

Fasi di sviluppo di un programma C (vedere Deitel & Deitel, Il Linguaggio C, pp. 13--22)

Di seguito parleremo delle varie fasi di sviluppo di un programma in linguaggio C (vedi Fig. 1).

Possiamo distinguere tre fasi principali: 1) SCRITTURA DEL PROGRAMMA, 2) CREAZIONE DELL'ESEGUIBILE, 3) ESECUZIONE DELL'ESEGUIBILE.

- 1) SCRITTURA DEL PROGRAMMA: In questa fase il programmatore crea il programma utilizzando un qualunque editor di testo e lo memorizza sul disco. Ad esempio crea un file di testo di nome *prova.c*. In ambiente linux un editor di testo è *gedit*, oppure *nano*, oppure *kate*. In ambiente Windows, ad esempio, c'è *notepad*.
- 2) CREAZIONE DELL'ESEGUIBILE: Questa fase è suddivisa in divisa in ulteriori tre sottofasi, ed inizia quando chiamo il compilatore sul file di testo creato, ad esempio: *gcc prova.c*
 - a) *Pre-processing*: Il file di testo, ad esempio *prova.c*, è inviato al pre-processore e vengono eseguite le istruzioni che iniziano con il carattere *#*. Queste sono le istruzioni per il preprocessore, e non fanno parte del linguaggio ANSI C.
 - b) *Verifica Sintattica*: Il compilatore verifica se il programma è sintatticamente corretto e lo trasforma in un codice intermedio, detto codice oggetto. Dopodichè lo memorizza sul disco.
 - c) Il corpo delle funzioni, richiamate nel programma, che fanno parte di qualche libreria (standard o definita dell'utente), e che sono state opportunamente dichiarate, viene opportunamente collegato (linkato) al codice oggetto precedentemente creato. Dopodichè è creato l'eseguibile in un file sul disco, ad esempio *prova*. Questa fase è realizzata dal *Linker*. Osserviamo che il nome del file eseguibile può essere deciso dall'utente. Se l'utente non lo specifica, in genere è chiamato di default *a.out*.

3) ESECUZIONE: Questa fase inizia una volta terminata la fase precedente. Non inizia automaticamente, ma è l'utente che deve esplicitamente avviarla, in genere semplicemente richiamando il nome del file eseguibile. Questa fase si suddivide in due sottofasi:

- a) Il file eseguibile è caricato in memoria. Questa operazione è realizzata dal *Loader*.
- b) La CPU esegue le istruzioni presenti nel programma eseguibile, memorizzando eventualmente nuovi valori dei dati mentre il programma è in esecuzione.

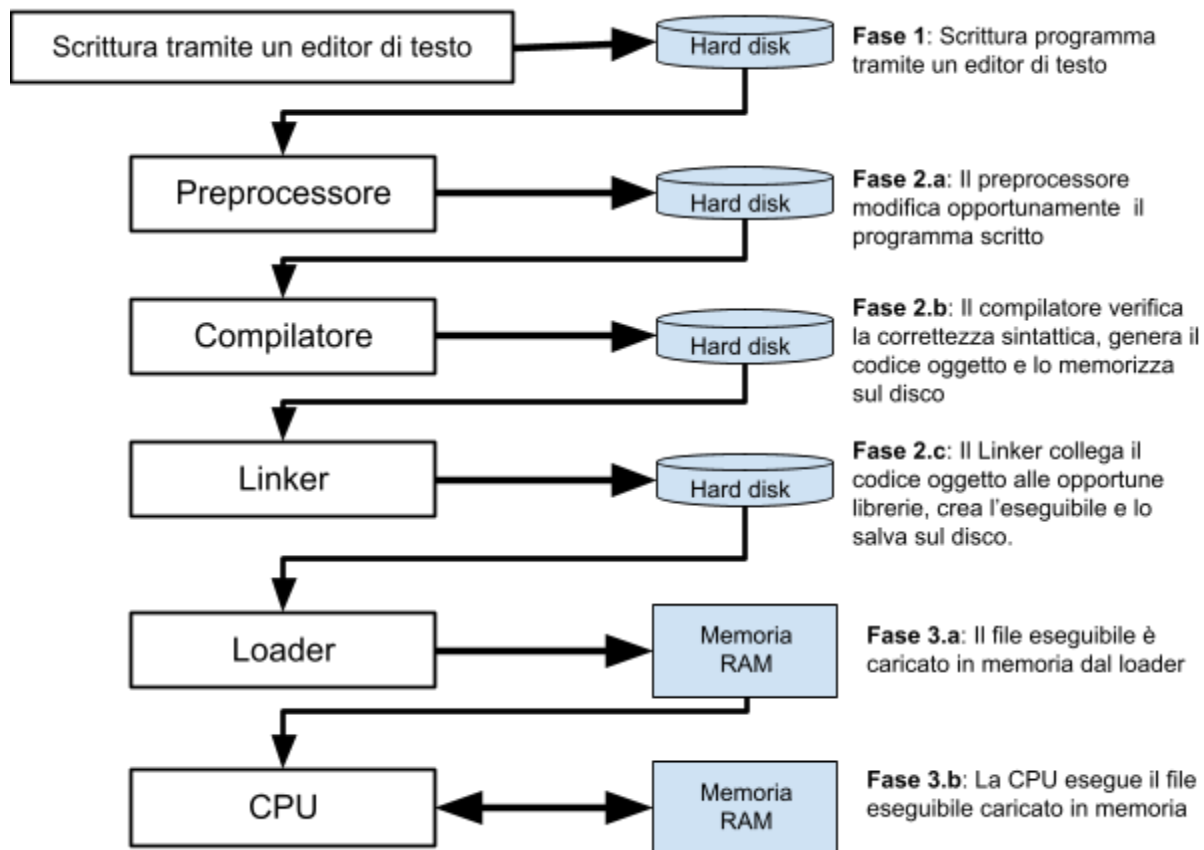


Fig.1:Flusso logico per la creazione ed esecuzione di un programma scritto in ANSI C

Si ricorda che nella fase di preprocessing un ruolo importante è giocato dalla direttiva `#include` la quale permette di includere dei file header, i quali specificano,

fondamentalmente, dichiarazioni di funzioni (prototipi) e costanti. Le funzioni dichiarate nei file header sono definite in opportune librerie. Ad esempio:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

In questo caso includo i file header che fanno riferimento alla libreria standard del ANSI C.

L'effetto di tale istruzione del preprocessore è quello di materialmente aggiungere il contenuto del file header specificato, al file di testo in cui tale istruzione è richiamata.

I file header in cui sono **dichiarate** le funzioni della libreria standard del ANSI C (dove sono **definite** le funzioni) sono:

Diagnostica: <assert.h>

Elaborazione dei caratteri: <ctype.h>

Codici di errore: <errno.h>

ANSI C limits: <limits.h> and <float.h>

Localization: <locale.h>

Nonlocal jumps: <setjmp.h>

Signal handling: <signal.h>

Variable arguments: <stdarg.h>

Common definitions: <stddef.h>

Date and time: <time.h>

Input/output: <stdio.h>

General utilities: <stdlib.h>

String processing: <string.h>

Matematica: <math.h>

I file header messi in grassetto sono quelli più usati. Un discorso particolare merita il file header **math.h**. In questo caso anche se include dichiarazioni della libreria standard, in molti compilatori le funzioni corrispondenti sono definiti in una sottolibreria dedicata, quindi è necessario aggiungerla esplicitamente. As esempio con gcc, bisogna scrivere quanto segue: *gcc nomFile.c -lm*

L'opzione `-lm` permette di aggiungere anche la libreria matematica, che fa sempre parte della libreria standard del ANSI C.