# Module - 1

**Syllabus: -**

* **Introduction:** Operating system functions, OS Architecture (Monolithic, Microkernel, Layered, Hybrid), Different types of O.S. (batch, multi-programmed, time-sharing, real-time, distributed, parallel), Evaluation of OS.
* **System Structure:** Computer system operation, I/O structure, storage structure, storage hierarchy, different types of protections, Operating system structure (simple, layered, virtual machine), O/S services, System calls.
* **Protection & Security:** Goals of protection, Domain of protection, Access matrix and its representation, Threats and system security.
* **Processes and Threads:** 7 state process model, Process scheduling, Operations on processes, Inter-process communication, Threads overview, Benefits of threads, User and kernel threads.

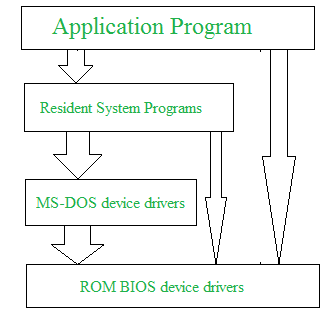
**Functions of Operating System: -**

1. Memory Management
2. Processor Management
3. Device Management
4. File Management
5. Security

**OS Architecture**

## ****Simple/Monolithic structure****

These operating systems do not have well-defined structures and are small, simple, and limited. The interfaces and levels of functionality are not well separated. [MS-DOS](https://www.geeksforgeeks.org/ms-dos-operating-system/) is an example of such an operating system. These types of operating systems cause the entire system to crash if one of the user programs fails.



### ****Advantages of Simple/Monolithic structure****

1. Better application performance.
2. It is easy for kernel developers to develop such an operating system.

### ****Disadvantages of Simple/Monolithic structure****

1. The structure is very complicated, as no clear boundaries exist between modules.
2. It does not implement data hiding in the operating system.

## Micro-kernel Structure

The structure of this operating system is designed by removing all non-essential components from the kernel and implementing them as system and user programs. This results in a smaller kernel known as the micro-kernel.  All new services can be added to the user space and does not require the kernel to be modified. It is more secure and reliable as if a service fails, then rest of the operating system remains untouched. Mac OS is an example of this type of OS.

### Advantages of Micro-kernel structure

1. It makes the operating system portable to various platforms.
2. As microkernels are small so these can be tested effectively.

### Disadvantages of Micro-kernel structure

1. Degradation of system performance.

## Hybrid-Kernel Structure

Hybrid-kernel structure is a combination of both monolithic-kernel structure and micro-kernel structure. Basically, it combines properties of both monolithic and micro-kernel and make a more advance structure. It implements speed and design of monolithic and modularity and stability of micro-kernel structure.

### Advantages of Hybrid-Kernel Structure

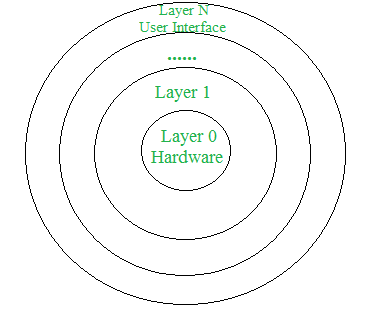
1. Better performance
2. Supports a wide range of hardware and applications.
3. Provides better isolation and security by implementing micro-kernel approach.

### Disadvantages of Hybrid-Kernel Structure

1. It increases overall complexity of system by implementing both structure (monolithic and micro and making the system difficult to understand.
2. The layer of communication between micro-kernel and other component increases time complexity and decreases performance compared to monolithic kernel.

## ****Layered structure****

In this structure, the OS is broken into a number of layers (levels). The bottom layer (layer 0) is the hardware, and the topmost layer (layer N) is the user interface. These layers are designed in such a way that each layer uses the functions of the lower-level layers only.   
UNIX is an example of this structure. 



### ****Advantages of Layered structure****

1. Layering makes it easier to enhance the operating system.
2. It is very easy to perform debugging and system verification.

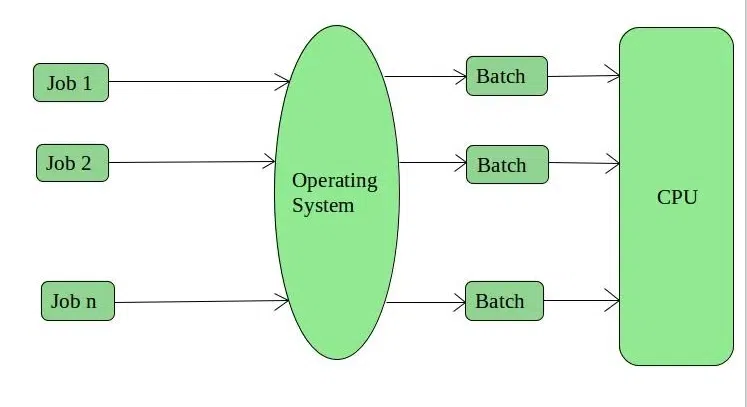
### ****Disadvantages of Layered structure****

1. In this structure, the application’s performance is degraded as compared to simple structure.
2. It requires careful planning for designing the layers, as the higher layers use the functionalities of only the lower layers.

**Types of Operating System: -**

### ****Batch Operating System****

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



**Advantages of Batch Operating System**

* Multiple users can share the batch systems.
* The idle time for the batch system is very less.

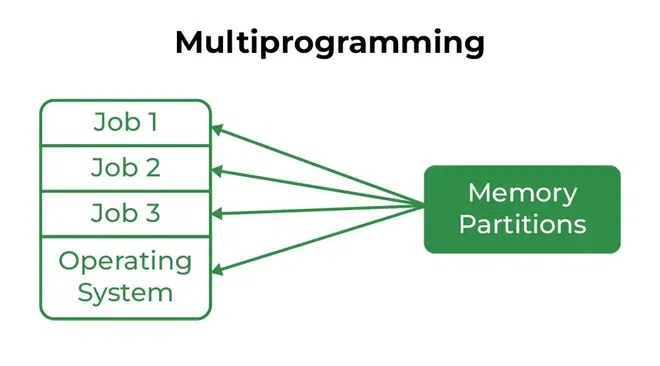
**Disadvantages of Batch Operating System**

* Difficult to know the time required for any job to complete
* It is sometimes costly.
* The other jobs will have to wait for an unknown time if any job fails.

**Examples of Batch Operating Systems:** Payroll Systems, Bank Statements, etc.

### Multi-Programming Operating System

A multi programming operating system (OS) allows multiple programs to run simultaneously on a single processor. This is done by allocating CPU time to different programs in such a way that it appears that they are all running simultaneously.



**Advantages of Multi-Programming Operating System**

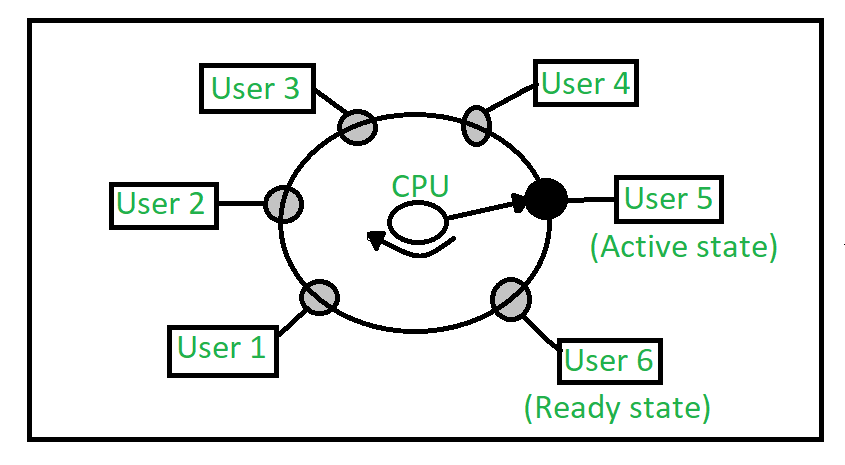
* Multi Programming increases the Throughput of the System.
* It helps in reducing the response time.

**Disadvantages of Multi-Programming Operating System**

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

### ****Time-Sharing Operating Systems****

A time-sharing operating system (OS) is a type of operating system where multiple users can interact with the computer simultaneously. It allows several users to use the computer at the same time by dividing the CPU time among them.



**Advantages of Time-Sharing OS**

* Each task gets an equal opportunity.
* Fewer chances of duplication of software.
* Resource Sharing.

**Disadvantages of Time-Sharing OS**

* High Overload
* Complexity
* Security Risks

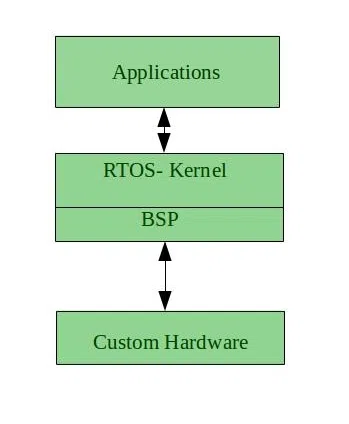
### ****Real-Time Operating System****

Real-time operating system is an operating system used for real-time computing applications that processes data having a deadline.

**Types of Real-Time Operating Systems**

* **Hard Real-Time Systems:**   
  Hard Real-Time OSs are meant for applications where time constraints are very strict and even the shortest possible delay is not acceptable. These systems are built for saving life like automatic parachutes or airbags which are required to be readily available in case of an accident. Virtual memory is rarely found in these systems.
* **Soft Real-Time Systems:**   
  These OSs are for applications where time-constraint is less strict.

For more, refer to the [Difference Between Hard Real-Time OS and Soft Real-Time OS](https://www.geeksforgeeks.org/difference-between-hard-real-time-and-soft-real-time-system/).



**Advantages of RTOS**

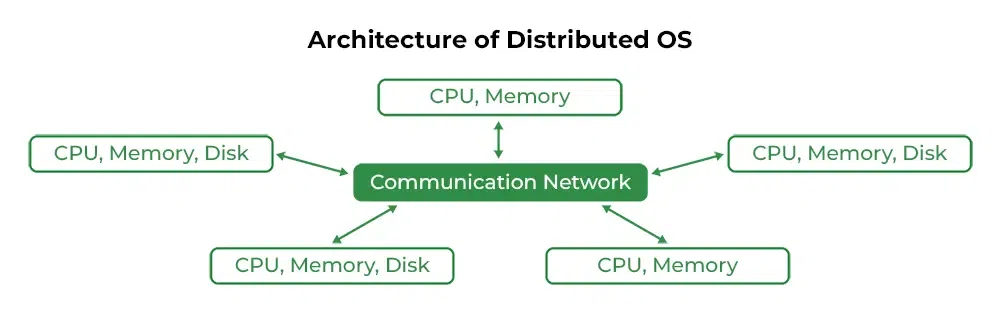
* **Maximum Consumption**
* **Task Shifting**
* **Focus on Application**
* **Error Free**

**Disadvantages of RTOS**

* **Limited Tasks**
* **Complex Algorithms**
* **Thread Priority**

**Examples of Real-Time Operating Systems are** Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

### ****Distributed Operating System**** Distributed operating system is a type of operating system that runs on a network of interconnected computers, often referred to as a distributed system. A distributed operating system extends its capabilities across multiple machines to provide a unified computing environment.



**Advantages of Distributed Operating System**

* Failure of a single system will not affect the network.
* Data exchange speed is increased.
* Computation is highly fast and durable.

**Disadvantages of Distributed Operating System**

* Failure of the main network will stop the entire communication.
* Expensive

**Computer System Operations**

Computer system operations are actions that are performed to accomplish a task.

There are five basic types of computer operations:

1. **Inputting:** Transferring information into a computer system
2. **Processing:** Can process data as required by the user
3. **Outputting:** Giving results in the form of output
4. **Storing:** Storing data
5. **Controlling:** Controlling all operations inside a computer

**Storage Structure and Hierarchy**

Storage structure refers to the way in which the data is stored and organized within a computer system. It involves the use of different storage media such as hard disks and memory to store and retrieve data.

**The storage hierarchy typically consists of the following levels: -**

1. **Registers:** Registers are the fastest and the smallest type of storage in a computer system that holds the data that the cpu needs to access quickly.
2. **Cache Memory:** Cache memory is a small amount of high-speed memory that is used to temporarily store frequently accessed data. Cache memory is faster than main memory, but it is also more expensive.
3. **Main Memory (RAM):** Main memory, also known as random access memory (RAM), is the primary storage used by a computer system. RAM is larger than cache memory, but it is also slower.
4. **Virtual Memory:** Virtual memory is a technique used by computer systems to extend the amount of available memory by using hard disk space.
5. **Secondary Storage:** Secondary storage is used to store data and programs that are not currently being used by the computer system.
6. **Tertiary Storage:** Tertiary storage is used for long-term storage of data that is not frequently accessed.

**Different Types of Protection: -**

1. System protection
2. User Authentication
3. Access control
4. Encryption
5. Firewall

## Services of Operating System

1. Program execution
2. Input Output Operations
3. Communication between Process
4. File Management
5. Memory Management
6. Process Management
7. Security and Privacy
8. Resource Management
9. User Interface
10. Networking
11. Error handling
12. Time Management

**Program Execution** - It is the Operating System that manages how the program is going to be executed. It loads the program into the memory after which it is executed.

**Input Output Operation** - Operating System manages the input output operations and establishes the communication between the user and the device drivers

**Communication between processes** - The Operating system manages the communication between processes. Communication between processes includes data transfer among them.

**File Management** - The operating system manages files and provides a platform for the user to create and delete files and is responsible for making decision regarding the storage of all types of data and decides that how the data should be stored and manipulated.

**System Call**

## A system call is a programmatic way in which a computer program requests for a service from the kernel of the operating system. A system call is a way for the programs to interact with the operating system. A computer program makes a system call when it makes a request to the operating systems kernel. System call **provides** the services of the operating system to the user programs via Application Program Interface(API).

## ****Services Provided by System Calls****

* Process creation and management
* Main memory management
* File Access, Directory, and File system management
* Device handling(I/O)
* Protection

**Goals of Protection -**

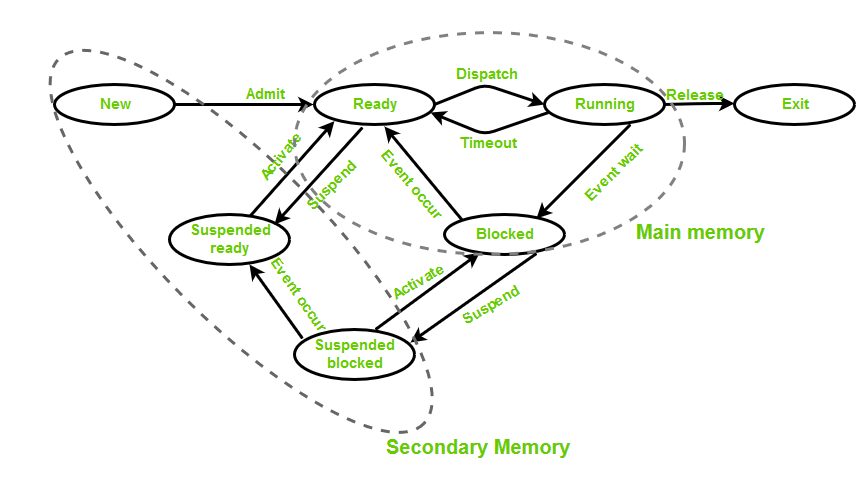
1. Data integrity
2. Availability
3. Confidentiality
4. Accountability
5. Isolation

**Domain of Protection -**

1. Process Isolation
2. Memory Protection
3. Network Security
4. File System Protection
5. Device Access Control

**7 State process model**

1. **New** - It is a program that is present in the secondary memory that will be picked up by the OS to create the process.
2. **Ready** - After the creation of the process the process enters into the ready state that is the process is loaded in to the main memory.
3. **Run** - The process is chosen form the ready queue by the cpu for the execution of the instructions within the process.
4. **Waiting** - A process enters into the waiting state when it is unable to proceed due to the unavailability of the resources such as input output operation, asking for user input or waiting for a specified event.
5. **Terminated** - This state indicates that the process execution has been completed.
6. **Suspend Ready** - The process that was initially in the ready state but was swapped out of main memory to any external storage.
7. **Suspend Wait** - It is the transition of the process from the suspend state to the ready state, indicating that the process is ready to be executed again.



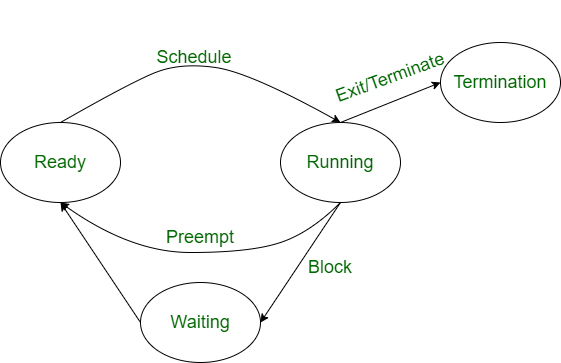
**Process Scheduling**

Process scheduling is the activity of the process manager that handles the removal of the running process from the CPU and the selection of another process based on a particular strategy.

Process scheduling is an essential part of a Multiprogramming operating system. Such operating systems allow more than one process to be loaded into the executable memory at a time and the loaded process shares the CPU using time multiplexing.

## ****Operation on a Process****

The execution of a process is a complex activity. It involves various operations. Following are the operations that are performed while execution of a process:



**Creation**

This is the initial step of the process execution activity. Process creation means the construction of a new process for execution. This might be performed by the system, the user, or the old process itself. There are several events that lead to the process creation. Some of the such events are the following:

1. When we start the computer, the system creates several background processes.
2. A user may request to create a new process.
3. A process can create a new process itself while executing.
4. The batch system takes initiation of a batch job.

### ****Scheduling/Dispatching****

The event or activity in which the state of the process is changed from ready to run. It means the operating system puts the process from the ready state into the running state. Dispatching is done by the operating system when the resources are free or the process has higher priority than the ongoing process. There are various other cases in which the process in the running state is preempted and the process in the ready state is dispatched by the operating system.

### ****Blocking****

When a process invokes an input-output system call that blocks the process, and operating system is put in block mode. Block mode is basically a mode where the process waits for input-output. Hence on the demand of the process itself, the operating system blocks the process and dispatches another process to the processor. Hence, in process-blocking operations, the operating system puts the process in a ‘waiting’ state.

### ****Preemption****

When a timeout occurs that means the process hadn’t been terminated in the allotted time interval and the next process is ready to execute, then the operating system preempts the process. This operation is only valid where CPU scheduling supports preemption. Basically, this happens in priority scheduling where on the incoming of high priority process the ongoing process is preempted. Hence, in process preemption operation, the operating system puts the process in a ‘ready’ state.

### Process Termination

Process termination is the activity of ending the process. In other words, process termination is the relaxation of computer resources taken by the process for the execution. Like creation, in termination also there may be several events that may lead to the process of termination. Some of them are:

1. The process completes its execution fully and it indicates to the OS that it has finished.
2. The operating system itself terminates the process due to service errors.
3. There may be a problem in hardware that terminates the process.
4. One process can be terminated by another process.