

Informatica

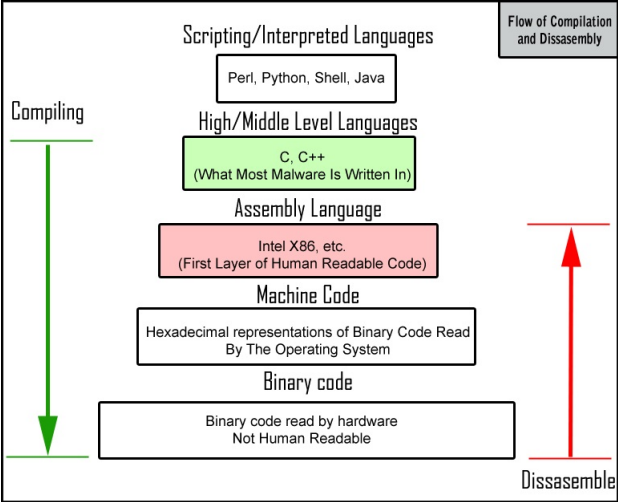
Dipartimento di Economia

Ing. Cristiano Gregnanin

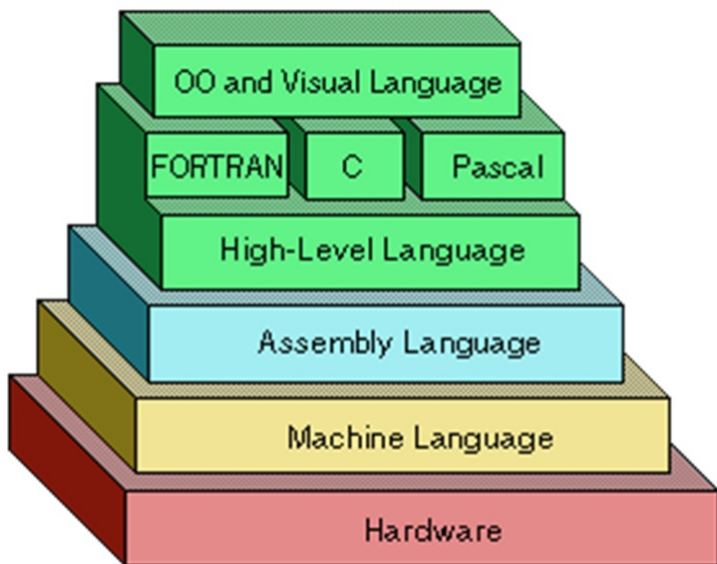
Corso di laurea in Economia

20 ottobre 2016

Linguaggi



Linguaggi



Linguaggi di alto livello

Si basano su una **macchina virtuale** le cui *mosse* non sono quelle della macchina hardware

Astrazione

Esistono linguaggi a vari livelli di astrazione:

- ▶ Linguaggio macchina: implica la conoscenza dei metodi di rappresentazione delle informazioni utilizzati.
- ▶ Linguaggio macchina e assembler: implica la conoscenza dettagliata delle caratteristiche della macchina (registri, dimensioni dati, set di istruzioni). **semplici** programmi implicano la specifica di molte istruzioni.
- ▶ **linguaggi di alto livello:** Il programma può astrarre dai dettagli legati all'architettura. **Sono indipendenti dalla macchina hardware sottostante. La macchina hardware si dice che è astratta**

Astrazione

- ▶ **Linguaggio macchina:**

0100 0000 0000 1000

Difficile da leggere e scrivere.

- ▶ **Linguaggio macchina e assembler:**

LOADA H

LOADB Z

ADD

..

Le istruzioni corrispondono univocamente a quelle macchina, ma vengono espresse tramite nomi simbolici (parole chiave).

- ▶ **linguaggi di alto livello:**

print('hello world')

Sono indipendenti dalla macchina hardware.

Realizzazione

Il calcolatore capisce esclusivamente il linguaggio macchina. Per utilizzare un linguaggio di alto livello occorre scrivere il programma con un editor di testo e salvarlo in un file. Tale file prende il nome di **file sorgente**

Realizzazione

Successivamente bisogna fare in modo che il calcolatore capisca ciò che è stato scritto nel **file sorgente**. Esistono **2 modi** per farlo:

- ▶ attraverso **l'interpretazione** del programma
- ▶ attraverso **la compilazione** del programma

Interprete vs compilatore

L'interprete è *un programma* che legge il file sorgente e lo **esegue** istruzione per istruzione.

finché non è finito il file sorgente:

1. **leggi** un'istruzione dal file sorgente
2. **traduci** l'istruzione in linguaggio macchina
3. **esegui** l'istruzione tradotta. (esecuzione di codice binario)

Interprete vs compilatore

Il compilatore è *un programma* che legge il file sorgente e lo **traduce** in linguaggio macchina, *salvandolo* in un altro file detto **compilato**

finché non è finito il file sorgente:

1. **leggi** un'istruzione dal file sorgente
2. **traduci** l'istruzione in linguaggio macchina
3. **SALVA** l'istruzione tradotta nel file compilato.

Alla fine del processo si otterrà un file detto **compilato**

Esecuzione di un programma

Per eseguire sulla macchina hardware un programma scritto in un linguaggio di alto livello è necessario tradurre il programma in sequenze di istruzioni di basso livello, direttamente eseguite dal processore attraverso:

- ▶ interpretazione (Basic, php, python, matlab...)
- ▶ compilazione (C, C++...)

Come sviluppare un programma

Qualunque sia il linguaggio di programmazione scelto occorre scrivere il testo del programma e memorizzarlo in un file. (*fase di sviluppo.*)

Successivamente se il linguaggio è **compilato** allora bisogna compilare il file sorgente ed eseguire il file *compilato*. **Mentre** se il linguaggio è **interpretato** è sufficiente usare l'interprete per eseguire il programma.

Ambienti di programmazione

è l'insieme dei programmi che consentono la scrittura, la verifica e l'esecuzione di nuovi programmi (fasi di sviluppo).

Sviluppo di un programma: Affinché un programma scritto in un qualsiasi linguaggio di programmazione sia comprensibile (e quindi eseguibile) da un calcolatore occorre tradurlo dal linguaggio originario al linguaggio della macchina. Tale operazione è svolta da speciali programmi detti **traduttori**

Traduzione di un programma

Il **traduttore** converte il testo del file sorgente nella corrispondente rappresentazione in linguaggio macchina. (programma eseguibile)

Sviluppo di programmi

I **traduttori** possono operare traducendo ed eseguendo immediatamente ogni singola istruzione del file sorgente (*interpretati*), **oppure** possono tradurre l'intero programma (*senza eseguirlo*) e producono il programma convertito in linguaggio macchina.

Osservazione: *in generale l'esecuzione di programmi compilati è più rapida.*

riassumendo: compilato vs interpretato

I **compilatori** traducono automaticamente un programma dal linguaggio L a quello macchina

Gli **interpreti** sono programmi capaci di eseguire direttamente un programma in un linguaggio L istruzione per istruzione.

Ambienti di programmazione

riassunto: compilazione

Il compilatore opera la traduzione di un programma sorgente in un **programma oggetto** direttamente eseguibile dal calcolatore.

PRIMA si traduce **TUTTO** il programma **POI** si esegue la versione tradotta.

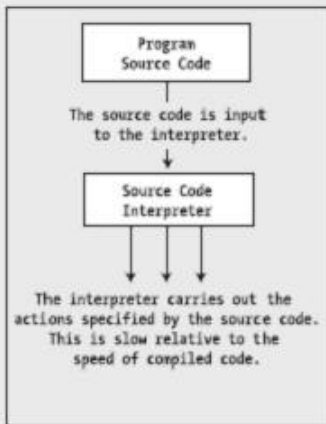
A volte il programma oggetto è composto da più moduli, dove ogni modulo è un programma compilato. Il *linker* provvede a collegarli fornendo un unico programma eseguibile.

Ambienti di programmazione

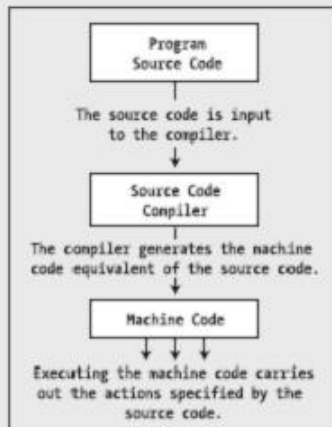
Riassumendo: interpretazione

L'interprete traduce ed esegue direttamente ciascuna istruzione del file sorgente, istruzione per istruzione. E' alternativo al compilatore. La traduzione e l'esecuzione sono due fasi immediatamente consecutive.

Compilato vs interpretato



Interpreted Program Execution



Compiled Program Execution