GDB Cheat Sheet

Basics

`gcc -g ...`
make an executable that can be debugged using GDB

`gdb progName`
start debugging *progName*

`gdb --args progName args`
start debugging *progName*, using command-line arguments *args*

`q`
quit GDB

`help command`
display information about *command*, incl. its syntax

`run`
start running program

`kill`
terminate currently running program

Breakpoints

`break point`
create a breakpoint at *point*

point can be a line number
eg. `break 5`

point can also be a function name
eg. `break main`

if the project contains multiple source files, break point must specify file
eg. `break foo.c:5`

`continue`
continue executing normally

`finish`
continue executing until current function returns

`step`
execute next line of source code

`next`
execute next line of source code, without descending into functions (treat a function call as a single line)

Examining Data

`print var`
show current value of variable *var*

`print/format var`
show current value of *var* in specified format

`print/x var` uses hexadecimal format (useful for examining addresses)

var can be a struct attribute
eg. `print node->key`

`display var`
automatically print value of *var* at each halt in execution

`undisplay var#`
stop displaying variable with display number *var#*

`watch var`
set a watchpoint on *var*: pause the program whenever value of *var* changes

`info args`
show value of all arguments to current function

`info locals`
show current value of all local variables

Examining the Stack

`backtrace`
display the current call stack (can be used after a runtime error, eg. segfault)

Modifying Breakpoints

`info breakpoints`
display information about all current breakpoints

`delete`
remove all breakpoints

`delete breakpoint#`
remove breakpoint with number *breakpoint#*

Gabrielle Singh Cadieux, 2016