

Fondamenti di Informatica 2011-2012

Esercitazione 1

Abstract

In questa prima esercitazione familiarizzeremo con l'interfaccia a linea di comando (prompt dei comandi) di Windows, imparandone il funzionamento di base. Impareremo quindi ad utilizzare il compilatore GCC per il linguaggio C e ad interpretare i messaggi di errore e di avviso che questo riporta. Infine tradurremo qualche breve algoritmo in questo linguaggio, andandone a testare il funzionamento.

1 Informazioni generali

1. Alcune esercitazioni proposte potrebbero richiedere più di 2 ore. Questo è voluto, per invogliarvi a completare l'esercitazione a casa e ripensare all'attività proposta.
2. Gli esercizi asteriscati presentano difficoltà superiori rispetto a quelle degli esercizi non asteriscati proposti nel contesto di una esercitazione.
3. Se volete conservare i file creati durante le esercitazioni, è opportuno effettuare un backup di tali file ad esempio copiandoli su una chiavetta o inviandovi via e-mail.

2 Il prompt dei comandi

Per utilizzare il compilatore GCC per il linguaggio C è necessario interagire con Windows tramite l'interfaccia a linea di comando. Tramite questa interfaccia è possibile, ad esempio, muoversi fra le cartelle presenti sul disco rigido del computer, creare, modificare e rimuovere file e cartelle, ed eseguire programmi.

Lanciare il prompt dei comandi Per lanciare il prompt dei comandi potete usare la scorciatoia presente nel menù avvio: **Start - Programs - Accessories - Command Prompt**. Comparirà una finestra, dove vedrete la seguente linea e un cursore lampeggiante:

```
C:\Documents and Settings\userxp>
```

Questa linea identifica la directory corrente, mentre il cursore lampeggiante indica che il sistema è in attesa di un comando.

Visualizzare il contenuto di una cartella Proviamo ora a visualizzare il contenuto della cartella. Provate a scrivere il comando

```
dir
```

e quindi a premere il tasto Invio. Che cosa succede?

Effettuate ora la stessa operazione utilizzando l'interfaccia grafica: dalla finestra Desktop aprite **My Computer** e spostatevi nella medesima cartella. Quali sono le differenze? Come vengono distinti file e cartelle?

Navigare fra le cartelle - Sottocartelle Proviamo ora a cambiare la directory corrente. Fra le sottocartelle della cartella **userxp** è presente una sottocartella di nome **Desktop**. Cambiate la cartella corrente tramite il comando

```
cd Desktop
```

seguito dal tasto Invio. Che cosa succede? Provate ora a visualizzare il contenuto della cartella come fatto in precedenza.

Ritorniamo ora nella cartella precedente. Scrivete il comando `cd ..`, che permette di spostarsi nella cartella che contiene quella corrente, seguito dal tasto Invio. Che cosa succede? Provate ora a visualizzare il contenuto della cartella come fatto in precedenza.

Effettuate ora le stesse operazioni utilizzando l'interfaccia grafica. Quali sono le differenze?

Creare una sottocartella Spostatevi nuovamente nella cartella **Desktop**. Proviamo ora a creare una nuova sottocartella di nome **Labo**. Usate il comando

```
md Labo
```

seguito dal tasto Invio. Provate ora a visualizzare il contenuto della cartella come fatto in precedenza. Che cosa è successo?

Verificate l'effetto del comando utilizzando anche l'interfaccia grafica.

3 Scrivere un programma C

Per scrivere un programma C, e creare quindi il *programma sorgente*, potete utilizzare un comune *editor* di testo. Per la sua semplicità vi consigliamo Notepad, che potete eseguire tramite l'icona che trovate nel menu avvio: **Start - Programs - Accessories - Notepad**.

Scrivete ora nell'area di testo il seguente programma C:

```
#include <stdio.h>
int main()
{
    printf("Hello World\n");
    return 0;
}
```

Salvate quindi il file nella cartella **Labo** appena creata, utilizzando il nome `helloworld.c`.

4 Compilare un programma C

Sulle macchine del laboratorio è stata installata la suite di programmi MinGW¹ che comprende un compilatore per il linguaggio C: GCC.

Il compilatore GCC in realtà svolge tre funzioni distinte: preprocessore, compilatore (come descritto a lezione) e linker. Nel seguito, quando useremo la parola ‘compilazione’ in riferimento ad un certo programma sorgente intenderemo l’azione che corrisponde a preprocessare, compilare e linkare tale programma.

Per compilare il programma, quindi per creare il *programma eseguibile* scritto in linguaggio macchina, dovete utilizzare il prompt dei comandi. Spostatevi nella cartella contenente il file `helloworld.c` e scrivete il comando:

```
gcc.exe -ansi -Wall helloworld.c
```

oppure

```
gcc -ansi -Wall helloworld.c
```

seguito dal tasto Invio.

`gcc` (o `gcc.exe`) è il nome di un programma eseguibile che si vuole eseguire, cioè caricare in memoria centrale e fare eseguire dal processore. In questo caso, il programma corrisponde al compilatore GCC.

L’opzione `-ansi` specifica di utilizzare la variante del linguaggio C che avete visto a lezione, mentre l’opzione `-Wall` richiede di mostrare durante la compilazione tutti gli errori non critici (warnings) presenti nel codice. `helloworld.c` infine richiede di compilare il file chiamato `helloworld.c` contenuto nella directory corrente.

Se non sono presenti errori sintattici nel codice (errors) il compilatore creerà un file chiamato `a.exe` nella directory corrente contenente il programma compilato. Il programma `a.exe` è il programma eseguibile corrispondente al programma sorgente `helloworld.c`.

Nota: *il compilatore potrebbe creare anche un file chiamato `a.out`, ignoratelo.*

Se lo desiderate potete cambiare il nome del file generato usando l’opzione `-o` seguita dal nome di un file. Ad esempio il comando:

```
gcc.exe -ansi -Wall helloworld.c -o helloworld.exe
```

genererà un file eseguibile chiamato `helloworld.exe`.

¹<http://www.mingw.org/>

5 Eseguire un programma

Eseguite il programma compilato digitando `a.exe` (oppure `helloworld.exe`) nel prompt dei comandi, seguito da Invio. Cosa succede? Cercate di dare una risposta a ciò che succede basandovi sulla descrizione del comando `gcc` fornita in Sezione ??.

6 Errori sintattici

Qualora il programma che state cercando di compilare contenga degli errori sintattici, questi verranno riportati da `gcc`, indicando la riga alla quale è stato rilevato un errore e la sua descrizione. È importante imparare a interpretare le informazioni date dal compilatore, in modo da potere correggere gli errori presenti nel codice sorgente. Spesso i messaggi di errore sono chiari e riportano con precisione il tipo e la posizione dell'errore. Può capitare, però, che la riga dove viene segnalato un errore sia nelle immediate vicinanze di quella che contiene effettivamente l'errore.

Esistono due tipologie di errore:

1. errori veri e propri, che impediscono al compilatore di generare il programma eseguibile;
2. situazioni che richiedono attenzione da parte del programmatore (*warning*) ma che non impediscono la generazione del programma eseguibile.

Provate ad esempio a verificare i potenziali messaggi di errore che vengono riportati se modificate il file `helloworld.c` come descritto nel seguito. Per ogni caso, cercate di capire se il programma eseguibile è stato generato oppure non è stato generato e perché.

- Dimenticate di includere `<stdio.h>`.
- Dimenticate di scrivere il tipo della funzione `main` (`int`).
- Dimenticate di mettere le parentesi tonde dopo `main`.
- Dimenticate di mettere una o entrambe le parentesi graffe.
- Dimenticate di mettere una o entrambe le virgolette prima di `Hello`.
- Dimenticate un punto e virgola.
- Scrivete `return;` invece che `return 0;`
- Non scrivete né `return;` né `return 0;`

Che differenze ci sono nei messaggi riportati se omettete l'opzione `-Wall` dal comando di compilazione?

7 Primi programmi in C

Nota importante: *Per risolvere gli esercizi proposti, per prima cosa abituatevi a scrivere su carta l'algoritmo in pseudo-codice che risolve il problema che vi viene proposto. Solo dopo traducete l'algoritmo in un programma C. Utilizzate dei commenti e indentare il codice per rendere il programma più leggibile.*

7.1 Esercizio 1

Scrivete un programma che:

- Legge due numeri interi da standard input e li assegna a due variabili chiamate `a` e `b`.
- Stampa su standard output la stringa `"Dopo la lettura, nella variabile a è memorizzato il valore (valore di a) e nella variabile b è memorizzato il valore (valore di b)"` e poi va a capo.
- Scambia i valori memorizzati nelle due variabili `a` e `b`.
- Stampa su standard output la stringa `"Dopo lo scambio, nella variabile a è memorizzato il valore (valore di a) e nella variabile b è memorizzato il valore (valore di b)"` e poi va a capo.

7.2 Esercizio 2

Modificate il programma realizzato per il primo esercizio in modo che calcoli e stampi la somma, la sottrazione, la divisione, e la moltiplicazione dei due numeri decimali forniti.

Provate ad eseguire il programma fornendo i seguenti valori e verificare il risultato:

`a=5, b=2`
`a=3, b=10`
`a=3, b=0`
`a=2.4, b=1.1`

7.3 Esercizio 3

Scrivete un programma permetta all'utente di inserire la lunghezza dei cateti `a` e `b` di un triangolo rettangolo. Il programma deve calcolare l'ipotenusa `c` del triangolo, il suo perimetro `2P` e la sua area `A` usando la formula di Erone:

$$A = \text{sqrt}(P * (P - a) * (P - b) * (P - c))$$

dove `a`, `b` e `c` sono i lati e `P` il semiperimetro, cioè $P = (a + b + c)/2$.

7.4 Esercizio 4

Scrivete un programma che richieda all'utente di inserire una latitudine in gradi $[-90,+90]$, minuti $[0,-60]$ e secondi $[0,-60]$ (tutti valori interi) e quindi stampi la latitudine in termini di gradi in formato decimale. Vi sono 60 secondi in un minuto e 60 minuti in un grado. Utilizzate i tipi e i qualificatori che vi sembrano piu' opportuni per conservare i valori forniti dall'utente. Utilizzate delle costanti per memorizzare i fattori di conversione fra secondi e gradi e minuti e gradi. Verificate che fornendo 37 gradi, 51 minuti e 19 secondi il risultato sia approssimativamente 37.855.

7.5 Esercizio 5*

Scrivete un programma per la stampa di biglietti cinematografici. Una volta lanciato il programma chiede al cassiere il costo di ogni singolo biglietto (valore decimale), il nome del cliente e il numero di posti desiderati. Una volta che l'utente ha inserito i dati richiesti il programma deve chiedere conferma, chiedendo all'utente di scrivere un carattere (S o N). Se l'utente non conferma (N) il programma termina, altrimenti (S) stampa un biglietto contenente il nome del cliente, il numero di posti richiesti e il costo del biglietto, come segue:

```
#####  
# Nome: Mario Rossi  
# Numero posti: 2  
# Prezzo: 22,50  
#####
```

7.6 Esercizio 6*

Modificate uno dei programmi realizzati di modo che non termini dopo avere stampato il risultato ma che continui a richiedere nuovi numeri fino a quando l'utente non inserisce due 0.