1. Define OS. List the Objectives of OS.

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.

**Objectives of Operating System**

The objectives of the operating system are −

* To make the computer system convenient to use in an efficient manner.
* To hide the details of the hardware resources from the users.
* To provide users a convenient interface to use the computer system.
* To act as an intermediary between the hardware and its users, making it easier for the users to access and use other resources.
* To manage the resources of a computer system.
* To keep track of who is using which resource, granting resource requests, and mediating conflicting requests from different programs and users.
* To provide efficient and fair sharing of resources among users and programs.

2. List the goals of OS.

**Operating system goals:**

* Execute user programs and make solving user problems easier
* Make the computer system convenient to use
* Use the computer hardware in an efficient manner

3. Explain the following types of systems: Time sharing systems Multi- processor systems, Distributed system.

**Time Sharing Operating System:-**

Time-sharing enables many people, located at various terminals, to use a particular computer system at the same time. Multitasking or Time-Sharing Systems is a logical extension of multiprogramming. Processor’s time is shared among multiple users simultaneously is termed as time-sharing. Multiple jobs are implemented by the CPU by switching between them, but the switches occur so frequently. So, the user can receive an immediate response.

Advantages of Timesharing operating systems are –

* It provides the advantage of quick response.
* This type of operating system avoids duplication of software.
* It reduces CPU idle time.

Disadvantages of Time-sharing operating systems are −

* Time sharing has problem of reliability.
* Question of security and integrity of user programs and data can be raised.
* Problem of data communication occurs.

**Distributed systems:-**

A distributed system is a collection of physically separate computer systems that are networked to provide the users with access to the various resources that the system maintains. Access to a shared resources increases computation speed, functionality, data availability, and reliability. Distributed systems depend on computer communication networks for their functionality. A network is a communication path between two or more systems. Networks are characterized based on the distances between their nodes. A local area network (LAN) connects computers with in a room, floor, or a building. A wide area network (WAN) usually links building, cities,or countries.

Features of Distributed systems:-

* Resource sharing: Improves resource utilization across boundaries of individual computer systems.
* Computation speed up: Parts of a computation can be executed in different computer systems to speed up the computation.

**Parallel Systems:-**

Parallel systems are also known as multi-processor systems or tightly coupled systems. Such systems have two or more processors in close communication, sharing the computer bus and sometimes the clock, memory, and peripheral devices.

Multiprocessor systems have three main advantages:

* Increased throughput: By increasing the number of processor, we expect to get more work done in less time.
* Economy of scale: Multiprocessor system can cost less than equivalent multiple single processor systems, because they can share peripherals, mass storage, and power supplies. If several programs operate on the same set of data, it is cheaper to store those data on one disk and to have all the processor share them than to have many computers with local disks and many copies of the data.
* Increased reliability: if functions can be distributed properly among several processors, then the failure of one processor will not halt the system but slow it down. If one processor fails, then each of the remaining processors can pick up share of the work of the failed processor.

The multiprocessor systems are of two types.

1. Asymmetric multiprocessing: In asymmetric multiprocessing each processor is assigned a specific task. A master processor controls the system. The master processor schedules and allocates work to the slave processors.

2. Symmetric multiprocessing: In symmetric multiprocessing system, each processor performs all tasks within the operating system. There is no master slave relationship between processors. Communication: It provides means of communication between remote entities.

4. Define Symmetric and Asymmetric Multiprocessing.

**1. Asymmetric multiprocessing:** In asymmetric multiprocessing each processor is assigned a specific task. A master processor controls the system. The master processor schedules and allocates work to the slave processors.

**2. Symmetric multiprocessing:** In symmetric multiprocessing system, each processor performs all tasks within the operating system. There is no master slave relationship between processors.

5. Explain the types of real time systems.

**A hard real time system:-** is typically dedicated to processing real time applications, and meets the response requirement of an application under all conditions.

**A soft real time system:-** makes the best effort to meet the response requirement of a real time application but cannot guarantee that it will be able to meet it under all conditions.

6. List the applications of RTOS.

pplications such

as launching a space shuttle, Industrial control systems (Robot Arms), whether monitoring

and forecasting applications, Missile guiding system etc.

7. List the Operations performed on OS.

1. Resource allocations
2. File Management
3. Process Management
4. I/O Device Management
5. Network Management
6. Main Memory management
7. Secondary-Storage Management
8. Security Management
9. OS services

8. Explain in detail the operations performed on OS.

1. **Resource allocation**:-

The Operating System allocates resources when a program need them. When the program terminates, the resources are de-allocated, and allocated to other programs that need them. There are two popular methods for resource allocation.

* Partitioning of resources
* Allocation from a pool

In partition method, the OS makes a decision to allocate resources to the programs before the execution begins. This approach is called static allocation. Static allocation is simple to implement. However it lacks flexibility that leads to problems like wastage of resources.

1. **File Management:-**

A file is a set of related information which is should define by its creator. It commonly represents programs, both source and object forms, and data. Data files can be numeric, alphabetic, or alphanumeric.

**Function of file management in OS:**

The operating system has the following important given activities in connections with file management:

* File and directory creation and deletion.
* For manipulating files and directories.
* Mapping files onto secondary storage.
* Backup files on stable storage media.

1. **Process Management:-**

The process management component is a procedure for managing the many processes that are running simultaneously on the operating system. Every software application program has one or more processes associated with them when they are running.

**Functions of process management in OS:**

The following are functions of process management.

* Process creation and deletion.
* Suspension and resumption.
* Synchronization process
* Communication process

1. **I/O Device Management:-**

One of the important use of an operating system that helps you to hide the variations of specific hardware devices from the user.

**Functions of I/O management in OS:**

* It offers buffer caching system
* It provides general device driver code
* It provides drivers for particular hardware devices.
* I/O helps you to knows the individualities of a specific device.

1. **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.

**Functions of Network management:**

* Distributed systems help you to various computing resources in size and function. They may involve microprocessors, minicomputers, and many general-purpose computer systems.
* A distributed system also offers the user access to the various resources the network shares.
* It helps to access shared resources that help computation to speed-up or offers data availability and reliability.

1. **Main Memory management:-**

Main Memory is a large array of storage or bytes, which has an address. The memory management process is conducted by using a sequence of reads or writes of specific memory addresses.

**Functions of Memory management in OS**: An Operating System performs the following functions for Memory Management:

* It helps you to keep track of primary memory.
* Determine what part of it are in use by whom, what part is not in use.
* In a multiprogramming system, the OS takes a decision about which process will get Memory and how much.
* Allocates the memory when a process requests
* It also de-allocates the Memory when a process no longer requires or has been terminated.

1. **Secondary-Storage Management:-**

The most important task of a computer system is to execute programs. These programs, along with the data, help you to access, which is in the main memory during execution. **Functions of Secondary storage management in OS**:Here, are major functions of secondary storage management in os.

* Storage allocation
* Free space management
* Disk scheduling

1. **Security Management:-**

The various processes in an operating system need to be secured from each other’s activities. For that purpose, various mechanisms can be used to ensure that those processes which want to operate files, memory CPU, and other hardware resources should have proper authorization from the operating system.

For example, Memory addressing hardware helps you to confirm that a process can be executed within its own address space. The time ensures that no process has control of the CPU without renouncing it.

1. **OS services:-**

An Operating System supplies different kinds of services to both the users and to the programs as well. It also provides application programs (that run within an Operating system) an environment to execute it freely. It provides users the services run various programs in a convenient manner.

9. What is a system call? List the different types of system calls.

A system call is a way for a user program to interface with the operating system. The program requests several services, and the OS responds by invoking a series of system calls to satisfy the request. System calls are predefined functions that the operating system may directly invoke if a high-level language is used.

Types of System Calls:-

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

1. Process Control

2. File Management

3. Device Management

4. Information Maintenance

5. Communication

10. Explain the different types of System Calls.

**Process Control:-** Process control is the system call that is used to direct the processes. Some process controlexamples 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 managementexamples 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 devicemanagement 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 ordate, set system data, etc.

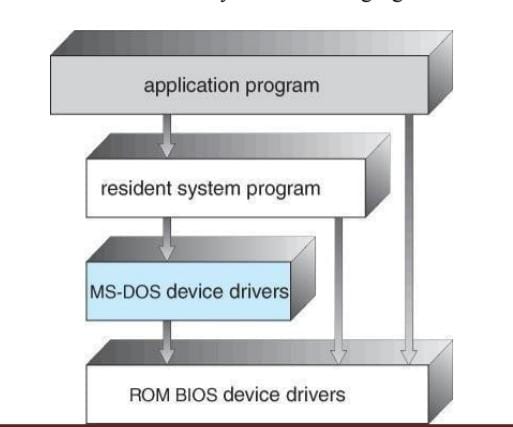
**Communication :-** Communication is a system call that is used for communication. There are some examples of communication, including create, delete communication connections, send, receive messages, etc.

11. List the services of operating systems.

1. User Interface
2. Program Execution
3. File system manipulation
4. Input / Output Operations
5. Communication
6. Resource Allocation
7. Error Detection
8. Accounting
9. Security and protection

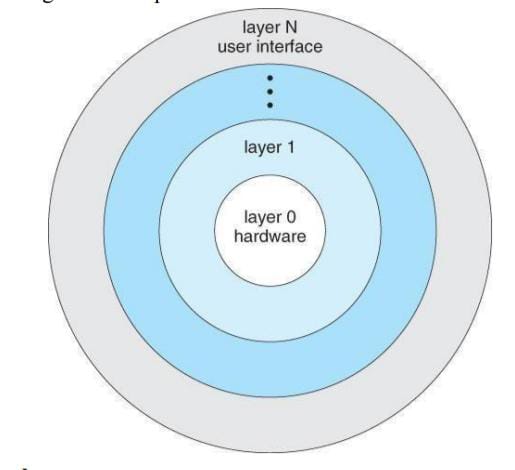
12. Explain Simple and Layered Operating system.

**Simple Structure:-** Many operating systems do not have well defined structures. Such systems started as small, simple, and limited systems and then grew beyond their original scope. MS – DOS is an example of such a system. It was written to provide the most functionality in the least space, so it was not divided into modules carefully. The following figure shows its structure.



**Layered Approach :-**

* Another approach is to break the OS into a number of smaller layers, each of which rests on the layer below it, and relies solely on the services provided by the next lower layer.
* This approach allows each layer to be developed and debugged independently, with the assumption that all lower layers have already been debugged and are trusted to deliver proper services.
* The problem is deciding what order in which to place the layers, as no layer can call upon the services of any higher layer, and so many chicken-and-egg situations may arise.
* Layered approaches can also be less efficient, as a request for service from a higher layer has to filter through all lower layers before it reaches the HW, possibly with significant processing at each step.



13. Explain monolithic and microkernel operating system.

**Monolithic kernel:-**A monolithic kernel is an operating system architecture where the entire operating systemis working in kernel space. The monolithic model differs from other operating systemarchitectures, such as the microkernel architecture, in that it alone defines a high-level virtualinterface over computer hardware.A set of primitives or system calls implement all operating system services such asprocess management, concurrency, and memory management. Device drivers can be added to thekernel as modules.

Advantages of Monolithic Kernel:-

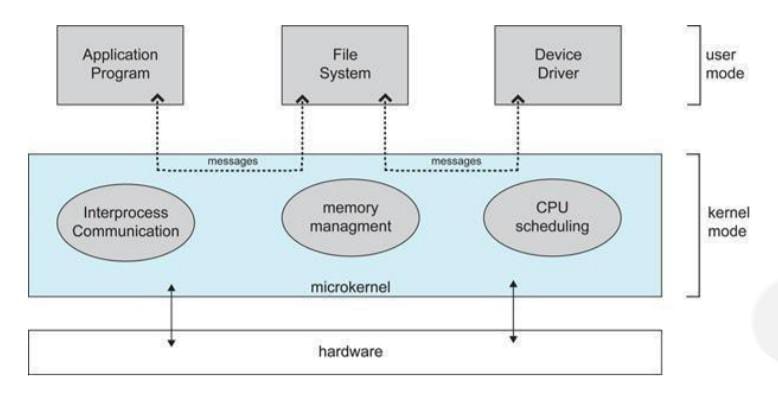
* The execution of the monolithic kernel is quite fast as the services such as memory management, file management, process scheduling, etc., are implemented under the same address space.
* A process runs completely in single address space in the monolithic kernel.
* The monolithic kernel is a static single binary file.

Disadvantages of Monolithic Kernel :-

* If any service fails in the monolithic kernel, it leads to the failure of the entire system.
* The entire operating system needs to be modified by the user to add any new service.

Microkernel:-

The basic idea behind micro kernels is to remove all non-essential services from the kernel, and implement them as system applications instead, thereby making the kernel as small and efficient as possible. Most microkernels provide basic process and memory management, and message passing between other services, and not much more. and protection can be enhanced, as most services are performed in user mode, not kernel mode. System expansion can also be easier, because it only involves adding more system applications, not rebuilding a new kernel. Mach was the first and most widely known microkernel, and now forms a major component of Mac OSX.



Advantages of Microkernel :-

* Microkernels are secure since only those parts are added
* Microkernel architecture is compact and isolated, so it may perform better.
* The system expansion is more accessible, so it may be introduced to the system application without disrupting the kernel.
* When compared to monolithic systems, microkernels have fewer system crashes.

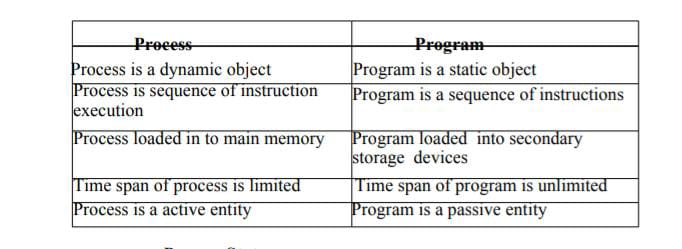
Disadvantages of Microkernel

* a context switch or a function call is needed.
* In a microkernel system, providing services are more costly than in a traditional monolithic system

14. Define Process.

**PROCESS**: The process is defined as a program in execution. The program by itself is not a process; a program is a passive entity, such as a file containing a list of instructions stored on disk (often called executable file), whereas a process is an active entity, with a program counter specifying the next instruction to execute. A program becomes a process when an executable file is loaded into memory.

15. Differentiate between Process and Thread.



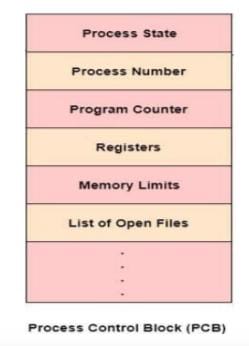
16. Explain the Process Control Block.

**Process Control Block (PCB):-** Process Control Block is a data structure that contains information of the process related to it.

* The process control block is also known as a task control block, entry of the process table, etc.
* It is very important for process management as the data structuring for processes are done in terms of the PCB. It also defines the current state of the operating system.

**Structure of the Process Control Block**

* The process control stores many data items that are needed for efficient process management.
* Some of these data items are explained with the help of the given diagram



The following are the data items −

**Process State**

* This specifies the process state i.e. new, ready, running, waiting or terminated.

**Process Number**

* This shows the number of the particular process.

**Program Counter**

* This contains the address of the next instruction that needs to be executed in the process.

**Registers**

* This specifies the registers that are used by the process. They may include accumulators, index registers, stack pointers, general purpose registers etc.

**List of Open Files**

* These are the different files that are associated with the process

**CPU Scheduling Information**

* The process priority, pointers to scheduling queues etc. is the CPU scheduling information that is contained in the PCB.
* This may also include any other scheduling parameters.

**Memory Management Information**

* The memory management information includes the page tables or the segment tables depending on the memory system used.
* It also contains the value of the base registers, limit registers etc.

**I/O Status Information**

* This information includes the list of I/O devices used by the process, the list of files etc.

**Accounting information**

* The time limits, account numbers, amount of CPU used, process numbers etc. are all a part of the PCB accounting information.

**Location of the Process Control Block**

* The process control block is kept in a memory area that is protected from the normal user access.
* This is done because it contains important process information. Some of the operating systems place the PCB at the beginning of the kernel stack for the process as it is a safe location.

17. What is a process? Explain different process states.

**PROCESS:** The process is defined as a program in execution. The program by itself is not a process; a program is a passive entity, such as a file containing a list of instructions stored on disk (often called executable file), whereas a process is an active entity, with a program counter specifying the next instruction to execute. A program becomes a process when an executable file is loaded into memory.

**Process States:-** When a process executed, it changes the state, generally the state of process is determined by the current activity of the process. Each process may be in one of the following states:

New : The process is being created.

Running : The process is being executed.

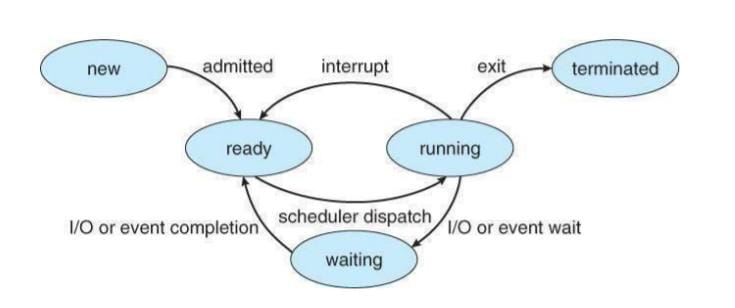
Waiting : The process is waiting for some event to occur.

Ready : The process is waiting to be assigned to a

processor. Terminated : The Process has finished execution.

Only one process can be running in any processor at any time, But many process may be in ready and waiting states. The ready processes are loaded into a “ready queue”.

**Diagram of process state**



**New ->Ready :** OS creates process and prepares the process to be executed, and then OS moves the process into ready queue.

**Ready->Running:** OS selects one of the Jobs from ready Queue and move them from ready to Running.

**Running->Terminated** : When the Execution of a process has Completed, OS terminates that process from running state. Sometimes OS terminates the process for some other reasons including Time exceeded, memory unavailable, access violation, protection Error, I/O failure and soon.

**Running->Ready :** When the time slot of the processor expired (or) If the processor received any interrupt signal, the OS shifted Running -> Ready State.

**Running -> Waiting**: A process is put into the waiting state, if the process need an event occur (or) an I/O Device require.

**Waiting->Ready:** A process in the waiting state is moved to ready state when the event for which it has been Completed.

18. Define Independent and Cooperating Process.

A process is **independent** if it cannot affect or be affected by the other processes executing in the system. Any process that does not share data with any other process is independent.

A process is **cooperating** if it can affect or be affected by the other processes executing in the system. Clearly, any process that shares data with other processes is a cooperating process.

19. Explain the necessity of IPC.

**Need for IPC:**

There are several reasons for providing an environment that allows process cooperation:

**\* Information sharing:-** Since several users may be interested in the same piece of information (for instance, a shared file), we must provide an environment to allow concurrent access to such information.

\* **Computation** **speedup**:- If we want a particular task to run faster, we must break it into subtasks, each of which will be executing in parallel with the others. Notice that such a speedup can be achieved only if the computer has multiple processing cores.

**\* Modularity**:- We may want to construct the system in a modular fashion, dividing the system functions into separate processes or threads.

**\* Convenience**:- Even an individual user may work on many tasks…

**20. Explain the Operations performed on Process.**

**Operations on the Process**

**1. Creation**: Once the process is created, it will be ready and come into the ready queue (main memory) and will be ready for the execution.

**2. Scheduling:** Out of the many processes present in the ready queue, the Operating system chooses one process and start executing it. Selecting the process which is to be executed next, is known as scheduling.

**3. Execution**: Once the process is scheduled for the execution, the processor starts executing it. Process may come to the blocked or wait state during the execution then in that case the processor starts executing the other processes.

**4. Deletion/killing:** Once the purpose of the process gets over then the OS will kill the process. The Context of the process (PCB) will be deleted and the process gets terminated by the Operating system.

**21. Explain the process of Context switch in OS.**

**Context Switching in OS (Operating System)**

1. The Context switching is a technique or method used by the operating system to switch a process from one state to another to execute its function using CPUs in the system.

2. When switching performs in the system, it stores the old running process's status in the form of registers and assigns the CPU to a new process to execute its tasks.

3. While a new process is running in the system, the previous process must wait in a ready queue.

4. The execution of the old process starts at that point where another process stopped it.

5. It defines the characteristics of a multitasking operating system in which multiple processes shared the same CPU to perform multiple tasks without the need for additional processors in the system.

6. A context switching helps to share a single CPU across all processes to complete its execution and store the system's tasks status.

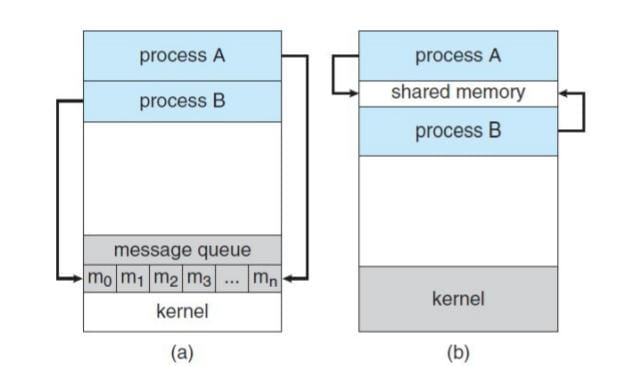
7. When the process reloads in the system, the execution of the process starts at the same point where there is conflicting.

22. Explain the 2 modes of IPC.

There are two fundamental models of inter process communication: shared memory and message passing.

In the **shared-memory model**, a region of memory that is shared by cooperating processes is established. Processes can then exchange information by reading and writing data to the shared region.

In the **message-passing model**, communication takes place by means of messages exchanged between the cooperating processes**.**

****