5118020-03 Operating Systems

Introduction to Operating Systems

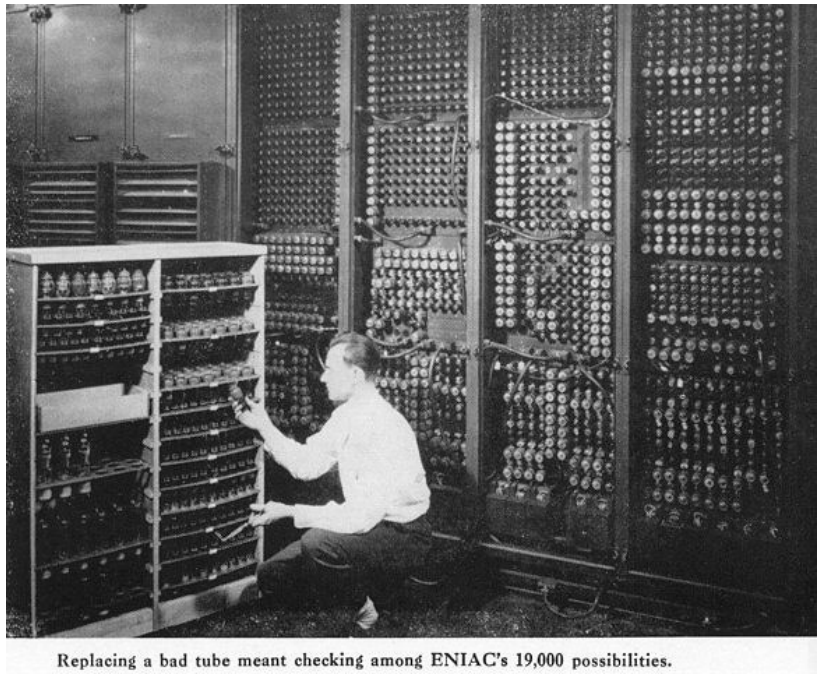
(OSTEP:Ch.2)

Shin Hong

What's Operating System?

- a program first loaded when a computer starts
- a suite of programs to a computer system to run multiple programs and manage hardware devices efficiently
  - supports construction and executions of application programs
  - works as a platform for application programs

Motivation - Demands



Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

- importing programs from other systems
- constructing a new system by combining existing programs
- managing a diversity and variety of hardware devices
- providing interactive features in a software system (e.g., networking)
- storing information persistently

Motivation - Problems

- portability issue (hardware-dependency)
- interoperatability issue
- scheduling issues
- resource management issue (efficiency and scalability)
- safety & security issues

Approach

- Virtualize a computer system
  - provide a consistent and simple view to applications and programmers (for portability)
  - provide common interfaces for an application program to communicate with hardware units and other application programs (for interoperability)
  - distribute hardware resources to efficiently serve requests from application programs (for concurrency, efficiency and scalability)
  - forbid an application to access the critical control and the internals of a computer system and the other programs (for safety and security)
- Provide different policies of coordinating application programs, such that a suitable one will be used at a specific context

Solution: Kernel

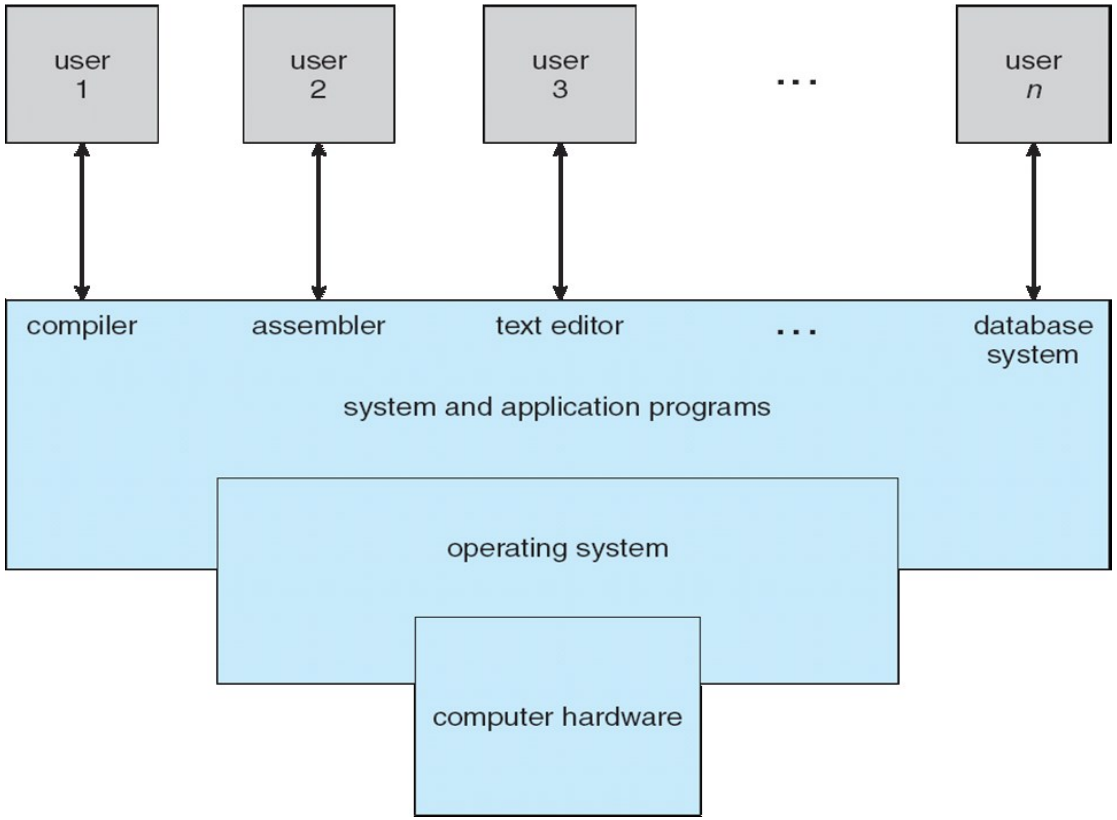
- Use a library program that provides “helper” functions with common interfaces to application programs
  - for controlling HW devices, and for communicating with other programs

- Let these helper functions exclusively process sensitive operations under protection
  - It’s more like “officer” than just “helper”
  - extends a computer architecture to provide a special instruction to call these officer functions (requires HW support)

- When they work, officer functions not only response to given requests, but also manage the resources that the application uses, especially in consideration with other programs in the system
  - Then it’s more like “governor” than “officer”

Solution: System Programs

- To make a human user and an application easily interact each other, there exists parts of operating systems running upon kernel called *system programs*
  - e.g., compiler, linker, loader, shell, service daemons



Solution: Abstraction

- Three models
  - Process
    - an abstract object representing a running instance of a program and all resources that it uses
  - Virtual memory
    - an abstraction of memory locations
  - File system
    - an abstract object for communication channel (streams)
      - to storage devices (i.e., perment memory)
      - to other programs
      - to other systems via network

Program Examples

- multi-processing
- multi-threading
- virtual memory
- file system operation

Quick History of Operating System