

## C++: strumenti (compilatori/editor, librerie)

Patrizia Scandurra  
[scandurra@dti.unimi.it](mailto:scandurra@dti.unimi.it)  
Info3

### Il compilatore GNU g++

- **g++ è il tool di gcc per il C++**
- **gcc = GNU Compiler Collection**  
compilatore free per eccellenza per Linux
- Installato automaticamente in quasi tutte le distribuzioni di Linux
- E' in grado di compilare C, C++, Java, Fortran, ADA, ecc. su oltre 40 famiglie di chip

## Uso del compilatore g++

- Scrivere il file sorgente, o i diversi file sorgente, con un editor di testo (ad es., *emacs*)
- Compilare l'applicazione dando il comando:

```
g++ -Wall -o <applicazione> <filesorgente>
```

- Eseguire l'applicazione:

```
./ <applicazione>
```

## Editor in Unix/Linux

- **Emacs** (by R. Stallman)
- **VI**
  - **VIM**, versione migliorata che gira su:
    - AmigaOS, DOS, Windows(95/98/NT/2000/XP/.NET/64-Bit/Embedded), BeOS, MacOS, OS/2, ecc..
- **Xwpe**
  - Interfaccia compatibile con vecchi editor del Borland C e Borland Pascal per DOS

## Librerie in Unix/Linux

- **FLTK** (“fulltick”)
  - Ambiente grafico di sviluppo, libreria ed editor per il C++ multi-piattaforma
    - Varianti di UNIX, MacOS, e MS Windows
  - Grafica 3D con OpenGL
- **GTK+**
  - Toolkit multiplatforma (prevalentemente) per lo sviluppo in C
  - Tramite la quale è stato sviluppato
    - il noto programma di grafica Gimp, Window Manager di GNOME e varie applicazioni GNOME
  - Editor per interfaccia grafica **Glade**
- **Qt**
  - Toolkit multiplatforma per lo sviluppo in C++
    - Ms/Windows, Unix/X11 (Linux, Sun solaris, e altre varianti di Unix)
  - Tramite la quale è stato sviluppato Window Manager di KDE
  - IDE per UNIX: **KDevelop**

## L'IDE Bloodshed Dev-C++

- C/C++ IDE free (GPL) per Windows 95, 98, NT, 2000, XP:  
<http://www.bloodshed.net/devcpp.html>
- GNU g++ è il “motore nascosto” dietro a Dev-C++:
  - **Mingw** port di GCC (GNU Compiler Collection)
- Dev-C++ può essere usato in combinazione con Cygwin o ogni altro compilatore basato su GCC

## Requisiti di sistema per Dev-C++

- Sistema operativo Microsoft Windows 95, 98, Millenium Edition, NT 4, 2000 o XP
- RAM: 8 Mb (consigliati almeno 32 Mb)
- CPU: Intel Pentium 100 MHz o compatibile (consigliati almeno 233 MHz)
- Spazio su disco: 30 Mb liberi (consigliati almeno 45 Mb)

## Installazione di Dev-C++

- Scaricare in una cartella temporanea il file di setup

*Dev-C++ 5.0 beta 9.2 (4.9.9.2) (9.0 MB)  
with Mingw/GCC 3.4.2*

dal sito

<http://www.bloodshed.net/devcpp.html>

- Lanciare il programma di setup e seguire le istruzioni (click su “Yes”, poi “Next”, poi “Finish”)

## Configurazione di Dev-C++ (1)

- Lanciare il programma (Start ⇒ Programmi ⇒ Dev-C++ ⇒ Dev-C++)
- Se si ha un messaggio relativo alla mancanza della libreria *wininet.dll*, scaricare `wininet.exe` ed eseguirlo
- La prima volta che si lancia il compilatore appare una finestra di dialogo: click su “chiudi”

## Configurazione di Dev-C++ (2)

- Selezionare “Strumenti ⇒ Opzioni di compilazione”
- Clicca sulla scheda “Generazione di codice / Ottimizzazione”, selezionare “Ottimizzazione” nel pannello a sx e scegliere “Ottimizzazione migliore”
- Seleziona poi “Linker” nel pannello sx, e cambia “Generate debugging information” in “Yes”

## Utilizzazione di Dev-C++

- È possibile creare due diversi tipi di applicazioni:
  - Applicazioni da console (solo testo)
  - Applicazioni Windows (grafica)

## Creare una applicazione

- 1) Creare un nuovo progetto
- 2) Scrivere i file sorgente
- 3) Compilare e linkare i file sorgente
- 4) Errori del compilatore? Correggerli tornando al passo 2.
- 5) Eseguire l'applicazione
- 6) Non funziona? Correggerla tornando al passo 2.

## Creazione di un progetto (1)

- Selezionare “File ⇒ New Project...”
- Selezionare “Console Application”
- Selezionare “C++ language”
- Click su “Ok”
- Inserire il nome del progetto e click su “Ok”
- Creare una nuova cartella e click su “Salva”  
*È consigliabile salvare ciascun progetto in una cartella diversa!*

## Creazione di un progetto (2)

- Nella cartella del progetto, Dev-C++ inserisce un file “.dev” ed un file “.rc”
- A sinistra nello schermo Dev-C++ mostra il nome del progetto e tutti i file sorgente
- Un file sorgente “main.cpp” è inserito automaticamente e mostrato a destra nello schermo

## Scrivere un file sorgente (1)

- La creazione di una applicazione implica la scrittura di un programma costituito da uno o più file sorgenti
- Modificare il file sorgente nella parte destra dello schermo

## Scrivere un file sorgente (2)

- Lo scheletro di un programma C++ (applicazione MS-DOS):

```
#include <iostream>
#include <stdlib.h>
using namespace std;
int main(int argc, char *argv[])
{
    system("PAUSE");
    return 0;
}
```



## Scrivere un file sorgente (3)

- Per generare le parentesi graffe { e } nelle tastiere italiane:
  - <AltGr> <Shift> è                      <AltGr> <Shift> +
  - <Alt> 1 2 3                                  <Alt> 1 2 5  
(sul tastierino numerico)
- Per generare ~:            <Alt> 1 2 6

## Scrivere un file sorgente (4)

- Salvare il file selezionando “File ⇒ Save” o Ctrl+s
- Assegnare un nome al file sorgente (ad esempio *main*) e click su “Salva”
- Il file `main.cpp` è inserito da Dev-C++ nel progetto

## Compilare l'applicazione

- Selezionare “Esegui  $\Rightarrow$  Compila” o Ctrl+F9
- Se la compilazione ed il linking del progetto si concludono senza errori, il bottone “>>Esegui<<” o Ctrl+F10 della finestra di dialogo viene abilitato
- Se vi sono errori, il bottone non viene abilitato: si deve correggere l'errore

## Correggere un errore

- I messaggi di errore vengono mostrati nella finestra in basso di Dev-C++
- Sono suddivisi tra errori del compilatore (i più comuni), errori del linker, ed altri
- Fare un doppio click sul *primo* errore dei messaggi del compilatore
- Dev-C++ posiziona il cursore sulla riga del file sorgente corrispondente
- Capire l'errore, correggerlo e ricompilare

## Eseguire l'applicazione

- Se Dev-C++ compila senza errori, il bottone ">>Esegui<<" è abilitato: selezionarlo
- Il programma esegue in una finestra DOS
- Controllare il funzionamento del programma, e premere un tasto
- Esiste anche il bottone "Compila e esegui" F9

## C++ source structure

- header files (.h, .hpp)
  - dependencies (`#include` directives)
  - type declarations
  - type definitions
  - function declarations
  - inline functions
- implementation files (.cpp, .cc)
  - dependencies (`#include` directives)
  - function definitions
  - `main()` function

## main()

- each C++ program needs exactly one main() function
- program entry point
- possible signatures are

```
int main(int argc, char* argv[])  
int main(void)
```

## Aggiungere file sorgenti

- È possibile aggiungere file sorgenti al progetto:
  - Selezionare “Progetto ⇒ Nuova unità”
  - Un nuovo file sorgente “Senza titolo” viene creato (si veda la parte destra dello schermo)
  - Scrivere il file e salvarlo dandogli un nome
- Dev-C++ compila automaticamente tutti i file sorgenti del progetto

## Guide introduttive

- Di base per Dev-C++ (“Help ⇒ Tutorial”)
- Dev-C++ resources site (libraries, sources, updates...):

`http://www.bloodshed.net/dev/`

## Altri ambienti Windows

Ambienti non free, molto avanzati:

- Borland C++ Builder
- Visual C++/Studio 6.0
- Visual Studio.NET / VC++ 7.0

Un altro ambiente free (anche per Linux):

- Eclipse C/C++ Development Tooling – CDT `http://www.eclipse.org/cdt/`