Cheatography

GDB Cheat Sheet

by oddcoder via cheatography.com/33871/cs/10577/

Run GDB

gdb <program_path>	Load program into gdb
gdb <program_path> <core_path>	Load program and core dump into gdb

Breakpoints

break	Set break point at the current location
break if <condition>	Set break point here that triggers if certain condition is met
break <code_location>	Set break point at given code location
break <code_location> if <condition>	Set break point at given code location that triggers if given condition is met
hbreak	works exactly like break but it is hardware assisted breakpoints
info breakpoints	List all breakpoints and their associated num
clear	Delete all break points
delete <breakpoint_num>	Delete breakpoint given its number
enable <breakpoint_num>	Enable breakpoint given its number
disable <breakpoint_num>	Disable breakpoint given its number

code_location

function_name	self-explanatory
**function_name + offset*	move *offset* bytes from *function_name*
**math_expr*	pointer evaluated from the math expression

Stepping

run	Run the loaded program
run <arguments>	Run loaded program with given arguments
attach <pid>	Attach debugger to given process
next	Next line of source code
step	Same as *next* but will dive into calls
nexti	Next assembly instruction
stepi	same as *nexti* but will dive into calls
finish	Continue till first ret instruction
continue	Continue till next breakpoint

Examining code

backtrace	Print current backtrace
disassemble <function_name>	Disassemble given function

Memory

print/<format> <expression>	Evaluate expression and print it in given format
display/<format> <expression>	Same as *print* however it keeps executing after each **step** instruction
info display	List all auto-display expressions and their numbers
enable display <num>	Enable display given its number
disable display <num>	Disable display given its number
x/nuf <address>	Examine memory. n: How many units to print (default 1). f: Format character (like "print"). u: Unit. Unit is one of: b: Byte h: Half-word (two bytes) w: Word (four bytes) g: Giant word (eight bytes).

Format

a	Pointer
c	Character
d	Signed integer
f	Floating point number
i	instruction
o	octal
s	C-type strings
t	Binary
u	Unsigned integer
x	Hexadecimal

General information

info sharedlibrary	List loaded shared libraries
info proc mappings	list of mapped memory regions.