

Hardware e software

Marco Alberti



Dipartimento
di Matematica
e Informatica



Università
degli Studi
di Ferrara

Programmazione e Laboratorio, A.A. 2024-2025

Ultima modifica: 6 dicembre 2023

Attenzione! Questo materiale didattico è per uso personale dello studente ed è coperto da copyright.
Ne sono vietati la riproduzione e il riutilizzo anche parziale, ai sensi e per gli effetti della legge sul diritto d'autore.

Sommario

- Hardware
 - Funzionalità di un computer
 - Architettura di Von Neumann
 - Memoria
 - Central Processing Unit
 - Input/Output
- Software
- Sistema Operativo
- Macchina astratta

Sommario

1 Hardware

2 Software

Computer

Per essere utile, un computer deve essere in grado di

- Ricevere dati che rappresentano il problema (**input**)
- **Elaborare** i dati per calcolare la risposta
- Fornire la risposta (**output**)

Elaborare: eseguire operazioni sui dati

- Operazioni complesse ottenute come composizione (**programma**) di operazioni elementari (**istruzioni**)
- Necessario memorizzare
 - risultati intermedi dell'elaborazione
 - programma

Esempio: moltiplicazione in colonna

- input: lettura degli operandi
- selezione del programma: lettura dell'operatore
- copia degli operandi nella memoria di lavoro
- operazioni elementari: addizione e moltiplicazione di numeri di una cifra, con gestione dei riporti
- scrittura e lettura dei risultati parziali nella memoria di lavoro
- programma: combinazione di operazioni elementari, ripetizione soggetta a condizioni
- output: copia del risultato dalla memoria di lavoro

Funzionalità di un computer

Al computer servono:

- Uno o più componenti in grado di:
 - eseguire operazioni elementari
 - leggere ed eseguire il programma
 - leggere e scrivere nelle memorie di lavoro
- Memorie per
 - programmi
 - dati letti da input
 - risultati intermedi
 - dati da scrivere in output
- Dispositivi per comunicare con l'esterno

Famiglie di computer

In ogni computer sono presenti i componenti elencati; i computer differiscono per

- come sono realizzati i componenti (tecnologia);
- come sono organizzati per formare il sistema (architettura).

Alcune tappe dell'evoluzione dei computer:

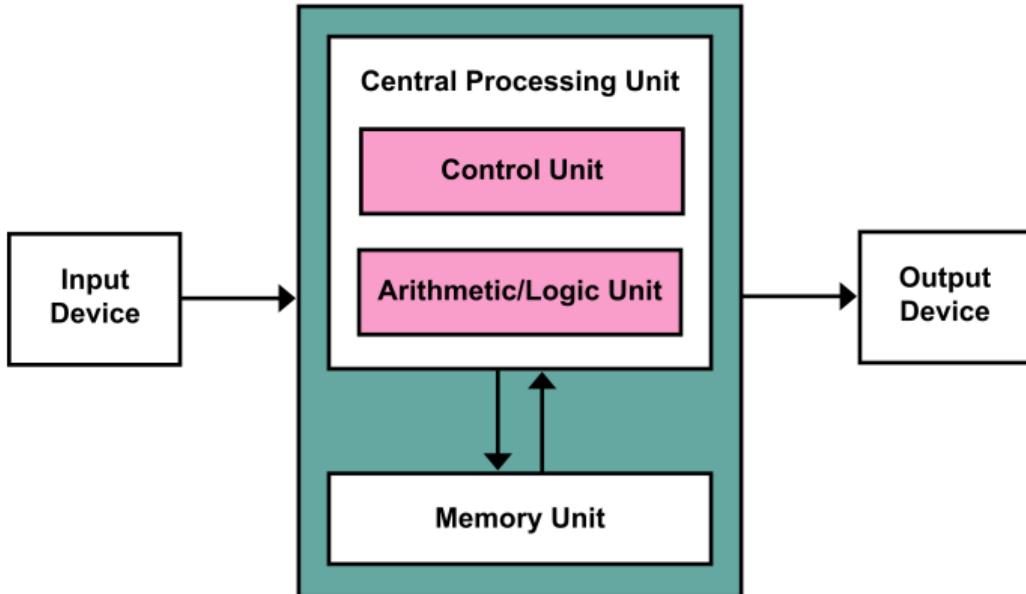
1833 Charles Babbage, *Analytical Engine*, primo calcolatore programmabile, meccanico, non fu mai costruito.

WW2 Vari calcolatori tra cui

- Z3 (Germania), elettromeccanico
- Colossus (UK), elettronico (a valvole), usato per decodificare il codice tedesco ENIGMA

1948 SSEM, primo calcolatore basato sull'architettura di Von Neumann, tuttora dominante.

Architettura di Von Neumann



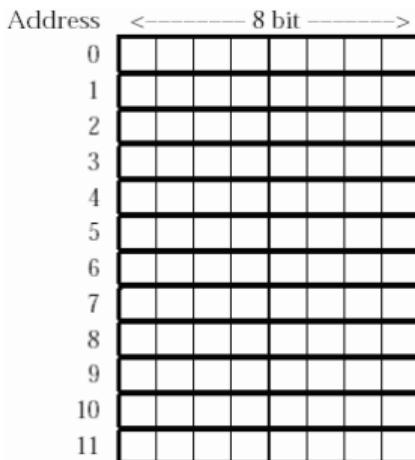
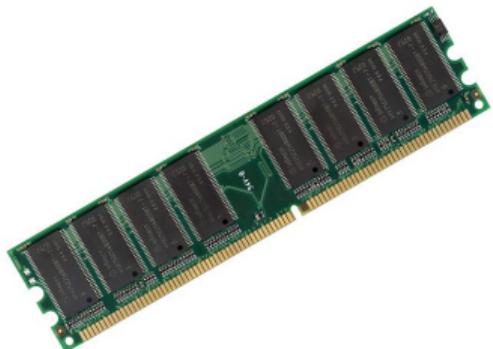
Memoria centrale

Nell'architettura di Von Neumann, programmi e dati (letti, di lavoro e da scrivere) sono memorizzati nello stesso componente: la **memoria centrale**.

Due tipologie principali:

- Read Only Memory (ROM):
 - non volatile (non si cancella quando non alimentata)
 - (tendenzialmente) sola lettura
 - usata per parte dei programmi di sistema
- Random Access Memory (RAM):
 - volatile
 - lettura e scrittura
 - usata per i dati e gran parte dei programmi in esecuzione

Memoria centrale



La memoria centrale è, logicamente, una sequenza di celle di capacità fissata (es. 8 bit), identificate da un numero intero positivo progressivo detto **indirizzo**. Un dato scritto in una cella si può leggere in seguito sapendo l'indirizzo della cella, come un appunto scritto su carta.

Unità di misura della quantità di informazione

La capacità di una memoria è la quantità di informazione che può contenere

Unità	Abbrev.	Valore	Che cosa misura (2024)?
1 bit	b		Destra o sinistra?
1 byte	B	8 bit	Un piccolo intero
1 KiloByte	KB	$2^{10} (= 1.024)$ B	Pagina di testo
1 MegaByte	MB	$2^{20} (= 1.048.576)$ B	Una fotografia
1 GigaByte	GB	$2^{30} (= 1.073.741.824)$ o 10^9 B	Un film
1 TeraByte	TB	10^{12} B	Wikipedia ($\times 10$)
1 PetaByte	PB	10^{15} B	Master Netflix ($\times 20$)
1 ExaByte	EB	10^{18} B	Google ($\times 10$)

Central Processing Unit



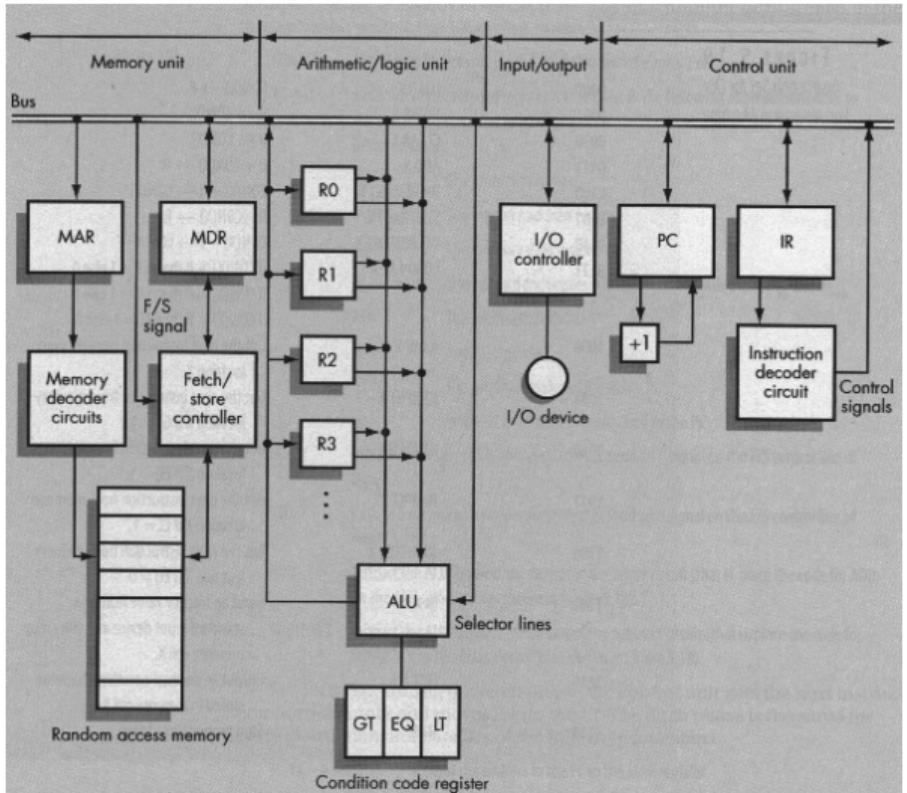
Detta anche processore.

Esegue i calcoli aritmetici e logici, interpreta ed esegue i programmi.

Opera alla **frequenza** (misurata in Hz o multipli, eventi al secondo) determinata dal segnale di **clock**.

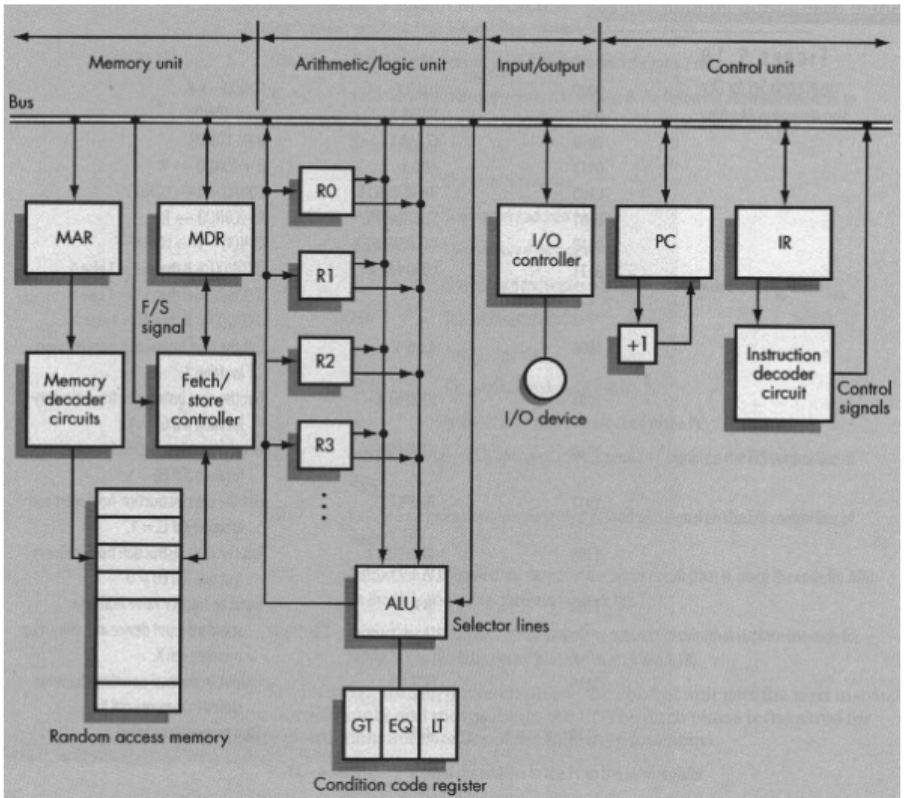
Ha un numero limitato di celle di memoria dette **registri**.

Registri



- IR: Instruction Register, istruzione in esecuzione
- R0 ... RN: registri di lavoro
- MAR: Memory Address Register, contiene l'indirizzo della RAM a cui accedere
- MDR: Memory Data Register, contiene il dato da scrivere o appena letto in RAM
- PC: Program Counter, indirizzo della prossima istruzione da eseguire
- Condition Code Register, risultato dell'ultima operazione di confronto

Interazione con RAM



fetch(address): legge il valore della cella all'indirizzo **address**

- ① imposta **address** in MAR
- ② decodifica MAR
- ③ copia in MDR il valore della cella all'indirizzo decodificato

store(address, value): scrive **value** alla cella all'indirizzo **address**

- ① imposta **address** in MAR
- ② imposta **value** in MDR
- ③ decodifica MAR
- ④ copia all'indirizzo decodificato il valore di MDR

Istruzioni macchina

La CPU è in grado di eseguire istruzioni in un linguaggio binario detto **linguaggio macchina**.

Ogni istruzione letta da RAM è un numero binario (sequenza di 0 e 1) diviso in due parti di lunghezza fissa:

- **codice operazione** (es. 00110110) che rappresenta l'istruzione da eseguire
- uno o più **campi di indirizzo** che rappresentano gli operandi dell'istruzione

Ciclo CPU

- ① **fetch**: l'istruzione in RAM all'indirizzo PC è copiata in IR, e PC è incrementato
- ② **decode**: l'istruzione da eseguire viene selezionata
- ③ **execute**: l'istruzione selezionata viene eseguita

Istruzioni macchina

Le istruzioni (operazioni elementari) disponibili sono l'**instruction set**, che varia da CPU a CPU. Famiglie:

- CISC: Complete Instruction Set Computing (es. x86)
- RISC: Reduced Instruction Set Computing (es. ARM)

Categorie di istruzioni:

- Operazioni aritmetiche (es. addizione) fra registri, con risultato in un registro
- Operazioni di confronto fra numeri (es. \leq), che impostano il condition code register
- Salti (impostano il PC, cioè l'indirizzo della prossima istruzione da eseguire)
- Trasferimento dati (da registro a RAM, da RAM a registro, fra registri)

Linguaggio Assembly

Le istruzioni macchina sono sequenze di 0 e 1 difficili da ricordare e leggere.

Le istruzioni macchina sono in corrispondenza biunivoca con le istruzioni di un linguaggio (specifico della CPU) detto **Assembly** o impropriamente (ma frequentemente) **Assembler**.

Alla seguente istruzione in linguaggio C

```
c = a + b;
```

corrispondono le seguenti istruzioni Assembly x86:

```
mov edx, DWORD PTR [rbp-8]
mov eax, DWORD PTR [rbp-12]
add eax, edx
mov DWORD PTR [rbp-4], eax
```

Dispositivi di Input/Output

- Tastiera e mouse
- Video, stampanti
- Touchscreen
- Fotocamera
- Modem
- Dispositivi di memorizzazione di massa

Tecnologie di memorizzazione di massa

Distinzione fondamentale:

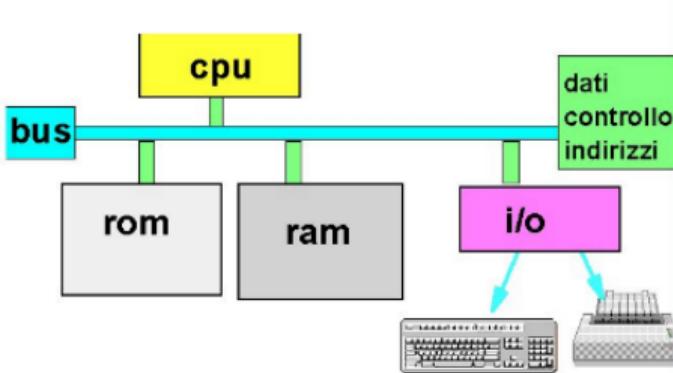
- **Accesso sequenziale:** dati letti nell'ordine in cui si trovano sul supporto
- **Accesso diretto:** ogni dato è accessibile direttamente, indipendentemente dalla posizione sul supporto

Tecnologie:

- Nastri magnetici (accesso sequenziale)
- Dischi magnetici (floppy disk, hard disk)
- Dischi ottici (CD, DVD, Blu¹-Ray)
- Memorie a stato solido (pen-drive, SSD)

¹sic

Comunicazione fra componenti



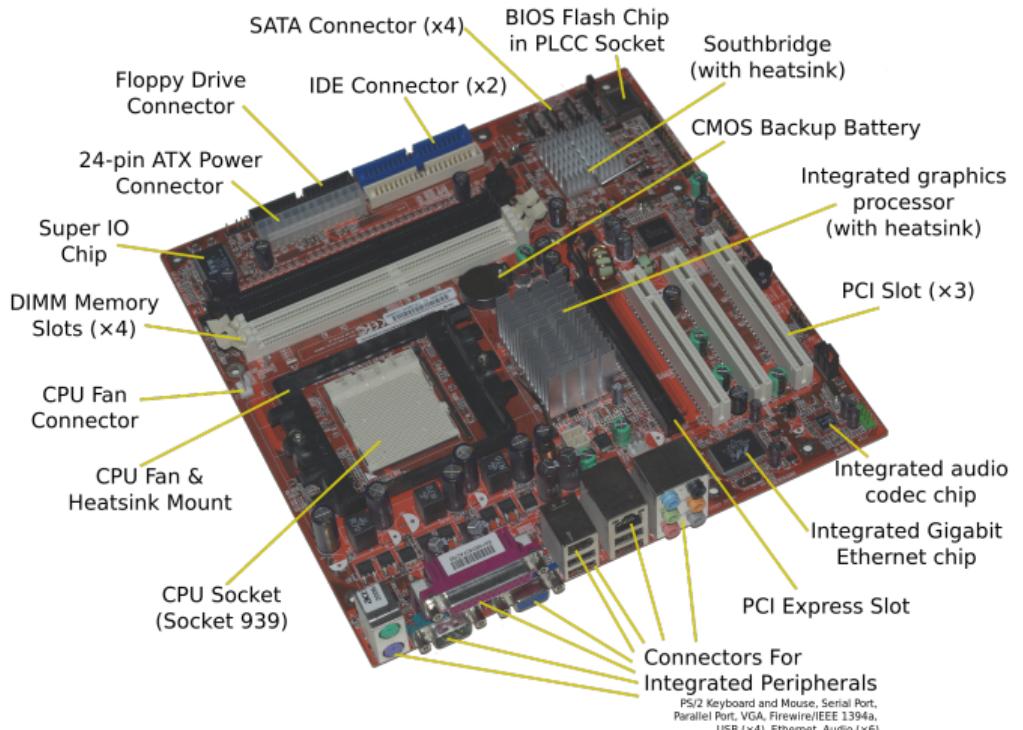
Nell'architettura di Von Neumann, il **bus** è, logicamente, la linea di comunicazione fra componenti.

Es. **store(12345678,12)**

- 12345678 sul bus indirizzi
- 12 sul bus dati
- **store** sul bus controllo

Oggi, fisicamente, è realizzato come sistema di componenti e connessioni.

Scheda madre



Legge di Moore



- IBM PC (1981): CPU Intel 8088 4.77Mhz, (fino a) 256KB RAM, HD 10MB (optional!), 5000\$ (corretto per inflazione)
- Tipico notebook (2020): CPU 3GHz (> 500x), 8GB RAM (> 30.000x), HD 1TB (100.000x), 400\$

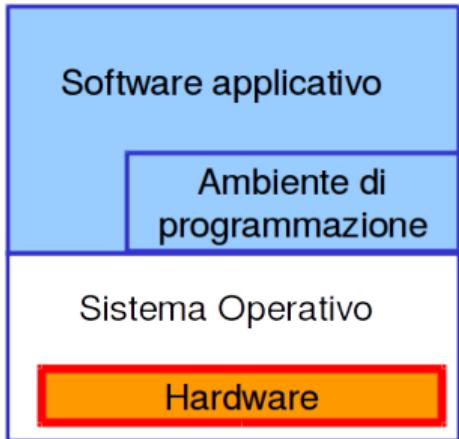
Gordon Moore (co-fondatore di Intel): "La complessità di un microcircuito, misurata ad esempio tramite il numero di transistori per chip, raddoppia ogni 18 mesi." Negli ultimi anni, però, la velocità di crescita sta calando.

Sommario

1 Hardware

2 Software

Software



- Complesso dei programmi che possono essere eseguiti su un computer.
- Diviso in
 - Applicazioni (che fanno qualcosa di utile per l'utente)
 - Software di sistema (che permette alle applicazioni e agli utenti di usare le risorse del computer)

Sistema operativo

Software che rende le risorse del computer fruibili alle applicazioni e agli utenti.

Caricato all'accensione del computer (**boot**) e sempre in esecuzione.

Funzionalità principali:

- Gestione dei processi
- Gestione dei dispositivi (periferiche)
- File system
- Gestione degli utenti e dei permessi
- Interazione con l'utente:
 - Interfacce grafiche
 - Linea di comando

Gestione dei processi

- Un processo è un programma in esecuzione
- A richiesta dell'utente o di un programma, il sistema operativo carica in memoria i programmi da un dispositivo di memorizzazione.
- Nei sistemi operativi moderni molti processi sono *contemporaneamente* (in realtà in *timesharing*) in esecuzione
- Il sistema operativo assegna e revoca ai processi la CPU, la memoria (memoria virtuale), i dispositivi di input-output

Stati dei processi

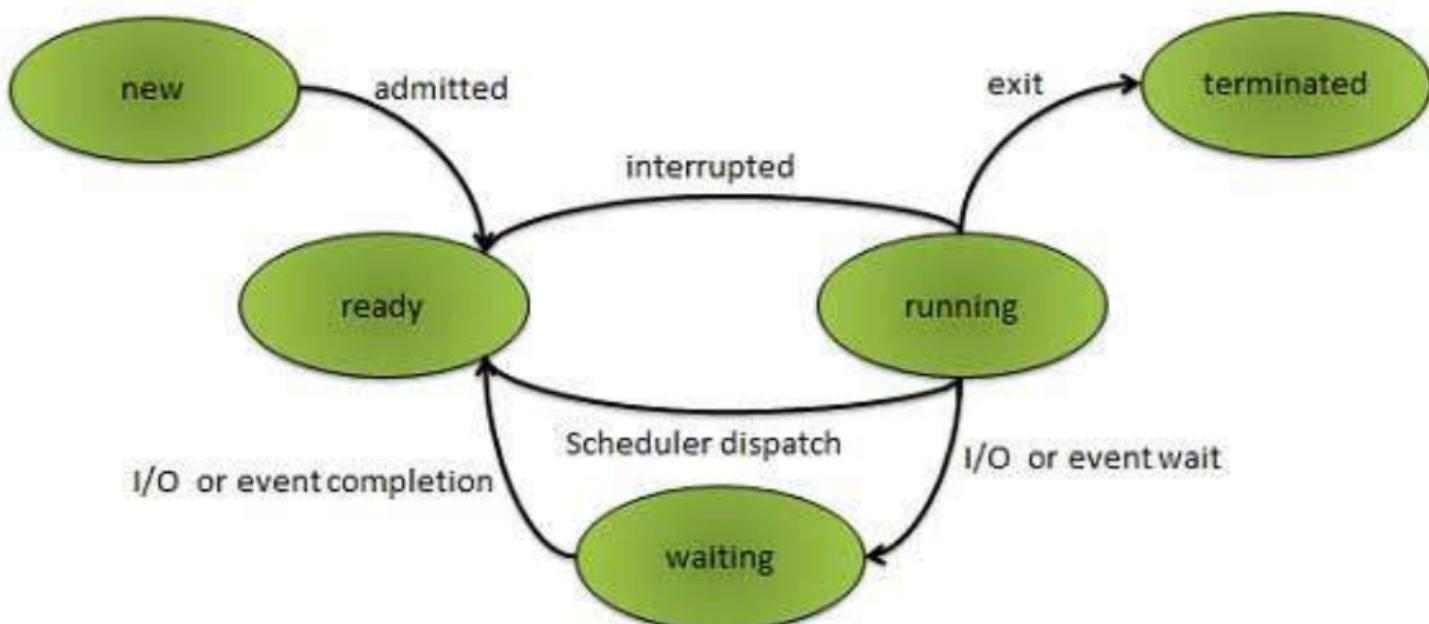
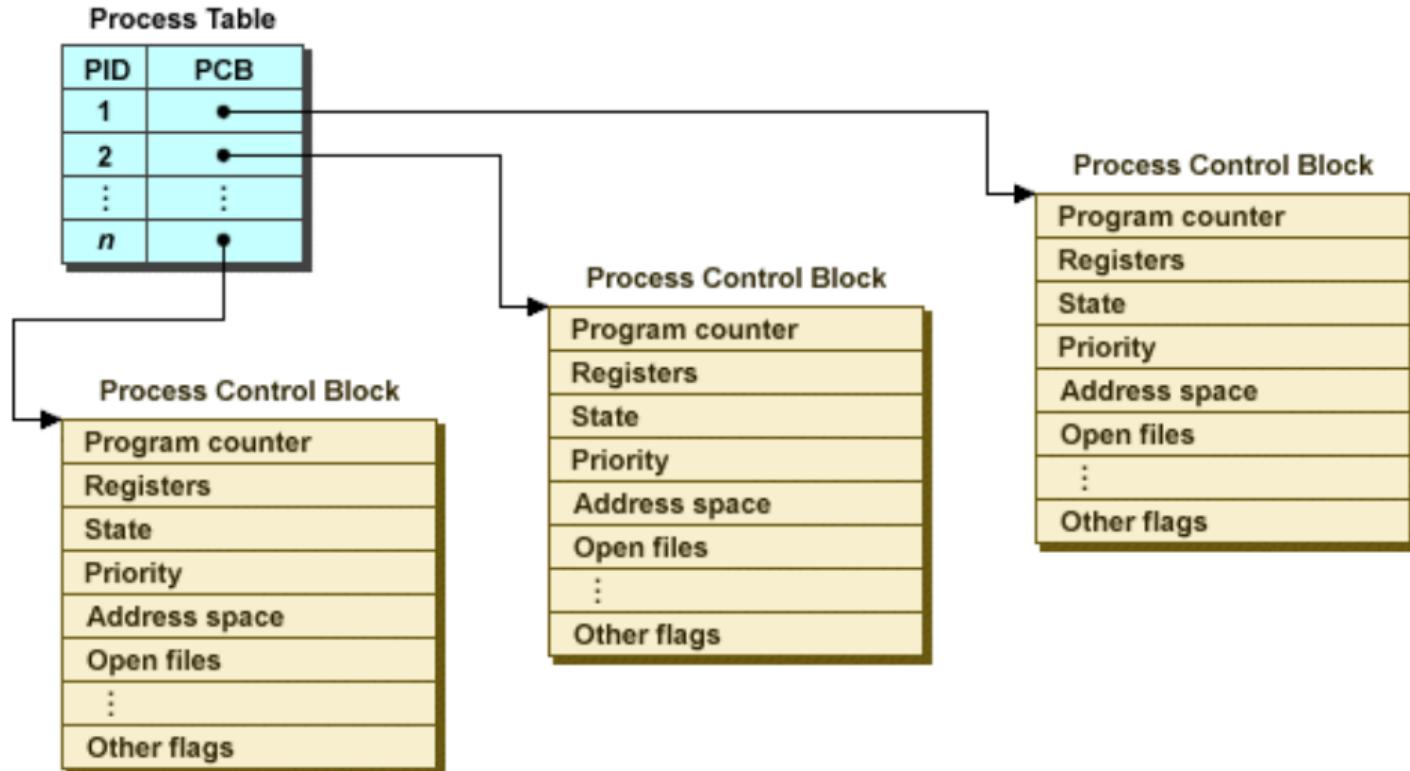
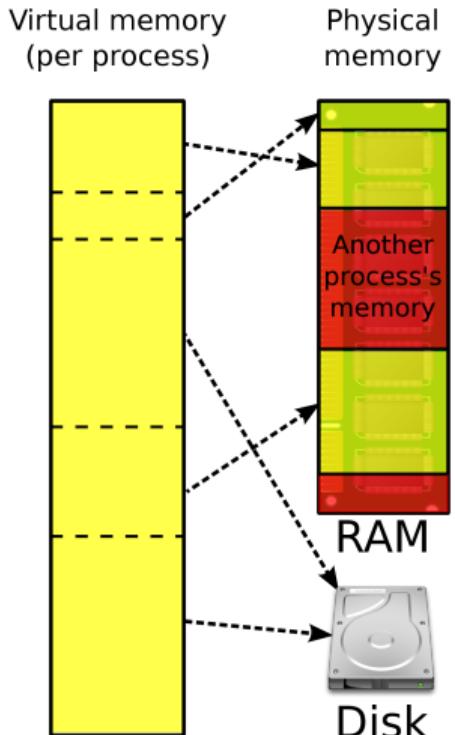


Tabella dei processi



Memoria virtuale



Gestione dell'hardware

- Il sistema operativo rende i dispositivi disponibili alle applicazioni attraverso un'**interfaccia**² uniforme
- Esempio interfaccia di stampa (quella che compare premendo CTRL-P o CMD-P): l'applicazione non deve chiedersi se stamperà su una stampante ad aghi, a getto, laser, convertitore a PDF, etc., perché è il sistema operativo a gestire la comunicazione con la stampante
- Le funzionalità sono realizzate attraverso programmi detti **driver** (generalmente forniti dal produttore del dispositivo)

²strumento di accesso, spesso semplificato, alle funzionalità di un sistema

File system

- Sarebbe complicato per i programmi gestire la memorizzazione su diversi supporti (diverse modalità di indirizzamento, memorizzazione non contigua, ...)
- File: unità di memorizzazione, a cui è assegnato un nome e una locazione logica (cartella)
- Esempi: programmi, documenti, fotografie
- Il file system è un'interfaccia uniforme verso i dispositivi di memorizzazione

Macchina astratta

- Complesso di hardware e software che fornisce determinate funzionalità.
- Utente vede la macchina astratta presentata (attraverso un'interfaccia) dal sistema operativo
- Programmatore vede la macchina astratta programmabile con il linguaggio utilizzato (che funge da interfaccia). Noi studieremo la macchina astratta definita dallo standard per il linguaggio di programmazione C.

UNIX e derivati

- UNIX, anni '70, nato e cresciuto in simbiosi con il linguaggio di programmazione C.
- Multi-utente, multi-programmato.
- Il nome indica il prodotto commerciale, ma anche complesso di funzionalità (numerose implementazioni).
- Standard POSIX
- Diverse famiglie di implementazioni (System5, BSD).
- Oggi liberamente disponibili: Linux-GNU, (Free-Open-Net)BSD.
- MacOSX ha kernel FreeBSD.
- Android è Linux adattato ai sistemi mobili.

DOS - Windows

- PC-DOS: sistema operativo fornito da Microsoft a IBM per il PC
- Mono-utente, mono-programmato.
- MS-DOS: compatibile con PC-DOS, fornito da Microsoft ai concorrenti di IBM per i cloni ("IBM-compatibili").
- Windows: interfaccia grafica per MS-DOS (anni '80), fino a Windows ME (2000)
- Windows NT: nuovo kernel (inizio anni '90), fino a versione 5 (Windows 2000)
- Le due versioni unificate in Windows XP (2001)