

CALCOLO NUMERICO con ELEMENTI DI PROGRAMMAZIONE (BATR) - (A.A. 2012-2013)

Prof. F. Pitoli

INSTALLAZIONE COMPILATORE C

Ing. Gabriele Colosimo, Ing. Andrea Nascetti

Area di Geodesia e Geomatica
Dipartimento di Ingegneria Civile Edile e Ambientale
Università di Roma "La Sapienza"

<gabriele.colosimo, andrea.nascetti>@uniroma1.it

Indice

1 Breve introduzione al modulo di programmazione

2 Installare il compilatore C

- Introduzione
- Windows
- Linux
- Mac Os X

3 Hello World!

- Familiarizzare con il terminale
- Editor di testo
- Codice sorgente
- Riassunto

Indice

1 Breve introduzione al modulo di programmazione

2 Installare il compilatore C

- Introduzione
- Windows
- Linux
- Mac Os X

3 Hello World!

- Familiarizzare con il terminale
- Editor di testo
- Codice sorgente
- Riassunto

Concetti generali

Il computer

- Un **Computer** è un dispositivo in grado di eseguire calcoli ed operazioni a velocità estremamente elevate
- I vari dispositivi che compongono un computer (tastiera, schermo, dischi, ...) sono chiamati **hardware**
- I programmi che possono essere eseguiti su un computer sono chiamati **software**

Il Sistema Operativo (SO)

- Il SO si occupa di interfacciare l'hardware con il software fornendo a quest'ultimo tutti i servizi di accesso all'hardware stesso

Concetti generali

Il computer

- Un **Computer** è un dispositivo in grado di eseguire calcoli ed operazioni a velocità estremamente elevate
- I vari dispositivi che compongono un computer (tastiera, schermo, dischi, ...) sono chiamati **hardware**
- I programmi che possono essere eseguiti su un computer sono chiamati **software**

Il Sistema Operativo (SO)

- Il SO si occupa di interfacciare l'hardware con il software fornendo a quest'ultimo tutti i servizi di accesso all'hardware stesso

Scopo del modulo di elementi di programmazione

- Acquisire i rudimenti della programmazione utili a implementare semplici software per la soluzione di problemi di calcolo numerico

Indice

1 Breve introduzione al modulo di programmazione

2 Installare il compilatore C

- Introduzione
- Windows
- Linux
- Mac Os X

3 Hello World!

- Familiarizzare con il terminale
- Editor di testo
- Codice sorgente
- Riassunto

Il ruolo del compilatore

Codice sorgente e linguaggio macchina

- Un computer è in grado di comprendere esclusivamente istruzioni di basso livello in formato binario (**linguaggio macchina**)
0111100000111101010
- Un linguaggio di programmazione permette di scrivere programmi come successione logica di istruzioni a più alto livello (**codice sorgente**)
`printf (" Ciao mondo")`
- Affinché il programma possa essere costruito, si rende necessario **tradurre** le istruzioni di alto livello nel linguaggio macchina
- **Compilatore**: programma in grado di **tradurre il codice sorgente in linguaggio macchina**

Passaggi necessari per la costruzione di un programma

- 1 schematizzare la sequenza logica di istruzioni da eseguire
- 2 scrivere il codice sorgente del programma secondo la **sintassi** del linguaggio prescelto
- 3 **compilare** il codice sorgente
- 4 generare un file che possa essere eseguito dal computer (**linker**)

Passaggi necessari per la costruzione di un programma

- 1 schematizzare la sequenza logica di istruzioni da eseguire
- 2 scrivere il codice sorgente del programma secondo la **sintassi** del linguaggio prescelto
- 3 **compilare** il codice sorgente
- 4 generare un file che possa essere eseguito dal computer (**linker**)

IMPORTANTE

- Il linguaggio macchina dipende da **hardware** (architettura del processore: 32/64 bit) e **software** (ad esempio dal SO: Linux, Mac Os, Windows, ...) del computer
- Il linguaggio di programmazione segue degli **standard**; l'insieme delle regole codificate negli standard è detta **sintassi**
- Lo stesso programma, per essere eseguito su computer con diverse configurazioni, **deve essere compilato nuovamente**

Il linguaggio C

Cenni storici

- Viene sviluppato nel 1967 da Martin Richards come linguaggio per scrivere il software dei sistemi operativi
- Diviene inizialmente famoso per essere alla base del SO **UNIX**
- Nel 1989 fu approvato il primo standard “per fornire una versione del linguaggio che non fosse ambigua e che fosse indipendente dalle macchine” (**C89**)
- Grazie alla sua efficienza, velocità e **portabilità**, è oggi alla base dei principali SO dell'ultima generazione

Il compilatore GCC

Caratteristiche e informazioni

- In questo corso utilizzeremo il compilatore **GNU Compiler Collection (GCC)**
- Principali caratteristiche:
 - 1** multiplatforma (Linux, Windows, Mac Os X, ...)
 - 2** supporta diversi linguaggi di programmazione (C, C++, Objective-C, Fortran, Java, Ada, e Go)
 - 3** progetto **Open Source** (gratuito e aperto alle modifiche dell'utente)
 - 4** documentazione liberamente disponibile [on-line](#)

GCC su Windows I

Cygwin

- Trasporta in ambiente Windows un insieme di applicazioni native dell'ambiente Linux
- L'installazione può avvenire attraverso un collegamento internet o utilizzando il materiale fornito nel corso

Internet

- 1 collegarsi al sito del progetto [Cygwin](#)
- 2 scaricare il file *setup.exe* e salvarlo nella cartella Utente^a
- 3 eseguire il file *setup.exe*

^ain windows `C:\Utenti\<nome_utente>`

Materiale Corso

- 1 eseguire il file
“CNEP_DVD\Windows\setup.exe
- 2 dalla schermata iniziale cliccare su
“Avanti” e poi “Install from Local Directory”

GCC su Windows II

Internet

- 4** dalla schermata iniziale cliccare su “Avanti” e poi “Install from internet”
- 5** lasciare le impostazioni di default sulle cartelle di installazione (`C:\cygwin`)
- 6** selezionare la cartella locale di installazione dei file (se la cartella non esiste questa viene creata)
- 7** lasciare l'impostazione di default sulle connessioni
- 8** scegliere tra i siti disponibili e avviare il download

Materiale Corso

- 3** lasciare le impostazioni di default sulle cartelle di installazione (`C:\cygwin`)
- 4** selezionare la cartella di installazione “`CNEP_DVD\Windows\Install`” nel materiale del corso

GCC su Windows III

Internet/Materiale Corso

- 8** ignorare il messaggio di “warning” e proseguire selezionando “OK”
- 9** scelta dei pacchetti da installare. Espandere la tendina “Devel” e selezionare i seguenti pacchetti:
 - gcc4-core: core C compiler subpackage (10508 K)
 - libstdc++6-devel: C++ Standard library static/import link libs
 - gdb: The GNU debugger (5600 K)
- 10** accettare la risoluzione delle dipendenze e cliccare su “Avanti”
- 11** creare le icone del programma e terminare l'installazione

GCC su Linux

Passaggi per l'installazione

1 avviare il computer

2 ...

3 ...

4 **FATTO!**

- La maggior parte delle distribuzioni Linux ha già installato GCC al proprio interno

GCC su Mac Os X I

Developer kit

- Apple distribuisce gratuitamente una serie di strumenti (tra cui GCC) per gli sviluppatori
- I pacchetti forniti nel materiale didattico sono stati scaricati dal sito [Apple Developer](#)

GCC su Mac Os X II

Identificare la versione del proprio SO

- Selezionare dal Menu Apple (quello con la Mela): “Informazioni su questo Mac”
- Seguendo la tabella, associare la versione al nome del SO

Versione	SO
10.8.*	Mountain Lion
10.7.*	Lion
10.6.*	Snow Leopard
10.5.*	Leopard

Tabella : Nome e versione SO Mac OS X

- Selezionare nella cartella “CNEP_DVD\Mac_OsX\<SO>” il pacchetto .dmg corrispondente al SO
- Montare il file immagine e installare il pacchetto

Indice

1 Breve introduzione al modulo di programmazione

2 Installare il compilatore C

- Introduzione
- Windows
- Linux
- Mac Os X

3 Hello World!

- Familiarizzare con il terminale
- Editor di testo
- Codice sorgente
- Riassunto

Accedere al terminale

Windows

- Il terminale può essere lanciato utilizzando l'icona di *Cygwin*^a presente sulla Scrivania
- Il terminale può essere lanciato selezionando il file *C:\cygwin\Cygwin.bat*

^aGenerata durante la fase di installazione

Linux

- Il terminale si trova generalmente tra le applicazioni degli *Accessori*

Mac Os X

- Aprire *Spotlight*^a e cercare *terminal*. Per comodità negli utilizzi successivi, rendere permanente il collegamento nella *dashboard*

^aRicerca con la lente

Comandi di base del terminale I

Sintassi Generale

- i comandi del terminale hanno tutti una sintassi generica del tipo:

comando [−opzione_1 ... −opzione_n] [file_1 ... file_n]

```
user@CALCOLO_NUMERICO: ls -l hello.c ./
-rw-r--r-- 1 poweruzzen poweruzzen 159 2012-09-13 17:38
    hello.c

./:
totale 336
drwxr-xr-x 2 poweruzzen poweruzzen 4096 2012-09-13 14:52
    fig
-rwxr-xr-x 1 poweruzzen poweruzzen 8497 2012-09-13 12:27
    hello
-rw-r--r-- 1 poweruzzen poweruzzen 159 2012-09-13 17:38
    hello.c
```

Comandi di base del terminale II

cat

- Mostra il contenuto di un file

```
user@CALCOLO_NUMERICO: cat file_prova
Nome   Cognome   Nascita   Professione
```

```
Mario  Rossi     22/12/1980  ingegnere
user@CALCOLO_NUMERICO:
```

Comandi di base del terminale III

cd <destinazione >

- Permette di accedere alla cartella *destinazione*
- Usato senza argomenti dirige alla *home directory* dell'utente
- `cd ..` torna indietro di una cartella

```
user@CALCOLO_NUMERICO: ls -l
totale 104
drwxrwxr-x 2 uzen-sweet uzen-sweet 4096 set 25 13:03
  Cartella1
-rw-r--r-- 1 uzen-sweet uzen-sweet 98764 giu 22 2011 logo.
  png
user@CALCOLO_NUMERICO: cd Cartella1/
user@CALCOLO_NUMERICO:
```

Comandi di base del terminale IV

cp <file1 ><file2 >

- Copia il *file1* in *file2*

```
user@CALCOLO_NUMERICO: ls
logo.png
user@CALCOLO_NUMERICO: cp logo.png logo2.png
user@CALCOLO_NUMERICO: ls
logo.png logo2.png
user@CALCOLO_NUMERICO:
```

clear

- Pulisce lo schermo del terminale

exit

- Termina la sessione ed esce dal terminale

Comandi di base del terminale V

history

- Mostra gli ultimi n comandi eseguiti dall'utente

```
user@CALCOLO_NUMERICO: history
102  cat file_prova
103  clear
104  history
user@CALCOLO_NUMERICO:
```


Comandi di base del terminale VI

mkdir <name>

- Crea una nuova cartella *name*

```
user@CALCOLO_NUMERICO: ls
logo.png
user@CALCOLO_NUMERICO: mkdir Nuova_Cartella
user@CALCOLO_NUMERICO: ls
logo.png  Nuova_Cartella
user@CALCOLO_NUMERICO:
```

Comandi di base del terminale VII

mv <file1 ><file2 >

- Rinomina il *file1* in *file2*

```
user@CALCOLO_NUMERICO: ls
logo . png
user@CALCOLO_NUMERICO: mv logo . png my_logo . png
user@CALCOLO_NUMERICO: ls
my_logo . png
user@CALCOLO_NUMERICO:
```

pwd

- Mostra la cartella di lavoro attuale

```
user@CALCOLO_NUMERICO: pwd
/home/Gabriele/Corsi/
user@CALCOLO_NUMERICO:
```

Comandi di base del terminale VIII

rm <file >

- Rimuove uno o più file

```
user@CALCOLO_NUMERICO: ls  
logo . png  
user@CALCOLO_NUMERICO: rm logo . png  
user@CALCOLO_NUMERICO: ls  
user@CALCOLO_NUMERICO:
```

Editor suggeriti per le diverse piattaforme

Windows

- [Notepad++](#)

Linux

- [Geany](#)

Mac Os X

- [TextWrangler](#)

Tutorial 1 I

- Aprire il terminale
- Creare una nuova cartella utilizzando il comando `mkdir`

```
user@CALCOLO_NUMERICO: mkdir Esempio1
```

- Spostarsi all'interno della cartella creata utilizzando il comando `cd`

```
user@CALCOLO_NUMERICO: cd Esempio1
```

- Aprire l'*editor* prescelto e scrivere il seguente codice sorgente:

Tutorial 1 II

```
1  /*
2  * programma: hello.c
3  * Stampa messaggio di saluto sullo schermo
4  */
5
6  #include <stdio.h>
7
8  int main()
9  {
10     printf("Hello World!\n");
11     return 0;
12 } /* fine funzione main */
```

- Salvare il codice in un file (*hello.c*) dentro la cartella appena creata
- Controllare il contenuto della cartella con il comando `ls`

```
user@CALCOLO_NUMERICO: ls
hello.c
user@CALCOLO_NUMERICO:
```

Tutorial 1 III

- Compiliamo il codice sorgente utilizzando GCC

```
user@CALCOLO_NUMERICO: gcc -o hello.out hello.c
```

- Se la compilazione è andata a buon fine, viene prodotto il file *hello.out*. Controlliamo il contenuto della cartella con il comando `ls`

```
user@CALCOLO_NUMERICO: ls  
hello.c hello.out  
user@CALCOLO_NUMERICO:
```

- Il file *hello.out* può essere eseguito il seguente comando:

```
user@CALCOLO_NUMERICO: ./hello.out
```

- Controlliamo il risultato:

```
user@CALCOLO_NUMERICO: ./hello.out  
Hello World!  
user@CALCOLO_NUMERICO:
```

Contenuti principali

Elementi di Programmazione

- (Brevissima) Introduzione al Computer e ai concetti di *hardware*, *software* e Sistema Operativo
- Il **linguaggio C** e il **compilatore** (cos'è e a cosa serve). Perché scegliere **GCC**
- Passaggi necessari per la **costruzione di un programma**
- Installazione del compilatore GCC sulle diverse piattaforme (Windows, Linux, Mac Os X)
- Come muoversi (“a gattoni”) nel **terminale**
- Scrivere e compilare il nostro primo programma (**Hello World!**)

Riferimenti Bibliografici



B.W. Kernighan and D.M. Ritchie

The C programming language.

Prentice Hall, Inc., 1988.



The Hello World Collection



The Free Software Foundation

How people code "Hello World"