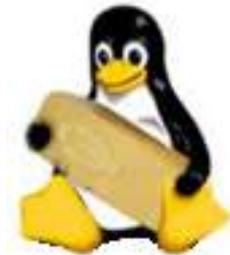


Strumenti integrati Free Software per lo sviluppo

GNU Emacs, GCC, GDB



SOftware **LI**bero **RA**gusa <http://www.solira.org>

Fasi dello sviluppo di un programma associate a diversi strumenti software:



→ **Editing del sorgente:** Editor, meglio se specifico per programmatori (GNU Emacs)



→ **Compilazione:** traduzione del codice sorgente in codice oggetto (GCC - Gnu Compiler Collection)



→ **Testing:** verifica rispondenza alle attese, correzione errori di *run-time* (GDB - Gnu DeBugger)

Meglio se gli strumenti sono integrati



Vantaggi dell'integrazione



→ Dialogo **compilatore–editor**: errore rilevato in fase di compilazione porta alla riga del sorgente. Facilità di modifica/correzione



→ Dialogo **debugger–editor**: programma monitorato con evidenziazione della riga del sorgente corrispondente all'istruzione in esecuzione

In definitiva: risparmio di tempo e praticità

GNU Emacs: l'editor tutto-fare



- ➔ **Ambiente di sviluppo globale:** editor con riconoscimento e colorazione strutture sintattiche, controllo avanzato editing (indentazione), aiuti per il programmatore. Non solo editor (piattaforma di sviluppo). Librerie per **integrazione** con altri software
- ➔ **Non fa niente per essere *simpatico*:** complessità di gestione mitigata in ambienti desktop (navigazione e uso della *Speedbar*), terminologia (window, frame, buffer) e interazioni diversa rispetto agli standard odierni: comandi nella *Echo Area (minibuffer)* richiamata dalla combinazione *Alt+x (M-x)*

Integrazione GCC - Emacs



→ Compilatore del progetto GNU: prima solo C ora supporto per diversi linguaggi (C, C++, Fortran, Java)



→ Il programma `prova.c` si compila con:

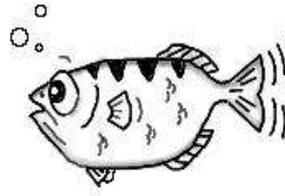
```
cc prova.c -o prova
```

dalla riga di comando della Echo Area (*Tools/Compile*)



→ L'output del compilatore in una *window* dell'ambiente: in presenza di errori possibilità di richiamare la riga del sorgente con l'errore anche se il sorgente non è caricato in un buffer (pulsante centrale del mouse o tasto *Invio*)

Integrazione GDB - Emacs



- Il programma `prova.c` si compila con:

```
cc prova.c -g -o prova
```

per includere informazioni per il debugging

- Emacs comprende librerie per il supporto ai debugger (GUD, Grand Unified Debugger), fra cui GDB
- Se una window di Emacs mostra anche il sorgente, se ne potrà osservare l'interazione con i comandi di controllo del debugger

Monitorare l'esecuzione di un programma

comandi di uso comune per le sessioni di debugging con GDB



<code>r(un)/kill</code>	Avviare/bloccare l'esecuzione del programma
<code>c(ontinue)/n(ext)/s(tep)</code>	Continuare l'esecuzione dopo un breakpoint
<code>b(reak)/delete</code>	Aggiungere/eliminare un breakpoint
<code>watch/p(rint)</code>	Settare un punto di osservazione/stampare una espressione
<code>info [break program proc]</code>	Informazioni su breakpoint/watchpoint, programma, ambiente
<code>help/quit</code>	Sistema di aiuto sui comandi/Termina sessione di debugging

GNU Make

velocizzare la gestione di progetti complessi



- ➔ *Makefile*: file-contenitore per specificare ***cosa*** ottenere, ***come*** ottenerlo, ***da cosa*** dipende il raggiungimento del singolo obiettivo
- ➔ Compilazione ***intelligente*** del Makefile: solo i sorgenti che risultano modificati (comparazione fra date e orari)
- ➔ Concetti fondamentali: obiettivo (***target***), prerequisiti (***prerequisites***), comandi (***command***)

<http://www.solira.org>

