

Software libero per programmare C e

C++ (autore: Vittorio Albertoni)

Premessa

Il linguaggio C nasce nei Bell Labs tra il 1969 e il 1973, ad opera di Dennis M. Ritchie, come linguaggio per semplificare la scrittura del sistema operativo Unix evitando di utilizzare l'allora esistente ma poco adatto linguaggio B o addirittura il complicatissimo linguaggio Assembler.

Le origini sono pertanto quelle di linguaggio di sistema ad uso di pochi tecnici.

Nel 1978, dopo averlo arricchito per essere adatto a sviluppare applicazioni per computer, Dennis M. Ritchie e Brian W. Kernighan lo rendono di pubblico dominio con il testo *The C Programming Language*, di cui esiste la traduzione italiana "Il linguaggio C" pubblicata da Pearson.

Dal 1980, sempre nei Bell Labs, Bjarne Stroustrup si dedicò ad una ricerca per la definizione di un nuovo linguaggio di programmazione orientato agli oggetti e decise di estendere a questo scopo il linguaggio C in un nuovo linguaggio, da lui provvisoriamente battezzato C with classes.

Nel 1983 questo nuovo linguaggio entrò in uso con il nome di C++, ad indicare un C arricchito.

Il fatto che C sia nato in simbiosi con il sistema operativo Unix e C++ ne sia una derivazione fa di questi linguaggi qualche cosa di conaturato con tutto il software libero derivato dal sistema Unix: GNU e GNU/Linux.

Sul linguaggio C e sulla sua estensione C++ esiste ampia letteratura, sia in libreria sia sul web.

Non è un linguaggio da neofiti ed è abbastanza difficile da imparare e da utilizzare.

Da sempre è considerato il linguaggio che genera i programmi più efficienti dal punto di vista della velocità di esecuzione ed è il linguaggio preferito dai programmatori professionisti.

* * *

Nel mese di Ottobre del 2011, ad una settimana di distanza, se ne sono andati per sempre Steve Jobs e Dennis Ritchie.

A Steve Jobs sono stati tributati onori fin esagerati. Di Dennis Ritchie non ha parlato nessuno.

Si sappia, tuttavia, che tutto ciò che ha fatto Steve Jobs si è potuto fare grazie al linguaggio C di Dennis Ritchie e a Unix.

Indice

1	Dotazione di base	3
2	IDE di supporto	3
2.1	Geany	4
2.2	Code::Blocks	4
2.3	Qt Creator	4

1 Dotazione di base

Come per tutti gli altri linguaggi di programmazione ci serve un editor di testo, per scrivere il programma, e un compilatore, per tradurre il programma in linguaggio macchina.

L'editor di testo lo troviamo sicuramente installato con il nostro sistema operativo (Gedit, Kate o altro in Linux, Blocco note in Windows, TextEdit in Mac).

Il compilatore che ci passa il mondo del software libero si chiama GCC, che sta per GNU Compiler Collection, in origine GNU C Compiler: nato per il linguaggio C, infatti, si è arricchito di vari front end per altri linguaggi, come C++, ADA, Fortran, ecc.

Chi lavora su Linux probabilmente se lo trova installato con il sistema operativo, almeno per il linguaggio C. Per verificare scriviamo a terminale il comando `gcc` e vediamo cosa succede. Con il comando `g++` verificiamo che sia installato anche il front end per C++ ed eventualmente seguiamo le istruzioni per installarlo.

All'indirizzo <https://gcc.gnu.org/> troviamo comunque quanto serve per installare il compilatore su Linux, Windows e Mac OS X.

Chi lavora su Windows, all'indirizzo <https://jmeubank.github.io/tdm-gcc/>, trova una valida alternativa nel pacchetto TDM-GCC.

Una volta installato il compilatore e salvato il programma in un file con estensione `.c` (se si tratta di un programma in linguaggio C) o `.cpp` (se si tratta di un programma in linguaggio C++) la compilazione avviene, rispettivamente, con i comandi a terminale

```
gcc -o <nome_eseguibile> <nome_file>.c
g++ -o <nome_eseguibile> <nome_file>.cpp
```

2 IDE di supporto

Se scriviamo il programma con un editor di testo dobbiamo sapere esattamente ciò che dobbiamo scrivere, cioè dobbiamo conoscere perfettamente il linguaggio. L'unico aiuto che possiamo avere da alcuni editor è quello di vedere in colori diversi le varie parole chiave del linguaggio, in modo da riscontrare eventuali errori di battitura: per raggiungere questo scopo dobbiamo cominciare a salvare con estensione `.c` o `.cpp` il file prima ancora di scriverlo, in modo da allertare l'editor circa il fatto che stiamo scrivendo un programma in linguaggio C o C++.

Se utilizziamo una IDE possiamo avere questo e altro.

IDE sta per Integrated Development Environment.

L'IDE è un ambiente di sviluppo integrato che, in presenza della stesura di progetti complessi, costituiti da un insieme di file e di oggetti che debbano interagire tra loro per addivenire alla codifica di un programma eseguibile, rappresenta l'unica possibilità di arrivare in tempi ragionevoli ed evitando i più banali errori, ad un risultato accettabile.

Con questo strumento, infatti, possiamo radunare il nostro lavoro in progetti anche composti da molti file, ed effettuare la compilazione, con i necessari linking tra vari file, nella stessa sede in cui abbiamo scritto i file stessi.

Sempre con il vantaggio della colorazione delle parole chiave del linguaggio, ad evidenziare eventuali errori di battitura, o addirittura con il vantaggio della code completion, attraverso la quale, non appena abbiamo avviato la scrittura di una parola chiave, ci viene suggerita la parola completa.

Qui richiamo due opportunità che il software libero ci mette a disposizione per tutti i sistemi operativi più diffusi.

2.1 Geany

Lo troviamo all'indirizzo <https://www.geany.org/>, scheda DOWNLOAD ▷ RELEASES, nelle versioni per Linux, Windows e Mac OS X. Chi usa Linux può trovarlo nel Gestore dei programmi.

Salvato il file, con estensione .c o .cpp prima di scriverne il contenuto, potremo poi scrivere il programma aiutati da code completion.

Alla sua installazione automaticamente individua la localizzazione del compilatore sul computer e possiamo compilare il programma utilizzando appositi tasti nella barra degli strumenti.

E' utilizzabile per molti altri linguaggi di programmazione.

2.2 Code::Blocks

Lo troviamo all'indirizzo <http://www.codeblocks.org/>. Aprendo la scheda DOWNLOAD ▷ BINARIES abbiamo la possibilità di scegliere la versione per il nostro sistema operativo, Linux, Windows o Mac OS X.

Se lavoriamo su Windows, per avere garanzia di funzionamento, anche se abbiamo già installato un compilatore, è bene che scarichiamo la versione comprensiva di MinGW.

La versione che ci viene proposta per i vari sistemi operativi è l'ultima uscita, nel momento in cui scrivo la 20.03 del marzo 2020. Se abbiamo un sistema operativo non di ultima generazione conviene scaricare una versione precedente: le troviamo cliccando sulla scritta FOR OLDER RELEASE VERSIONS PLEASE CHECK HERE in fondo alla pagina.

Se lavoriamo su Linux possiamo utilmente installare la versione che ci è proposta dal gestore dei programmi.

Code::Blocks ci offre la code completion più efficiente possibile.

Sia pure con qualche non banale lavoro di configurazione possiamo affidarci a Code::Blocks per creare programmi dotati di GUI (Graphical User Interface).

Ottimi risultati si ottengono, sempre stando nel software libero, con wxWidgets: all'indirizzo <https://wiki.wxwidgets.org/Install> troviamo una guida alla predisposizione di Code::Blocks in tal senso. Installando il plugin wxSmith abbiamo a disposizione un tool per disegnare visualmente la GUI.

2.3 Qt Creator

Le librerie Qt non sono esattamente software libero e sono pure costose.

Per gli amanti del software libero e dell'open source la Qt Company ha tuttavia un occhio di riguardo e mette loro a disposizione i suoi prodotti gratuitamente contro l'impegno a che essi vengano utilizzati esclusivamente per progetti open source e comunque non vengano utilizzati a scopi commerciali.

Se vogliamo favorire andiamo all'indirizzo <https://www.qt.io/> e clicchiamo sul pulsante DOWNLOAD.TRY.

Scorriamo la pagina fino a raggiungere la zona DOWNLOAD FOR OPEN SOURCE USERS, scegliamo GO OPEN SOURCE e poi DOWNLOAD THE QT ONLINE INSTALLER. Viene riconosciuto il sistema operativo su cui siamo e ci viene proposto l'installer adatto.

Per scaricarlo e procedere all'installazione dobbiamo superare una serie di adempimenti burocratici che vanno dall'apertura di un account gratuito alla conferma della nostra email ed all'assunzione degli impegni previsti per la disponibilità gratuita del software.

Ad installazione avvenuta avremo a disposizione un ottimo ambiente di sviluppo per programmare applicazioni console (per le quali potremmo benissimo limitarci ad usare Geany) ma e soprattutto per programmare applicazioni dotate di interfaccia utente grafica potendo utilizzare un designer visuale.