



OBIETTIVI:

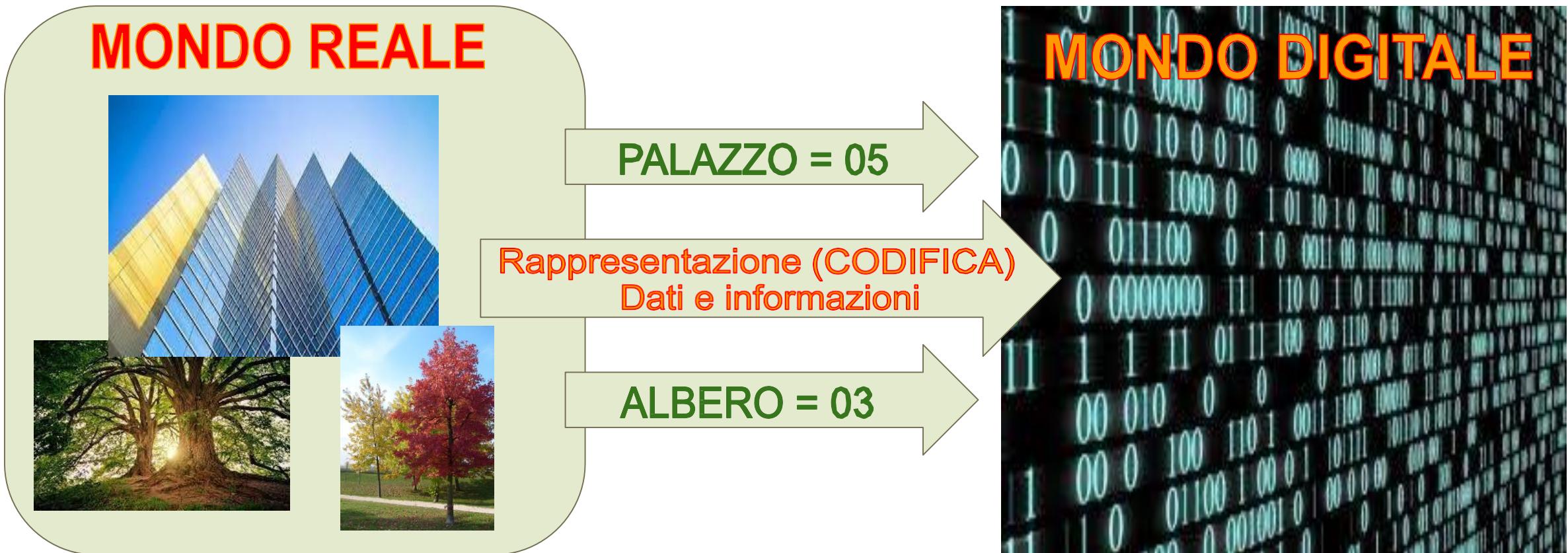
1. Comprendere la logica dell'informazione digitalizzata e della elaborazione automatica dei dati;
2. Conoscenza della logica di funzionamento e reciproche interazioni tra le componenti principali dell'elaboratore elettronico(CPU, periferiche memoria ecc);
3. Conoscenza della rappresentazione dei dati in memoria;
4. Sviluppare la capacità di gestire condizioni logiche complesse applicando l'algebra di Boole;
5. Conoscenza delle differenze e peculiarità tra linguaggi compilati, interpretati e pseudocompilati.

● CU1 : “Basi di Informatica”

- CU1 : “Basi di Informatica”

- Presentazione azienda
- Presentazione docente:
- Presentazione corso : sua composizione , durata , finalità
- Presentazione dei discenti

• L'informatica e l'Informazione: *Definizione*



- Definizione : la scienza che si occupa del trattamento dell'informazione mediante procedure automatizzate
- Perché scegliere l'informatica: per risolvere problemi attraverso l'uso di sistemi di elaborazione o calcolatori elettronici

• L'informatica e l'Informazione: *Esempio di elaborazione di dati*

- Somma di due numeri interi
- Dati a e b , risolvere l'equazione $ax+b=0$
- Calcolare il massimo comun divisore fra due numeri dati.
- Dato un insieme di parole, metterle in ordine alfabetico.
- Calcolare l'intersezione di due insiemi.
- Dato un elenco di nomi e relativi numeri di telefono trovare il numero di telefono di una determinata persona
- Dati gli archivi dei dipendenti di un'azienda, calcolare lo stipendio medio del dipendente dell'azienda.
- Data l'immagine satellitare di un'area geografica, calcolare le previsioni del tempo per domani.

• L'informatica e l'Informazione: *Elaborazione, dato, informazione*



□ Che cos'è **un'elaborazione** dati? Varie definizioni.

Procedimento informatico che, a partire da determinati dati in input, produce determinati dati come output dopo una certa Manipolazione/trasformazione.

Esempi elaborazione: calcolo busta paga mensile impiegato,...

□ Che cos'è un **dato**? Varie definizioni.

Un dato è una descrizione elementare, codificata, di un'entità, di un fenomeno, di una transazione, di un avvenimento o di altro, **appartenente al mondo reale**.

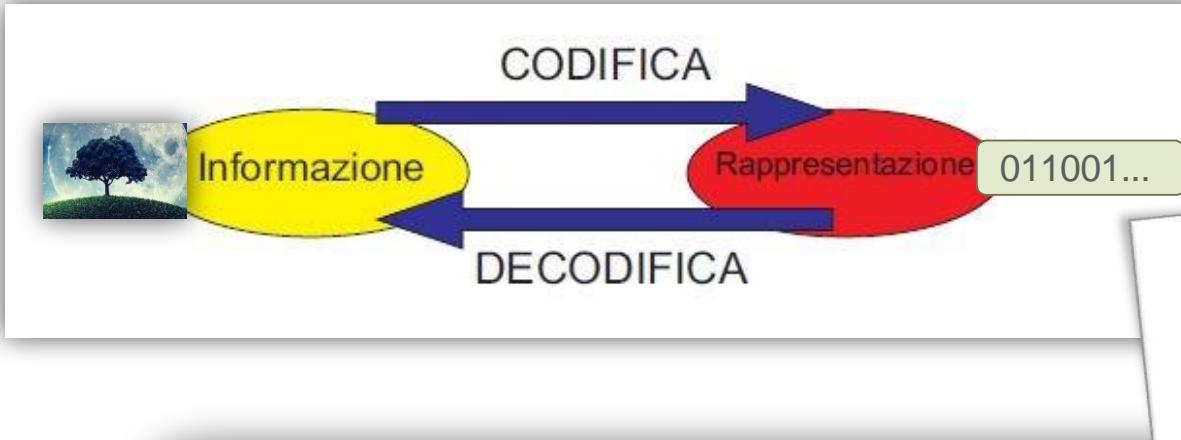
Esempio dato: codice fiscale impiegato, numero di telefono, i giorni lavorati nel mese dall'impiegato, ecc.

□ Che cos'è un **informazione**?

Un'informazione è l'insieme di dati, correlati tra loro, che sintetizzano o descrivono con concetto della realtà o un'idea.

Esempio informazione: posizione contrattuale , previdenziale impiegato, dati anagrafici, ecc.

• L'informatica e l'Informazione: *Codifica e Rappresentazione Dati*



- La codifica è una **convenzione!**
- E' il modo in cui associamo un'informazione ad una sua rappresentazione binaria.

- Rappresentazione binaria: l'elemento base è il bit (binary digit, cifra binaria) che assume valori 0 o 1
- Per rappresentare più valori si usano sequenze di bit

1bit 2 valori: 0, 1

2bit 4 valori: 00, 01, 10, 11

3bit 8 valori: 000, 001, 010, 011, 100, 101, 110, 111

In generale: n bit rappresentano 2^n diversi valori

- 4 bit: nibble, 8 bit: byte

• L'informatica e l'Informazione: *Multipli del Byte*

I computer hanno una memoria finita.

Quindi, l'insieme dei numeri interi e reali e dei caratteri che si possono rappresentare in un computer è necessariamente finito



Unità di misura

Di solito si usano i multipli del byte

Kilo	KB	2^{10} (circa un migliaio, 1024)
Mega	MB	2^{20} (circa un milione, $1KB \times 1024$)
Giga	GB	2^{30} (circa un miliardo, $1MB \times 1024$)
Tera	GB	2^{40} (circa un migliaio di miliardi, $1GB \times 1024$)

• L'informatica e l'Informazione: *Sistemi di numerazione*

DECIMALI (base 10)

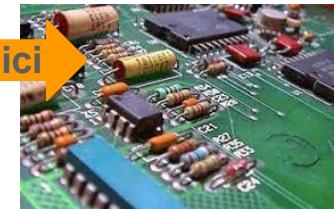
$$(134)_{10} = 1 \times 10^2 + 3 \times 10^1 + 4 \times 10^0$$



BINARI (base 2)

adatto alla gestione con circuiti elettronici

$$(101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (5)_{10}$$



OTTALE (base 8)

$$(647)_8 = 6 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 = (423)_{10}$$

ESADECIMALE (base 16)

$$(123)_{16} = 1 \times 16^2 + 2 \times 16^1 + 3 \times 16^0 = (291)_{10}$$

• L'informatica e l'Informazione: Conversione tra Decimale e Binario

Da decimale a Binario

Numero	/2	Resto
142	71	0
71	35	1
35	17	1
17	8	1
8	4	0
4	2	0
2	1	0
1	0	1

$(142)_{10} =$
 $(10001110)_2$
 $(216)_8$
 $(8E)_{16}$

Da Binario a Decimale

$$(101110)_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 1 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 32 + 0 + 8 + 4 + 2 + 0 = 46$$

$$(101110)_2 = (46)_{10}$$

Numeri Ottali e Esadecimali

- I numeri binari sono molto lunghi rispetto alla quantità di info che rappresentano
- Le rappresentazioni ottali e esadecimali sono versioni **compatte** di numeri binari

OTTALE (a tre a tre)

$$(1 \ 100 \ 111 \ 010)_2 = (1472)_8$$

ESADECIMALE (a quattro a quattro)

$$(11 \ 0011 \ 1010)_2 = (33A)_{16}$$

N.B. Le cifre da 10 a 15 si rappresentano con le lettere A,...,F

• L'informatica e l'Informazione: Rappresentazione Caratteri

- Tre possibili rappresentazioni:

- **ASCII standard**: un carattere è rappresentato con **7 bit** (ASCII = American Standard Code for Information Interchange)
- **ASCII estesa**: un carattere è rappresentato con **8 bit**
- **UNICODE** : un carattere è rappresentato con **16 bit** (MS Windows ne usa una simile)

Rappr. Binaria

Il codice ASCII standard `e codificato su **7 bit**, e quindi puo rappresentare al massimo $2^7 = 128$ simboli diversi. Esiste una versione del codice ASCII che usa un byte. Tale codice viene detto Codice ASCII esteso e permette la rappresentazione di $2^8=256$ caratteri, (128 di base con le varianti relative alle diverse lingue + i caratteri speciali e grafici).



ogni carattere è rappresentato in 1 byte e quindi possiamo rappresentare 256 caratteri. Questo basta per:

a...z A...Z 0...9 .,: () etc

+ caratteri di controllo: Enter, Tab, etc



Byte	Cod	Char	Byte	Cod	Char	Byte	Cod	Char
00100000	32	Spc	01000000	64	@	01100000	96	-
00100001	33	!	01000001	65	A	01100001	97	a
00100010	34	"	01000010	66	B	01100010	98	b
00100011	35	#	01000011	67	C	01100011	99	c
00100100	36	\$	01000100	68	D	01100100	100	d
00100101	37	%	01000101	69	E	01100101	101	e
00100110	38	&	01000110	70	F	01100110	102	f
00100111	39	,	01000111	71	G	01100111	103	g
001010								
0 0 000	NUL	(null)	32 20 040	#32; Space	64 40 100	#64; Space	96 60 140	#96; -
1 1 001	SOH	(start of heading)	33 21 041	#33; !	65 41 101	#65; A	97 61 141	#97; a
2 2 002	STX	(start of text)	34 22 042	#34; "	66 42 102	#66; B	98 62 142	#98; b
3 3 003	ETX	(end of text)	35 23 043	#35; #	67 43 103	#67; C	99 63 143	#99; c
4 4 004	ETB	(end of transmission)	36 24 044	#36; \$	68 44 104	#68; D	100 64 144	#100; d
5 5 005	ENQ	(enquiry)	37 25 045	#37; %	69 45 105	#69; E	101 65 145	#101; e
6 6 006	ACK	(acknowledge)	38 26 046	#38; &	70 46 106	#70; F	102 66 146	#102; f
7 7 007	DEL	(bell)	39 27 047	#39; ,	71 47 107	#71; G	103 67 147	#103; g
8 8 010	BS	(backspace)	40 28 050	#40; (72 48 110	#72; H	104 68 150	#104; h
9 9 011	TAB	(horizontal tab)	41 29 051	#41;)	73 49 111	#73; I	105 69 151	#105; i
10 A 012	LF	(NL line feed, new line)	42 2A 052	#42; *	74 4A 112	#74; J	106 6A 152	#106; j
11 B 013	VT	(vertical tab)	43 2B 053	#43; +	75 4B 113	#75; K	107 6B 153	#107; k
12 C 014	FF	(NP form feed, new page)	44 2C 054	#44; ,	76 4C 114	#76; L	108 6C 154	#108; l

• L'informatica e l'Informazione: *Rappresentazione Immagini*

L'immagine è suddivisa in pixel



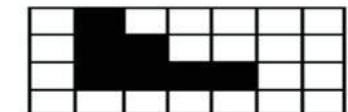
Pixel



Ad ogni pixel associamo una rappresentazione binaria:

0 ₂₂	1 ₂₃	0 ₂₄	0 ₂₅	0 ₂₆	0 ₂₇	0 ₂₈
0 ₁₅	1 ₁₆	1 ₁₇	0 ₁₈	0 ₁₉	0 ₂₀	0 ₂₁
0 ₈	1 ₉	1 ₁₀	1 ₁₁	1 ₁₂	0 ₁₃	0 ₁₄
0 ₁	0 ₂	0 ₃	0 ₄	0 ₅	0 ₆	0 ₇

0000000011110001100000100000



- Assegnando un bit ad ogni pixel si possono rappresentare solo immagini in bianco e nero
- Per rappresentare immagini a diversi livelli di grigio o a colori: a ogni pixel è associata una sequenza di bit
 - con 8 bit per pixel: $2^8 = 256$ livelli di grigio
 - con 24 bit per pixel: $2^{24} = 16777216$, 16.7 milioni di colori



Formati

- Immagini bitmap
- Immagini bitmap compresse: GIF, JPEG...
- Immagini vettoriali: SVG...

• L'informatica e l'Informazione: Rappresentazione Dati in Memoria

Memorizzazione delle informazioni sulla memoria del calcolatore (cenni)

- ▶ Per poter comprendere e saper utilizzare correttamente costanti, variabili e tutti gli aspetti del linguaggio che hanno attinenza con i tipi di dati, è necessario comprendere come tali dati vengano *memorizzati* sul calcolatore.
- ▶ Il calcolatore ha una **memoria finita**, quindi, per un dato tipo è possibile rappresentare solo un numero ristretto di valori possibili. Ad esempio, il tipo di dato **int** non ci permette di rappresentare l'intero dominio dei **numeri interi**.
- ▶ La memoria di un calcolatore può essere astrattamente considerata come una *sequenza finita di celle*:
 - ▶ ogni cella può memorizzare un **bit** (Binary digIT). Il bit è *l'unità di informazione* basilare per i calcolatori e può assumere due soli valori: 0 oppure 1.
 - ▶ le celle di memoria sono raggruppate in **parole** di lunghezza prefissata. Una parola è chiamata **byte** e consiste di **8 bit**. Il byte è la più piccola quantità di memoria direttamente *indirizzabile* in molte architetture di calcolatori.
 - ▶ Esempi:
 - ▶ con 1 bit è possibile memorizzare $2^1 = 2$ valori distinti;
 - ▶ con 1 byte (= 8 bit) è possibile memorizzare $2^8 = 256$ valori distinti;
 - ▶ con 2 byte (= 16 bit) è possibile memorizzare $2^{16} = 65,536$ valori distinti;
 - ▶ con 4 byte (= 32 bit) è possibile memorizzare $2^{32} = 4294967296$ valori.
- ▶ Date le limitazioni fisiche e le scelte progettuali adottate per i calcolatori, il **numero di valori distinti** che è possibile memorizzare in un *tipo di dato* dipende dal **numero di byte** associati a tale tipo.

L'informatica e l'Informazione: Range di valori dei Tipi di Dati

Dimensioni e range di valori dei tipi di dati base

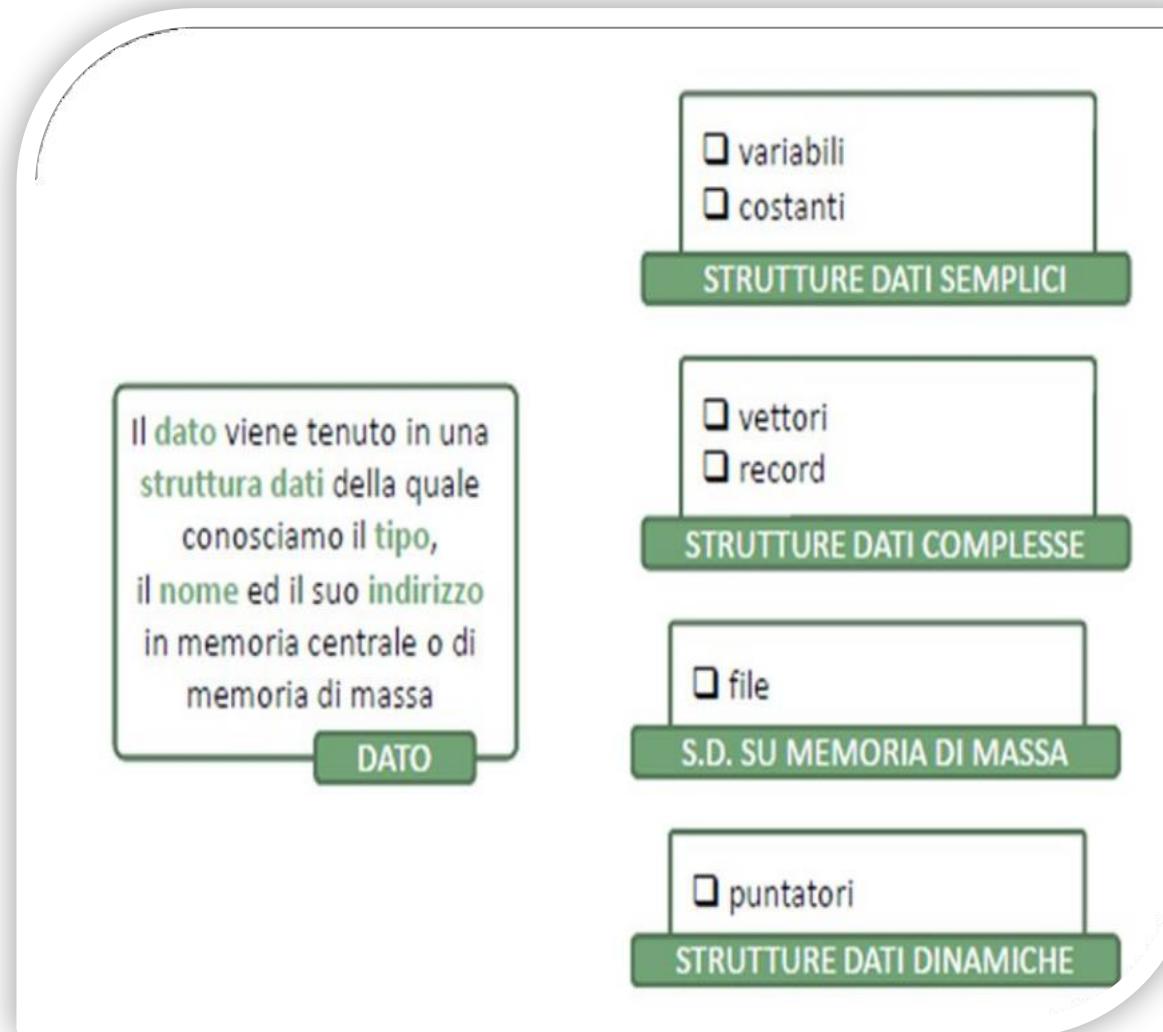
- Lo standard ISO C89 definisce dimensioni minime (in byte) per i tipi di dato base del linguaggio C. Le dimensioni effettive dipendono dall'hardware della macchina: processori a 32 bit o 64 bit tipicamente utilizzano dimensioni differenti.

Tipo	Byte	Min	Max
<code>signed char</code>	1	-128	127
<code>unsigned char</code>	1	0	255
<code>short int</code>	2	-32767	32767
<code>unsigned short int</code>	2	0	65535
<code>int</code>	4	-2147483647	2147483647
<code>unsigned int</code>	4	0	4294967295
<code>long int</code>	4	-2147483647	2147483647
<code>unsigned long int</code>	4	0	4294967295
<code>float</code>	4	1.175494×-38	3.402823×38
<code>double</code>	8	2.225074×-308	1.797693×308
<code>long double</code>	16	$3.362103 \times 10^{-4932}$	$1.189731 \times 10^{4932}$

Tabella: Dimensioni tipiche su processore a 32 bit

- Il tipo `char` può essere di default `signed` o `unsigned`, a seconda dell'implementazione.
- I tipi in virgola mobile non possono essere `unsigned`. La tabella mostra in valore assoluto il minimo (escludendo lo 0) e massimo numero rappresentabile per `float`, `double` e `long double`.

• L'informatica e l'Informazione: *Strutture Dati e Variabili*



Una variabile è un'area della RAM riservata per un uso specifico. Ci si riferisce ad essa mediante un **nome**; essa ha *sempre* un **valore**; i valori ammessi sono tutti dello stesso **tipo**.

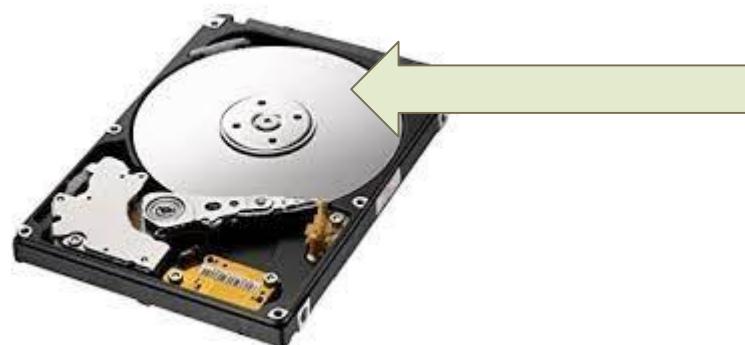
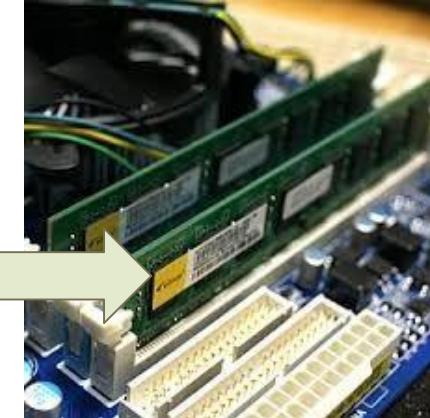
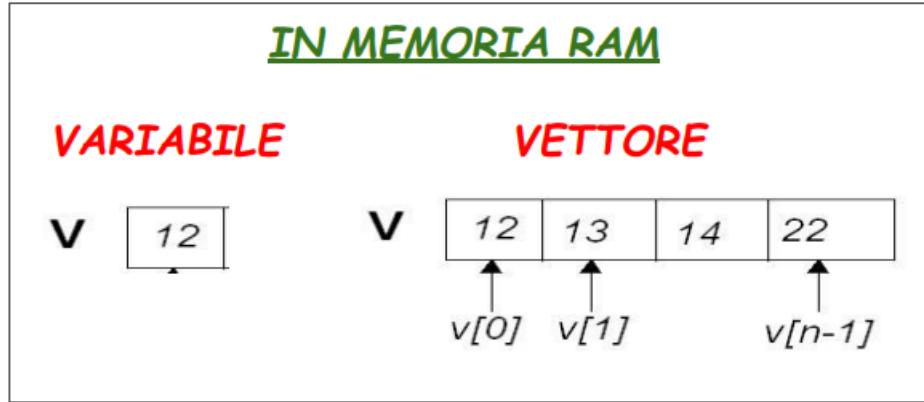
VARIABILE

Dichiarare una variabile vuol dire riservare lo spazio in memoria.

Ogni variabile viene dichiarata una volta sola.

L'**assegnazione** è la procedura mediante la quale si definisce il valore della variabile. Il valore di una variabile di solito cambia diverse volte.

• L'informatica e l'Informazione: *Strutture Dati e Variabili*

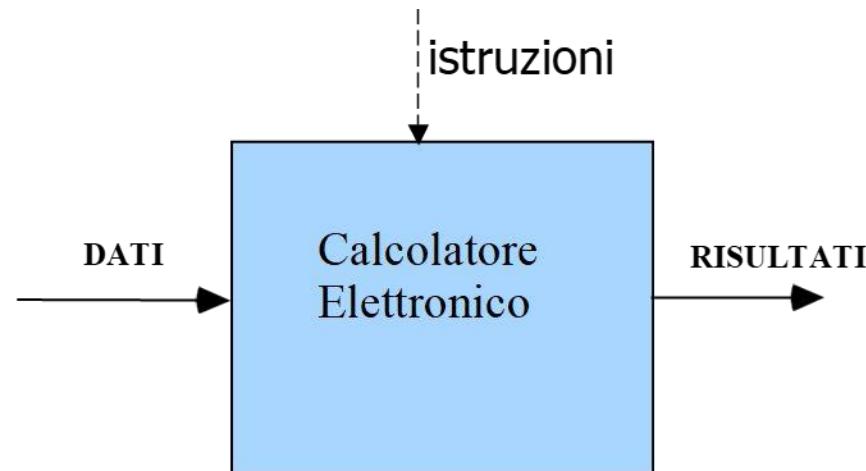


IN MEMORIA DI MASSA (DISCHI)

RECORD	FILE																								
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Adrian</th> <th>Ailincai</th> <th>2900</th> <th>21.0</th> </tr> </thead> <tbody> <tr> <td>Marco</td> <td>Albergamo</td> <td>31000</td> <td>27.0</td> </tr> <tr style="outline: 2px solid red;"> <td>Leonardo</td> <td>Amico</td> <td>8000</td> <td>21.0</td> </tr> <tr> <td>Massimiliano</td> <td>Arpino</td> <td>1500</td> <td>19.2</td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td>Valerio</td> <td>Zerillo</td> <td>45000</td> <td>34.5</td> </tr> </tbody> </table>	Adrian	Ailincai	2900	21.0	Marco	Albergamo	31000	27.0	Leonardo	Amico	8000	21.0	Massimiliano	Arpino	1500	19.2					Valerio	Zerillo	45000	34.5
Adrian	Ailincai	2900	21.0																						
Marco	Albergamo	31000	27.0																						
Leonardo	Amico	8000	21.0																						
Massimiliano	Arpino	1500	19.2																						
Valerio	Zerillo	45000	34.5																						

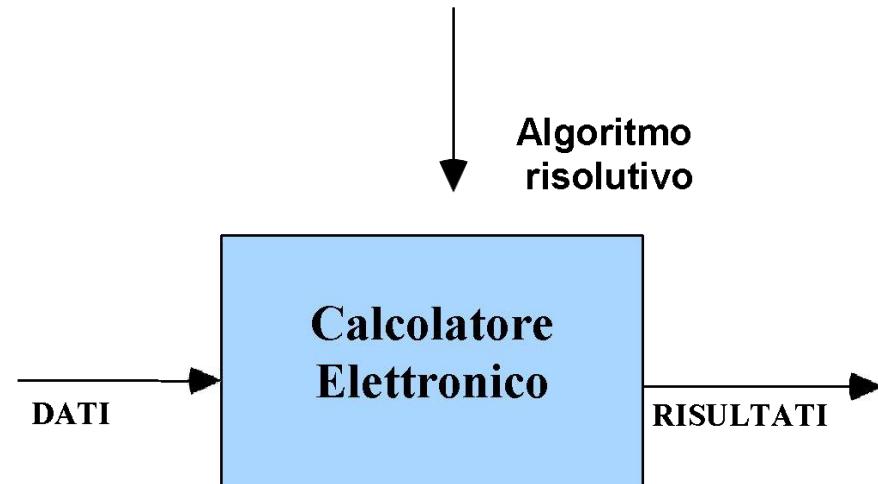
• Il calcolatore elettronico

- Un sistema di elaborazione, o **computer** o **calcolatore elettronico**, è quindi un sistema in grado di risolvere problemi in modalità automatica (programmata) e di trattare in maniera automatica informazioni
- L'elaboratore è un mero esecutore di «comandi» o meglio istruzioni programmato dall'esterno. Le sequenze di istruzioni impartite al computer prendono il nome di **programmi**

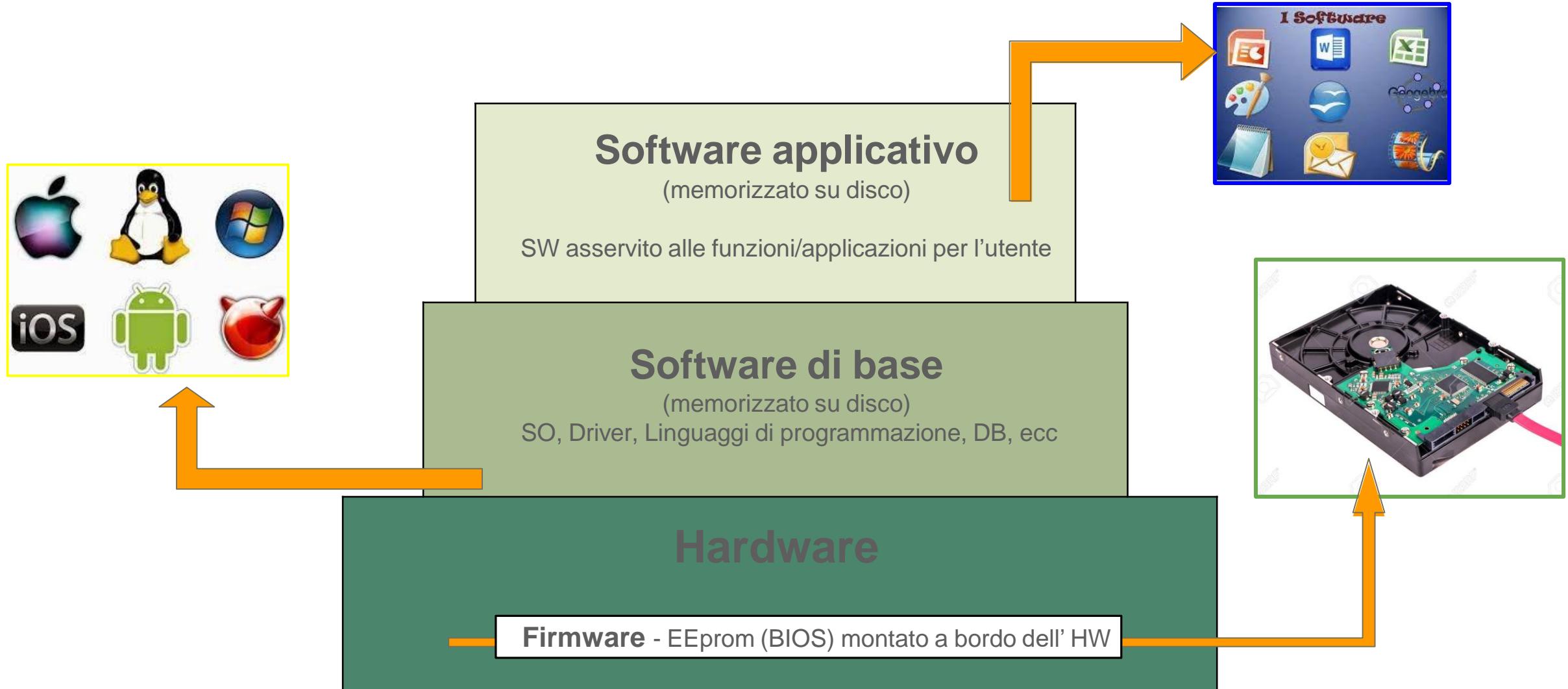


• Il calcolatore elettronico: *Programmazione*

- E` l'attività con cui si predisponde l'elaboratore ad eseguire un particolare insieme di azioni su particolari informazioni (*dati*), allo scopo di risolvere un certo problema .

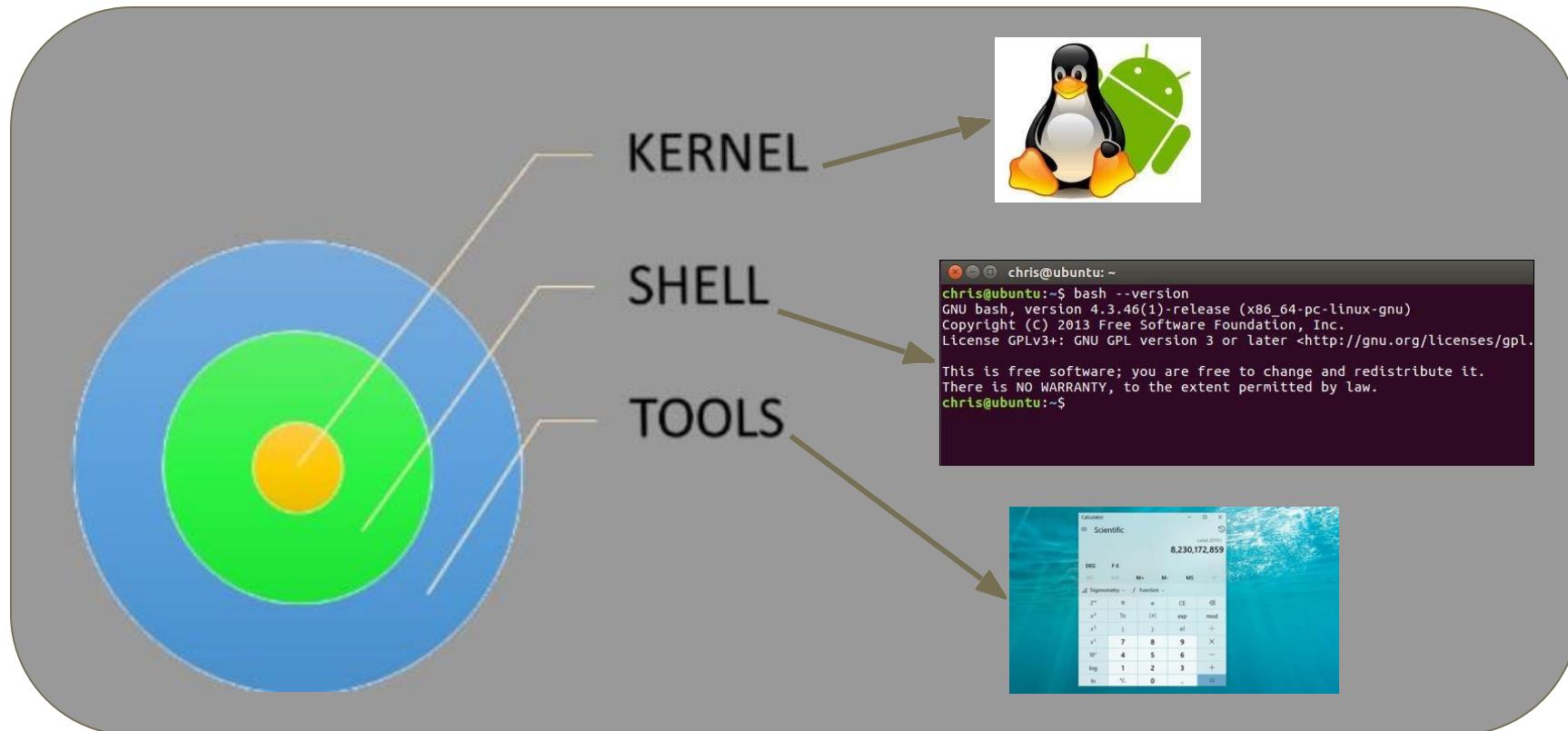


• Il calcolatore elettronico: *Componenti*



• Il calcolatore elettronico: *Il Sistema Operativo*

Il sistema operativo (SO) riconducibile al Software di Base, è un insieme di programmi che ottimizzano l'uso delle risorse di un elaboratore e permettono l'interazione tra gli utenti e il software applicativo e l'hardware di un computer. Viene caricato all'accensione del computer ed è così stratificato:



Principali funzioni del S.O.

- 1.gestisce le risorse del computer: la memoria, il processore, le periferiche, ecc.;
- 2.consente la comunicazione con il computer, attraverso l'interfaccia uomo-macchina;
- 3.controlla l'esecuzione dei programmi applicativi;
- 4.controla gli errori software e hardware.

• Il calcolatore elettronico: *Componenti*

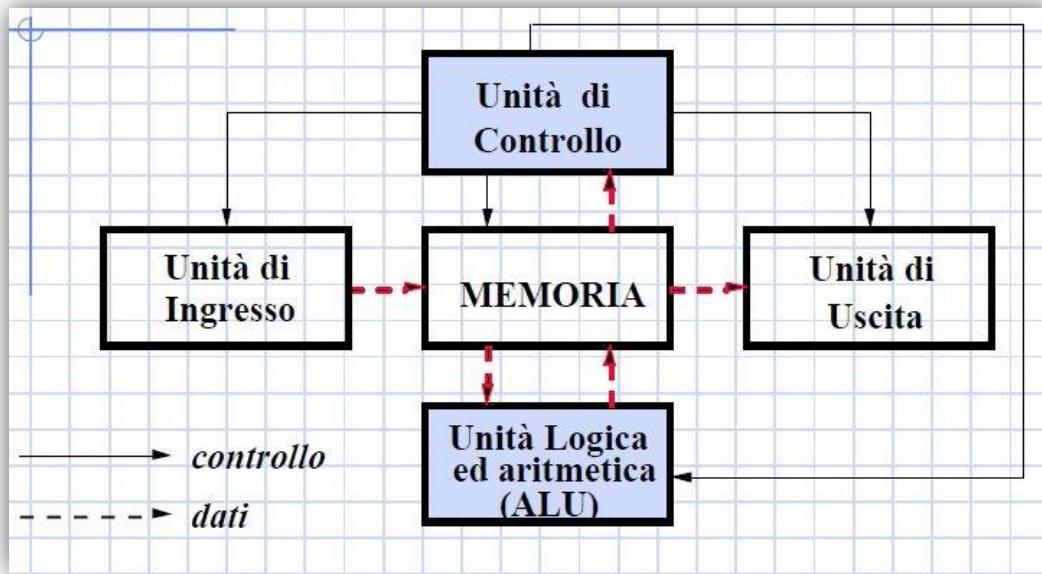
- In un sistema di elaborazione si possono così identificare due componenti principali:
 - **l'hardware** : è la parte fisica del computer, l'insieme di tutti i componenti di natura elettronica e meccanica che lo costituiscono (l'unità centrale, con tutti i suoi componenti, la tastiera, il mouse, il monitor, ecc.); essi sono visibili e tangibili.
 - **il software** : comprende tutte quelle parti che non hanno una consistenza fisica, in particolare i programmi che permettono allo strumento di operare. Si distingue in:
 - ❖ software di base: p.e. il Sistema Operativo (S.O.)
 - ❖ software applicativo: p.e. gli antivirus, i programmi per la navigazione in Internet e per la posta elettronica, i programmi di ritocco fotografico, i lettori audio/video,i videogiochi,ecc.

Il firmware : parte intermedia tra hardware e software si tratta del software che viene “intergrato”(p.e. in **ROM**) nei dispositivi elettronici digitali durante la loro costruzione.

Il suo compito è “avviare” il componente elettronico(anche sue parti o periferiche) e permettere a quest'ultimo di entrare in comunicazione con altre schede o componenti del dispositivo su cui sono installate. (**BIOS** è l'acronimo di **Basic Input-Output System** l'esempio storico)

Contiene istruzioni operative sul funzionamento del dispositivo ma anche protocolli di comunicazione che rendono possibile lo scambio di informazioni tra varie parti del sistema. Non direttamente modificabile dall'utente finale

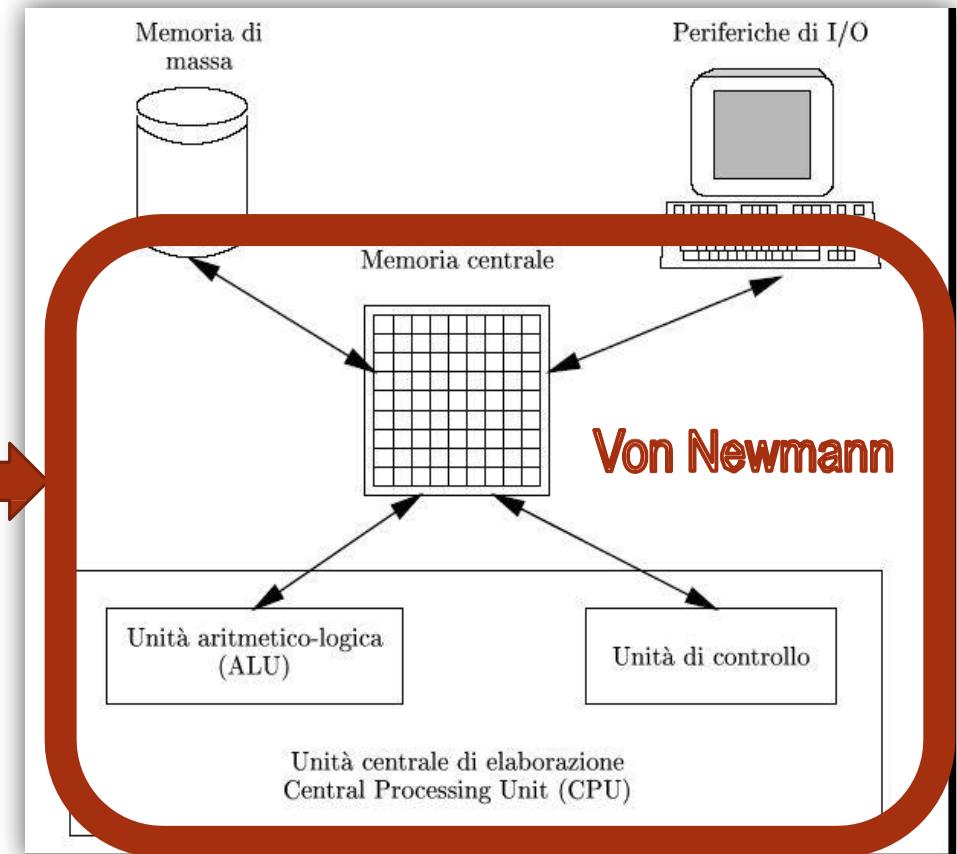
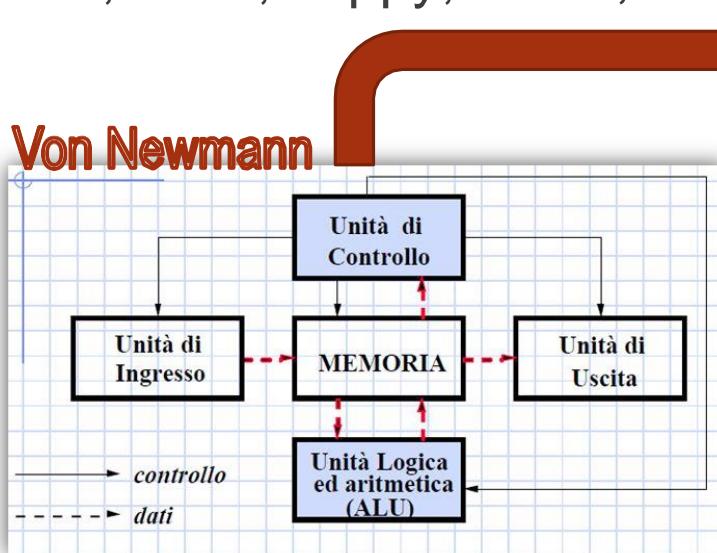
• Il calcolatore elettronico: *Un Modello, la macchina di Von Neumann*



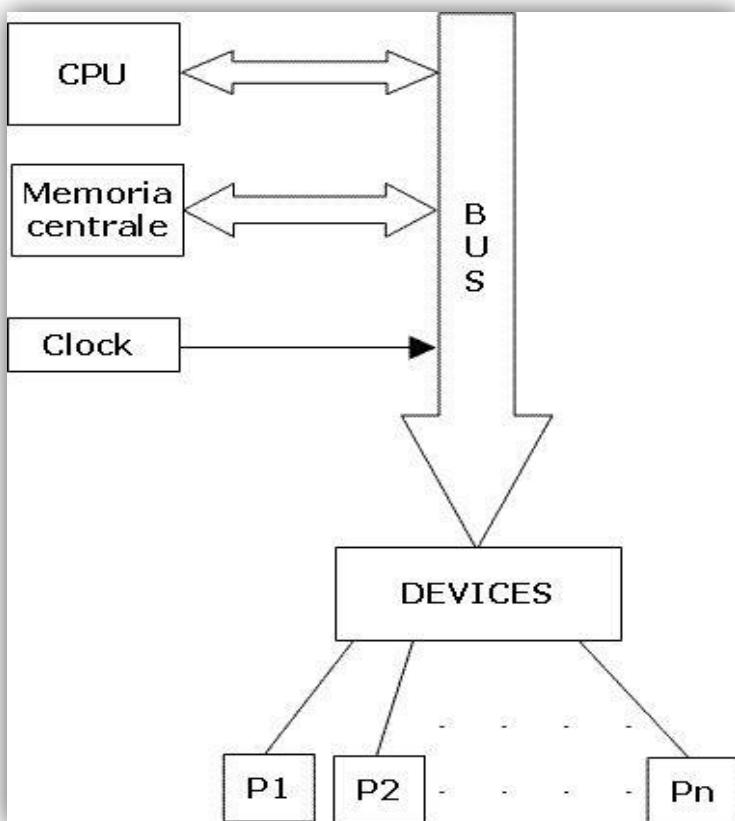
- **Unità di Ingresso** per l'acquisizione dei dati e dei programmi e per il loro trasferimento in memoria
- **Unità di Memoria** per la registrazione sia dei dati che delle istruzioni del programma
- **Unità di Controllo** presiede a tutte le operazioni, interpreta le istruzioni prelevate dalla memoria e ne guida l'esecuzione inviando appositi segnali alle altre unità.
- **Unità Aritmetico-Logica (ALU)**, dedicata alla esecuzione delle operazioni aritmetiche e logiche
- **Unità di Uscita** per il trasferimento all'esterno dei risultati presenti in memoria

• Il calcolatore elettronico: Architettura

- **Unità Centrale:** CPU
- **Memoria centrale:** RAM, cache(1°,2° livello), registri
- **Periferiche I/O :** tastiera, monitor, stampanti,...
- **Memoria di massa(secondaria):** hard disk, memorie ottiche, flash, floppy, nastri,...



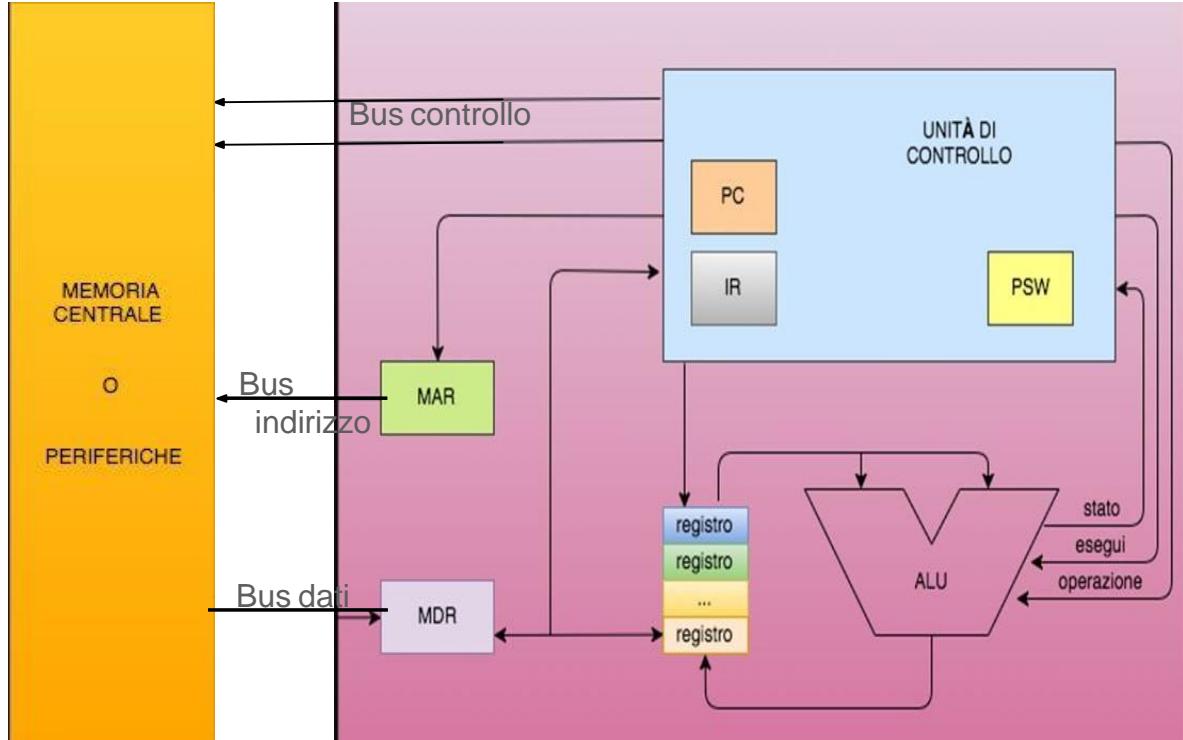
• Il calcolatore elettronico: *Componenti fondamentali*



Dallo schema di **Von Neumann** si arriva quindi alle quattro componenti fondamentali:

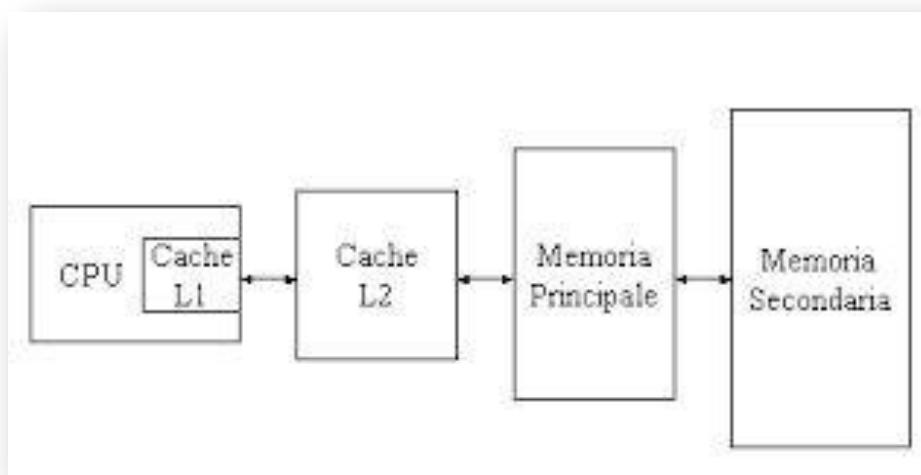
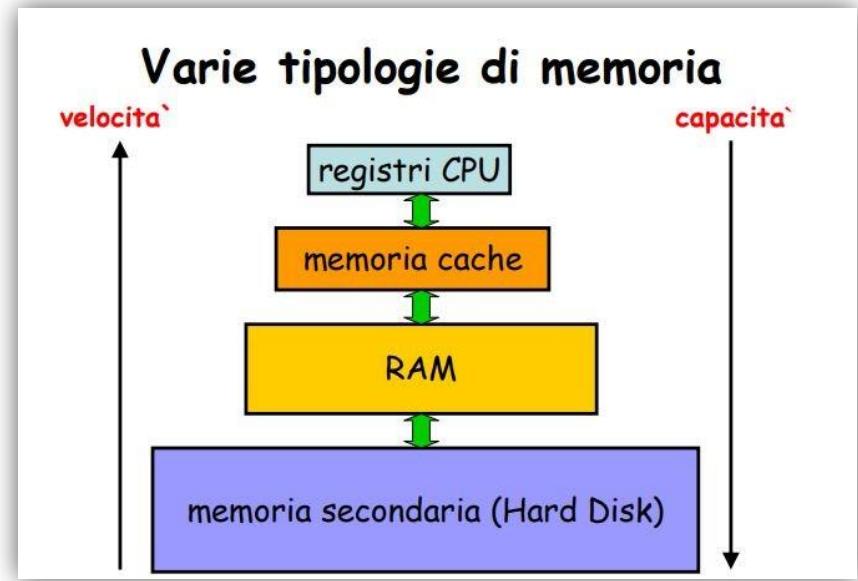
- **CPU** (Central Processing Unit – Unità centrale di elaborazione), che svolge le funzioni di elaborazione e controllo. Essa contiene i dispositivi elettronici in grado di acquisire, interpretare ed eseguire i programmi, elaborando i dati che arrivano dall'esterno.
- **Memoria centrale** (o memoria principale RAM), in cui vengono memorizzati sia i programmi che devono essere eseguiti che i dati che devono essere elaborati dai programmi.
- **Devices**, che sono i dispositivi d'interfacciamento verso l'esterno; gestiscono lo scambio d'informazione fra l'elaboratore e le periferiche.
- **Bus**, che è l'insieme dei collegamenti elettrici che si occupano della comunicazione fra i vari componenti, realizzando un supporto fisico per la trasmissione dei dati tra di essi.
- **Clock**: la frequenza del segnale di temporizzazione e corrisponde al numero di operazioni elementari per secondo che la CPU è in grado di eseguire

• Il calcolatore elettronico: CPU - Unità Centrale di Elaborazione



- L'*unità aritmetico-logica (Arithmetic-Logic Unit, ALU)* effettua le operazioni di tipo aritmetico e logico sugli operandi forniti ai suoi ingressi.
- I *registri* sono piccole celle di memoria interne alla CPU su cui vengono memorizzate le informazioni necessarie per l'esecuzione delle istruzioni (es., l'istruzione che il processore sta elaborando, i dati su cui opera l'istruzione corrente, l'indirizzo della prossima istruzione da eseguire, ...)
- L'*unità di controllo* coordina le varie unità nell'esecuzione dei programmi, comandando opportunamente i segnali di controllo dei registri, dell'ALU e dei bus che interfacciano la memoria.

• Il calcolatore elettronico: *Tipologie di Memoria*



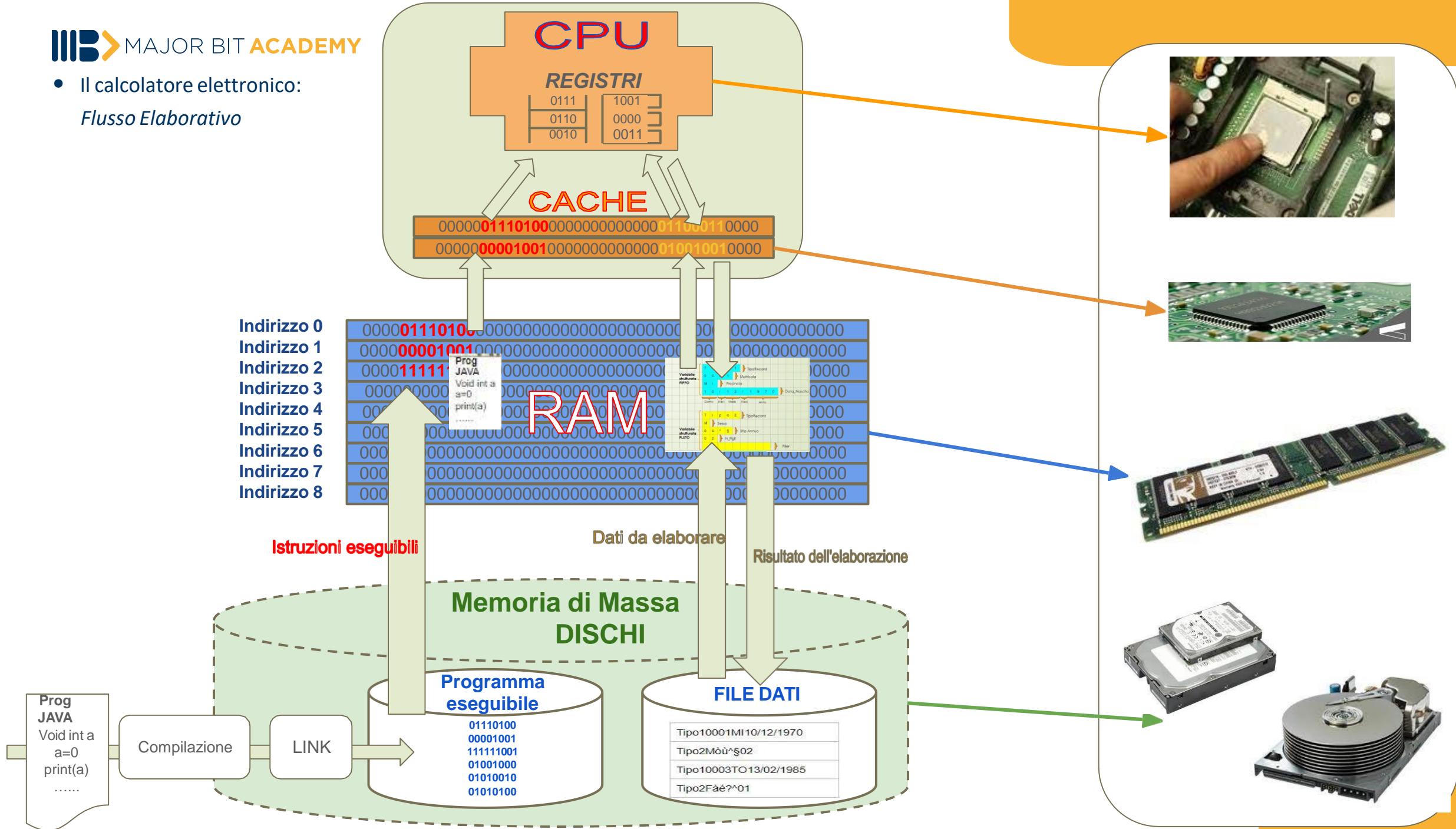
REGISTRI: Hanno la funzione di memorizzare all'interno della CPU dati e istruzioni necessari all'esecuzione

La memoria **RAM** (Random Access Memory) è una delle componenti della memoria centrale del computer insieme alla cache e alla ROM. Ha accesso casuale utilizzata nei computer per registrare dati e informazioni in modo temporaneo e volatile

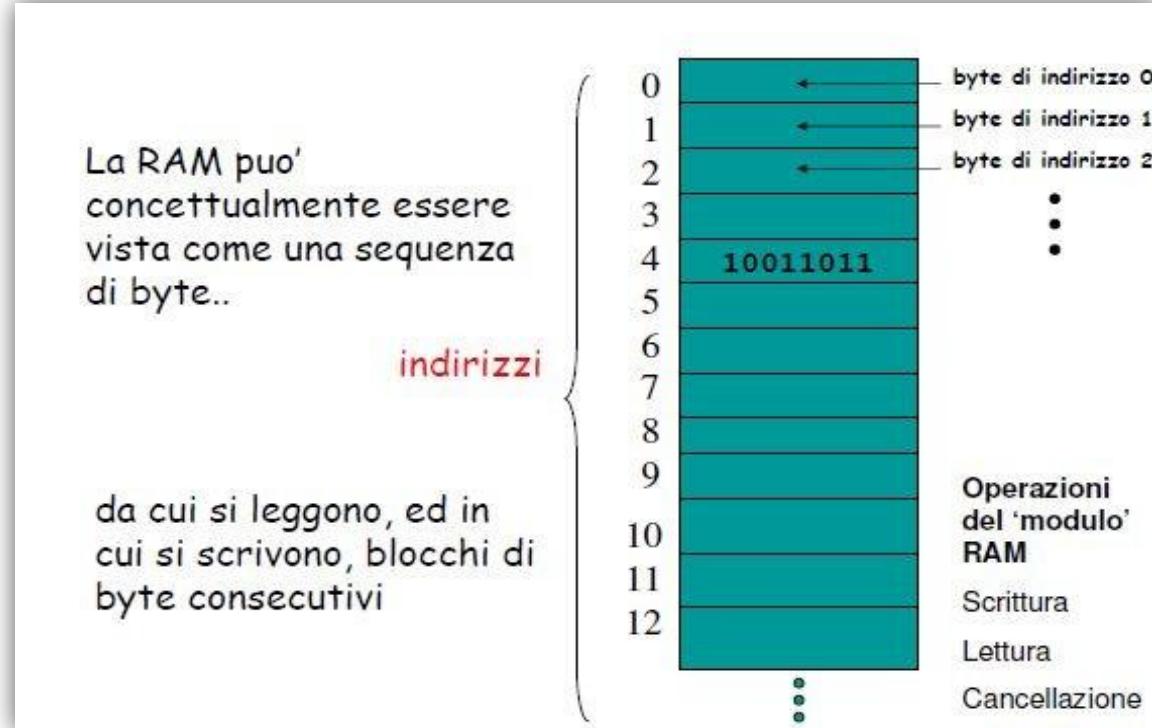
CACHE: conosciuta anche come CPU cache. È una memoria temporanea e volatile di scrittura e lettura, situata nel processore (CPU) e nella scheda madre. La memoria cache è utilizzata per registrare dati da recuperare velocemente durante l'elaborazione, le istruzioni e i dati utilizzati più di frequente dal processore

Unità di memoria di massa : forniscono il supporto fisico per la memorizzazione permanente dei dati, e presentano caratteristiche estremamente diverse a seconda della casa costruttrice e del tipo di unità.

- Il calcolatore elettronico:
Flusso Elaborativo



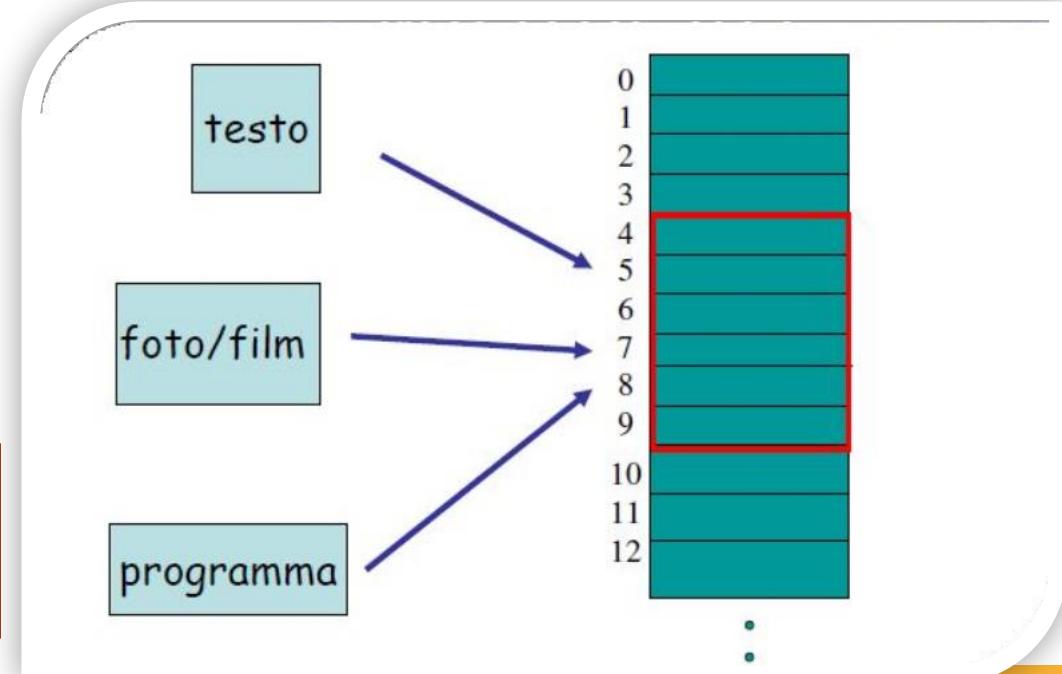
• Il calcolatore elettronico: *Organizzazione della Memoria principale RAM*



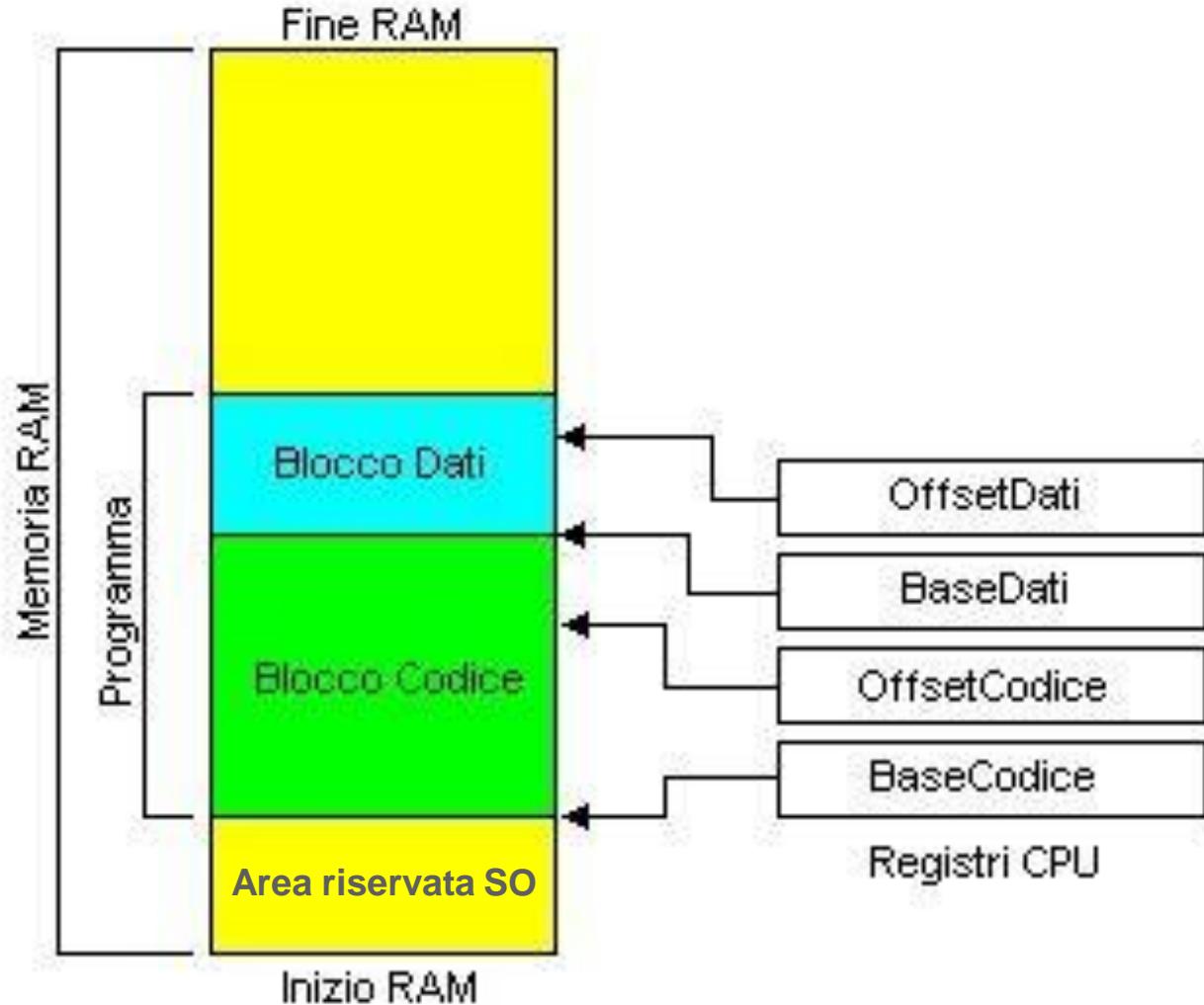
Un indirizzo di memoria identifica in modo univoco una locazione (cella) della memoria centrale del computer per leggere o per scrivere un «informazione»

Organizzata come: "un insieme ordinato di contenitori" ("celle" o "locazioni") aventi tutte la stessa «dimensione»

Il byte e' la minima quantità di memoria accessibile (attraverso il corrispondente indirizzo) un'unità di misura di base della capacità di memoria di un elaboratore elettronico



• Il calcolatore elettronico: *Organizzazione della Memoria principale RAM*



ATTENZIONE

1. *Vietato scrivere sulle aree riservate della RAM*
2. *Scrivere grandi quantità di dati in RAM può provocare un errore di OVERFLOW*



• Algebra di Boole

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1
(somma logica)		

A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

(prodotto logico)

A	not A
0	1
1	0

(negazione)

Le tabelle elencano tutte le possibili combinazioni in ingresso e il risultato associato a ciascuna combinazione

Quando usiamo l'algebra di Boole ?

E' un'algebra basata su tre operazioni logiche:

Ed operandi che possono avere due soli valori:

Tramite questa algebra possiamo comporre "espressioni logiche" che risulteranno, nel loro insieme, vere o false.

- Per esprimere **condizioni complesse** in alcuni costrutti tipici dei linguaggi di programmazione:
 - **SCELTA - (If (A and B) then)**
 - **ITERAZIONE - (while - do while (A and B))**

Ma anche nel linguaggio SQL per scrivere **espressioni complesse** di selezione dei dati:

- SQL (Select * from tabella where (B or A and not(C))

• Livelli di astrazione nei linguaggi di programmazione

I “livelli” dei linguaggi

Linguaggi di alto livello: più vicini al nostro modo di comprendere e impostare problemi, più facili da imparare e da trasferire e da ‘portare’ su un altro computer (esempio: HTML, Java, C++, Visualbasic ecc.)

Linguaggi di basso livello: vicini al linguaggio macchina e all’architettura fisica del computer (esempio: Assembler)

Linguaggio macchina: usa solo 0 e 1 ed è comprensibile direttamente dall’hardware, ma molto difficile per l’uomo

Numerico-Vespignani, Informatica per le scienze umanistiche, Il Mulino, 2003

41

Classificazione Livello di astrazione

- Confronto fra linguaggi ad alto e a basso livello:



• Livelli di astrazione nei linguaggi di programmazione

Linguaggio di ALTO LIVELLO

Hello World in C:

```
#include main()
{
printf('Hello, World');
}
```

Vantaggi e Svantaggi

- Più semplice da capire
 - Più compatto
- Meno veloce in fase di esecuzione
- Meno adatto per l'elaborazione REAL TIME e la gestione dei macchinari

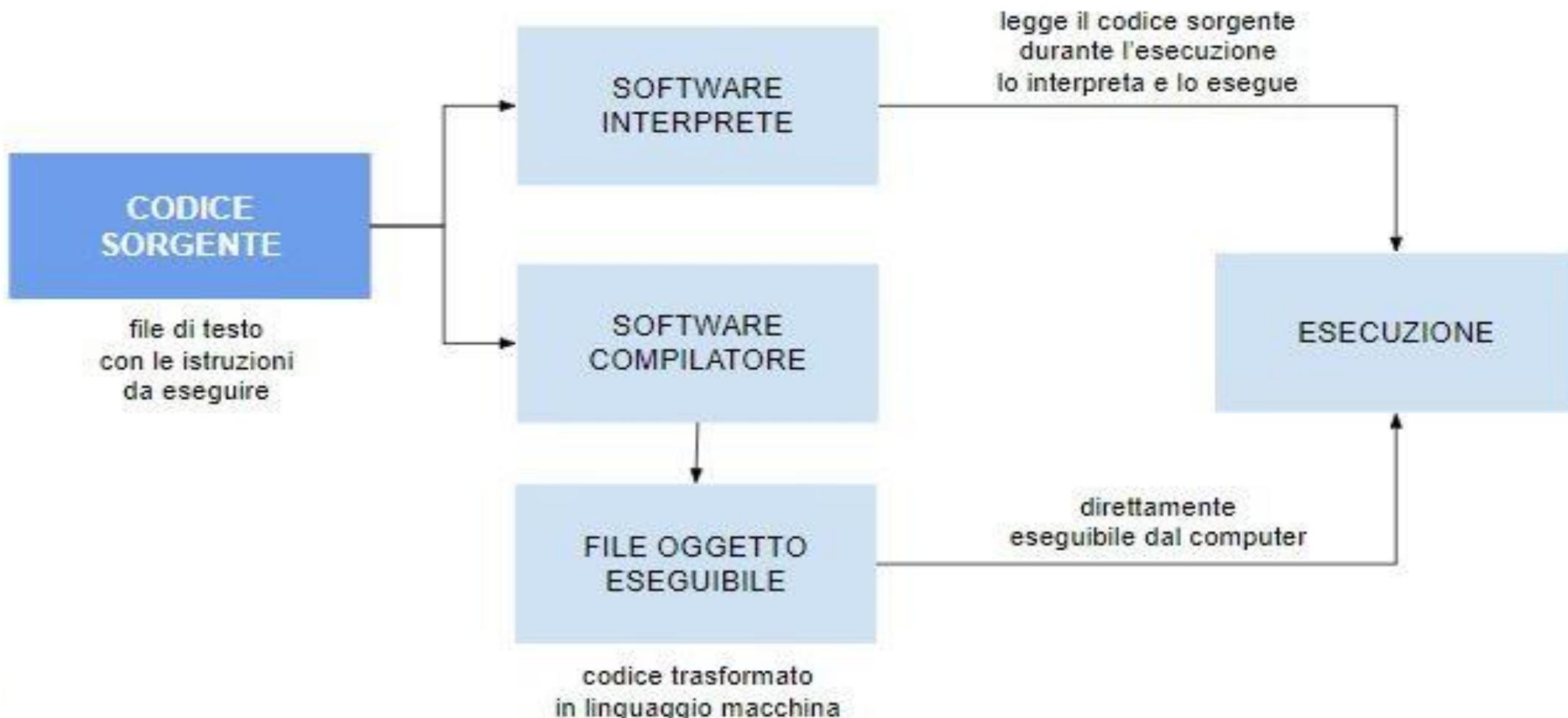


Linguaggio di BASSO LIVELLO

Hello World in ASSEMBLY con sintassi intel (wikipedia):

```
IDEAL
MODEL SMALL
STACK 100h
DATASEG HW DB "Hello, world", 13,
10, '$' CODESEG
Begin:
    MOV AX, @data
    MOV DS, AX
    MOV DX,
    OFFSET HW
    MOV AH, 09H INT 21H
    MOV AX, 4C00H INT 21H
END Begin
```

• Programmi Compilati e Interpretati



• Programmi Compilati e Interpretati: *Definizione*

- Linguaggi come il C, C++, Delphi, Visual Basic sono **LINGAGGI COMPILATI** (linguaggi imperativi con prestazioni migliori dei linguaggi interpretati:
 1. **scrittura del codice** utilizzando un editor qualsiasi o utilizzando un ambiente di sviluppo IDE che ne facilita la scrittura
 2. **compilazione** ovvero verifica degli errori di sintassi. Ogni istruzione viene trasformata nel corrispondente codice in linguaggio macchina che può essere, così, eseguito dal processore
- Con i **LINGUAGGI INTERPRETATI**, il codice sorgente viene interpretato direttamente al momento dell'esecuzione. Per es il PHP il cui codice viene "elaborato" e restituisce una pagina html pura. E caratterizzato da:
 1. **alta portabilità**
 2. **carico di lavoro maggiore per il processore** (ad ogni run il codice deve essere nuovamente reinterpretato).

• Programmi Compilati e Interpretati: *Il linguaggio JAVA*

- Un **linguaggio che è a metà strada tra queste metodologie** è **Java** che è sia compilato che interpretato; il codice sorgente viene compilato in un formato intermedio (chiamato bytecode), il quale a sua volta viene interpretato dalla Java Virtual Machine (JVM), che ha il compito di interpretare “al volo” le istruzioni bytecode in istruzioni per il processore; la JVM viene sviluppata per ogni Sistema Operativo e permette di astrarre la macchina virtuale creata dal SO ad un livello di standardizzazione superiore (data di fatto dalla creazione della virtual machine sopra un’altra virtual machine) che rende, in pratica, JAVA altamente portabile.
- Questa metodologia implica la possibilità di controllare eventuali errori del codice sorgente (grazie alla compilazione), di creare programmi relativamente leggeri (il bytecode è un formato che crea file di dimensioni ragionevoli), ma ha la pecca di avere delle prestazioni non proprio soddisfacenti, questo perché il codice viene interpretato dalla JVM che a sua volta deve delegare l’esecuzione vera e propria al Sistema Operativo.

• Programmi Compilati e Interpretati: *C++, JAVA, Python, C#, VB.NET*

- **C++ è un linguaggio ad alto livello orientato agli oggetti** (OOP) sviluppato da **Bjarne Stroustrup** e pubblicato nel **1983** (il più anziano tra i tre) come miglioramento del linguaggio C, già esistente.C++ è un **linguaggio compilato** con una sintassi abbastanza complessa, ma resta, anche a distanza di più di 30 anni, uno dei più utilizzati soprattutto per l'ottima implementazione della parte orientata agli oggetti e la sua discreta efficienza.
- **Java** è stato sviluppato da **James Gosling** e pubblicato definitivamente nel maggio del **1995**. Anche Java (ora appartenente alla **Oracle**) è **orientato agli oggetti**. Java è tuttora utilizzato per creare contenuti sul web e per sviluppare giochi e applicazioni per mobile Android.
E' senza dubbio, se non il primo, uno dei primi per quanto riguarda l'utilizzo in quanto con esso sei in grado di creare "tutto ciò di cui hai bisogno!". Qualsiasi progetto ti balena nella testa potrai realizzarlo con Java.**Java**, grazie alla **JVM** (Java Virtual Machine) è un **linguaggio sia compilato sia interpretato**.
- **Python** fu ideato da **Guido van Rossum** all'inizio degli anni novanta (1991) ed è uno dei linguaggi che ha rivoluzionato il mondo con la sua sintassi molto semplice.E' un linguaggio multi-paradigma, è distribuito con licenza Open-Source approvata dalla OSI, perciò il suo utilizzo è gratuito e libero anche per prodotti commerciali. Anche Python è un **linguaggio orientato agli oggetti**.**Python è un linguaggio interpretato**.
- **C# VB.net: linguaggi orientati ad oggetti e compilati**
- **SQL linguaggio di interrogazione** utilizzato nella consultazione e interrogazione di database e archivi dati; le sue istruzioni recuperano ,inseriscono aggiornano dati . Diventano linguaggi di programmazione nelle implementazioni in T-SQL per microsoft e PL-SQL per oracle nella scrittura delle stored procedure

