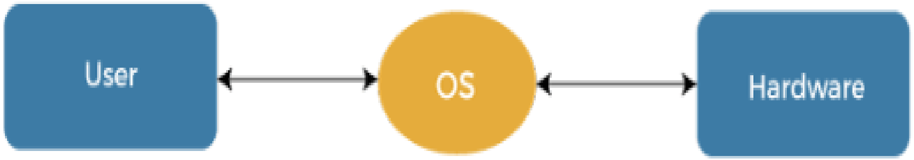
UNIT – I:

Operating System - Introduction, Structures - Simple Batch, Multi programmed, Time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, System components, Operating System services, System Calls.

INTRODUCTION TO OPERATING SYSTEMS

Definition:

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.



Varieties of Operating Systems have emerged over the years having different features and functionalities like Windows, Linux, MacOS, iOS, Android etc.
Every computer system including desktops, laptops, tablets, supercomputers, hand held and even video game consoles use some type of Operating system.

Operating systems are there from the very first computer generation and they keep evolving with time. There are various types of operating systems used today depending on the device, manufacturer, and user preferences.

Purpose:

To provide an environment in which a user can execute programs conveniently and efficiently. It is responsible for the management and coordination of activities and the sharing of the resources of the computer.

OS is a System Software (Tested Program + Documentation).

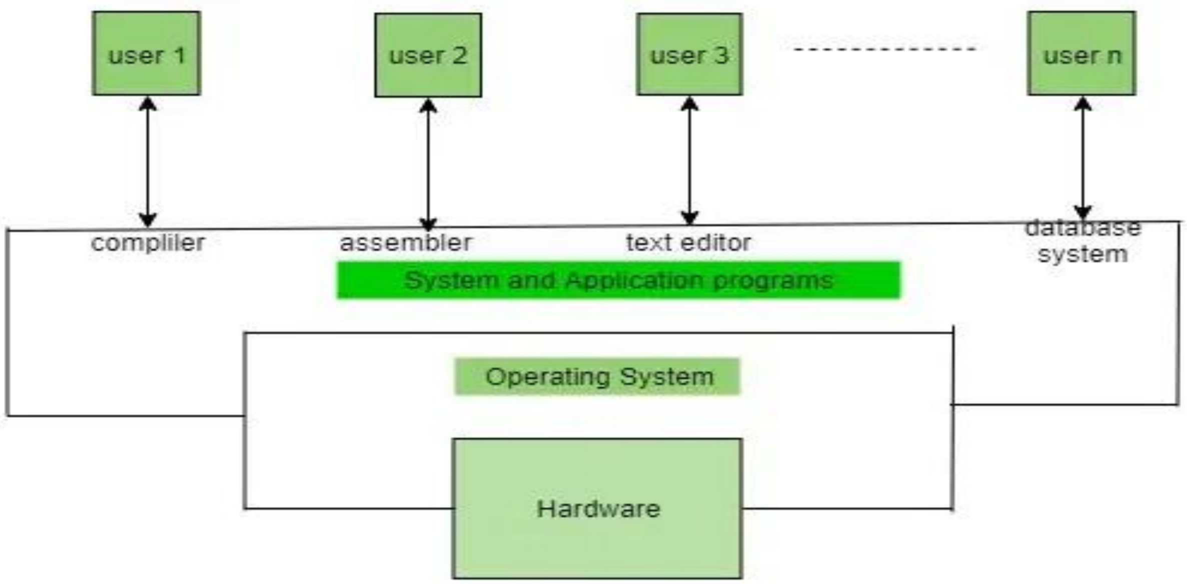
History of Operating system:

- General Motors Research Lab implemented the first OS in 1956 for their IBM 701 (central computer)
- In 1960's IBM started developing their own OS and distributed it.
- In late 1960's, first version of UNIX OS was developed and was written in C language
- In 1981, first OS built by Microsoft was DOS known as MS-DOS.
- In 1985, first version of Windows OS came into existence when a GUI created and paired with MS-DOS.

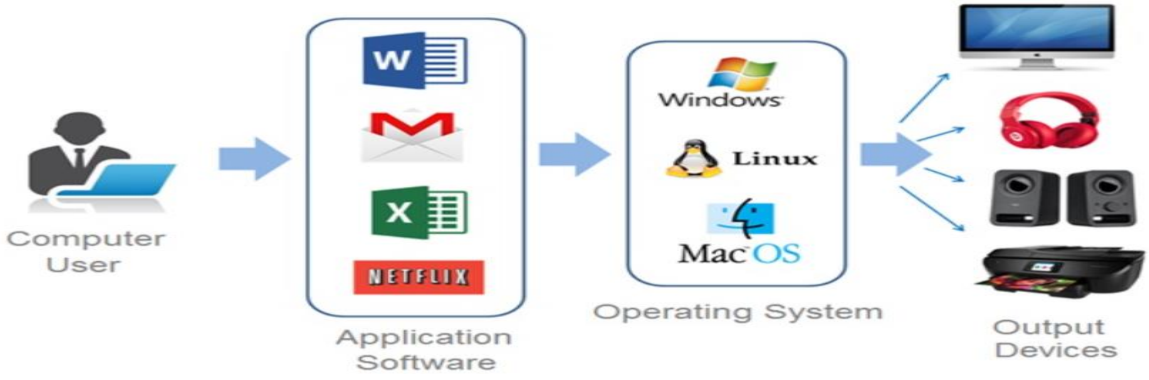
Abstract View (or) Generic Architecture of Components of OS:

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 softwares 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.

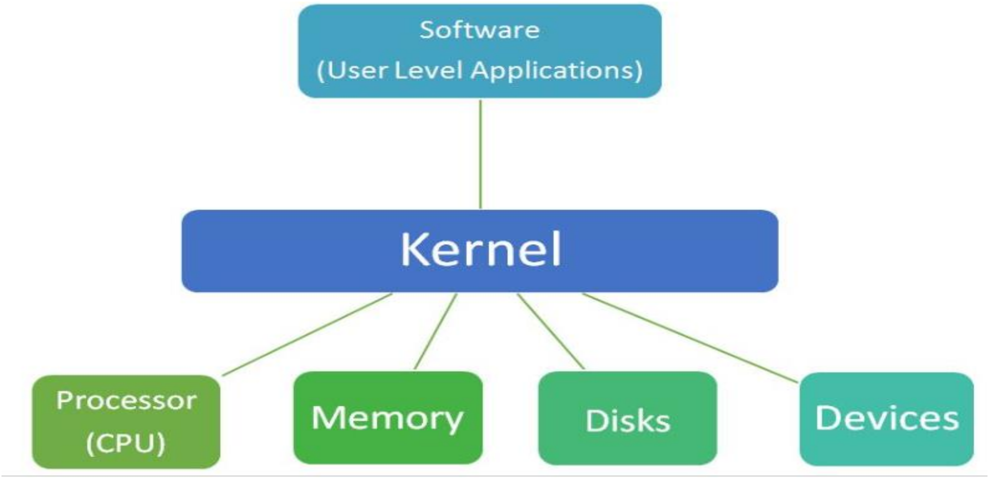


The below figure shows the Realtime examples of the generic components of a Computer System OS.



Kernel:

- It is the first program that is loaded in memory when system bootsup. A kernel is the core component of an operating system that acts as a bridge between the hardware and software.
- It manages system resources such as memory, CPU, and input/output devices, and provides a layer of abstraction between the hardware and higher-level software components.



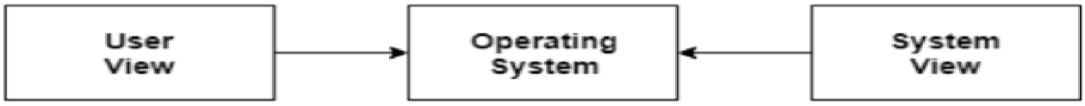
Functions of a Kernel in OS:

- Process Management
- Scheduling Processes
- Resource Allocation
- Device Management
- Memory Management

USER VIEW & SYSTEM VIEW OF OPERATING SYSTEM

An operating system is a construct that allows the user application programs to interact with the system hardware.

The operating system can be observed from the point of view of the user or the system. This is known as the **user view** and the **system view** respectively. More details about these are given as follows:



USER VIEW:

The user view depends on the system interface that is used by the users. The different types of user view experiences can be explained as follows −

- If the user is using a **personal computer**, the operating system is largely designed to make the interaction easy. There is no need for the operating system to worry about resource utilization. This is because the personal computer uses all the resources available.
- If the user is using a system connected to a **mainframe or a minicomputer**, the operating system is largely concerned with resource utilization. This is because there may be multiple terminals connected to the mainframe and the operating system makes sure that all the resources such as CPU, memory, I/O devices etc. are divided uniformly between them.
- If the user is sitting on a **workstation** connected to other workstations through networks, then the operating system needs to focus on both individual usage of resources and sharing though the network.
- If the user is using a **handheld computer such as a mobile**, then the operating system handles the usability of the device including a few remote operations.

SYSTEM VIEW:

The different types of system view for operating system can be explained as follows –

- In system views the operating system as a **resource allocator**. There are many resources such as CPU time, memory space, file storage space, I/O devices etc. that are required by processes for execution.
- The operating system can also work as a **control program**. It manages all the processes and I/O devices so that the computer system works smoothly and there are no errors. It makes sure that the I/O devices work in a proper manner without creating problems.
- Operating systems can also be viewed as a way to make using **hardware easier**.
- Computers were required to easily solve user problems. However, it is not easy to work directly with the computer hardware. So, operating systems were developed to easily communicate with the hardware.
- An operating system can also be considered as a program running at all times in the background of a computer system (known as the kernel) and handling all the application programs.

COMPUTER - SYSTEM ARCHITECTURE

Single Processor Systems:

Most systems use a single processor. On a single processor system, there is one main CPU capable of executing general-purpose instructions set, including instructions from user processes. Almost all systems have other special-purpose processors as well. They may come in the form of device-specific processors, such as disk, keyboard, and graphics controllers; or, on mainframes, they may come in the form of more general-purpose processors, such as I/O processors that move data rapidly among the components of the system.

Multi Processor Systems:

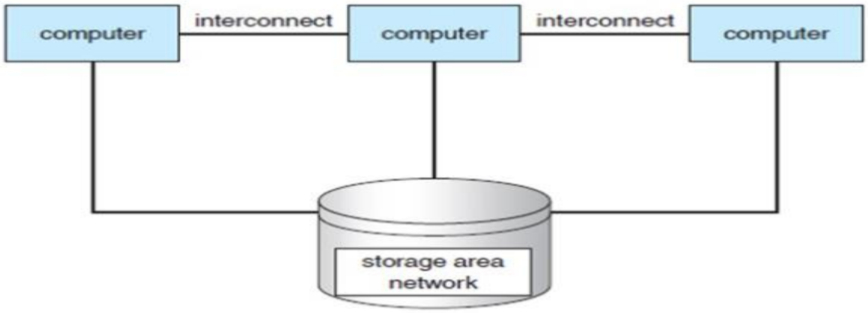
These systems are growing in use and importance. Also known as parallel systems or tightly coupled systems. The System have two or more processors in close communication and sharing the computer bus memory and peripherals devices.

Major Advantages:

1. Increased Throughput
2. Economy of scale
3. Increased reliability

Clustered Systems:

Cluster systems are similar to parallel systems because both systems use multiple CPUs. They have independent computer systems and a shared storage media, and all systems work together to complete all tasks. All cluster nodes use two different approaches to interact with one another, like message passing interface (MPI) and parallel virtual machine (PVM).



The clustered systems are a combination of **hardware clusters** and **software clusters**. Hardware clusters aid in the sharing of high-performance disks among all computer systems, while software clusters give a better environment for all systems to operate.

There are mainly three types of the clustered operating system:

1. Asymmetric Clustering System - One node out of all nodes is in hot standby mode, while the remaining nodes run the essential applications.
2. Symmetric Clustering System - Multiple nodes help run all applications in this system, and it monitors all nodes simultaneously. Because it uses all hardware resources, this cluster system is more reliable.
3. Parallel Cluster System - A parallel cluster system enables several users to access similar data on the same shared storage system. The system is made possible by a particular software version.

PROTECTION AND SECURITY IN OS

Protection – Is any mechanism for controlling access of processes or users to resources defined by the OS.

Protection can improve the reliability by detecting latent errors at the interface. An unprotected resource cannot defend against use by an unauthorized user. So, a protected-oriented system provides a means to distinguish between authorized and unauthorized usage.

Security – Is the defense of system against internal and external attacks. Security protects the information stored in the system (both data and code), as well as the physical resources of the computer system, from unauthorized access, malicious destruction or alteration, and accidental introduction of inconsistency.

Protection and security require that computer resources such as CPU, Software’s, memory etc. are protected.

This extends to the operating system as well as the data in the system. This can be done by ensuring integrity, confidentiality and availability in the operating system. The system must be protected against unauthorized access, viruses, worms etc.

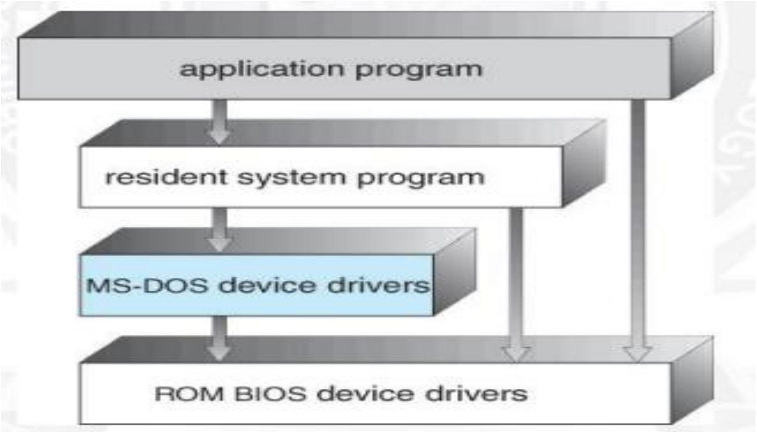
Most Operating systems maintain a list of user names and associated User Identifiers (user IDs). When a user logs in to the system, the authentication stage determines the ID for user.

STRUCTURES (or) APPROACHES OF OPERATING SYSTEMS

The operating systems are large and complex. A common approach is to partition the task into small components, or modules, rather than have one monolithic system.

The structure of an operating system can be defined the following structures –

1. **Simple Structure:**
   - This structure was followed by most of the OS designed long ago. But it lacks definition and is only appropriate for usage with tiny and restricted systems. Since the interfaces and degrees of functionality were not well separated, which may result in unauthorized access to I/O procedures.
   - This organizational structure is used by the **MS-DOS operating system**.

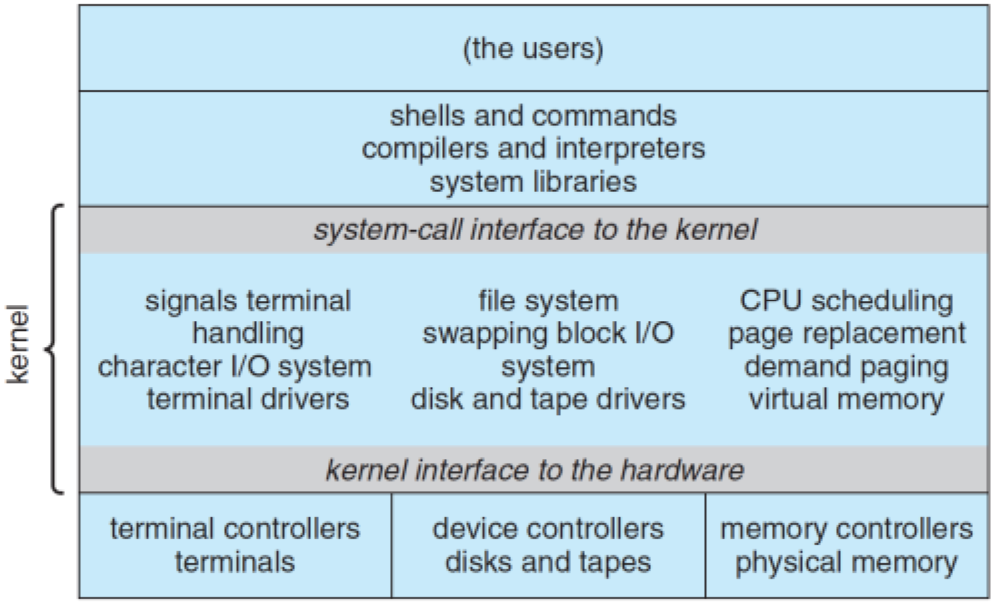


   - These layers include ROM BIOS device drivers, MS-DOS device drivers, application programs, and system programs.
   - Here the base hardware ROM-BIOS controls the system hardware and acts as an interface between OS and system Hardware. Where system BIOS contains in ROM Chip.

- The Pre – installed programs on Windows based computer starts before actual OS begins.
- Application programs and System programs have direct access to BIOS drivers, if any program running in phase fails, it crashes the entire system.

2. Monolithic Structure:

- The Monolithic operating System in which the kernel acts as a manager by managing all things like file management, memory management, device management, and operational processes of the Operating System.
- Every functionality packed into one level in a kernel is known as Monolithic kernel.
- This organizational structure is used by the **UNIX operating system.**

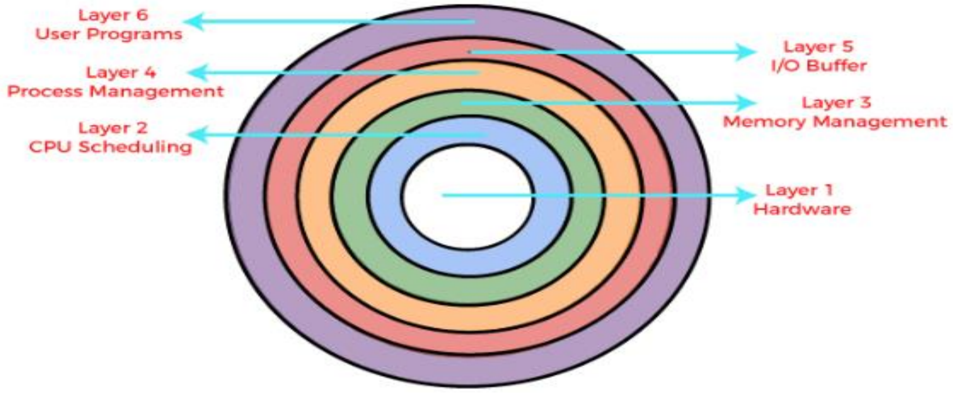


- Where the maintenance of too many functionalities in single kernel is difficult to debug. No independent upgradation of functionalities.
- This is an old operating system that was used in banks to carry out simple tasks like batch processing and time-sharing, which allows numerous users at different terminals to access the Operating System.

3. Layered Structure:

- In this type of structure, OS is divided into layers or levels. The hardware is on the bottom layer (layer 0), while the user interface is on the top layer (layer N). These layers are arranged in a hierarchical way in which the top-level layers use the functionalities of their lower-level levels.

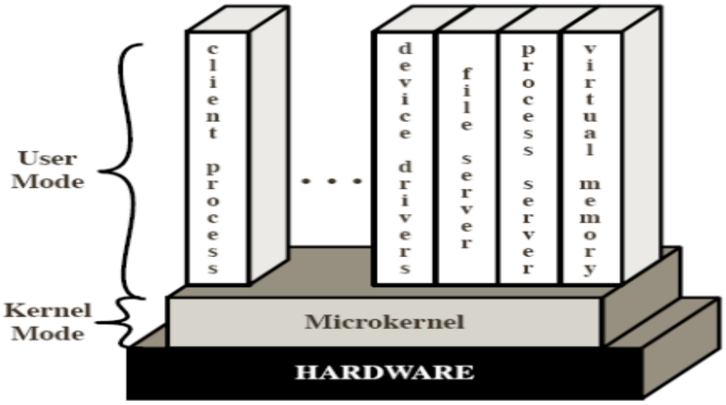
- In this approach, functionalities of each layer are isolated, and abstraction is also available.
- Layering makes it easier to enhance the operating system as implementation of a layer can be changed easily without affecting the other layers.



- Each layer can interact only with layer above or below. Where hardware is protected without direct contact with user interface.
- Debugging is simpler because the lower layers are examined first, followed by the top layers.

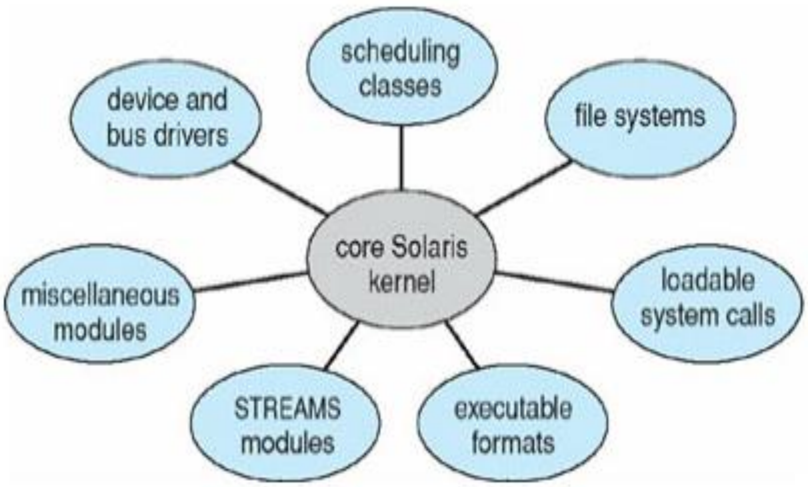
4. Micro Kernel Structure:

- This structure designs the operating system by removing all non-essential components from the kernel and implementing them as system and user programs. This result in a smaller kernel called the micro-kernel.
- Each Micro-Kernel is made independently and is isolated from other Micro-Kernels. So this makes the system more secure and reliable. If any Micro-Kernel fails, then the remaining operating System remains untouched and works fine.
- This organizational structure is used by the **Mac operating system.**



- It provides the communication between client program and other services by using message passing. Where most of the functions will be executed in user mode.
- 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.

5. **Modular Structure:**
   - It is considered as the best approach for an OS. It involves designing of a modular kernel.
   - The kernel has only set of core components and other services are added as dynamically loadable modules to the kernel either during run time or boot time.



   - It resembles layered structure due to the fact that each kernel has defined and protected interfaces but it is more flexible than the layered structure as a module can call any other module.

TYPES OF OPERATING SYSTEMS

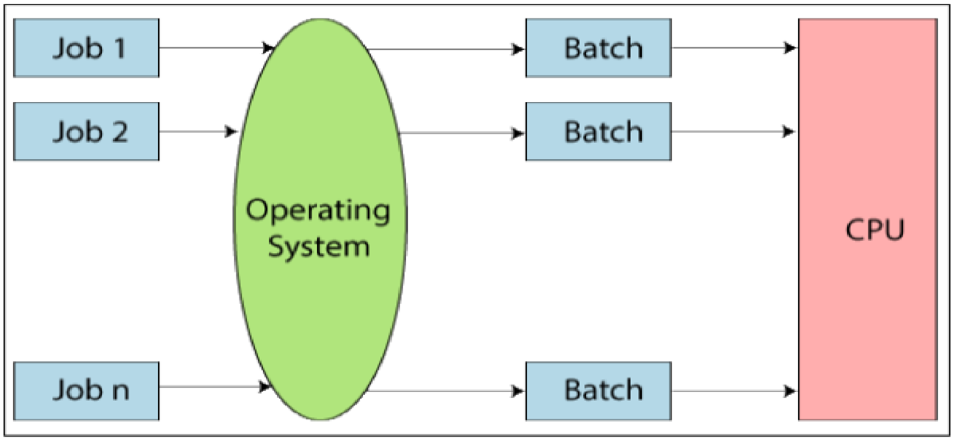
An operating system is a well-organized collection of programs that manages the computer hardware. It is a type of system software that is responsible for the smooth functioning of the computer system.
An Operating System performs all the basic tasks like managing files, processes, and memory.

There are several types of Operating Systems which are mentioned below.

1. **Simple Batch OS:**
   - This type of OS, user does not interact with computer directly. Where the user submits a job to the computer operator.
   - Processes with similar types of jobs form into Batch.

- Batches with similar requirements and when computer available run them batch-wise.
- It is the responsibility of the operator to sort jobs with similar needs.

Example: Bank Statements, Pay rolls, Printings etc…



Advantages of Multi-Programming Operating System

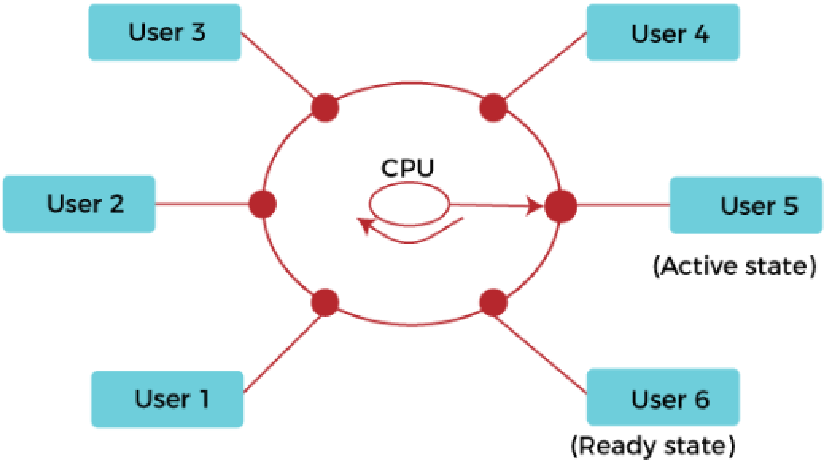
- Multi Programming increases the Throughput of the System.
- It helps in reducing the response time.

Disadvantages of Multi-Programming Operating System

- There is not any facility for user interaction of system resources with the system.
- It is sometimes costly.
- The other jobs will have to wait for an unknown time if any job fails.

2. Time Sharing OS (Multitasking OS):

- Each task is given some time to execute so that all the tasks work smoothly.
- The time taken by each job to execute the job is known as **quantum**.
- After the interval of time is over, the operating system moves to the next task.



- Here the CPU is switched among multiple programs given by different users on a scheduled basis.

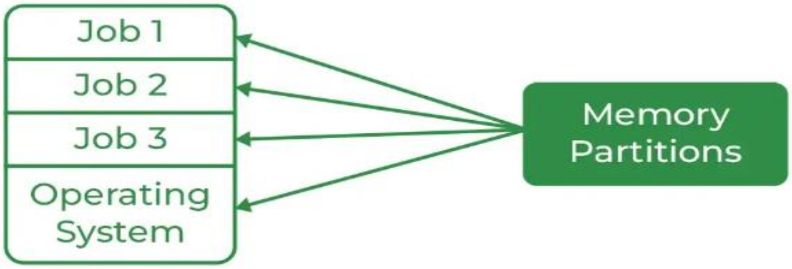
Advantages of Time-Sharing Operating System

- CPU remains idle for less time.
- No partiality occurs between the jobs.
- It quickly responses.

Disadvantages of Time-Sharing Operating System

- The Problem of Data commination.
- Not reliable

3. **Multi Programming OS:**
   - OS provides the ability to run more than one program (job) concurrently in the main memory.
   - Used basically for better execution of programs and CPU utilization.
   - If different jobs are ready to execute at the same time, then the job is selected for CPU Scheduling.
     Example: Modern OS
   - It 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.



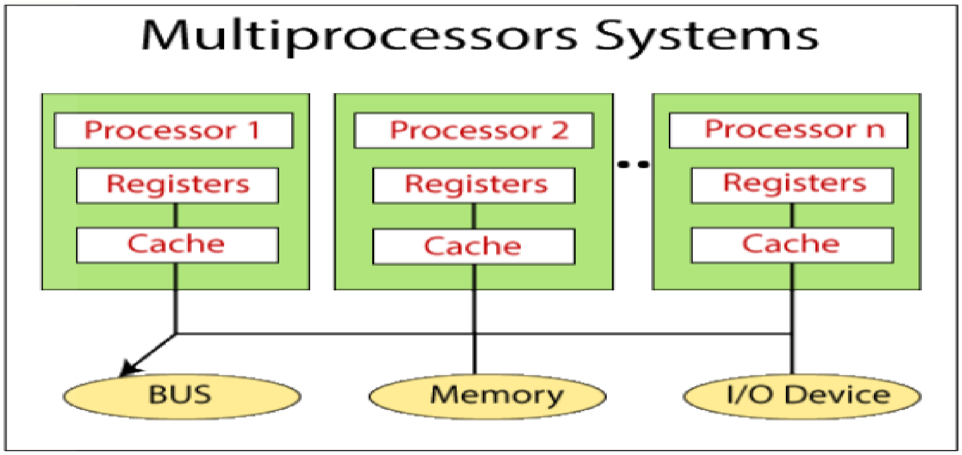
Advantages of Multi-Programming Operating System

- Multi Programming increases the Throughput of the System.
- It helps in reducing the response time.

Disadvantages of Multi-Programming Operating System

- There is not any facility for user interaction of system resources with the system.

4. **Multi-Processor OS (Parallel Processor):**
   - It means the use of two or more processors within a single computer system.
   - These multiple processors are in close communication and share the memory, computer bus, and other peripheral devices.
   - These systems are known as tightly coupled systems.



- This will increase the throughput of the system.

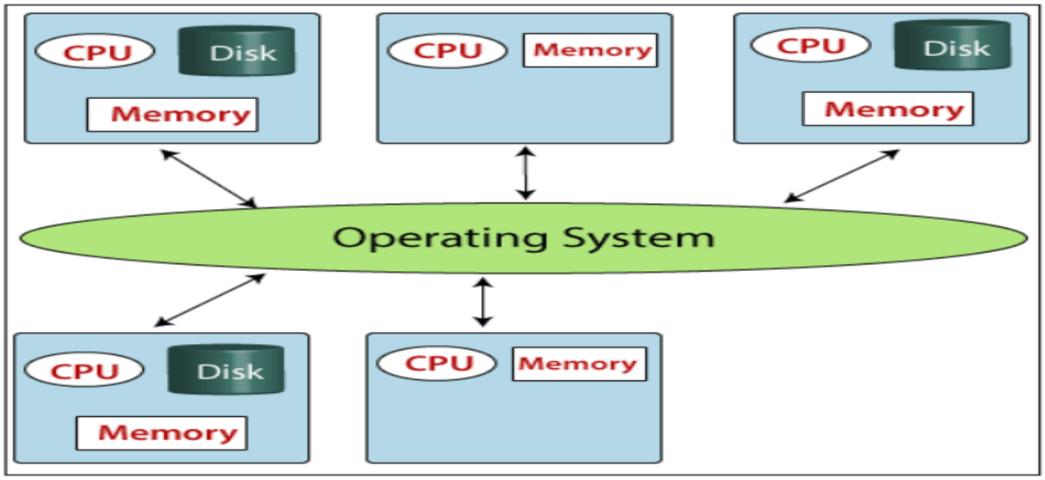
Advantages of Multi-Processing Operating System

- It increases the throughput of the system.
- As it has several processors, so, if one processor fails, we can proceed with another processor.

Disadvantages of Multi-Processing Operating System

- Due to the multiple CPU, it can be more complex and somehow difficult to understand.

5. **Distributed OS:**
   - Distributed systems are also known as loosely coupled systems.
   - Here multiple central processors are used to serve multiple real-time applications and multiple users.
   - Data processing & jobs are distributed among the processors accordingly.
   - The processors communicate with one another through various communication lines (such as high-speed buses or telephone lines).



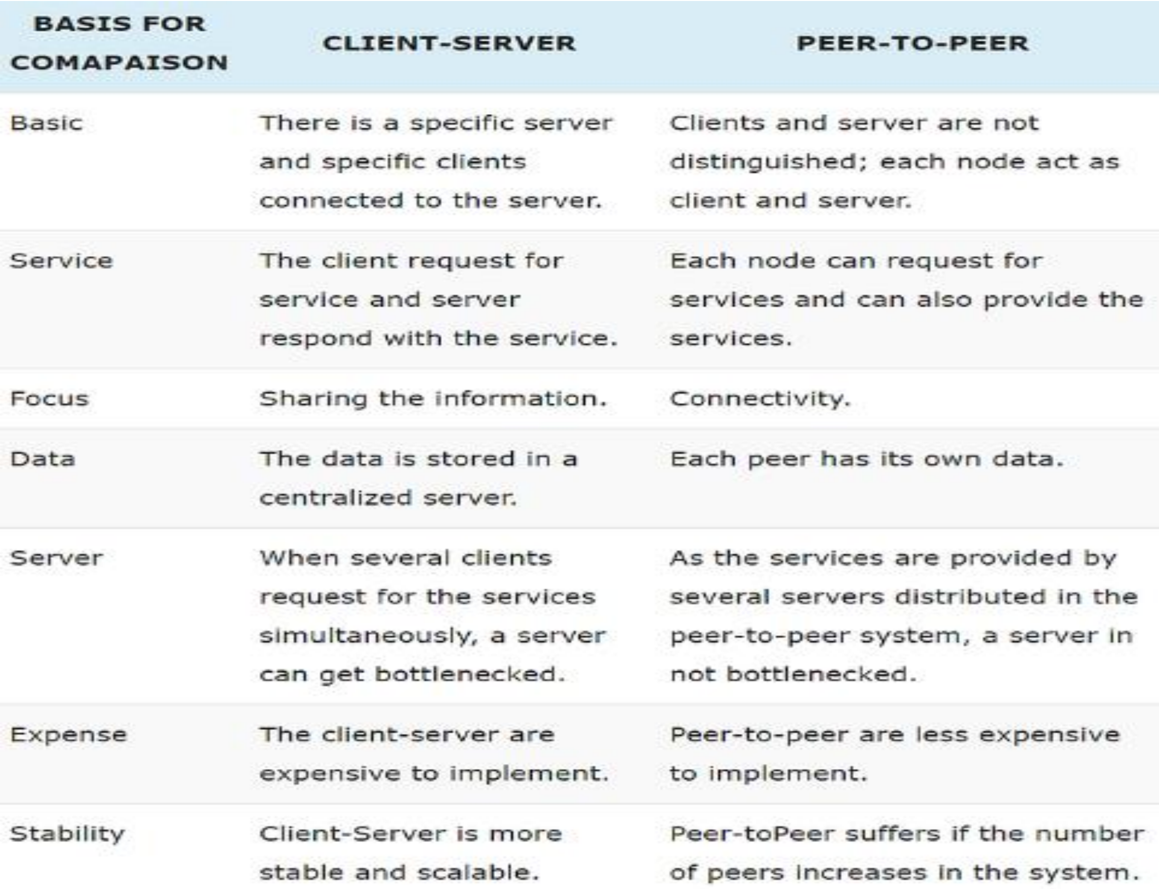
Advantages of Distributed Operating System

- Failure of one will not affect the other network communication, as all systems are independent of each other.
- Since resources are being shared, computation is highly fast and durable.

Disadvantages of Distributed Operating System

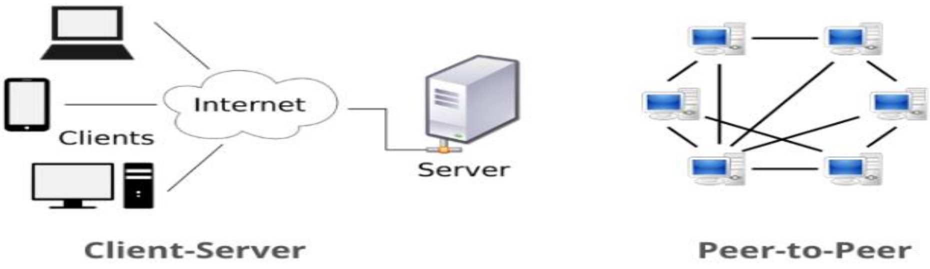
- Failure of the main network will stop the entire communication.
- To establish distributed systems the language is used not well-defined yet.

Difference between Client-Server and Peer-to-Peer Network



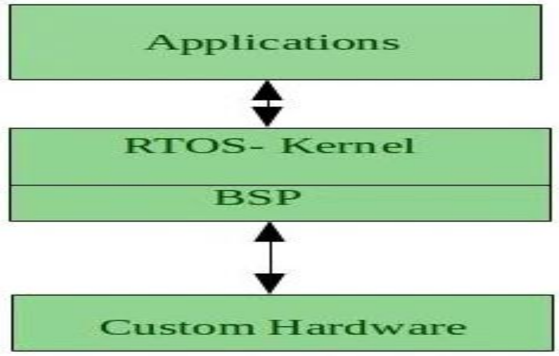
BASIS FOR COMAPAISON	CLIENT-SERVER	PEER-TO-PEER
Basic	There is a specific server and specific clients connected to the server.	Clients and server are not distinguished; each node act as client and server.
Service	The client request for service and server respond with the service.	Each node can request for services and can also provide the services.
Focus	Sharing the information.	Connectivity.
Data	The data is stored in a centralized server.	Each peer has its own data.
Server	When several clients request for the services simultaneously, a server can get bottlenecked.	As the services are provided by several servers distributed in the peer-to-peer system, a server in not bottlenecked.
Expense	The client-server are expensive to implement.	Peer-to-peer are less expensive to implement.
Stability	Client-Server is more stable and scalable.	Peer-toPeer suffers if the number of peers increases in the system.

Structure



6. **Real time OS:**

- It serves real-time systems. The time interval required to process and respond to inputs is very small. This time interval 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.
- It is used when a large number of events are processed in a short interval of time.
- In Real-Time Systems, each job carries a certain deadline within which the job is supposed to be completed.



Types of Real-Time Operating Systems

Hard Real-Time OSs are meant for applications where time constraints are very strict and even the shortest possible delay is not acceptable. Virtual memory is rarely found in these systems.

Soft Real-Time OSs are for applications where time-constraint is less strict.

Advantages of RTOS

- **Maximum Consumption:** Maximum utilization of devices and systems, thus more output from all the resources.
- **Task Shifting:** The time assigned for shifting tasks in these systems is very less.
- Real-time **operating system in** the **embedded system:** Since the size of programs is small, RTOS can also be used in embedded systems like in transport and others.
- **Error Free:** These types of systems are error-free.
- **Memory Allocation:** Memory allocation is best managed in these types of systems.

Disadvantages of RTOS

- **Limited Tasks:** Very few tasks run at the same time and their concentration is very less on a few applications to avoid errors.

- **Use heavy system resources:** Sometimes the system resources are not so good and they are expensive as well.
- **Complex Algorithms:** The algorithms are very complex and difficult for the designer to write on.
- **Thread Priority:** It is not good to set thread priority as these systems are very less prone to switching tasks.

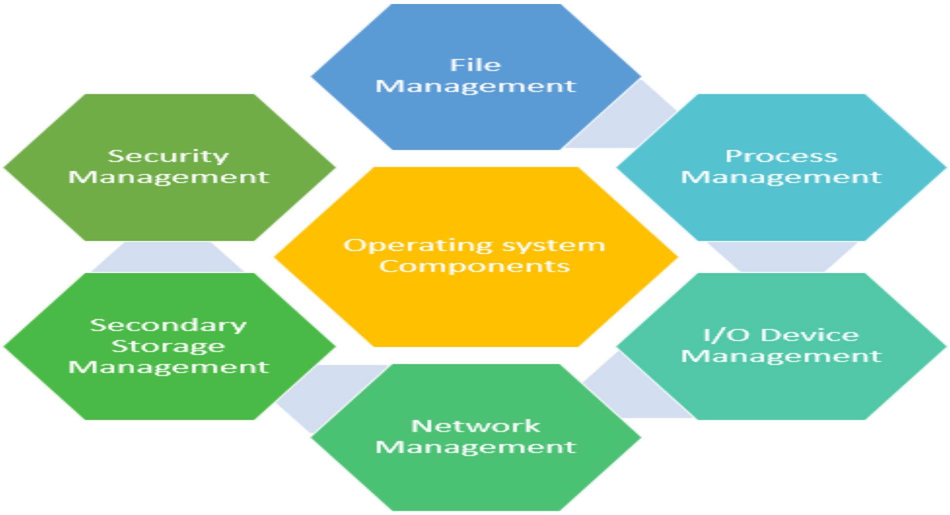
7. **Personal Computing Environment:**

   In the personal computing environment, there is a single computer system. All the system processes are available on the computer and executed there. The different devices that constitute a personal computing environment are laptops, mobiles, printers, computer systems, scanners etc.

   Personal computing is a stand-alone machine. In a personal computing environment, the complete program resides on the stand-alone machine and executed from the same machine.

SYSTEM COMPONENTS (or) OPERATING SYSTEMS FUNCTIONS

The operating system manages resources of system software and computer hardware resources. It allows computing resources to be used in an efficient way. A user can interact with the operating system by making system calls or using OS commands.



1. **Process Management:**

   The process management component is the way to manage the multiple processes running simultaneously on the OS. Process management manages all the running processes and makes sure that all of them run efficiently. It also uses the memory allocated to them and shuts them down when required.

2. **File Management:**

   Files are used for long-term storage. Files are used for both input and output. This file management service can also be treated as an abstraction as it hides the information about the disks from the user. The operating system also provides a system call for file management.

3. I/O Device Management:

I/O device management hides the variations of some hardware devices from the user.

Following are the features of I/O device management:

- Buffers caching system
- Provides general device driver code
- Provides drivers for particular hardware devices.
- Helps know the individualities of a device.

4. Network Management:

Network management is the process of administering and managing computer networks. It includes performance management, fault analysis, provisioning of networks, and maintaining the quality of service**.**

Network management helps connect the network fully or partially so that users can design routing and connection strategies with no connection and security issues.

5. Storage Management:

Programs help access data in the main memory during execution. The main memory is too small and cannot store all the data and programs permanently. Thus, secondary storage acts as a backup to the main memory. Assemblers and compilers are stored on the disk until they are loaded into the memory and use the disk for processing.

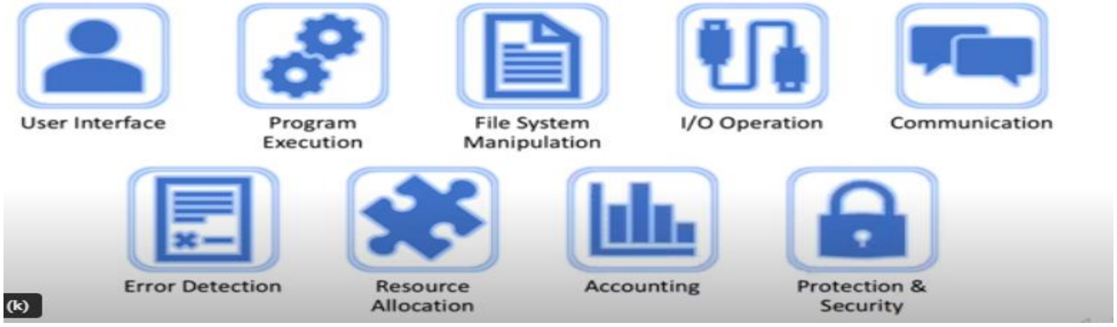
6. Security Management:

It is necessary to protect the processes from each other's activities. Security management ensures that the operating files, memory, CPU, and other hardware resources have proper authorization from the OS. No process can do its own I/O, this maintains the integrity of peripheral devices.

OPERATING SYSTEMS SERVICES

The operating system is a primary resource manager. It gives several services to utility programmers and users. Applications access these services through application programming interfaces or system calls.

By invoking those interfaces, the application can request a service from the operating system, pass parameters, and acquire the operation outcomes.



User Interface

User interface is essential and all operating systems provide it. Users either interface with the operating system through the command-line interface or GUI. The command interpreter executes the next user-specified command.

Program execution

The Operating System is responsible for the smooth execution of both user and system programs. It utilizes various resources available for the efficient running of all types of functionalities.

I/O Operations

Operating System manages the input-output operations and establishes communication between the user and device drivers. Device drivers are software that is associated with hardware that is being managed by the OS so that the sync between the devices works properly.

Accounting

An Operating device collects utilization records for numerous assets and tracks the overall performance parameters and responsive time to enhance overall performance. These personal records are beneficial for additional upgrades and tuning the device to enhance overall performance.

Security and Protection

Operating device affords safety to the statistics and packages of a person and protects any interference from unauthorized users. The safety feature counters threats, which are published via way of individuals out of doors the manage of the running device.

Communication

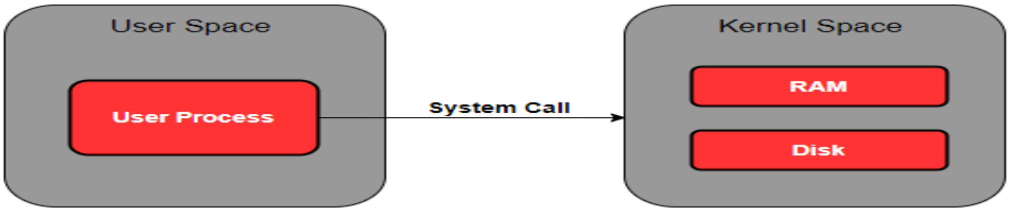
The operating system manages the exchange of data and programs among different computers connected over a network. This communication is accomplished using message passing and shared memory.

Error Handling

The Operating System also handles the error occurring in the CPU, in Input-Output devices, etc. It also ensures that an error does not occur frequently and fixes the errors. It also prevents the process from coming to a deadlock. It also looks for any type of error or bugs that can occur while any task.

SYSTEMS CALLS

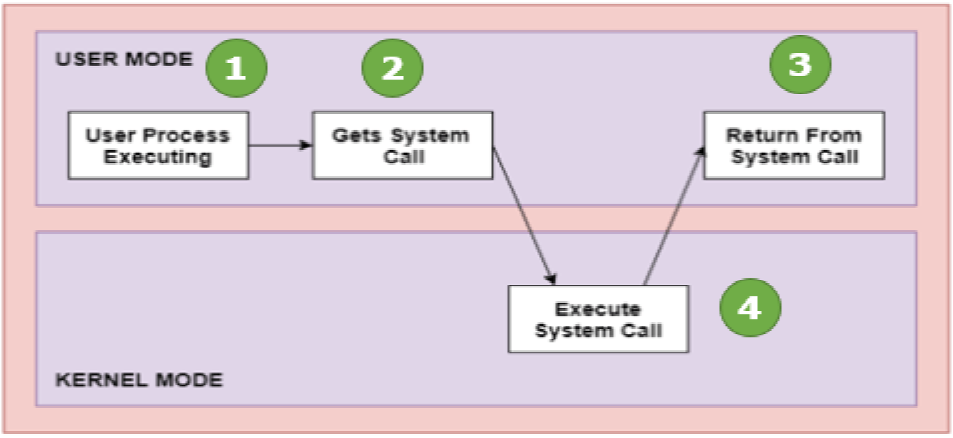
A system call is a mechanism that provides the interface between a process and the operating system. It is a programmatic method in which a computer program requests a service from the kernel of the OS.



System call offers the services of the operating system to the user programs via API (Application Programming Interface). System calls are the only entry points for the kernel system.

In operating systems, the user is not allowed to directly access the kernel from user mode, and the user needs an interface between the user process and operating systems to access the kernel services. System calls provide this interface.

How System Call Works?



Step 1) The processes executed in the user mode till the time a system call interrupts it.

Step 2) After that, the system call is executed in the kernel-mode on a priority basis.

Step 3) Once system call execution is over, control returns to the user mode.

Step 4) The execution of user processes resumed in Kernel mode.

Types of System calls:

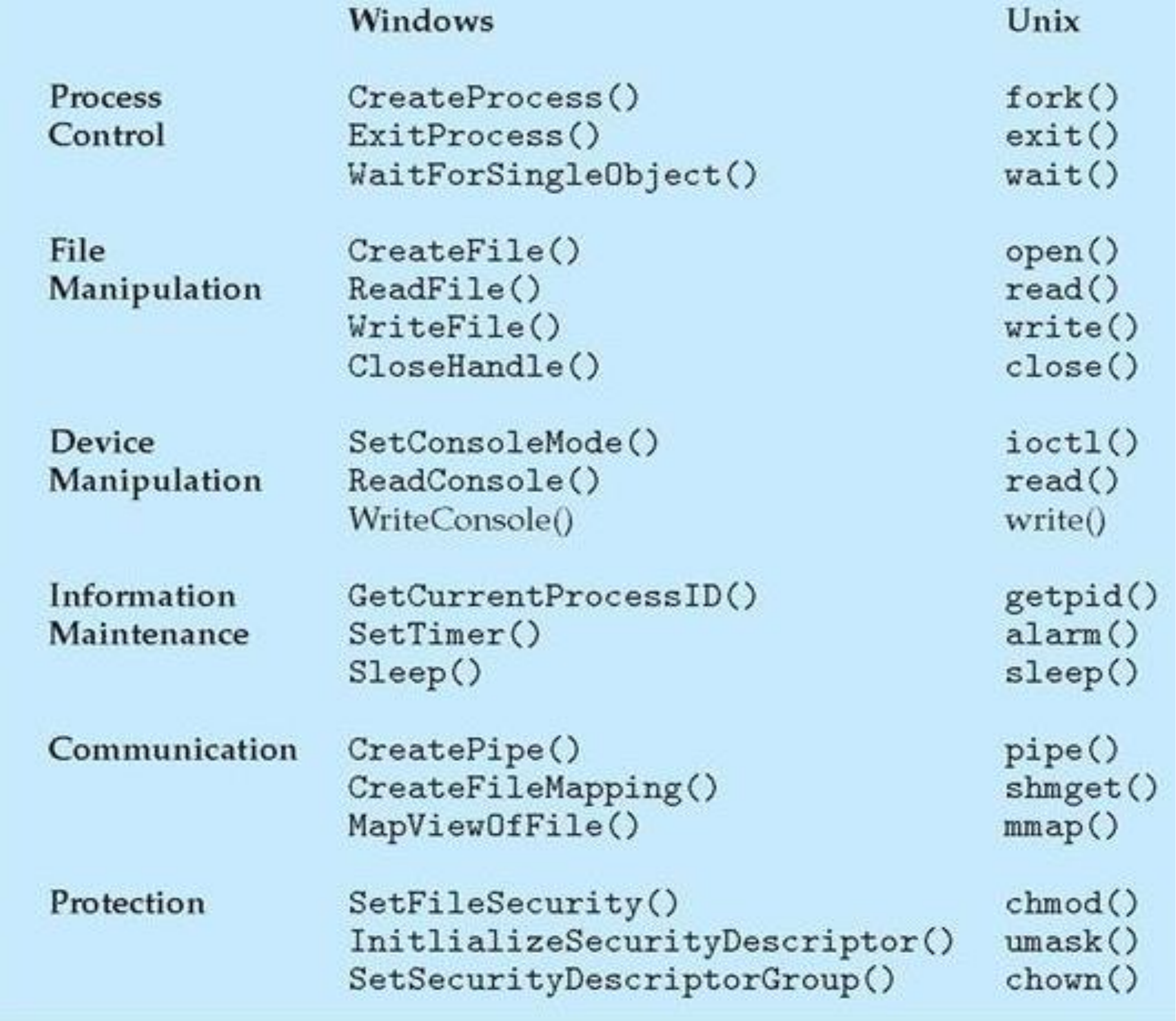
Process control: This system handles all services related to processes such as process termination, creation, and so on.

File management: This system call is responsible for all services related to file manipulation, such as writing into a file, reading from a file, creating a file, and so on.

Device management: This system call is responsible for all services related to device manipulation, such as reading or writing to the device buffers.

Information maintenance: This system call is responsible for all services related to information handling and its transfer between the process or operating system and the user program.

Communication: This system call is responsible for inter-process communication between co-operating processes. These calls are also responsible for handling the communication link between processes.



	Windows	Unix
Process Control	CreateProcess()	fork()
	ExitProcess()	exit()
	WaitForSingleObject()	wait()
File Manipulation	CreateFile()	open()
	ReadFile()	read()
	WriteFile()	write()
	CloseHandle()	close()
Device Manipulation	SetConsoleMode()	ioctl()
	ReadConsole()	read()
	WriteConsole()	write()
Information Maintenance	GetCurrentProcessID()	getpid()
	SetTimer()	alarm()
	Sleep()	sleep()
Communication	CreatePipe()	pipe()
	CreateFileMapping()	shmget()
	MapViewOfFile()	mmap()
Protection	SetFileSecurity()	chmod()
	InitlializeSecurityDescriptor()	umask()
	SetSecurityDescriptorGroup()	chown()

Important System Calls Used in OS

1. wait()

In certain systems, a process must wait for another process to finish running before continuing. When a parent process creates a child process, the parent process's execution is suspended until the child process has finished running.

With a wait() system call, the parent process is automatically suspended. Control returns to the parent process once the child process has completed its execution.

2. fork()

The fork system call in OS is used by processes to make copies of themselves. By using this system, the Call parent process can create a child process, and the parent process's execution will be halted while the child process runs.

3. exec()

When an executable file replaces an earlier executable file within the context of an active process, this system call is executed. The original process identifier is still present even though a new process is not created; instead, the new process replaces the old one's stack, data, head, data, etc.

4. kill()

The OS uses the kill() system call to urge processes to terminate by sending them a signal. A kill system call can have different meanings and is not always used to end a process.

5. exit()

An exit system call is used when the programme must be stopped. When the exit system call is used, the resources that the process was using were released.