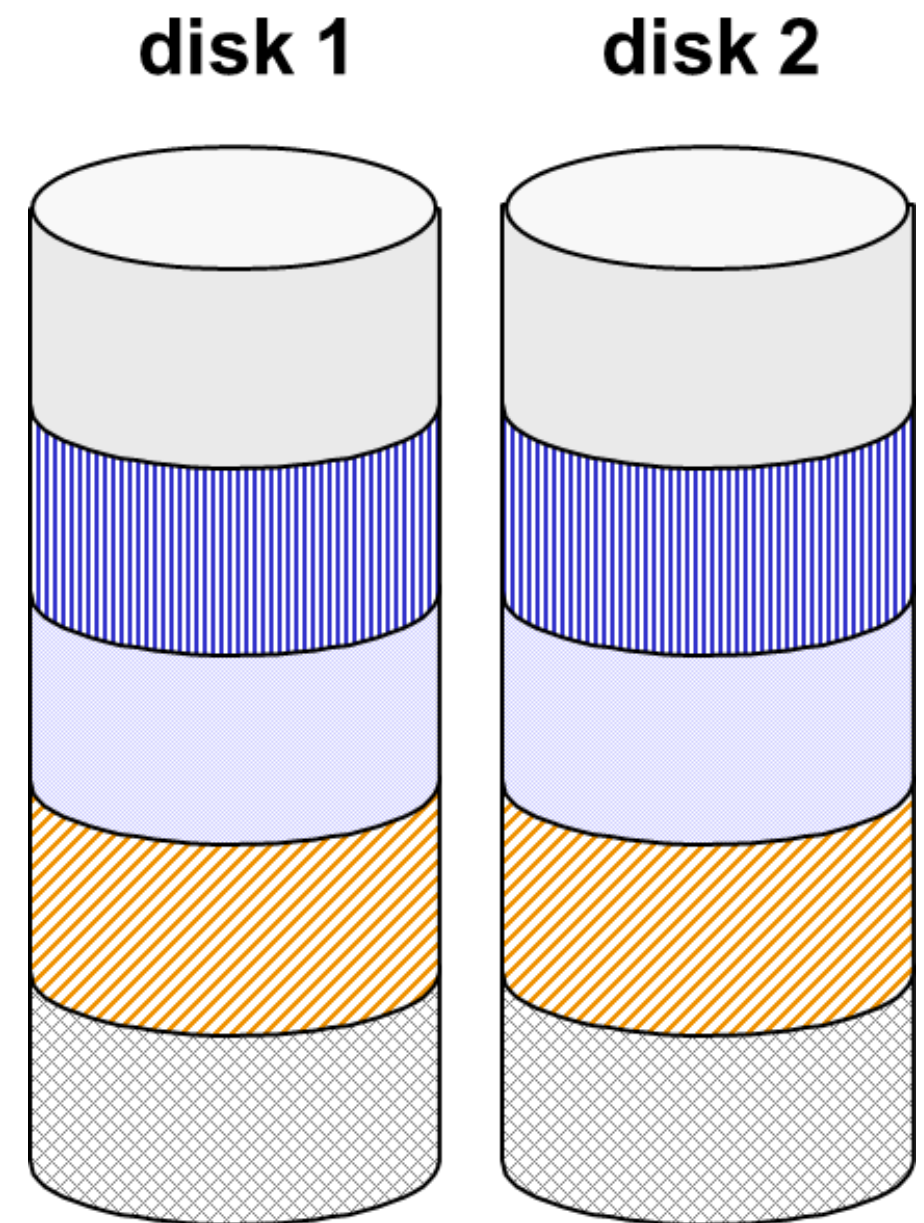
Si crea **un solo volume** logico su tutti i dischi.

I dati sono allocati su dischi diversi, per **parallelizzare** operazioni di I/O.

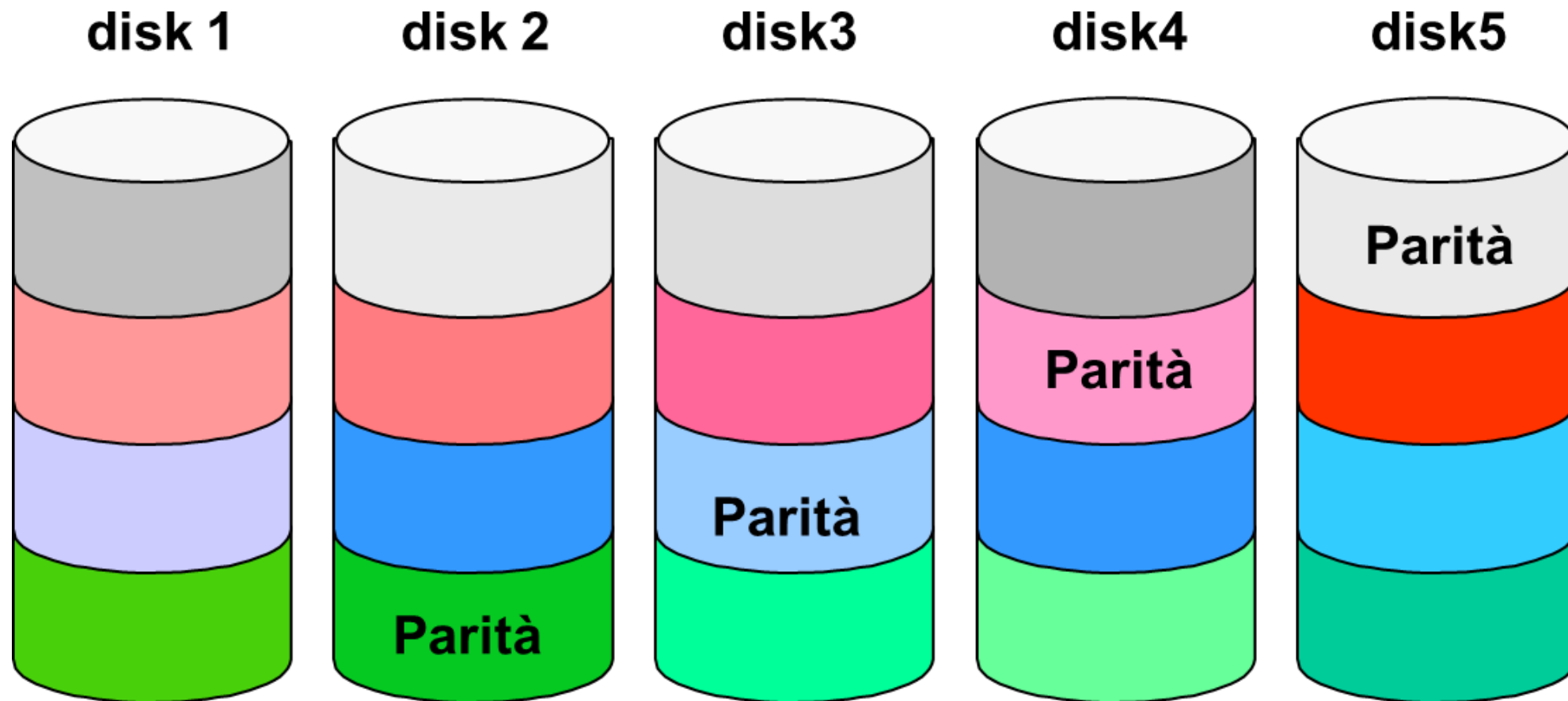
RAID livello 1 (mirroring)

Tutti i dati sono **replicati sui due dischi**. Il sistema scrive un dato sempre su due dischi.

- Lettura può essere parallelizzata sui due dischi
- Possibile mirroring anche aree sistema
- Tolleranza al guasto di un disco
- Elevato **costo** (utilizzo dischi del 50%).



RAID livello 5 (striping con parità)



- Ogni sezione di parità contiene l'**XOR (or-esclusivo)** delle 4 sezioni dati corrispondenti.
- Nel caso di perdita di **UNA** delle sezioni dati, il sistema ricostruisce la perdita utilizzando la sezione di parità.
- Minore costo rispetto a mirroring (in questo esempio, costo del 20%).
- Ogni scrittura richiede modifica sezione di parità.