

SOFTWARE

È l'insieme delle istruzioni che è necessario fornire alla macchina per il suo funzionamento. Vi sono due categorie di software:

- **SOFTWARE DI SISTEMA** (o di base), che deve gestire le funzioni base del computer ed è, generalmente, già installato nella macchina che si acquista;
- **SOFTWARE APPLICATIVO**, delegato ad assolvere funzioni particolari (p.e. editor di testi, elaborazione di immagini, ecc.).

Il software di sistema è suddividibile in tre sezioni principali:

➤ **il firmware,**

che consiste in quella parte di programmi memorizzata dal costruttore nella ROM, che esegue un auto-test della macchina e provvede poi a caricare nella RAM il sistema operativo; ricordiamo che la ROM contiene anche il BIOS che gestisce il controllo dei dispositivi di input e di output.

➤ **Il sistema operativo (S.O.),**

che contiene i programmi base per il funzionamento del sistema e per l'interfaccia verso l'utente. Tra i programmi che lo costituiscono, molto importanti sono:

- ⇒ il **supervisore** (command processor), che gestisce le componenti del sistema (CPU, memoria, periferiche, ecc.) interpreta ed esegue i comandi degli utenti, gestisce gli altri programmi;
- ⇒ lo **spool**, che permette di gestire la stampa contemporaneamente ad altre operazioni;
- ⇒ i **programmi di gestione delle librerie** (file manager), che gestiscono e organizzano i file;
- ⇒ i **programmi di diagnosi e di correzione degli errori di funzionamento;**
- ⇒ i **programmi per la configurazione hardware del sistema.**

Tra i sistemi operativi più noti: DOS, varie versioni di Windows, Mac Operating System, Linux, ecc.

L'utente può comunicare con il S.O. digitando apposite righe di comando (p.e. nel DOS per copiare un file "pippo.txt" dal dischetto "A" al disco "C" si deve scrivere la riga: copy a: pippo.txt c: pippo.txt) oppure mediante una interfaccia grafica (**GUI – Graphic User Interface**) come in Windows.

Il software applicativo è costituito da programmi che svolgono specifiche funzioni; una suddivisione di massima può essere la seguente:

<p>PRODUCTIVITY TOOL</p> <p>Per la normale attività quotidiana dei professionisti</p>	<ul style="list-style-type: none"> ➤ Trattamento testi ➤ Fogli elettronici ➤ Database (gestione dati) ➤ Grafica ➤ Comunicazioni ➤ Applicazioni integrate ➤
<p>BUSINESS APPLICATION</p> <p>Per la gestione di aziende, contabilità ecc.</p>	<ul style="list-style-type: none"> ➤ Finanza ➤ Contabilità ➤ Gestione affari ➤
<p>SCIENTIFIC AND TECHNICAL APPLICATION</p>	<ul style="list-style-type: none"> ➤ Analisi statistica e numerica ➤ Ingegneria ➤ Produzione ➤
<p>JOB-SPECIFIC APPLICATION</p> <p>Per specifiche esigenze e particolari tipi di utilizzo</p>	<ul style="list-style-type: none"> ➤ Medicina ➤ Legge ➤ Immobili ➤
<p>PERSONAL AFFAIR</p> <p>Per l'utente singolo, al di fuori dell'ambiente di lavoro</p>	<ul style="list-style-type: none"> ➤ Gestione privata ➤ Autoistruzione ➤ Giochi ➤

Un uso in sempre più rapida diffusione per il software applicativo è quello delle applicazioni multimediali. Queste possono essere rivolte alla "presentazione" (desktop publishing) che consiste nella successione di schermate (slide show) che possono essere corredate di effetti speciali, animazioni e suono e sostituiscono i vecchi metodi di presentazione con la lavagna luminosa.

Possono essere usate anche per realizzare materiale divulgativo (enciclopedie, atlanti, ecc.) o per autoistruzione (corsi di lingue, per imparare ad utilizzare pacchetti applicativi, ecc.)

SVILUPPO DEL SOFTWARE

Per arrivare ad un programma funzionante è necessario passare per un certo numero di fasi che, in linea di massima, sono:

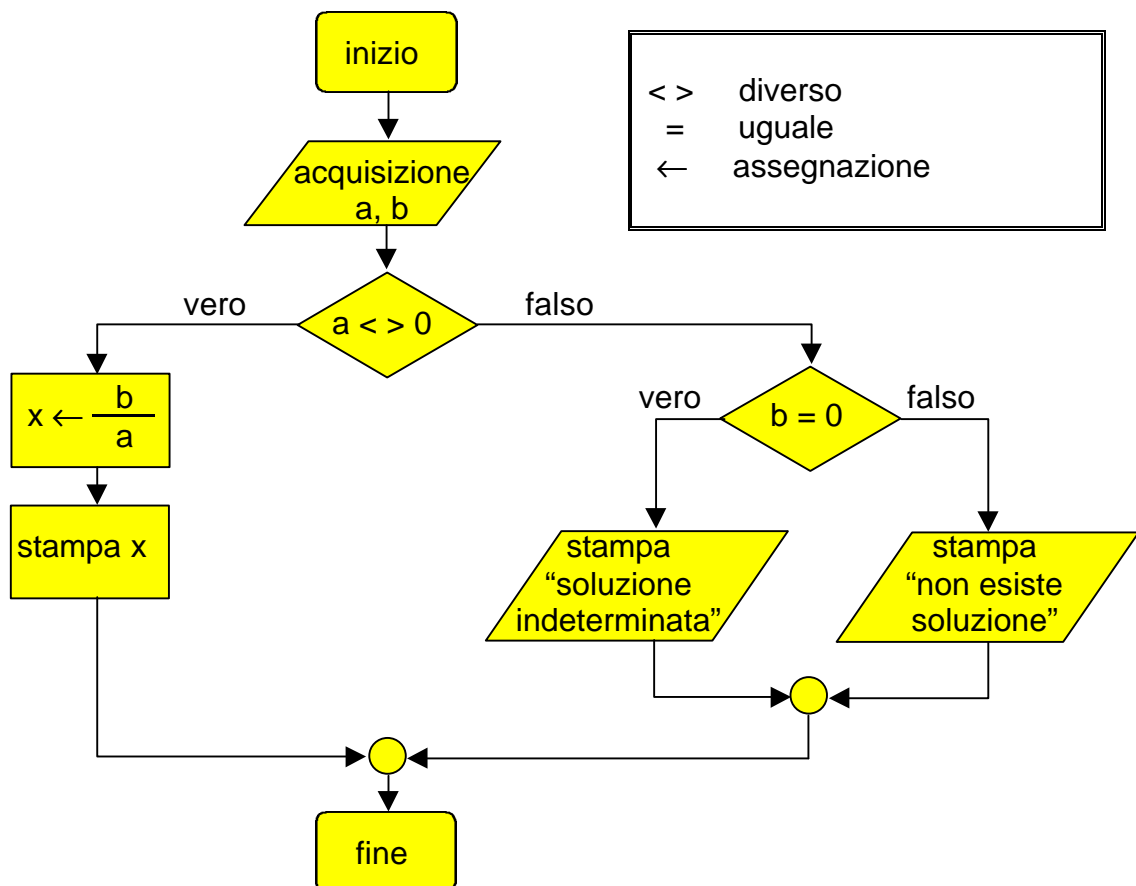
➤ ANALISI del problema.

Va esaminato accuratamente il problema per stabilire cosa si vuole ottenere, quali dati è necessario immettere e quali passi logici e matematici è necessario eseguire. È di fondamentale importanza individuare tutte le possibilità che si possono presentare nell'esecuzione del programma per non correre il rischio che si possano avere dei malfunzionamenti.

➤ RAPPRESENTAZIONE SIMBOLICA

Usando una opportuna simbologia si esegue uno schema dei passi logici (diagramma di flusso o flow-chart) da seguire per lo svolgimento del programma.

Esempio di un diagramma di flusso di un programma per risolvere una equazione di 1° grado:



➤ SCRITTURA IN PSEUDOCODICE (o linguaggio di progetto)

Se la soluzione è complessa, può essere utile, prima di scrivere la programmazione definitiva, realizzare uno schema in linguaggio naturale, contenente però le strutture necessarie a sviluppare un algoritmo secondo le regole della programmazione strutturata.

Pseudocodice di un programma per risolvere una equazione di 1° grado:

```

{
acquisisci i termini a e b
IF (a < > 0)           → se a è diverso da 0
  {
  x ← b / a           → assegna ad x il valore b / a
  stampa il valore di x
  }
ELSE                 → altrimenti
  {
  IF (b = 0)         → se b è diverso da 0
    {
    stampa "soluzione indeterminata"
    }
  ELSE              → altrimenti
    {
    stampa "non esiste soluzione"
    }
  }
}

```

➤ PROGRAMMAZIONE

Una volta descritto logicamente il problema, si può effettuare la sua scrittura in un opportuno **linguaggio di programmazione**, che consiste in una serie di istruzioni scritte secondo una ben precisa sintassi, che varia da linguaggio a linguaggio.

Per esempio vediamo un programma che riporta le istruzioni relative al diagramma precedente:

Scrivere un programma che risolva l'equazione di primo grado:

$$ax + b = 0$$

```
#include<stdio.h> _____ /*1*/
float a,b,x; _____ /*2*/
main() _____ /*3*/
{ _____ /*4*/
    printf("Inserisci il coefficiente della variabile:"); _____ /*5*/
    scanf("%f", &a); _____ /*6*/
    printf("Inserisci il termine noto:"); _____ /*7*/
    scanf("%f", &b); _____ /*8*/
    x=-b/a; _____ /*9*/
    printf("La radice dell'equazione é:%f" x); _____ /*10*/
} _____ /*11*/
```

- /*1*/ Associa al programma la libreria standard di ingresso/uscita
- /*2*/ Dichiarare che le variabili identificate come **a** e **b** (coefficienti dell'equazione) e **x** (radice dell'equazione), sono di tipo reale.
- /*3*/ Inizializza il programma.
- /*4*/ Inizio del programma.
- /*5*/ Chiede che venga digitato il valore del coefficiente della **x**.
- /*6*/ Assegna alla variabile **a** il valore digitato, specificando che si tratta di un numero reale.
- /*7*/ Chiede che venga digitato il valore del termine noto **c**:
- /*8*/ Assegna al termine noto **c** il valore digitato, specificando che si tratta di un numero reale.
- /*9*/ Assegna alla variabile **x** il valore determinato dall'espressione a destra dell'uguale.
- /*10*/ Visualizza il risultato.
- /*11*/ Fine del programma.

➤ IMPLEMENTAZIONE

Il programma scritto in precedenza (**programma sorgente**), non è comprensibile direttamente dall'elaboratore, ma va tradotto in un linguaggio da esso comprensibile (**linguaggio macchina**).

Questa traduzione può avvenire secondo due modalità diverse:

- 1) mediante un programma (**programma interprete**) che traduce volta per volta le istruzioni ogni volta che il programma viene eseguito;

2) mediante un programma (***programma compilatore***) che traduce una volta per tutte le istruzioni del programma sorgente nel linguaggio macchina; la traduzione così ottenuta prende il nome di ***programma oggetto***.

➤ **TESTING**

Finita la stesura del programma, è necessario verificare il suo corretto funzionamento nelle situazioni più diverse possibili per accertarsi che non dia luogo a malfunzionamenti in casi di utilizzo particolare.

➤ **DOCUMENTAZIONE**

Un programma deve sempre essere corredato da una documentazione completa, in modo da permettere facilmente interventi correttivi o di modifica.

