BCA Fourth Semester

Operating Systems

Unit -1 Introduction to Operating System [2 Hrs]

Operating System Introduction

- Computer Software can roughly be divided into two types:

  a) Application Software: Which perform the actual work the user wants.

  b) System Software: Which manage the operation of the computer itself.

- The most fundamental system program is the operating system, whose job is to control all the computer's resources and provide a base upon which the application program can be written.
- Operating system acts as an intermediary between a user of a computer and the computer hardware.
- **An operating system is similar to a government.** Like a government it performs no useful function by itself. It simply provides an environment within which other programs can do useful work.
- A computer system can be divided roughly into four components: ***the hardware, the operating system, the application program, and the users*** as shown in the figure:

Operating System Introduction

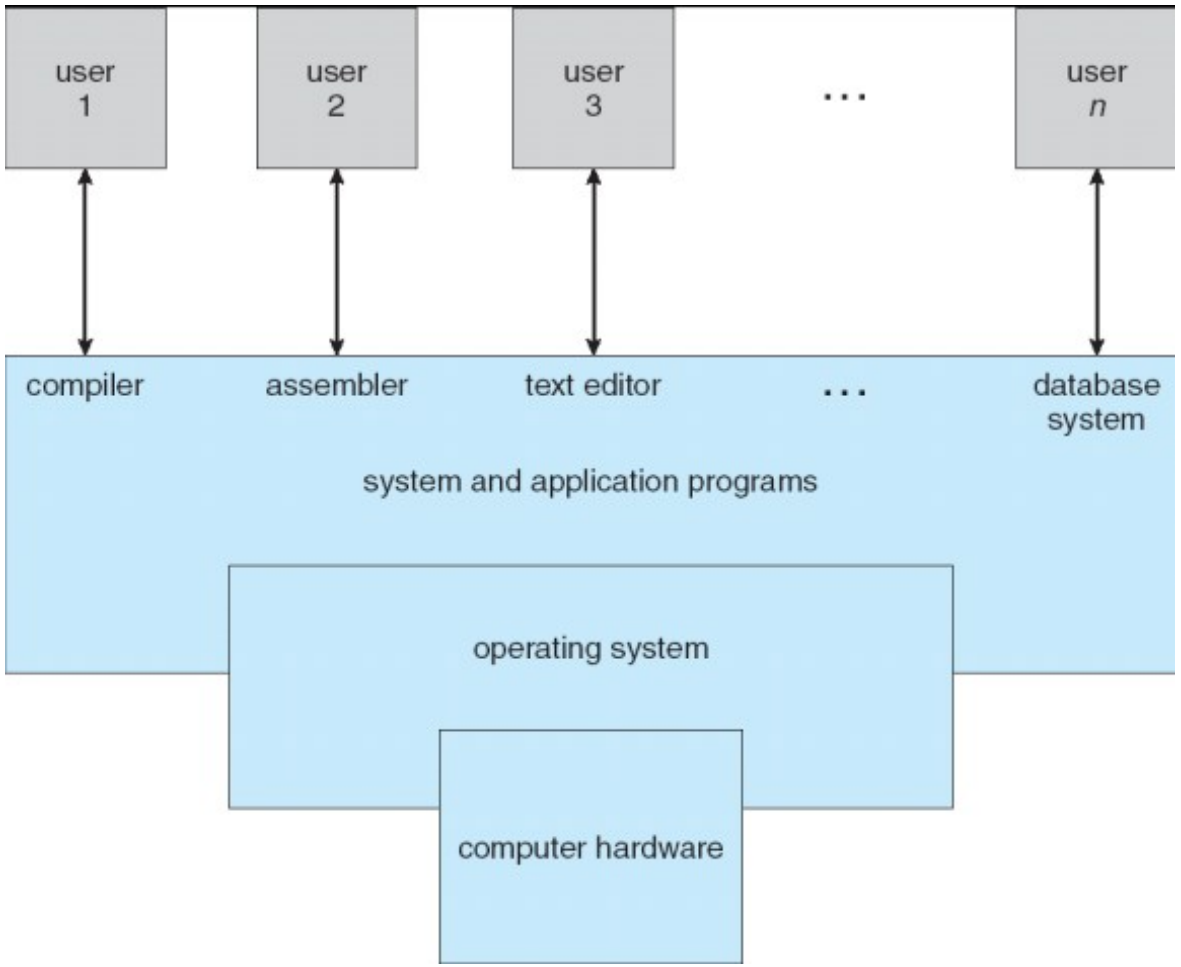


Figure: Abstract view of the components of a Computer System.

(Objectives) Two Views of Operating System

1) Operating System as an Extended Machine or Virtual Machine(or As a User/computer interface)

- The operating system masks or hides the details of the Hardware form the programmers and general users and provides a convenient interface for using the system.
- The program that hides the truth about the hardware from the user and presents a nice simple view of named files that can be read and written is of course the operating system.
- In this view the function of OS is to present the user with the equivalent of an extended machine or virtual machine that is easier to program than underlying hardware.
- Just as the operating system shields the user from the disk hardware and presents a simple file-oriented interface, it also conceals a lot of unpleasant business concerning interrupts, timers, memory management and other low level features.

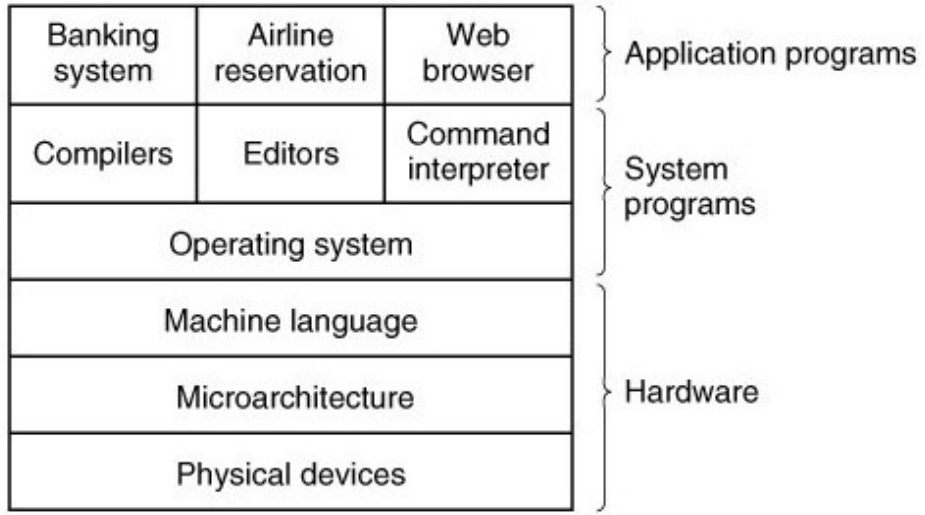


Figure: Computer system consists of Hardware, system program and application program

Two Views of Operating System

2) Operating System as a Resource Manager

- A computer system has many resources. Modern computers consist of processors, memories, timers, disks, mice, network interfaces, printers, and a wide variety of other devices. **In the alternative view, the job of the operating system is to provide for an orderly and controlled allocation of the processors, memories, and I/O devices among the various programs competing for them.**
- Imagine what would happen if three programs running on some computer all tried to print their output simultaneously on the same printer. The first few lines of printout might be from program 1, the next few from program 2, then some from program 3, and so forth. The result would be chaos. The operating system can bring order to the potential chaos by buffering all the output destined for the printer on the disk. **When one program is finished, the operating system can then copy its output from the disk file where it has been stored to the printer, while at the same time the other program can continue generating more output, oblivious to the fact that the output is not really going to the printer (yet).**

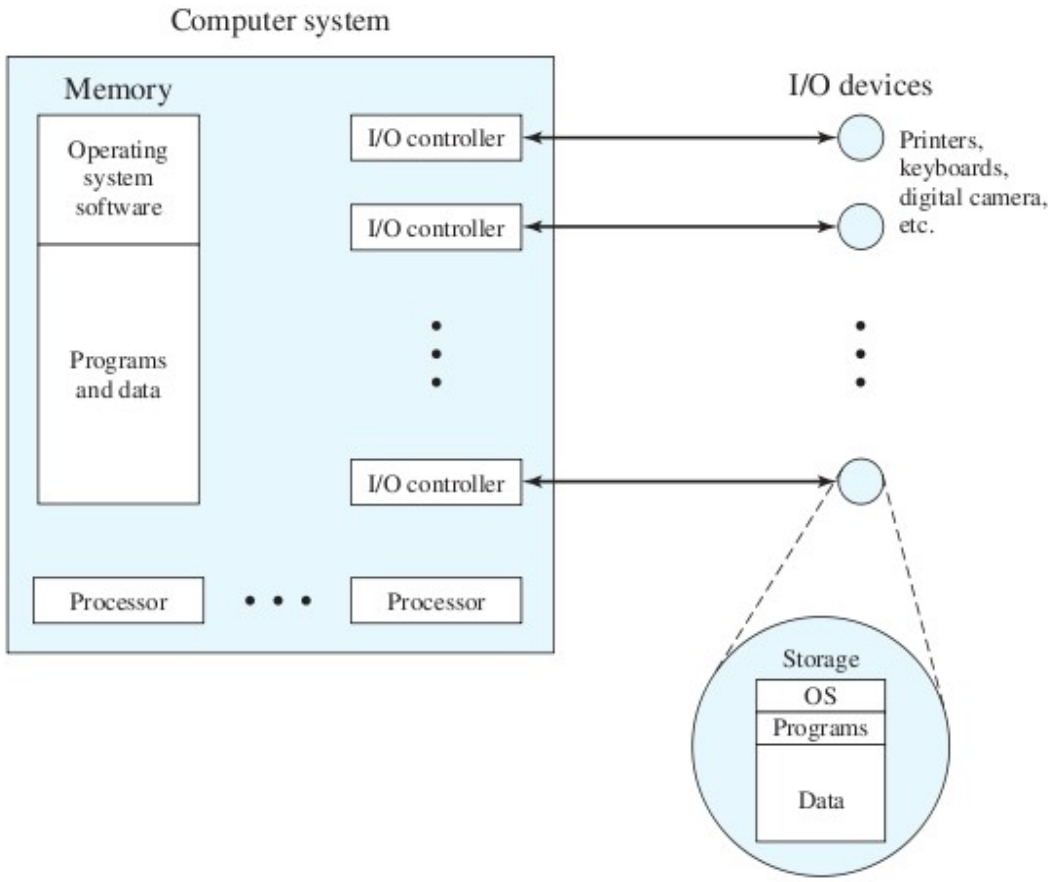


Fig: The Operating system as Resource manger

History and Evolution of Operating System

- **The first computers did not have operating systems.** Each program that was running on these first computers had to include all the code needed to run on the computer, communicate with the connected hardware and perform the calculation that the program was intended to perform. This situation made even the simplest programs become very complex.
- In response to this problem, the owners of the central computers began to develop system software that facilitated the writing and execution of the programs included in the computer, and thus the first operating systems were born.
- **The first operating system was created by General Motors in 1956 to run a single IBM central computer.** In the 1960s, IBM was the first computer manufacturer to take on the task of developing operating systems and began distributing operating systems included in its computers.
- **The first operating systems were developed in the 1950s, when computers could only run one program at a time.** Later in the following decades, computers began to include more and more software programs, sometimes called libraries, that came together to create the start of today's operating systems.
- **In the late 1960s, the first version of the Unix operating system was developed**. Written in programming language C, and available for free during its early years. Unix easily adapted to the new systems and quickly achieved wide acceptance.

History and Evolution of Operating System

- Many modern operating systems, including Apple OS X and all different versions of Linux, date back or rely on the Unix OS.
- Microsoft Windows was developed in response to an IBM request for an operating system to run its range of personal computers or PCs.
- **The first operating system created by Microsoft was not called Windows, it was called MS-DOS and it was built in 1981 when it bought the 86-DOS operating system from Seattle Computer Products and modified it to meet IBM requirements.**

- **The Windows name was first used in 1985 when a graphical user interface was created and paired or joined with the MS-DOS.**
- Today Apple, OS X, Microsoft Windows and the various forms of Linux (including Android) dominate the vast majority of the modern operating systems market, as we saw earlier.

Generation of Operating System

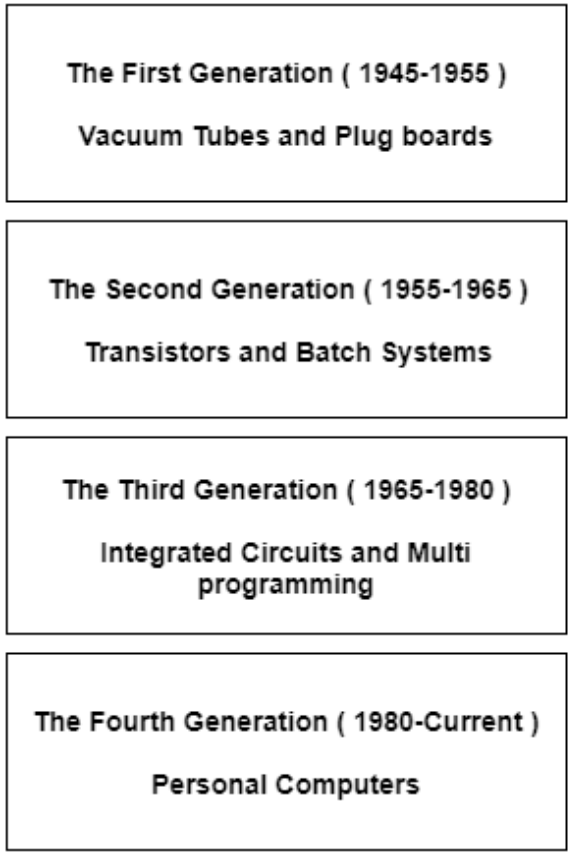
- Operating Systems have evolved over the years. So, their evolution through the years can be mapped using generations of operating systems. **There are four generations of operating systems. These can be described as follows –**

The First Generation (1945 - 1955): Vacuum Tubes and Plugboards

- Digital computers were not constructed until the second world war. Calculating engines with mechanical relays were built at that time. However, the mechanical relays were very slow and were later replaced with vacuum tubes. These machines were enormous but were still very slow.

- These early computers were designed, built and maintained by a single group of people. Programming languages were unknown and there were no operating systems so all the programming was done in machine language. All the problems were simple numerical calculations.

- By the 1950's punch cards were introduced and this improved the computer system. Instead of using plugboards, programs were written on cards and read into the system.



OPERATING SYSTEM GENERATIONS

Generation of Operating System

The Second Generation (1955 - 1965): Transistors and Batch Systems

- Transistors led to the development of the computer systems that could be manufactured and sold to paying customers. These machines were known as mainframes and were locked in air-conditioned computer rooms with staff to operate them.
- The Batch System was introduced to reduce the wasted time in the computer. A tray full of jobs was collected in the input room and read into the magnetic tape. After that, the tape was rewound and mounted on a tape drive.
- Then the batch operating system was loaded in which read the first job from the tape and ran it. The output was written on the second tape. After the whole batch was done, the input and output tapes were removed and the output tape was printed.

The Third Generation (1965 - 1980): Integrated Circuits and Multiprogramming

- Until the 1960's, there were two types of computer systems i.e the scientific and the commercial computers. These were combined by IBM in the System/360. This used integrated circuits and provided a major price and performance advantage over the second generation systems.
- The third generation operating systems also introduced multiprogramming. This meant that the processor was not idle while a job was completing its I/O operation. Another job was scheduled on the processor so that its time would not be wasted.

Generation of Operating System

The Fourth Generation (1980 - Present): Personal Computers

- Personal Computers were easy to create with the development of large-scale integrated circuits. These were chips containing thousands of transistors on a square centimeter of silicon. Because of these, microcomputers were much cheaper than minicomputers and that made it possible for a single individual to own one of them.

- The advent of personal computers also led to the growth of networks. This created network operating systems and distributed operating systems. The users were aware of a network while using a network operating system and could log in to remote machines and copy files from one machine to another.

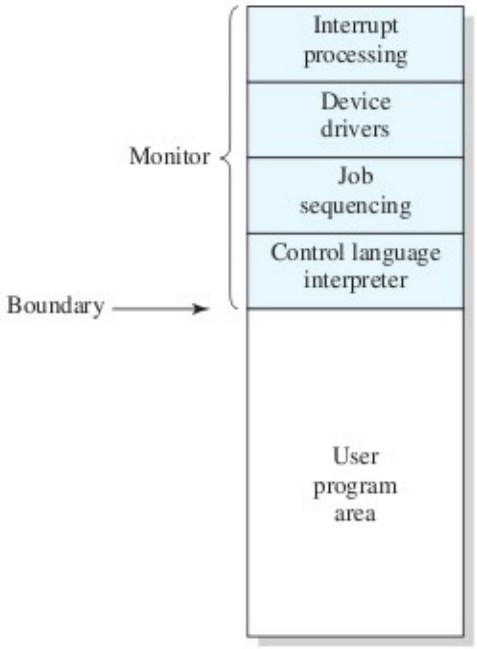
Types of OS

- Following are the types of OS:
  1. Batch processing OS
  2. Multiprogramming OS
  3. Multitasking or time sharing System
  4. Network Operating system
  5. Distributed Operating system
  6. Multiprocessor Operating System
  7. Real Time Operating System

Batch Processing OS:

- Early computers were very expensive, and therefore it was important to maximize processor utilization.
- The wasted time due to scheduling and setup time in Serial Processing was unacceptable.
- To improve utilization, the concept of a batch operating system was developed.

- **Batch is defined as a group of jobs with similar needs**. The operating system allows users to form batches. **Computer executes each batch sequentially, processing all jobs of a batch considering them as a single process called batch processing**.
- **The central idea behind the simple batch-processing scheme is the use of a piece of software known as the monitor.**

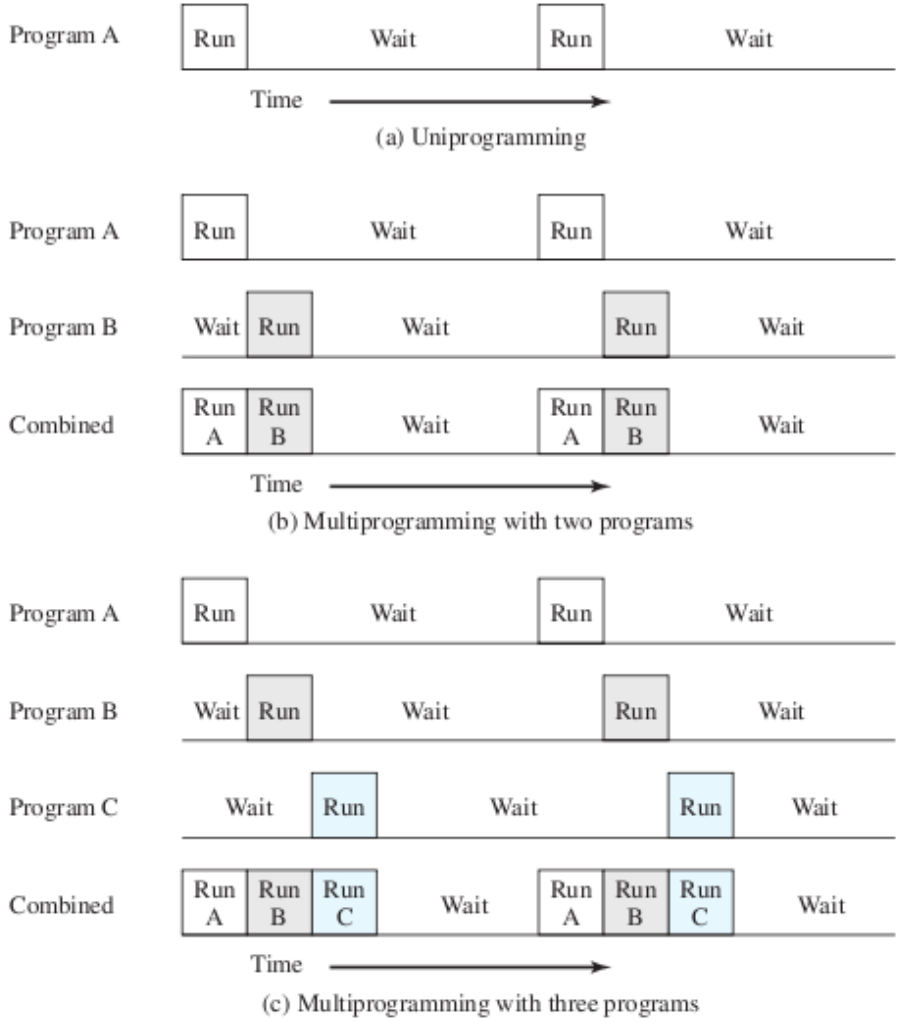


- With this type of OS, the user no longer has direct access to the processor. Instead, the user submits the job on cards or tape to a computer operator, who batches the jobs together sequentially and places the entire batch on an input device, for use by the monitor.
- Each program is constructed to branch back to the monitor when it completes processing, at which point the monitor automatically begins loading the next program.

- With a batch operating system, processor time alternates between execution of user programs and execution of the monitor. There have been two sacrifices: Some main memory is now given over to the monitor and some processor time is consumed by the monitor. Both of these are forms of overhead.

Multiprogramming OS:

- A single program cannot keep either CPU or I/O devices busy at all times.
- **Multiprogramming increases CPU utilization by organizing jobs in such a manner that CPU has always one job to execute.**
- If computer is required to run several programs at the same time, the processor could be kept busy for the most of the time by switching its attention from one program to the next.
- Additionally I/O transfer could overlap the processor activity i.e., while one program is awaiting for an I/O transfer, another program can use the processor.
- **So CPU never sits idle or if comes in idle state then after a very small time it is again busy.**



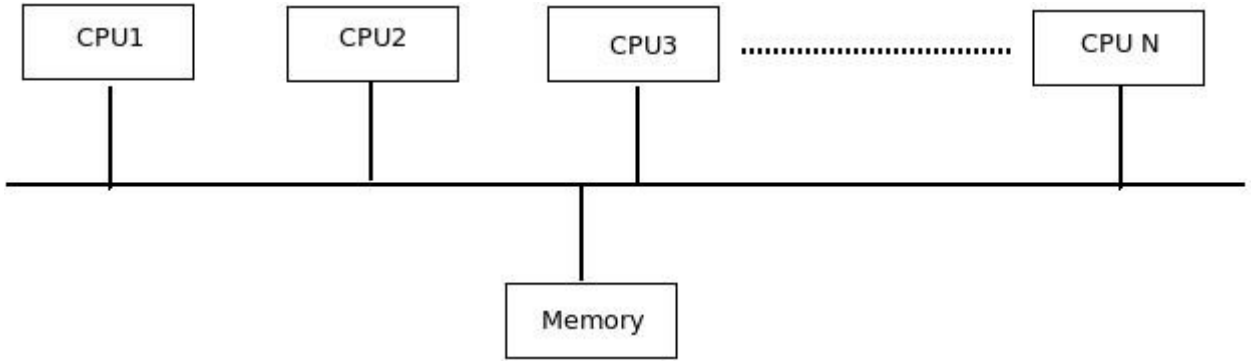
Multitasking or Time Sharing OS:

- Multiprogramming didn't provide the user interaction with the computer system.
- Time sharing or Multitasking is a logical extension of Multiprogramming that provides user interaction.
- There are more than one user interacting the system at the same time.
- The switching of CPU between two users is so fast that it gives the impression to user that he is only working on the system but actually it is shared among different users.
- CPU bound is divided into different time slots depending upon the number of users using the system.
- Just as multiprogramming allows the processor to handle multiple batch jobs at a time, multiprogramming can also be used to handle multiple interactive jobs. In this latter case, the technique is referred to as time sharing, because processor time is shared among multiple users.
- A multitasking system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time shared computer. Each user has at least one separate program in memory.
- Multitasking are more complex than multiprogramming and must provide a mechanism for jobs synchronization and communication and it may ensure that system does not go in deadlock.

Multiprocessor OS:

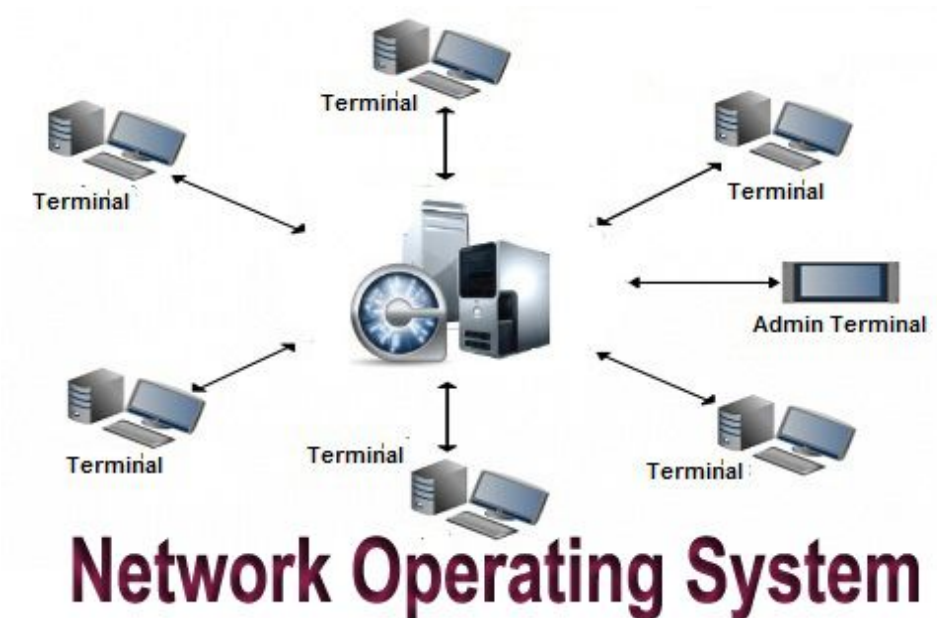
- Multiprocessor operating system aims to support high performance through the use of multiple CPUs.
- It consists of a set of processors that share a set of physical memory blocks over an interconnected network.
- An important goal is to make the number of CPUs transparent to the application. Achieving such transparency is relatively easy because the communication between different (parts of) application uses the same primitives as those in uni-processor OS.

- The idea is that all communication is done by manipulating data at the shared memory locations and that we only have to protect that data segment against simultaneous access. Protection is done through synchronization primitives like semaphores and monitors.



Network OS:

- **A network operating system (NOS) is a computer operating system (OS) that is designed primarily to support workstations, personal computers and, in some instances, older terminals that are connected on a local area network (LAN).**
- The software behind a NOS allows multiple devices within a network to communicate and share resources with each other.
- The composition of hardware that typically uses a NOS includes a number of personal computers, a printer, a server and file server with a local network that connects them together.
- The role of the NOS is to then provide basic network services and features that support multiple input requests simultaneously in a multiuser environment.



Distributed OS:

- A recent trend in computer system is to distribute computation among several processors. In contrasts to the tightly coupled system the processors do not share memory or a clock. Instead, each processor has its own local memory.
- The processors communicate with one another through various communication lines such as computer network.
- Distributed operating system are the operating system for a distributed system(a network of autonomous computers connected by a communication network through a message passing mechanisms).

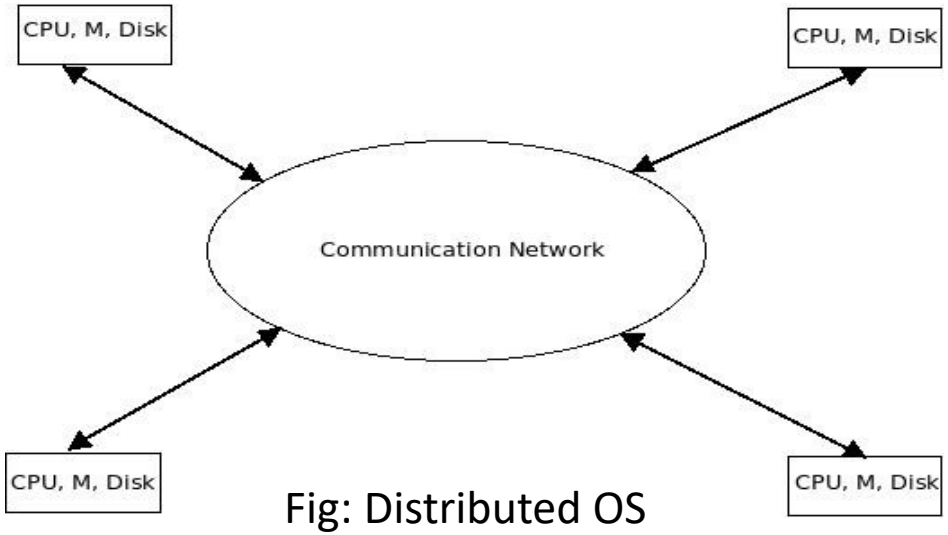


Fig: Distributed OS

- A distributed operating system controls and manages the hardware and software resources of a distributed system. When a program is executed on a distributed system, user is not aware of where the program is executed or the location of the resources accessed.

- Example of Distributed OS: Amoeba, Chorus, Alpha Kernel.

Real Time OS:

- Primary objective of Real Time Operating System is to provide quick response time and thus to meet a scheduling deadline.
- User convenience and resource utilization are secondary concern to these systems. Real time systems has many events that must be accepted and processed in a short time or within certain deadline. Such applications include:
- Rocket launching, flight control, robotics, real time simulation, telephone switching equipment's etc.

- Real time systems are classified into two categories:

  a). Soft Real time System: If certain deadlines are missed then system continues its working with no failure but its performance degrade.

  b). Hard Real time System: If any deadline is missed then system will fail to work or does not work properly. This system guarantees that critical task is completed on time.

Function of OS

Following are some of important functions of an operating System.

- Memory Management
- Processor Management
- Device Management
- File Management
- Security
- Control over system performance
- Job accounting
- Error detecting aids
- Coordination between other software and users

Function of OS (contd..)

Memory Management

Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.

Main memory provides a fast storage that can be access directly by the CPU. So for a program to be executed, it must in the main memory. Operating System does the following activities for memory management.

- Keeps tracks of primary memory i.e. what part of it are in use by whom, what part are not in use.
- In multiprogramming, OS decides which process will get memory when and how much.
- Allocates the memory when the process requests it to do so.
- De-allocates the memory when the process no longer needs it or has been terminated.

Function of OS (contd..)

Processor Management

In multiprogramming environment, OS decides which process gets the processor when and how much time. This function is called process scheduling. Operating System does the following activities for processor management.

- Keeps tracks of processor and status of process. Program responsible for this task is known as traffic controller.
- Allocates the processor (CPU) to a process.
- De-allocates processor when processor is no longer required.

Function of OS (contd..)

Device Management

OS manages device communication via their respective drivers. Operating System does the following activities for device management.

- Keeps tracks of all devices. Program responsible for this task is known as the I/O controller.
- Decides which process gets the device when and for how much time.
- Allocates the device in the efficient way.
- De-allocates devices.

Function of OS (contd..)

File Management

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions. Operating System does the following activities for file management.

- Keeps track of information, location, uses, status etc. The collective facilities are often known as file system.
- Decides who gets the resources.
- Allocates the resources.
- De-allocates the resources.

Function of OS (contd..)

Other Important Activities

Following are some of the important activities that Operating System does.

- **Security** -- By means of password and similar other techniques, preventing unauthorized access to programs and data.
- **Control over system performance** -- Recording delays between request for a service and response from the system.
- **Job accounting** -- Keeping track of time and resources used by various jobs and users.
- **Error detecting aids** -- Production of dumps, traces, error messages and other debugging and error detecting aids.
- **Coordination between other software and users** -- Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.