WinDbg Cheat Sheet

I. General

A. bp $exentry - Breaks at executable entry point

B. @ - Dereference memory

C. !address - List and describe all memory pages

D. ln [Address] - Lists the closest symbol for a given memory address

II. Registers

A. r [Register name | Flag Name] : format [= Expression_Or_Value]

1. Used to display register values or change them
2. Examples:
    a) r eax - Read eax
    b) r eax = 2 - Set eax to 2
    c) r eax:f - Read eax as floating point
    d) r eax:ub/uw/ud/uq - Read eax as unsigned byte/word/dw/quadword
        (1) Use i instead of u for signed format

B. rMx [Register name | Flag Name] : format [= Expression_Or_Value]

1. r with Mask, where x is the 32 bit mask
2. This is used to view more than the general registers
    a) 0x2 - General Registers
    b) 0x4 - Floating-point Registers
    c) 0x8 - Segment Registers
    d) 0x10 - MMX
    e) 0x20 - Debug Registers
    f) 0x40 - SSE XMM
    g) 0x80 - Kernel Mode: Control Registers
    h) 0x100 - Kernel Mode: TSS
    i) 0x1FF - All possible registers
3. Example:
    a) rM1ff - Read all possible registers
4. Note that some registers can only be displayed in kernel mode debugging

C. rmx - Set default mask

III. Process Control

A. Execution
   1. g - go, resume execution
   2. gu - go up, executes current function until return to caller

B. Stepping into / over
   1. [t | p] - Step into/over
   2. [ta | pa] (address) - Step into/over until address is reached
   3. [tc | pc] - Step into/over until a call is encountered.
   4. [th | ph] - Step into/over until any branch.
   5. [tt | pt] - Step into/over until a ret instruction.
   6. [tct | pct] - Step into/over until a call or ret.

C. Breakpoints
   1. bp [address/symbol] - Sets a breakpoint on address/symbol,
   2. bl - List break points
   3. bc (number) - Clear breakpoint
   4. be (number) - Enable breakpoint
   5. bd (number) - Disable breakpoint

D. Threads
   1. ~ - List all threads
   2. ~Ns - Switch threads, where N is the thread number

IV. Reading and Writing Memory

A. Display Memory: d [a | b | c | d | D | f | p | q | t | u | w | W] (address)
   1. b, w, d, q = byte, word, double-word, quad word.
   2. f, D - Single and double precision floating point.
   3. a, u - Ascii or unicode
   4. p - Pointer value
   5. s - Symbols corresponding to the address will be displayed
   6. a, u - Ascii or unicode, za and zu add a null terminator automatically
   7. t [type] [address] - display type of item at a given address
      a) Ex. dt nt!_DRIVER_OBJECT
      b) Ex. dt nt!_DRIVER_OBJECT 828b2648 - Overlay data onto the structure

B. Edit Memory: e [b | d | D | f | p | q | w] (address) [value]
    1. b, w, d, q - byte, word, dword, or qword
    2. f, d = Set single or double precision floating point number
    3. p - Set pointer-sized values there

V. Kernel Mode Debugging General

A. lm
    1. In kernel mode, this command displays the list of loaded device drivers
B. !process 0 0
    1. List all running processes
    2. Can provide a process name to search for as an optional third argument.
C. !drvobj [driver_name]
    1. Finds the driver object, gives address of driver in kernel-space
D. !devobj [device_name]
    1. Gets device object information by using the name of the device specified
E. !devhandles [Address]
    1. Obtains a list of all user-space applications that have a handle to the device at the given address.
F. !idt
    1. Shows the Interrupt Descriptor Table