

CS 35L Software Construction Laboratory (Lab5-B)  
Wed, Feb 8, 2012, Ver 1.0

A tutorial: <http://www.unknownroad.com/rtfm/gdbtut/gdbtoc.html>  
A quick demo: <http://irl.cs.ucla.edu/~jwcai/qsort.c>

Preparing your source code for debugging

- Add g command line argument when compile the source code using gcc/g++
  - With a -g option, the output file will be larger and runs slower
- e.g. gcc -g qsort.c -o qsort

When to use a debugger

- To figure out how a program is executed
- Reproduce some bugs
- Print program stack after crash

Loading a program

- Simplest situation: gdb name\_of\_executable  
e.g. gdb ls
- When the executable needs some parameters  
e.g. gdb gdb --args ls -lt
- In emacs: M-x gdb (require very reliable source code)

GDB command line

- break -- shortened as 'b'
- set breakpoint on file and line number  
b qsort.c:5
  - set breakpoint on function  
b qsort

info breakpoints  
-- display all breakpoints

- list -- shortened as 'l'
- print out the lines of code around the line the program is stopped at

e.g.

(gdb) info breakpoints

Num	Type	Disp	Enb	Address	What
1	breakpoint	keep	y	0x00000000100000cde	in main at qsort.c:6
	breakpoint already hit 1 time				
3	breakpoint	keep	y	0x00000000100000db2	in qsort at qsort.c:19
	breakpoint already hit 3 times				

(gdb) disable 1 -- Note: disable a breakpoint

(gdb) delete 3 -- Note: delete a breakpoint

r -- short for 'run'

n -- short for 'next', move on to the next statement

s -- short for 'step', step into one function call

n N -- make N movements

s N -- continue N steps

c -- short for 'continue', execute until next breakpoint

RET -- i.e. hit enter button (repeat next, step, or continue)

print var -- print the value of a variable (one time)

e.g.

```
(gdb) print pivot
$4 = 105 'i'
```

disp var -- display the value of a variable  
(every time the variable is in current scope)  
watch var -- a smart break point  
break the execution one the variable changes

e.g.

```
(gdb) watch e
Hardware watchpoint 5: e
(gdb) c
Hardware watchpoint 5: e
Old value = 3
New value = 7
qsort (str=0x7fff5fbff890 "adfhisst", l=6, r=7) at qsort.c:21
1: s = 6
```

backtrace

-- display a listing of function calls from the beginning of execution  
-- extremely useful for after-crash debugging

e.g.

```
(gdb) backtrace
#0  qsort (str=0x7fff5fbff890 "acelr", l=3, r=4) at qsort.c:24
#1  0x100000ec0 in qsort (str=0x7fff5fbff890 "acelr", l=0, r=4) at qsort.c:42
#2  0x100000d52 in main () at qsort.c:12
```