What is Operating System?

An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a 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.

Generally, a **Computer System** consists of the following components:

- **Computer Users** are the users who use the overall computer system.
- **Application Softwares** are the softwares which users use directly to perform different activities. These softwares are simple and easy to use like Browsers, Word, Excel, different Editors, Games etc. These are usually written in high-level languages, such as Python, Java and C++.
- **System Softwares** are the softwares which are more complex in nature and they are more near to computer hardware. These software are usually written in low-level languages like assembly language and includes **Operating Systems** (Microsoft Windows, macOS, and Linux), Compiler, and Assembler etc.
- **Computer Hardware** includes Monitor, Keyboard, CPU, Disks, Memory, etc.

So now let's put it in simple words:

> If we consider a Computer Hardware is body of the Computer System, then we can say an Operating System is its soul which brings it alive ie. operational. We can never use a Computer System if it does not have an Operating System installed on it.

Operating System - Examples

There are plenty of Operating Systems available in the market which include paid and unpaid (Open Source). Following are the examples of the few most popular Operating Systems:

- **Windows:** This is one of the most popular and commercial operating systems developed and marketed by Microsoft. It has different versions in the market like Windows 8, Windows 10 etc and most of them are paid.
- **Linux** This is a Unix based and the most loved operating system first released on September 17, 1991 by Linus Torvalds. Today, it has 30+ variants available like Fedora, OpenSUSE, CentOS, UBuntu etc. Most of them are available free of charges though you can have their enterprise versions by paying a nominal license fee.
- **MacOS** This is again a kind of Unix operating system developed and marketed by Apple Inc. since 2001.
- **iOS** This is a mobile operating system created and developed by Apple Inc. exclusively for its mobile devices like iPhone and iPad etc.
- **Android** This is a mobile Operating System based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets.

Some other old but popular Operating Systems include Solaris, VMS, OS/400, AIX, z/OS, etc.

Operating System - Functions

To brief, Following are some of important functions of an operating System which we will look in more detail in upcoming chapters:

- Process Management
- I/O Device Management
- File Management
- Network Management
- Main Memory Management
- Secondary Storage Management
- Security Management
- Command Interpreter System
- Control over system performance
- Job Accounting
- Error Detection and Correction
- Coordination between other software and users
- Many more other important tasks

Operating Systems - History

Operating systems have been evolving through the years. In the 1950s, computers were limited to running one program at a time like a calculator, but later in the following decades, computers began to include more and more software programs, sometimes called libraries, that formed the basis for today's operating systems.

The first Operating System was created by General Motors in 1956 to run a single IBM mainframe computer, its name was the IBM 704. IBM was the first computer manufacturer to develop operating systems and distribute them in its computers in the 1960s.

There are few facts about Operating System evaluation:

- Stanford Research Institute developed the oN-Line System (NLS) in the late 1960s, which was the first operating system that resembled the desktop operating system we use today.
- Microsoft bought QDOS (Quick and Dirty Operating System) in 1981 and branded it as Microsoft Operating System (MS-DOS). As of 1994, Microsoft had stopped supporting MS-DOS.
- Unix was developed in the mid-1960s by the Massachusetts Institute of Technology, AT&T Bell Labs, and General Electric as a joint effort. Initially it was named MULTICS, which stands for Multiplexed Operating and Computing System.
- FreeBSD is also a popular UNIX derivative, originating from the BSD project at Berkeley. All modern Macintosh computers run a modified version of FreeBSD (OS X).
- Windows 95 is a consumer-oriented graphical user interface-based operating system built on top of MS-DOS. It was released on August 24, 1995 by Microsoft as part of its Windows 9x family of operating systems.
- Solaris is a proprietary Unix operating system originally developed by Sun Microsystems in 1991. After the Sun acquisition by Oracle in 2010 it was renamed Oracle Solaris.

Why to Learn Operating System

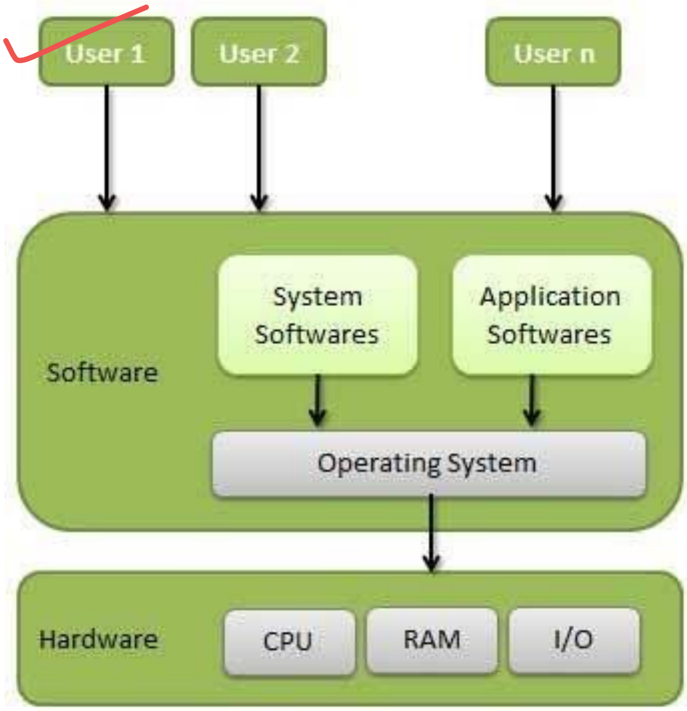
If you are aspiring to become a Great Computer Programmer then it is highly recommended to understand how exactly an Operating System works inside out. This gives opportunity to understand how exactly data is saved in the disk, how different processes are created and scheduled to run by the CPU, how to interact with different I/O devices and ports.

There are various low level concepts which help a programmer to Design and Develop scalable softwares. Bottom line is without a good understanding of Operating System Concepts, it can't be assumed someone to be a good Computer **Application Software** developer, and even it is unimaginable imagine someone to become a **System Software** developer without knowing Operating System in-depth.

If you are a fresher and applying for a job in any standard company like Google, Microsoft, Amazon, IBM etc then it is very much possible that you will be asked questions related to Operating System concepts.

Architecture

We can draw a generic architecture diagram of an Operating System which is as follows:



Operating System Generations

Operating systems have been evolving over the years. We can categorise this evaluation based on different generations which is briefed below:

0th Generation

The term 0th generation is used to refer to the period of development of computing when Charles Babbage invented the Analytical Engine and later John Atanasoff created a computer in 1940. The hardware component technology of this period was electronic vacuum tubes. There was no

Operating System available for this generation computer and computer programs were written in machine language. This computers in this generation were inefficient and dependent on the varying competencies of the individual programmer as operators.

First Generation (1951-1956)

The first generation marked the beginning of commercial computing including the introduction of Eckert and Mauchly's UNIVAC I in early 1951, and a bit later, the IBM 701.

System operation was performed with the help of expert operators and without the benefit of an operating system for a time though programs began to be written in higher level, procedure-oriented languages, and thus the operator's routine expanded. Later mono-programmed operating system was developed, which eliminated some of the human intervention in running job and provided programmers with a number of desirable functions. These systems still continued to operate under the control of a human operator who used to follow a number of steps to execute a program. Programming language like FORTRAN was developed by John W. Backus in 1956.

Second Generation (1956-1964)

The second generation of computer hardware was most notably characterised by transistors replacing vacuum tubes as the hardware component technology. The first operating system GMOS was developed by the IBM computer. GMOS was based on single stream batch processing system, because it collects all similar jobs in groups or batches and then submits the jobs to the operating system using a punch card to complete all jobs in a machine. Operating system is cleaned after completing one job and then continues to read and initiates the next job in punch card.

Researchers began to experiment with multiprogramming and multiprocessing in their computing services called the time-sharing system. A noteworthy example is the Compatible Time Sharing System (CTSS), developed at MIT during the early 1960s.

Third Generation (1964-1979)

The third generation officially began in April 1964 with IBM's announcement of its System/360 family of computers. Hardware technology began to use integrated circuits (ICs) which yielded significant advantages in both speed and economy.

Operating system development continued with the introduction and widespread adoption of multiprogramming. The idea of taking fuller advantage of the computer's data channel I/O capabilities continued to develop.

Another progress which leads to developing of personal computers in fourth generation is a new development of minicomputers with DEC PDP-1. The third generation was an exciting time, indeed, for the development of both computer hardware and the accompanying operating system.

Fourth Generation (1979 – Present)

The fourth generation is characterised by the appearance of the personal computer and the workstation. The component technology of the third generation, was replaced by very large scale integration (VLSI). Many Operating Systems which we are using today like Windows, Linux, MacOS etc developed in the fourth generation.

Following are some of important functions of an operating System.

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

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 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.

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.

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.

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** – 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 softwares and users** – Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

Components of Operating System

There are various components of an Operating System to perform well defined tasks. Though most of the Operating Systems differ in structure but logically they have similar components. Each component must be a well-defined portion of a system that appropriately describes the functions, inputs, and outputs.

There are following 8-components of an Operating System:

1. Process Management
2. I/O Device Management
3. File Management
4. Network Management
5. Main Memory Management
6. Secondary Storage Management
7. Security Management
8. Command Interpreter System

Following section explains all the above components in more detail:

Process Management

A process is program or a fraction of a program that is loaded in main memory. A process needs certain resources including CPU time, Memory, Files, and I/O devices to accomplish its task. The process management component manages the multiple processes running simultaneously on the Operating System.

A program in running state is called a process.

The operating system is responsible for the following activities in connection with process management:

- Create, load, execute, suspend, resume, and terminate processes.
- Switch system among multiple processes in main memory.
- Provides communication mechanisms so that processes can communicate with each others
- Provides synchronization mechanisms to control concurrent access to shared data to keep shared data consistent.
- Allocate/de-allocate resources properly to prevent or avoid deadlock situation.

I/O Device Management

One of the purposes of an operating system is to hide the peculiarities of specific hardware devices from the user. I/O Device Management provides an abstract level of H/W devices and keep the details from applications to ensure proper use of devices, to prevent errors, and to provide users with convenient and efficient programming environment.

Following are the tasks of I/O Device Management component:

- Hide the details of H/W devices
- Manage main memory for the devices using cache, buffer, and spooling
- Maintain and provide custom drivers for each device.

File Management

File management is one of the most visible services of an operating system. Computers can store information in several different physical forms; magnetic tape, disk, and drum are the most common forms.

A file is defined as a set of correlated information and it is defined by the creator of the file. Mostly files represent data, source and object forms, and programs. Data files can be of any type like alphabetic, numeric, and alphanumeric.

A files is a sequence of bits, bytes, lines or records whose meaning is defined by its creator and user.

The operating system implements the abstract concept of the file by managing mass storage device, such as types and disks. Also files are normally organized into directories to ease their use. These directories may contain files and other directories and so on.

The operating system is responsible for the following activities in connection with file management:

- File creation and deletion
- Directory creation and deletion

- The support of primitives for manipulating files and directories
- Mapping files onto secondary storage
- File backup on stable (nonvolatile) storage media

Network Management

The definition of network management is often broad, as network management involves several different components. Network management is the process of managing and administering a computer network. A computer network is a collection of various types of computers connected with each other.

Network management comprises fault analysis, maintaining the quality of service, provisioning of networks, and performance management.

Network management is the process of keeping your network healthy for an efficient communication between different computers.

Following are the features of network management:

- Network administration
- Network maintenance
- Network operation
- Network provisioning
- Network security

Main Memory Management

Memory is a large array of words or bytes, each with its own address. It is a repository of quickly accessible data shared by the CPU and I/O devices.

Main memory is a volatile storage device which means it loses its contents in the case of system failure or as soon as system power goes down.

The main motivation behind Memory Management is to maximize memory utilization on the computer system.

The operating system is responsible for the following activities in connections with memory management:

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

Secondary Storage Management

The main purpose of a computer system is to execute programs. These programs, together with the data they access, must be in main memory during execution. Since the main memory is too small to permanently accommodate all data and program, the computer system must provide secondary storage to backup main memory.

Most modern computer systems use disks as the principle on-line storage medium, for both programs and data. Most programs, like compilers, assemblers, sort routines, editors, formatters, and so on, are stored on the disk until loaded into memory, and then use the disk as both the source and destination of their processing.

The operating system is responsible for the following activities in connection with disk management:

- Free space management
- Storage allocation

Disk scheduling

Security Management

The operating system is primarily responsible for all task and activities happen in the computer system. The various processes in an operating system must be protected from each other's activities. For that purpose, various mechanisms which can be used to ensure that the files, memory segment, cpu and other resources can be operated on only by those processes that have gained proper authorization from the operating system.

> Security Management refers to a mechanism for controlling the access of programs, processes, or users to the resources defined by a computer controls to be imposed, together with some means of enforcement.

For example, memory addressing hardware ensure that a process can only execute within its own address space. The timer ensure that no process can gain control of the CPU without relinquishing it. Finally, no process is allowed to do it's own I/O, to protect the integrity of the various peripheral devices.

Command Interpreter System

One of the most important component of an operating system is its command interpreter. The command interpreter is the primary interface between the user and the rest of the system.

Command Interpreter System executes a user command by calling one or more number of underlying system programs or system calls.

> Command Interpreter System allows human users to interact with the Operating System and provides convenient programming environment to the users.

Many commands are given to the operating system by control statements. A program which reads and interprets control statements is automatically executed. This program is called the shell and few examples are Windows DOS command window, Bash of Unix/Linux or C-Shell of Unix/Linux.

Other Important Activities

An Operating System is a complex Software System. Apart from the above mentioned components and responsibilities, there are many other activities performed by the Operating System. Few of them are listed below:

- **Security** – 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 softwares and users** – Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.