The first chapter introduces the general topics of operating systems and a few important concepts such as multiprogramming, time-sharing, multicore, 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 are:

- **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 operating system as a control program, which controls program execution, prevent users from misusing 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 a *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, which 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 actual data transfer. CPU initiates a DMA controller, which instructs a device controller to move data between the devices and main memory; upon completion, it generates an interrupt to CPU.

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 the 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.

Protection and **Security**

- Protection measures the control of the access of processes or users to the resources made available by the computer system. Security measures are responsible for defending a computer system from internal or external attacks. There are wide range of security threats such as denial-of-service, worms, viruses, identity theft.

Multiprocessor Systems

- Multiprocessors systems have recently begun to dominate the landscape of computing, also known as **parallel systems**. Their major advantages are increased throughput and reliability, and cost efficient.
- **Symmetric multiprocessing** treats all processors as equals, and I/O can be processed by any CPU.
- **Asymmetric multiprocessing** has one master CPU and the other slave CPUs. 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 have increased reliability. They are also more complex in both hardware and software than single processor systems.
- A recent trend in CPU design is to include multiple computing cores on a single chip. Such multiprocessor systems are termed **multicore**. They are more efficient than multiple chips each with single cores because on-chip communication is faster than between-chip communication. Also one chip with multiple cores uses significantly less power than multiple single-core chips. Apparently, not all multiprocessor systems are multicores.

Cluster Systems

- Clustered systems differ from multiprocessor systems (**tightly coupled**) in that they are composed of two or more individual systems or nodes joined together. Such systems are considered **loosely coupled**, and each node may be a single processor system or a multicore system.
- Clustering is usually used to be provide high-availability service – that is, service will continue even if one or more systems in the cluster fail.

Client-Server and Peer-to-Peer (P2P)

- There are two distinctive features in a *Client-Server model*: (1) only clients can initiate the communications, i.e., requesting services from a server. For instance, a Web browser requests a Web page from a Web server. (2) Servers are always available, and waiting for clients to contact.
- In a *Peer-to-Peer paradigm*, there is no client or server, each entity serves the same role by joining and leaving at any time. Each peer can request certain services from

other peers while also providing services for other peers at the same time. For instance, in a *Bittorrent* application, each peer can retrieve certain chunks of a file from different peers while sending chunks (different) of a file to other peers.

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** technique 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. GNU/Linux and BSD UNIX are open-source operating systems. .