Introduzione ai calcolatori

Sommario

- Cosa sono le Tecnologie dell’Informazione e della Comunicazione.
- Architettura di Von Neumann e struttura di una CPU.
- Hardware e periferiche. I fattori che influiscono sulle prestazioni di un computer. La memoria.
- Codice binario e rappresentazione dell’informazione.
- Linguaggi di programmazione

Tecnologie dell'informazione e della comunicazione (TIC/ICT)

- Insieme dei metodi e delle tecniche utilizzate nella trasmissione, ricezione ed elaborazione di dati e informazioni.
- Tecnologie dell'informazione
- Telecomunicazioni

- **Società dell’informazione**
- attuale società post-industriale, in cui i beni immateriali prevalgono su quelli industriali.

Sistemi di elaborazione dell'informazione

- Gli elementi principali di un sistema di calcolo si suddividono in 2 categorie:
  - **Hardware**, rappresenta la parte fisica dell'elaboratore, costituita da componenti elettronici ed elettromeccanici
  - **Software**, costituito da tutti i programmi che consentono all'hardware di realizzare compiti specifici

Software

- **Software di Base**
- funzionale all‟utilizzo dell’elaboratore e delle sue periferiche, comprende:
- **Sistema operativo**
- Programmi **traduttori** dei linguaggi di programmazione

- **Software applicativo**
- mostra all’utente il calcolatore come una macchina virtuale utilizzabile per la **risoluzione** di **problemi** specifici;
- Comprende tutte le **applicazioni adoperate dagli utenti**: videoscrittura, foglio elettronico, ecc.

Software

Sistema operativo

- È l'insieme di programmi che
  - servono alla **gestione dell'hardware**
  - permettono **l'interazione tra l'utente e l'hardware** del calcolatore.

- Alcune funzioni sono:
- **Comunicare con l'utente**: ricevere ed eseguire i comandi, comunicare eventuali errori
- **Gestire l'allocazione della memoria** e di tutte le risorse del calcolatore
- **Acquisire i dati in ingresso** dalla tastiera e dagli altri dispositivi di input
- **Inviare i risultati** al video e agli altri dispositivi di output
- **Leggere e scrivere i dati nella memoria** di massa

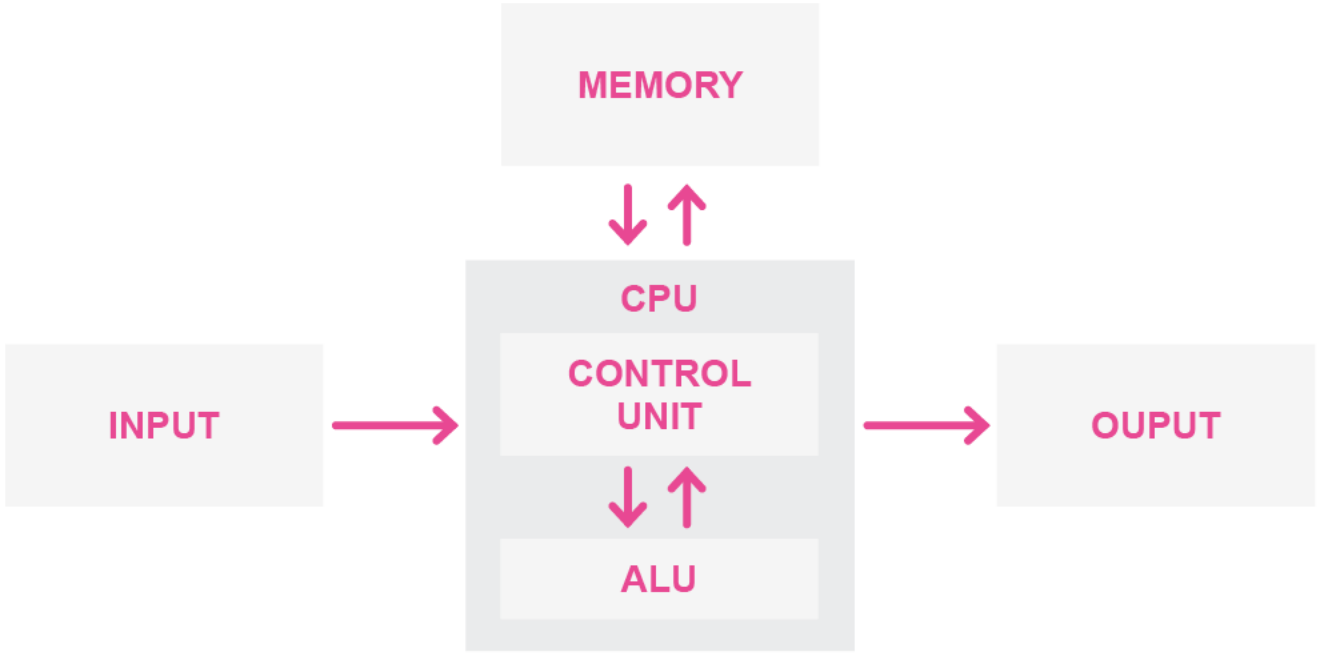
Architettura di un calcolatore

- Le componenti fondamentali della architettura dei calcolatori sono:
  - **Unità centrale di elaborazione** (o CPU) che preleva le istruzioni dalla memoria e le esegue, legge/scrive dati dalla memoria
  - **Memoria principale** (RAM, ROM) che contiene i dati e le istruzioni dei programmi
  - **Memoria secondaria** o di massa (HD, CD, DVD, ecc.) per memorizzare dati e programmi in maniera permanente
  - **Dispositivi di input** (tastiera, mouse, touch pad, ecc.) per l'inserimento dei dati
  - **Dispositivi di output** (monitor, stampante, ecc.) per ricevere i risultati

Architettura di Von Neumann

- Progettata da John Von Neumann nel 1952
- Basata sul concetto di programma memorizzato in memoria
- Sia i dati che le istruzioni del programma erano memorizzati

Architettura di Von Neumann



Memoria principale (RAM)

- Contiene **dati e istruzioni** che sono oggetto di elaborazione da parte del processore

- L'elemento base della Ram è la **cella di memoria**, le cui caratteristiche sono:
  - Può assumere solo due stati (0/1)
  - È possibile scrivere nella cella per cambiare lo stato
  - È possibile leggere lo stato della cella

- È organizzata come una sequenza di locazioni di memoria.
- Ogni locazione è una sequenza di n bit, in genere 8 o 16

- Ogni locazione di memoria è individuata da un indirizzo univoco che ne specifica la posizione. L'indirizzo di memoria è un numero binario intero positivo

Memoria ROM (Read Only Memory)

- È una memoria a **sola lettura**, che contiene informazioni **permanenti** e **non modificabili**
- I dati sono impostati nel chip durante il processo di fabbricazione

- È usata per **memorizzare istruzioni di sistema** come ad esempio le istruzioni necessarie per l'**avvio del sistema** operativo e per il riconoscimento di tutte le periferiche.

Memoria di Massa

- Le memorie di massa (o memorie secondarie) sono utilizzate per memorizzare grandi volumi di dati in modo persistente.

Memoria Cache

- È una memoria che usa una **tecnologia veloce** SRAM contro una più lenta DRAM della memoria principale.

- Non visibile al software, è **gestita dall'hardware**, che memorizza i dati recenti usati dalla memoria primaria.
-
- La Cache ha la funzione di **aumentare le prestazioni del sistema, riducendo il traffico del bus di sistema e della memoria principale che è uno dei maggiori** colli di bottiglia.

Codici di caratteri

- È un **codice alfabetico**, cioè un insieme di caratteri che può comprendere:
- Caratteri alfabetici minuscoli e maiuscoli (‘a’,…,‘z’,‘A’,…,‘Z’);
- Caratteri numerici (‘0’,…,‘9’)
- Segni di punteggiatura (‘,’, ‘:’, ‘!’, etc.)
- Altri simboli stampabili (‘@’, ‘+’, etc.)
- Caratteri di controllo (NUL, FF, etc.)

- Principali codici di caratteri
  - ASCII
  - UNICODE

Codice ASCII (American Standard Code for Information Interchange)

- Sistema di **codifica dei caratteri**, in cui **ogni simbolo dell'alfabeto** è **codificato** in una **stringa di bit**.
- **ASCII: 7 bit**, 128 parole di codice
- **ASCII esteso (Latin-1): 8 bit,** 256 parole di codice

Codice ASCII (American Standard Code for Information Interchange)

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Sistemi di numerazione

- È un insieme di simboli (cifre) e regole che assegnano ad ogni sequenza di cifre uno e un solo valore numerico. Può essere
  - **non posizionale.** Es. il sistema di numerazione degli antichi romani
  - **posizionale**. Es. il sistema di numerazione decimale

Sistemi di numerazione posizionale

- Dato un numero **B > 1 detto base**
- **Insieme di n simboli**: 0, 1, 2, … , n-1

- Una stringa di n simboli xn-1, xn-2, …., x1, x0 si interpreta come
- xn-1*b^(n-1) + xn-2*b^(n-2) + … x1*b^1 + x0*b^0

- Il valore rappresentato da un simbolo **dipende dalla posizione** del simbolo nella stringa.

Sistema di numerazione Decimale

- La base B è 10
- I simboli sono 0, 1, 2, … , 9

- Esempio: la stringa 2014 rappresenta il numero
- 2*10^3 + 0*10^2 + 1*10^1 + 4*10^0

Sistema di numerazione Binario

- La base B è 2. I simboli sono 0 e 1

- Esempio: la stringa binaria 11110 rappresenta il numero
- 1*2^4+ 1*2^3 + 1*2^2 + 1*2^1 + 0*2^0 (=30 in decimale)

Sistema di numerazione Decimale e Binario

Sistema in base 10	Sistema in base 2
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111

Conversione da Decimale a base B

- La conversione di un numero da base 10 a base B viene effettuata **dividendo ripetutamente** la **parte intera del numero decimale** per la **base B,** e registrando i **resti** delle divisioni **dal basso verso l'alto,** fino a ottenere un quoziente uguale a zero
- ESEMPIO
- **Conversione del numero 7**
- 7/2 = 3 resto 1
- 3/2 = 1 resto 1
- 1/2 = 0 resto 1
- Il numero 7 in binario è 111
-
- **Conversione del numero 14**
- 14/2 = 7 resto 0
- 7/2 = 3 resto 1
- 3/2 = 1 resto 1
- 1/2 = 0 resto 1
- Il numero 14 in binario è 1110

Conversione da base B a Decimale

- Si effettua **moltiplicando ogni cifra** del numero per la sua base (B) elevata alla posizione in cui si trova la cifra.

- Esempio da Binario a Decimale:
- 10 = 1*2^1 + 0*2^0 = 1*2 + 0*1 = 2 + 0 = 2
- 101= 1*2^2 + 0*2^1 + 1*2^0 = 1*4 + 0*2 + 1*1 = 4 + 0 + 1 = 5

Strutture logiche di informazione

- Nella rappresentazione binaria l'unità di informazione elementare è il **bit**.

- **Strutture logiche**
- **Word** dipende dall'architettura
- **Byte** 8 bit
- **Half-Byte** 4 bit

- **Multipli delle strutture logiche**
- 1 Kilobyte (KB) = 2^10 byte =1024 byte circa mille byte
- 1 Megabyte (MB) = 1024 KB circa 1 milione di byte
- 1 Gigabyte (GB) = 1024 MB circa 1 miliardo di byte
- 1 Terabyte (TB) = 1024 GB circa 1000 miliardi di byte
- 1 Petabyte (PB) = 1024 TB circa 1000000 miliardi di byte

Linguaggio di programmazione

- Un linguaggio in informatica è l'**insieme di stringhe** che **godono** di una **certa proprietà**.
- **Grammatica** è un formalismo per definire un linguaggio fornendo un metodo per la costruzione delle stringhe.

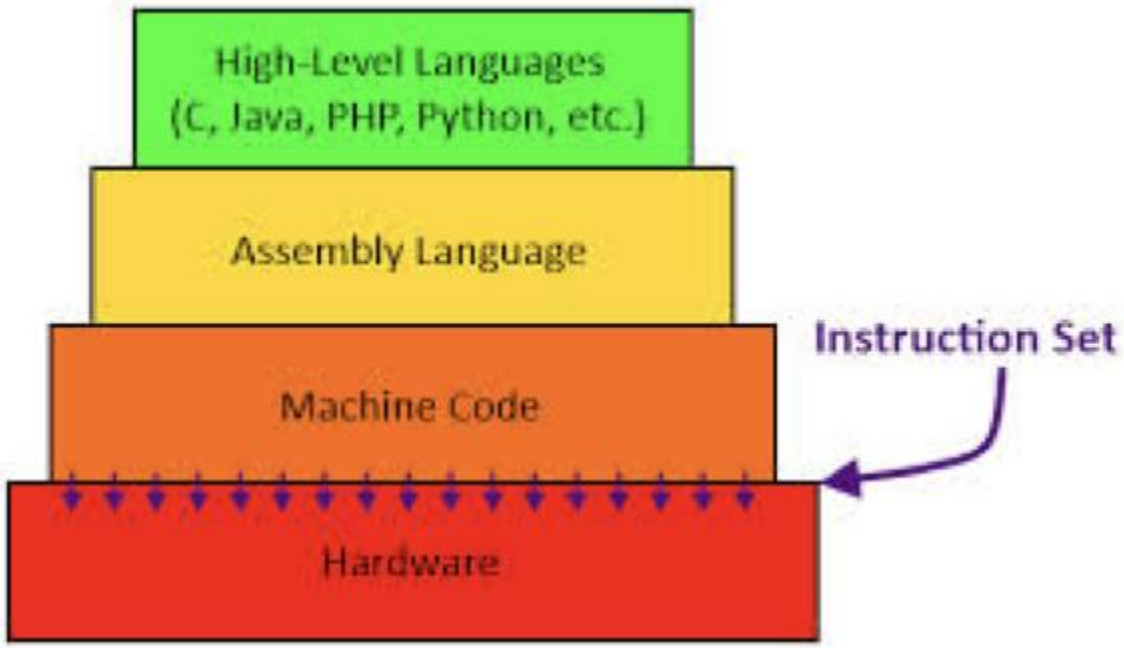
- Una **Grammatica formale** G è una **quadrupla** <**T, N, P, S**>, con
- **T** insieme (alfabeto) finito e non vuoto di simboli terminali
- **N** insieme (alfabeto) finito e non vuoto di simboli non terminali
- **P** (regole di produzione) relazione binaria finita su
- (TUN)* × N × (TUN)* x (TUN)*
- **S** (assioma) è il simbolo non terminale di inizio

- Il linguaggio generato da una grammatica è l'insieme L(G) = {x | x in T*, S →*x}

Gerarchia di livelli

- Un elaboratore è pensato come un sistema gerarchico in cui sono presenti diversi livelli di macchina virtuale interagenti fra loro
- Le diverse macchine virtuali sono via via più astratte, più alto è il livello di una macchina, più vicino è il suo linguaggio alla logica dell'utente e più lontano dalla logica dell'elaboratore reale
- La macchina reale, ovvero il livello hardware, è la macchina più efficiente ma meno flessibile

Gerarchia di livelli



Traduzione di programmi

- Per poter essere eseguito sulla macchina reale (HW), un programma scritto in linguaggio Ln deve essere **tradotto** in un programma (equivalente) scritto in un linguaggio Ln-1, a sua volta tradotto in un programma scritto in linguaggio Ln-2, e così via sino ad ottenere una traduzione in linguaggio L0 (**linguaggio macchina**) che è l'unico comprensibile dalla macchina fisica

- Ogni istruzione in linguaggio Ln è sostituita (tradotta) da una sequenza di istruzioni in linguaggio Ln-1

- Solo programmi scritti in linguaggio L0 possono essere eseguiti direttamente dai circuiti elettronici di cui è composto l'HW, senza bisogno di traduzione

- La traduzione di programmi da un livello più alto al livello sottostante è effettuato da un particolare programma può avvenire secondo due approcci

    - **Compilazione**
    - **Interpretazione**
- ed è eseguito da un programma chiamato rispettivamente **Compilatore** o **Interprete**

Compilazione

- Dato un programma scritto in un linguaggio di programmazione (programma sorgente), ogni sua istruzione è tradotta in linguaggio macchina
- **Viene generato** un altro programma, detto **programma oggetto**, che potrà essere successivamente eseguito
- Vengono rilevati **solo gli errori sintattici**. Gli errori semantici saranno rilevati in fase di esecuzione

Interpretazione

- Dato un programma scritto in un linguaggio di programmazione (programma sorgente), ogni sua istruzione è tradotta in linguaggio macchina ed **immediatamente eseguita**
- **Non viene generato** il **programma oggetto**
- Vengono rilevati **sia gli errori sintattici che semantici**

Interpretazione e Compilazione

- **Velocità di esecuzione**
- Bassa per i linguaggi interpretati
- Alta per i linguaggi compilati

- **Facilità di messa a punto dei programmi**
- Alta per i linguaggi interpretati
- Bassa per i linguaggi compilati

Linguaggi di programmazione di basso livello

- **Linguaggio macchina**
- Un’istruzione è una stringa di bit che viene interpretata ed eseguita dalla CPU
- Esempio di istruzione: 0100100100101000

- **Linguaggio Assembly**
- Un’istruzione è una stringa alfanumerica che viene tradotta in una corrispondente istruzione in linguaggio macchina
- Esempio di istruzione: MOV AX,12