



Il sistema operativo

Si tratta di un insieme di programmi in linguaggio macchina che hanno lo scopo di semplificare la vita del programmatore e/o dell'utente; queste sono le funzioni principali di un Sistema Operativo:

- **Gestione dei periferici**; il SO fornisce i **drivers** per la comunicazione con i vari dispositivi installati nel sistema e gestisce i **buffers di comunicazione**. Inoltre fornisce all'utente una **libreria di funzioni** che consentono di **accedere ai dispositivi esterni senza conoscerne i dettagli** (HAL).
- **Gestione della CPU**; una particolare componente del SO (**scheduler**), utilizzando un **timer** ed una linea di **interrupt**, comanda l'esecuzione di diversi programmi per intervalli di tempo dipendenti dalla priorità relativa (**multitasking**).
- **Multiutenza**; il SO si occupa di permettere ad **utenti diversi** di utilizzare **contemporaneamente** il sistema, fornendo la possibilità di proteggere dati e risorse.



Svantaggi dei linguaggi macchina

- Fortissima dipendenza dal tipo di CPU e dai dettagli della struttura del sistema (quantità di memoria, interfacce di comunicazione...).
- Posseggono solo funzioni elementari; le operazioni complesse richiedono la scrittura di programmi ad-hoc.
- Variabili e subroutines vengono specificate tramite il loro indirizzo in memoria. Ciò rende i programmi molto difficili da leggere oltre che dipendenti dal contesto in cui sono utilizzati.

Va comunque notato che i programmi in linguaggio macchina sono i soli che un computer possa eseguire.



Vantaggi dei linguaggi di alto livello

- I linguaggi ad alto livello (o evoluti) sono **indipendenti** dal **tipo di CPU** e dalla **struttura del sistema** utilizzato.
- Mettono a disposizione del programmatore una serie di routines prefabbricate che eseguono **funzioni complesse** (calcolo, input/output...).
- Utilizzano dei **simboli** (nomi) per specificare le **variabili** e le **subroutines**, il che rende i programmi molto più leggibili.

Naturalmente, visto che alla fine è necessario fabbricare un programma in linguaggio macchina, per ogni linguaggio evoluto esiste un programma, detto **compilatore**, che esegue la traduzione. Il compilatore si occupa pure di tutti i dettagli specifici del tipo di CPU e del sistema usati.



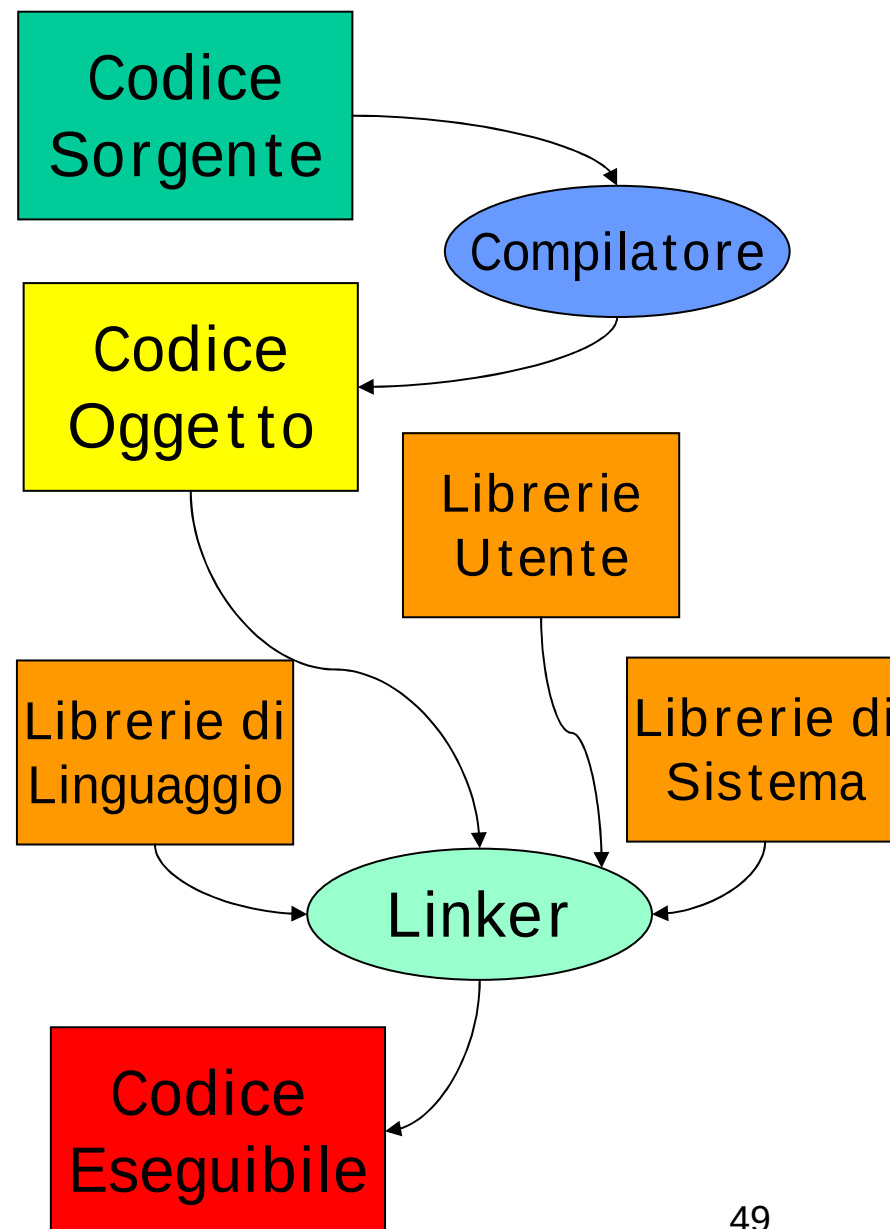
Compilazione e link

Il **Codice Sorgente** è scritto in linguaggio evoluto; usa funzioni complesse e nomi simbolici per variabili e functions.

Il **Codice Oggetto** è scritto in linguaggio machina, ma contiene riferimenti simbolici a variabili e subroutines. Usa subroutines per eseguire funzioni complesse.

Il **Codice Eseguibile** è scritto in linguaggio macchina e usa indirizzi di memoria per indicare variabili e functions.

Nel Sorgente e nell'Oggetto possono esistere riferimenti a **variabili o functions esterne**, cioè non definite nel codice medesimo. Il Codice Eseguibile **non contiene riferimenti esterni**.





Istruzioni per la compilazione

La compilazione di un sorgente C++ ed il link sono eseguiti dallo stesso comando: `g++`

La sola compilazione si ottiene con

```
g++ -c -o <oggetto>.o <sorgente>.c
```

Se si vuole eseguire compilazione e link si usa

```
g++ -o <eseguibile> <sorgente>.c <oggetti>.o -l<librerie>
```

Le librerie si aggiungono con l'opzione `-l` ed hanno nomi del tipo `lib<nome>.a` oppure `lib<nome>.so`. Le librerie di linguaggio e quelle di sistema sono aggiunte automaticamente dal comando `cc[]{}{}`.



Elementi di un linguaggio evoluto

In ogni linguaggio evoluto si possono identificare diverse componenti:

- **Sintassi**: indica l'insieme delle regole generali che governano la scrittura del programma.
- **Struttura**: indica il modo in cui è possibile suddividere un programma complesso in elementi più semplici e fornisce le regole per la comunicazione tra i sottoprogrammi.
- **Variabili**: sono le regole per la definizione e l'utilizzo delle aree di memoria destinate ai dati da elaborare.
- **Controllo di flusso**: sono le istruzioni per l'esecuzione di cicli, test, etc...
- **Funzioni**: sono i comandi che si usano per eseguire funzioni complesse (input/output, funzioni matematiche...).



Esempio di un programma in C++

```
main () {
```

Subroutine

```
int i;
```

```
double x;
```

Variabili

```
for (i=0; i<100; i++) {
```

Controllo di
flusso

```
    x = i/100.0;
```

```
    cout << "cos(" << i << "/100) = ", cos(x), endl;
```

```
}
```

```
}
```

Funzione I/O

Funzione matematica

Output:

```
cos(0/100) = 1.000000
```

```
cos(1/100) = 0.999950
```

```
cos(2/100) = 0.999800
```

```
...
```