

Introduzione alla Programmazione

claudio.lucchese@unive.it

Motto: "Talk is cheap. Show me the code!"

```
1  /* Il mio primo programma */  
2  
3  #include <stdio.h>  
4  
5  int main(void)  
6  {  
7      printf("Hello, world!\n");  
8  
9      return 0;  
10 }
```

try it now! (eg. by using replit.com)

Motto: "Talk is cheap. Show me the code!"

```
1  /* Il mio primo programma */
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7      printf("Hello, world!\n");
8
9      return 0;
10 }
```

Commento: ignorato dal compilatore

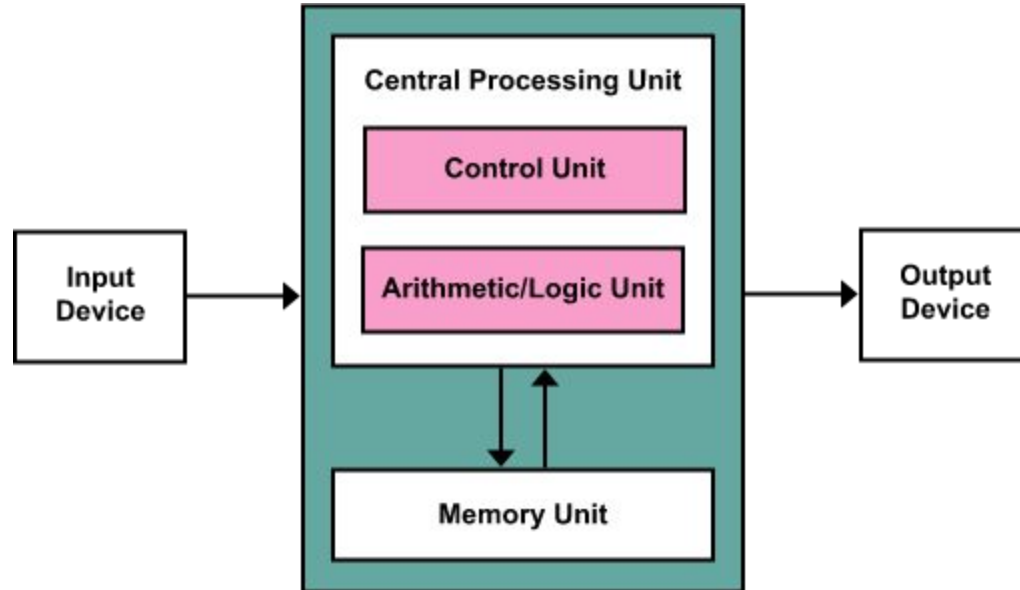
include: uso di librerie esterne (I/O)

funzione main: punto di ingresso del programma; vedremo dopo le funzioni

Scrittura nello stream di output:
uso di una funzionalità esterna

codice di uscita:
il programma termina correttamente>

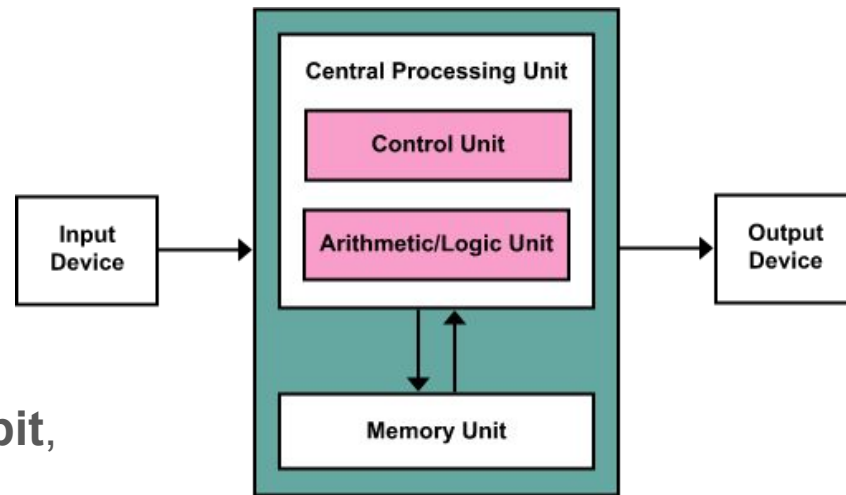
von Neumann Architecture



Non vogliamo scrivere programmi diversi per ogni hardware esistente.
Scriviamo programmi per una **macchina astratta**.

von Neumann Architecture

- La memoria contiene le informazioni necessarie al programma
 - (e il programma stesso)
- La memoria è una lunga sequenza di **bit**, dove ognuno memorizza **0** o **1**
- Un **bit** (**b**inary **d**igit) è l'unità di informazione di un computer
- Per convenzione, la memoria è organizzata in gruppi di 8 bit, detti **byte**



MEMORIA: 1010101001011010101101001110101010101001110100101

- Es. per sommare due numeri a e b dobbiamo:
 - leggere il numero a memorizzato in posizione x
 - leggere il numero b memorizzato in posizione y
 - calcolare la somma $c = a + b$ e memorizzarla in memoria in posizione z

Come *rappresentare* un numero intero?

Siamo abituati alla notazione decimale (posizionale in base 10):

- il significato di **1024** è $1 \times 10^3 + 0 \times 10^2 + 2 \times 10^1 + 4 \times 10^0$
- (con un alfabeto di 10 simboli)

Con i bit possiamo rappresentare solo 2 simboli: sistema posizionale in base 2

- il significato di 10000000000 è
 $1 \times 2^{10} + 0 \times 2^9 + 0 \times 2^8 + 0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 +$
 $+ 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 1024$

Come rappresentare 57 con 8 bits (1 byte)?

Dobbiamo trovare i
coefficienti 0/1 di

$$\begin{aligned} &?x2^7 + ?x2^6 + ?x2^5 + ?x2^4 + \\ &?x2^3 + ?x2^2 + ?x2^1 + ?x2^0 \end{aligned}$$

Come rappresentare 57 con 8 bits (1 byte)?

Dobbiamo trovare i

coefficienti 0/1 di

$$?x2^7 + ?x2^6 + ?x2^5 + ?x2^4 + \\ ?x2^3 + ?x2^2 + ?x2^1 + ?x2^0$$

Metodo delle sottrazioni

La rappresentazione binaria
di 57 è **00111001**

Input	Powers of 2	Is larger than or equal to 2^i ?	Coefficient
57	$2^7 = 128$	$57 - 128 < 0$	0
57	$2^6 = 64$	$57 - 64 < 0$	0
57	$2^5 = 32$	$57 - 32 = 25 \geq 0$	1
25	$2^4 = 16$	$25 - 16 = 9 \geq 0$	1
9	$2^3 = 8$	$9 - 8 = 1 \geq 0$	1
1	$2^2 = 4$	$1 - 4 < 0$	0
1	$2^1 = 2$	$1 - 2 < 0$	0
1	$2^0 = 1$	$1 - 1 = 0 \geq 0$	1

Domande

- Quante sono le sequenze di 0/1 possibili con 3 bit? e con 8? e con 64?
- Qual è il numero più grande che possiamo rappresentare con 8 bit?
- Quanti bit sono necessari per rappresentare i numeri:
 - 1,3,15,127,255,256,1023,10⁹ ?

- Come possiamo rappresentare i numeri reali? es: 2.5?

Come rappresentare 2.5 con 8 bit ?

Il significato di **2.5** è $2 \times 10^0 + 5 \times 10^{-1}$

Una strategia possibile è di usare
4 bit per la parte intera e
4 bit per la parte decimale

Come rappresentare 2.5 con 8 bit ?

Il significato di **2.5** è $2 \times 10^0 + 5 \times 10^{-1}$

Una strategia possibile è di usare
4 bit per la parte intera e
4 bit per la parte decimale

La rappresentazione binaria
di 2.5 è **00101000**

Input	Powers of 2	Is larger than or equal to 2^i ?	Coefficient
2.5	$2^3 = 8$	$2.5 - 8 < 0$	0
2.5	$2^2 = 4$	$2.5 - 4 < 0$	0
2.5	$2^1 = 2$	$2.5 - 2 = 0.5 \geq 0$	1
0.5	$2^0 = 1$	$0.5 - 1 < 0$	0
0.5	$2^{-1} = 0.5$	$0.5 - 0.5 = 0 \geq 0$	1
0.0	$2^{-2} = 0.25$	--	0
0.0	$2^{-3} = 0.125$	--	0
0.0	$2^{-4} = 0.0625$	--	0

Real numbers to binary

- *“Una strategia possibile è di usare 4 bit per la parte intera e 4 bit per la parte decimale”*
Questa è chiamata rappresentazione a **virgola fissa (Fixed-point)**
- **Attenzione!!** il più piccolo numero rappresentabile è $2^{-4}=1/16=0.0625$
- **Molta Attenzione!!** Possiamo rappresentare solo 2^4 numeri tra 0 e 1! Provate a rappresentare 0.1!

Non solo abbiamo dei limiti rispetto al numero più grande e al numero più piccolo rappresentabile, ma, in questo intervallo, **possiamo rappresentare un numero finito (piccolo) di numeri reali!**

Ogni numero binario è rappresentabile come $\pm 1.????? \times 2^?$

I computer in genere usano rappresentazione **floating point a 32 bit (4 byte)**

- 1 bit per il segno (positivo/negativo)
- 8 bit per l'esponente $2^?$
- 23 bit per la mantissa $1.?????$

Nei floating point a **doppia precisione** vengono usati **64 bit** (più precisa e più costosa)

Come possiamo rappresentare il testo?

Come rappresentare il testo
HELLO WORLD! ?

ASCII (American Standard
Code for Information
Interchange) è una codifica
a 8 bit per il testo.

bits	decimal	Encoded text	bits	decimal	Encoded text	bits	decimal	Encoded text	bits	decimal	Encoded text
00100000	32	space	00110000	48	0	01000000	64	@	01010000	80	P
00100001	33	!	00110001	49	1	01000001	65	A	01010001	81	Q
00100010	34	"	00110010	50	2	01000010	66	B	01010010	82	R
00100011	35	#	00110011	51	3	01000011	67	C	01010011	83	S
00100100	36	\$	00110100	52	4	01000100	68	D	01010100	84	T
00100101	37	%	00110101	53	5	01000101	69	E	01010101	85	U
00100110	38	&	00110110	54	6	01000110	70	F	01010110	86	V
00100111	39	'	00110111	55	7	01000111	71	G	01010111	87	W
00101000	40	(00111000	56	8	01001000	72	H	01011000	88	X
00101001	41)	00111001	57	9	01001001	73	I	01011001	89	Y
00101010	42	*	00111010	58	:	01001010	74	J	01011010	90	Z
00101011	43	+	00111011	59	;	01001011	75	K	01011011	91	[
00101100	44	,	00111100	60	<	01001100	76	L	01011100	92	\
00101101	45	-	00111101	61	=	01001101	77	<	01011101	93]
00101110	46	.	00111110	62	>	01001110	78	N	01011110	94	^
00101111	47	/	00111111	63	?	01001111	79	O	01011111	95	_

Risposta: 01001000 01000101 01001100 01001100 01001111 00100000
01010111 01001111 01010010 01001100 01000100 00100001

Come rappresentare un'immagine?

Immaginiamo di avere un'immagine da 12 MegaPixel:

- Ogni pixel codifica un colore
- Il colore è codificato come intensità dei colori primari: Red, Green, Blue
- Ogni colore primario è rappresentato con un intero da 0 a 255
 - 256 valori possibili
 - 8 Bit = 1 byte
 - la tripla (0,0,0) significa nero, mentre (255,255,255) significa bianco
- Abbiamo bisogno di 24 bit = 3 byte per pixel
 - $2^{24} \sim 16$ milioni di colori
 - $3 \times 12\text{M} = 36$ MB per immagine non compressa
 - (per fortuna esiste jpeg!)

Quanto è grande la memoria?

Un PC ha generalmente 8GB di memoria:

- 1 Petabyte (PB) = 1000 Terabyte (TB)
- 1 Terabyte (TB) = 1000 Gigabyte (GB)
- **1 Gigabyte (GB) = 1000 Megabyte (MB)**
- 1 Megabyte (MB) = 1000 Kilobyte (KB)
- 1 Kilobyte (KB) = 1000 Byte

8 GB = 8×10^9 byte = 2 miliardi di numeri floating point a singola precisione

8 GB ~ 222 foto non compresse da 16Mpixel

Quanto è grande un Terabyte?

1 TERABYTE

A \$200 hard drive that holds 260,000 songs.

20 TERABYTE

Photos uploaded to Facebook each month.

120 TERABYTE

All the data and images collected by the Hubble Space Telescope.

330 TERABYTE

Data that the large Hadron collider will produce each week.

460 TERABYTE

All the digital weather data compiled by the national climate data center.

530 TERABYTE

All the videos on Youtube.

600 TERABYTE

ancestry.com's genealogy database (includes all U.S. census records 1790-2000)

1 PETABYTE

Data processed by Google's servers every 72 minutes.

Domande

- Per risolvere un dato problema, quali sono le informazioni necessarie e come strutturarle in memoria?

Per adesso ci limitiamo ad una domanda più semplice e basilare:

- **Come possiamo memorizzare un'informazione in memoria, deciderne la codifica e la posizione (o indirizzo)?**

Variabili

Def: una variabile è un'informazione (**valore o oggetto**) memorizzata in una specifica locazione di memoria, con un **tipo** che ne stabilisce le codifica e le operazioni possibili e con un **identificatore**.

```
1  /* Il mio primo programma */  
2  
3  #include <stdio.h>  
4  
5  int main(void)  
6  {  
7      int a = 10;  
8  
9      return 0;  
10 }
```

un **identificatore** è formato da lettere, cifre e *underscore*, il primo carattere non può essere una cifra. Il linguaggio è case sensitive!

I tipi fondamentali

1.	char	1 Byte	da -128 a +127
2.	signed char	1 Byte	da -128 a +127
3.	unsigned char	1 Byte	da 0 a 255
4.	signed short int o short int o short	2 Byte	da -32768 a 32767
5.	unsigned short int o unsigned short	2 Byte	da 0 a 65535
6.	signed int o int	4 Byte	da -2147483648 a +2147483647
7.	unsigned int o unsigned	4 Byte	da 0 a 4294967295
8.	signed long int o long int o long	8 Byte	da -2^{63} a $2^{63}-1$
9.	unsigned long int o unsigned long	8 Byte	da 0 a $2^{64}-1$
10.	float	4 Byte	da -3.40×10^{-38} a 3.40×10^{38}
11.	double	8 Byte	da -1.80×10^{-308} a 1.80×10^{308}
12.	long double	dipende dall'implementazione ...	

Assegnamento

L'**espressione di assegnamento** : `identificatore = espressione`

permette di leggere e modificare il contenuto (*nella locazione di memoria in corrispondenza*) di una variabile

```
1  /* Il mio primo programma */
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7      int a = 10, b;
8
9      b = 66;
10
11     return 0;
12 }
```

Altre espressioni

Le espressioni sono combinazioni valide di costanti, variabili, operatori e chiamate di funzione.

Costanti numeriche: 10 (int) 10u (unsigned) 10l (long)
 10.0 (double) 10.0f (float) 10.0d (double)

Operatori aritmetici: () + - * / %

Operatori relazionali e di uguaglianza: < <= > >= == !=

Operatori logici: && || !

Per ogni operatore è definita la precedenza e l'associatività.

Es. di espressione: (-2 + (2*3-4*3+2)/7) / (2*1)

Esercizi

1. Calcolare la circonferenza di un cerchio dato il suo raggio
2. Date due variabili intere A e B, scambiarne il contenuto.
3. Dati 3 interi corrispondenti a ore, minuti e secondi, visualizzare il numero di secondi trascorsi dall'inizio della giornata
4. Qual è il passo in minuti/Km sufficiente per battere l'attuale record della maratona?