



**Università degli Studi di Parma**

**Facoltà di Scienze MM.FF.NN. – Corso di Laurea in Informatica**

---

# **Nozioni di Base di Informatica**

**Roberto Alfieri**

**Giulio Destri**

# Hardware e Software

---

- Hardware: la parte "fisica" del calcolatore
- Software: l'insieme di programmi che un calcolatore richiede per funzionare

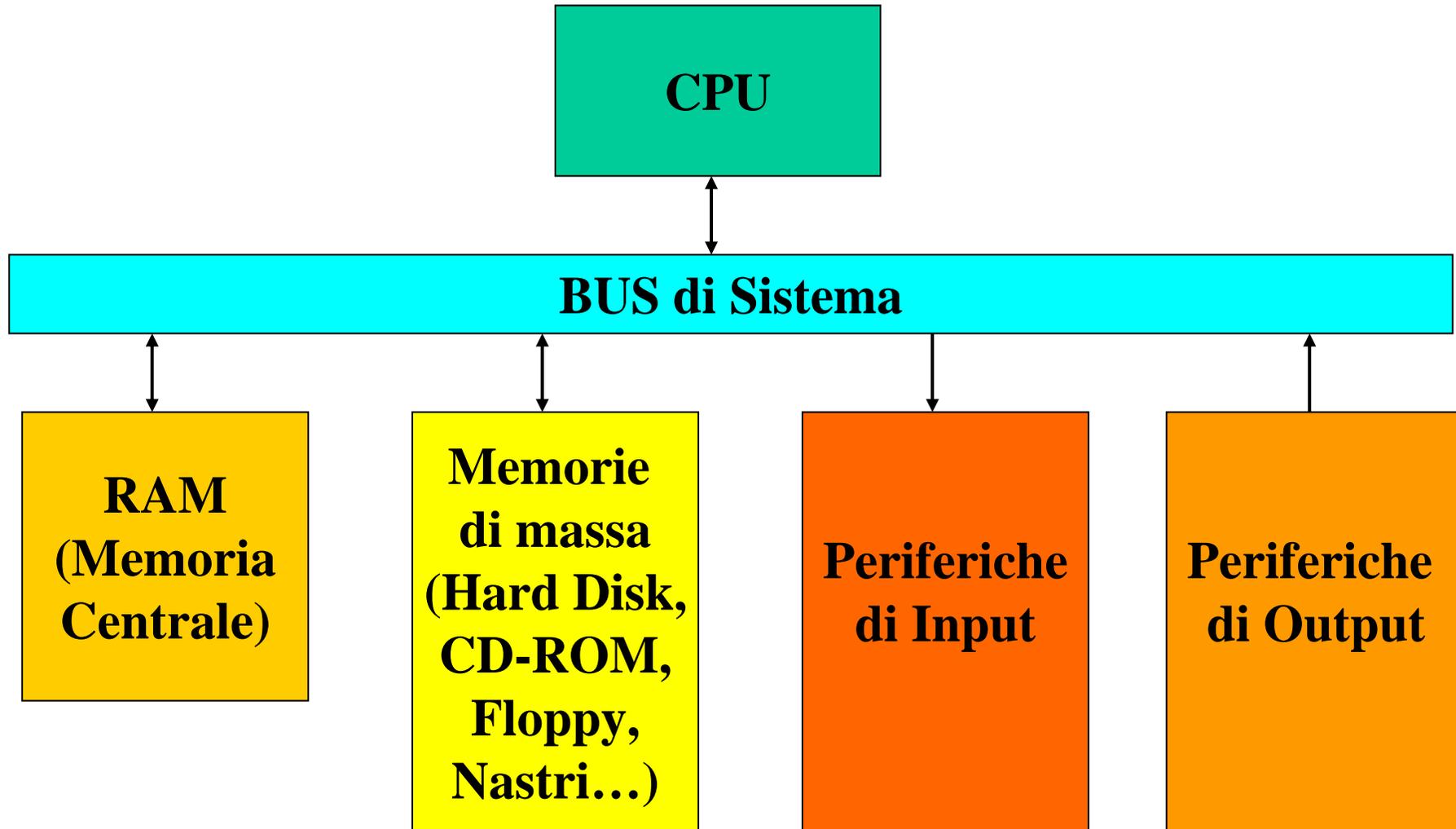
# La struttura del Computer

---

- Unità Centrale di Elaborazione (CPU)
- Bus di collegamento
- Memoria RAM e ROM
- Memorie di Massa
- Dispositivi di I/O
- Porte
- Rete
- Periferiche

# La struttura del Computer

---



# Le periferiche

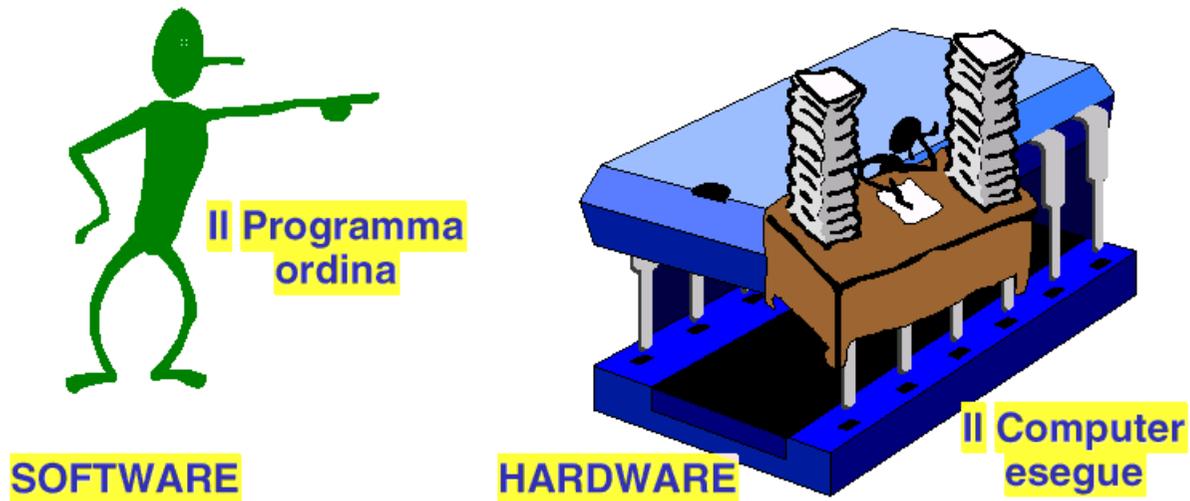
---

- Tastiera, tavoletta grafica, touchscreen
- Mouse, penna ottica, cloche, Joystick, touchpad
- Monitor, proiettore, uscita audio
- Stampante
- Plotter
- Modem
- Scheda di Rete
- Scanner, telecamera, microfono

# La CPU

---

L'unità centrale è progettata per eseguire i programmi.



# La memoria centrale

---

- Insieme di celle di uguale capacità
- Contiene numeri binari
- Ogni cella ha un **indirizzo** ben preciso
- Il tempo di accesso è finito e uguale per ogni cella
- I numeri contenuti possono corrispondere a istruzioni o dati
- E' volatile (si cancella allo spegnimento della macchina)

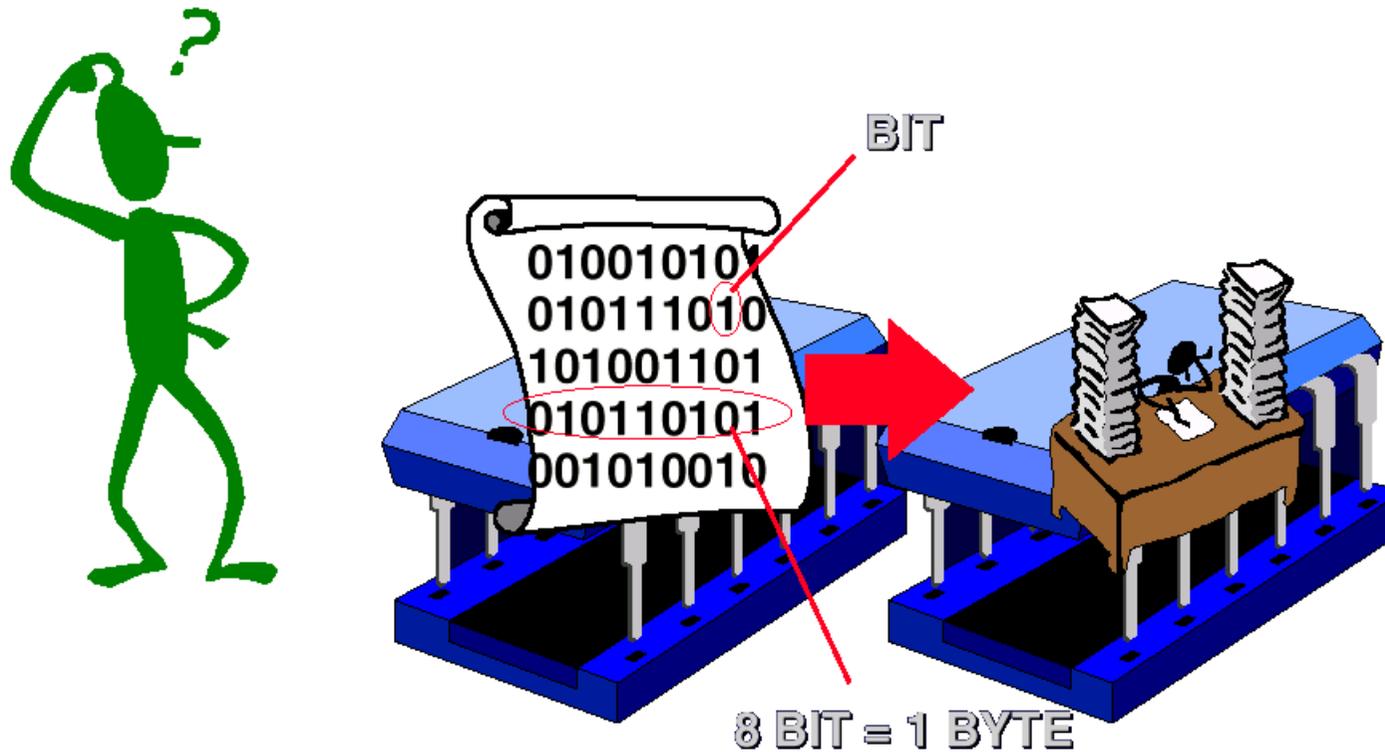
# Memorie di massa

---

- Floppy disk
- Hard disk
- CD-ROM, DVD-ROM, CD-ROM-WR
- Disk Array (RAID)
- Nastro, unità magneto-ottica, IomegaZIP
- Schede
- E' permanente

# Le informazioni entro il Computer

---



# Rappresentazione di informazioni

---

- Codice ASCII: insieme di simboli, rappresentati dai numeri compresi fra 0 e 255 e comprendenti anche le cifre e le lettere dell'alfabeto
- Codice EBCDIC
- UNICODE: evoluzione di ASCII, con codici da 0 a 65536

# Tipi di Interfaccia Utente

---

- Interfaccia utente a riga di comando
- Interfaccia utente a menu
- Combinazione delle due precedenti
- Interfaccia grafica

# Programma Applicativo

---

- Un programma eseguibile è contenuto in un file (o in più file)
- Quando un programma viene invocato il **loader** carica in una zona della memoria centrale l'insieme delle istruzioni contenute nel (nei) file
- Il programma entra poi in esecuzione ed ottiene il controllo della CPU
- Il programma in esecuzione, con i suoi dati, le sue istruzioni ed il suo stato corrente si chiama **processo**

# Il Sistema Operativo

---

- E' formato da un nucleo (kernel) e da un insieme di programmi, moduli e librerie
- Gestisce le risorse di basso livello e fornisce una visione indipendente dal supporto fisico
- Rende utilizzabile l'elaboratore per l'utente ed i programmi applicativi
- Coordina il funzionamento dei programmi applicativi
- Definisce e gestisce l'interfaccia utente

# Il Sistema Operativo: funzioni

---

- Gestione della memoria centrale
- Gestione della memoria di massa (file system)
- Gestione dei programmi in esecuzione (processi)
- Gestione dell'Input e Output
- Chiamate di funzioni comuni per i programmi applicativi

# Multitasking

---

- Capacità di caricare in memoria più programmi applicativi      capacità di eseguire più programmi applicativi “contemporaneamente”
- Multitasking reale (time-sharing): tempo CPU suddiviso fra le applicazioni (es. WindowsNT, UNIX)
- Multitasking virtuale (cooperativo): una sola applicazione per volta è attiva e “gira”, le altre sono sospese (es. Windows 3.x)

# Multithreading

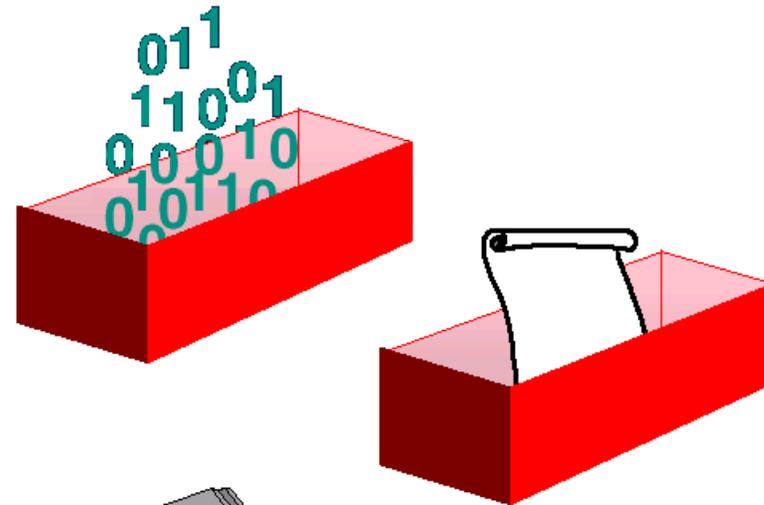
---

- Capacità di avere più flussi di esecuzione entro un programma in esecuzione → capacità di eseguire diverse istruzioni “contemporaneamente” entro uno stesso processo
- Il programma risulta suddiviso in diversi blocchi di istruzioni che agiscono in modo concorrente (thread)
- Il multithreading è garantito dal sistema di esecuzione dei programmi

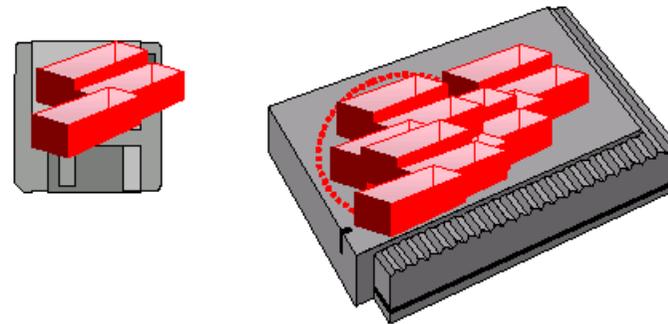
# I File

---

**Sono contenitori di dati o programmi**



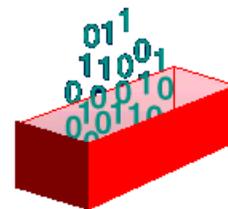
**Sono memorizzati su disco**



# Formato dei File

---

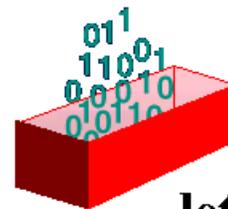
Ogni programma "usa"  
diversamente i bit ed  
ha un proprio formato di  
rappresentazione dei dati



lettera.txt



Edit



lettera.doc

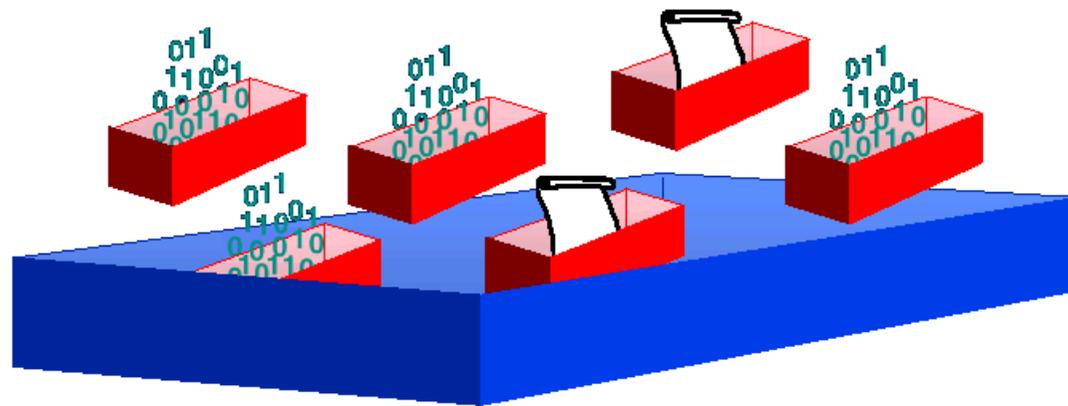


Word

# Le Directory o Cartelle

---

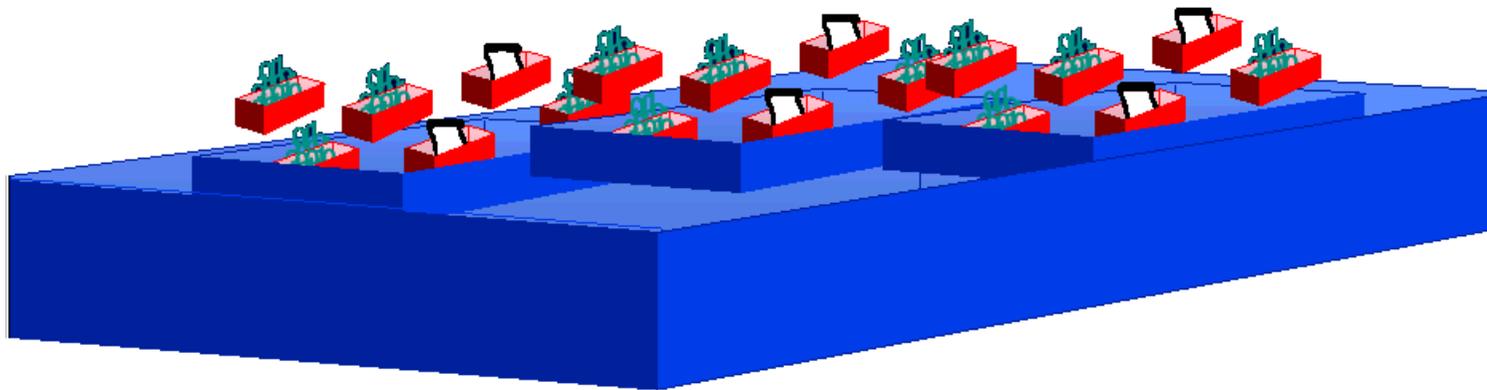
**I file possono essere raggruppati  
in elenchi, chiamati "directory"**



# Le sottodirectory o sottocartelle

---

**Una directory può contenere altre directory, dette sottodirectory**

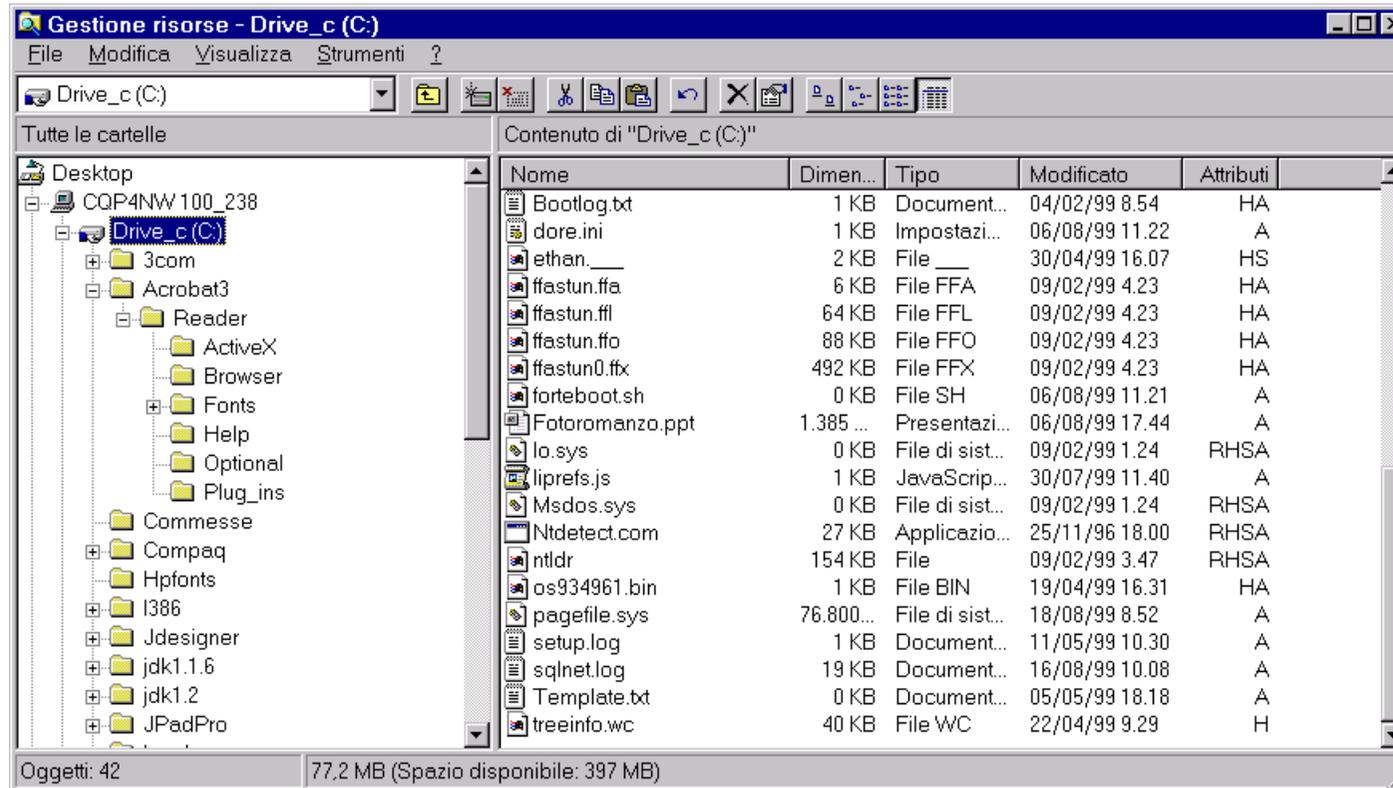


# Il FileSystem

---

- E' l'insieme di directory e file in cui sono organizzati i dati presenti nelle memorie di massa
- Nel caso di sistemi DOS, Windows e UNIX ha una struttura ad albero, in cui la directory principale prende il nome di radice
- In altri sistemi le strutture possono essere differenti

# Il FileSystem - 2



# Memoria Virtuale

---

- La memoria di lavoro può essere maggiore della RAM effettiva a disposizione
- Una parte del disco viene dedicata a contenere l'estensione della memoria (area o file di swap)
- Con apposite politiche di gestione una parte del contenuto della RAM viene scaricato nell'area di swap e viceversa
- Le prestazioni sono ovviamente inferiori

# Sistemi Operativi Vari

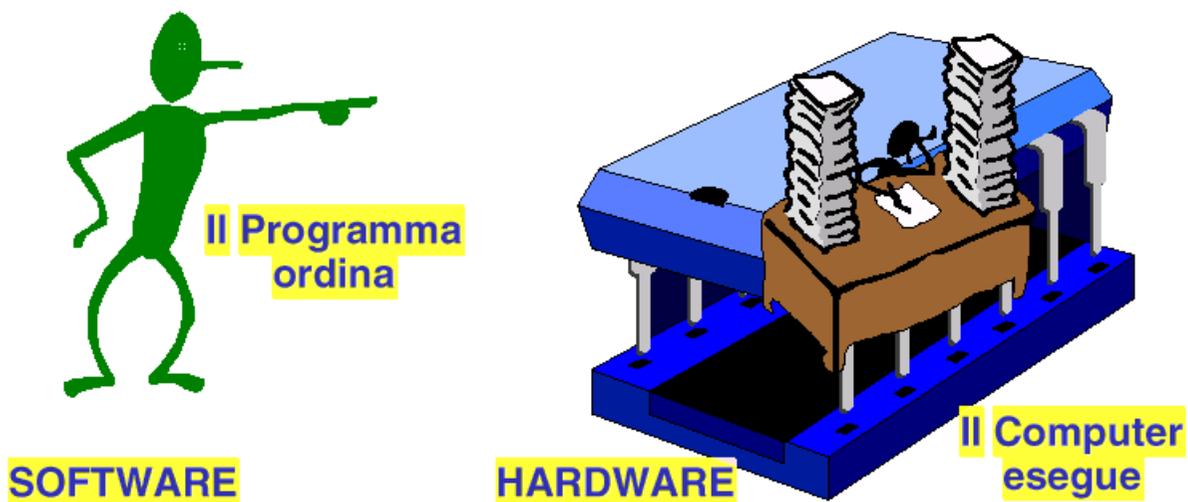
---

- MS-DOS, DOS+Windows 3.x
- MacOS
- Windows95/98/XP/Vista/7
- Windows2000/2003/2008
- UNIX e Linux
- OS/390, MVS
- OS/400
- VMS

# I Programmi

---

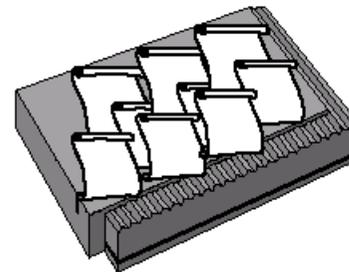
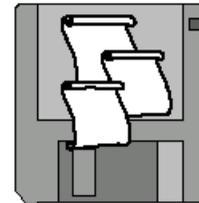
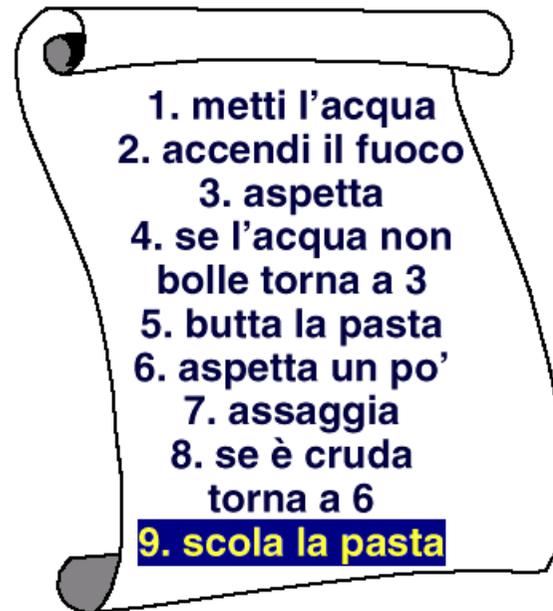
L'unità centrale è progettata per eseguire i programmi.



# Cos'è un Programma?

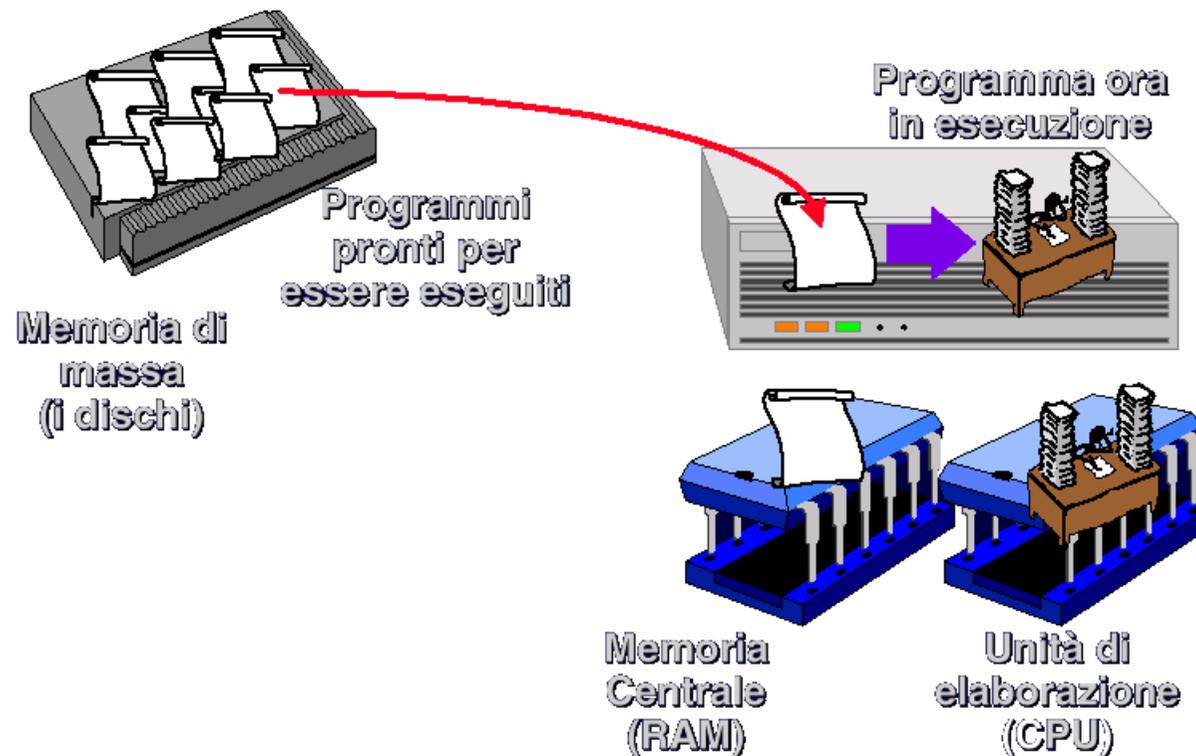
---

**E' una sequenza di istruzioni  
memorizzata in un file**



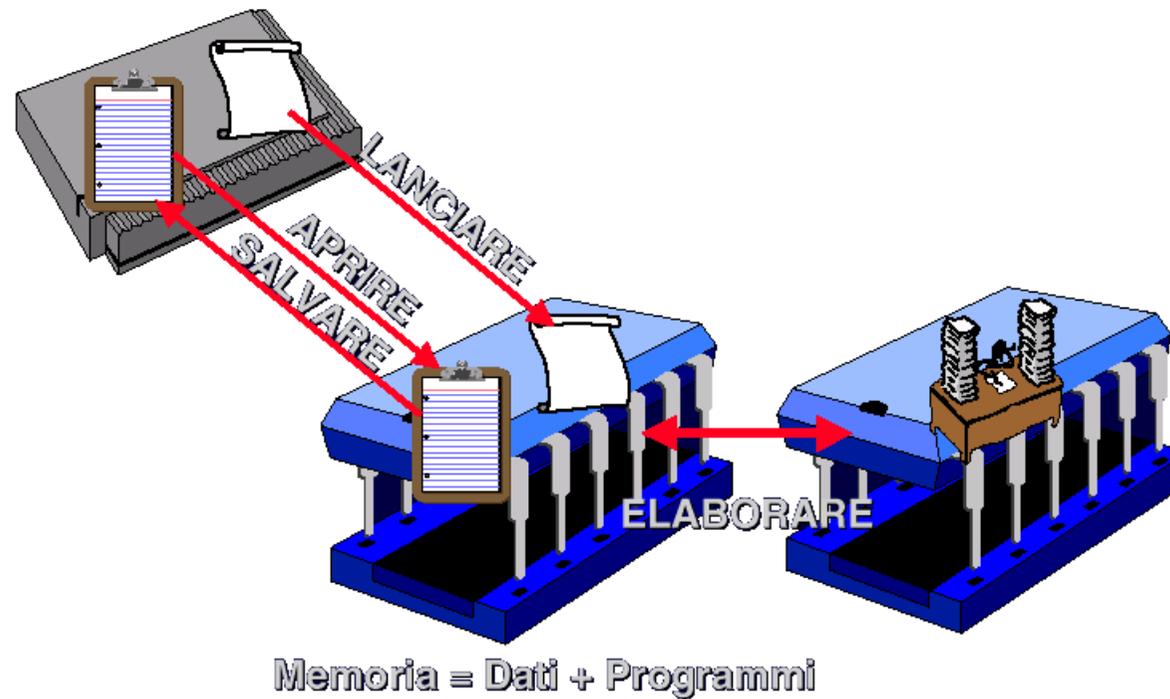
# Esecuzione dei Programmi

---

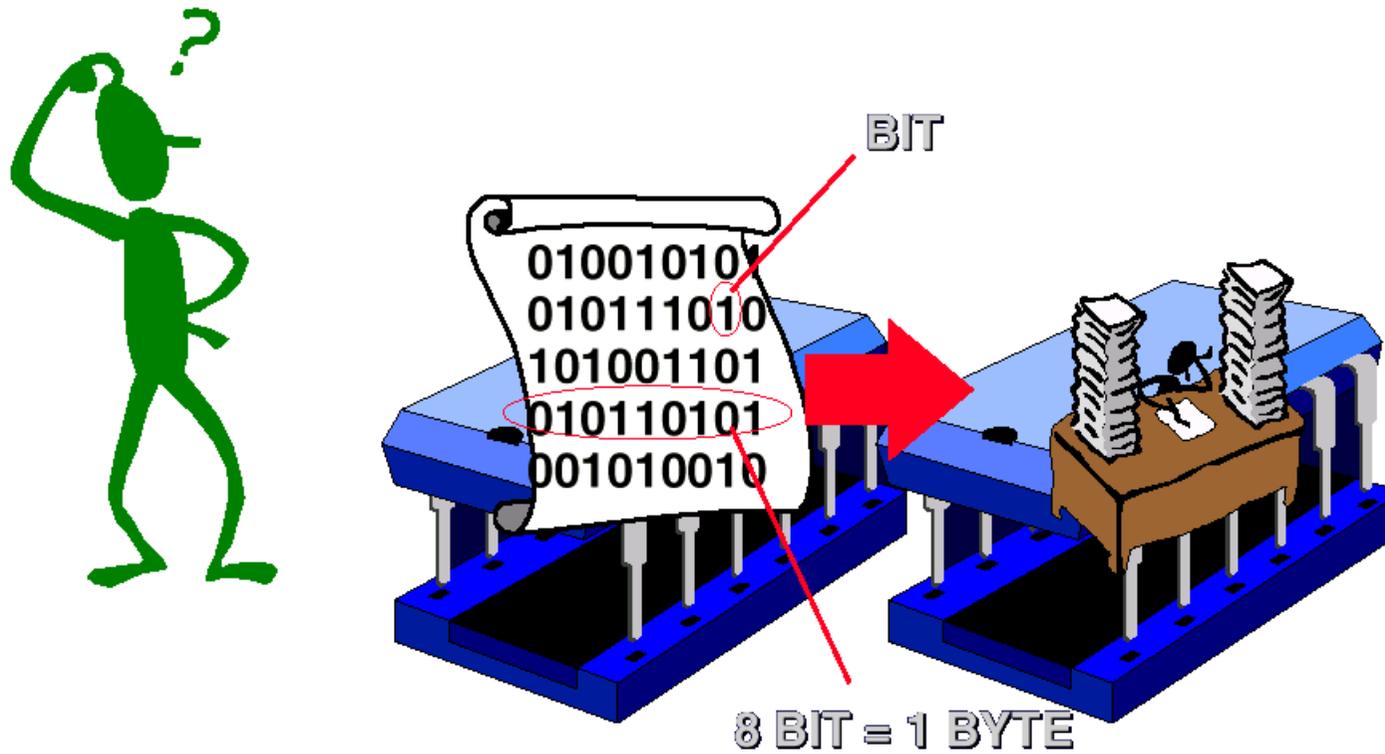


# Il percorso dei dati

---



# Cosa vede la CPU?



# Tipi di programmazione

---

- Linguaggio macchina (composto di numeri binari)
- Linguaggio assembly o assembler (corrispondenza biunivoca con linguaggio macchina ma più comprensibile agli umani)
- Linguaggio ad alto livello come C, C++, Pascal, Java, COBOL (sintassi ben definita)
- Linguaggio 4GL come ABAP
- Scripting di Applicazione come VBA entro Word
- Scripting Web (JavaScript, VBScript, PHPScript)

# Il Flusso delle Istruzioni

---

- Le istruzioni vengono eseguite secondo una ben precisa sequenza
- Il computer sa sempre quale è l'istruzione correntemente in esecuzione
- Il **program counter** è l'indice della istruzione correntemente in esecuzione
- In dipendenza di condizioni il flusso di esecuzione può essere diverso fra una esecuzione ed un'altra

# Variabili, costanti, array

---

- Ogni elemento di un programma ha associata un'etichetta o **identificatore**
- Una variabile è un elemento il cui valore può variare nel tempo
- Una costante riceve un valore all'inizio dell'esecuzione che poi non varia più
- Un array è un vettore di elementi identici

# Sottoprogrammi

---

- Un programma per essere manutenibile deve avere una struttura chiara
- Compiti diversi devono essere svolti da parti diverse del programma
- Esempi: stampe, lettura di dati da file...
- Una procedura è una parte di programma che svolge un compito ben determinato

# Sottoprogrammi - 2

---

- Una procedura viene invocata col suo nome
- Una procedura accetta delle variabili che prendono il nome di **argomenti**
- Esempi:
  - LeggiDatiDaFile('c:\pippo.dat')
  - StampaSuVideo('salve mondo')
  - Addiziona(totale, nuovodato)
- In programmazione ad oggetti alle procedure corrispondono i **metodi**

# Il Compilatore

---

- E' un programma eseguibile
- Traduce le istruzioni scritte in un linguaggio ad alto livello in istruzioni comprensibili per il computer
- Può produrre codice macchina o codici intermedi
- Il processo di compilazione si divide in varie fasi

# La stesura del programma

---

File sorgente (testo, linguaggio di alto livello)

↓  
**Compilazione**

↓  
File oggetto (binario, linguaggio macchina)

↓  
**Linking**

↓  
File eseguibile (binario, linguaggio macchina,  
pronto per l'esecuzione)

# La stesura del programma - 2

---

File sorgente (testo, linguaggio di alto livello)

↓  
**Compilazione**

↓  
File codice binario (binario, pseudo istruzioni  
macchina)

↓  
**Interpretazione**

↓  
Codice macchina nativo in memoria e sua  
esecuzione

# La stesura del programma - 3

---

File sorgente (testo, linguaggio di scripting)



**Caricamento e Interpretazione**



Codice macchina nativo in memoria e sua esecuzione entro l'interprete

# Il Multithreading

---

- Nel multithreading esistono più flussi di esecuzione entro uno stesso programma
- Il gestore del multithreading (es. il sistema operativo) amministra la successione dei vari flussi
- Il coordinamento dei vari thread fra loro entro il programma è, in modo più o meno facilitato, a carico del programmatore

# Gli ambienti integrati di sviluppo

---

- Editor interno
- Debugger
- Compilatore integrato
- Archiviazione del software e lavoro di gruppo

# I componenti di un programma

---

- L'interfaccia utente
- La logica applicativa o di business
- I dati