Chapter 3: Computer Memory

3.1 Memory Mapping.

- computer memory can be thought of as an array of bytes.
- In modern CPU's there are hardware mapping registers used to give each process a protected address space.
    - The logical address can be the same but the physical address differs.
- Memory Pages
    - Hardware mapping registers on an x86_64 CPU can map pages of 2 different sizes.
        - 2 MB for the kernel on Linux, OS X, and Windows.
            - sometimes user processes are allowed 2 MB page.
        - 4096 bytes for most other uses.
        - Modern CPU's may allow 1 GB pages.
    - The memory system translates the upper bits of the address from a process's logical address to a physical address.
    - Example using 4KB pages.
        - An address is translated based on the page number and the address within the page.
        - Consider the logical address 0x4000002220.
            - 4096 = 2^12 so the offset within the page is the rightmost 12 bits. 0x220
            - The page number is the rest of the bits 0x4000002
            - The hardware translates the page number to a physical page address.
            - Then the offset is tagged onto the end of the address.
    - This adds memory protection so that processes are not using memory from other processes pages.

    - Users are also prohibited from reading other users data.

3.2 Process Memory Model in Linux

- In Linux the memory for a process is divided into four ligical regions.
    - text
    - data
    - heap
    - stack
- The stack is mapped to the highest address of a process.
    - Linux x86_64 has this as 0x7fffffffffff or 131 TB.
    - The number is selected based on the max number of bits allowed in logical addresses being 48 bits.
- Arrangement of the program memory.
    - The text segment goes at the lowest memory address.
    - The data segment is placed directly above the text segment.
        - Data starts with the .data segment.
            - Contains initialized data.
        - Then the .bss segment.
            - contains data whihc is statically allocated in a process.
            - this data is not stored in an executable file.
                - allocated when the process is loaded into memory.
            - initially .bss segment are all 0 bits.
    - Then the heap
        - dynamically allocated memory at run time.
        - can grow very large.
            - limited by physical memory and swap space on x86_64
    - The Stack Segment on top of that.
        - Typically restricted to 16 MB by the Linux kernel

        - can be edited by changing /etc/security/limits.conf
    - The stack automotacially grows when the system responds to a page fault.
- executing /cat/proc/999/maps where 999 is the pid will display the memory used by a process.