

Quick GDB Reference

compiling a program to run under GDB

- `gcc -g`
- Remember not to use with optimization (`-O` switches)

starting and stopping

<code>gdb programName</code>	opens gdb using the program <i>progName</i>
<code>gdb -tui programName</code>	TUI - “text user interface”. Still plain text, so it works over an SSH session, but allows for multiple windows (including your source). Only seems to work under recent versions of emacs. Can toggle TUI mode on/off with CTRL-X A
<i>under emacs:</i>	
M-x gdb	
<code>run</code>	runs the program (remember to “break main” if you want to step early on).
<code>run [arg1] [arg2] ...</code>	passing command-line arguments
<code>quit</code>	exits the debugger

the essentials

command	abbreviation	description
break	b	break, <i>i.e.</i> , stop executing at a particular line number or function. Current file is default, but can specify other.
break <i>[file:]line</i>		
break <i>[file:]func</i>		
next	n	execute next line of code
step	s	execute next line of code, but also execute called functions line by line. Think “step into”
continue	c	resume running until the next breakpoint
	c n	will continue running and skip the next n breakpoints.
until	u	use this to continue running until the end of a loop or function
print <i>expr</i>	p <i>expr</i>	shows the value of an expression <i>expr</i>
print/f <i>expr</i>		$f \in \{x, d, u, o, t, a, c, f\}$
display <i>[/f] expr</i>		like print, except that it continues to show the value as long as the variables in <i>expr</i> are in scope.
x <i>address</i>		examine memory
x/f <i>address</i>		
backtrace	bt	trace the entire stack (<i>i.e.</i> , how did we get to this function in the first place?

information about the local function

GDB’s info command can be very helpful when examining the local stack frame (In case you don’t know about it yet, it’s an area of memory local to your current function. We’ll discuss this stuff in depth this semester).

command	abbreviation	description
frame	f	information about the local frame
info frame	info f	detailed information about the frame
		- frame address
		- address frame of callee
		- address of frame of caller
		- address of args
		- address of locals
		- saved registers
info frame <i>addr</i>	info f <i>addr</i>	same as above, but information given about the frame at address <i>addr</i>
info args		information about the func arguments
info locals		information about the local variables