

Il software.

Programmi e linguaggi di programmazione.

Il software è l'insieme dei programmi che possono essere eseguiti dal calcolatore.

Si chiama **programma** un insieme sequenziale di istruzioni che possano venir eseguite da un computer.

Poiché tutte le operazioni che possono essere eseguite da un computer devono essere espresse in forma binaria 1001101001...., al livello più basso un qualsiasi programma per computer è espresso tramite una sequenza di 0 e di 1. Questo è il livello del cosiddetto **linguaggio macchina**. Il linguaggio macchina è l'unico linguaggio che viene direttamente compreso ed eseguito dal processore di un computer.

Scrivere programmi in codice macchina è un compito decisamente difficile per un essere umano. Per questa ragione sono stati sviluppati **linguaggi di programmazione** più comodi del linguaggio macchina. Il primo passo in questa direzione è l'**assembler** o *linguaggio assemblativo*. L'idea di base di un linguaggio assemblativo è "aggregare in blocchi" le singole istruzioni del linguaggio macchina cosicché, per esempio, se si vuole un'istruzione che faccia sommare un numero ad un altro, invece di scrivere la successione di bit "010111000...", si scrive semplicemente SOMMA r0,r1; dove "r0" e "r1" sono i nomi dei registri che contengono i numeri scritti in formato binario che si vogliono sommare. Pertanto un programma scritto il linguaggio assemblativo è molto simile a un programma in linguaggio macchina che però sia stato reso leggibile agli esseri umani. Esempio:

Linguaggio macchina

```
1010 0000 0000 0000 0000 0000 0000 0000 --> richiesta di caricare la prima
variabile nel primo registro HW
1010 0000 0010 0001 1111 1111 1111 1100 --> corrisponde al caricamento della
seconda variabile nel secondo
registro HW
0100 0000 0000 0001 0000 0000 0000 0000 --> corrisponde ad eseguire la somma
del contenuto dei due registri e a
scrivere il risultato sul primo
registro
```

Assembler

```
ldl r0,A$001           come sopra, notare l' uso di nomi simbolici per
                        le variabili e per i registri
ldl r1,B$001
addl r0,r1,r0
```

Ma cosa accade quando l'hardware riceve un programma scritto in assembler? Non lo capisce, poiché sa interpretare solo stringhe di 0 e di 1. Ma ecco il punto fondamentale: si può scrivere in linguaggio macchina un *programma di traduzione*. Questo programma detto **assemblatore** accetta i nomi mnemonici delle istruzioni (per esempio ADDL), i numeri decimali e le altre comode abbreviazioni che un programmatore riesce a ricordare facilmente e converte il tutto nelle monotone ma essenziali sequenze di 0 e di 1.

Per quanto costituisca un passo in avanti, l'assembler è però ancora molto lontano dalle esigenze di chi scrive programmi. Le istruzioni in assembler sono ancora troppo correlate con le capacità elementari dell' hardware e lontane dalla logica dei problemi che si vogliono normalmente risolvere. Per far sì che il linguaggio di programmazione rifletta il più fedelmente possibile la logica dei problemi applicativi invece che quella dell'hardware sono stati creati linguaggi di livello "più alto" detti appunto **linguaggi di alto livello** in cui ad ogni istruzione di alto livello corrispondono più

istruzioni in linguaggio macchina. Esempi di linguaggi di alto livello sono il **Fortran, C, C++, Pascal, Java, Javascript, etc.**

Per esempio in Pascal per sommare due numeri è sufficiente scrivere:

```
a+b;
```

Sussiste ovviamente sempre il problema di tradurre il programma dal linguaggio di alto livello al linguaggio macchina fatto di bit (0 e 1).

I programmi scritti in linguaggi di alto livello sono tradotti in linguaggio macchina mediante due meccanismi di base: **i compilatori e gli interpreti**.

- *Un compilatore traduce il programma sorgente di alto livello in linguaggio macchina e la traduzione viene scritta su un file detto *programma eseguibile*. Una volta ottenuto il file eseguibile, esso può essere eseguito dalla macchina ogni qualvolta l'utente lo voglia. L'esecuzione è dunque un momento separato e distinto dal momento di traduzione in linguaggio macchina. Per esempio un programma scritto in Java o in Pascal deve prima essere compilato (tradotto in linguaggio macchina) e poi solo successivamente la sua traduzione (file eseguibile) eseguita.*
- *Un interprete invece si comporta diversamente, traduce il codice di alto livello un'istruzione per volta in linguaggio macchina e, generalmente, la esegue immediatamente. Per esempio un programma scritto in Javascript non necessita di compilazione. Se lo si "lancia" esso viene interpretato, ossia le istruzioni del linguaggio vengono tradotte una per volta in linguaggio macchina ed eseguite subito. In particolare, nel caso di Javascript il programma che si occupa di fare da interprete è Internet Explorer o un qualsiasi altro browser di ultima generazione (per esempio Netscape Navigator).*

Sistema operativo e programmi applicativi

Una parte molto importante del software di un calcolatore è costituita dal **sistema operativo**, un complesso insieme di programmi che ha lo scopo di coordinare tutte le attività del calcolatore, e senza il quale esso è del tutto inutilizzabile. Il sistema operativo entra in funzione ogni qual volta viene acceso il computer. Esso fa da intermediario tra l'hardware del calcolatore, l'utente e gli altri programmi, e smista il flusso delle informazioni tra CPU, memoria centrale, memoria di massa e periferiche di input e di output. In particolare il sistema operativo presiede alla

1. Gestione dei processi (programma in esecuzione)
 2. Gestione della Memoria centrale (traduttore dall'indirizzo simbolico al fisico)
 3. Gestione della memoria secondaria (periferiche hard-disk)
 4. Gestione dei Files (organizzazione gerarchica dei dati)
 5. Gestione dei Drivers (programmi per comunicare con le periferiche)
 6. Protezione delle risorse (autorizzazioni)
- Tutti i programmi "vivono" sul SO (è l'intermediario tra il pgr e la macchina).

Una ruolo importante del sistema operativo è quello di gestire l'interfaccia con l'utente, vale a dire quegli aspetti che concernono la comunicazione tra il computer e chi lo adopera. Dal punto di vista del tipo di interfaccia, si possono individuare due categorie di sistemi operativi. Nei sistemi operativi tradizionali l'utente interagisce con il calcolatore esclusivamente attraverso comandi forniti con la tastiera. Il più diffuso sui calcolatori di uso personale è il sistema MS-DOS, che per molti anni è stato il principale sistema operativo dei PC fabbricati dalla ditta IBM. Oggi sui personal computer i sistemi operativi tradizionali sono stati largamente sostituiti dai **sistemi operativi a finestre** anche detti **sistemi operativi a oggetti**, nei quali l'interfaccia con l'utente è di tipo grafico (per esempio Windows). Ogni documento che l'utente sta usando (un testo, un programma, ecc.) viene visualizzato sul monitor sotto forma di una finestra vale a dire una zona

dello schermo con cui l'utente può interagire sia utilizzando il mouse, sia digitando dati e comandi con la tastiera.

Un'altra importante classe di programmi è costituita dal cosiddetto **software applicativo**. Si tratta di programmi specializzati che, a differenza del sistema operativo, non sono essenziali per il funzionamento del calcolatore in sé, ma consentono all'utente di usare il computer per svolgere compiti specifici. Citiamo qui di seguito alcuni dei tipi più comuni di software applicativo.

- **Text editor**: si tratta di programmi per la scrittura elettronica, che consentono di usare il calcolatore come una sorta di macchina per scrivere, producendo in output documenti in formato ASCII.
- **Word processor**: sono programmi per la scrittura elettronica, più raffinati e complessi rispetto a un text editor. Producono in output documenti in un formato più ricco di un testo ASCII, e consentono, ad esempio, di definire l'impostazione (dimensioni dei margini, spaziatura tra le linee), utilizzare vari tipi di carattere, inserire in un testo caratteri di dimensioni diverse, evidenziare parti del testo in corsivo o in grassetto, e così via. Per esempio WORD.
- **Sistemi per la gestione di basi di dati** (data base management systems): sono programmi che consentono la costruzione, la consultazione e l'aggiornamento di grossi archivi di dati, come l'archivio dei clienti di una banca o per la gestione del magazzino di una ditta. Per esempio ACCESS.
- **Fogli elettronici** (spreadsheet): sono programmi che consentono di rappresentare ed elaborare dati di vario genere in un formato simile a una tabella. Per esempio EXCEL.

