Tutoraggio 1

Introduzione alla programmazione in C

Gabriele Russo Russo

Corso di Calcolatori Elettronici A.A. 2018/19



Tutoraggio

Martedì 9,30-11,15 (Aula B1)

- Esercizi
- Domande

Piazza

https://piazza.com/uniroma2.it/fall2018/ce1920/

- Domande
- Risposte

Ambiente di sviluppo per il C

Il minimo indispensabile per programmare in C:

- Editor di testo (Notepad++, gedit, Atom, ViM, ...)
- Compilatore C



GNU Compiler Collection

Faremo riferimento ad uno specifico compilatore C: **GCC** (GNU Compiler Collection)

- Scaricabile liberamente
- Disponibile (con qualche differenza) su tutte le piattaforme



Vediamo come installarlo su Linux, Windows e Mac

Il compilatore GCC puo' essere installato facilmente tramite il gestore dei pacchetti della vostra distribuzione. Ad esempio:

Ubuntu, Debian, Linux Mint, ...

\$ sudo apt-get install build-essential

Fedora

\$ sudo dnf install gcc

Installazione di MinGW su Windows

MinGW (Minimalist GNU for Windows)
scaricabile gratuitamente da http://www.mingw.org/

- 1. Dalla pagina *Downloads*, scaricare mingw-get-setup.exe
- 2. Avviare mingw-get-setup.exe e seguire le istruzioni
- Per utilizzare i tool di MinGW dal prompt dei comandi, una volta installato e' necessario modificare la variabile d'ambiente PATH. Bisogna aggiungere una nuova entry: <directory mingw>\bin sostituendo <directory mingw> con la directory dove avete installato MinGW (ad es. C:\MinGW\).

Soluzione alternativa:

installare MinGW insieme a CodeBlocks (ne parleremo a breve)

Diverse possibilità:

- 1. potrebbe già essere installato (provate ad eseguire gcc)
- 2. (Per versioni meno recenti di OS X), installando i *Command Line Tools* da Apple Developer
- 3. tramite il tool Homebrew, eseguendo brew install gcc (l'installazione puo' richiedere molto tempo); per i dettagli: http://macappstore.org/gcc/
- scaricando una release non ufficiale di GCC per Mac da http://hpc.sourceforge.net/ e seguendo le istruzioni riportate nella pagina
- 5. Se trovate soluzioni migliori, condividetele su Piazza!

Compilare con GCC: primo esempio

Consideriamo il classico esempio hello_world.c.

Compilazione ed esecuzione di hello_world.c				
\$	gcc -o hello h	nello_world.c		
\$./hello	# Su Linux o Mac		
\$	hello.exe	# Su Windows		
He	Hello, world!			

Se il codice C non è sintatticamente corretto, il compilatore non produce alcun file eseguibile e stampa dei messaggi di **errore**. Esempio: proviamo ad eliminare una parentesi graffa... GCC può visualizzare dei **warning** durante la compilazione. In tal caso, il file eseguibile viene comunque generato. Tuttavia, un warning è molto spesso indicazione di potenziali errori commessi dal programmatore, i cui effetti sarebbero visibili solo al momento dell'esecuzione.

Vediamo gli esempi warning1.c e warning2.c

Compilare abilitando la stampa di warning

\$ gcc -Wall -Wextra -o hello hello_world.c

Se il vostro codice causa warning, c'è qualcosa da sistemare!

Se volete utilizzare un IDE, una possibile opzione è CodeBlocks.

Puo' essere scaricato gratuitamente dal sito ufficiale: http://www.codeblocks.org/downloads/26.

Per gli utenti Windows, e' consigliato scaricare il pacchetto codeblocks-17.12mingw-setup.exe, che installa anche il compilatore MinGW.

Makefile

- Il comando make è una utility che automatizza il processo di compilazione
- Usando make evitiamo di invocare manualmente GCC ad ogni compilazione
- Si basa sul Makefile: una specifica del processo di compilazione che vogliamo automatizzare
- Particolarmente utile per progetti medio-grandi in cui si ha a che fare con molteplici file sorgenti
- Saper scrivere un Makefile NON è tra gli obiettivi del corso! Saper usare dei Makefile già pronti può essere utile (anche per la prova al calcolatore)

Makefile: esempio con Hello World

Un Makefile che possiamo usare per compilare hello_world.c

Makefile				
all: Hello				
Hello: hello_world.c gcc -Wall -Wextra -o \$@ \$^				

Salviamo questo Makefile nella stessa cartella dove si trova il file .c, ed eseguiamo make nel terminale.

Makefile: un altro esempio

Un Makefile che possiamo usare per compilare tutti i file .c presenti nella cartella corrente

Makefile

```
CC=gcc
CFLAGS=-Wall -Wextra -O2
SOURCES=$(wildcard *.c)
EXECUTABLES=$(SOURCES:.c=)
```

all: \$(EXECUTABLES)

% : %.c \$(CC) \$(CFLAGS) \$^ -o \$@

Primi esempi di programmi C

Stampa di valori a schermo
 esempi_print.c

Lettura di valori da tastiera:

- esempio_getchar.c
- esempi_scanf.c
- esempi_scanf2.c
- Ciclo "for" e "while"

primes.c