

1 Compilazione ed esecuzione di un programma in C in ambiente UNIX (Rev. 2.1.0)

Sebbene la procedura generale per creare un file contenente un programma in linguaggio C, compilarlo ed eseguirlo sia la stessa su tutti i computers i dettagli specifici **dipendono** da tre elementi: **sistema operativo**, **editore** di testo e **compilatore**. Qui considereremo il caso di un computer con sistema operativo **UNIX** e compilatore GNU C. Per gli altri sistemi rimandiamo alla documentazione allegata al computer.

Scrittura del Programma

Per scrivere un programma in linguaggio C, o in un qualsiasi altro linguaggio di programmazione, si utilizza un **editore di testo**, da non confondersi con un word-processor che genera documenti formattati, con cui si **crea** uno o più **files di testo** (files sorgente) contenenti la sequenza delle istruzioni in linguaggio C che formano il programma.

Il **nome** dei files **non** deve contenere spazi bianchi o altri caratteri speciali come “/” o “*” e **deve** terminare con il **suffisso** “.c” per indicare che il file contiene istruzioni in linguaggio C. Ad esempio si può usare “myprogram.c” o “my-program.c” ma non “my program.c” o “my/program.c”.

Esistono molti editori di testo, uno molto comune su tutti i sistemi UNIX è l'editore **GNU Emacs**. Per editare un file in ambiente UNIX basta eseguire il comando:

```
$ emacs source.c
```

dove **source.c** è il nome del file che si vuole editare. Se il file **source.c** **non** esiste Emacs **crea** un nuovo file di nome **source.c**.

Compilazione

In ambiente **UNIX** il compilatore GNU C viene invocato con il comando:

```
$ cc source.c
```

ovvero

```
$ gcc source.c
```

dove **source.c** è il nome del file sorgente del programma in linguaggio C. Se il nome del file **non** finisce con il suffisso “.c” il sistema operativo potrebbe non riconoscere il file come file sorgente un programma in C generando di conseguenza un messaggio di errore. L'**eseguibile** prodotto dal compilatore si trova nel file chiamato “a.out”.

Il comando **cc** in realtà non invoca solo il compilatore, ma chiama nell'ordine:

Preprocessore : che modifica la sorgente secondo le direttive contenute nella sorgente stessa

Compilatore : che traduce il programma generato dal preprocessore nel codice oggetto in linguaggio macchina
Linker : che crea l'eseguibile usando il codice oggetto prodotto dal compilatore e quello disponibile nelle librerie

Gli **errori** che si possono verificare a questo stadio sono errori di **sintassi** o di **compilazione**. In questo caso la compilazione viene **interrotta** ed il programma eseguibile **non** viene prodotto.

Il comando **cc** prende una serie di **opzioni** specificate mediante delle **flags**. Ad esempio se si desidera creare un file eseguibile con un nome diverso da **a.out** si può usare la flag **“-o”**:

```
$ cc -o executable source.c
```

dove **executable** è il nome del file contenente l'eseguibile. Non è necessario che il file eseguibile termini con il suffisso **“.out”** e spesso l'eseguibile viene chiamato con lo **stesso** nome del file sorgente **senza** nessun suffisso.

Alternativamente è possibile cambiare il nome al file **a.out** utilizzando il comando UNIX **“mv”** (*move*):

```
$ cc source.c  
$ mv a.out executable
```

Altre possibili flags sono **“-E”** per invocare solo il preprocessore, oppure **“-c”** per creare il codice oggetto senza chiamare il linker e quindi senza generare l'eseguibile. Nel primo caso il programma prodotto viene inviato sullo schermo, mentre nel secondo caso viene creato un file con lo **stesso** nome del file sorgente ma con il suffisso **“.o”**. Altre flags molto utili in fase di sviluppo del programma sono

- **“-Wall”** per aumentare il livello di messaggi prodotti in fase di compilazione.
- **“-ansi”** o **“-std=c89”** o **“-std=iso9899:1990”** per selezionare il dialetto C89.
- **“-std=iso9899:199409”** per selezionare il dialetto C89 Amendment 1.
- **“-std=c99”** o **“-std=iso9899:1999”** per selezionare il dialetto C99. Al momento dello scrivere il compilatore GCC C supporta solo parzialmente lo Standard C (1999).
- **“-pedantic”** per segnalare come **“warning”** ogni disuniformità rispetto allo Standard C selezionato.
- **“-pedantic-errors”** per segnalare come **“errore”** ogni disuniformità rispetto allo Standard C selezionato.

Per una lista completa rimandiamo al manuale del compilatore consultabile con il comando **“man”**:

```
$ man cc
```

ovvero

```
$ man gcc
```

o con il comando “**info**”:

```
$ info gcc
```

Esecuzione

Un programma eseguibile viene eseguito semplicemente **richiamando** il suo nome, per cui se l'eseguibile si trova nel file **a.out** questo viene eseguito semplicemente con il comando:

```
$ a.out
```

ovvero se l'eseguibile si trova nel file **executable**

```
$ executable
```

Se il programma è invocato da una linea di comando quando questo termina sul terminale ritorna il **prompt** di sistema.

Eventuali **errori** durante l'esecuzione del programma sono chiamati errori di **esecuzione** e **non** necessariamente **terminano** l'esecuzione del programma. In genere questi errori possono risultare più difficili da trovare degli errori di compilazione.

Interruzione dell'esecuzione di un programma

A volte può essere necessario **interrompere** un programma in esecuzione, ad esempio perchè è entrato in un ciclo infinito. In ambiente UNIX questo si ottiene generalmente con premendo **contemporaneamente** la key “**CTRL**” e il carattere “**c**”. In questo caso l'esecuzione viene interrotta e sullo schermo ritorna il prompt di sistema.