# Introduction to Operating System

An operating system acts as an intermediary between the user of a computer and computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs in a convenient and efficient manner. .

The operating system as User Interface –

1. User
2. System and application programs
3. Operating system
4. Hardware

The Operating system must support the following tasks. The task are:

1. Provides the facilities to create, modification of programs and data files using an editor.
2. Access to the compiler for translating the user program from high-level language to machine language.
3. Provide a loader program to move the compiled program code to the computer's memory for execution.
4. Provide routines that handle the details of I/O programming.
5. The operating system coordinates the use of the hardware among the various system programs and application programs for various users

**I/O System Management –**

The module that keeps track of the status of devices is called the I/O traffic controller.

The I/O subsystem consists of

- A memory Management component that includes buffering, caching and spooling.
- A general device driver interface.
- Drivers for specific hardware devices.

**Assembler**:-An assembler is a program that converts assembly language into machine code. It takes the basic commands and operations from assembly code and converts them into binary code that can be recognized by a specific type of processor.

**Compiler:-** Compiler transforms code written in a high-level programming language into the machine code, at once, before program runs, whereas an Interpreter coverts each high-level program statement, one by one, into the machine code, during program run.

**Loader:-** The loader is a program that places programs into memory and prepares them for execution i.e. the assembler outputs the machine language translation of a program on a secondary device and a loader places it in the core.

## Important functions of an operating System

**Security –** It uses password protection to protect user data and similar other techniques and prevents unauthorized access to programs and user data.

**Error detecting aids –**
Operating system constantly monitors the system to detect errors and avoid the malfunctioning of computer system.

**Coordination between other software and users –**
It coordinate and assign interpreters, compilers, assemblers and other software to the various users of the computer systems.

**Management**
OS plays a vital role in management of memory, processors, devices, files etc via deciding the order in which allocation of memory, processors should be done and deallocation when a process is either terminated or no more required. It keeps track of where information is stored, user access settings and status of every file also known as file system.

## Types of operating system

| S.no. | Type of O.S. | Definition | Advantages | Disadvantages |
|---|---|---|---|---|
| 1 | Batch O.S. | There is an operator which takes similar jobs having the same requirement and group them into batches without interacting the computer directly | **1.** Multiple users can share the batch systems<br><br>**2.**Idle time for the batch system is very less<br><br>**3.**Easy to manage large work repeatedly in batch systems | **1.**Hard to debug<br><br>**2.**Sometimes costly<br><br>**3.**Other jobs will have to wait for an unknown time if any job fails |
| 2 | Time sharing O.S. | Each task is given some time known as quantum(After this time interval is over OS switches over to the next task.) to execute so that all the tasks work smoothly.These systems are also known as Multitasking Systems. | **1.**Each task gets an equal opportunity<br><br>**2.**Fewer chances of duplication of software<br><br>**3.**CPU idle time can be reduced | **1.**Reliability problem<br><br>**2.**One must have to take care of the security and integrity of user programs and data<br><br>**3.**Data communication problem |

| | | | | |
|---|---|---|---|---|
| 3 | Distributed O.S. | Various autonomous interconnected computers communicate with each other using a shared communication network. Independent systems possess their own memory unit and CPU. These are referred to as “loosely coupled systems” | **1.**Failure of one will not affect the other network communication<br><br>**2.**Highly fast and durable<br><br>**3.**Load on host computer reduces | **1.**Failure of the main network will stop the entire communication<br><br>**2.**Language used to establish this system is not well defined yet<br><br>**3.**Expensive , highly complex and not understood well yet |
| 4 | Network O.S. | These systems run on a server and allow shared access of files, printers, security, applications, and other networking functions over a small private network.These are referred to as tightly coupled systems | **1.**Highly stable centralized servers<br><br>**2.**Security concerns are handled through servers<br><br>**3.**New technologies and hardware up-gradation are easily integrated into the system | **1.**Servers are costly<br><br>**2.**User has to depend on a central location for most operations<br><br>**3.**Maintenance and updates are required regularly |

| | | | | |
|---|---|---|---|---|
| 5 | Real time O.S. | These systems fulfil the need of strict time requirements like missile systems, air traffic control systems, robots, etc. Two types of Real-Time Operating System which are as follows:<br><br>**1.**Hard Real-Time Systems<br><br>**2.**Soft Real-Time Systems: | **1.**Maximum consumption<br><br>**2.**Task shifting<br><br>**3.**Error free | **1.**Limited task<br><br>**2.**Complex Algorithms<br><br>**3.**Use heavy system resources |

# Real Time Systems

**Hard real-time system:-**This type of system can never miss its deadline as it may have disastrous consequences. Example: Flight controller system.

**Soft real-time system:-**This type of system can miss its deadline occasionally with some acceptably low probability as it has no disastrous consequences. Example: Telephone switches.

Terms related to real time system:

**Job –** A job is a small piece of work that can be assigned to a processor and may or may not require resources.

**Task –** A set of related jobs that jointly provide some system functionality.

**Release time of a job –** It is the time at which job becomes ready for execution.

**Execution time of a job –** It is the time taken by job to finish its execution.

**Deadline of a job –** It is the time by which a job should finish its execution. Deadline is of two types: absolute deadline and relative deadline.

**Response time of a job –** It is the length of time from release time of a job to the instant when it finishes.

**Relative Deadline** - Maximum allowable response time of a job.

**Absolute deadline** - Relative deadline + release time.

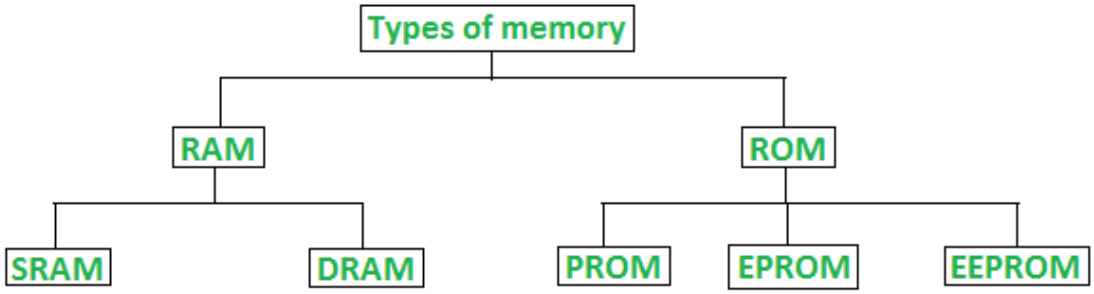
Resources which are essential for execution of a job are called active resources such as processors while which may or may not be required for execution are called passive resources such as mutex.
Two resources are identical if they can be used interchangeably else they are heterogeneous.

There are two types of tasks in real-time systems:
1.Periodic tasks:-jobs are released at regular intervals.
2.Dynamic tasks:-sequential program that is invoked by the occurrence of an event.

## Difference between Multiprogramming, multitasking, multithreading and multiprocessing

| No. | Characteristic | Multi programming | Multi processing | Multithreading | Multi tasking |
|---|---|---|---|---|---|
| 1 | Definition | The concurrent residency of more than one program in the main memory is called as multi programming. | The availability of more than one processor per system, which can execute several set of instructions in parallel is called as multi processing. | A process is divided into several different sub-processes called threads, which has its own path of execution. This concept is called as multi threading. | The execution of more than one task simultaneously is called as multi tasking. |
| 2 | Number of CPU: | One | More than one | Can be one or more than one | One |
| 3 | Job processing time: | More time. | Less time. | Moderate amount of time. | Moderate amount of time. |
| 4 | No.of process being executed: | One process is executed at a time. | More than one process can be executed at a time | Various components of the same process are being executed at a time. | One by one job is being executed at a time. |
| 5 | Throughput: | Less. | Maximum. | Moderate. | Moderate |
| 6 | Efficiency: | Less | Maximum | Moderate | Moderate |

# Memory




**Classification of computer memory**

**RAM :-**

It is also called as *read write memory* or the *main memory* or the *primary memory*.It is a volatile memory as the data loses when the power is turned off.

**Read Only Memory (ROM) –**

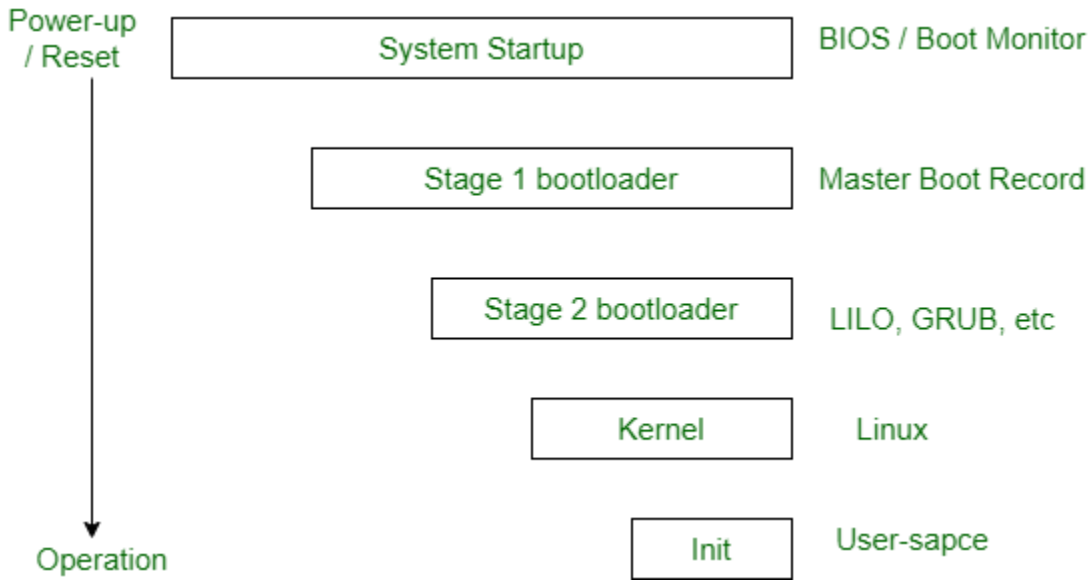
Stores crucial information essential to operate the system, like the program essential to boot the computer.It is not volatile**.**

A major difference between 32-bit processors and 64-bit processors is the number of calculations per second they can perform, which affects the speed at which they can complete tasks. 64-bit processors can come in dual-core, quad-core, six-core, and eight-core versions for home computing. Multiple cores allow for an increased number of calculations per second that can be performed, which can increase the processing power and help make a computer run faster.

# What happens when we turn on computer?

The first thing a computer has to do when it is turned on is to start up a special program called an operating system.

**An overview of the boot process**




The BIOS (Basic Input Output System) chip tells it to look in a fixed place. The **POST** (Power On Self Test) first checks the bios and then tests the CMOS RAM. If there is no problem, then continues to check the CPU, hardware devices. If some errors found then an error message is displayed on the screen or a number of beeps are heard known as POST beep codes.

## Boot Block in Operating System

On most of the computer systems, a small piece of code known as bootstrap program locates the kernel, loads it into main memory and starts its execution and this program is stored in ROM because it doesn't require initialization and as the location is fixed so the processor can start executing when powered up or reset. Also, ROM is basically read-only memory and hence it cannot be affected by the computer virus.
But the problem is that changing the bootstrap code basically requires changes in the ROM hardware chips and hence nowadays there is a tiny bootstrap loader program in the boot whose only job is to bring the full bootstrap program from the disk through

which we can easily change the full bootstrap program and the new version can be easily written onto the disk.

The full bootstrap program is stored in the **boot blocks** at a fixed location on the disk which has a boot partition called boot disk. The code in the boot ROM instructs the read controller to read the boot blocks into the memory and then starts the execution of code. The full bootstrap program is basically able to load the complete OS from a non-fixed location on disk to start the os running and even though it is very small.

## UEFI(Unified Extensible Firmware Interface)

UEFI like BIOS is a firmware that runs when the computer is booted. It initializes the hardware and loads the os into the memory. However, being the more modern solution and overcoming various limitations of BIOS, UEFI is all set to replace the former.

### Limitations of BIOS

- Can boot from drives of less than 2 TB. 3+ TB drives are now standard, and a system with a BIOS can’t boot from them.
- Runs in 16-bit processor mode, and has only 1 MB space to execute.
- Can’t initialize multiple hardware devices at once, thus leading to slow booting process.

### Advantages of UEFI over BIOS

- Breaking Out Of Size Limitations
- Speed and performance : UEFI can run in 32-bit or 64-bit mode and has more addressable address space than BIOS.
- More User-Friendly Interface
- Security: It allows only authentic drivers and services to load at boot time, to make sure that no malware can be loaded at computer startup.

# System Structure

**Kernel:-**

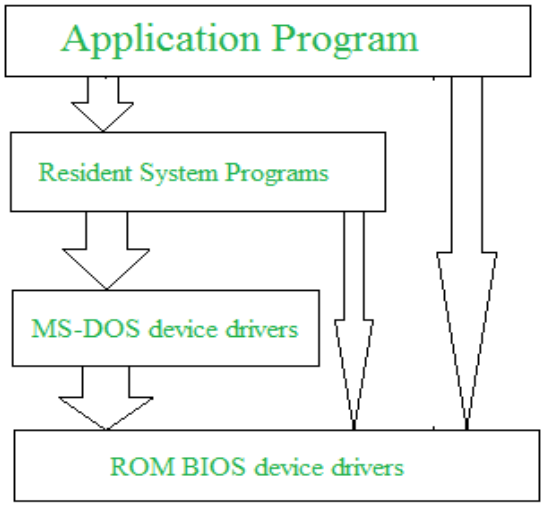
A Kernel is a computer program that is the heart and core of an Operating System. Whenever a system starts, the Kernel is the first program that is loaded after the bootloader because the Kernel has to handle the rest of the system for the Operating System. The Kernel remains in the memory until the Operating System is shut-down.

The Kernel is responsible for low-level tasks such as disk management, memory management, task management, etc.

## Types of Operating System:-

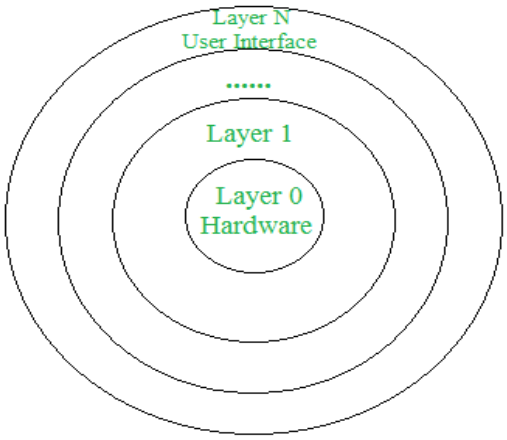
### Simple structure:

Such operating systems do not have well defined structure and are small, simple and limited systems. The interfaces and levels of functionality are not well separated. MS-DOS is an example of such an operating system. MS-DOS application programs are able to access the basic I/O routines. These types of operating systems cause the entire system to crash if one of the user programs fails.



**Layered structure:**

An OS can be broken into pieces and retain much more control on the system. In this structure the OS is broken into a no.of layers (levels). The bottom layer (layer 0) is the hardware and the topmost layer (layer N) is the user interface.These layers are so designed that each layer uses the functions of the lower level layers only.




**Advantage:**

This simplifies the debugging process as if lower level layers are debugged and an error occurs during debugging then the error must be on that layer only as the lower level layers have already been debugged.

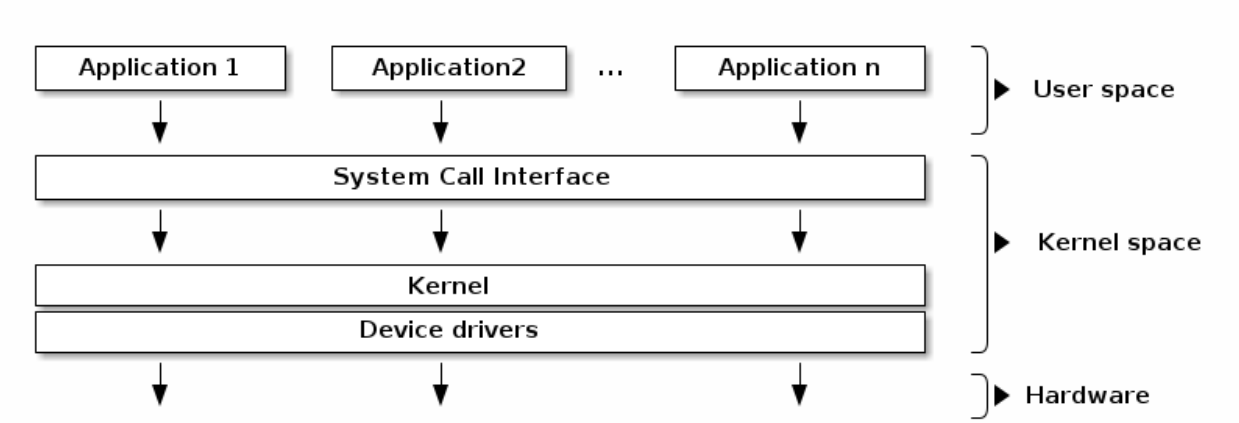
**Disadvantage:**

The main disadvantage of this structure is that at each layer, the data needs to be modified and passed on which adds overhead to the system. Moreover careful planning of the layers is necessary as a layer can use only lower level layers.

ex.UNIX.

**Monolithic Kernel:**

This kernel provides CPU scheduling, memory management, file management and other operating system functions through system calls. As both services are implemented under the same address space, this makes operating system execution faster.




**Advantage:**

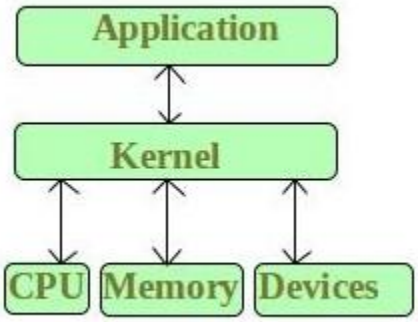
1. It provides CPU scheduling, memory management, file management and other operating system functions through system calls.

**Disadvantages:**

1. If anyone service fails it leads to entire system failure**.**

## Micro-kernel:

This structure designs the operating system by removing all non-essential components from the kernel and implementing them as system and user programs. This results in a smaller kernel called the micro-kernel.




**Advantages:**
1. The architecture of this kernel is small and isolated hence it can function better.
2. Expansion of the system is easier, it is simply added in the system application without disturbing the kernel.

**Disadvantages:**
1.Inter process-Communication makes the process completion slow.