

Debugging

- Se un programma compila non è detto che funzioni
- se un programma funziona non è detto che faccia quello che avete in mente;
- se un programma può essere eseguito, anche terminando prematuramente, si può usare un debugger;
- si possono monitorare le variabili e vedere il contenuto degli array;
- i più bravi possono guardare il contenuto dei registri;
- l'eseguibile deve contenere le informazioni sul sorgente;
- l'esecuzione deve poter procedere in modo da potersi fermare in corrispondenza ad ogni riga del sorgente;
- Per accedere al debugging si compila con

g++ -g -O0 program.c

GNU Debugger gdb

È il debugger a linea di comando standard per il software linux. È scomodo da usare, ma funziona.

- Dare il comando `gdb file_eseguibile`
- Comparirà il prompt del debugger;
- `[l]ist` visualizza il codice sorgente, anche come `list riga1,riga2`;
- `[r]un` fa andare il programma dall'inizio alla fine;
- `[br]eakpoint` ferma il programma a una linea oppure una funzione;
- `[s]tep` va alla prossima linea del programma entrando in ogni funzione;
- `[n]ext` fa lo stesso ma non entra nelle funzioni;
- `finish` esce dalla funzione dove siete;

- `cont` continua fino alla fine o fino al prossimo breakpoint;
- `[u]ntil` va alla linea del sorgente che sta più in basso di quella attuale (utile per uscire dai cicli)
- `[p]rint variabile` mostra il valore della variabile
- `[h]elp` e `[i]nfo` danno aiuto sui vari comandi

Profiling

- Serve a misurare la velocità di esecuzione delle singole funzioni del programma;
- compilare con `gcc -pg sorgente.c`
- il programma deve essere poi eseguito, quindi si da' il comando

`gprof eseguibile (| less)`

- si ottiene una statistica del tempo speso in ogni funzione, e il numero di volte che ogni funzione è stata chiamata;
- il profiler non corregge i programmi, serve solo a renderli più veloci;
- guardare in quali funzioni un programma passa più tempo;
- concentrarsi su quelle chiamate più volte;
- si ottimizza un programma solo se già funziona, per poter verificare i risultati.

IDE di programmazione

- Permette di editare, compilare ed eseguire in uno stesso ambiente premendo un tasto;
- consente di gestire grandi progetti con molti file
- accorcia il ciclo di sviluppo
- ne esistono diverse sotto linux: tra queste kdevelop e anjuta
- kdevelop è pensata per KDE
- anjuta è pensata per GNOME
- Geany è una shell semplice ma con le cose essenziali: evidenziazione della sintassi, compilazione, esecuzione dei programmi senza uscire dall'editor