

# 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:

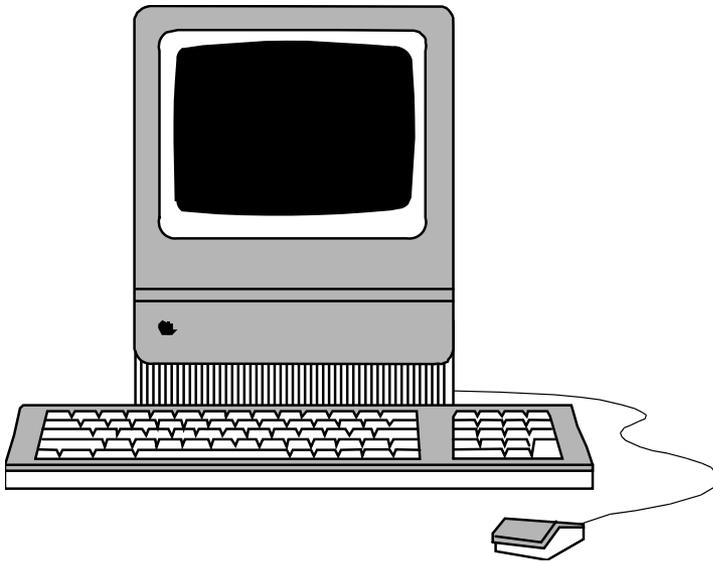
---

- 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
- ...

# 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/DVD
- Dischi fissi (“hard disk”)
- Dischetti (“floppy”)

## Componenti accessori

- Stampante
- Modem
- Scanner
- Tavolette grafiche
- Penne USB

...



**HARDWARE**

# 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.**

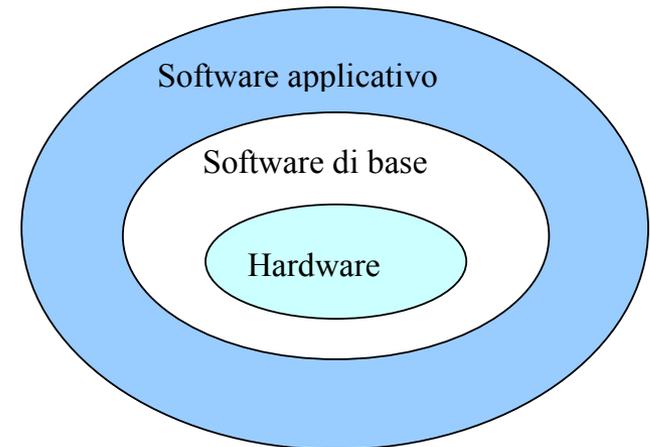
# SOFTWARE

---

**Software: programmi che**  
vengono eseguiti dal sistema.

**Distinzione fra:**

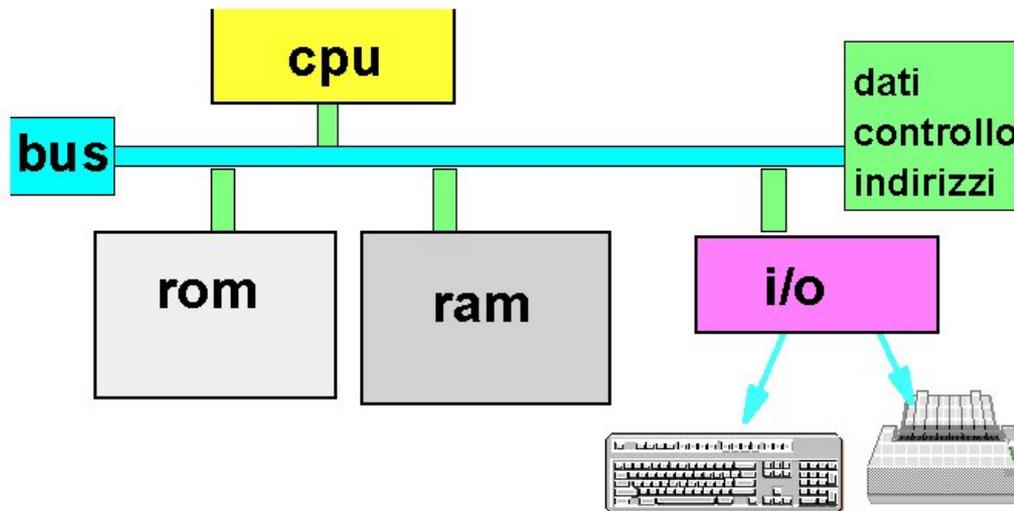
- Software di base (es. Sistema Operativo)
- Software applicativo



# HARDWARE

---

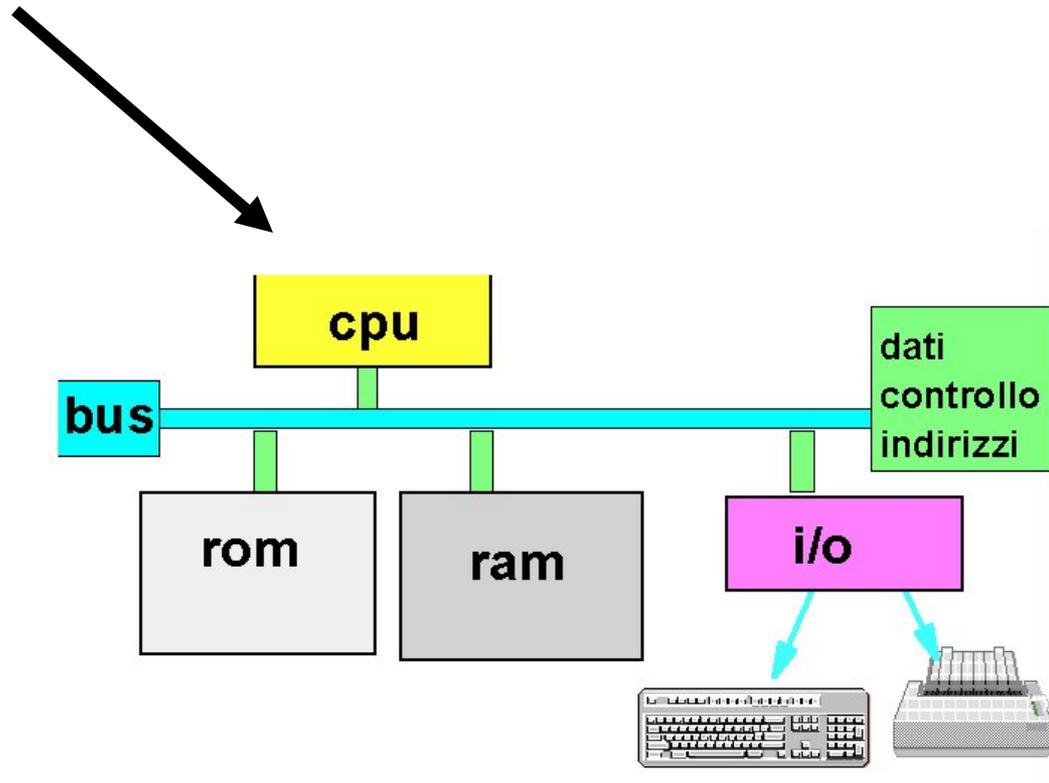
E' composto da un insieme di *unità funzionali*



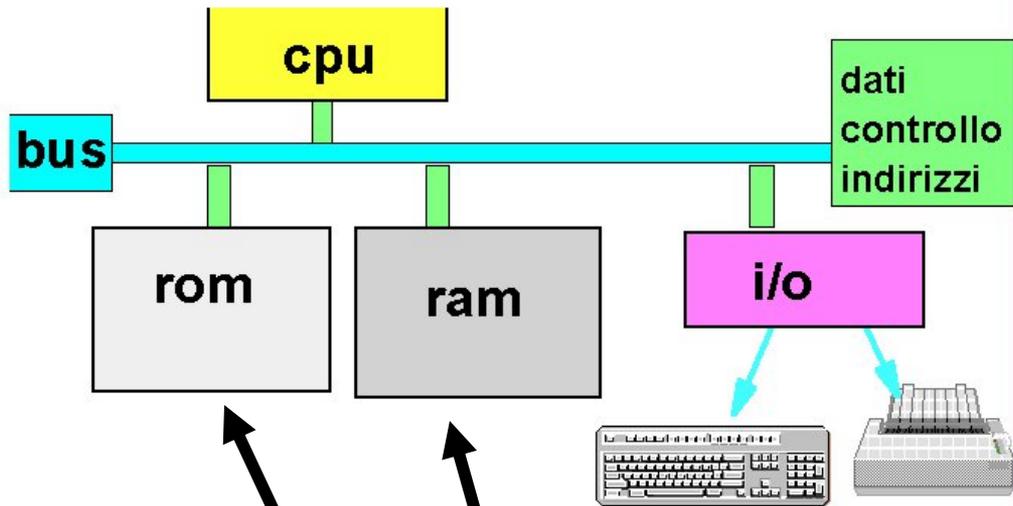
# HARDWARE

## 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

---

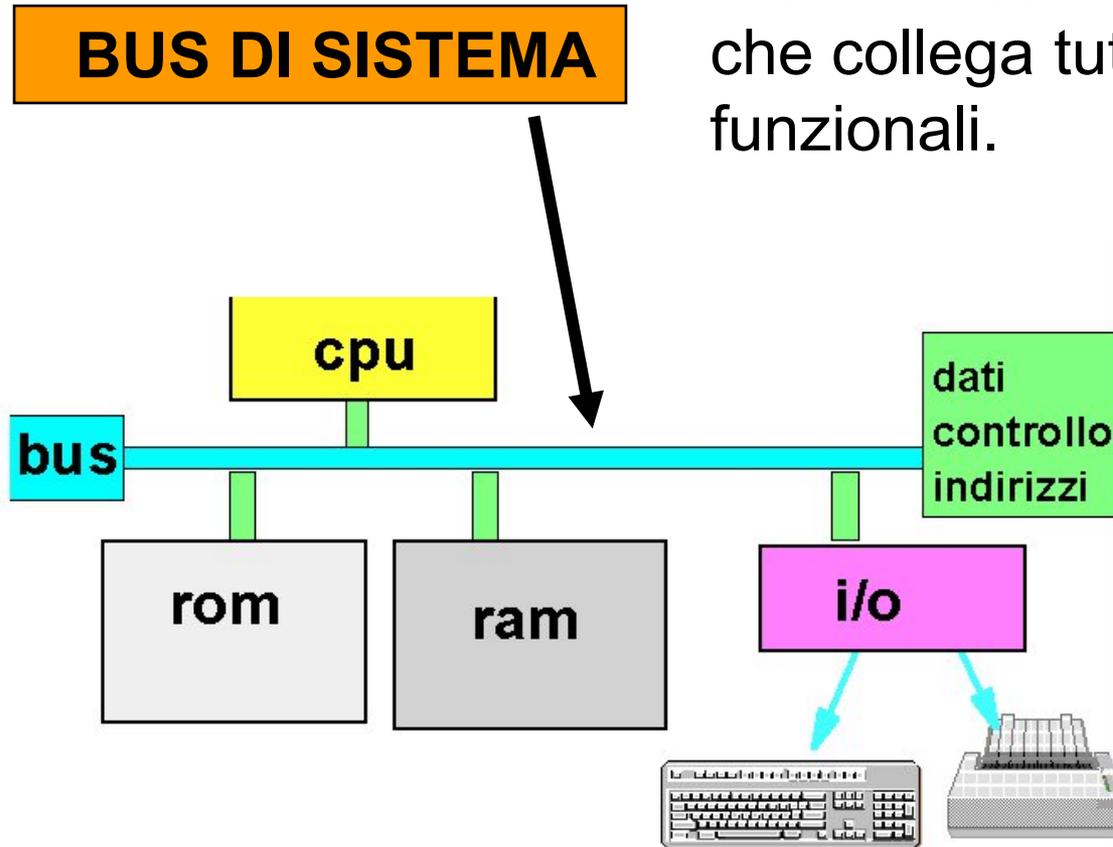
## ATTENZIONE

- **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

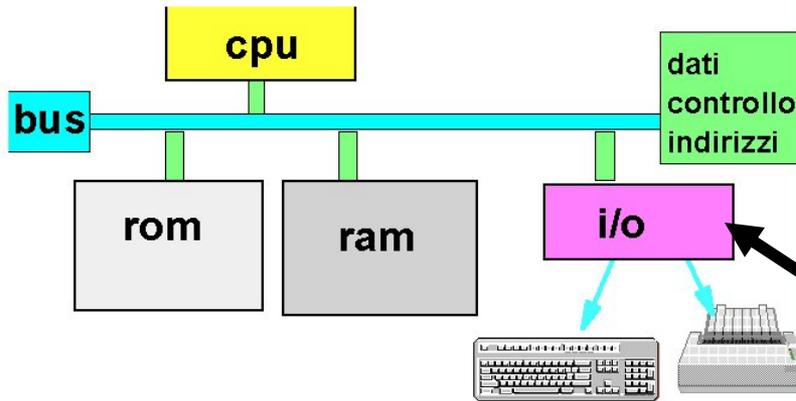
# HARDWARE

---

È una “linea di comunicazione”  
che collega tutti gli elementi  
funzionali.



# HARDWARE



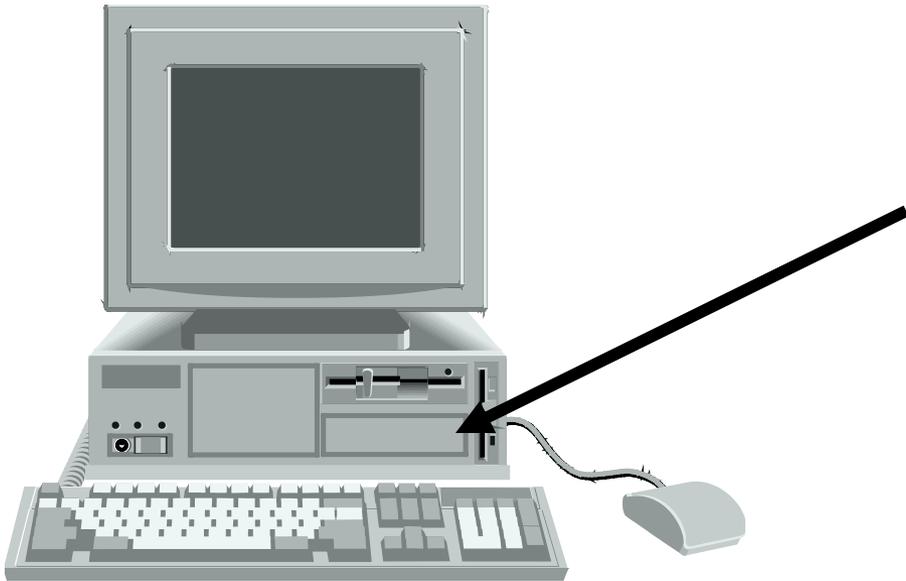
Sono usate per far comunicare il calcolatore con l'esterno (in particolare con l'utente)

## 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
- Penne USB

- 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  $10^6$ )

# LA MEMORIA DI MASSA

**Scopo:** memorizzare *grandi masse* di dati in modo *persistente*

(I dati memorizzati su questo tipo di memoria sopravvivono all'esecuzione dei programmi)

## Caratteristiche:

- *tempo di accesso*
- *capacità*

## Byte (e multipli)

- Kbyte (1.024 Byte)
- Mbyte (1.048.576 Byte)
- Gbyte (1.073.741.824 Byte)

## Tempo di accesso

- disco fisso: ~10 ms
- floppy: ~100 ms
- flash: 50ns (r) – 1ms (w)

## Capacità

- disco fisso: >10 GB
- Flash: 1-16 GB
- floppy: 1.4 MB

# DISPOSITIVI di memoria di massa

---

## DUE CLASSI FONDAMENTALI:

- **ad accesso sequenziale** (ad esempio, **NASTRI**):  
per recuperare un dato è necessario accedere prima a tutti quelli che lo precedono sul dispositivo
- **ad accesso diretto** (**DISCHI/FLASH**):  
si può recuperare direttamente un qualunque dato memorizzato

# DISPOSITIVI MAGNETICI

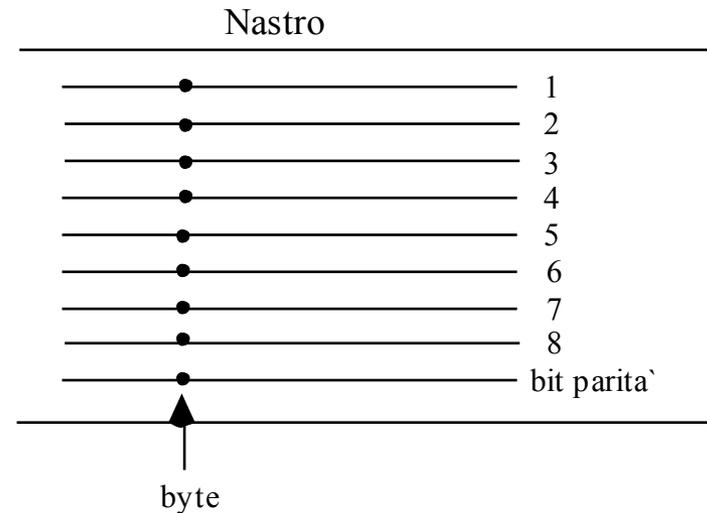
---

- L'area del dispositivo è suddivisa in **micro-zone**
- Ogni micro-zona memorizza una **informazione elementare** sotto forma di ***stato di magnetizzazione***:
  - area magnetizzata / area non magnetizzata**
- Ai due possibili stati di magnetizzazione vengono **associate le due cifre binarie 0 e 1**
  - bit (Binary digit)**
- Quindi, **ogni micro-zona memorizza 1 bit**
- Per memorizzare informazioni più complesse si considerano *collezioni di bit*:
  - BYTE** (collezione di **8 bit**) e suoi multipli

# NASTRI MAGNETICI

Nastri di materiale magnetizzabile arrotolati su supporti circolari, o in cassette.

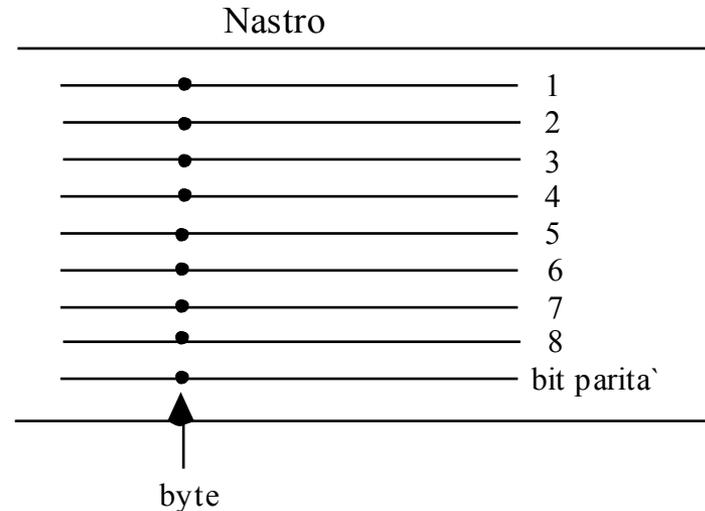
Sul nastro sono tracciate delle **piste orizzontali parallele** (di solito 9, di cui 8 corrispondono ad un byte e la nona è il bit di parità).



*Bit di parità': rende pari il numero di 1 contenuti nelle piste orizzontali. Serve per il controllo di eventuali errori di memorizzazione.*

# NASTRI MAGNETICI (segue)

I dati sul nastro sono organizzati in zone contigue dette **record**, separate da zone prive di informazione (*inter-record gap*).



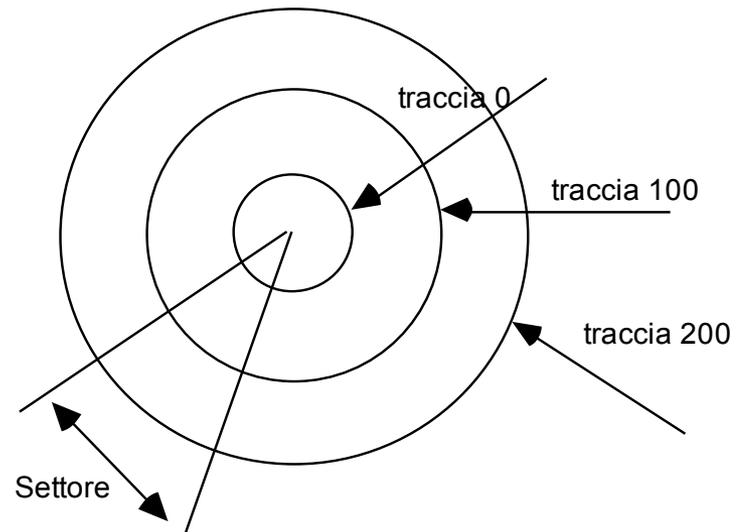
- Tutte le **elaborazioni** sono **sequenziali**:  
le operazioni su uno specifico record sono **lente**
- Oggi servono solo per mantenere copie di riserva (**backup**) dei dati

# DISCHI MAGNETICI

---

Un disco consiste in un certo numero di **piatti** con **due superfici** che ruotano attorno ad un perno centrale.

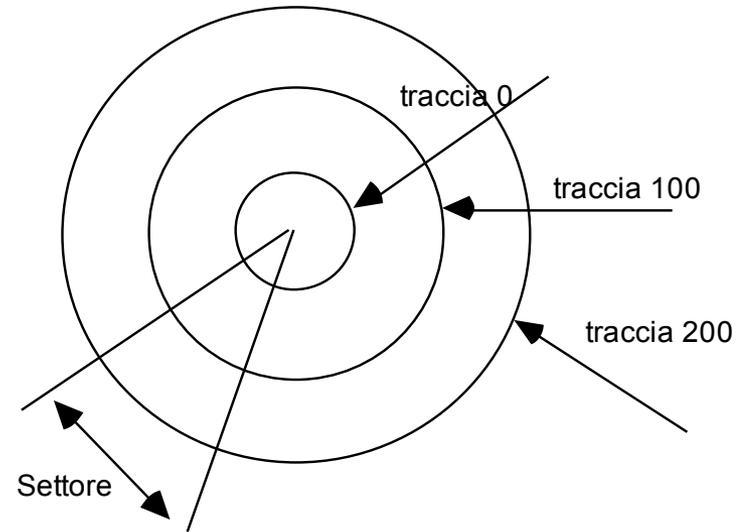
Ogni superficie dispone di una propria **testina di lettura / scrittura**.



Le superfici sono organizzate in **cerchi concentrici (tracce)** e in **spicchi di uguale grandezza (settori)**.  
Le tracce equidistanti dal centro formano un **cilindro**.

# DISCHI MAGNETICI (segue)

I dati sono scritti in posizioni successive **lungo le tracce**: ogni bit corrisponde a uno stato di *magnetizzazione* del materiale magnetico della superficie del disco.



Ogni **blocco** del disco è identificato con la terna  
**⟨superficie, traccia, settore⟩**

Per effettuare il trasferimento dei dati in memoria centrale occorre disporre di un'area di memoria (*buffer*) di dimensioni pari al blocco.

# DISCHI MAGNETICI (segue)

---

## Ingresso (uscita) da (verso)

⟨*superficie, traccia, settore*⟩

- 1) spostamento della testina (seek) verso la traccia richiesta
- 2) attesa che il settore arrivi sotto la testina;
- 3) trasferimento dei dati in / da memoria centrale (solitamente eseguito da un processore dedicato - Direct Memory Access, DMA).

## Tempo di accesso:

$$T_{i/o} = T_{seek} + 1/2 T_{rotazione} + T_{trasferimento}$$

( $T_{seek}$  è il più lungo)

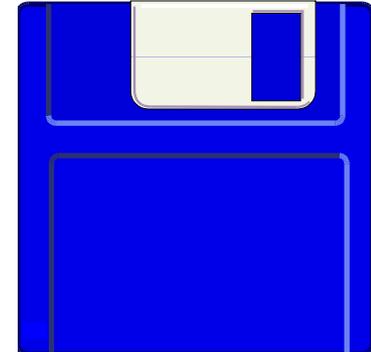
# DISCHETTI (FLOPPY)

---

Sono dischi magnetici di ***piccola capacità***, portatili, usati per trasferire informazioni tra computer diversi.

Sono costituiti da un **unico disco** con due superfici.

Sopravvivono solo quelli da 3.5" di diametro (1.4 MB)



**IMPORTANTE:** per poter essere usati, i dischetti devono prima essere ***suddivisi in tracce e settori*** dal Sistema Operativo → **FORMATTAZIONE**  
Operazione lunga, noiosa → **dischetti già formattati**

# Penne USB

---

Sono dispositivi che contengono memorie flash.

Organizzate strutturalmente come una RAM ma sono persistenti e il computer le vede come un disco

Sono memorie altamente asimmetriche:

- tempo di accesso in lettura molto basso dell'ordine di nanosecondi
- tempo di accesso in scrittura dell'ordine dei millisecondi (cancellazione e riscrittura)

# DISPOSITIVI OTTICI

---

## **1984, CD-ROM (Compact-Disk Read-Only Memory)**

- Capacità: > 600 MB
- Costo: < \$1
- Velocità di trasferimento:
  - originariamente 150 KB / s ( “1X” )
  - oggi 24, 32, 40 volte tanto...

## **1984, WORM (Write Once Read Many)**

- Sono dischi ottici scrivibili (una sola volta)
- Parenti stretti dei CD audio (CD-DA, 1982)
- Accesso diretto ai settori (capacità 2.048 KB)

# DISPOSITIVI OTTICI (segue)

---

## 1986, CD - I (Compact-Disk Interactive)

- Per memorizzare immagini, filmati, grafica, suono, testi e dati (*multimedialità*).

Ormai il CD è il principale mezzo per lo scambio di grandi quantità di informazioni

- installazione di nuovi programmi di utilità
- archiviazione di immagini, suoni, opere multimediali
- copie di riserva (backup)
- distribuzione di materiale pubblicitario o “di prova”

**Affidabilità: fino a 10-15 anni.**

# DISPOSITIVI OTTICI - Il presente

---

## **1997, DVD (Digital Versatile Disk)**

- Evoluzione del CD-ROM
- Capacità da 4.8 GB (Single Layer-Single Side) a 17 GB circa (Double Layer - Double Side)
- Velocità di trasferimento molto elevata

Adatto per film e opere multimediali.

## **2002, “HD-DVD” e “Blue-ray Disc”**

Evoluzione del DVD: consorzio sviluppatore composto da produttori di Consumer Electronics, e dai produttori di contenuti (principalmente, “Hollywood Majors”).

- Capacità da 25 GB a 50 GB
- Velocità di trasferimento molto elevata

Nel Febbraio 2008 la “*guerra dei formati*” termina, sancendo la vittoria dello standard Blue-ray...

# CAPACITÀ DELLE MEMORIE

---

<b>Tipo di memoria</b>	<b>Capacità</b>
Memoria centrale	512 Mbyte - 4 GByte
Dischi magnetici	120 GByte - 1 TByte
Dischi floppy	1.4 MByte
Penne USB	1 – 16 GByte
Dischi ottici	650 MByte - 50 GByte

# PERSONAL COMPUTER

---

## PC (ex “IBM-COMPATIBILI”)

Usano processori della famiglia *Intel 80x86*:

- 8086
- 80286
- ...
- Pentium
- Pentium MMX
- Pentium II
- Pentium III
- Pentium IV
- Dual Core/64 Bit
- ...



### **Le prestazioni** dipendono da:

- frequenza dell'orologio di sistema (*clock*)
- dimensione della RAM
- velocità/parallelismo delle linee dati/comandi (bus)

# ALTRI SISTEMI DI CALCOLO

---

## **Workstation**

sistemi con capacità di supportare più attività contemporanee, spesso dedicati a più utenti. Prestazioni normalmente superiori a quello di un tipico Personal Computer.

## **Mini-calcolatori**

Macchine capaci di servire decine di utenti contemporaneamente, collegati tramite terminali

## **Super-calcolatori**

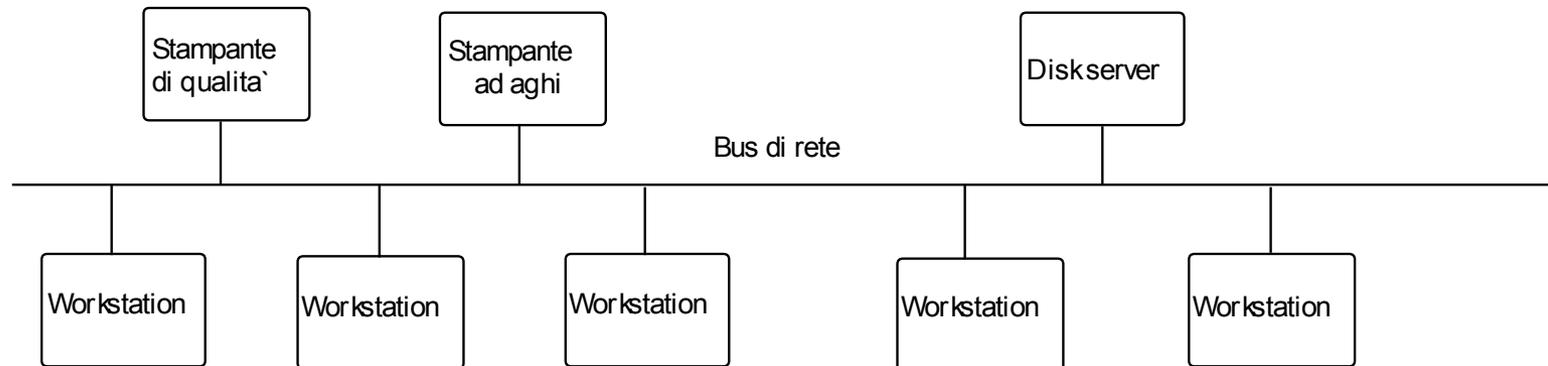
Hanno molti processori, grandi memorie di massa e servono tipicamente centinaia o migliaia di terminali

# RETI DI CALCOLATORI

---

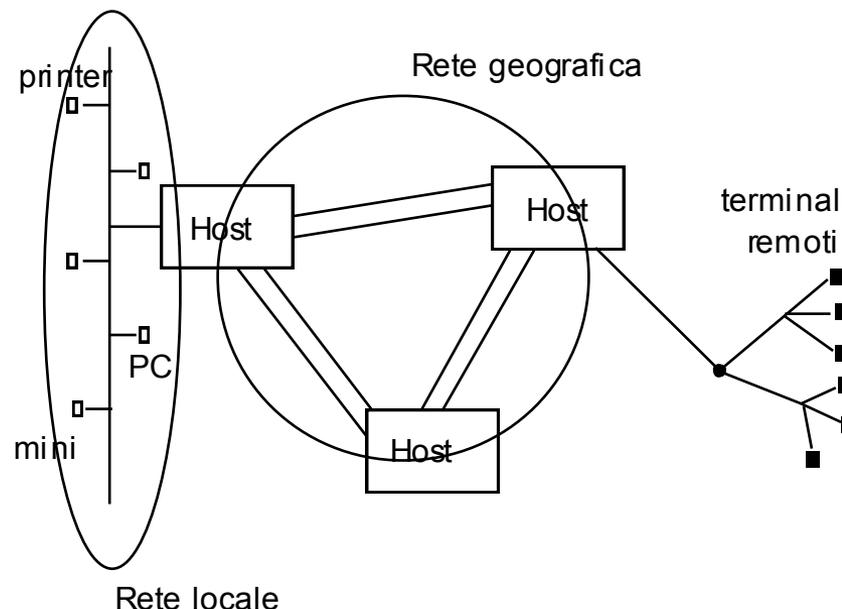
- **Reti Locali:**  
connettono elaboratori *fisicamente vicini*  
(nello stesso ufficio o stabilimento).

## LAN (Local Area Network)



# RETI DI CALCOLATORI (segue)

- **Reti geografiche:** collegano elaboratori medio-grandi situati anche **a grande distanza**.



**WAN (Wide Area Network)**

# INTERNET: la rete delle reti

---

- **Internet:** la rete risultante dalla interconnessione mondiale di tutte le reti.
- Milioni di elaboratori (“**siti**”) collegati a **ragnatela**



**World-Wide Web (WWW)**

# 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
- Windows NT/XP/Vista
- Linux
- Mac OS 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

---

## **Conseguenza:**

diversi S.O. possono realizzare *diverse macchine virtuali* **sullo stesso elaboratore fisico**

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, Macintosh, Windows, ...)

# RUOLO DEL SISTEMA OPERATIVO

**Il S.O. traduce le richieste dell'utente in opportune sequenze di istruzioni, a loro volta trasformate in valori e impulsi elettrici per la macchina fisica.**



e viceversa:



# RUOLO DEL SISTEMA OPERATIVO

---

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

## **Esempi:**

- accesso a memoria centrale
- accesso ai dischi
- I/O verso video, tastiera, ...

# ESEMPIO



e viceversa:



Utente:

“esegui progr1”

Sistema Operativo:

- input da tastiera
- ricerca codice di “progr1” su disco
- carica in memoria centrale codice e dati
- <elaborazione>

Utente:

“stampa 10”

Sistema Operativo:

- output su video

# CLASSIFICAZIONE dei S.O.

---

**In base al numero di utenti:**

- **Mono-utente (*mono-user*):** un solo utente alla volta può utilizzare il sistema
- **Multi-utente (*multi-user*):** più utenti possono interagire contemporaneamente con la macchina.

Nel caso di più utenti contemporanei, **il Sistema Operativo deve fornire a ciascuno l'astrazione di un sistema “dedicato”**.

# CLASSIFICAZIONE dei S.O.

---

In base al numero di programmi in esecuzione:

- **Mono-programmato (*mono-task*):** si può eseguire *un solo programma* per volta
- **Multi-programmato (*multi-task*):** il S.O. è in grado di portare avanti contemporaneamente l'esecuzione di più programmi (pur usando una sola CPU).

Nel caso di multi-programmazione **il S.O. deve gestire la suddivisione del tempo** della CPU fra i vari programmi.

# CLASSIFICAZIONE dei S.O.

---

## Esempi:

- **MS-DOS:** monoutente, monoprogrammato
- **Windows95/98:** monoutente, multiprogrammato
- **Windows NT/XP:** multiutente, multiprogrammato
- **UNIX (linux):** multiutente, multiprogrammato

# 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 (*fasì 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

---

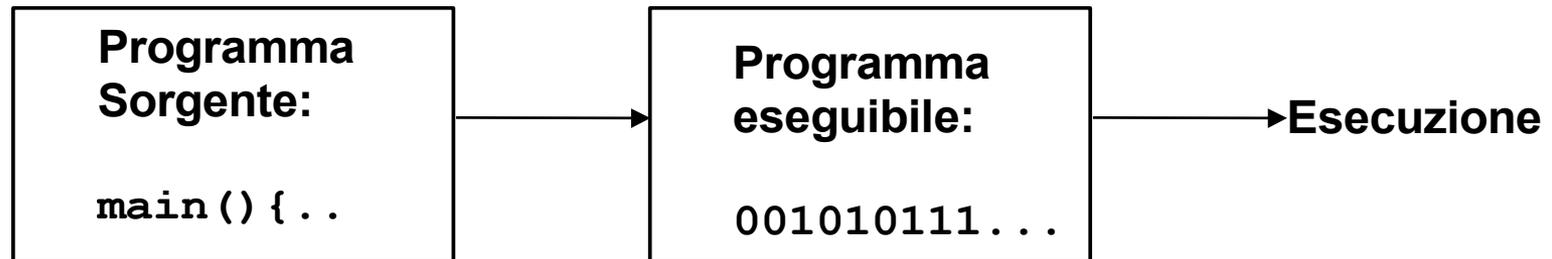
PROGRAMMA	TRADUZIONE
<code>main()</code>	
<code>{ int A;</code>	<code>00100101</code>
<code>...</code>	
<code>A=A+1;</code>	<code>11001..</code>
<code>if....</code>	<code>1011100..</code>

## 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

---

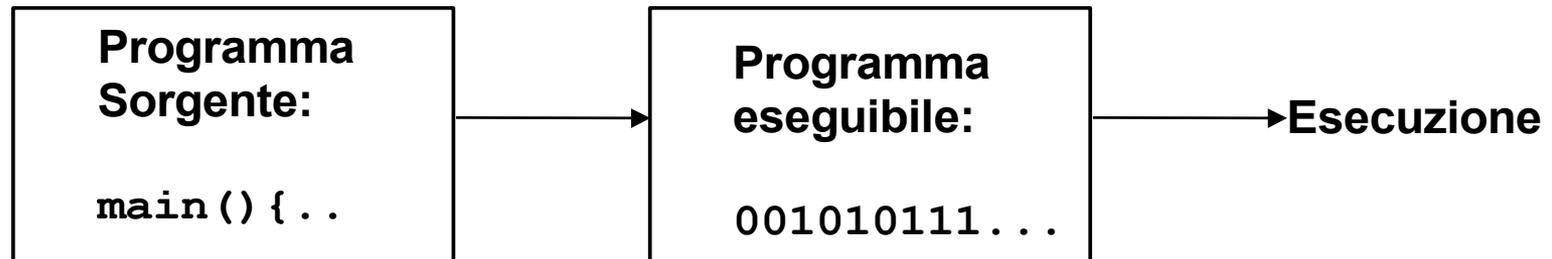


## Due categorie di traduttori:

- 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)

---

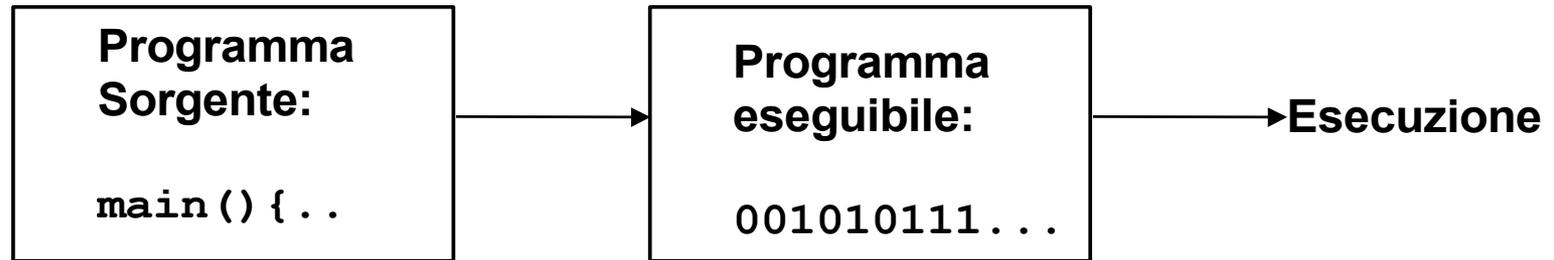


Quindi:

- 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 **ve-**  
**loce** 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***.

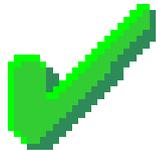
E poi....

# 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)

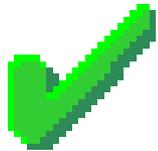
- **Linker:** (*collegatore*) nel caso in cui la costruzione del programma oggetto richieda l'unione di **più moduli** (compilati separatamente), il linker provvede a **collegarli** formando un unico *programma eseguibile*.
- **Debugger:** (“*spulciatore*”) consente di **eseguire passo-passo** un programma, **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*.**