

Laboratorio di Calcolo

Linguaggi di programmazione

DIPARTIMENTO DI FISICA



SAPIENZA
UNIVERSITÀ DI ROMA

Software

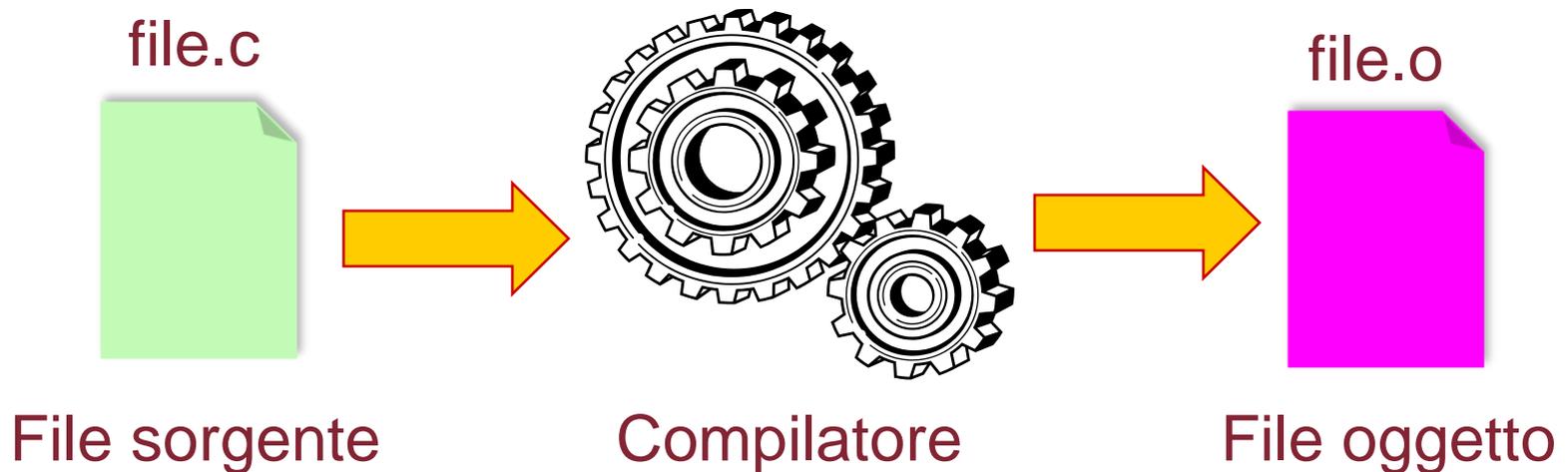
- Insieme di istruzioni e dati trattati dal computer
- Le istruzioni possono essere a diversi livelli
 - Istruzioni macchina: insieme limitato di funzioni specifiche della CPU
 - Leggi un byte da una locazione di memoria
 - Somma tra loro il contenuto di due registri
 - Esegui le istruzioni a partire dalla locazione m
 - Istruzioni di alto livello: funzioni tradotte in sequenze di istruzioni in linguaggio macchina prima di essere eseguite

Linguaggi di alto livello

- Linguaggi sintatticamente complessi **facili** da usare dagli umani: la traduzione in linguaggio macchina viene fatta “dietro le quinte”
- Si possono classificare in procedurali (C, FORTRAN, PASCAL) e Object Oriented (C++,JAVA)

Compilatori

- Traduttori da linguaggio di alto livello a linguaggio macchina
- Linguaggi evoluti
 - Versatilità, Semplicità, Flessibilità
 - **Indipendenza** dalla macchina



Linguaggio Macchina

- A ogni istruzione è associato un codice di N bit
- La CPU comincia caricando il byte presente in una locazione predeterminata e lo interpreta come istruzione
- Si interpretano i byte successivi secondo l'istruzione

Esempio (1)

Consideriamo la somma di due numeri

$3 + 2$ (che fa 5, lo sappiamo)

meglio rappresentarlo come

A vale 3

B vale 2

$A + B$

(perchè posso variare i valori di A e B)

A e B sono **variabili** contenute in registri di memoria

Esempio (2)

Le operazioni precedenti si possono rappresentare con **istruzioni**

| | |
|-----------------|--|
| set_A, X | a vale X |
| set_B, X | b vale X |
| sum_AB | somma A e B, A contiene la somma |
| out_A | scrive la somma nella memoria grafica, mostrandola all'utente |

Esempio (3)

- Supponiamo che i seguenti codici binari corrispondano alle seguenti istruzioni

| | | |
|-----|----------|--|
| 001 | set_A, X | Carica il contenuto X della locazione di memoria successiva nel registro A |
| 010 | set_B, X | Carica il contenuto X della locazione di memoria successiva nel registro B |
| 011 | sum_AB | Somma i contenuti dei registri A e B e metti il risultato in A |
| 100 | out_A | Scrivi nella memoria grafica il contenuto del registro A |

Esempio

- Ed ora supponiamo che la memoria del nostro computer sia così riempita:

| Address | Content |
|---------|---------|
| 001 | 001 |
| 010 | 011 |
| 011 | 010 |
| 100 | 010 |
| 101 | 011 |
| 110 | 100 |
| 111 | 000 |

Esempio

- Ed ora supponiamo che la memoria del nostro computer sia così riempita:

| Address | Content |
|---------|------------|
| 001 | 001 |
| 010 | 011 |
| 011 | 010 |
| 100 | 010 |
| 101 | 011 |
| 110 | 100 |
| 111 | 000 |

001 viene interpretato come Un'istruzione e quindi come `set_A, X`.

Di conseguenza il byte successivo contiene il valore di X.

Esempio

- Ed ora supponiamo che la memoria del nostro computer sia così riempita:

| Address | Content |
|---------|------------|
| 001 | 001 |
| 010 | 011 |
| 011 | 010 |
| 100 | 010 |
| 101 | 011 |
| 110 | 100 |
| 111 | 000 |

011 (3) viene dunque caricato nel registro A – e' un DATO

Esempio

- Ed ora supponiamo che la memoria del nostro computer sia così riempita:

| Address | Content |
|---------|------------|
| 001 | 001 |
| 010 | 011 |
| 011 | 010 |
| 100 | 010 |
| 101 | 011 |
| 110 | 100 |
| 111 | 000 |

010 viene nuovamente interpretato come una istruzione e dunque come **set_B, X**. Nel registro B viene caricato il numero 010 (2)

Esempio

- Ed ora supponiamo che la memoria del nostro computer sia così riempita:

| Address | Content |
|---------|------------|
| 001 | 001 |
| 010 | 011 |
| 011 | 010 |
| 100 | 010 |
| 101 | 011 |
| 110 | 100 |
| 111 | 000 |

011 rappresenta l'istruzione di somma del contenuto dei registri A e B. In A finisce il valore 5 (3+2) che si scrive 101 e che non si vede qui

Esempio

- Ed ora supponiamo che la memoria del nostro computer sia così riempita:

| Address | Content |
|---------|------------|
| 001 | 001 |
| 010 | 011 |
| 011 | 010 |
| 100 | 010 |
| 101 | 011 |
| 110 | 100 |
| 111 | 000 |

l'istruzione di somma non ha parametri perciò il byte successivo è ancora un'istruzione: **outA** (codice 100)

Esempio (4)

Quindi abbiamo visto la traduzione dal linguaggio di alto livello (**set_A**, **X** oppure **sum_AB**) al linguaggio macchina (010 oppure 011).

Questa traduzione è fatta dal **compilatore**.

Noi scriviamo istruzioni di alto livello e poi le compiliamo.

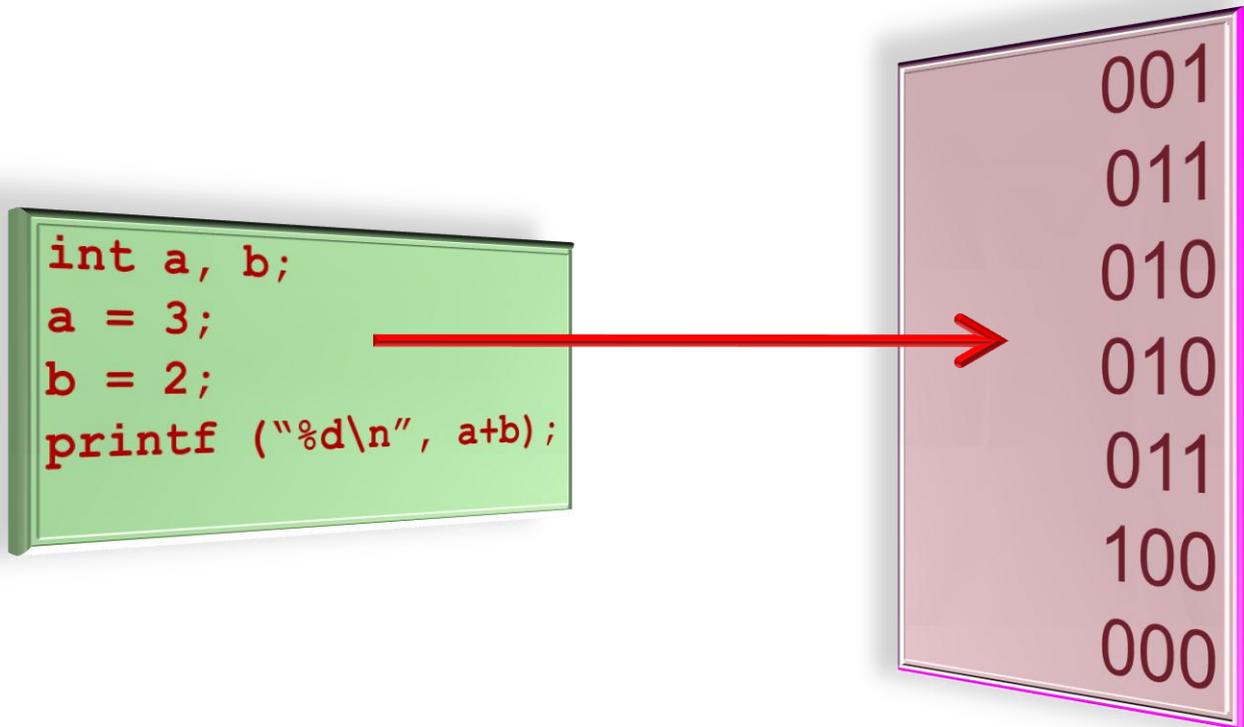
Esempio (5)

Il livello di un linguaggio è un concetto **relativo**

Un esempio parziale in C:

```
int a, b;  
a = 3;  
b = 2;  
printf ("%d\n", a+b);
```

Compilazione



Traduzione da linguaggio ad alto livello a linguaggio macchina

Linker

- Compilare NON BASTA
- Uno o più file oggetto si devono *collegare* alle *librerie* (*linking*)
- **Librerie**: insieme di file oggetto preparati dal produttore del compilatore con le istruzioni per l'esecuzione di compiti comuni
 - Librerie matematiche
 - Librerie grafiche
 - Servizi di I/O
- L'operazione di *link* può essere esplicita o implicita