



# Introduzione al Linguaggio C

Corso di Informatica  
Laurea in Fisica

prof. ing. Corrado Santoro

A.A. 2009-10

# Dai flow-chart ai linguaggi ...



- Abbiamo imparato ad usare uno strumento **formale** – i flowchart – per rappresentare un algoritmo
- Tuttavia un flowchart **non è direttamente eseguibile** da un calcolatore
- Occorre pertanto **tradurre** il flowchart in un **linguaggio** che sia comprensibile da una CPU
- La traduzione deve essere possibile, cioè il nuovo linguaggio deve possedere i costrutti:
  - Start/Stop
  - Input, Output
  - Calcolo
  - Confronto

# Ma cos'è un linguaggio di programmazione?



- E' un insieme di
  - Simboli
  - Parole-chiave (keyword)
- Ogni keyword/simbolo ha un significato ben preciso (**semantica**)
- Le keyword/simboli possono combinarsi fra loro secondo una **sintassi** ben precisa

# Tanti linguaggi ...



- Esistono **centinaia** di linguaggi di programmazione  
→ un certo algoritmo (flowchart) può essere tradotto in uno qualunque di questi linguaggi
- Tuttavia, il programma per essere eseguibile, deve essere scritto in un linguaggio comprensibile dalla CPU
- Questo vuol dire che ogni CPU conosce **centinaia** di linguaggi di programmazione?

# ... un solo linguaggio.



- **NO!**
- Ogni CPU conosce **solamente un linguaggio** di programmazione:
  - Imposto/progettato dal produttore della CPU
  - Strettamente specifico per quel tipo o famiglia di CPU
  - Composto da istruzioni **estremamente primitive** in grado di
    - operare su dati di dimensione massima prefissata
    - eseguire letture e scritture di locazioni di memoria
    - eseguire letture e scritture dalle periferiche di I/O
- Il linguaggio della CPU è denominato **linguaggio/codice macchina o assembly**

# Tanti linguaggi, un solo linguaggio

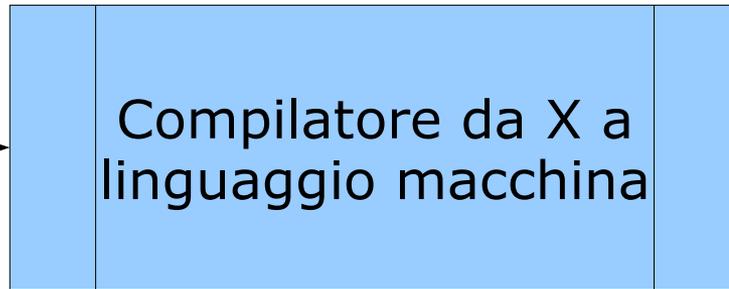
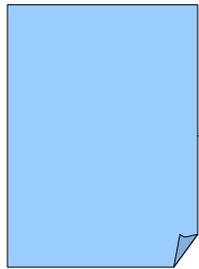


- Tanti linguaggi → linguaggio macchina: come funziona questo meccanismo??
- Per ogni linguaggio X esiste un software che permette di tradurre in linguaggio macchina qualunque programma scritto in X
- Questi software di traduzione appartengono a due differenti categorie:
  - **Compilatori**
  - **Interpreti**

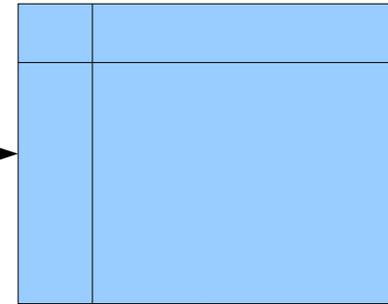
# Compilatori



File di testo "sorgente",  
contenente il programma  
scritto nel linguaggio X



File binario "eseguibile"  
dalla CPU



CPU

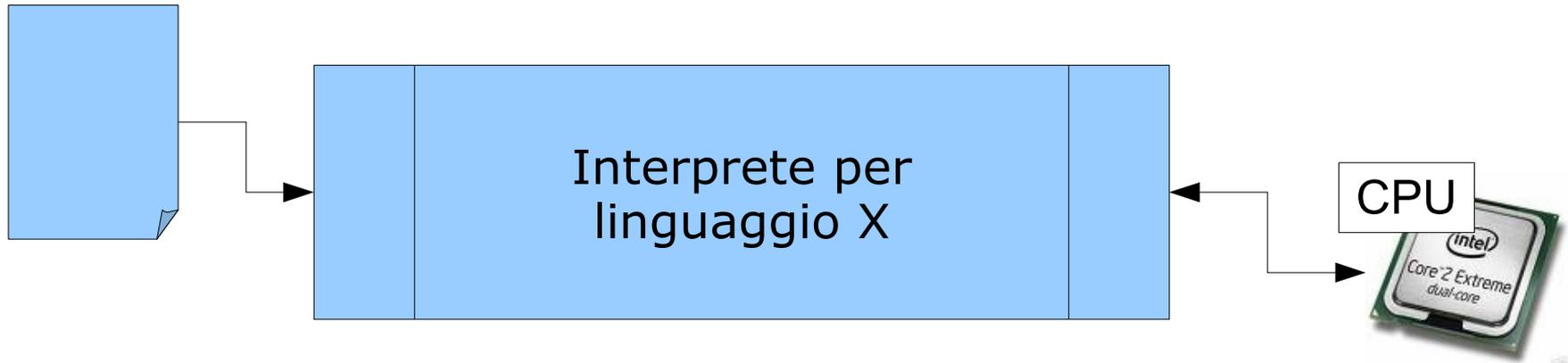


- Il file sorgente viene tradotto, istruzione per istruzione, in linguaggio macchina
- Il risultato è un file "eseguibile", composto solo da istruzioni in linguaggio macchina, che può essere direttamente eseguito dalla CPU

# Interpreti



File di testo "sorgente",  
contenente il programma  
scritto nel linguaggio X



- Ogni comando del file sorgente viene tradotto "on-the-fly" in linguaggio macchina ed eseguito immediatamente
- Ogni volta che occorre eseguire di nuovo quel programma, occorre rifare sempre la traduzione

# Il Linguaggio C: struttura di un programma



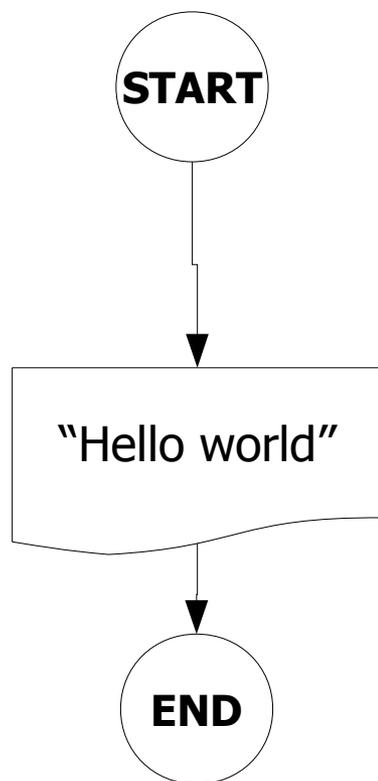
- Un programma C è un insieme di **FUNZIONI** che implementano l'algoritmo da realizzare
- Ogni funzione è composta da un insieme di comandi primitivi del linguaggio e da invocazioni (utilizzi) di altre funzioni esistenti nel programma stesso
- **Dichiarazione di una funzione**

```
NOME_FUNZIONE (ELENCO PARAMETRI)  
{  
    CORPO DELLA FUNZIONE  
}
```

- **Use (invocazione di una funzione)**

```
NOME_FUNZIONE (ELENCO PARAMETRI) ;
```

# Verso il linguaggio C, il primo programma



hello.c

```
#include <stdio.h>

main ()
{
    printf(\"Hello, world\\n\");
    getchar();
}
```

## Proviamolo con il Dev-C++