

Introduzione a Dev-C++

Università degli Studi di Brescia

Docente: Massimiliano Giacomini

Note:

- Dev-C++ richiede Windows 95/98/NT/2000/XP
- All'indirizzo www.codeblocks.org è reperibile anche Code::Blocks, per:
 - Windows 2000 / XP / Vista
 - Linux (Ubuntu & Debian, 32 & 64 bits)
 - Mac OS X 10.4+
- Un qualunque compilatore va bene per preparare l'esame (non ci saranno domande/esercizi relative ad un compilatore piuttosto che ad un altro)

Dal linguaggio ad alto livello al “linguaggio macchina”

- Il calcolatore rappresenta i dati (e le istruzioni) in linguaggio binario
- I nostri programmi in C hanno invece una forma del tipo:

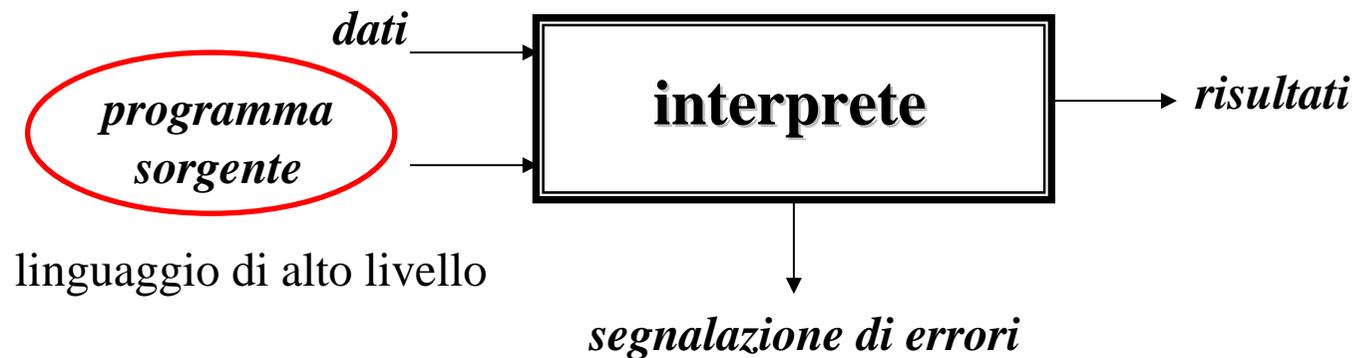
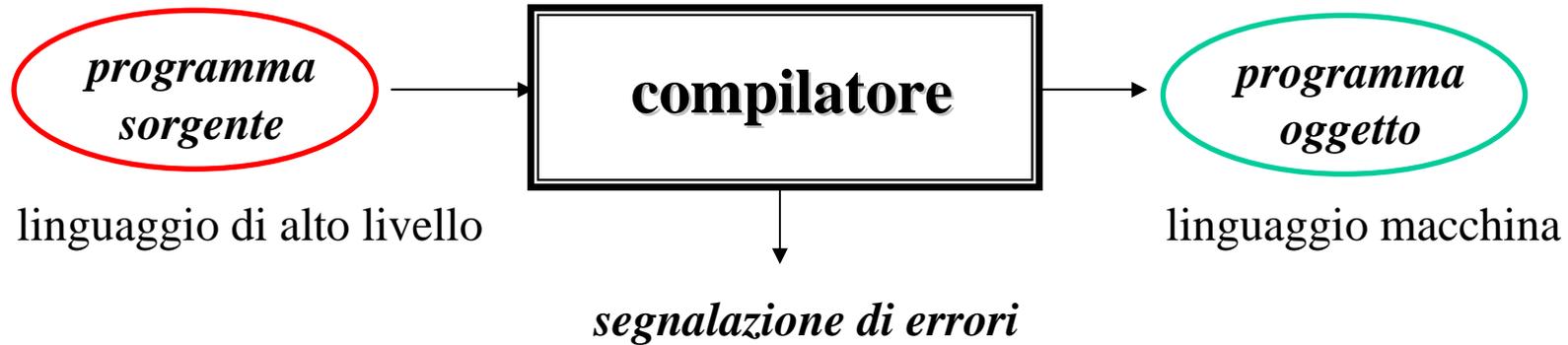
```
#include<stdio.h>

main(){
    int primo, secondo;
    int somma;

    printf("Inserisci il primo numero\n");
    scanf("%d",&primo);
    printf("Inserisci il secondo numero\n");
    scanf("%d",&secondo);
    somma=primo+secondo;
    printf("Somma uguale a %d\n",somma);
    system("pause");
}
```

- Evidentemente, devono essere “trasformati” in linguaggio macchina

Compilatore vs. interprete

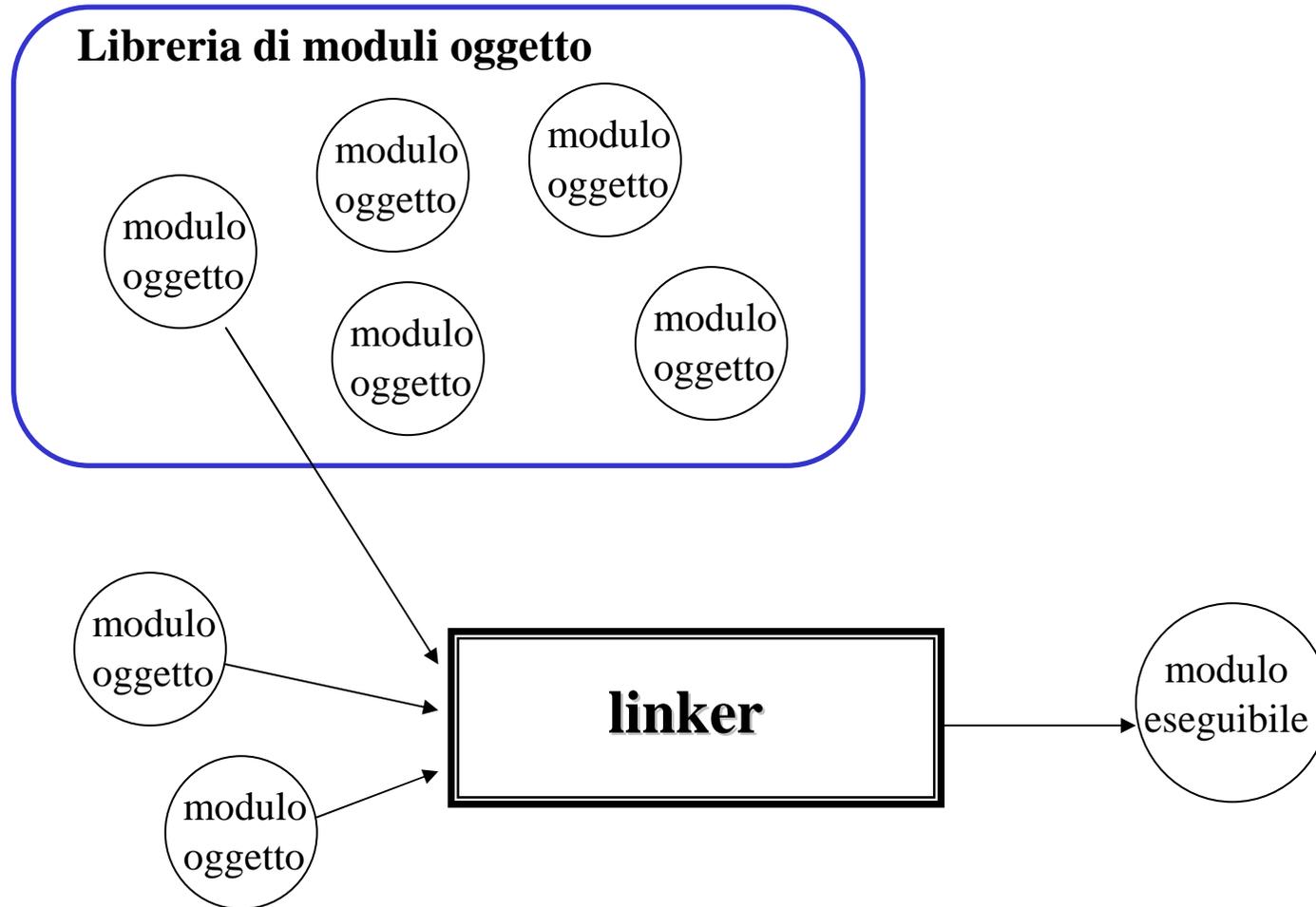


- Noi useremo un *compilatore*:
programma che traduce tutte le istruzioni di un linguaggio di alto livello (il C) in linguaggio binario, che può essere interpretato dal calcolatore
- In realtà, il programmatore può usare delle *librerie* standard, che forniscono alcune “funzionalità” comuni in modo che non debbano essere programmate ogni volta (es: input/output)
- Queste librerie corrispondono a moduli in linguaggio binario, che vengono “collegati” ai moduli sviluppati dal programmatore da un programma chiamato *linker*

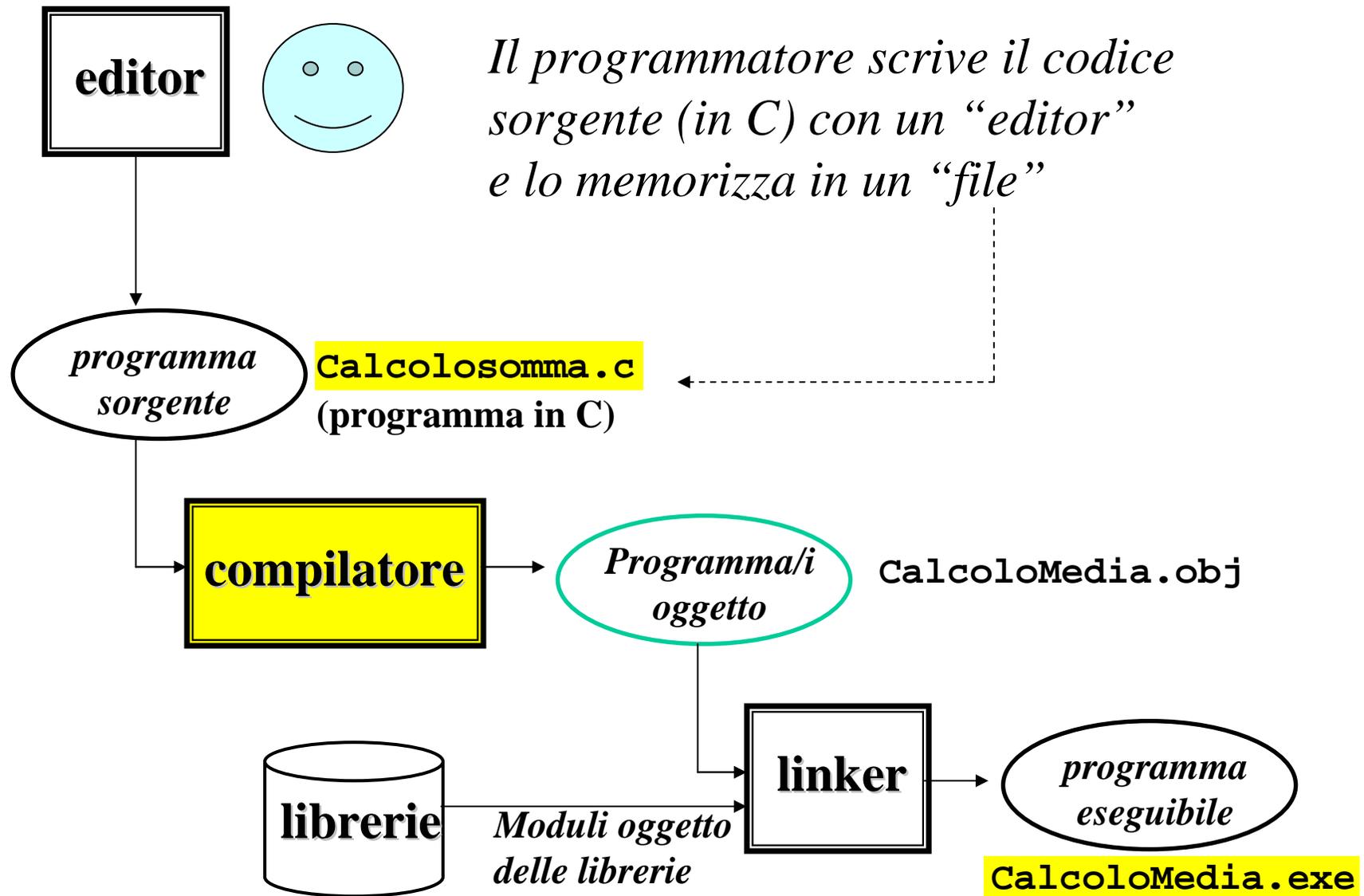
Compilatore e linker

- I compilatori consentono tipicamente la compilazione separata di parti di programmi (*moduli*)
- I diversi moduli possono essere progettati, costruiti e messi a punto separatamente, e archiviati in opportune *librerie*
- Nel momento in cui un programma deve essere eseguito, un programma apposito, detto *linker*, si occupa di ritrovare e collegare opportunamente fra loro i moduli oggetto
- Il risultato del linker è un unico modulo, detto *modulo eseguibile*, pronto per il caricamento in memoria e l'esecuzione

Il ruolo del linker



Lo sviluppo di un programma (semplificato)



IDE

- Integrated Development Environment: Ambiente di sviluppo integrato
 - > comprende una varietà di strumenti coordinati per supportare il processo di sviluppo dei programmi (creazione, traduzione, esecuzione, test, ...), tra cui:
 - editor
 - compilatore
 - linker
 - debugger
 - strumenti per la gestione delle configurazioni
 - analizzatori statici, strumenti per il test, ...
 - > riesce a supportare e “automatizzare” (parte del) proc. di sviluppo
- Dev-C++: IDE per programmi C/C++
 - free
 - tra i più semplici a disposizione(ma noi ne useremo comunque solo una piccola parte!)

Installazione di Dev-C++

- Scaricare il programma di installazione (vedi sito esercitazioni) ed eseguirlo, selezionare il linguaggio per installazione ed accettare i termini di licenza
- Select type of install: scegliere **full**
- Scegliere la directory in cui si desidera sia installato Dev-C++ (conviene lasciare quella preimpostata)
- Scegliere se installarlo per tutti gli utenti o meno + *FINISH*
- Si arriva alle finestre di configurazione:
 - Scegliere il linguaggio (english forse è la scelta migliore) + *NEXT*
 - Scegliere se installare la caratteristica di “completamento automatico del codice” + *NEXT* [NB: non necessaria per noi]
 - in caso affermativo, appare un'altra finestra in cui si chiede se usare una cache per ottimizzare il processo [NB: come volete] + *OK*

Creazione di un progetto

- Per usare Dev-C++ occorre creare un “progetto” che include tutti i file necessari nel processo di sviluppo: non solo .c, .obj, .exe ma anche tutti i file necessari a Dev-C++ per gestire l’intero processo
- Per creare un progetto:
 - > Menu *File/New/Project*
 - > Appare una finestra per selezionare le caratteristiche del progetto:
 - tipo: console, windows, vari tipi di librerie, empty project
 - linguaggio: C++, C
 - nome del progetto (es. primoprogramma): corrisponderà al nome dell’eseguibile (es. primoprogramma.exe)
 - > Viene data la possibilità di selezionare la directory in cui saranno creati tutti i file del progetto
(consiglio: per ogni progetto createvi una directory separata, p.es. primoprogramma)

Creazione/aggiunta di un file sorgente

- Una volta creato un progetto, il passo successivo è quello di includere un file sorgente .c (creandone uno nuovo o aggiungendo file esistente):
 - > Per creare un file nuovo .c:
Menu *File/New/Source File* oppure Menu *Project/Source File*
(viene creato un file senza titolo, comunque il programma ne chiede il nome prima della compilazione o salvataggio)
 - > Per aggiungere un file esistente:
Menu *Project/Add to Project*
 - > Per modificare un file (scrivere il programma!): finestra centrale
 - > I file possono essere selezionati con la finestra di sinistra e rinominati con tasto destro del mouse + *Rename file*

Compilazione ed esecuzione

- Compilazione + linking:
Menu *Execute/Compile*
Se è tutto ok compare la scritta “Done”, altrimenti nella finestra in basso compare una lista di errori (con il doppio click viene sottolineata la riga del codice sorgente corrispondente)
- Esecuzione (dopo aver compilato):
Menu *Execute/Run*

Salvataggio e caricamento

- Per i nostri scopi, si può salvare l'intero progetto con
Menu *File/Save all*
- Per caricare un progetto esistente:
Menu *File/Open project or file*
e caricare il file progetto (.dev)

Ancora sul “tipo” di progetto

- A partire da un “empty project” si può fare tutto
- Tuttavia, Dev-C++ rende disponibili diversi “tipi” di progetti che includono automaticamente frammenti di codice e librerie, specie per l’interazione con il sistema operativo:
 - > Console Application: finestra DOS
 - > Windows Application: finestra windows
 - > Static Library
 - > DLL
- Noi sviluppiamo “console applications” e quindi potremmo partire anche dal tipo “Console Application”... potete provare direttamente!