

---

# **Il processore - CPU** **(CENTRAL PROCESSING UNIT)**

---

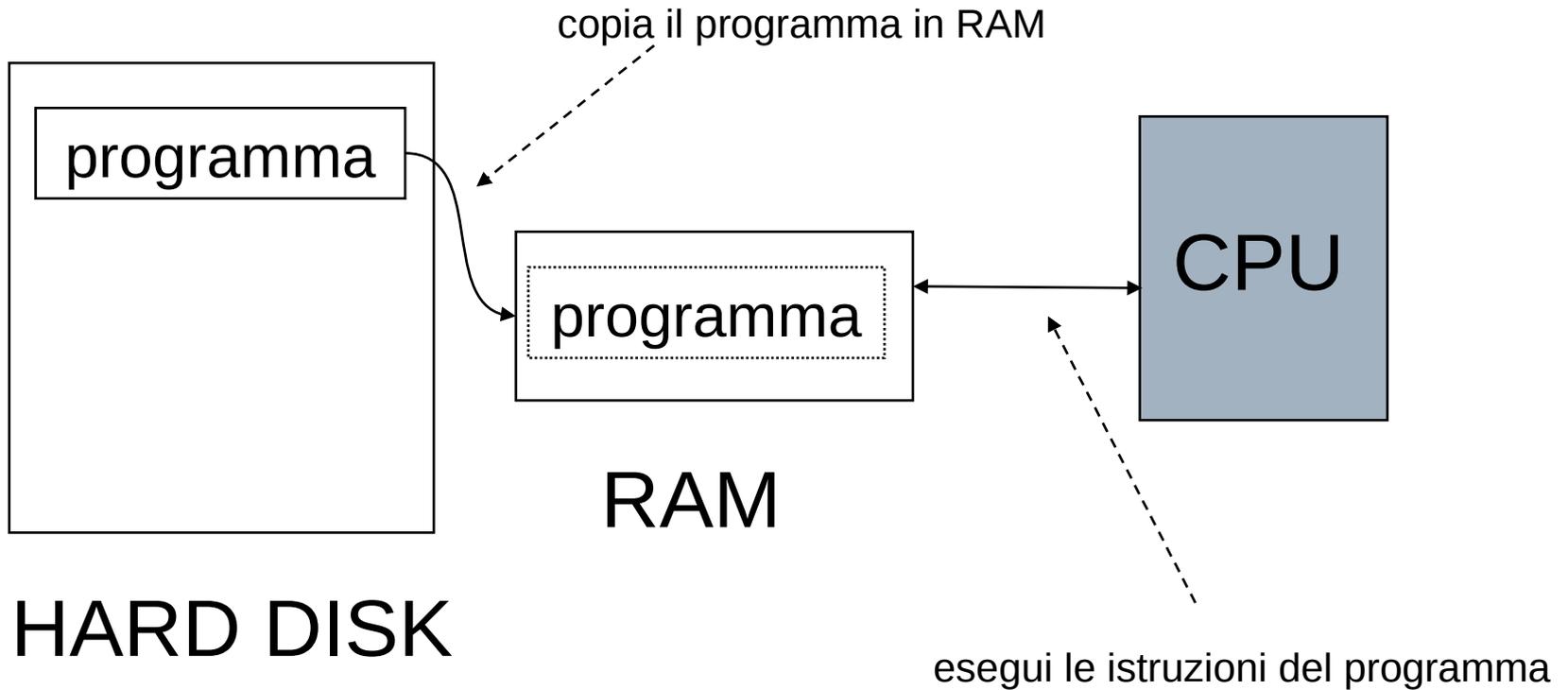
# Funzionamento

---

- **Programmi** e **dati** risiedono in file memorizzati in memoria secondaria.
  - Per essere eseguiti (i programmi) e usati (i dati) vengono copiati nella memoria primaria.
  - La **CPU** e' in grado di eseguire le istruzioni di cui sono composti i programmi
-

# Funzionamento

---



# Il processore - CPU (CENTRAL PROCESSING UNIT)

- ❑ I programmi che la CPU è in grado di eseguire sono scritti in **linguaggio macchina**
- ❑ **Esempio istruzione macchina:**

**Codice istruzione | argom. 1 | argom. 2**

- Lunghezza: word/parola della macchina
  - Numero di argomenti dipende dall'istruzione
  - Sequenza di bit!
  - Per leggibilità: trascrizione
-

# Un programma in linguaggio macchina (**ASSEMBLY**)

---

1000	LOAD 3568 R1
1004	LOAD 3574 R2
1008	ADD R1 R2
1012	STORE R1 3568
1016	JUMP 1000
	.....

---

# Il set di istruzioni macchina

---

- Ogni tipo di processore è in grado di eseguire un numero molto limitato di istruzioni
  - Sequenze di istruzioni di base: **programmi**
-

# Il set di istruzioni macchina

---

- Istruzioni per l'elaborazione dei dati
    - aritmetiche
    - logiche (AND, OR, NOT)
    - relazionali (maggiore, minore, uguale, ...)
  - Controllo del flusso delle istruzioni
    - sequenza
    - selezione
    - ciclo
  - Trasferimento di informazioni
    - dati ed istruzioni fra CPU e memoria
    - dati fra CPU e dispositivi di I/O (tramite interfacce)
-

# Caratteristiche dei microprocessori

---

- **Repertorio di istruzioni**
    - L'insieme delle istruzioni che costituiscono il linguaggio macchina del processore
  - **Frequenza di clock**
    - l'esecuzione di una istruzione può richiedere più cicli macchina
  - **Ampiezza del bus**
    - numero di bit nel bus interno del processore
  - **Co-processor**
    - processori specializzati per operazioni complesse (es: co-processore matematico)
  - **Memoria cache**
    - una memoria veloce locale al processore, che consente di accedere più velocemente ai dati da elaborare
-

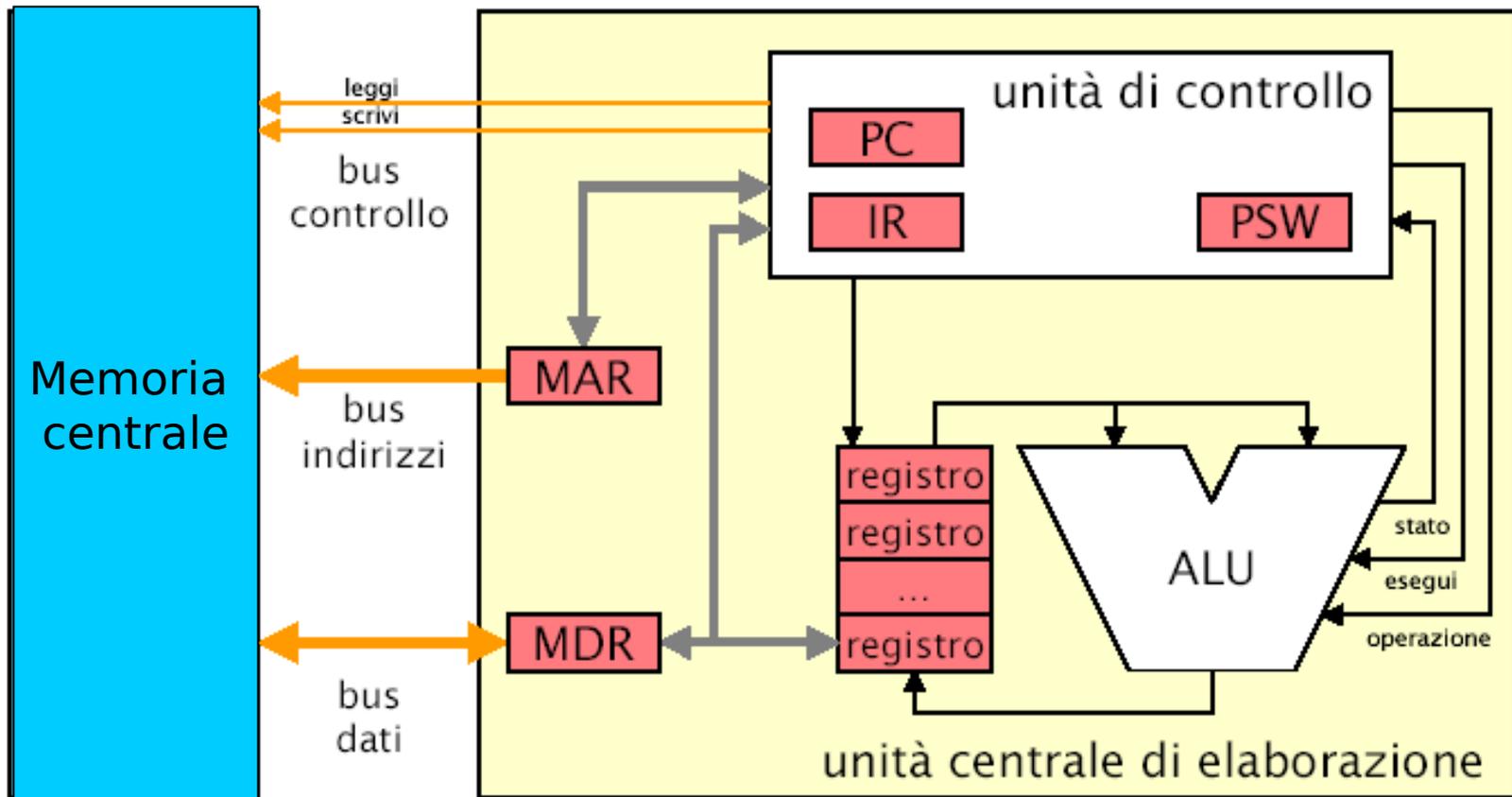
# Frequenza di clock

---

La frequenza con cui si eseguono i cicli di esecuzione è scandita dal **clock** (“orologio”, ma più “ritmo”)

- ad ogni impulso di clock l'unità di controllo esegue un ciclo di esecuzione
  - la velocità di elaborazione di un microprocessore dipende dalla frequenza del suo clock (500, 1000,... MHz) (es.: 500 milioni di cicli al secondo).
-

# Struttura del processore



# Le componenti della CPU:

## I REGISTRI

---

- piccole unita' di memoria (2, 4, 8 byte) con tempi di accesso molto piu' bassi delle celle della memoria primaria
  - Ospitano le informazioni necessarie per eseguire l'istruzione corrente
  - In numero molto limitato (16, 32, 64) si dividono in registri **speciali** e **generali**
-

# Le componenti della CPU:

## I REGISTRI SPECIALI

---

### □ Il Program Counter (PC)

- Contiene l'indirizzo in memoria centrale della **prossima istruzione** da eseguire
  - All'inizio dell'esecuzione di un programma viene caricato con l'indirizzo della prima istruzione di quel programma
  - Ad ogni istruzione eseguita il PC viene modificato per contenere l'indirizzo della istruzione successiva
-

# Le componenti della CPU:

## I REGISTRI SPECIALI

---

### □ L'Instruction Register (IR)

- Contiene l'istruzione correntemente in esecuzione
- La CPU legge l'istruzione contenuta nell'Instruction Register e la esegue

### □ IL Registro di stato (PSW)

- Descrive lo stato corrente della esecuzione
  - Segnala eventuali errori (ad es.: overflow)
-

# Le componenti della CPU:

## I REGISTRI SPECIALI

---

### □ Registro Indirizzi Memoria (MAR)

- contiene l'indirizzo della cella da cui leggere o in cui scrivere un dato

### □ Registro Dati Memoria (MDR)

- contiene il dato letto dalla memoria o da scrivere in memoria
-

# Le componenti della CPU:

## I REGISTRI GENERALI

---

### □ I registri generali

- in numero di 8, 16, 32, 64
  - sono usati come memorie temporanee per contenere gli operandi delle istruzioni e i risultati parziali durante l'esecuzione delle istruzioni
-

# Le componenti della CPU:

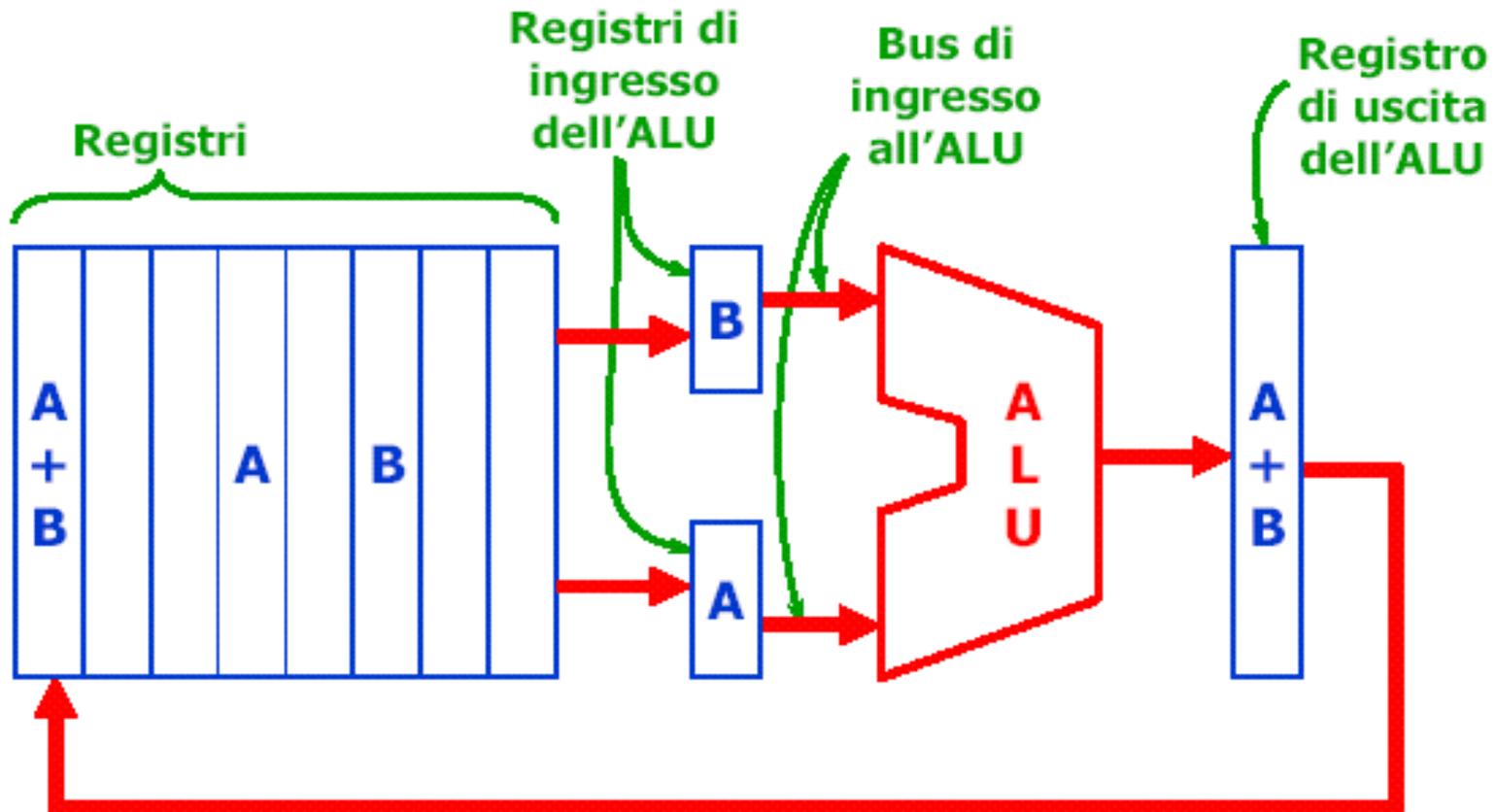
## ARITHMETIC-LOGIC UNIT (ALU)

---

- Si occupa di eseguire le operazioni di tipo aritmetico/logico: somme, confronti...
  - Preleva gli operandi dai / deposita il risultato delle operazioni nei: registri generali
  - A volte e' affiancata da un **co-processore matematico**
-

# Le componenti della CPU:

## ARITHMETIC-LOGIC UNIT (ALU)



# Le componenti della CPU:

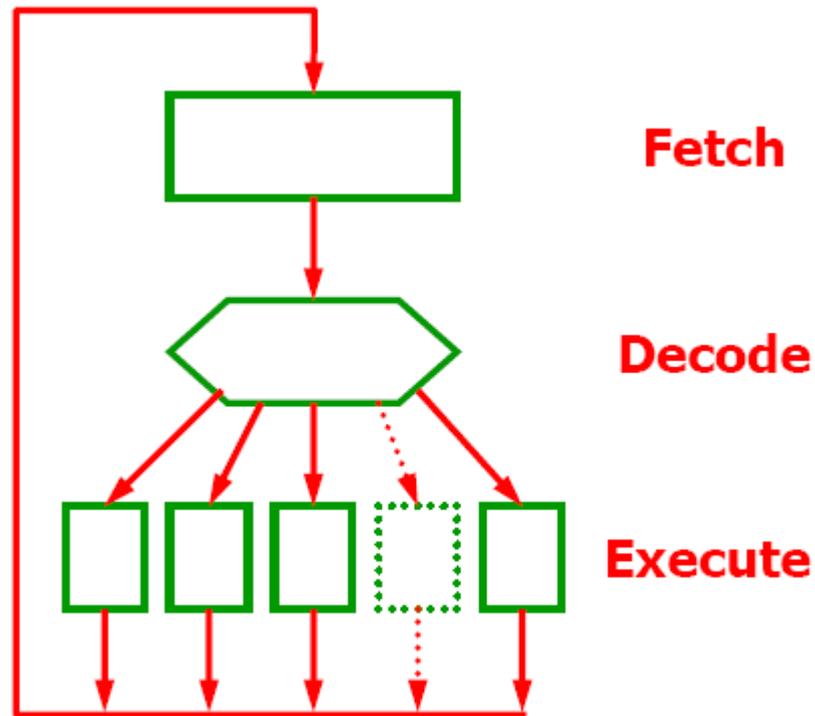
## LA CONTROL UNIT (CU)

---

- L'unità di controllo del processore (CPU) esegue una istruzione svolgendo le seguenti **tre operazioni** di base:
    - **Fetch** (lettura)
    - **Decode** (decodifica)
    - **Execute** (esecuzione)
  - Un programma è eseguito reiterando il ciclo *fetch-decode-execute* (**ciclo macchina**) per eseguire ordinatamente le sue istruzioni
-

# Il ciclo Fetch-Decode-Execute

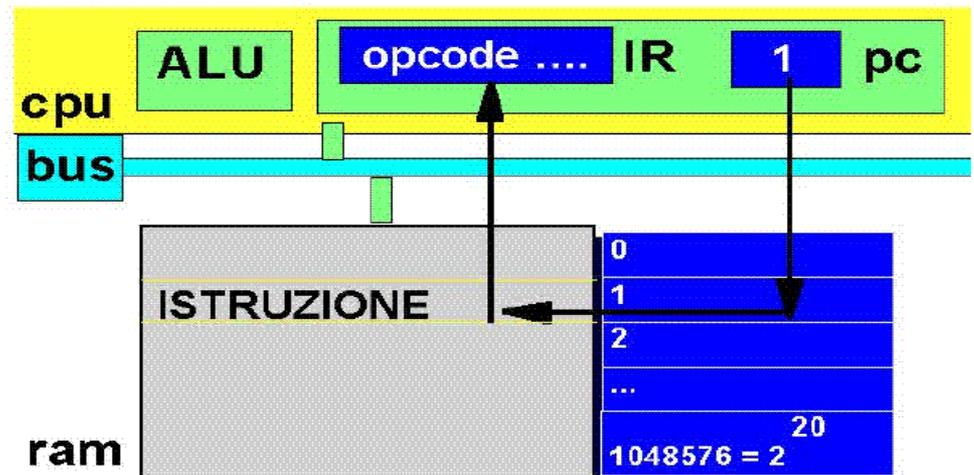
---



# Ciclo *fetch-decode-execute*

## 1) **FETCH:**

- si accede alla prossima istruzione, riferita dal registro contatore dell'istruzione (PC)
- si porta tale istruzione dalla memoria centrale al Registro Istruzioni (IR)



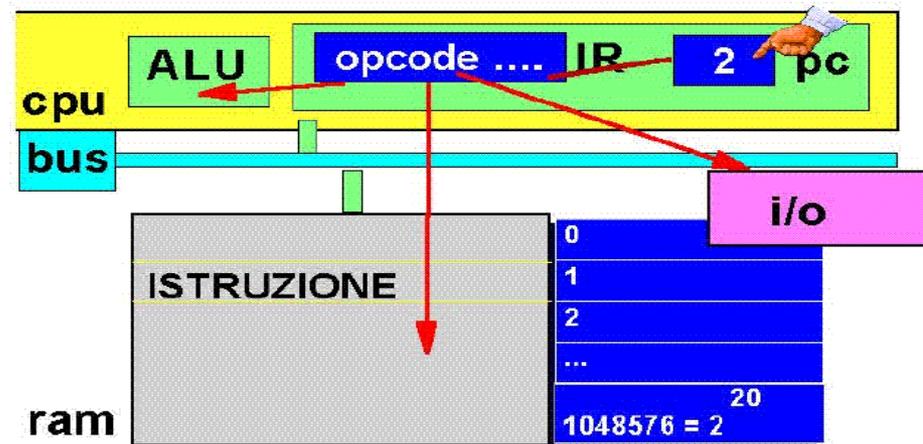
# Ciclo *fetch-decode-execute*

---

- **2) DECODE:** decodifica dell'istruzione
    - si individua il tipo dell'operazione e gli operandi (dati) usati
    - si trasferiscono i dati nei registri opportuni
-

# Ciclo *fetch-decode-execute*

- **3) EXECUTE:** esecuzione dell'istruzione
  - si incrementa il registro contatore dell'istruzione (PC)
  - ciascuna azione viene richiesta al componente opportuno



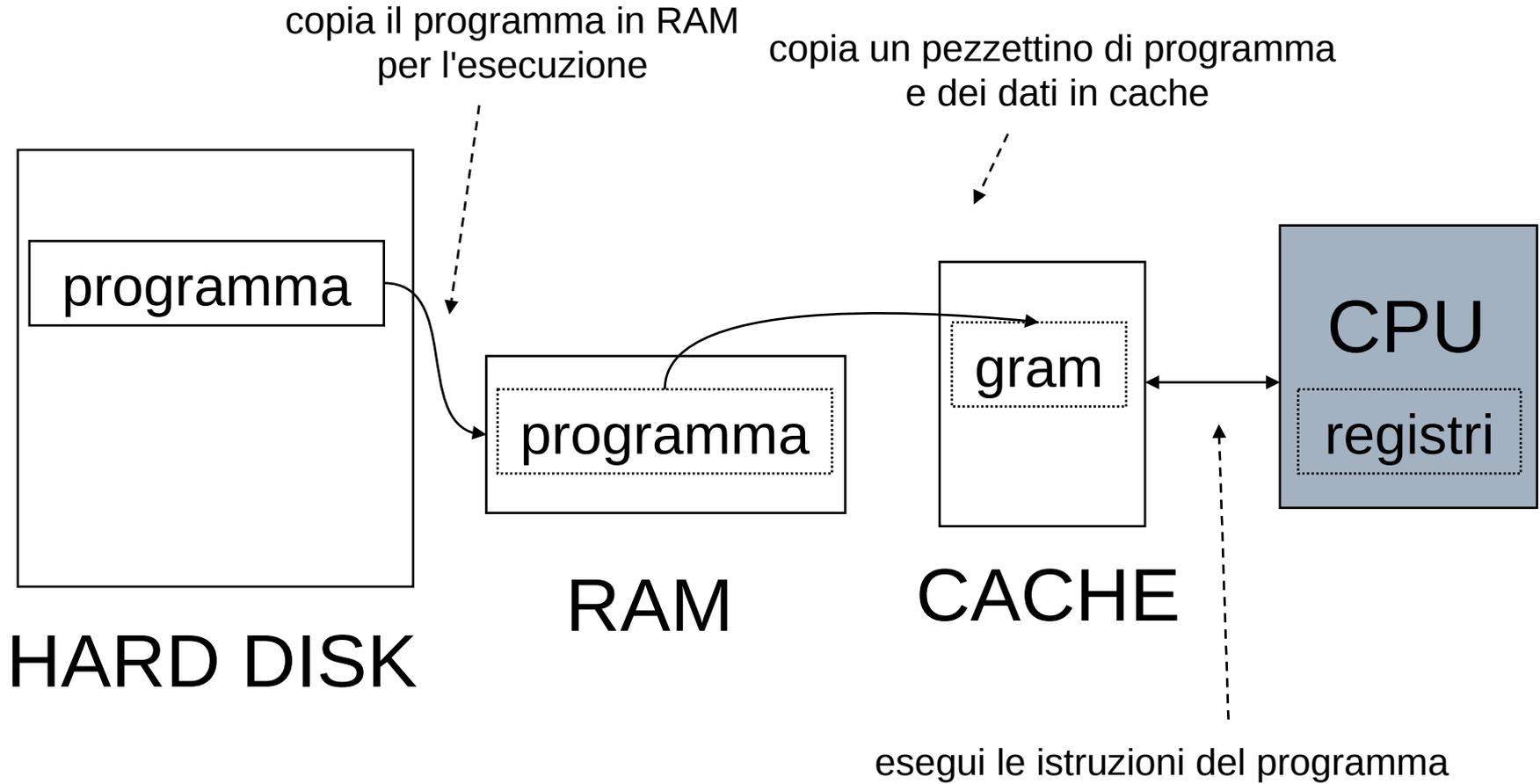
# MEMORIA CACHE

---

- ❑ Livello di **memoria intermedio** tra i registri e la ram
  - ❑ Per memorizzare i dati usati piu' spesso senza doverli recuperare in memoria
  - ❑ 64, 128, 256, 512KB, 1MB, 2MB
  - ❑ **Interna** o **esterna** alla CPU
  - ❑ Influisce moltissimo sulle prestazioni e il costo della CPU (e quindi del computer)
  - ❑ I computer attuali hanno spesso **più livelli** di cache
-

# MEMORIA CACHE

---



# Memorie di un computer

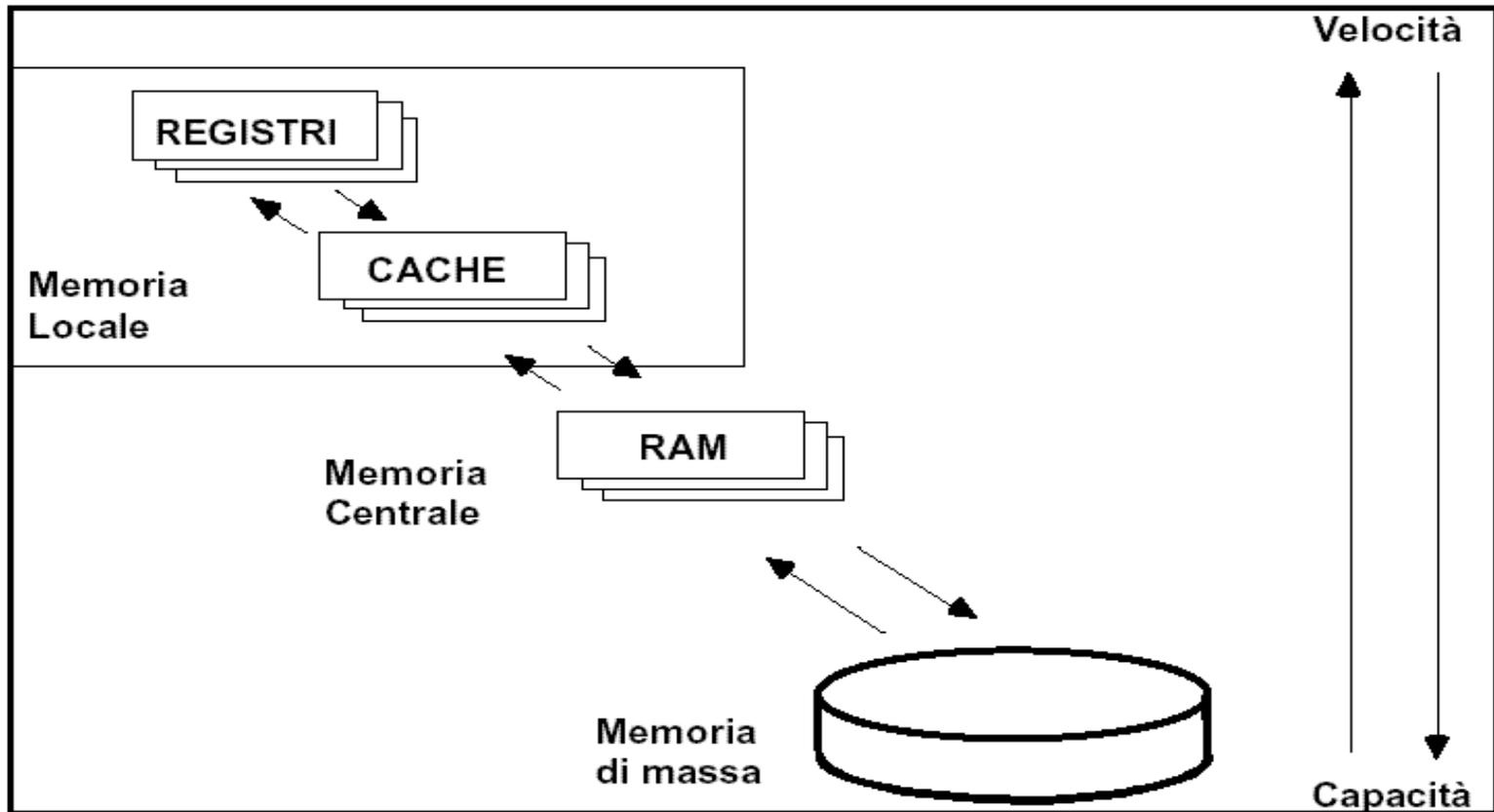
---

<b>Registri</b>	< 1 KByte	100 * picosecondi
<b>Cache</b>	< 1 MByte	nanosecondi
<b>RAM</b>	< 4 Gbyte	10 * nanosec
<b>Hard disk</b>	> 40 GByte	10 * microsec
<b>Dischi ottici</b>	650MB-17GB	micro/millisecondi
<b>Nastri</b>	> 10 GByte	10 * millisecondi

---

# Gerarchia delle memorie

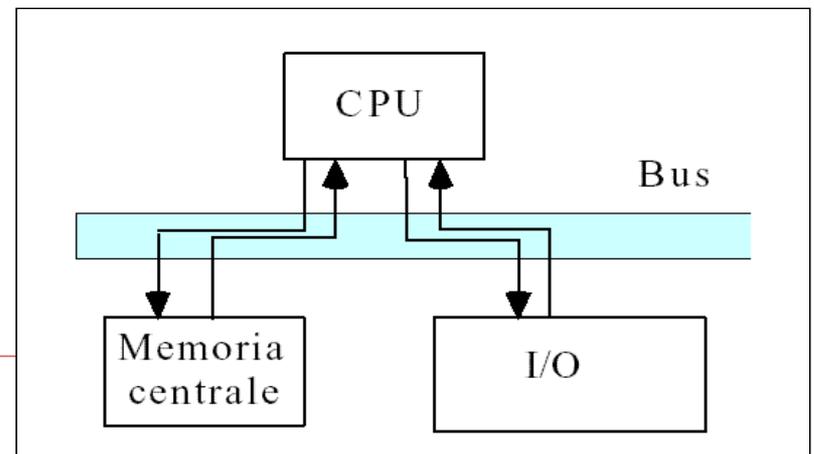
---



# Bus di sistema

---

- ❑ Interconnette CPU, memorie ed interfacce verso dispositivi periferici (I/O, memoria di massa, ...)
- ❑ Collega **due unità funzionali alla volta**
  - una trasmette e l'altra riceve
- ❑ Il trasferimento dei dati avviene sotto il controllo della CPU



# Bus di sistema

---

- Un bus trasporta dati, indirizzi e comandi
  - Componenti del bus (sottogruppi di linee):
    - **Bus dati (data bus)**
    - **Bus indirizzi (address bus)**
    - **Bus comandi (command bus)**
  
  - **Bus dati (data bus)**
    - Serve per trasferire dati
      - tra la memoria centrale ed il registro dati (MDR) della CPU
      - tra periferiche e CPU (o memoria centrale)
    - Bidirezionale
-

# Bus di sistema

---

## **Bus indirizzi (address bus)**

- Serve per trasmettere il contenuto del registro indirizzi (MAR) alla memoria (o ad una periferica)
  - si seleziona una cella per successive operazioni di lettura o scrittura
- Unidirezionale

## **Bus comandi (command bus)**

- Serve per inviare comandi
    - verso la memoria (es: lettura o scrittura)
    - o verso una periferica (es. stampa verso la stampante / interfaccia)
  - Unidirezionale
-

# Dispositivi di input/output (I/O, PERIFERICHE)

---

- **Terminali.** Tastiera + Video:
    - risoluzione, dimensione in pollici,...
  - **Stampanti:**
    - ad aghi, a getto, d'inchiostro, laser,...
  - **Modem:** per collegarsi in rete
  - **Scanner:** per digitalizzare le immagini
-

# Dispositivi di Input

---

**Tastiera**



**Mouse**



**trackball**



**Touch pad**



**Microfono**



**Tavoletta grafica**



**Webcam**



**Penna ottica**



**Scanner**



**Fotocam. Dig.**



**Schermi touch screen**



**Joystick**



# Dispositivi di Output

---

**Monitor**



**Plotter**



**Stampante**



**Casse acustiche**



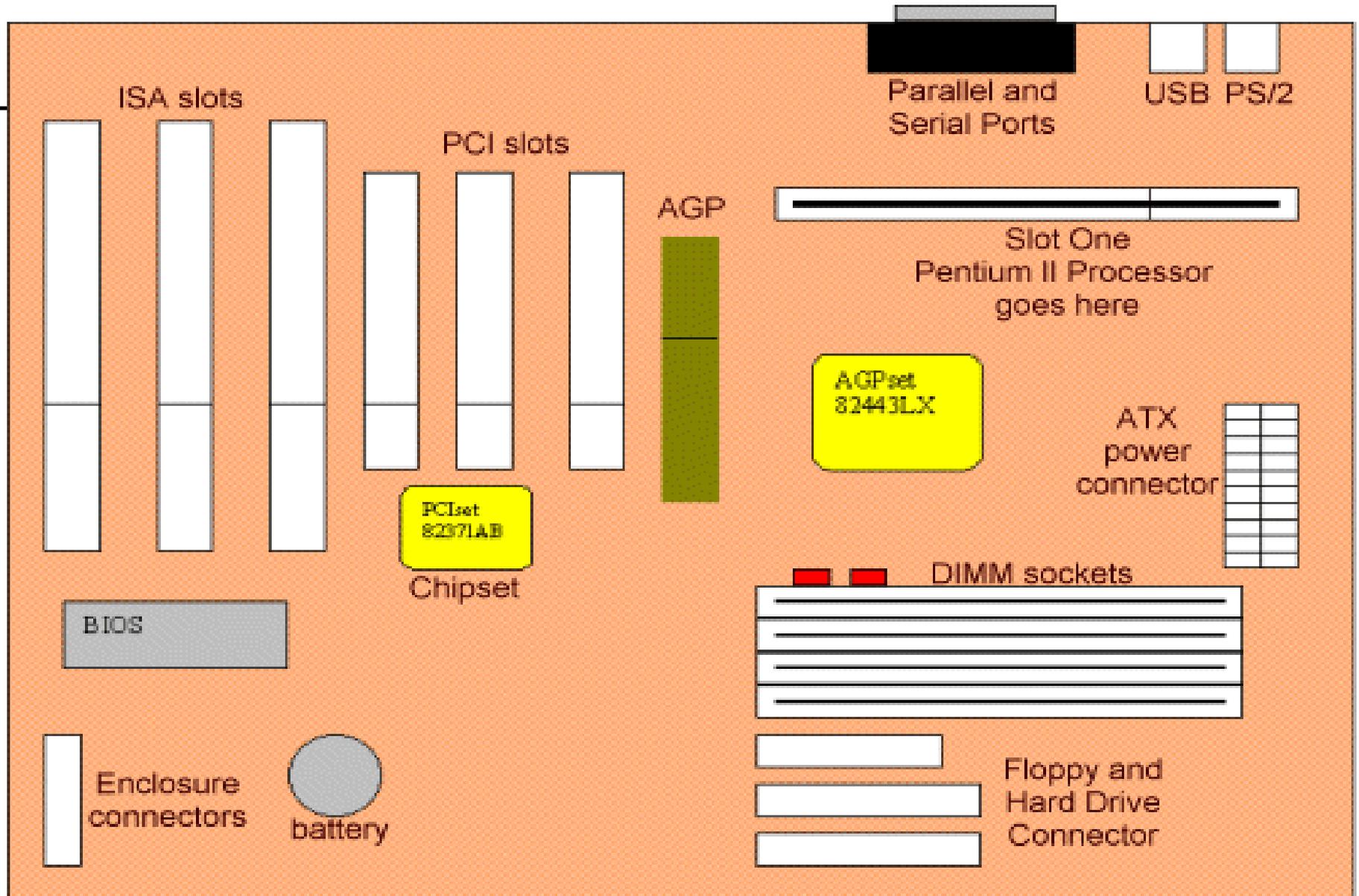
**Cuffie**



**Videoproiettore**



# Piastra madre



# Il software di base

---

- ❑ Dobbiamo impartire ordini al computer usando solo il codice binario???
  - ❑ Ovviamente no: il computer è dotato di alcuni programmi (il **software di base - sistema operativo**) che rendono il computer facile da usare
  - ❑ Questi programmi trasformano il computer in una **macchina virtuale**, più vicina alle esigenze dell'utente che può così ignorare i dettagli implementativi
-

# Il Sistema Operativo

---

- ❑ Senza il Sistema Operativo (SO) il computer sarebbe scomodissimo e complicatissimo da usare
  - ❑ Permette di sfruttare le risorse del computer in modo semplice e (si spera) intuitivo (si pensi ad esempio alle interfacce grafiche)
  - ❑ Permette di usare i programmi che ci interessano (di scrittura, di studio, i videogames) senza preoccuparci di come questo avvenga all'interno del computer
-

# Il software

---

- I **linguaggi di programmazione ad alto livello**, permettono di:
    - scrivere i propri programmi, cioè di usare il computer come vogliamo noi
    - di poter usare questi programmi su quasi qualsiasi computer: **portabilità**
-

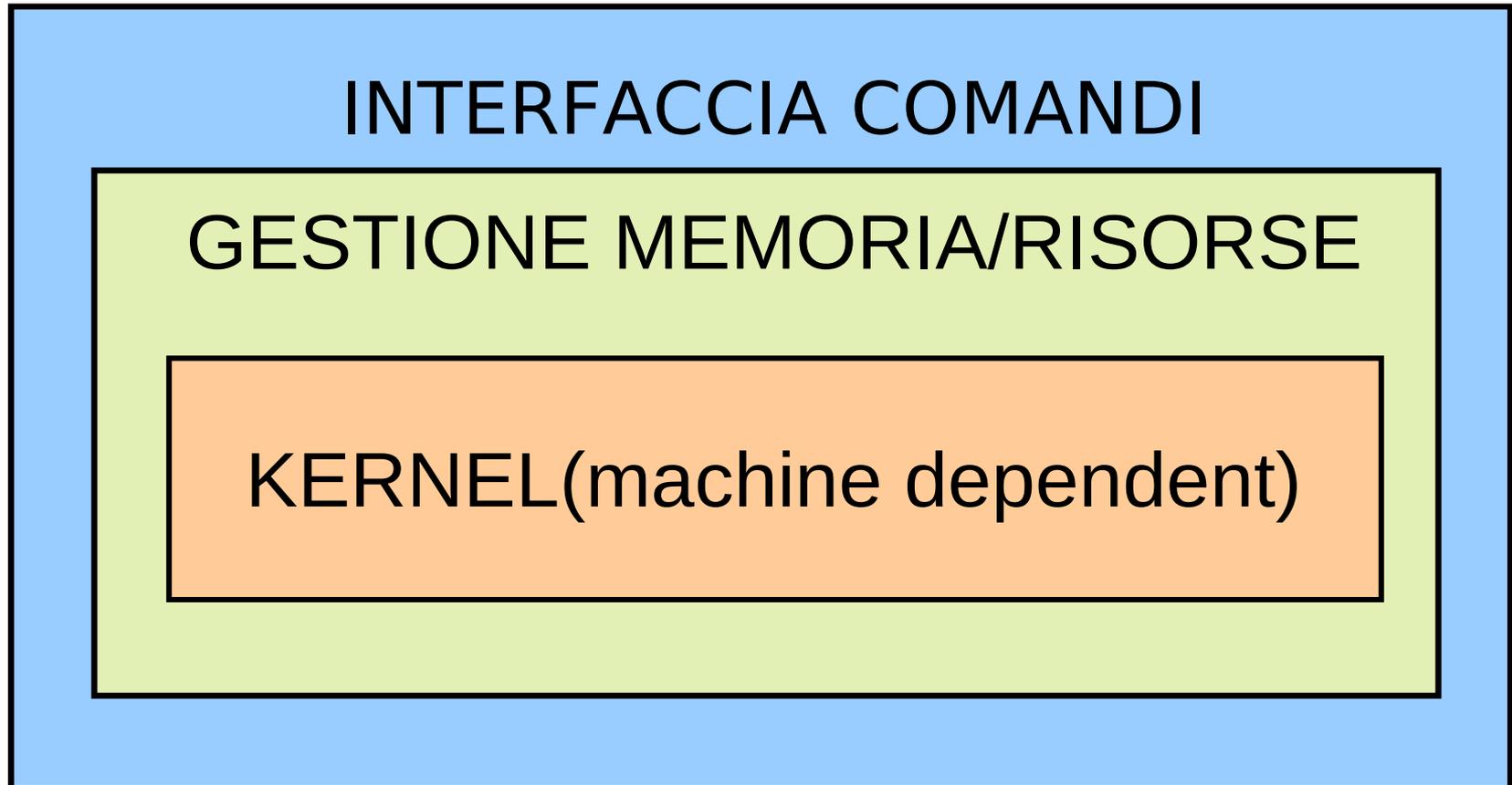
# Cosa fa il sistema operativo?

---

- ❑ Gestisce in modo efficiente le risorse del computer: cpu, memoria, periferiche
  - ❑ Gestisce l'interfaccia dell'utente
  - ❑ Gestione dell'esecuzione di programmi
  - ❑ Nasconde la macchina fisica all'utente
-

# La cipolla del sistema operativo

---



# Tipi si sistemi operativi

---

- ❑ **Mono-utente o Multi-utente (Mono/multi-user)**
  - ❑ **Mono o Multi-programmati (Mono/multi-tasking)**
  - ❑ **Distribuiti (o di rete)**
-

# Le funzioni principali dei sistemi operativi

---

- ❑ Gestione del processore e dei processi
  - ❑ Gestione della memoria principale
  - ❑ Gestione della memoria virtuale
  - ❑ Gestione della memoria secondaria (il file system)
-