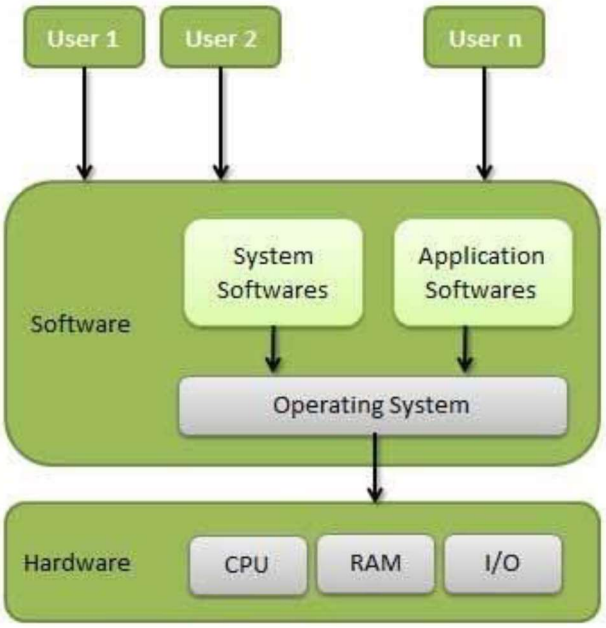
CHAPTER-3 Operating System



The operating System is defined as a set of programs that controls the entire operation of the computer such as handling Input/output operation, user request, interrupts etc. The Operating System manages the resources of a computer system and schedules its operation. It is a master supervisory program which provides a friendly interface for the user to interact with the computer and its peripherals and different application software as well as an environment for program to execute.

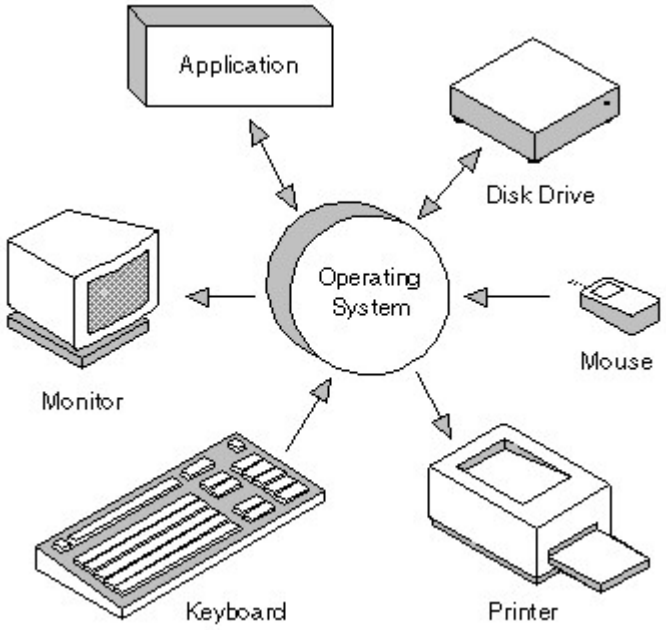
An Operating System (OS) is computer program (system software) that acts as an intermediary/ interface between users and computer hardware. This is the first software we see when we turn on the computer and again the last software we see when the computer is turned off. It provides users an environment in which a user can execute programs conveniently and efficiently. In technical terms, it is software which manages hardware. An operating System controls the allocation of resources and services such as memory, processors, devices and information.

An Operating System (OS) is an interface between computer user and computer hardware. An operating system (OS) is the program that, after being initially loaded into the computer by a boot program, manages all of the other application programs in a computer. An operating system is software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.

The operating system manages a computer's hardware resources, including:

- *Input devices such as a keyboard and mouse.*
- *Output devices such as display monitors, printers and scanners.*
- *Network devices such as modems, routers and network connections.*
- *Storage devices such as internal and external drives.*

Some popular Operating Systems include Linux, Windows, OS X, VMS, OS/400, AIX, z/OS, etc.
An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.



Following are some of important functions/features 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

1 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 accessed directly by the CPU. For a program to be executed, it must be in the main memory.

An 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, the OS **decides which process will get memory** when and how much.
- **Allocates** the memory when a process requests it to do so.
- **De-allocates** the memory when a process no longer needs it or has been terminated.

2 Processor Management

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

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

3 Device Management

An Operating System manages device communication via their respective drivers. It 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.

4 File Management

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions.

An 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.

Other Important Activities

Following are some of the important activities that an Operating System performs –

- **Security** and Protection − By means of password and similar other techniques, it prevents 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's and users** − Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.
- **User Interface or Command Interpreter -**

  Operating system provides an interface between the computer user and the computer hardware. The user interface is a set of commands or graphical user interface via which the user interacts with the applications and the hardware.

TYPES OF OPERATING SYSTEM

OS are classified into different types depending on their capability of processing—(1) Single user, (2) Multiuser, (3) Multitasking, (4) Multiprocessing, (5) Real time OS and (6) Embedded OS

Single User and Single Task OS is for use by a single user for a standalone single computer for performing a single task. Operating system for Personal Computers (PC) are single user OS. For example, if the user is editing a document, then a document cannot be printed on the printer simultaneously. Single user OS are simple operating system designed to manage one task at a time. MS-DOS is an example of single user OS.

Single User and Multitasking OS allows execution of more than one task or process concurrently. For this, the processor time is divided amongst different tasks. This division of time is also called time sharing. The processor switches rapidly between processes. For example, the user can listen to music on the computer while writing an article using a word processor software. The user can switch between the applications and also transfer data between them. Windows 95 and all later versions of Windows are examples of multitasking OS.

Multiuser OS is used in computer networks that allow same data and applications to be accessed by multiple users at the same time. The users can also communicate with each other. Linux, UNIX, and Windows 7 are examples of multiuser OS.

Multiprogramming OS: In this OS, more than one jobs resides in main memory and hence the memory is utilized. The CPU picks one job and starts executing it, when the jobs requires to perform input/output operation, the CPU does not sit idle and picks the next job and starts executing its instruction, in this way a single CPU is in demand all the time and hence it is fully utilized.

Multiprocessing OS have two or more processors for a single running process. Processing takes place in parallel and is also called parallel processing. Each processor works on different parts of the same task, or, on two or more different tasks. Since execution takes place in parallel, they are used for high speed execution, and to increase the power of computer. Linux, UNIX and Windows 7 are examples of multiprocessing OS.

Real Time OS are designed to respond to an event within a predetermined time. Real time OS are brain of real time system. Its response to input is very immediate. These operating systems are used to control processes. Processing is done within a time constraint. OS monitors the events that affect the execution of process and respond accordingly. They are used to respond to queries in areas like medical imaging system, missiles, banking sector, industrial control systems etc. Lynx OS is an example of real time OS. Type: **Hard Real-time OS:** failure if response time is too long and **Soft Real-time OS:** less accurate if the response time is too long.

Embedded OS is embedded in a device in the ROM. They are specifically configured for certain hardware and are less resource intensive. Mostly depends upon the sensors installed to the hardware machine. They are used in appliances like microwaves, washing machines, traffic control systems etc. They have relatively low memory and slow CPU.

Types of Operating System on the basis of User Interface:

a) Graphic User Interface
b) Text/Command User Interface

a. Graphic User interface

User interacts with operating system through various elements called GUI system. **Graphical User Interface (GUI)** allows the user to use icons or other visual indicators to interact with electronic devices. For example, all versions of Microsoft Windows utilize a GUI. GUI uses windows, icons, and menus to carry out commands, such as opening, deleting, and moving files.
Although many GUI operating systems are navigated through the use of a mouse, the keyboard can also be utilized by using keyboard shortcuts or arrow keys.

Features

- In GUI operating system commands line are through graphical components like menu, toolbar, folder etc.
- Because of graphical components GUI Operating are easy to learn and master.
- In GUI system we can preview the content before it gets printed that is WYSIWYG (WhatYou See Is What You Get)
- GUI Operating System supports multiple programs to execute simultaneously
- GUI operating system demands high resources.
- GUI are generally 32 or 64 bits operating system
- GUI also provides menus, buttons and other graphical objects to the user to perform different tasks.
- GUI is very easy to interact with the computer. So it is user friendly.
- It supports all components of multimedia like: Video, Audio, Graphics, Animation Text.
- It supports multitasking, multiprogramming, multithreading
- It needs large amount of memory.
- Example: Microsoft Windows, Mac OS, Solaris and Linux

b) Text/Character User Interface (TUI/CUI) or Command line Interface (CLI)

User interacts with operating system through keyboard called CUI system. A command line operating system provides a command prompt on the computer screen. The commands are given to the computer by typing on the keyboard. The commands are typed according to the predefined format. The users have to memorize commands and rules of writing these commands. It is not an easy way to interface with the computer. It supports only text not others components. It can support up to 16 bits.

Features

- *CUI* operating System commands are issued by text only
- In CUI system the user has to memorize all the commands and syntax
- WYSIWYG (What You See Is What You Get) is not possible in case of CUI system
- CUI operating system like DOS is generally single tasking.
- It is less user friendly
- It is faster than GUI
- It cannot support multimedia environment
- It requires less amount of memory
- It is generally 8 or 16 bit operating system.
- It does not contain components like menus, icons, pointing devices

Difference between Graphical User Interface (GUI) and Text User Interface (TUI)

SN	Graphical User Interface (GUI)	Text User Interface (TUI)
1	We can work with picture and pointing devices which is much easier.	We must enter or type the command so we must remember all the commands
2	It is easier to remember the graphics symbol.	It is almost impossible to know or remember all the commands of CUI
3	The appearance of GUI is very attractive.	The appearance may not be as good as that of
4	You can run any program by clicking on the mouse.	You must remember the command and type the command to run or load any program.
5	GUI users have a windows that enable a user to easy view, control, and	Although many command line environments are capable of multitasking
6	manipulate multiple things at once and is commonly much faster to do when compared to CUI	they do not offer same ease and ability to view multiple thins at once on one screen
7	Although new users may have a difficult at time learning to use the mouse operate and use a GUI, most user pick up this interface much easier when compared to a command line	Because of the memorization and familiarity needed to operate a command line interface new users find it more difficult to successfully navigate and operate a command line interface.
8	It supports pointing devices.	It doesn't support pointing devices.
9	It supports multimedia.	It doesn't support multimedia.
10	Examples: Windows, Android etc.	Examples: Dos, Unix etc.

What is Thread?

A thread is a flow of execution through the process code, with its own program counter that keeps track of which instruction to execute next, system registers which hold its current working variables, and a stack which contains the execution history. A thread shares with its peer threads few information like code segment, data segment and open files. When one thread alters a code segment memory item, all other threads see that. A thread is also called a **lightweight process**. Threads provide a way to improve application performance through parallelism. Threads represent a software approach to improving performance of operating system by reducing the overhead thread is equivalent to a classical process.

Each thread belongs to exactly one process and no thread can exist outside a process. Each thread represents a separate flow of control. Threads have been successfully used in implementing network servers and web server. They also provide a suitable foundation for parallel execution of applications on shared memory multiprocessors. The following figure shows the working of a single-threaded and a multithreaded process.

Difference between Process and Thread

S.N.	Process	Thread
1	Process is heavy weight or resource intensive.	Thread is light weight, taking lesser resources than a process.
2	Process switching needs interaction with operating system.	Thread switching does not need to interact with operating system.
3	In multiple processing environments, each process executes the same code but has its own memory and file resources.	All threads can share same set of open files, child processes.
4	If one process is blocked, then no other process can execute until the first process is unblocked.	While one thread is blocked and waiting, a second thread in the same task can run.
5	Multiple processes without using threads use more resources.	Multiple threaded processes use fewer resources.
6	In multiple processes each process operates independently of the others.	One thread can read, write or change another thread's data.

Advantages of Thread

- Threads minimize the context switching time.
- Use of threads provides concurrency within a process.
- Efficient communication.
- It is more economical to create and context switch threads.
- Threads allow utilization of multiprocessor architectures to a greater scale and efficiency.

Types of Thread

Threads are implemented in following two ways −

- **User Level Threads** − User managed threads.
- **Kernel Level Threads** − Operating System managed threads acting on kernel, an operating system core.

TYPES OF OPEN SOURCE OPERATING SYSTEM

In general, **open source** refers to any program whose **source** code is made available for use or modification as users or other developers see fit. **Open source** software is usually developed as a public collaboration and made freely available.

An operating system whose source code is freely distributed with a license to study, change and further distributed to anyone for any purpose is called open source OS.

Linux
Linux is the best known and most popular open source operating system. As an operating system, Linux is software that sits underneath all of the other software on a computer, receiving requests from those programs and relaying these requests to the computer's hardware. Widely used by Network and System Administrators. For e.g CentOS, Fedora, Redhat, Ubuntu, etc..

Cosmos
This is an open source operating system written mostly in programming language C#. Its full form is **C# Open Source Managed Operating System.** Till 2016, Cosmos did not intend to be a fully-fledged operating system but a system that allowed other developers to easily build their own operating systems. It also hid the inner workings of the hardware from the developer's thus providing data abstraction.

FreeDOS
This was a free operating system developed for systems compatible with **IBM PC computers.** FreeDOS provides a complete **environment** to **run legacy software and other embedded systems.** It can booted from a floppy disk or USB flash drive as required. FreeDos is licensed under the GNU General Public license and contains free and open source software. So there is no license fees required for its distribution and changes to the system are permitted.

Genode
Genode is free as well as open source. It contains a **microkernel layer and different user components.** It is one of the few open source operating systems not derived from a licensed operating system such as Unix. Genode can be used as an operating system for computers, tablets etc. as required. It is also used as a base for **virtualization, inter-process communication, software development etc.** as it has a small code system. This OS is written in C++.

Ghost OS
This is a free, open source operating system developed for **personal computers.** It started as a research project and developed to contain various advanced **features like graphical user interface, C library etc.** The Ghost operating system features **multiprocessing and multitasking** and is based on the Ghost Kernel. Most of the programming in Ghost OS is done in **C++**.

ITS
The **incompatible time-sharing system** was developed by the MIT Artificial Intelligence Library. It is principally a time sharing system. There is a remote login facility which allowed guest users to informally try out the operating system and its features using ARPAnet. ITS also gave out many new features that were unique at that time such as **device independent graphics terminal, virtual devices, inter machine file system access etc.**

OSv
This was an operating system released in 2013. It was mainly focused on **cloud computing** and was built to run on top of a virtual machine as a guest. This is the reason it **doesn't include drivers for bare hardware.** In the OSv operating system, everything runs in the kernel address space and **there is no concept of a multi-user system.** It is written in C++.

Phantom OS
This is an operating system that is based on the concepts on **persistent virtual memory** and is code oriented. It was mostly developed by **Russian developers.** Phantom OS is not based on concepts of famous operating systems such as Unix.Its **main goal is simplicity and effectiveness in process management.** Phantom is based on the principle that "Everything is an object", in contrast to the Unix-like approach of "Everything is a file". It is written in C and JAVA.