Unit VI: Operating System, Features of Operating System, Types of Operating Systems, Support for Networking.

OPERATING SYSTEM

- It is the software that provides an interface between the computer hardware, and the application programs or users.
- An operating system (OS) is the software that provides an interface between the computer hardware, and the application programs or users.
- An OS provides standard **services** (an interface) which are implemented on the hardware, including:
  -> Processes, CPU scheduling, memory management, file system, networking

• An operating system is a program that acts as an intermediary between a user of a computer and the computer hardware.
• The purpose of an operating system is to provide an environment in which a user can execute programs.
The primary goal of an operating system is thus to make the computer system convenient to use.
• A secondary goal is to use the computer hardware in an efficient manner.
• In brief, an operating system is the set of programs that controls a computer. Some examples of operating systems are UNIX, Mach, MS-DOS, MS Windows, Windows/NT, OS/2 and MacOS.
• An operating system is an important part of almost every computer system.
• A computer system can be divided roughly into four components: the hardware, the operating system, the application programs and the users

Objectives of Operating Systems
• To hide details of hardware by creating abstraction.
Manages access to shared hardware resources by creating arbitration
• To allocate resources to processes (Manage resources).
• Provide a pleasant and effective user interface.

FEATURES/FUNCTIONS OF OPERATING SYSTEMS:

The main functions of a modern OS are as follows:

Process Management: As a process manager, the OS handles the creation and deletion of processes, suspension and resumption of processes, and scheduling and synchronization of processes.

Memory Management: As a memory manager, the OS handles the allocation and de-allocation of memory space as required by various programs.

File Management: The OS is responsible for creation and deletion of fles and directories. It also takes care of other fle-related activities such as organizing, storing, retrieving, naming and protecting the fles.

Device Management: The OS provides input/output subsystem between process and device driver. It handles the device caches, buffers and interrupts. It also detects the device failures and notifes the same to the user.

Security Management: The OS protects system resources and information against destruction and unauthorized use.

User Interface: The OS provides the interface between the user and the hardware.

The user interface is the layer that actually interacts with the computer operator. The interface consists of a set of commands or menus through which a user communicates with a program.

Some more Illustrated features/functions of OS are as follows:
:

- Process Management

A process is only ONE instant of a program in execution. There are many processes can be running the same program. The five major activities of an operating system in regard to process management are:

- Creation and deletion of user and system processes.
- Suspension and resumption of processes.
- A mechanism for process synchronization.
- A mechanism for process communication.
- A mechanism for deadlock handling.

- Memory Management

Memory is a large array of words or bytes. Each word or byte has its own address. Memory is a repository of quickly accessible data shared by the CPU and I/O devices.
The major activities of an operating system in regard to memory-management are:

- Keep track of which part of memory are currently being used and by whom.
- Decide which processes are loaded into memory when memory space becomes available.
- Allocate and deallocate memory space as needed.

- File Management

A file is a collected of related information defined by its creator. Computer can store files on the disk (secondary storage), which provide long term storage.

- The creation and deletion of files.
- The creation and deletion of directions.
- The support of primitives for manipulating files and directions.
- The mapping of files onto secondary storage.
- The backup of files on stable storage media.

- Secondary-Storage Management

Generally speaking, systems have several levels of storage, including primary storage, secondary storage and cache storage. Instructions and data must be placed in primary storage or cache to be referenced by a running program.

- Networking

A distributed system is a collection of processors that do not share memory, peripheral devices, or a clock. The processors communicate with one another through communication lines called network.

- Protection System

Protection refers to mechanism for controlling the access of programs, processes, or users to the resources defined by a computer system.

- Command Interpreter System

A command interpreter is an interface of the operating system with the user. The user gives commands with are executed by operating system (usually by turning them into system calls).

- Program Execution

The system must be able to load a program into memory and to run it. The program must be able to end its execution, either normally or abnormally (indicating error).

• Error Detection
An error is one part of the system may cause malfunctioning of the complete system. To avoid such a situation the operating system constantly monitors the system for detecting the errors.

- System Calls and System Programs

System calls provide the interface between a process and the operating system. These calls are generally available as assembly-language instructions and are usually listed in the manuals used by assembly-language programmers.

CPU Scheduling

- CPU or processor is one of the primary computer resources. All computer resources like I/O, memory, and CPU are scheduled for use.

CPU scheduling is important for the operating system. In a multiprogramming and time sharing system, the processor executes multiple processes by switching the CPU among the processes, so that no user has to wait for long for a program to execute. To enable running of several concurrent processes, the processor time has to be distributed amongst all the processes efficiently.

Scheduler is a component of the operating system that is responsible for scheduling transition of processes. At any one time, only one process can be in running state and the rest are in ready or waiting state. The scheduler assigns the processor to different processes in a manner so that no one process is kept waiting for long.

Scheduling can be non-pre-emptive scheduling or pre-emptive scheduling. In non-pre-emptive scheduling, the processor executes a process till termination without any interruption. Hence the system resources are not used effi ciently. In pre-emptive scheduling , a running process may be interrupted by another process that needs to execute. Pre-emption allows the operating system to interrupt the executing task and handle any important task that requires immediate action. In pre-emptive scheduling, the system resources are used efficiently.

Multitasking

- We expect modern computers to do many different things at once, and we need some way to arbitrate between all the different programs running on the system. It is the operating systems job to allow this to happen seamlessly.
- The operating system is responsible for *resource management* within the system. Many tasks will be competing for the resources of the system as it runs, including processor time, memory, disk and user input. The job of the operating system is to arbitrate these resources to the multiple tasks and allow them access in an orderly fashion. You have probably experienced when this *fails* as it usually ends up with your computer crashing (the famous "blue screen of death" for example).

(Refer your Green Covered Book (computer system and IT Apps) too for functions of OS)

TYPE OF OPERATING SYSTEMS:

Single User

Single user operating systems can be split into two types:

- single user, single application operating systems
- single user, multi tasking operating systems

Single user, single application

This type of operating system only has to deal with one person at a time, running one user application at a time.

An example of a this kind of operating system would be found on a mobile phone. There can only be one user using the mobile and that person is only using one of its applications at a time.

Single user, multi-tasking

You will find this kind of operating system on a personal computer.

The operating system is designed mainly with a single user in mind, but it can deal with many applications running at the same time. For example, you might be writing an essay, while searching the internet, downloading a video file and also listening to a piece of music.

Example operating systems are

Windows

Linux

Mac OS X

The difference compared to the Single-Use, Single Application operating system is that it must now handle many different applications all running at the same time.

The memory available is also very different, for example it is quite normal to have Gigabytes of RAM available on a personal computer which is what allows so many applications to run.

Multi-user

A multi-user operating system allows multiple users to access a computer system at the same time. Time-sharing systems and Internet servers can be classified as multi-user systems as they enable multiple-user access to a computer through the sharing of time. Single-user operating systems have only one user but may allow multiple programs to run at the same time.

Network Operating System

It refers to software that implements an operating system of some kind that is oriented to computer networking. For example, one that runs on a server and enables the server to manage data, users, groups, security, applications, and other networking functions. The network operating system is designed to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks.

Real-time

A real-time operating system is a multitasking operating system that aims at executing real-time applications. Real-time operating systems often use specialized scheduling algorithms so that they can achieve a deterministic nature of behavior. The main objective of real-time operating systems is their quick and predictable response to events. They have an event-driven or time-sharing design and often aspects of both. An event-driven system switches between tasks based on their priorities or external events while time-sharing operating systems switch tasks based on clock interrupts.

Multi-tasking vs. single-tasking

A multi-tasking operating system allows more than one program to be running at the same time, from the point of view of human time scales. A single-tasking system has only one running program. Multi-tasking can be of two types: pre-emptive and co-operative. In pre-emptive multitasking, the operating system slices the CPU time and dedicates one slot to each of the programs. Unix-like operating systems such as Solaris and Linux support pre-emptive multitasking, as does AmigaOS. Cooperative multitasking is achieved by relying on each process to give time to the other processes in a defined manner. 16-bit versions of Microsoft Windows used cooperative multi-tasking. 32-bit versions of both Windows NT and Win9x, used pre-emptive multi-tasking. Mac OS prior to OS X used to support cooperative multitasking.

Distributed

A distributed operating system manages a group of independent computers and makes them appear to be a single computer. The development of networked computers that could be linked and communicate with each other gave rise to distributed computing. Distributed computations are carried out on more than one machine. When computers in a group work in cooperation, they make a distributed system.

Templated

In an o/s, distributed and cloud computing context, templating refers to creating a single virtual machine image as a guest operating system, then saving it as a tool for multiple running virtual machines (Gagne, 2012, p. 716). The technique is used both in virtualization and cloud computing management, and is common in large server warehouses.

Embedded

Embedded operating systems are designed to be used in embedded computer systems. They are designed to operate on small machines like PDAs with less autonomy. They are able to operate with a limited number of resources. They are very compact and extremely efficient by design. Windows CE and Minix 3 are some examples of embedded operating systems.

Batch processing

It is the execution of a series of programs ("jobs") on a computer without manual intervention.

It is often not desirable to deal with the inputs until a certain number have occurred or a set time has passed. So they are stored until the system comes online to process the data in one 'batch'.

It queues up programs so that as soon as one program completed, the next would start.

Batch processing is usually fully automatic unlike 'real-time' or transaction processing which are interactive.

For example

- A stock control programme may store records of every item sold in a shop that day. Then, at the end of each day it calculates what needs to be ordered.
- An online competition stores all the entries until it is time to find the winner.
- Electricity, gas and telephone bills are usually calculated on a monthly basis.
- Producing monthly bank statements to send out to customers