



# Sistemi Operativi 2

## Introduzione

Il corso andrà a trattare i seguenti macro-argomenti:

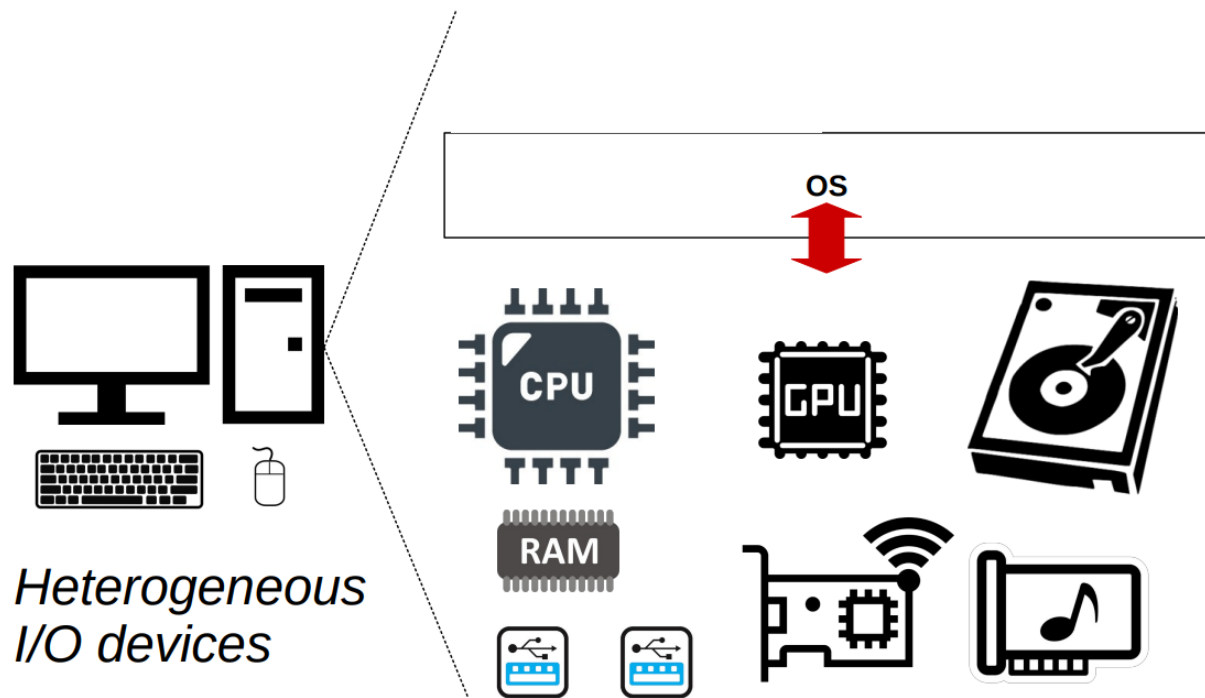
- I/O Management
  - I/O Software e Hardware
  - Dischi, SSD e altri dispositivi
- File System
  - File e directories
  - Implementazione del file system

## Sottosistema I/O

Nei sistemi operativi ci sono componenti hardware che sono specifiche per l'input o output di informazioni. Prendiamo in caso questa lista:

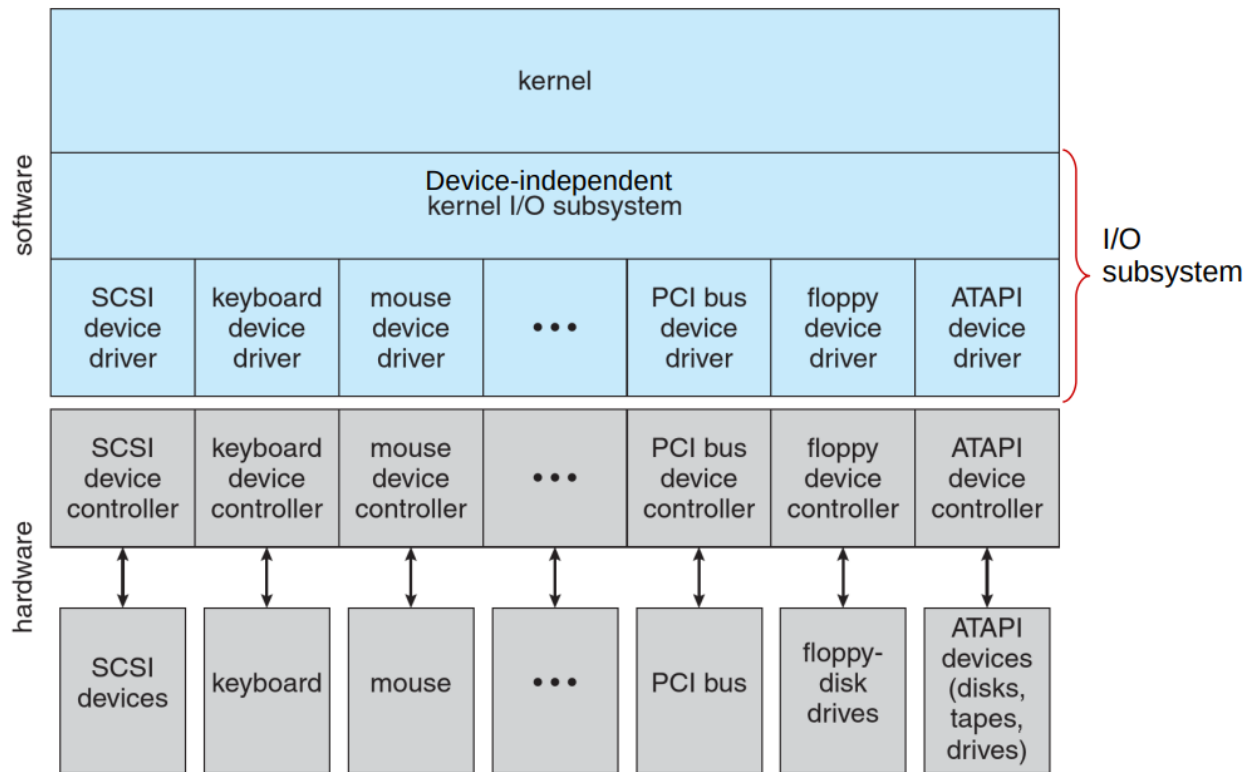
- Tastiera (**I**)
- Cassa (**O**)
- Monitor (**O**)
- Microfono (**I**)
- Network card interface (**B**)
- Flash cards (**B**)

Il sistema operativo gestisce input e output di questi dispositivi ma anche: **cattura errori** e li **gestisce**.



Un "problema" è che il SO deve gestire molti dispositivi eterogenei tra loro (es. velocità diverse), troveremo una soluzione.

Il **sottosistema I/O** è una parte del SO che separa SO con le complessità della gestione dei dispositivi I/O. Provvede un'interfaccia uniforme ai livelli sovrastanti per accedere ai dispositivi I/O. In oltre deve essere in grado di comunicare con vari dispositivi I/O e questo avviene grazie a gestori dedicati chiamati **drivers**.



## Funzionamento di un sottosistema I/O

### Catena di assemblaggio

- Protocolli:
  - Come/Quali parti entrano
  - Come/Quali parti escono
- Gestori:
  - Lo staff
- Divisione dei compiti
  - I compiti sono divisi

### Sottosistema I/O

- Protocolli:
  - Interfacce con dispositivi I/O
- Gestori:
  - Drivers
- Divisione dei compiti
  - Astrazione dei dispositivi I/O dalle applicazioni

1. Il sottosistema I/O lavora con dispositivi che operano a **blocchi di dimensione prefissata**, un blocco è la **minor parte indirizzabile** nel sistema, i **trasferimenti** avvengono su **unità di blocchi**. Un blocco può essere **scritto o letto** (accesso randomico perchè si può leggere e scrivere).

2. Il sottosistema I/O lavora con dispositivi che operano a **caratteri** (o **flusso di caratteri**) che trattano i dati come flussi di byte **senza tener conto di blocchi**, **non sono indirizzabili** e **non** si può **effettuare la ricerca** (accesso **seriale**) le operazioni più tipiche sono **inserimento** o **presa** di **caratteri**.

Vediamo come funzionano i vari dispositivi:

- Tastiera (**C**) → viene letta come stream di caratteri
- Hard disk (**B**) → viene letta come insieme di blocchi
- Mouse (**C**)
- Dispositivi USB (**B**)
- Network card interface (?) → stream di blocchi di diverse dimensioni

La differenza tra **blocchi** e **flusso** di caratteri è che il **primo quando leggiamo** un blocco, **possiamo accedervi nuovamente** (negli hard disk quando leggiamo un blocco esso non scompare ma rimane lì fin tanto che non lo eliminiamo), nel **secondo quando leggiamo** un dato esso **non è più reperibile nuovamente** (quando facciamo il click di un mouse non possiamo rileggerlo perchè è ormai consumato).

## Dispositivi a blocchi

```
FILE *fp = NULL;
int ret = 0;
...
fp = fopen("/dev/sda", "rb");
ret = fseek(fp, 0L, SEEK_END); // <- non avremo problemi a fare
if (ret == -1) perror("Cannot seek device");
fclose(fp);
```

## Dispositivi a caratteri

```
FILE *fp = NULL;
int ret = 0;
...
```

```
fp = fopen("/dev/tty", "rb");
ret = fseek(fp, 0L, SEEK_END); // <- le operazioni di seek non s
if (ret == -1) perror("Cannot seek device");
fclose(fp);
```

Come abbiamo visto con le **NIC** (Network Interface Card) non si riesce a classificare, è una **imperfezione** dire che è o uno o l'altra. Ma per molti dei dispositivi va bene come classificazione, e rendere indipendenti le parti del SO che dialogano con questi dispositivi I/O come il file system.

## Dispositivi I/O

Un dispositivo I/O è composto da due parti:

- Il **dispositivo** (parte meccanica)
- Il **controller** (parte elettronica)

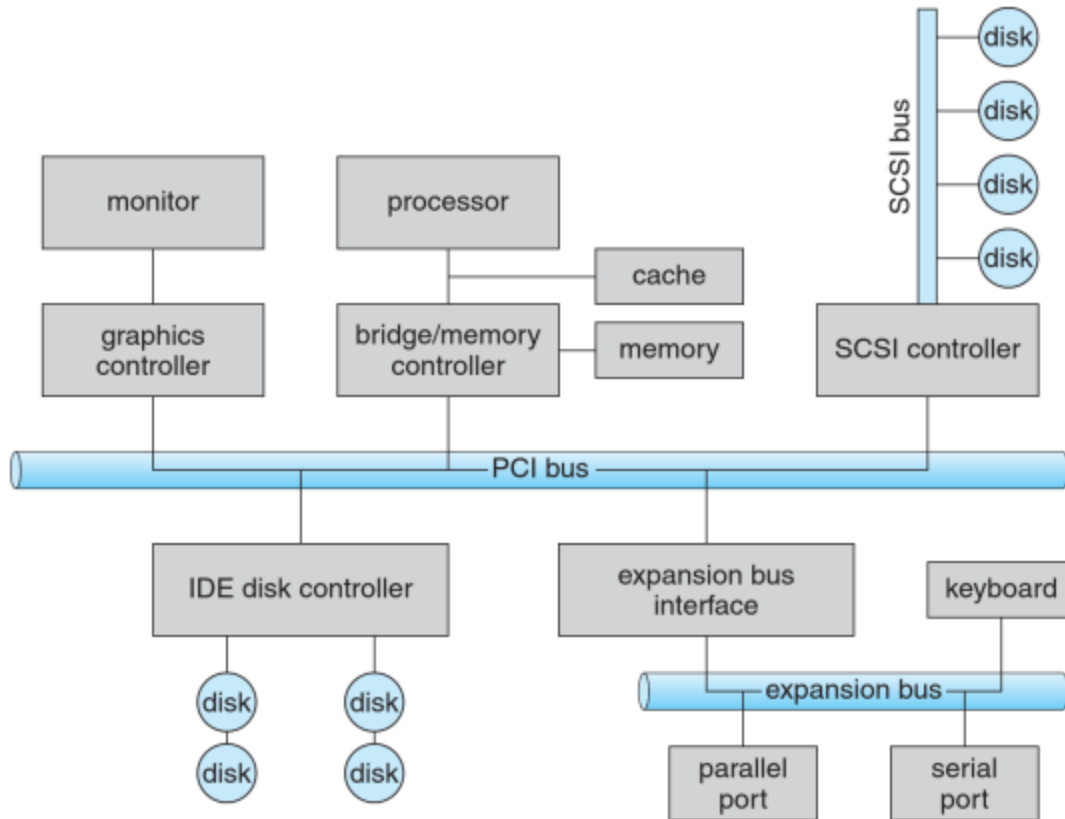
Il **dispositivo comunica** con il **computer** attraverso il **controller (traduzione tra logico a digitale)**, in un punto specifico chiamato **porta**. Tipicamente il dispositivo comunica con il cavo che arriva alla porta del computer. Oltre a questo il controller accetta comandi dal SO e li porta al dispositivo, l'interfaccia tra dispositivo e controller è standard.

Se ci sono più dispositivi che condividono gli stessi cavi per la connessione viene chiamata una connessione **bus**.

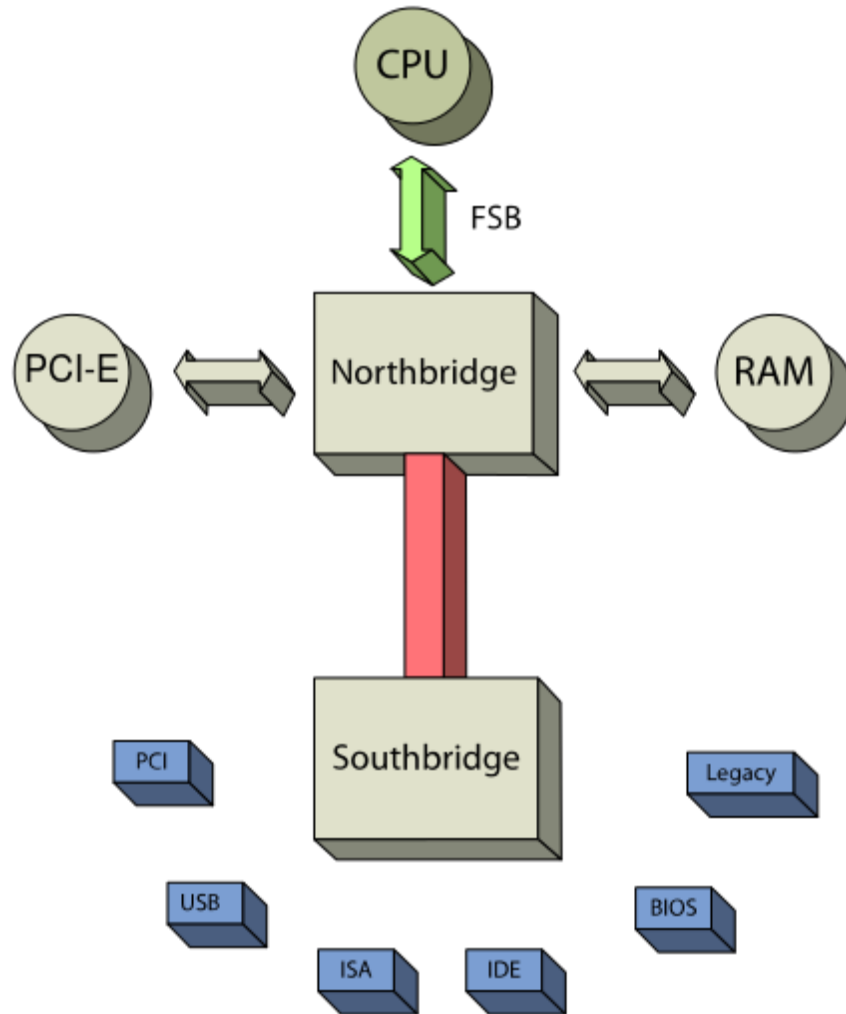
Il **controller** che risiede dentro il nostro computer **comunica con la CPU attraverso un bus**, i moderni sistemi hanno più bus di comunicazione e hanno diversi protocolli come **PCI** o **SCSI** (Small Computer System Interface) ormai vecchio rimpiazzato da SATA serve per interfacciarsi con i dischi rigidi.

Un bus **Peripheral Component Interconnect** (PCI) connette il sottosistema processore-memoria a dispositivi veloci. Ora è stato sostituito da **PCI Express** (PCIe).

Un bus di espansione connette dispositivi relativamente lenti, come la tastiera e le porte seriali e USB.

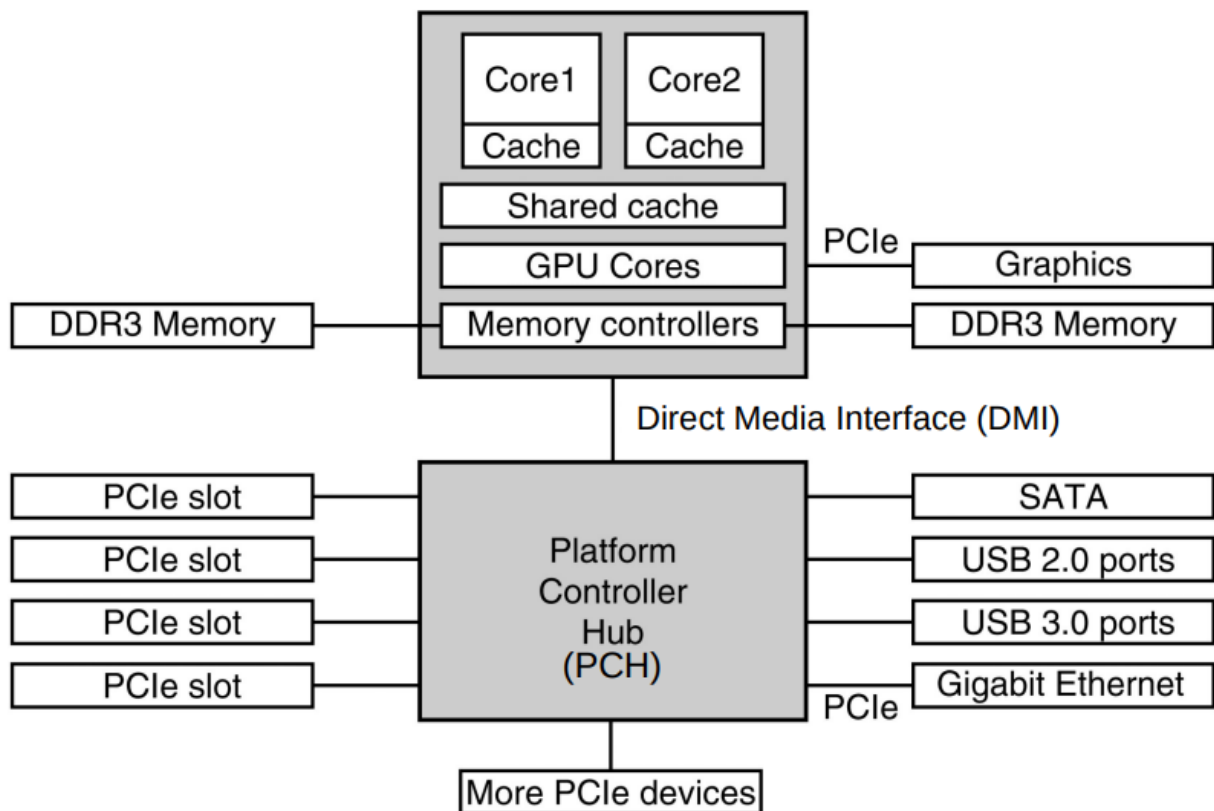


Ecco come le vecchie strutture dei controller I/O erano composte:



- **Northbridge:** ovvero il controller di memoria che controlla la comunicazione ad alta velocità tra CPU e RAM ad esempio
- **Southbridge:** controller per l'I/O che controlla la comunicazione a bassa velocità tra CPU e dispositivi I/O ad esempio
- **Front Side Bus:** connessione tra Northbridge e CPU.

Le nuove strutture dei controller I/O sono composte da:



Le funzioni del Northbridge sono integrate nella CPU ed è stato integrato il **PCH (Platform Controller Hub)** che svolge le funzioni prima implementate dal Southbridge. La comunicazione tra PCH e CPU avviene su un bus speciale chiamato **DMI**.


Un controller può avere diverse forme:

1. **Un chip singolo:** all'interno della motherboard che controlla i segnali che vengono forniti dal dispositivo dalla porta a cui sono connessi, nei sistemi moderni è un pacchetto della CPU.
2. **Un circuito dedicato:** nei sistemi moderni i dispositivi possono essere inseriti mediante dei sistemi di collegamento dedicati come il connettore **PCIe** per le schede video.

Alcuni dispositivi hanno il loro controller costruito internamente chiamati **controller periferici** e permettono ai controller del computer (host controller) di impartire dei comandi di alto livello ai controller del dispositivo. (nei HDD ad esempio si possono impartire comandi di controllo errori)




 I/O Hardware

 I/O Software

 Dischi

 SSD (Solid State Disk)

 File System

 Formulario Esame