CS2106 Cheatsheet by randomwish

https://github.com/randomwish/schoolNotes

Operating Systems

Is a program that acts as an **intermediary** between a computer user and computer hardware

Types of Operating Systems

- No OS → direct interaction between program and hardware; any changes has to be done to the hardware (+ → minimal overhead; - > not portable and inefficient)
- Batch OS → executes user programs one at a time (loads job from media → executes job → collect results) (- → CPU idle when performing I/O; possible improvement is multiprogramming)
- Time-Sharing OS → allows multiple users to interact with machine using terminals; OS manages sharing of CPU time and storage; **virtualization** of hardware is involved (each program executes as if it has all resources to itself)

Purpose for OS

- Abstraction → hide low level details, perform tasks through OS such that user can perform common high level functionality
- Resource allocator → manages resources available and arbitrate conflicting requests
- Control Program → controls execution of programs to prevent errors and provide security

Structure of an OS

OS (also known as the kernel) would run in **kernel mode** → complete access to hardware resources. Kernel code **cannot** use syetm calls, cannot use normal libraries, and no normal I/O. Other software would then run in **user mode**, where it has limited access to hardware resources. Invocation of OS:

- Hardware ↔ OS: OS executing machine instructions
- OS ↔ User Programs and OS ↔ Library: OS called using system call interface

Ways to structure an OS:

- Monolithic → kernel is one big special program, where services are integral (Advantages: well understood and performance; disadvantages: coupled components and possible complications)
- Microkernel → kernel is small and clean, only provides basic and essential features (Inter Process Communication, Address Space management and thread management); higher level services are run as server process outside of OS → uses IPC to communicate (Advantages: kernel is robust, extensible, and better isolation and protection between kernel and high level; disadvantages: lower performance
- Layered System → seen as a generalization of monolithic system, where components are organised in hierarchy of layers
- client server model → variation of microkernel; client process request service from server process and server process is built on top of microkernel

Virtual machines

(also known as a **hypervisor**): seen as a software emulation of hardware, where it provides illusion of complete hardware Types of hypervisors:

- Type 1 Hypervisor → provides individual virtual machines to guest OSes (idea is that it provides direct linkage to the hardware, hardware ↔ hypervisor ↔ guest OS)
- Type 2 Hypervisor → runs in host OS, and the guest OS would then be run in the virtual machine (idea is that it is the intermediary between guest OS and the host operating system, hardware ↔ host operating system ↔ type 2 hypervisor)