

# CATENA PROGRAMMATIVA

## *Linguaggi:*

Linguaggio **macchina**:

- Scritto in binario
- Direttamente comprensibile dal calcolatore
- Attività di programmazione più lunga
- Facile commettere errori

Linguaggio **assembler**:

- Scritto in Assembly
- E' una rappresentazione del linguaggio macchina
- Più comprensibile per l'utente rispetto a linguaggio macchina grazie all'utilizzo di simboli e non sequenze di bit
- Tradotto dall'assemblatore in linguaggio macchina

Linguaggi ad **alto livello**:

- Tradotti dal compilatore in Assembler
- Più comprensibile per l'utente rispetto ad assembler e di conseguenza del macchina - Indipendenti dalle caratteristiche dell'architettura su cui il programma sarà eseguito (portabilità)
- Permettono utilizzo di librerie di funzionalità già scritte (riusabilità del codice)

## *Step Catena Programmatica:*

### *1. Compilatore*

Un programma ad **alto livello** viene tradotto nel linguaggio **assembly** utilizzando il compilatore.

### *2. Assemblatore*

Dopo la fase di compilazione, il programma scritto in linguaggio **assembly** (definibile come **file sorgente**) viene tradotto in linguaggio **macchina** (definibile come **file oggetto**) dall'assemblatore.

Dunque l'Assemblatore gestisce:

- Etichette
- pseudoistruzioni
- Numeri in base diverse

Ogni lettura del programma sorgente è chiamata **passo** e l'assemblatore è chiamato a tradurre **due** passi.

Questo è dovuto alle etichette di salto che generano il problema dei riferimenti in avanti, ossia, riferimenti ad etichette successive o contenute in altri file.

Le etichette esterne a un modulo rimangono non risolte, perché di queste se ne occuperà il linker.

### 3. Linker

Inserisce in memoria in modo simbolico il **linguaggio macchina** e i moduli dati: -

Determina gli indirizzi dei dati e delle etichette che compaiono nelle istruzioni -

Corregge i riferimenti interni ed esterni

- Risolve i riferimenti in sospeso, ovvero alle etichette esterne

Genera il **file eseguibile**.

### 4. Loader

1. Leggendo l'intestazione del **file eseguibile** determina la lunghezza del segmento di testo (istruzioni) e del segmento dati (variabili).
2. Crea lo Spazio.
3. Copia le istruzioni e dati dal **file eseguibile** in memoria.
4. Copia nello stack degli eventuali parametri passati al programma principale.
5. Inizializza i registri e imposta lo **stack pointer** affinché punti alla prima locazione libera.
6. Passa al main la quale copia i parametri nei registri argomento e chiama la procedura principale del programma.
7. Quando Il main restituisce il controllo, la procedura di startup **termina** il

programma con una chiamata alla funzione di sistema exit.