GDB Survival Guide

1. **Help**
    a. **help**
        i. Drill in for more specifi help by specifying which command you want help on
            1. **help disassemble**
2. **Disassembly**
    a. **disassemble** <address>
        i. <address> may be absolute, relative, or symbolic
            1. disassem main
            2. disassem $eip
            3. disassem 0x8040100
        ii. disassemble will look for a function containing the address you specify and disassemble that function
        iii. **NOTE**: any gdb command can be shortened to just the number of characters required to uniquely identify that command from all other gdb commands.
3. **Program execution**
    a. **run** – runs the program
    b. **break** or just **b** – set a breakpoint at a specified address
        i. **b main**
        ii. **b *0x8040100**
            1. **NOTE** the use of the * preceding the address
    c. **step** – step one source line stepping into function calls
    d. **stepi** – step one assembly instruction stepping into function calls
    e. **next** – step one source line stepping over function calls
    f. **nexti** – step one assembly instruction stepping over function calls
        i. **NOTE** step, stepi, next, nexti can all take an optional count argument to indicate how many instructions to step through
    g. **cont** – continue execution after a breakpoint or other stop has been encountered
        i. **NOTE** cont can also take a count parameter to specify how man breakpoints should be skipped before stopping again
4. **Data display** there are several ways to view data in gdb
    a. **Full register dump**
        i. **info reg**
    b. **Individual register display**
        i. **print** $<reg name>
            1. **print $eax**

ii. **display** $<reg name>
    1. **display $eax**
        a. print value of eax each time the program stops
    2. **undisplay <expr>**
        a. Stop display expr each time program stops

c. **Memory dump**
    i. **x** /<fmt> <addr>
        1. <fmt> is a format and size specifier
            a. <fmt> can contain a repeat count, a format letter, and a size letter. Repeat count defaults to 1.
            b. Format letters include
                i. x – hex
                ii. o – octal
                iii. t – binary
                iv. f – float
                v. d – decimal
                vi. u – unsigned decimal
                vii. c – ascii characters
                viii. s – null terminated string
                ix. i – instruction
                x. a - address
            c. Size letters include
                i. b – byte
                ii. h – 2 byte short
                iii. w – 4 byte words
        2. <addr> is the memory address at which to start the dump
        3. **Examples**
            a. **x /32xw $esp**
                i. Display 32 4 byte words beginning at the address held in esp
                b. **x /32i $eip**
                    i. Display 32 instructions beginning at the address held in eip.
                    ii. **NOTE:** No attempt is made to find the function in which the address lies. This is a good way to disassemble code that may lie in a data region (such as shellcode)
                c. **x /128c 0x804932**
                    i. Display 128 characters starting at address 0x804932
    ii. **print** and **display** also work for memory locations and accept format letters but not size letters.