

Starting: gdb gdb <file>		Examine Code disas disas sum disas 0x80483b7 disas 0x80483b7 0x80483c7 print /x \$eip print /d \$eip print /t \$eip	Disassemble current function Disassemble function sum Disassemble function around 0x80483b7 Disassemble code within specified address range Print program counter in hex Print program counter in decimal Print program counter in binary
Run and stop: help quit run run 1 2 3 run < in.txt kill Control-D Control-C make	Get information about gdb Exit gdb Run program Run program with command-line arguments 1 2 3 Run program with input redirected from in.txt Stop the program Exit gdb Stop the currently running gdb command (Does not exit GDB.) Run make to rebuild without leaving gdb	Examine Data print /d \$eax print /x \$eax print /t \$eax print 0x100 print /x 555 print /x (\$esp+8) print *(int *) 0xbffff890 print *(int *) (\$esp+8) print (char *) 0xbffff890	Print contents of %eax in decimal Print contents of %eax in hex Print contents of %eax in binary Print decimal representation of 0x100 Print hex representation of 555 Print (contents of %esp) + 8 in hex Print integer at address 0xbffff890 Print integer at address %esp + 8 Print string at address 0xbffff890
Breakpoints break sum break *0x80483c3 delete 1 disable 1 enable 1 delete clear sum	Set breakpoint at the entry to function sum Set breakpoint at address 0x80483c3 (gdb numbers each breakpoint you create) Delete breakpoint 1 Disable breakpoint 1 Enable breakpoint 1 Delete all breakpoints Clear breakpoints at entry to function sum	x/w 0xbffff890 x/w \$esp x/wd \$esp x/2w \$esp x/2wd \$esp x/g \$esp x/gd \$esp x/a \$esp x/s 0xbffff890 x/20b sum x/10i sum display /FMT EXPR	Examine 4-byte word starting at address 0xbffff890 Examine 4-byte word starting at address in \$esp Examine 4-byte word starting at address in \$esp in decimal Examine two 4-byte words starting at address in \$esp Examine two 4-byte words starting at address in \$esp in decimal Examine 8-byte word starting at address in \$esp. Examine 8-byte word starting at address in \$esp in decimal Examine address in \$esp as offset from previous global symbol Examine string stored at 0xbffff890 Examine first 20 opcode bytes of function sum Examine first 10 instructions of function sum Print expression EXPR using format FMT each time execution stops
Execution stepi stepi 4 nexti step continue until 3 finish call sum(1, 2)	Execute one instruction Execute four instructions Execute one instruction treating entire function call as one instruction Execute one C statement Execute until next breakpoint Execute until breakpoint 3 Execute until current function returns Call sum(1,2) and print return value	Formats: x/[NUM][SIZE][FORMAT] If not given, uses sensible default or last-used explicit format NUM = number of objects to display SIZE = size of each object b = byte h = half-word w = word g = giant (quad-word) FORMAT = format for displaying each object d = decimal x = hex o = octal	
Context backtrace where info program info functions info stack info frame info registers info breakpoints	Print the current address and stack backtrace Print the current address and stack backtrace Print current status of the program) Print functions in program Print backtrace of the stack) Print information about current stack frame Print registers and their contents Print status of user-settable breakpoints		