«Sviluppo, compilazione ed esecuzione» Antonio Tufano

Ambienti di sviluppo

- La complessità e la precisione richiesta dal C/C++ è tale che un buon IDE può fare sensibilmente la differenza in termini di produttività e qualità del codice.
- Ne analizzeremo 3:
 - Eclipse;
 - Visual Studio;
 - Code-Blocks.

Ambienti di sviluppo: Eclipse

- Eclipse è un ottimo strumento multipiattaforma e multilinguaggio.
- La versione CDT (scaricabile direttamente dal sito ufficiale) integra tutti gli strumenti essenziali per lo sviluppo.
- Tutto questo all'interno di una interfaccia.



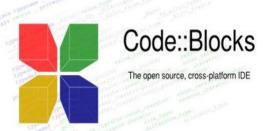
Ambienti di sviluppo: Visual Studio

- Visual Studio è un ottimo strumento per la programmare in C e in C++ e dà il meglio di se si utilizzano frequentemente tecnologie quali DirectX o le API di Windows.
- L'applicazione nella sua forma completa è a pagamento. Tuttavia è disponibile gratuitamente la versione Express.



Ambienti di sviluppo: Code-Blocks

- Code-Blocks è stato ed è tutt'ora uno degli IDE per C e C++ open source più conosciuto fra amatori e professionisti della programmazione. Punta di diamante di questa applicazione è la gran varietà di wizard e di estensioni di terze parti che sono stati sviluppati nel corso degli anni.
- L'interfaccia è un po' vecchio stile e non ha subito radicali cambiamenti negli ultimi anni. Un IDE da tenere d'occhio



Fasi di compilazione

- Il preprocessore elabora le direttive #include, #define, . . . modificando il sorgente
- Il **compilatore** traduce il codice C in linguaggio macchina:
 - 1) con ottimizzazione (della velocità di esecuzione o della dimensione dell'eseguibile)
 - 2) senza ottimizzazione (per il debug)
- Il **linker** assembla in un unico file eseguibile:
 - i file oggetto prodotti da diversi file sorgente
 - le librerie (I/O, matematiche, network, ecc.)

Fasi di compilazione

Il compilatore verifica la correttezza del codice C e produce due tipi di errori:

- Error: errori sintattici, impediscono la generazione del codice eseguibile
- Warning: errori non sintattici che non impediscono la generazione del codice eseguibile; i warning segnalano un possibile (e altamente probabile) problema che il compilatore risolve in base a regole generiche (ma attenzione: la soluzione generica potrebbe non essere quella corretta)

Un codice pulito non deve produrre né errori né warning

- Un programma in C può essere scritto anche con un semplice editor di testi come il Blocco note di Windows (o simile).
- Per poter vedere il risultato del codice dobbiamo salvare necessariamente il file con un'estensione ".c", ad esempio "hello.c".
- A questo punto non dobbiamo fare altro che compilare il programma.

- La compilazione di un programma in C segue varie fasi:
 - ➢ Il codice sorgente viene controllato dal preprocessore che ha i seguenti compiti:
 - rimuovere eventuali commenti presenti nel sorgente;
 - interpretare speciali direttive per il preprocessore denotate da "#", come #include o #define.
 - controllare eventuali errori del codice

- La compilazione di un programma in C segue varie fasi:
 - ➢ Il risultato del preprocessore sarà un nuovo codice sorgente "pulito" ed "espanso" che viene tradotto dal compilatore C in codice assembly;
 - L'assembler sarà incaricato di creare il codice oggetto salvandolo in un file (.o sotto Unix/Linux e .obj in Dos/Windows);
 - ➤ Il Link editor ha il compito di collegare tutti i file oggetto risolvendo eventuali dipendenze e creando il programma (che non è altro che un file eseguibile).

- Tutte queste operazioni, per quanto molto avanzate, possono individuare facilmente errori di sintassi, ma mai potranno trovare errori logici (come ad esempio un ciclo che non finisce), anche se spesso vengono segnalati dei **Warning** che non costituiscono errore, ma che segnalano parti di codice strane e quindi sulle quali porre attenzione per eventuali errori logici.
- Qui ci viene in aiuto il debugger che ha il compito di aiutare a gestire la correzione degli errori con svariati strumenti come i breakpoint ed i watchpoint.

- Se possedete un ambiente di sviluppo come Eclipse o Visual Studio dovete solamente cercare il comando "Compila" od "Esegui".
- Il programma provvederà, non solo ad eseguire tutte le operazioni citate, ma, una volta compilato, manderà in esecuzione il vostro programma presentandovi il risultato a video.

Per chi stesse usando, invece, compilatori come il GCC, la compilazione (intesa in tutte le sue fasi) deve essere fatta da riga di comando (dalla shell per capirsi), e, successivamente, sempre da riga di comando, si deve lanciare il file eseguibile risultante (in linux con gcc, di default il file eseguibile si chiama a.out, mentre in ambiente Dos/Windows generalmente ha lo stesso nome del sorgente ma con estensione .exe, invece che ".c").

Compilazione: GCC

- GCC (GNU Compiler Collection, in origine GNU C Compiler) è un compilatore multi-target.
- Creato dal fondatore della Free Software Foundation, Richard Matthew Stallman, come parte del Progetto GNU.
- Le versioni recenti sono incluse nelle principali distribuzioni del sistema operativo GNU/Linux, e di molti altri sistemi, mentre su macOS non è installato per default, ma è parte del pacchetto Xcode.

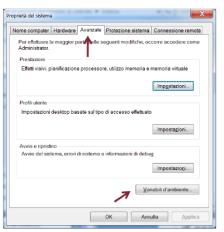


Installazione di GCC in ambiente Windows

- Esistono varie procedure. Noi suggeriamo questa:
 - 1. Scaricare il file: MinGW.zip
 - 2. Estrarre il folder MinGW e copiarlo nella root del disco di sistema C, in modo che diventi C:\MinGW
 - 3. Aggiornare la variabile di ambiente PATH in modo che includa nel percorso anche C:\MinGW\bin
- NOTA II path (letteralmente cammino) è il percorso, o meglio l'insieme di percorsi, nel file system lungo i quali il sistema operativo ricerca i file eseguibili, cioè i file dei quali comandiamo l'esecuzione. In Windows il path è specificato dal valore della variabile di ambiente PATH

Installazione di GCC in ambiente Windows

- Aprire Pannello di Controllo → Sistema
- 2. Scegliere Impostazioni di sistema avanzate
- 3. Scegliere il tab Avanzate e poi Variabili di ambiente
- 4. Scegliere Modifica e selezionare PATH

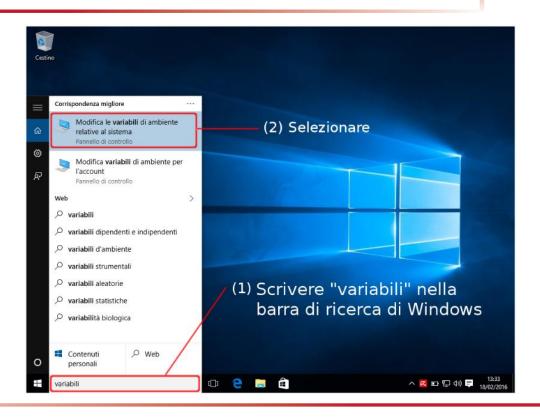




Aggiungere in fondo al PATH la stringa:
;C:\MinGW\bin
Premere OK



Installazione di GCC in ambiente Windows 10



Installazione di GCC su Linux

- Installazione mediante gestori di pacchetti:
 - Processi di installazione, aggiornamento e configurazione automatizzati.
 - Non dobbiamo compilare il codice sorgente e configurare manualmente gcc
- Installazione da linea di comando
- Principali gestori di pacchetti analizzati:
 - Advanced Packaging Tool (APT): per distribuzioni Debian e sue derivate (e.g. Ubunutu, Kubuntu)
 - ZYpp package management engine: distribuzioni openSUSE/SUSE
 - YUM package installer: distribuzioni RPM quali Red Hat Linux, Fedora, CentOS

Installazione di GCC su Linux

Installazione su Debian, Ubuntu e derivate:

*sudo apt-get install build-essential

Installazione su openSUSE/SUSE:

sudo) zipper install gcc

Installazione su Red Hat Linux, CentOS, Fedora e derivate:

sudo yum install gcc

I comandi sono eseguiti come superuser

Installazione di gcc su Mac OS X

- GCC è automaticamente installato con l'installazione dei Command Line Tools.
- Command Line Tools reperibili sul disco di installazione del sistema Mac OS X fornito all'acquisto della macchina Apple.
- In alternativa è possibile scaricare i Command Line Tools dal sito Apple's developer.
- Nota: il compilatore è tipicamente distribuito con Apple Xcode.

Installazione di gcc su Mac OS X

- Registrazione Apple account:
 - Accedere alla pagina
 https://developer.apple.com/downloads/index.action e registrare un account Apple. Eseguire poi l'accesso utilizzando le nuove credenziali.
- Command Line Tools per Xcode:
 - Selezionare la categoria "Developer Tools" e quindi "Command Line Tools for Xcode". Scaricare il file con estensione .dmg
- Installazione:
 - Doppio click sul file Command Line Tools.mpkg nella finestra di dialogo mostrata dopo il download del file .dmg. Seguire poi la guida all'installazione.

Compilare in Linux

- Accedi alla directory in cui hai salvato il codice sorgente del programma da compilare.
- Ad esempio, se il codice da compilare è memorizzato all'interno del file "main.c" salvato nella directory "/usr/wikiHow/source", dovrai usare il seguente comando cd /usr/wikiHow/source per raggiungere il percorso corretto da cui eseguire la compilazione.

Compilare in Linux

- Digita il comando gcc main.c -o HelloWorld
- Sostituisci il parametro "main.c" con il nome completo del file che contiene il codice sorgente da compilare.
- Il parametro "HelloWorld" indica il nome che verrà assegnato al file compilato; anche in questo caso, puoi scegliere il nome che preferisci.
- Eseguendo il comando indicato, il codice sorgente verrà compilato.

Compilare in Linux

- Se durante la compilazione vengono individuati degli errori e hai la necessità di avere maggiori dettagli al riguardo, usa il comando gcc -Wall -o errorlog main.c. Verrà generato il file "errorlog" che contiene le informazioni dettagliate riguardo agli errori rilevati durante la compilazione, visualizzabili usando il comando cat errorlog
- Per compilare un programma costituito da un codice sorgente distribuito su più file, puoi usare il comando gcc -o outputfile file1.c file2.c file3.c
- Per compilare contemporaneamente programmi differenti, ognuno costituito dal proprio codice sorgente, puoi usare il comando gcc – c filel.c filel.c filel.c

Eseguire in Linux

- Esegui il file del programma appena compilato.
- Per farlo, digita il comando ./HelloWorld sostituendo il parametro "HelloWorld" con il nome del file compilato che hai appena creato.

Compilare in Windows

- Avvia il prompt dei comandi di Windows come amministratore del sistema.
- Segui queste istruzioni:
 - Premi la combinazione di tasti di scelta rapida
 Win+S, quindi digita la parola chiave cmd.
 - Seleziona l'icona "Prompt dei Comandi" dall'elenco dei risultati apparso, quindi scegli l'opzione "Esegui come amministratore".
 - Premi il pulsante "Sì" per aprire il prompt dei comandi di Windows in modalità amministratore del sistema.

Compilare in Windows

- Accedi alla cartella in cui hai salvato il codice sorgente del programma da compilare.
- Ad esempio, ipotizzando di aver salvato il file "helloworld.c", contenente il codice sorgente, all'interno della cartella "C:\Sorgenti\Programmi", dovrai usare il comando cd C:\Sorgenti\Programmi.

Compilare in Windows

- Digita il comando gcc helloworld.c -o helloworld.exe. Sostituisci il nome "helloworld" con quello che hai assegnato al file che contiene il codice sorgente e con quello che desideri assegnare all'applicazione finale.
- A compilazione conclusa, se non saranno stati rilevati errori, verrai reindirizzato al prompt dei comandi.
- Perché la compilazione vada a buon fine, occorre che tutti gli errori rilevati dal compilatore vengano corretti.

Eseguire in Windows

- Per eseguire il file compilato digita il relativo nome.
- Nel nostro esempio dovrai digitare il comando "helloworld.exe" e premere il tasto "Invio".