I LINGUAGGI DI PROGRAMMAZIONE

In informatica, un **linguaggio di programmazione** può essere descritto come un **insieme di regole**, che caratterizzano **sintassi, semantica e lessico** ben definiti. Tali formalismi sono indispensabili per l'implementazione di un canale di **comunicazione tra uomo e computer**.

Il **linguaggio macchina** è quello a più basso livello, fatto di sequenze binare (uno e zero) che compongono le istruzioni per il microprocessore.

Subito sopra questo livello abbiamo i linguaggi detti **Assembly** o **Assembler** con una impropria confusione tra linguaggio e traduttore (che vedremo a breve). L'Assembly fa una mappatura praticamente 1:1 con il linguaggio macchina, permettendoci di scrivere in forma mnemonica quelle istruzioni binarie, molto più complicate da decifrare.

Esempio di codice realizzato con il linguaggio Assembler

```
$MOD8253
DSEG
                ORG     20h
Var1            DS      1
STATE           BIT     Var1.0
OUTPUT          BIT     P1.0

CSEG
                ORG     0h
                AJMP    START

                ORG     0Bh
                AJMP    INTERRUPT

START           MOV     IE,#82h
                MOV     TMOD,#01
                MOV     TH0,#FEh
                MOV     TL0,#0Ch
                SETB    STATE
                SETB    TR0

LOOP            NOP
                SJMP    LOOP

INTERRUPT       CLR     TR0
                MOV     TH0,#FEh
                MOV     TL0,#0Ch
                SETB    TR0
                CPL     STATE
                MOV     C,STATE
                MOV     OUTPUT,C
                RETI
                END
```

Tra i suoi pro vi sono lo stretto legame con l'architettura del computer, la velocità e la potenza, mentre tra i contro si annoverano la lunghezza dei programmi, unitamente alla loro difficoltà di definizione e messa a punto.

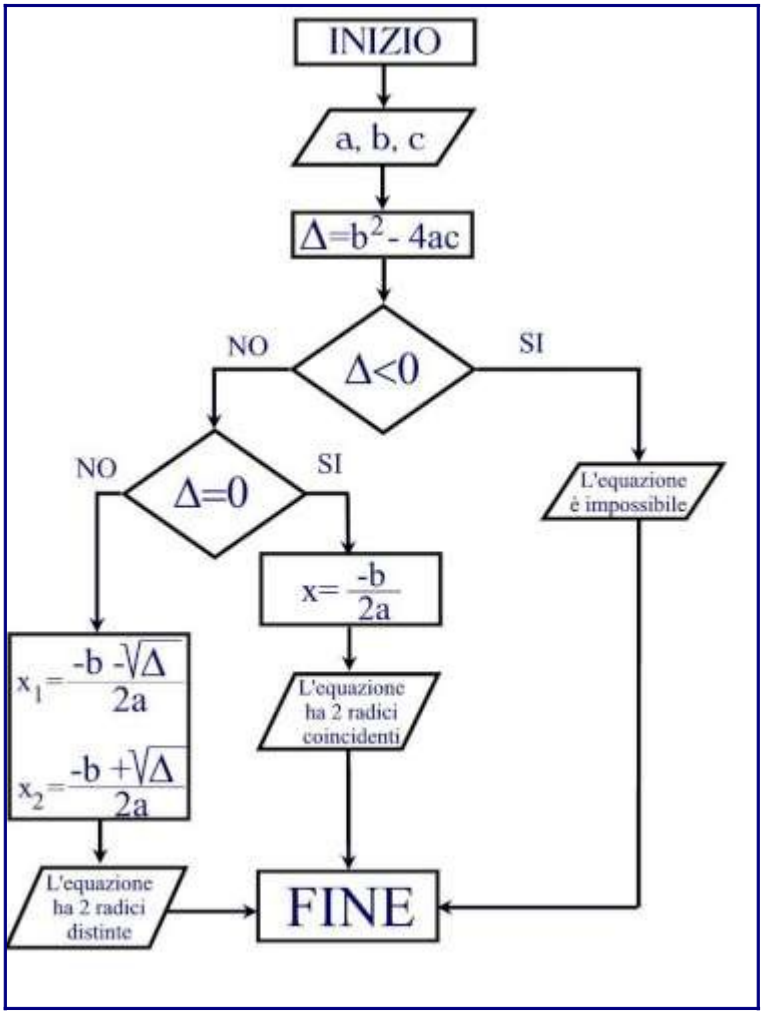
Programmi e algoritmi

Un **programma** altro non è che una implementazione di un certo algoritmo (descritto da un **diagramma di flusso**), tramite un linguaggio eseguibile appunto su un computer.

Fonte: *http://www.html.it*

I LINGUAGGI DI PROGRAMMAZIONE

Esempio di diagramma di flusso



Un linguaggio di programmazione è quindi un **tramite fra linguaggio macchina e linguaggio naturale**. Sua peculiarità principale è perciò di essere **privo di ambiguità** e di descrivere la soluzione di un problema secondo **criteri rigorosi**.

Traduttori e compilatori

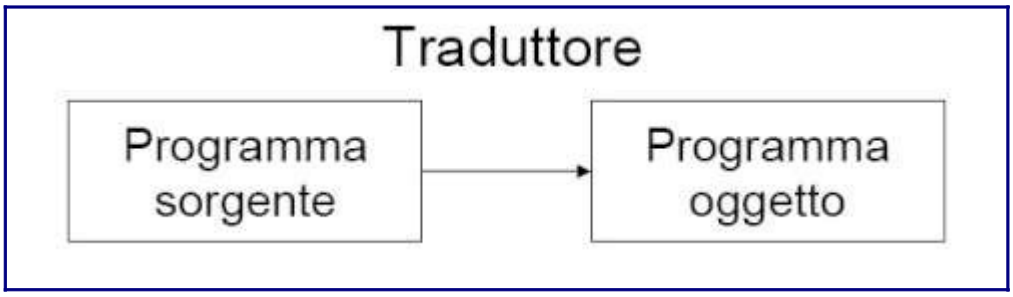
Il linguaggio macchina è l'unico direttamente interpretabile dal microprocessore, mentre qualsiasi altro linguaggio di programmazione più evoluto necessita di un **traduttore**.

Ulteriore distinzione può essere fatta tra **programma sorgente**, realizzato con un linguaggio di programmazione, e **programma oggetto** , realizzato invece in linguaggio macchina.

Il ruolo principale del traduttore

Fonte: *http://www.html.it*

I LINGUAGGI DI PROGRAMMAZIONE



Il traduttore riveste pertanto il ruolo di software che deve **tradurre un programma sorgente in un programma oggetto** per renderlo interpretabile dal microprocessore. Sono individuabili tre tipi di traduttore: **assemblatori, compilatori e interpreti**.

Gli assemblatori hanno a che fare con i linguaggi a basso livello orientati alla macchina (Assembler), mentre compilatori e interpreti concernono i linguaggi ad alto livello orientati all'uomo. Tutti hanno il compito di **fornire in uscita istruzioni in linguaggio macchina** adatte ad essere coerentemente eseguite dall'unità centrale di elaborazione (CPU).

Nel linguaggio Assembler ogni istruzione corrisponde in modo quasi biunivoco a una sola istruzione nel linguaggio macchina. In un **linguaggio ad alto livello** a una istruzione corrispondono invece tipicamente moltissime istruzioni-macchina, rendendo pertanto il lavoro di traduzione più articolato.

Le caratteristiche principali di un linguaggio ad alto livello sono:

- Migliore comprensione per un programmatore in quanto più vicino a quello naturale
- Indipendenza dall'architettura del computer, senza dover conoscere le specifiche del processore
- Individuazione di un linguaggio secondo determinati parametri

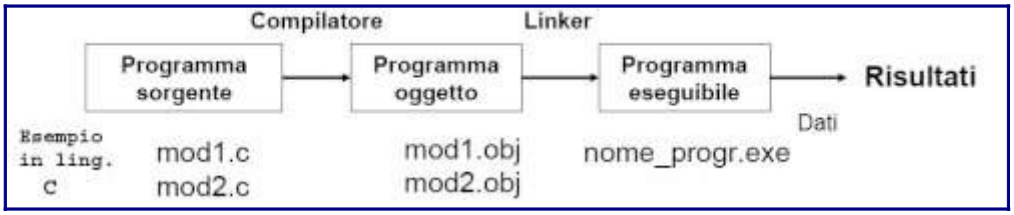
Un medesimo programma è eseguibile su computer diversi (e diverse CPU), utilizzando un compilatore dedicato.

Compilatori e interpreti sono software forzatamente **più complessi** rispetto agli assemblatori e le loro specifiche sono importanti per migliorare l'**efficienza del programma finale**.

Il **compilatore** riceve in ingresso (input) il codice sorgente del programma (file) e ha il compito di restituire in uscita (output) il cosiddetto programma oggetto composto da diversi elementi.

Un ulteriore strumento, il **linker**, ha poi il compito di collegare tra loro i vari moduli prodotti dal compilatore per ottenere un solo programma eseguibile, pronto per essere lanciato dall'utente.

Esempio di compilatore



Fonte: *http://www.html.it*