

`gdb` Cheatsheet

15-213 Fall 99

Starting and stopping `gdb`

```
gdb
gdb <file>
gdb -h           (lists command line options)

quit
Ctrl-d
Note: Ctrl-C does not exit from gdb, but halts the current
      gdb command
```

General commands

```
run           (start your program)
kill          (stop the program)
```

Breakpoints

```
break FUNCTION  (set a breakpoint at the entry to the function)
break *ADDRESS  (set a breakpoint at the specified address)

disable <NUM>   (disable the breakpoint with that number)
enable <NUM>    (enable the breakpoint with that number)

clear FUNCTION  (clear any breakpoints at the entry to the function)

delete <NUM>    (deletes the breakpoint with that number)
delete          (deletes all breakpoints)
```

Working at breakpoints

```
stepi          (execute one instruction)
stepi <NUM>     (execute NUM instructions)

nexti          (execute one instruction, stepping over functions)
nexti <NUM>     (execute NUM instructions, stepping over functions)

until LOCATION (continue running until LOCATION is reached)

continue       (resume execution)
continue <NUM> (continue, ignoring this breakpoint NUM times)

finish         (run until the current function returns)

backtrace      (print the current address and stack backtrace)
where          (print the current address and stack backtrace)
```

Examining code and data

<code>disas</code>	(display the function around the current line)
<code>disas ADDR</code>	(display the function around the address)
<code>disas ADDR1 ADDR2</code>	(display the function between the addresses)
<code>print/a \$pc</code>	(print the program counter)
<code>print \$sp</code>	(print the stack pointer)
<code>print \$eax</code>	(print the contents of %eax)
<code>print/x \$eax</code>	(print the contents of %eax as hex)
<code>print/a \$eax</code>	(print the contents of %eax as an address)
<code>print/d \$eax</code>	(print the contents of %eax as decimal)
<code>print/t \$eax</code>	(print the contents of %eax as binary)
<code>print/c \$eax</code>	(print the contents of %eax as a character)
<code>print 0x100</code>	(print decimal repr. of hex value)
<code>print/x 555</code>	(print hex repr. of decimal value)
<code>x ADDR</code>	(print the contents of ADDR in memory)
<code>x/NFU ADDR</code>	(print the contents at ADDR in memory: N = number of units to display F = display format U = b (bytes), h (2 bytes), w (4 bytes))
<code>x/10i ADDR</code>	(print the next 10 instructions)

Autodisplaying information

<code>display \$eax</code>	(print contents of %eax every time the program stops)
<code>display</code>	(print the auto-displayed items)
<code>delete display <NUM></code>	(stop displaying item NUM)

Useful information commands

<code>help</code>	
<code>info program</code>	(current status of the program)
<code>info functions</code>	(functions in program)
<code>info frame</code>	(information about the current stack frame)
<code>info variables</code>	(global and static variables)
<code>info registers</code>	(registers and their contents)
<code>info breakpoints</code>	(status of user-settable breakpoints)
<code>info address SYMBOL</code>	(use for looking up addresses of functions)

Running gdb in emacs

M-x gdb
C-h m to see the features of GDB mode