

Fondamenti Teorici

Antonio Pescapè e Marcello Esposito

Parte Quarta

v1.0

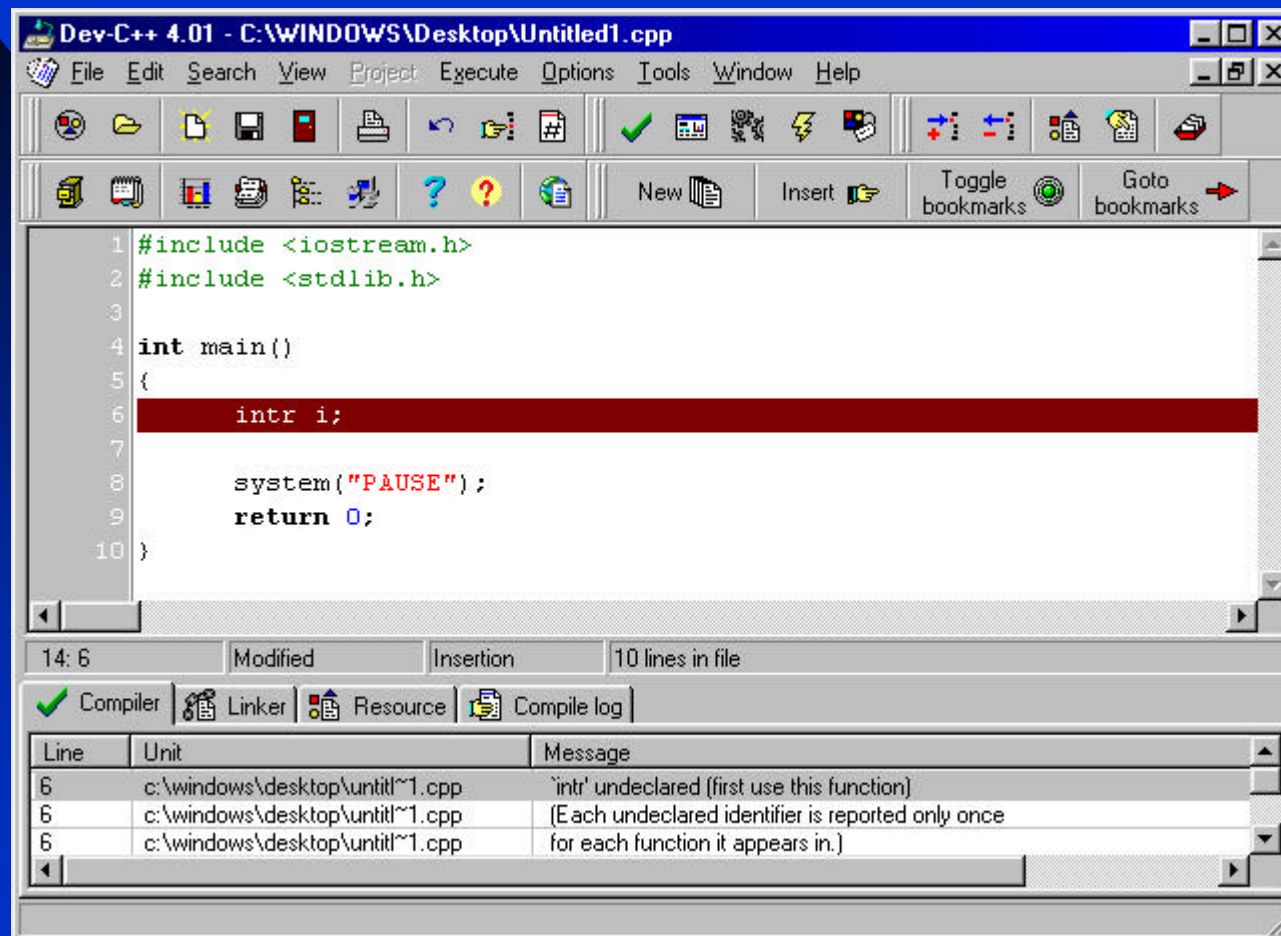
Agenda

- Ambiente di sviluppo del software
- Le fasi di sviluppo del programma
- La documentazione dei programmi
- Errori di programmazione
- Errori nei programmi
- Tecniche per limitare gli errori
- Testing

Ambiente di sviluppo del software

- Lo sviluppo del software viene assistito nelle sue fasi da distinti moduli del software di base: *editor, compilatore, file system, sistema per la diagnosi degli errori, supporto al run-time, debugger*. Può inoltre essere presente un apposito software per la gestione ed il controllo del processo di progettazione.
- L'insieme di questi elementi viene di solito indicato come *ambiente di sviluppo del programma*.
- Ambiente di sviluppo più sofisticati mettono a disposizione del programmatore strumenti (*tools*) di supporto per tutte le fasi di progettazione, di realizzazione e di gestione del software (*CASE, Computer Aided Software Engineering*)

Esempio: il Dev-C++



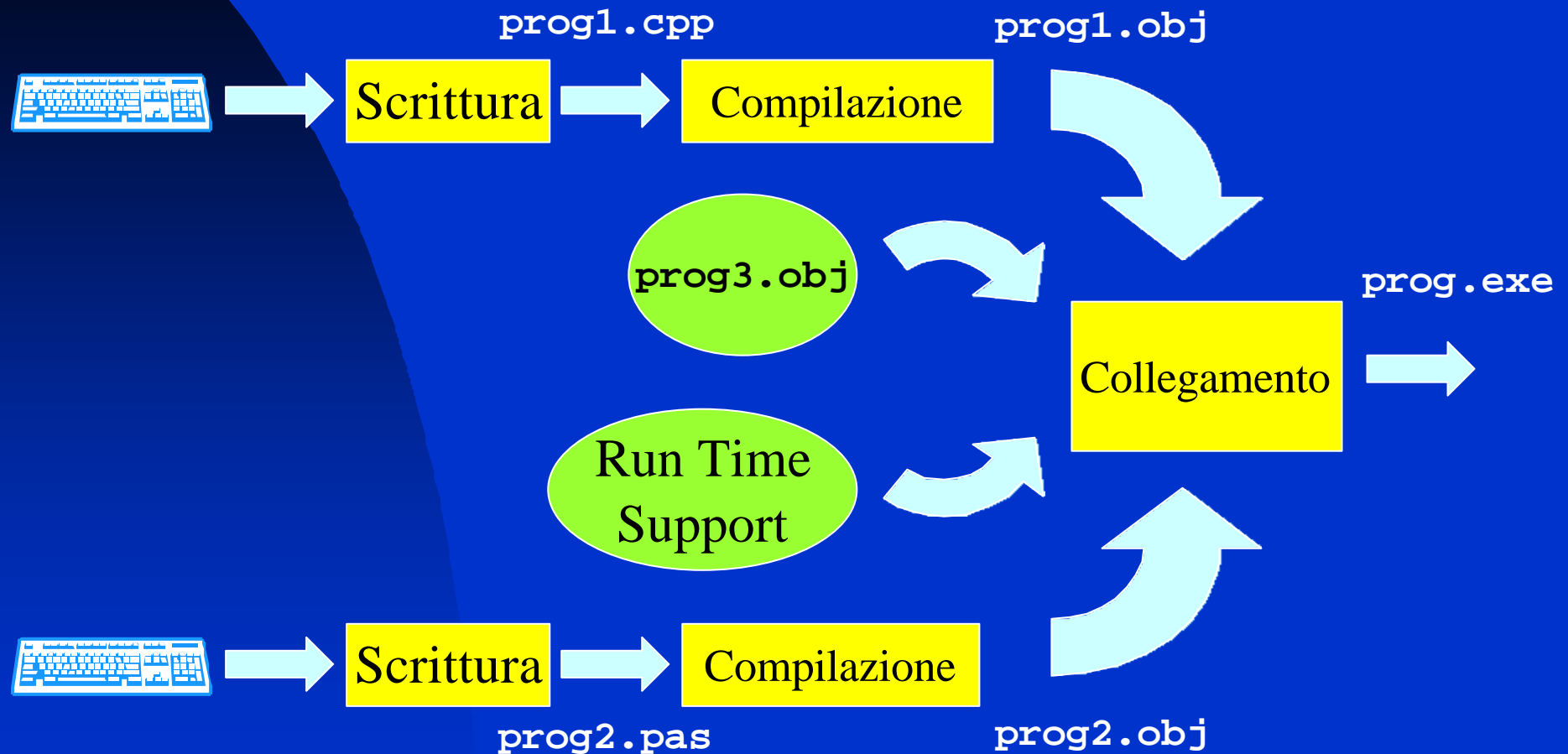
The screenshot shows the Dev-C++ 4.01 IDE. The main window displays a C++ program with the following code:

```
1 #include <iostream.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     intr i;
7
8     system("PAUSE");
9     return 0;
10 }
```

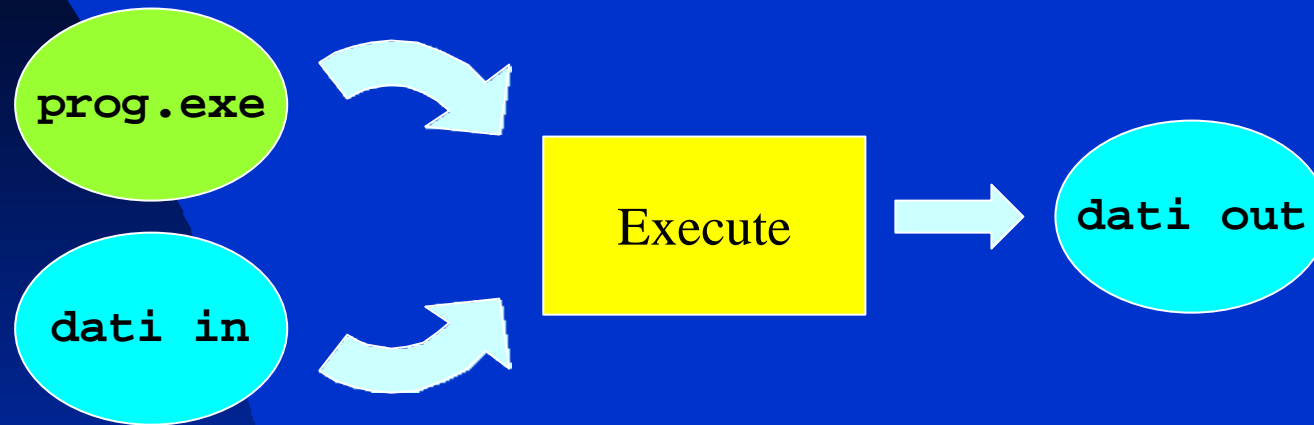
The line containing `intr i;` is highlighted in red, indicating a compilation error. The status bar at the bottom of the editor shows "14: 6", "Modified", "Insertion", and "10 lines in file". Below the editor, the "Compiler" tab is active, showing a table of error messages:

| Line | Unit | Message |
|------|--------------------------------|---|
| 6 | c:\windows\desktop\untit~1.cpp | 'intr' undeclared (first use this function) |
| 6 | c:\windows\desktop\untit~1.cpp | (Each undeclared identifier is reported only once |
| 6 | c:\windows\desktop\untit~1.cpp | for each function it appears in.) |

Il ciclo di vita di un programma



L'esecuzione



Dopo la fase di esecuzione, così come dopo la fase di compilazione, potrebbe essere necessario intraprendere la fase di debugging.

Editing e compilazione

- Preparazione del programma

- Editing del testo origine e memorizzazione in un file `p.org`
- Editor, che è un text-editor e non un word processor

- Compilazione

- Traduzione in linguaggio oggetto di ciò che è memorizzato in `p.org`
- Analisi di correttezza sintattica, segnalando eventuali errori
- Memorizza il testo oggetto in un file `p.obj`
- Il compilatore è una realtà che corre nell'ambiente del SO. Può essere riguardato come un suo sottoprogramma :

intestazione : **nomelinguaggio** (**input**, **output**, **opzioni**)

attivazione : **nomelinguaggio** (**p.org**, **p.obj**, **opzioni di compilazione**)

Collegamento

- Il collegatore (linker) procede alla collazione dei moduli di programma oggetto costituenti il programma eseguibile
- Collega tra di loro:
 - i moduli dell'utente (`prog1.obj`);
 - eventuali moduli di altri utenti (`prog2.obj`);
 - librerie di funzioni disponibili (`prog3.obj`);
 - il supporto al tempo di esecuzione.
- Al termine, il tutto viene memorizzato in un file eseguibile: `prog.exe`
- Il linker è concettualmente chiamato con una istruzione del sistema operativo del tipo

`link (p.obj1, p.obj2,, p.objn, opzioni)`

Esecuzione

- Richiamando il programma `prog.exe`, l'eseguibile viene caricato in memoria ed attivato; all'atto della terminazione del programma il controllo torna al SO
- Il supporto al tempo di esecuzione gestisce i collegamenti con il SO e garantisce la diagnostica degli errori dinamici come ad esempio l'overflow e l'invasione di memoria. Gestisce inoltre le operazioni di I/O, le allocazioni e le deallocazioni dinamiche ed in generale tutte le operazioni non direttamente programmate nel testo origine.

Correzione degli errori

- La fase di testing ha lo scopo di far emergere la presenza di errori o malfunzionamenti.
- La fase di debugging ha lo scopo di ricercare ed individuare le cause dei malfunzionamenti.
- La fase di debugging è una delle fasi del processo di correzione e consiste nel far girare il programma controllandolo con l'apposito software (di debug). In questo modo è possibile visualizzare gli stati intermedi attraverso i quali evolve il programma stesso: ad esempio, è possibile visualizzare i valori delle variabili del programma dopo l'esecuzione di ciascuna istruzione (trace).

La documentazione dei programmi (1)

- Leggibilità
 - ◆ interazione uomo-macchina e interazione uomo-uomo
 - ◆ Tecniche di costruzione dei programmi (top-down, decomposizione modulare, ...)
- All'interno dei programmi: Motivazioni (cosa si intende fare), Asserzioni (condizioni cui devono soddisfare alcuni dati), e Commenti relativi ai dati (definire gli attributi delle variabili).
- Mi raccomando....cauti con i commenti

La documentazione dei programmi (2)

- Albero di attivazione delle procedure
- Albero di struttura dei moduli
- Albero delle relazioni d'uso

- Documentazione Esterna (cosa fa il programma)
- Documentazione Interna (come lo fa)
- Documentazione delle biblioteche

Errori di programmazione

- *Errori nella definizione del problema*
- *Errori nella scelta dell'algoritmo*
- *Errori nella stesura del programma*
- *Errori nei dati*
- *Errori attribuibili al sistema (...molto rari)*

Errori nei programmi

- *Errori sintattici*
- *Errori segnalati in fase di esecuzione* (a/0
invece che a/b)
- *Errori nei risultati*