

Guida alla compilazione ed all'esecuzione

Alessandro Salerno

26 Settembre 2024

1 Introduzione

Questo documento descrive il funzionamento di 3 modalità per la compilazione e l'esecuzione degli esercizi del corso “Programmazione 1” presso l'Università degli Studi di Torino. Nello specifico, questo documento è destinato all'uso da parte degli Studenti del Canale C (Partizione P-Z) a.a. 2024/2025.

Tutti i metodi di compilazione descritti in questo documento fanno uso della riga di comando. Inoltre, si presume che il compilatore GCC sia già installato.

Nello specificare la sintassi dei comandi, le parentesi angolari sono state usate come placeholder per valori obbligatori, mentre le parentesi quadre sono state utilizzate come placeholder per valori opzionali.

2 Compilare direttamente con GCC

Per compilare un semplice programma C con GCC, è sufficiente usare il comando:

```
gcc <nome-file.c> -o <nome-file>[.exe]
```

Gli utenti di Sistemi Operativi Unix-like come Linux, BSD e macOS possono — e sono invitati a — omettere l'estensione “.exe”.

Per esempio, dato un file sorgente `es1.c` su una macchina Windows, è possibile compilarlo con il comando:

```
gcc es1.c -o es1.exe
```

Infine, per avviare il file eseguibile appena generato è sufficiente inserire il nome dell'eseguibile nella riga di comando come nell'esempio: ¹ ²

```
es1.exe
```

¹Nei sistemi Unix-like, la Working Directory non è inclusa nel PATH, pertanto è necessario aggiungere il prefisso “./” prima del nome dell'eseguibile quando si intende avviare un programma.

²Il metodo di esecuzione qui descritto è valido — con le dovute modifiche al percorso — anche per gli altri metodi descritti in questo documento.

3 Compilare direttamente con GCC e Linker

È possibile generare file oggetto utilizzando GCC e generare un eseguibile in un secondo momento. Questo metodo risulta particolarmente utile per progetti più grandi.

Per farlo, è sufficiente apportare delle modifiche al comando descritto nella sezione precedente: ³

```
gcc -c <file-sorgente.c> -o <file-oggetto.o>
```

Una volta generato il file oggetto, è possibile generare un file eseguibile utilizzando un linker ⁴:

```
gcc <file-oggetto.o> -o <file-eseguibile>[.exe]
```

4 Compilare con Make

Make è un programma che automatizza la creazione di file che dipendono da altri file ed è particolarmente diffuso per la compilazione di software, specie se scritto in linguaggi della famiglia C. ⁵

Per utilizzare Make è sufficiente invocare l'omonimo comando da un terminale la cui Working Directory contiene un file di nome "Makefile" senza alcuna estensione:

```
> ls
main.c      Makefile
> make
> ls
main.c      Makefile      a.out
```

Make semplifica molto il processo di compilazione, ma è necessario tenere a mente alcune caratteristiche importanti:

1. Il comportamento di un dato Makefile dipende dal codice contenuto al suo interno
2. Per evitare conflitti, è consigliabile creare una directory (Cartella) dedicata per ogni esercizio/progetto e copiare il Makefile al suo interno
3. Make potrebbe non essere sempre disponibile. È fortemente sconsigliato farne uso prima di aver appreso i comandi (compatibilmente con quanto richiesto dai Docenti)

³Un file oggetto — nella sua definizione più semplice — è un file che contiene codice binario riposizionabile e non eseguibile.

⁴Generalmente è consigliabile utilizzare un programma di Linking dedicato come ld, ma questo può risultare più arduo per gli utenti Windows e non rappresenta un valore aggiunto significativo per questi progetti.

⁵Esistono Build System come CMake che permettono la generazione di Makefile, ma sono stati omessi da questo documento per semplicità.

4.1 Usare il Makefile allegato

È possibile scaricare un Makefile molto semplice qui. Questo Makefile è stato re-allizzato appositamente per gli esercizi del corso “Programmazione 1” ed esegue automaticamente il codice dopo averlo compilato. Per usarlo, è sufficiente:

1. Scaricare il Makefile
2. Creare una directory (Cartella) dedicata all’esercizio
3. Spostare il Makefile all’interno della cartella appena creata
4. Creare uno o più file con estensione “.c”
5. Scrivere il codice

Per utilizzare lo script in un ambiente Unix-like è sufficiente eseguire il seguente comando da un terminale sito nella directory dell’esercizio ⁶:

```
make
```

Sui sistemi Windows che fanno uso di MinGW, invece, è sufficiente eseguire il seguente comando in una situazione analoga:

```
mingw32-make
```

5 Tips and tricks

- Su Windows, è possibile aprire una finestra del prompt dei comandi in una data cartella del File Explorer usando lo shortcut CTRL + L, servendo “cmd” nella barra selezionata e premendo invio
- Su Windows, è possibile aprire una finestra del Nuovo Terminale di Windows facendo tasto destro su uno spazio vuoto del File Explorer e selezionando “Apri nel Terminale”
- Il Makefile è un normale file di testo, pertanto può essere modificato con qualsiasi editor

6 Risorse

- Come scrivere Makefile
- Pagina sui File Oggetto - Wikipedia EN

⁶Per raggiungere una directory è possibile usare il comando “cd”.

7 Crediti ed informazioni

Questo documento è una revisione ufficiale del materiale originale scritto dallo Studente Alessandro Salerno e distribuito liberamente sul Gruppo Telegram di riferimento in formato PDF in data 25 Settembre 2024. Il Makefile di riferimento menzionato nella sezione 4.1 è distribuito secondo i termini della licenza “Creative Commons Zero v1.0 Universal”, ulteriori informazioni sono disponibili sul Repository GitHub dedicato.