Appunti:

MASTER COPY

Tel. 388/9837745

Compilazione da riga di comando

- Per poter compilare un file sorgente .cpp o .c bisogna trovarsi all'interno della cartella bin di Dev-cpp. Qui infatti è presente un file eseguibile, g++.exe che è il compilatore del nostro programma.

- FILE SORGENTE: il file sorgente è un file di testo con estensione .cpp o .c, per C e C++, in cui viene scritto il codice, ossia le istruzioni, che il nostro calcolatore dovrà eseguire.

- Per poter scrivere un programma, quindi per poter scrivere le istruzioni in un file sorgente, avremo bisogno di un editor, ossia di un software con cui potremo scrivere e salvare le istruzioni sul file. Esistono molteplici editor con vari strumenti; il mediamente esperto opterà per un linguaggio di programmazione in cui troviamo programmando, prenderà il nome di IDE, cioè ambiente di sviluppo, composto dall'insieme di strumenti quali debugger, compilatore e linker.

- Compilatore: un compilatore è uno strumento che permette di tradurre il codice scritto dal programmatore in un linguaggio più comprensibile per il calcolatore. Siccome il computer comprende solo 0 e 1 e il codice del programmatore è composto da vocaboli molto simili al linguaggio umano (do, while, for, int, main, ecc.), questo deve essere tradotto.

- Compilatore: strumento su la traduzione delle istruzioni del programma in linguaggio macchina (0 e 1). Esegue preliminarmente un'analisi sintattica e lessicale individuando eventuali errori

- La compilazione: la compilazione, come abbiamo appena detto, prevede la traduzione in linguaggio macchina. Questa traduzione non avviene però direttamente, poiché prima si passa ad un linguaggio intermedio detto ASSEMBLER, per poi tradurre quest'ultimo in linguaggio macchina.

- Fasi di compilazione:

(Errori) 1. Verifica della correttezza del codice. (errori dal compilatore)

(.o) 2. Se il codice è corretto, viene creato un file .o tradotto in linguaggio

(Linking) 3. Una volta creato il .o il linker collega il codice oggetto con le librerie invocate dal programma, poiché molte istruzioni sono definite in altre librerie

(.exe) 4. Terminato il linking viene prodotto un file risultante .exe, eseguibile dal calcolatore.

- Prima della compilazione viene avviato un preprocessore che espande tutte le istruzioni precedute dal # (collettivo), ovvero l'inclusione di librerie e la sostituzione delle costanti.

1: Tipi di errori che possono essere riscontrati sono:

- sintattici = errore di trascrizione del codice (es. scordo del ;).
- semantici = errore nello scrivere le regole dello (Floooat).
- semantici = errore logico, visibile solo durante l'esecuzione (es. dividere per 0).

2: Solo quando il codice è corretto viene creato un file .o, dove il codice del sorgente viene tradotto in linguaggio macchina comprensibile al calcolatore, ma non ancora eseguibile.

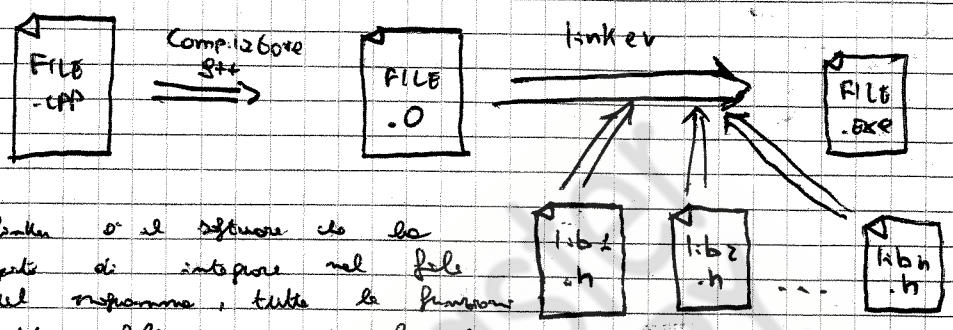
3: Il linker ha la funzione di collegare il programma ai file [illegible] corrispondenti a queste istruzioni, definite nei file .o

nella libreria iostream.

Si deve volta eseguito il linking viene creato un file eseguibile dal collaboratore, poiché tutte le istruzioni "non completamente definite" del programmatore vengono sostituite con il relativo codice presente nelle librerie incluse.

- Schema di compilazione

MASTER COPY
Tel. 388/9837745



*3: Il linker o il software che ha il compito di integrare nel file oggetto del programma, tutte le funzioni che una data libreria ci fa utilizzare.

- g++ <nomefile>.cpp : Compila e crea un file .exe con nome e output dalla libreria e lo contenuto in ordine alfabetico ad ogni nuovo file .exe.

- g++ <nomefile>.cpp -o <nomefile>.exe : Compila e crea un file .exe con nome dato da <nomefile>.

- g++ -c <nomefile>.cpp : Compila il file .cpp fermandosi prima della fase di linking, esegue solo la traduzione in codice e l'analisi degli errori, generando un file .o, con lo stesso nome del .cpp

- g++ -c <nomefile>.cpp -o <nomefile>.o : esegue la stessa operazione della precedente istruzione dando nome al file .o, uno a scelta dopo -o.

- g++ <nomefile>.o : esegue la compilazione del file, eseguendo la fase di linking e generando un file .exe con stesso nome e contenuto alfabeticamente ad ogni creazione di un eseguibile

- g++ <nomefile>.o -o <nomefile>.exe: (vedi sopra) con nome scelto da noi

classe di astrazione -0.

Metalinguaggio del c++

- I 2 principali aspetti di un linguaggio sono la semantica e la sintassi.

- Sintassi: insieme delle regole che permette di costruire correttamente le "frasi" di un linguaggio.

- semantica: definisce il significato delle frasi.

- Il Metalinguaggio definisce quest'ultimo aspetto di un linguaggio di programmazione, ovvero la sintassi e la semantica.

- Il tipo di notazione utilizzato per il nostro metalinguaggio sarà la BNF (Backus-Naur Form).

Esempio

frase
Soggetto verbo .

soggetto
articolo nome

articolo
one of
il lo

nome
one of
lupo coniglio bosco cielo scoiattolo

verbo
one OF
mangia vola corre

- Le categorie sintattiche, ossia le cose da definire, vengono scritte in corsivo secondo la notazione, mentre i simboli terminali sono scritti con Font da macchina da scrivere (Font. monospaziato), ed infine i simboli del metalinguaggio in grassetto.

MASTER COPY
Tel. 388/9837745

- alcuni elementi possono essere omessi e sono opzionali e in questo caso si parla di categorie sintattiche opzionali.

Esempio

Soggetto
articolo aggettivo|opt nome

Aggettivo
one of
mio tuo

- I simboli terminali per distinguerli l'uno dall'altro, devono essere separati da spaziature.

- Nel metalinguaggio BNF si possono definire anche una sequenza di elementi.

Esempio

Frase
Soggetto verbo sequenza-complementi|opt

Sequenza-Complementi
complemento
complemento sequenza-complementi

Complemento
articolo nome preposizione nome

Preposizione
one of
nel nello

:
:

Categorie Sintattiche in Gt

- Le principali categorie sintattiche sono:
  - Identificatori
  - Parole riservate
  - Espressioni letterali
  - Operatori
  - Separatori