

# Debugging

# Cosa è un debugger ?

- È un programma che permette di eseguire un altro programma P in ambiente protetto
  - In modo da poter fermare P durante l'esecuzione ed ispezionare:
    - Il contenuto delle variabili
    - Lo stack ed altro...
  - Vedremo alcune funzionalità di base...

# Fermarsi: break point

- Il debugger permette di fissare dei "punti di fermata" (break point) nel programma P
- Si può richiedere di eseguire
  - Fino ad un punto di fermata
  - Una istruzione alla volta entrando o meno nelle funzioni (**step**, **next**)

# Ispezionare le variabili

- Quando il programma è fermo si può ispezionare le variabili
  - Richiedere di mostrare il valore man mano che avanziamo (**display**)
  - Ispezionare il valore dello stack (backtracing)

# Compilare per il debugging

- Per fare tutto questo il compilatore deve inserire delle informazioni in più che permettono al programma debugger di "dialogare" con il programma P
  - Per interrompere la sua esecuzione
  - Per ispezionare i valori
- Si utilizza l'opzione **-g** del gcc

# ddd/gdb: un esempio

- Supponiamo di voler debuggare un programma in **file.c**
  - compiliamo

```
gcc -Wall -pedantic -g -o ese file.c
```

- Attiviamo il debugger (grafico o testuale) sull'eseguibile così ottenuto.

```
ddd ./ese
```

- testuale

```
gdb ./ese
```