

CHAPTER 1

FUNDAMENTALS OF OPERATING SYSTEM

- ***Introduction to Operating Systems***
- ***Operating System Structure and Operations***
- ***Functions of Operating Systems***
- ***Operating System Services and Interface***
- ***System Calls and its Types***
- ***System Programs***
- ***Operating System Structure***
- ***System Boot***

An operating system is a program that controls the execution of application programs and acts as an interface between the 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 conveniently and efficiently.

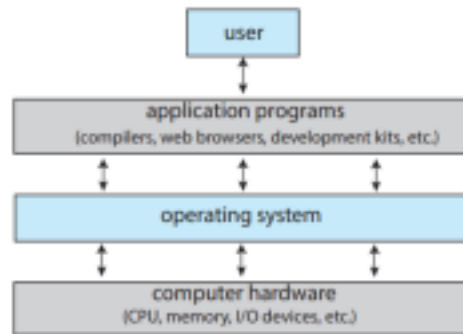


Fig.: An overview of components of computer system.

• Goals of an Operating System

The user goals or requirements should be as follows:

1. The OS usage should be convenient
2. Should be easy to use and easy to learn
3. Should be safe and secure to use and information handling, and security should be robust
4. Should be fast in response to the user requests

The system design requirements should be as follows:

- The OS should be easy to design, maintain and also to implement.
- Updates should be delivered in an easy way.
- Adequate security and privacy should be implemented.
- Resource sharing should be enabled for processes.
- Should support present software and hardware and should be scalable for future requirements.
- Should be backward compatible.

• Objectives of an Operating System

- **Efficiency:** The operating system increases the production efficiency. This is because the system configuration takes less time. By default, the operating system handles system tasks such as allocating resources to processes and resolving conflicts between different programs and users. This saves the user time and results in a more efficient result.
- **Hardware abstraction:** The operating system performs a good job of concealing the computer's intricate details. The user can fully utilise the computer hardware without having to cope with the accompanying difficulties. The operating system coordinates communication between user programs and computer hardware.

- **Convenience:** In the absence of an operating system, users would have to deal with the hardware directly without access to the pre-configured utility packages that come with an operating system. This would make using a computer extremely inconvenient. Operating systems allow users to go right to work on the tasks they want to do without dealing with the burden of setting up the system beforehand.
- **System resource management:** The operating system serves as a neutral arbitrator. It serves a management role in the computer system by ensuring equitable resource distribution among various operations and consumers.

• **Evolution of an Operating System**

Generation	Period	Computer Architecture	Development of an OS
First	1940s – 1950s	Vacuum tubes based technology, plug boards and punched cards, magnetic core memories	No operating system
Second	1950s – 1960s	Mainframe computers, line printers, magnetic tapes, assemblers, linkers, loaders, compilers, FORTRAN, COBOL	Automated job sequencing Resident monitor Batch systems Offline operation with magnetic tapes
Third	1960s – 1980s	IC-based technology, Minicomputer, Magnetic disk	Multi-programming Multi-programmed batch systems Time-sharing multi user systems UNIX

Fourth	1980s – Present	LSI- and VLSI based technology, Microcomputer	MS-DOS Apple Macintosh Windows Multi tasking Multi threading X Windows Network operating systems Distributed operating systems
--------	-----------------	---	---

3

• Types of an Operating System

1. Batch Operating System: This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having the same requirement and group them into batches. It is the responsibility of the operator to sort jobs with similar needs.

Examples: Payroll System, Bank Statements, etc.

2. Multiprogramming Operating System: Multiprogramming is an extension to batch processing where the CPU is always kept busy. Each process needs two types of system time: CPU time and IO time.

Examples: Windows O/S, UNIX O/S, Microcomputers such as XENIX, MP/M, and ESQview etc.

3. Multiprocessing Operating System: In Multiprocessing, Parallel computing is achieved. There are more than one processor present in the system which can execute more than one process at the same time. This will increase the throughput of the system.

4. Time-Sharing Operating Systems: Each task is given some time to execute so that all the tasks work smoothly. Each user gets the time of CPU as they use a single system. These systems are also known as Multitasking Systems. The task can be from a single user or different users also. The time that each task gets to execute is called quantum. After this time interval is over OS switches over to the next task.

Examples: Multics, Unix, etc.

5. Distributed Operating System: These are referred to as loosely coupled systems or distributed systems. These system's processors differ in size and function. The major benefit of working with these types of the operating system is that it is always possible that one user can access the files or software which are not actually present on his system but some other system connected within this network i.e., remote access

is enabled within the devices connected in that network.

Examples: LOCUS, etc.

- 6. Network Operating System:** These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions. These types of operating systems allow shared access of files, printers, security, applications, and other networking functions over a small private network. these computers are popularly known as **tightly coupled systems**.

Example: Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD, etc.

- 7. Real-Time Operating System:** These types of OSs serve real-time systems. The time interval required to process and respond to inputs is very small. This time interval

4

is called response time. Real-time systems are used when there are time requirements that are very strict like missile systems, air traffic control systems, robots, etc.

Examples: Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

➤ OPERATING SYSTEM STRUCTURE

1. Simple Structure

There are many operating systems that have a rather simple structure. These started as small systems and rapidly expanded much further than their scope. A common example of this is MS-DOS. It was designed simply for a niche amount for people.

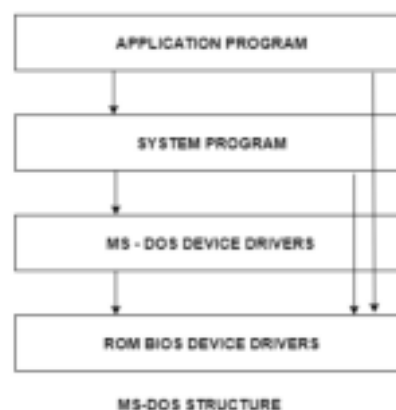


Fig.: Simple Structure

2. Monolithic OS

The monolithic operating system is a very basic operating system in which file management, memory management, device management, and process management are directly controlled within the kernel. The kernel can access all the resources present in the system. Operating systems that use monolithic architecture were first time used in the 1970s. The monolithic operating system is also known as the monolithic kernel. The

monolithic kernel acts as a virtual machine that controls all hardware parts.

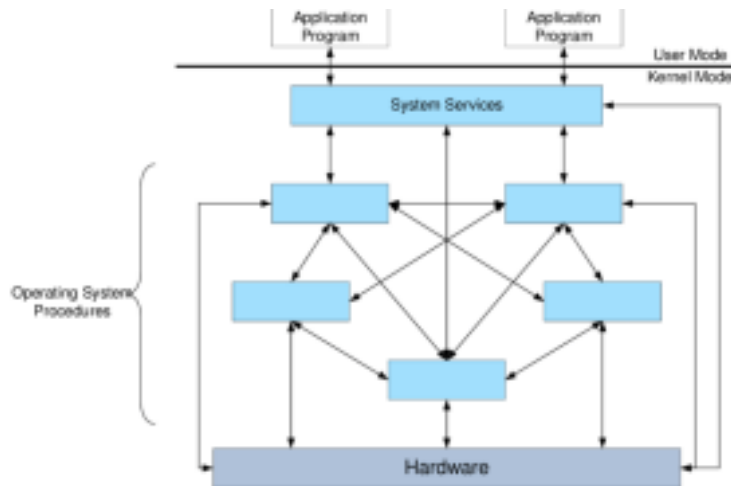


Fig.: Monolithic OS Structure

5

3. Layered Structure

A system can be made modular in many ways and one of the methods is the layered approach in which the operating system is broken into several layers. Generally, the structure of an operating system depends on how the various common components are interconnected with each other and how it is combined with the kernel. The layered structure of an operating system is the improvised approach of the early monolithic systems.

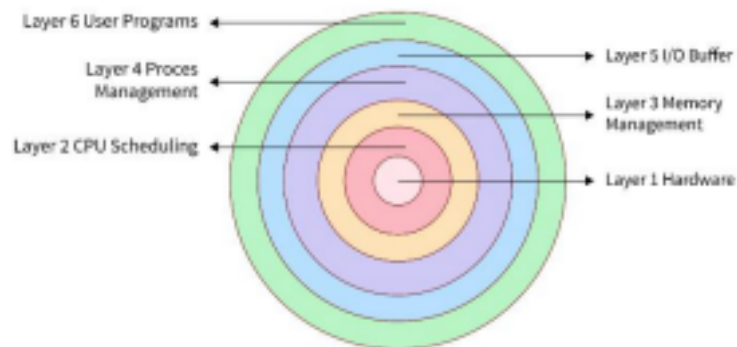


Fig.: Layered Structure

4. Micro Kernel OS

Micro kernel is the type of the operating system architecture which is useful in the same way as the other architectures are and is used for file management, memory management and the scheduling of the processes. The basic idea of microkernel design is to achieve high reliability by splitting the operating system up into small, well defined modules. The microkernel OS runs in kernel mode.

The main function of microkernel is to provide a communication facility between the client program and various services that are running in user space.

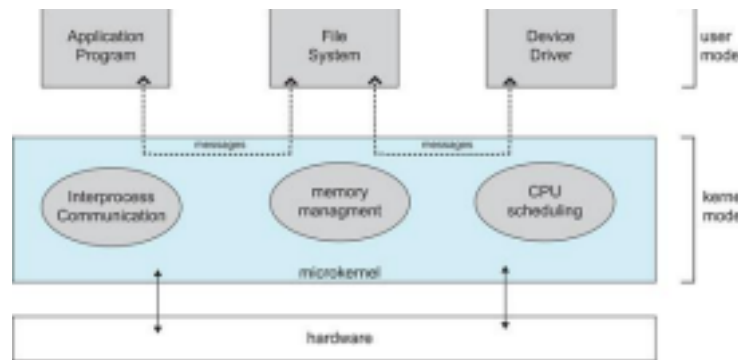


Fig. Micro Kernel OS Structure

5. Client Server OS

Client/Server communication involves two components, namely a client and a server. They are usually multiple clients in communication with a single server. The clients send requests to the server and the server responds to the client requests.

6

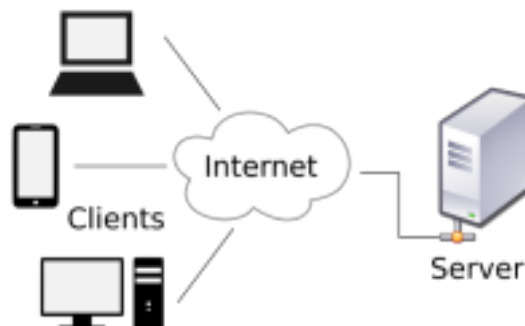


Fig. Client Server OS Structure

➤ OPERATING SYSTEM OPERATIONS

1. Multiprogramming and Multitasking

One of the most important aspects of operating systems is the ability to run multiple programs, as a single program cannot keep CPU or the I/O devices busy at all times. Users typically want to run more than one program at a time as well. Multiprogramming increases CPU utilization, as well as keeping users satisfied, by organizing programs so that the CPU always has one to execute. In a multi-programmed system, a program in execution is termed a process.

Multitasking is a logical extension of multiprogramming. In multitasking systems, the CPU executes multiple processes by switching among them, but the switches occur frequently, providing the user with a fast response time.

2. Dual mode and multi-mode

The concept of modes of operation in operating system can be extended beyond the dual mode. This is known as the multimode system. In those cases the more than 1 bit

is used by the CPU to set and handle the mode.

An example of the multimode system can be described by the systems that support virtualisation. These CPU's have a separate mode that specifies when the virtual machine manager (VMM) and the virtualisation management software is in control of the system.

3. Timer

We must ensure that the operating system maintains control over the CPU. We cannot allow a user program to get stuck in **an infinite loop or to fail** to call system services and **never return control to the operating system**. To accomplish this goal, we can use a timer. A timer can be set to interrupt the computer after a specified period. The period may be fixed or variable. **A variable timer is generally implemented by a fixed-rate clock and a counter. The operating system sets the counter.**

User Interface or Command Interpreter

The user interacts with the computer system through the operating system. Hence OS acts as an interface between the user and the computer hardware. This user interface is offered through a set of commands or a graphical user interface (GUI). Through this interface, the user makes interacts with the applications and the machine hardware.

Booting the Computer

The process of starting or restarting the computer is known as booting. If the computer is switched off completely and if turned on then it is called cold booting. Warm booting is a process of using the operating system to restart the computer.

Security

The operating system uses password protection to protect user data and similar other techniques. it also prevents unauthorized access to programs and user data. The operating system provides various techniques which assure the integrity and confidentiality of user data. The following security measures are used to protect user data:

Protection against unauthorized access through login. Protection against intrusion by keeping the firewall active.

1. Memory Management: It is the management of the main or primary memory. Whatever program is executed, it has to be present in the main memory. Main memory is a quick storage area that may be accessed directly by the CPU. When the program is completed, the memory region is released and can be used by other programs. Therefore, there can be more than one program present at a time. Hence, it is required to manage the memory.

The operating system:

- **Allocates** and deallocates the memory.
- **Keeps a record of which part of primary memory** is used by whom and how much. •

Distributes the memory while multiprocessing.

- In multiprogramming, the operating system selects which processes acquire memory when and how much memory they get.

2. Processor Management/Scheduling: Every software that runs on a computer, whether in the background or in the frontend, is a process. Processor management is an execution unit in which a program operates. The operating system determines the status of the processor and processes, selects a job and its processor, **allocates the processor to the process**, and de-allocates the processor after the process is completed.

Certain algorithms used for CPU scheduling are as follows:

- First Come First Serve (FCFS)
- Shortest Job First (SJF)

- Round-Robin Scheduling
- Priority-based scheduling etc.

3. Device Management: An operating system regulates device connection using drivers. The processes may require devices for their use. This management is done by the OS. The OS:

- Allocates and deallocates devices to different processes.
- Keeps records of the devices.
- Decides which process can use which device for how much time.

4. File Management: The operating system manages resource allocation and de-allocation. It specifies which process receives the file and for how long. It also keeps track of information, location, uses, status, and so on. These groupings of resources are referred to as file systems. The files on a system are stored in different directories. The OS:

- Keeps records of the status and locations of files.
- Allocates and deallocates resources.
- Decides who gets the resources.

8

5. Storage Management: Storage management is a procedure that allows users to maximize the utilization of storage devices while also protecting data integrity on whatever media on which it lives. Network virtualization, replication, mirroring, security, compression, deduplication, traffic analysis, process automation, storage provisioning, and memory management are some of the features that may be included. The operating system is in charge of storing and accessing files. The creation of files, the creation of directories, the reading and writing of data from files and directories, as well as the copying of the contents of files and directories from one location to another are all included in storage management.

➤ OPERATING SYSTEM SERVICES AND INTERFACE

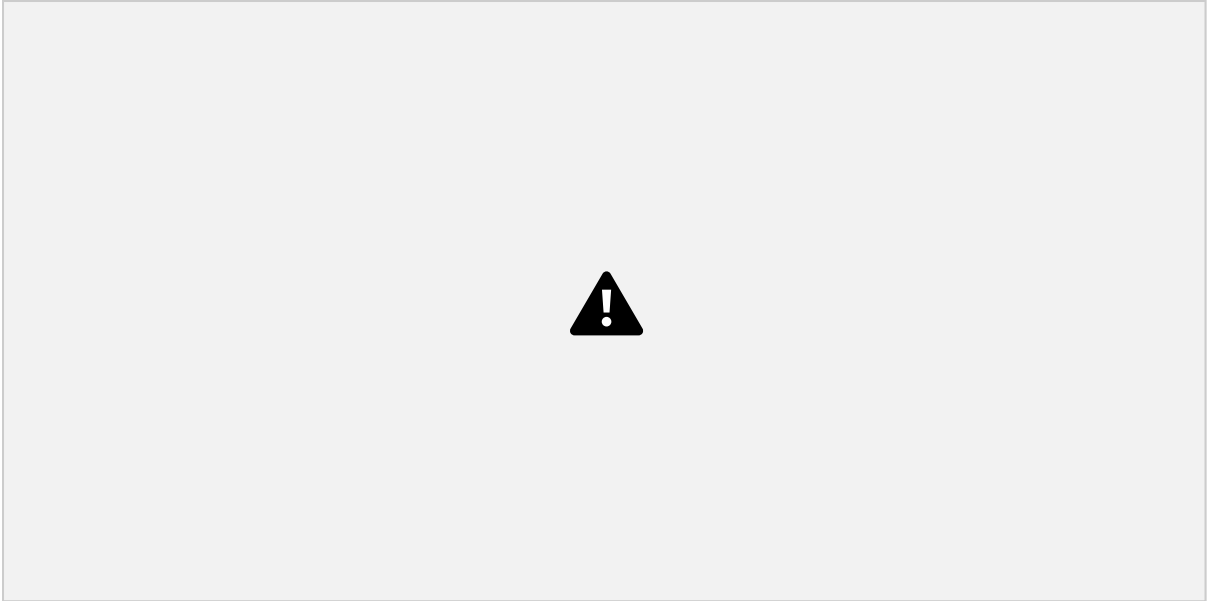


Fig.: A view of an OS Services

1. **User interface:** Almost all operating systems have a user interface (UI). This interface can take several forms. Most commonly, a graphical user interface (GUI) is used. Mobile systems such as phones and tablets provide a touch-screen interface, enabling users to slide their fingers across the screen or press buttons on the screen to select choices. Another option is a command-line interface (CLI), which uses text commands and a method for entering them. Some systems provide two or all three of these variations.
 2. **Program execution:** The system must be able to load a program into memory and to run that program. The program must be able to end its execution, either normally or abnormally (indicating error).
 3. **I/O operations:** A running program may require I/O, which may involve a file or an I/O device. For specific devices, special functions may be desired (such as reading from a network interface or writing to a file system). For efficiency and protection, users usually cannot control I/O devices directly. Therefore, the operating system must provide a means to do I/O.
- 9
4. **File-system manipulation:** Programs need to read and write files and directories. They also need to create and delete them by name, search for a given file, and list file information. Some operating systems include permissions management to allow or deny access to files or directories based on file ownership. Many operating systems provide a variety of file systems, sometimes to allow personal choice and sometimes to provide specific features or performance characteristics.
 5. **Communications:** There are many circumstances in which one process needs to exchange information with another process. Such communication may occur between processes that are executing on the same computer or between processes that are executing on different

computer systems tied together by a network. Communications may be implemented via shared memory, in which two or more processes read and write to a shared section of memory, or message passing, in which packets of information in predefined formats are moved between processes by the operating system.

6. **Error detection:** The operating system needs to be detecting and correcting errors constantly. Errors may occur in the CPU, memory, in I/O devices and in the user program. For each type of error, the operating system should take the appropriate action to ensure correct and consistent computing. Sometimes, it has no choice but to halt the system. At other times, it might terminate an error-causing process or return an error code to a process for the process to detect and possibly correct.

➤ INTERFACE OF AN OS

1. Command Line Interpreter:

Most operating systems, including Linux, UNIX, and Windows, treat the command interpreter as a special program that is running when a process is initiated or when a user first logs on systems. On systems with multiple command interpreters to choose from, the interpreters are known as shells.

The main function of the command interpreter is to get and execute the next user specified command. Many of the commands given at this level manipulate files: create, delete, list, print, copy, execute, and so on. The various shells available on UNIX systems operate in this way.

2. Graphical User Interface:

A second strategy for interfacing with the operating system is through a user-friendly graphical user interface, or GUI. Here, rather than entering commands directly via a command-line interface, users employ a mouse-based window and-menu system characterized by a desktop metaphor. The user moves the mouse to position its pointer on images, or icons, on the screen (the desktop) that represent programs, files, directories, and system functions. The first GUI appeared on the Xerox Alto computer in 1973. However, graphical interfaces became more widespread with the advent of Apple Macintosh computers in the 1980s.

3. Screen Touch Interface:

Because either a command-line interface or a mouse-and-keyboard system is impractical for most mobile systems, smartphones and handheld tablet computers typically use a touch-screen interface. Here, users interact by making gestures on the touch screen—for example, pressing and swiping fingers across the screen. Although earlier smartphones included a physical keyboard, most smartphones and tablets now simulate a keyboard on the touch screen.

➤ SYSTEM CALLS AND ITS TYPES

A system call is a function that a user program uses to ask the operating system for a particular service. User programmers can communicate with the operating system to request its services using the interface that is created by a system call.

A system calls in computing is the programmatic method by which a computer programme asks the kernel of the operating system it is running on for a service. Programs can interact with the operating system by making a system call. When a computer programme requests something from the kernel of the operating system, it performs a system call.

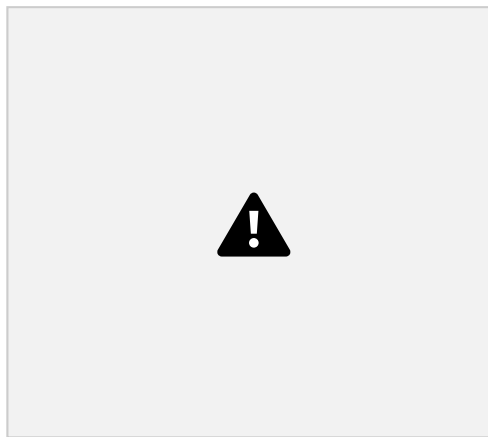


Fig.: Overview of System Call

System call uses Application Programming Interfaces to provide operating system services to user programmes (API). It offers a point of contact between processes and the operating system so that user-level processes can ask for the operating system's assistance. The only ways to access the kernel system are through system calls. System calls are required for all programmes that require resources.

➤ Types of System Call

1. Process control

- create process, terminate process
- load, execute
- get process attributes, set process attributes
- wait event, signal event
- allocate and free memory

2. File management
 - create file, delete file
 - open, close
 - read, write, reposition
 - get file attributes, set file attributes
3. Device management
 - request device, release device
 - read, write, reposition
 - get device attributes, set device attributes
 - logically attach or detach devices
4. Information maintenance
 - get time or date, set time or date
 - get system data, set system data
 - get process, file, or device attributes
 - set process, file, or device attributes
5. Communications
 - create, delete communication connection
 - send, receive messages
 - transfer status information
 - attach or detach remote devices
6. Protection
 - get file permissions
 - set file permissions

➤ SYSTEM PROGRAMS

1. **File Management:** A file is a collection of specific information stored in the memory of a computer system. File management is defined as the process of manipulating files in the computer system, its management includes the process of creating, modifying and deleting files.
- **Status Information:** Information like date, time amount of available memory, or disk space is asked by some users. Others providing detailed performance, logging, and debugging information which is more complex. All this information is formatted

and displayed on output devices or printed. Terminal or other output devices or files or a window of GUI is used for showing the output of programs.

- **File Modification:** For modifying the contents of files we use this. For Files stored on disks or other storage devices, we used different types of editors. For searching

contents of files or perform transformations of files we use special commands.

- **Programming-Language support:** For common programming languages, we use Compilers, Assemblers, Debuggers, and interpreters which are already provided to users. It provides all support to users. We can run any programming language. All languages of importance are already provided.
- **Program Loading and Execution:** When the program is ready after Assembling and compilation, it must be loaded into memory for execution. A loader is part of an operating system that is responsible for loading programs and libraries. It is one of the essential stages for starting a program. Loaders, relocatable loaders, linkage editors, and Overlay loaders are provided by the system.
- **Communications:** Virtual connections among processes, users, and computer systems are provided by programs. Users can send messages to another user on their screen, User can send e-mail, browsing on web pages, remote login, the transformation of files from one user to another.

➤ SYSTEM BOOT

Booting is the process of starting a computer. It can be initiated by hardware such as a button press or by a software command. After it is switched on, a CPU has no software in its main memory, so some processes must load software into memory before execution. This may be done by hardware or firmware in the CPU or by a separate processor in the computer system.

Restarting a computer also is called rebooting, which can be "**hard**", e.g., after electrical power to the CPU is switched from off to on, or "**soft**", where the power is not cut. On some systems, a soft boot may optionally clear RAM to zero. Hard and soft booting can be initiated by hardware such as a button press or a software command. Booting is complete when the operative runtime system, typically the operating system and some applications, is attained.

Types of Booting

1. **Cold Booting:** When the computer starts for the first time or is in a shut-down state and switch on the power button to start the system, this type of process to start the computer is called cold booting. During cold booting, the system will read all the instructions from the

13

ROM (BIOS) and the Operating System will be automatically get loaded into the system. This booting takes more time than Hot or Warm Booting.

2. **Warm Booting:** Warm or Hot Booting process is when computer systems come to no response or hang state, and then the system is allowed to restart during on condition. It is

also referred to as rebooting. There are many reasons for this state, and the only solution is to reboot the computer. Rebooting may be required when we install new software or hardware. The system requires a reboot to set software or hardware configuration changes, or sometimes systems may behave abnormally or may not respond properly. In such a case, the system has to be a force restart.