Chapter One: Introduction

Overview of Operating Systems (OS)

What is Operating System?

Operating system is the core software component of your computer. It performs many functions and is, in very basic terms, an interface between your computer and the outside world. A computer is a modern system consists of one or more processor, some main memory, disks, printers, a keyboard, display and other I/O systems. This is a complex system. Thus writing programs that keeps track of all these components and use them correctly is a mandatory. For these reason, computers are provided with a layer of software called the O***perating Systems (OS***).The job of OS is to manage all these devices and provide user program with a simpler interface to the hardware.

An operating system is the actual software that controls the allocation and use of a computer's hardware. It keeps components working in unity, acting as a communicator between the user, the computer's hardware and software.

Operating system is system software that controls the execution of programs and that provides services such as resource allocation, scheduling, I/O control and data/file management. It hides the complexity of how hardware components work.

Operating System interprets commands and instructions. It also coordinates compilers, assemblers, utility programs, and other software to the various user of the computer system. OS provides easy communication between the computer system and the computer operator (human). It also establishes data security and integrity.

Basic Functions of operating system

Operating systems perform basic tasks, such as recognizing input from the keyboard, sending output to the display screen, keeping track of files and directories on the disk, and controlling peripheral devices such as disk drives and printers.

For large systems, the operating system has even greater responsibilities and powers. It makes sure that different program and users running at the same time do not interfere with each other. The operating system is also responsible for security, ensuring that unauthorized users do not access the system.

Today most operating systems perform the following important functions:

1. **Process Management**:

That is, assignment of processor to different tasks/processes being performed by the computer system. This allows two or more processes to be executed at a time. Here the operating system must decide if it can run the different processes on single processor at a time, using multitasking concept.

The operating system needs to allocate enough of the processor's time to each process and application so that they can run as efficiently as possible. This is particularly important for multitasking. When the user has multiple applications and processes running, it is up to the operating system to ensure that they have enough resources to run properly.

2**. Memory management**:

That is, allocation of main memory and other storage areas to the system programs as well as user programs and data. This involves allocating, and often to create a virtual memory for program. Paging which means organizing data so that the program data is loaded into pages of memory. Another method of managing memory is swapping. This involves swapping the content of memory to disk storage.

The operating system needs to ensure that each process has enough memory to execute the process, while also ensuring that one process does not use the memory allocated to another process. This must also be done in the most efficient manner. A computer has four general types of memory. In order of speed, they are: high-speed cache, main memory, secondary memory, and disk storage. The operating system must balance the needs of each process with the different types of memory available.

3. **Input/output management or Device Management**:

That is, co-ordination and assignment of the different output and input device while one or more programs are being executed. Most computers have additional hardware, such as printers and scanners, connected to them. These devices require drivers, or special programs that translate the electrical signals sent from the operating system or application program to the hardware device. The operating system manages the input to and output from the computer.

4. **File Management**:

That is, the storage of file on various storage devices. It also allows all files to be easily changed and modified using text editors or some other files manipulation. An Operating System can create and maintain a file System, where users can create, delete and move files around a structured file system.

At the simplest level, an operating system does two things:

1. It manages the hardware and software resources of the system.
2. It provides a stable, consistent way for applications, for users, to deal with the hardware without having to know all the details of the hardware.

Types of Operating Systems

There are several types of operating systems, with Windows, Linux and Macintosh group being the most widely used. Here is an overview on each system:

Windows: Windows is the popular Microsoft brand preferred by most personal users. Although Windows has made steps regarding security, it has a status for being one of the most vulnerable systems.

Unix/Linux: The UNIX operating system is well known for its stability. UNIX is often used more as a server than a workstation. Linux was based on the UNIX system, with the source code being a part of GNU open-source project. Both systems are very secure yet far more complex than Windows.

Macintosh: Recent versions of the Macintosh operating system, including the Mac OS X, follow the secure architecture of UNIX. Systems developed by Apple are efficient and easy to use, but can only function on Apple branded hardware.

Within the broad family of operating systems, there are generally four types, categorized based on the types of computers they control and the sort of applications they support. The categories are:

- **Real-time operating system** (RTOS) - Real-time operating systems are used to control machinery, scientific instruments and industrial systems. An RTOS typically has very little user-interface capability, and no end-user utilities, since the system will be a "sealed box" when delivered for use. A very important part of an RTOS is managing the resources of the computer so that an operation executes in precisely the same amount of time, every time it occurs.

- **Single user-single tasking operating system**

Normally allows only one user to access the system at a time. Operating system of this type is used by many microprocessor systems. It runs one program at a time. Examples of single user operating system include MS-DOS (Micro Soft Disk Operating System) produced by Microsoft.

Advantage

Since only one user process is being executed at any time, all the resources are available to the process. So its execution speed is good.

Disadvantage

A number of resources are present in a computer system and only one of these resources being utilized by the process, all other resources remain idle. So there is a low degree of resource utilization. It is serial processing system.

- **Single user, multi-tasking operating system**

In this operating system, all resources are dedicated to only one user. But the user can use more than one program (resource). It overcomes over a single user-single tasking operating system. Windows (98, 2000, XP, Vista etc) are good examples of this operating system.

- **Multi-user, multi-tasking operating system**

It is mostly used on the networked environment. More than one user can utilized the resources available on the computer like hard wares and soft wares.

E.g. Linux, UNIX, etc .

When OS start its work?

Although operating systems differ, many of their basic functions are similar. Most users today have a computer with a hard disk drive. When the computer is turned on, the operating system will be loaded from the hard drive into the computer's memory, thus making it available for use. The process of loading the operating system into memory is called bootstrapping, or booting the system. The word booting is used because, figuratively speaking, the operating system pulls itself up by its own bootstraps.

When you turn on the power to a computer, the first program that runs is usually a set of instructions kept in the computer's read-only memory (ROM). This code examines the system hardware to make sure everything is functioning properly. This power-on self test

(POST) checks the CPU, memory, and basic input-output systems (BIOS) for errors and stores the result in a special memory location. Once the POST has successfully completed, the software loaded in ROM (sometimes called the BIOS or firmware) will begin to activate the computer's disk drives. In most modern computers, when the computer activates the hard disk drive, it finds the first piece of the operating system: the bootstrap loader.

The bootstrap loader is a small program that has a single function: It loads the operating system into memory and allows it to begin operation. In the most basic form, the bootstrap loader sets up the small driver programs that interface with and control the various hardware subsystems of the computer. It sets up the divisions of memory that hold the operating system, user information and applications. Then it turns control of the computer over to the operating system.