Chapter 13 Address Spaces

13.1 Early Systems

- Early system memory didn't provid much of an abstraction to users.
    - The OS was a set of routines sitting in memory.
    - There would be one running program sitting in physical memory.

13.2 Multiprogramming and Time Sharing

- Multiprogramming is where multiple programs are ready to run at a given time.
    - The OS would switch between them.
    - Increased the effective utilization of the CPU
    - Increased efficiency.
- Time Sharing:
    - One way to time share is to run one process for a short while giving full resources, then stop it, save all state to disk, load another processes state, and run the new process.
        - This method is slow.
        - Saving the entire contents of memory to disk is slow and has performance penalties.
    - We would rather leave the process in memory.

13.3 Address Space

- The address space is an easy to use abstraction of physical memory.
- It is the running programs view of memory in the system.

- The address space of a program contains all of the memory state of the running program.
    - the code of the program is in memory somewhere.
    - The stack is there.
    - The heap is also there.

13.4 Goals

- Goals to guide the OS virtualizing memory.
- Transparency:
    - The OS should implement virtual memory in a way that is invisible to a running program.
    - The program shouldn't know the memory is virtual.
- Efficiency:
    - The OS should make virtual memory as efficient as possible.
        - In terms of time.
        - In terms of space.
- Protection:
    - The OS should make sure to protect processes from one another as well as protect the OS from processes.
    - The process should not be able to access memory outside the allocated address space.
    - Isolation among processes.