

INFORMATICA

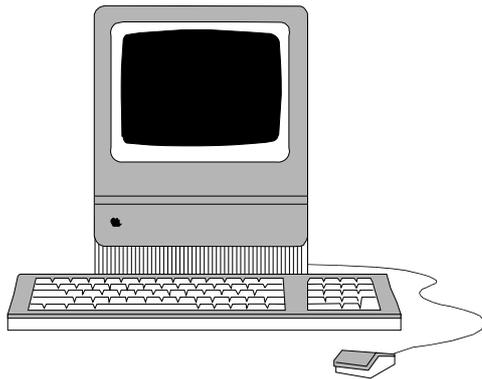
- Varie definizioni:
 - “Scienza degli elaboratori elettronici”
(*Computer Science*)
 - “Scienza dell’informazione”
- Definizione proposta:
 - ***Scienza della rappresentazione e dell’elaborazione dell’informazione***

L’informatica comprende:

- Metodi per la rappresentazione delle informazioni
- Metodi per la rappresentazione delle soluzioni
- Linguaggi di programmazione
- Architettura dei calcolatori
- Sistemi operativi
- Reti di calcolatori
- Calcolo numerico
- Algoritmi
-

ELABORATORE ELETTRONICO ("COMPUTER")

Strumento per la rappresentazione e l'elaborazione delle informazioni



L'ELABORATORE

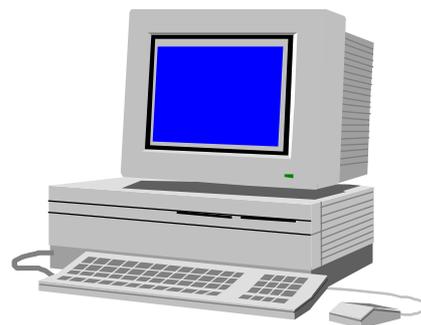
Componenti principali

- Unità centrale
- Video ("monitor")
- Tastiera e Mouse
- Lettore CD
- Dischi fissi ("hard disk")
- Dischetti ("floppy")

Componenti accessori

- Stampante
- Modem
- Scanner
- Tavolette grafiche

...



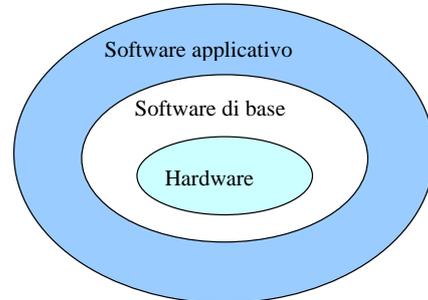
HARDWARE

SOFTWARE

Software: programmi che vengono eseguiti dal sistema.

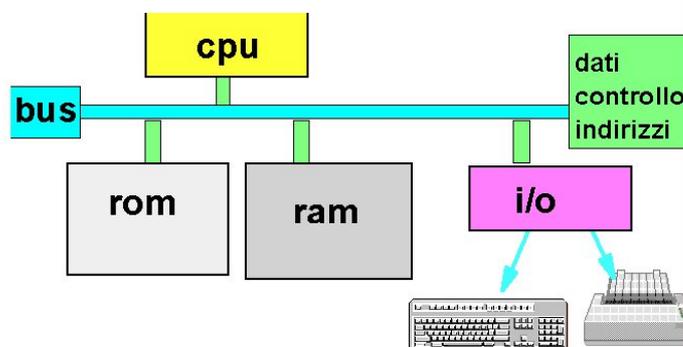
Distinzione fra:

- Software di base (es. Sistema Operativo)
- Software applicativo



HARDWARE

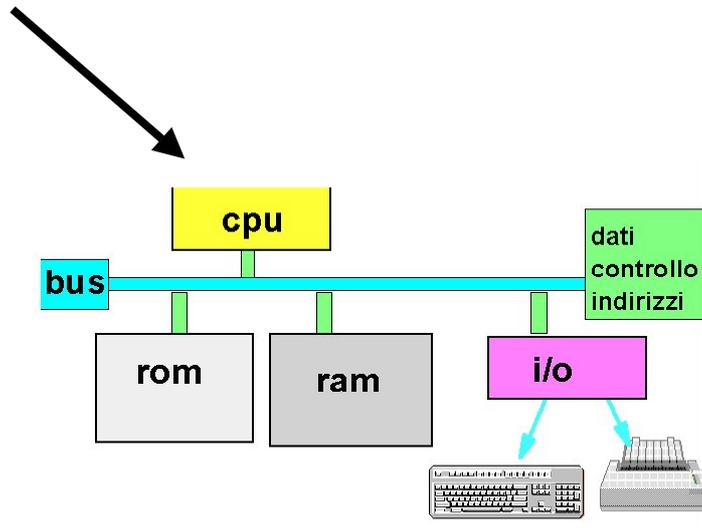
È composto da un insieme di *unità funzionali*



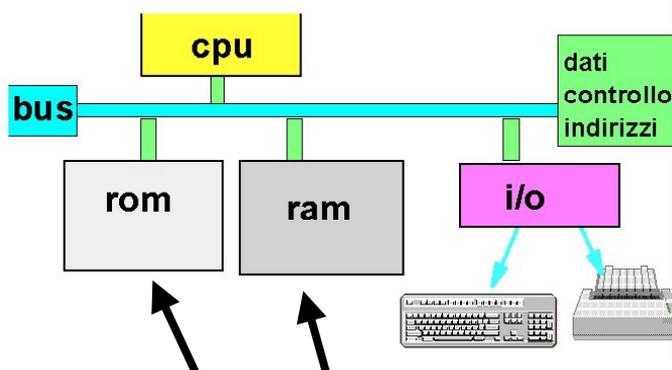
HARDWARE

CPU (Central Processing Unit), o Processore

CPU: Svolge le elaborazioni e il trasferimento dei dati, cioè *esegue i programmi*



HARDWARE



RAM & ROM

- Dimensioni relativamente limitate
- Accesso molto rapido

RAM (Random Access Memory), e
ROM (Read Only Memory)

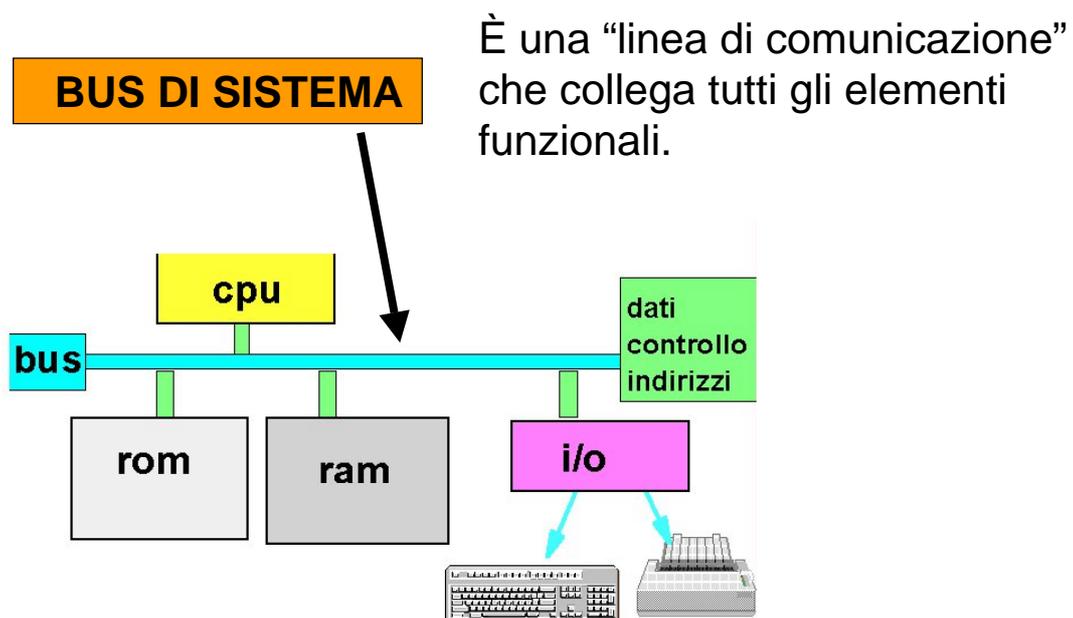
Insieme formano la **Memoria centrale**

HARDWARE

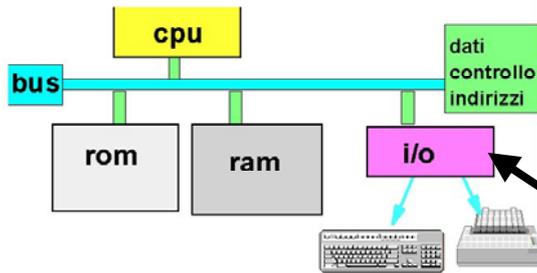
ATTENZIONE

- **RAM è volatile** (perde il suo contenuto quando si spegne il calcolatore)
 - usata per memorizzare dati e programmi
- **ROM è persistente** (mantiene il suo contenuto quando si spegne il calcolatore) ma il suo **contenuto è fisso e immutabile**
 - usata per memorizzare programmi di sistema

HARDWARE



HARDWARE

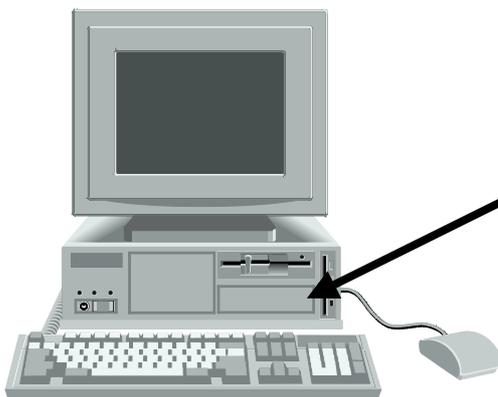


Sono usate per far comunicare il calcolatore con l'esterno (in particolare con l'utente)

UNITÀ DI INGRESSO / USCITA (I/O)

- Tastiera e Mouse
- Video e Stampante
- Scanner
- Tavoleta grafica
- **Dispositivi di memoria di massa**
- ...

HARDWARE



MEMORIA DI MASSA

- Dischi
- CD
- Nastri
- ...

- memorizza **grandi quantità** di informazioni
- **persistente** (le informazioni non si perdono spegnendo la macchina)
- **accesso molto meno rapido** della memoria centrale (**millisecondi** contro **nanosecondi** / differenza 10^6)

TECNOLOGIA DIGITALE

CPU, memoria centrale e dispositivi sono realizzati con **tecnologia elettronica digitale**

Dati ed operazioni vengono codificati a partire da due valori distinti di grandezze elettriche:

- tensione alta (V_H , 5V)
- tensione bassa (V_L , 0V)

A tali valori vengono convenzionalmente **associate le due cifre binarie 0 e 1:**

- **logica positiva:** $1 \leftrightarrow V_H$, $0 \leftrightarrow V_L$
- **logica negativa:** $0 \leftrightarrow V_H$, $1 \leftrightarrow V_L$

TECNOLOGIA DIGITALE (segue)

Dati ed operazioni vengono codificati tramite **sequenze di bit**

01000110101

CPU è in grado di operare soltanto in aritmetica binaria, effettuando operazioni *elementari* :

- somma e differenza
- scorrimento (shift)
- ...

Lavorando direttamente sull'hardware, l'**utente è forzato a esprimere i propri comandi al livello della macchina, tramite sequenze di bit**

IL SOFTWARE

Software:

insieme (complesso) di programmi.

Organizzazione a strati, ciascuno con funzionalità di livello più alto rispetto a quelli sottostanti

Concetto di
MACCHINA VIRTUALE



IL FIRMWARE

Firmware:

il confine fra hardware e software.

È uno strato di *micro-programmi*, scritti dai costruttori, che agiscono direttamente al di sopra dello strato hardware

Sono memorizzati su una speciale *memoria centrale permanente* (ROM, EPROM, ...)

IL SISTEMA OPERATIVO

Strato di programmi che opera *al di sopra di hardware e firmware* e **gestisce l'elaboratore**

Solitamente, è venduto insieme all'elaboratore

Spesso si può scegliere tra *diversi sistemi operativi* per lo stesso elaboratore, con diverse caratteristiche

Esempi:

- Windows 95/98/XP
- Windows NT/2000
- Linux
- MacOS X
- ...



FUNZIONI DEL SISTEMA OPERATIVO

Le funzioni messe a disposizione dal S.O. dipendono dalla complessità del sistema di elaborazione:

- gestione delle risorse disponibili
- gestione della memoria centrale
- organizzazione e gestione della memoria di massa
- interpretazione ed esecuzione di comandi elementari
- gestione di un sistema multi-utente

Un utente “vede” l'elaboratore solo tramite il Sistema Operativo

→ **il S.O. realizza una “macchina virtuale”**

FUNZIONI DEL SISTEMA OPERATIVO

Qualsiasi operazione di accesso a risorse della macchina implicitamente richiesta dal comando di utente viene esplicitata dal S.O.

Attraverso il S.O. il livello di interazione fra utente ed elaboratore viene elevato:

- senza S.O.: sequenze di bit
- con S.O.: comandi, programmi, dati

I sistemi operativi si sono evoluti nel corso degli ultimi anni (interfacce grafiche, Mac, Windows, ...)

ESEMPIO



e viceversa:



<u>Utente:</u>	<u>Sistema Operativo:</u>
"esegui progr1"	- input da tastiera - ricerca codice di "progr1" su disco - carica in memoria centrale codice e dati <elaborazione>
<u>Utente:</u>	<u>Sistema Operativo:</u>
"stampa 10"	- output su video

PROGRAMMI APPLICATIVI

Risolvono problemi specifici degli utenti:

- *word processor*: elaborazione di testi (*Es. MSWord*)
- *fogli elettronici*: gestione di tabelle, calcoli e grafici (*Es. Excel*)
- *database*: gestione di archivi (*Es. Access*)
- *suite* (integrati): collezione di applicativi capaci di funzionare in modo integrato come un'applicazione unica. (*Es. Office*)

- Sono scritti in **linguaggi di programmazione** di alto livello
- Risentono in misura ridotta delle caratteristiche della architettura dell'ambiente sottostante (*portabilità*)

AMBIENTI DI PROGRAMMAZIONE

È l'insieme dei programmi che consentono la scrittura, la verifica e l'esecuzione di nuovi programmi (*fasi di sviluppo*).

Sviluppo di un programma

- Affinché un programma scritto in un qualsiasi linguaggio di programmazione sia comprensibile (e quindi eseguibile) da un calcolatore, occorre **tradurlo** dal linguaggio originario al linguaggio della macchina
- Questa operazione viene normalmente svolta da speciali programmi, detti **traduttori**

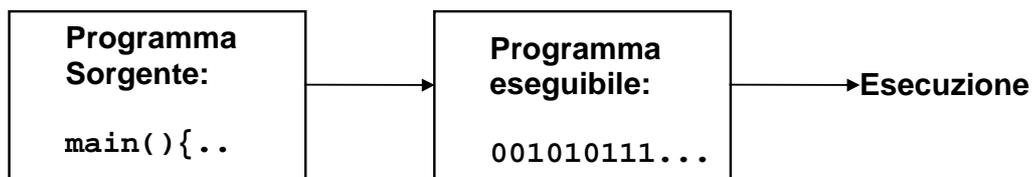
TRADUZIONE DI UN PROGRAMMA

PROGRAMMA	TRADUZIONE
main() { int A; ... A=A+1; if....	00100101 11001.. 1011100..

Il traduttore converte

- **il testo** di un programma scritto in un particolare linguaggio di programmazione (**sorgenti**)
- nella corrispondente **rappresentazione in linguaggio macchina** (programma **eseguibile**)

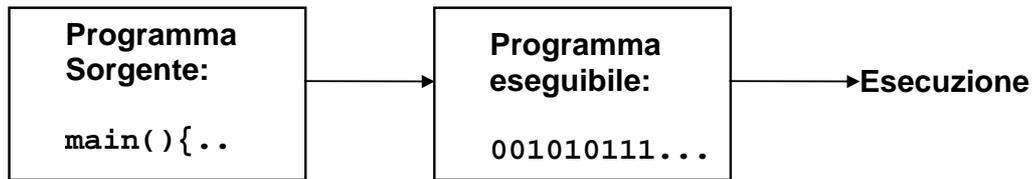
SVILUPPO DI PROGRAMMI



Due categorie di traduttori:

- i **compilatori** traducono l'intero programma (senza eseguirlo!) e producono in uscita il programma convertito in linguaggio macchina
- gli **interpreti** traducono ed eseguono immediatamente ogni singola istruzione del *programma sorgente*

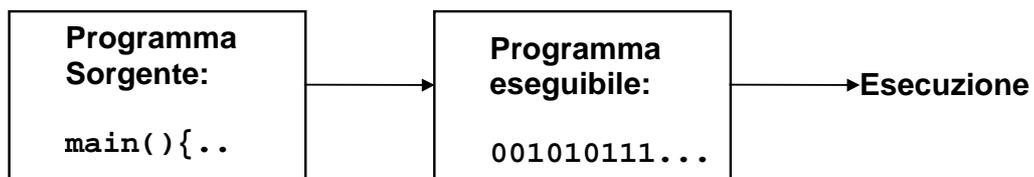
SVILUPPO DI PROGRAMMI (segue)



Quindi:

- nel caso del **compilatore**, lo schema precedente viene percorso **una volta sola** prima dell'esecuzione
- nel caso dell'**interprete**, lo schema viene invece attraversato **tante volte quante sono le istruzioni** che compongono il programma

SVILUPPO DI PROGRAMMI (segue)



L'esecuzione di un programma **compilato** è più **veloce** dell'esecuzione di un programma **interpretato**

AMBIENTI DI PROGRAMMAZIONE

COMPONENTI

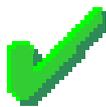
- **Editor**: serve per creare file che contengono **testi** (cioè sequenze di caratteri)
In particolare, l'editor **consente di scrivere il *programma sorgente***

E poi....

AMBIENTI DI PROGRAMMAZIONE

I° CASO: COMPILAZIONE

- **Compilatore**: opera la **traduzione di un *programma sorgente*** (scritto in un linguaggio ad alto livello) **in un *programma oggetto*** direttamente eseguibile dal calcolatore



PRIMA si traduce *tutto il programma*
POI si esegue la *versione tradotta*

AMBIENTI DI PROGRAMMAZIONE (2)

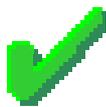
I° CASO: COMPILAZIONE (segue)

- **Linker:** (*collegatore*) nel caso in cui la costruzione del programma oggetto richieda l'unione di **più moduli** (compilati separatamente), il linker provvede a **collegarli** formando un unico *programma eseguibile*
- **Debugger:** (*“spulciatore”*) consente di **eseguire passo-passo** un programma, **controllando via via quel che succede**, al fine di **scoprire ed eliminare errori** non rilevati in fase di compilazione

AMBIENTI DI PROGRAMMAZIONE (3)

II° CASO: INTERPRETAZIONE

- **Interprete:** **traduce ed esegue** direttamente **ciascuna istruzione** del *programma sorgente*, **istruzione per istruzione**
È alternativo al compilatore (raramente sono presenti entrambi)



Traduzione ed esecuzione sono *intercalate*, e avvengono *istruzione per istruzione*