**UNIT-1**

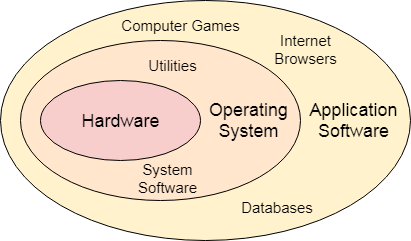
**INTRODUCTION OF OPERATING SYSTEM**

An Operating System acts as a communication bridge (interface) between the user and computer hardware. The purpose of an operating system is to provide a platform on which a user can execute programs in a convenient and efficient manner.

An operating system is a piece of software that manages the allocation of computer hardware. The coordination of the hardware must be appropriate to ensure the correct working of the computer system and to prevent user programs from interfering with the proper working of the system.   
Example: Just like a boss gives orders to his employee, in a similar way we request or pass our orders to the Operating System. The main goal of the Operating System is to make the computer environment more convenient to use and Secondary goal is to use the resources in the most efficient manner.

In the Computer System (comprises of Hardware and software), Hardware can only understand machine code (in the form of 0 and 1) which doesn't make any sense to a naive user.

We need a system which can act as an intermediary and manage all the processes and resources present in the system.



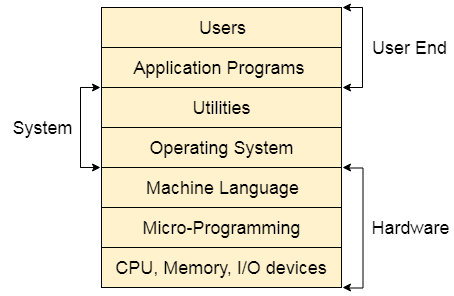
An **Operating System** can be defined as an **interface between user and hardware**. It is responsible for the execution of all the processes, Resource Allocation, [CPU](https://www.javatpoint.com/cpu-full-form) management, File Management and many other tasks.

The purpose of an operating system is to provide an environment in which a user can execute programs in convenient and efficient manner.

* **Structure of a Computer System**

A Computer System consists of:

* Users-(people who are using the computer)
* Application Programs (Compilers, Databases, Games, Video player, Browsers, etc.)
* System Programs (Shells, Editors, Compilers, etc.)
* Operating System ( A special program which acts as an interface between user and hardware )
* Hardware ( CPU, Disks, Memory, etc)



**Important functions of an Operating System:**

1. **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.
2. **Control over system performance –**   
   Monitors overall system health to help improve performance. records the response time between service requests and system response to having a complete view of the system health. This can help improve performance by providing important information needed to troubleshoot problems.
3. **Job accounting –**   
   Operating system Keeps track of time and resources used by various tasks and users, this information can be used to track resource usage for a particular user or group of users.
4. **Error detecting aids –**   
   The operating system constantly monitors the system to detect errors and avoid the malfunctioning of a computer system.
5. **Coordination between other software and users** –   
   Operating systems also coordinate and assign interpreters, compilers, assemblers, and other software to the various users of the computer systems.
6. **Memory Management –**   
   The operating system manages the Primary Memory or Main Memory. Main memory is made up of a large array of bytes or words where each byte or word is assigned a certain address. Main memory is fast storage and it can be accessed directly by the CPU. For a program to be executed, it should be first loaded in the main memory. An Operating System performs the following activities for memory management:

It keeps track of primary memory, i.e., which bytes of memory are used by which user program. The memory addresses that have already been allocated and the memory addresses of the memory that has not yet been used. In multiprogramming, the OS decides the order in which processes are granted access to memory, and for how long. It Allocates the memory to a process when the process requests it and deallocates the memory when the process has terminated or is performing an I/O operation. 

1. **Processor Management –**In a multi-programming environment, the OS decides the order in which processes have access to the processor, and how much processing time each process has. This function of OS is called process scheduling. An Operating System performs the following activities for processor management.

Keeps track of the status of processes. The program which performs this task is known as a traffic controller. Allocates the CPU that is a processor to a process. De-allocates processor when a process is no more required. 

1. **Device Management –**   
   An OS manages device communication via their respective drivers. It performs the following activities for device management. Keeps track of all devices connected to the system. designates a program responsible for every device known as the Input/Output controller. Decides which process gets access to a certain device and for how long. Allocates devices in an effective and efficient way. Deallocates devices when they are no longer required.
2. **File Management –**  
   A file system is organized into directories for efficient or easy navigation and usage. These directories may contain other directories and other files. An Operating System carries out the following file management activities. It keeps track of where information is stored, user access settings and status of every file, and more… These facilities are collectively known as the file system.

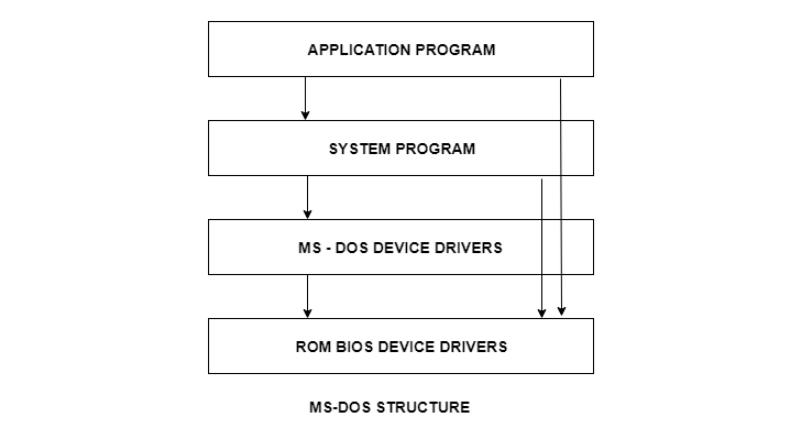
* **Operating System Structure**

An operating system is a construct that allows the user application programs to interact with the system hardware. Since the operating system is such a complex structure, it should be created with utmost care so it can be used and modified easily. An easy way to do this is to create the operating system in parts. Each of these parts should be well defined with clear inputs, outputs and functions.

## **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. There was no indication that it would become so popular.

An image to illustrate the structure of MS-DOS is as follows −

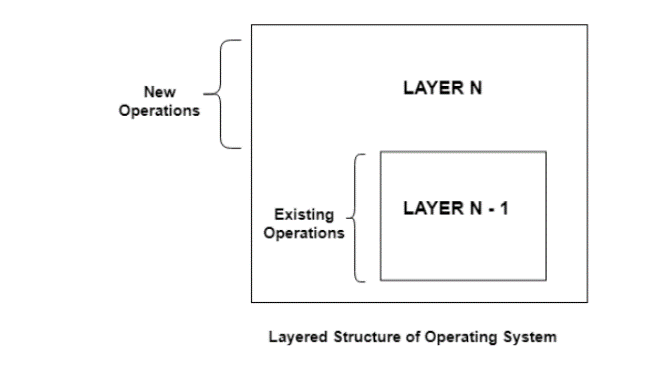


It is better that operating systems have a modular structure, unlike MS-DOS. That would lead to greater control over the computer system and its various applications. The modular structure would also allow the programmers to hide information as required and implement internal routines as they see fit without changing the outer specifications.

## **Layered Structure**

One way to achieve modularity in the operating system is the layered approach. In this, the bottom layer is the hardware and the topmost layer is the user interface.

An image demonstrating the layered approach is as follows −



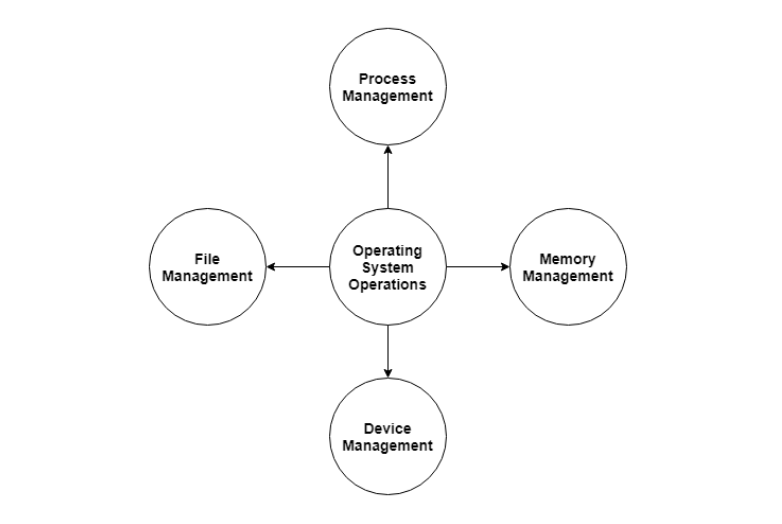
As seen from the image, each upper layer is built on the bottom layer. All the layers hide some structures, operations etc from their upper layers.

One problem with the layered structure is that each layer needs to be carefully defined. This is necessary because the upper layers can only use the functionalities of the layers below them.

# Operating System Operations

An operating system is a construct that allows the user application programs to interact with the system hardware. Operating system by itself does not provide any function but it provides an atmosphere in which different applications and programs can do useful work.

The major operations of the operating system are process management, memory management, device management and file management. These are given in detail as follows:



## **Process Management**

The operating system is responsible for managing the processes i.e assigning the processor to a process at a time. This is known as process scheduling. The different algorithms used for process scheduling are FCFS (first come first served), SJF (shortest job first), priority scheduling, round robin scheduling etc.

There are many scheduling queues that are used to handle processes in process management. When the processes enter the system, they are put into the job queue. The processes that are ready to execute in the main memory are kept in the ready queue. The processes that are waiting for the I/O device are kept in the device queue.

## **Memory Management**

Memory management plays an important part in operating system. It deals with memory and the moving of processes from disk to primary memory for execution and back again.

The activities performed by the operating system for memory management are −

* The operating system assigns memory to the processes as required. This can be done using best fit, first fit and worst fit algorithms.
* All the memory is tracked by the operating system i.e. it nodes what memory parts are in use by the processes and which are empty.
* The operating system deallocated memory from processes as required. This may happen when a process has been terminated or if it no longer needs the memory.

## **Device Management**

There are many I/O devices handled by the operating system such as mouse, keyboard, disk drive etc. There are different device drivers that can be connected to the operating system to handle a specific device. The device controller is an interface between the device and the device driver. The user applications can access all the I/O devices using the device drivers, which are device specific codes.

## **File Management**

Files are used to provide a uniform view of data storage by the operating system. All the files are mapped onto physical devices that are usually non volatile so data is safe in the case of system failure.

The files can be accessed by the system in two ways i.e. sequential access and direct access −

* **Sequential Access**

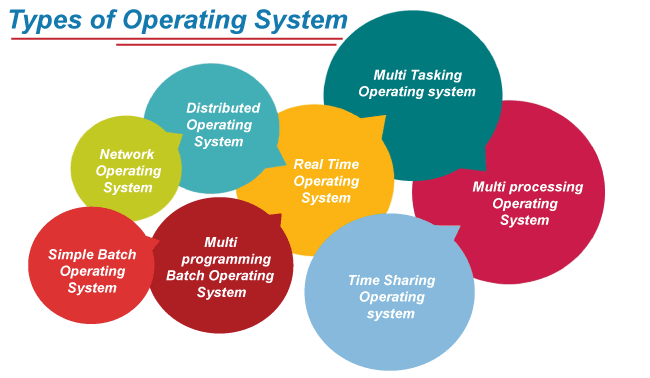
The information in a file is processed in order using sequential access. The files records are accessed on after another. Most of the file systems such as editors, compilers etc. use sequential access.

* **Direct Access**

In direct access or relative access, the files can be accessed in random for read and write operations. The direct access model is based on the disk model of a file, since it allows random accesses.

# Types of Operating Systems

An operating system is a well-organized collection of programs that manages the computer hardware. It is a type of system software that is responsible for the smooth functioning of the computer system.

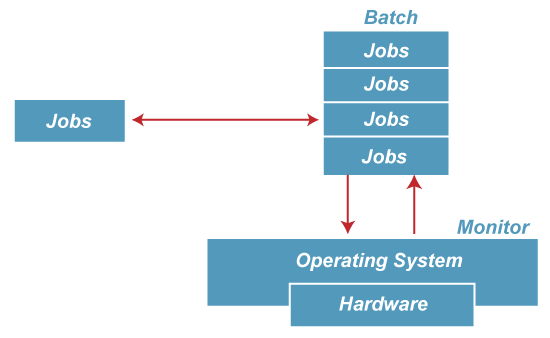


## **1. Batch Operating System**

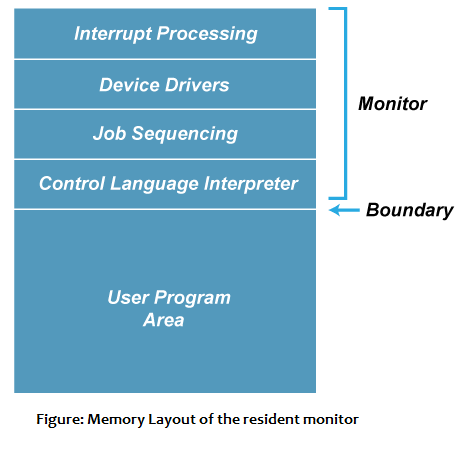
In the 1970s, Batch processing was very popular. In this technique, similar types of jobs were batched together and executed in time. People were used to having a single computer which was called a mainframe.

In Batch operating system, access is given to more than one person; they submit their respective jobs to the system for the execution.

The system put all of the jobs in a queue on the basis of first come first serve and then executes the jobs one by one. The users collect their respective output when all the jobs get executed.



The purpose of this operating system was mainly to transfer control from one job to another as soon as the job was completed. It contained a small set of programs called the resident monitor that always resided in one part of the main memory. The remaining part is used for servicing jobs.



### **Advantages of Batch OS**

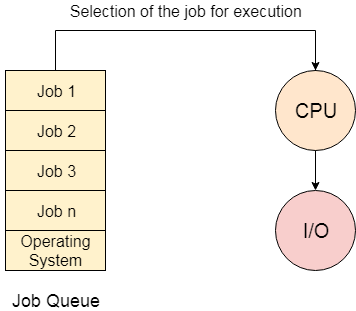
* The use of a resident monitor improves computer efficiency as it eliminates CPU time between two jobs.

### **Disadvantages of Batch OS**

**1. Starvation**

Batch processing suffers from starvation.

**For Example:**



There are five jobs J1, J2, J3, J4, and J5, present in the batch. If the execution time of J1 is very high, then the other four jobs will never be executed, or they will have to wait for a very long time. Hence the other processes get starved.

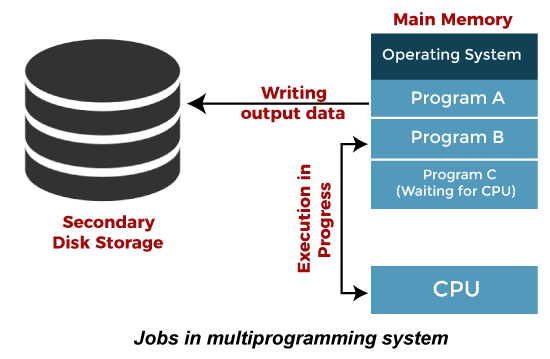
**2. Not Interactive**

Batch Processing is not suitable for jobs that are dependent on the user's input. If a job requires the input of two numbers from the console, then it will never get it in the batch processing scenario since the user is not present at the time of execution.

## **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.

In a multiprogramming environment, when a process does its I/O, The CPU can start the execution of other processes. Therefore, multiprogramming improves the efficiency of the system.



### **Advantages of Multiprogramming OS**

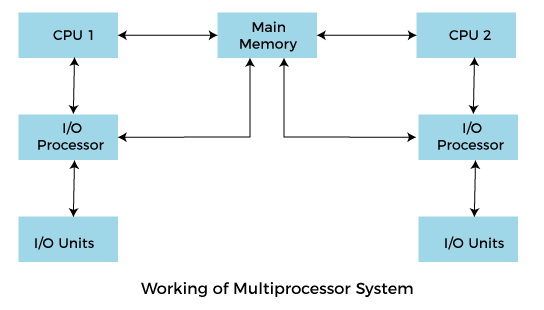
* Throughout the system, it increased as the CPU always had one program to execute.
* Response time can also be reduced.

### **Disadvantages of Multiprogramming OS**

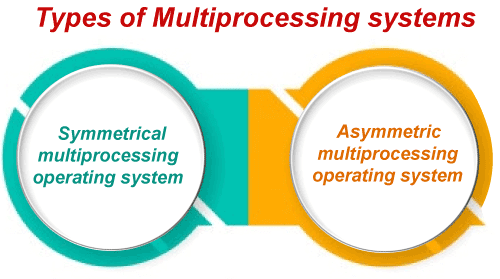
* Multiprogramming systems provide an environment in which various systems resources are used efficiently, but they do not provide any user interaction with the computer system.

## **Multiprocessing Operating System**

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



In Multiprocessing, Parallel computing is achieved. More than one processor present in the system can execute more than one process simultaneously, which will increase the throughput of the system.



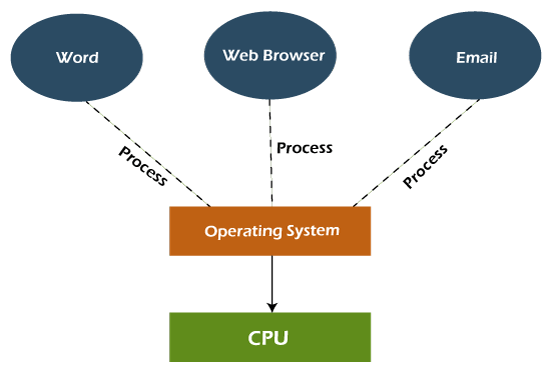
Advantages of Multiprocessing operating system:

* **Increased reliability:** Due to the multiprocessing system, processing tasks can be distributed among several processors. This increases reliability as if one processor fails, the task can be given to another processor for completion.
* **Increased throughout:** As several processors increase, more work can be done in less.

Disadvantages of Multiprocessing operating System

* Multiprocessing operating system is more complex and sophisticated as it takes care of multiple CPUs simultaneously.

### **Multitasking Operating System**

The multitasking operating system is a logical extension of a multiprogramming system that enables **multiple** programs simultaneously. It allows a user to perform more than one computer task at the same time.

### **Advantages of Multitasking operating system**

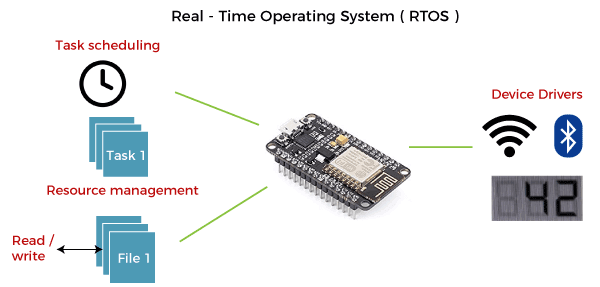
* This operating system is more suited to supporting multiple users simultaneously.
* The multitasking operating systems have well-defined memory management.

### **Disadvantages of Multitasking operating system**

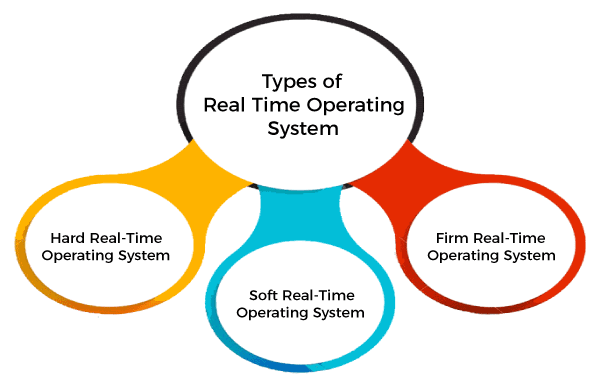
* The multiple processors are busier at the same time to complete any task in a multitasking environment, so the CPU generates more heat.

## **Real Time Operating System**

In Real-Time Systems, each job carries a certain deadline within which the job is supposed to be completed, otherwise, the huge loss will be there, or even if the result is produced, it will be completely useless.



The Application of a Real-Time system exists in the case of military applications, if you want to drop a missile, then the missile is supposed to be dropped with a certain precision.

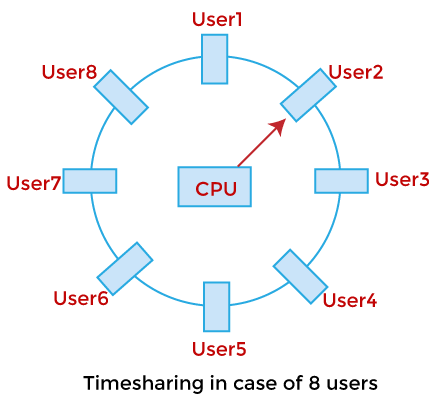


### **Advantages of Real-time operating system:**

* Easy to layout, develop and execute real-time applications under the real-time operating system.
* In a Real-time operating system, the maximum utilization of devices and systems.

## **Time-Sharing Operating System**

In the Time Sharing operating system, computer resources are allocated in a time-dependent fashion to several programs simultaneously. Thus it helps to provide a large number of user's direct access to the main computer. It is a logical extension of multiprogramming. In time-sharing, the CPU is switched among multiple programs given by different users on a scheduled basis.



A time-sharing operating system allows many users to be served simultaneously, so sophisticated CPU scheduling schemes and Input/output management are required.

Time-sharing operating systems are very difficult and expensive to build.

### **Advantages of Time Sharing Operating System**

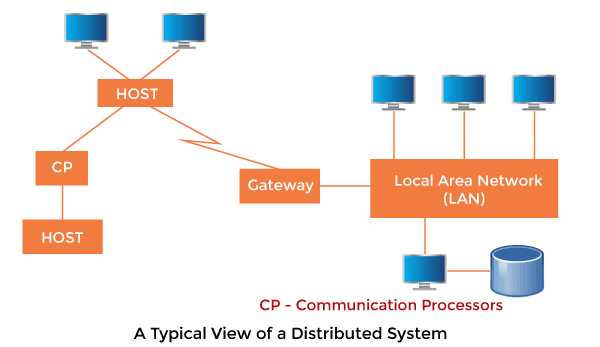
* The time-sharing operating system provides effective utilization and sharing of resources.
* This system reduces CPU idle and response time.

### **Disadvantages of Time Sharing Operating System**

* Data transmission rates are very high in comparison to other methods.
* Security and integrity of user programs loaded in memory and data need to be maintained as many users access the system at the same time.

## **Distributed Operating System**

The Distributed Operating system is not installed on a single machine, it is divided into parts, and these parts are loaded on different machines. A part of the distributed Operating system is installed on each machine to make their communication possible. Distributed Operating systems are much more complex, large, and sophisticated than Network operating systems because they also have to take care of varying networking protocols.



### **Advantages of Distributed Operating System**

* The distributed operating system provides sharing of resources.
* This type of system is fault-tolerant.

### **Disadvantages of Distributed Operating System**

* Protocol overhead can dominate computation cost
* **Operating System Services-**

Moreover, Operating System also provides certain services to the computer system in one form or the other.   
The Operating System provides certain services to the users which can be listed in the following manner:

* **Program Execution:** The Operating System is responsible for the execution of all types of programs whether it be user programs or system programs. The Operating System utilizes various resources available for the efficient running of all types of functionalities.
* **Handling Input/Output Operations**: The Operating System is responsible for handling all sorts of inputs, i.e, from the keyboard, mouse, desktop, etc. The Operating System does all interfacing in the most appropriate manner regarding all kinds of Inputs and Outputs.   
  For example, there is a difference in the nature of all types of peripheral devices such as mice or keyboards, the Operating System is responsible for handling data between them.
* **Manipulation of File System:** The Operating System is responsible for making decisions regarding the storage of all types of data or files, i.e, floppy disk/hard disk/pen drive, etc. The Operating System decides how the data should be manipulated and stored.
* **Error Detection and Handling**: The Operating System is responsible for the detection of any type of error or bugs that can occur while any task. The well-secured OS sometimes also acts as a countermeasure for preventing any sort of breach to the Computer System from any external source and probably handling them.
* **Resource Allocation:** The Operating System ensures the proper use of all the resources available by deciding which resource to be used by whom for how much time. All the decisions are taken by the Operating System.
* **Accounting:** The Operating System tracks an account of all the functionalities taking place in the computer system at a time. All the details such as the types of errors that occurred are recorded by the Operating System.
* **Information and Resource Protection:** The Operating System is responsible for using all the information and resources available on the machine in the most protected way. The Operating System must foil an attempt from any external resource to hamper any sort of data or information.
* All these services are ensured by the Operating System for the convenience of the users to make the programming task easier. All different kinds of Operating systems more or less provide the same services.

## **System Call-**

A system call is a method for a computer program to request a service from the kernel of the [operating system](https://www.javatpoint.com/os-tutorial) on which it is running. A system call is a method of interacting with the operating system via programs. A system call is a request from computer software to an operating system's kernel.

The **Application Program Interface (API)** connects the operating system's functions to user programs. It acts as a link between the operating system and a process, allowing user-level programs to request operating system services. The kernel system can only be accessed using system calls. System calls are required for any programs that use resources.and user programs.

Below are some examples of how a system call varies from a user function.

1. A system call function may create and use kernel processes to execute the asynchronous processing.
2. A system call has greater authority than a standard subroutine. A system call with kernel-mode privilege executes in the kernel protection domain.
3. System calls are not permitted to use shared libraries or any symbols that are not present in the kernel protection domain.
4. The code and data for system calls are stored in global kernel memory.

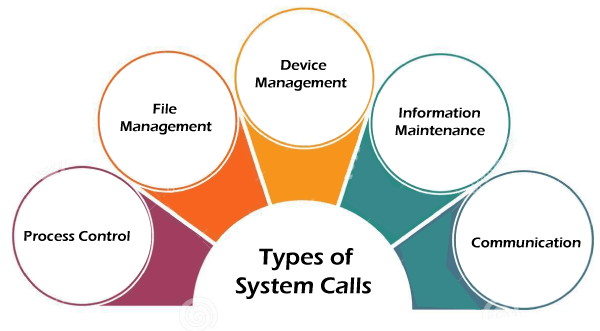
## **Why do you need system calls in Operating System?**

There are various situations where you must require system calls in the operating system. Following of the situations are as follows:

1. It is must require when a file system wants to create or delete a file.
2. Network connections require the system calls to sending and receiving data packets.
3. If you want to read or write a file, you need to system calls.
4. If you want to access hardware devices, including a printer, scanner, you need a system call.
5. System calls are used to create and manage new processes.

## **Types of System Calls**

There are commonly five types of system calls. These are as follows:



### **Process Control**

Process control is the system call that is used to direct the processes. Some process control examples include creating, load, abort, end, execute, process, terminate the process, etc.

### **File Management**

File management is a system call that is used to handle the files. Some file management examples include creating files, delete files, open, close, read, write, etc.

### **Device Management**

Device management is a system call that is used to deal with devices. Some examples of device management include read, device, write, get device attributes, release device, etc.

### **Information Maintenance**

Information maintenance is a system call that is used to maintain information. There are some examples of information maintenance, including getting system data, set time or date, get time or date, set system data, etc.

### **Communication**

Communication is a system call that is used for communication. There are some examples of communication, including create, delete communication connections, send, receive messages, etc.

## **Examples of Windows and UNIX system calls**

There are various examples of Windows and UNIX system calls. These are as listed below in the table:

|  |  |  |
| --- | --- | --- |
| **Process** | **Windows** | **UNIX** |
| **Process Control** | CreateProcess() ExitProcess() WaitForSingleObject() | Fork() Exit() Wait() |
| **File Manipulation** | CreateFile() ReadFile() WriteFile() CloseHandle() | Open() Read() Write() Close() |
| **Device Management** | SetConsoleMode() ReadConsole() WriteConsole() | Ioctl() Read() Write() |
| **Information Maintenance** | GetCurrentProcessID() SetTimer() Sleep() | Getpid() Alarm() Sleep() |
| **Communication** | CreatePipe() CreateFileMapping() MapViewOfFile() | Pipe() Shmget() Mmap() |
| **Protection** | SetFileSecurity() InitializeSecurityDescriptor() SetSecurityDescriptorgroup() | Chmod() Umask() Chown() |

---------------------------END OF THE UNIT-------------------------------