

The first chapter introduces the general topics of operating systems and a few important concepts such as multiprogramming, time sharing, clustered system, cloud computing and so on. It also illustrates why operating systems are what they are by showing how they developed.

The main purposes or objectives of an operating system:

- **User convenience:** to provide an environment for users or programmers to execute programs on computer hardware in a convenient, safe, protected and efficient manner.
- **Resource allocation:** to allocate computer resources in a fair and efficient manner. The resources can be hardware such as CPU and main memory, or software such as signal and lock. You can also view an OS as a control program, which controls program execution, prevent users from misusing the resources and handling I/O.

Interrupt, Trap and DMA

- An **interrupt** is a hardware-generated change-of-flow within the system. An *interrupt handler* (a specific program also referred as *interrupt service routine*, part of OS or kernel code) is called to handle an interrupt when an interrupt occurs; this involves *context switch*. An interrupt can be used to signal the completion of an I/O operation. All modern operating systems are **interrupt-driven**.
- A **trap** is a software-generated interrupt. A trap can be used to call operating system routines or to catch arithmetic errors.
- **DMA** or direct memory access is used for moving large amounts of data between I/O devices and main memory. It is considered efficient because it relieves the CPU from data transfer. CPU initiates a DMA controller, which instructs a device controller to move data between the devices and main memory.

The Dual Mode

- The distinction between **kernel mode** and **user mode** provides a rudimentary form of *protection* in the following manner. Certain instructions (*privileged*) could only be executed when the CPU is executing in kernel mode. Similarly, hardware devices can be accessed only when the program is executing in kernel mode. Thus, a program has limited capability when executing in user mode, thereby enforcing protection of critical resources. Users rely on the services provided by operating system to access such resources. For example, the following operations are privileged and can only be executed in the kernel mode: to set value of timer, clear memory, disable interrupts, modify entries in device-status table, access I/O devices.

Symmetric multiprocessing vs. Asymmetric multiprocessing

- Symmetric multiprocessing treats all processors as equals, and I/O can be processed by any CPU.
- Asymmetric multiprocessing has one master CPU and the remainder CPUs are slaves. The master distributes tasks among the slaves, and I/O is usually done by the master only.
- Multiprocessors can save the overall cost by not duplicating power supplies, housings, and peripherals. They can execute programs more quickly and can have increased reliability. They are also more complex in both hardware and software than uniprocessor systems.

Protection vs. Security

- Protection is concerned with controlling the user or process access of the resources of the computer system. The role of security is to defend the system from internal or external attacks. There are wide range of security threats such as denial-of-service, worms, viruses, and etc.

Cloud Computing

- Cloud computing is a type of computing that delivers computing, storage, or/and application services over a network. Cloud computing often uses **virtualization** to provide its functionality. There are many different types of cloud environments, as well as a variety of services offered. Cloud computing may be either public, private, or a hybrid of the two. Additionally, cloud computing may offer applications, platforms, or system infrastructures.

Open Source OS

- Open source operating systems have the advantages of having many people working on them, debugging them, ease of access and distribution, and rapid update cycles. Further, there is also certainly an advantage to being able to view and modify the source code. Typically open source operating systems are free for some forms of use, usually just requiring payment for support services.
- Commercial operating system companies usually do not like the competition that open source operating systems bring because these features are difficult to compete against. Some open source operating systems do not offer paid support programs. Some companies avoid open source projects because they need paid support, so that they have some entity to hold accountable if there is a problem or they need help fixing an issue.