

1.6 Fasi di programmazione

Qualsiasi versione del C si abbia a disposizione, e qualsiasi sistema operativo si impieghi, le fasi del lavoro del programmatore sono costituite da:

- • editing del programma;
- • precompilazione;
- • compilazione;
- • traduzione in codice oggetto;
- • *link*;
- • esecuzione.

I vari ambienti di programmazione si differenziano per gli strumenti che mettono a disposizione per tali fasi. Per la prima di esse si potrà “minutare” il programma con l’editor preferito, l’importante è che il file prodotto sia privo di quei caratteri speciali che vengono inseriti per la formattazione del testo e per altre funzioni. Nei sistemi più diffusi questo file è in formato ASCII.

La fase di precompilazione viene eseguita dal *preprocessore C* (vedi Capitolo 8), che ha il compito di espandere alcune forme abbreviate. È, per esempio, il preprocessore che si preoccupa di sostituire nel programma ai nomi delle costanti i loro valori, specificati con la macroistruzione *define* che abbiamo introdotto in questo capitolo. L’uscita del preprocessore, costituita dal codice sorgente espanso, viene elaborata dal *compilatore C* vero e proprio, che ricerca gli errori eventualmente presenti e traduce tale codice in istruzioni scritte nel linguaggio assembler. Questa versione del programma originario in linguaggio assembler viene passata all’*assembler*, che effettua la traduzione in una forma chiamata *codice oggetto rilocabile*. Questa forma non è ancora eseguibile dal sistema di elaborazione, perché deve

essere collegata alle librerie alle quali si fa riferimento negli `include`. Infatti il compilatore lascia in sospeso tutte le funzioni che vengono invocate nel programma ma che non vi sono definite; è il caso di `printf()` e `scanf()` che abbiamo già utilizzato. Il linker ricerca tali funzioni nelle librerie indicate: se le trova le collega, altrimenti restituisce dei messaggi di errore.

La precompilazione, la compilazione, l'assemblaggio e il link possono venire effettuati dal prompt di sistema richiamando un comando specifico, che spesso è `cc` seguito dal nome del file o dei file contenenti il testo del programma:

```
$ cc rettang.c
```

Se il comando non rileva errori riappare il prompt, altrimenti scorre sul video una lista degli errori. Il codice oggetto è adesso nel file `a.out` e si può mandare in esecuzione semplicemente digitandone il nome:

```
$ a.out  
AREA RETTANGOLO
```

```
Valore base: 10  
Valore altezza: 13  
Base: 10  
Altezza: 13  
Area: 130  
$
```

Se successivamente viene eseguita la compilazione di un altro programma, il nuovo codice oggetto rimpiazzerà il primo in `a.out`, per cui è bene ogni volta effettuare una copia di `a.out` su un diverso eseguibile. Il comando `cc` ha moltissime opzioni; una di esse, `-o`, permette di specificare direttamente il nome del file oggetto:

```
$ cc rettang.c -o rettang.exe
```

È poi possibile effettuare il link separatamente, il che consente, come vedremo in seguito, una notevole flessibilità nella programmazione.

Se si è in un ambiente dedicato, per sviluppare le varie fasi naturalmente basterà scegliere le opzioni relative dai menu messi a disposizione. Per esempio, con “Microsoft Visual C++”, una volta editato il programma possiamo scegliere in sequenza le opzioni `Compile`, `Build` ed `Execute` dal menu `Build`.