

**Q1) what is os? working principles?// what os do?  
sol)**

- 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 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:**

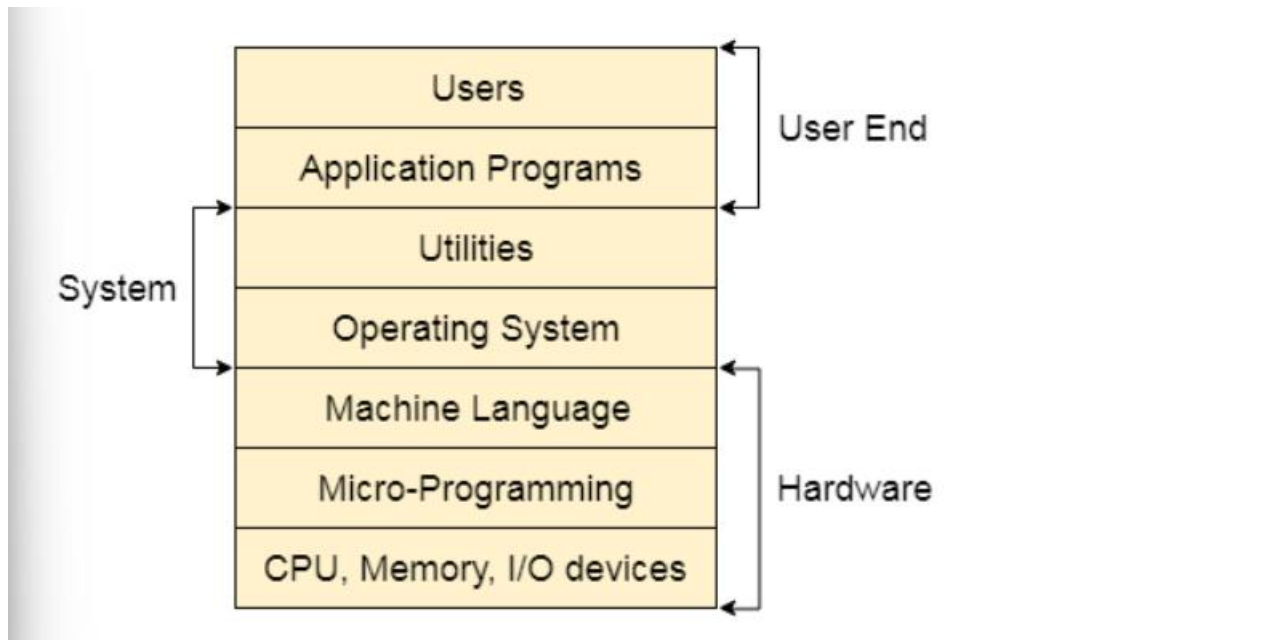
1.Users (people who are using the computer)

2.Application Programs (Compilers, Databases, Games, Video player, Browsers, etc.)

3.System Programs (Shells, Editors, Compilers, etc.)

4. Operating System ( A special program which acts as an interface between user and hardware )

5. Hardware ( CPU, Disks, Memory, etc)



## What does an Operating system do?

- 1.Process Management
- 2.Process Synchronization
- 3.Memory Management
- 4.CPU Scheduling
- 5.File Management
- 6.Security

\* /\*\*\*\*\*

## **Q2)Features of Operating System? sol)**

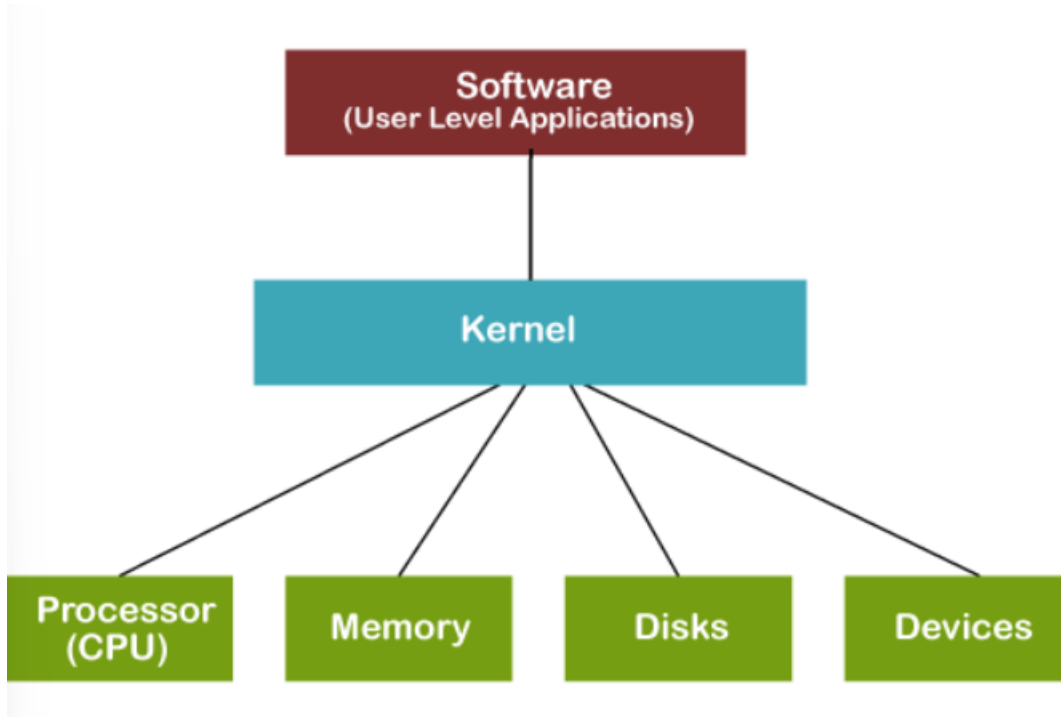
- 1.Error detection and handling
- 2.Handling I/O operations
- 3.Virtual Memory Multitasking
- 4.Program Execution
- 5.Allows disk access and file systems
- 6.Memory management
- 7.Security
- 8.Resource allocation
- 9.Easy to run
- 10.Information and Resource Protection
- 11.Manipulation of the file system

\*\*\*\*\*

### Q3) what is kernal? // what is kernal data structure?

**Sol)**

- Kernel is a computer program that is a core or heart of an operating system.
- structure of kernal:



- 
- Each operation of hardware and software is managed and administrated by the kernel.
- It acts as a bridge between applications and data processing
- It is the central component of an OS.

- It manages other tasks also such as memory management, task management, and disk management.

## Functions of a Kernel:

1. Device Management
2. Memory Management
3. Resource Management
4. Accessing Computer Resources

## Types of Kernel:

1. Monolithic Kernels
2. Microkernel
3. Hybrid Kernel
4. Nanokernel

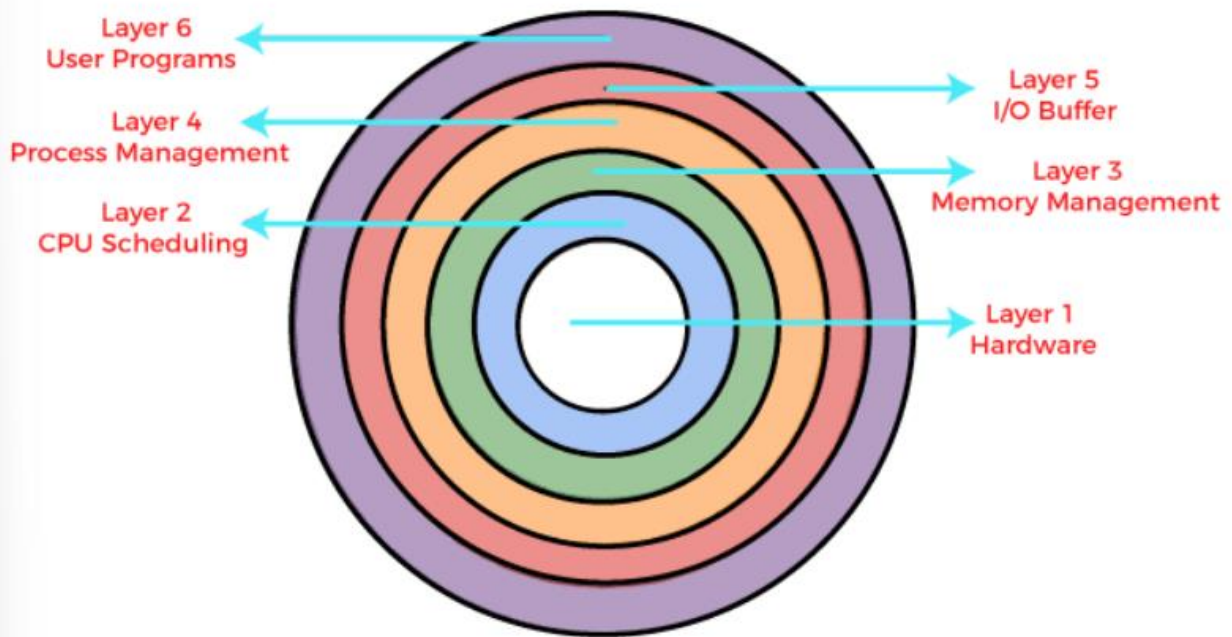
## 5. Exokernel

\*\*\*\*\*

### **q4)Structure of Operating System? sol)**

- The operating system can be implemented with the help of various structures
- The structure of the OS depends mainly on how the various common components of the operating system are interconnected and melded(kavavadam) into the kernel.
- The bottom layer (layer 0) is the hardware, and the topmost layer (layer N) is the user interface.
- These layers are so designed that each layer uses the functions of the lower-level layers only.





## Architecture of Layered Structure:

1. This type of operating system was created as an improvement over the early monolithic systems.
2. The operating system is split into various layers in the layered operating system, and



each of the layers has different functionalities

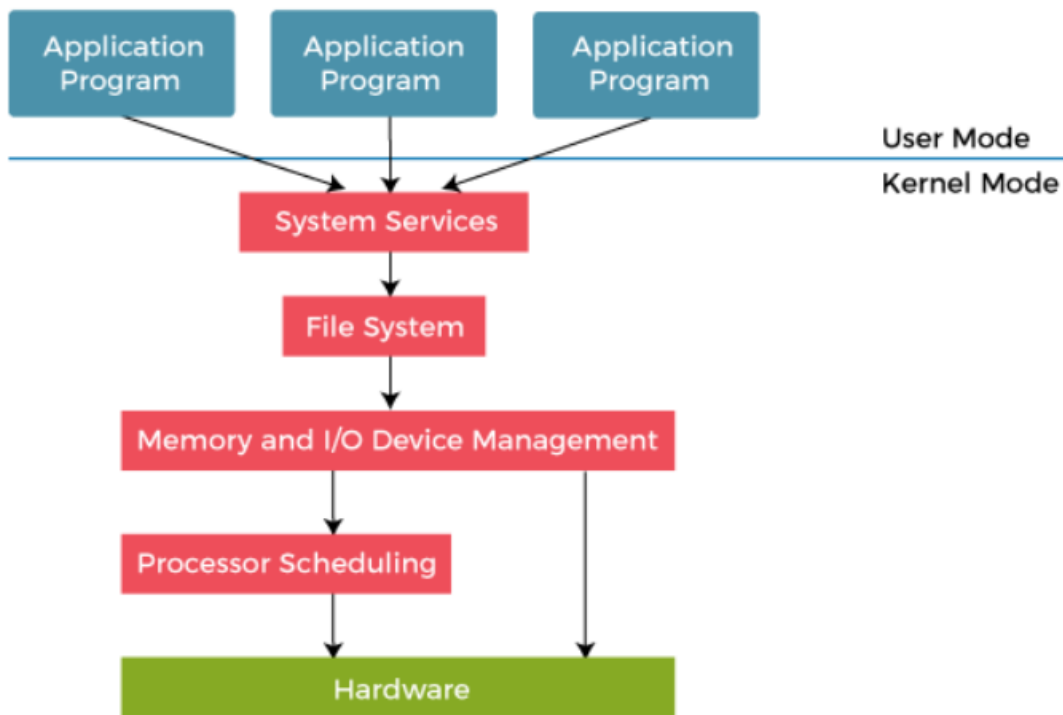
There are some rules in the implementation of the layers as follows.

1. A particular layer can access all the layers present below it, but it cannot access them.

2. That is, layer  $n-1$  can access all the layers from  $n-2$  to 0, but it cannot access the  $n$ th

3. Layer 0 deals with allocating the processes, switching between processes when interruptions occur or the timer expires.

There are six layers in the layered operating system:



## Advantages of Layered Structure:

- 1.Modularity
- 2.Easy debugging
- 3.Easy update
- 4.No direct access to hardware
- 5.Abstraction

## Disadvantages of Layered Structure:

- 1.Functionality
  - 2.Communication
  - 3.Slower in execution
  - 4.Complex and careful implementation
- 

**q5)\*\*what is protection and security ?**

**sol)**

- Protection is especially important in a multiuser environment when multiple users use computer resources such as CPU, memory, etc.
- It is the operating system's responsibility to offer a mechanism that protects each process from other processes.

- In a multiuser environment, all assets that require protection are classified as objects, and those that wish to access these objects are referred to as subjects.
- The operating system grants different 'access rights' to different subjects.
- A mechanism that controls the access of programs, processes, or users to the resources defined by a computer system is referred to as protection.
- You may utilize protection as a tool for multi-programming operating systems, allowing multiple users to safely share a common logical namespace, including a directory or files.
- It needs the protection of computer resources like the software, memory, processor, etc.

- It is critical to secure the device from unauthorized access, viruses, worms, and other malware.

## Need of Protection in Operating System:

Various needs of protection in the operating system are as follows:

1. There may be security risks like unauthorized reading, writing, modification, or preventing the system from working effectively for authorized users.
2. It helps to ensure data security, process security, and program security against unauthorized user access or program access.

3.It is important to ensure no access rights' breaches, no viruses, no unauthorized access to the existing data.

4.Its purpose is to ensure that only the systems' policies access programs, resources, and data.

## Security measures of Operating System:

There are various security measures of the operating system that the users may take. Some of them are as follows:

1.The network used for file transfers must be secure at all times.

2. During the transfer, no alien software should be able to harvest information from the network.

3. It is referred to as network sniffing, and it could be avoided by implementing encrypted data transfer routes.

4. The OS should be capable of resisting forceful or even accidental violations.

5. Passwords are a good authentication method, but they are the most common and it is very easy to crack passwords.

6. Security measures at various levels are put in place to prevent malpractices, like no one being allowed on the premises or access to the systems.

7.The best authentication techniques include a username-password combination, eye retina scan, fingerprint, or even user cards to access the system.

8.One-time passwords, encrypted passwords, and cryptography are used to create a strong password

---

**Q6)os serivices?**

**sol)**



**Following are the services provided by an operating system:**

1. Program execution
  2. Control Input/output devices
  3. Program creation
  4. Error Detection and Response
  5. Accounting
  6. Security and Protection
  7. File Management
  8. Communication
- 

**Q7) what are system calls?**

**sol)**

- 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.
- A system call can be written in assembly language or a high-level language like C or Pascal.
- System calls are predefined functions that the operating system may directly invoke if a high-level language is used.
- A system call is a method for a computer program to request a service from the kernel of the operating system 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
- System calls are required for any programs that use resources.
- When a computer software needs to access the operating system's kernel, it makes a system call.
- A system call function may create and use kernel processes to execute the asynchronous processing.
- System calls are not permitted to use shared libraries or any symbols that are not present in the kernel protection domain.

- The code and data for system calls are stored in global kernel memory.

## 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

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()

Q8)whta is linkers?

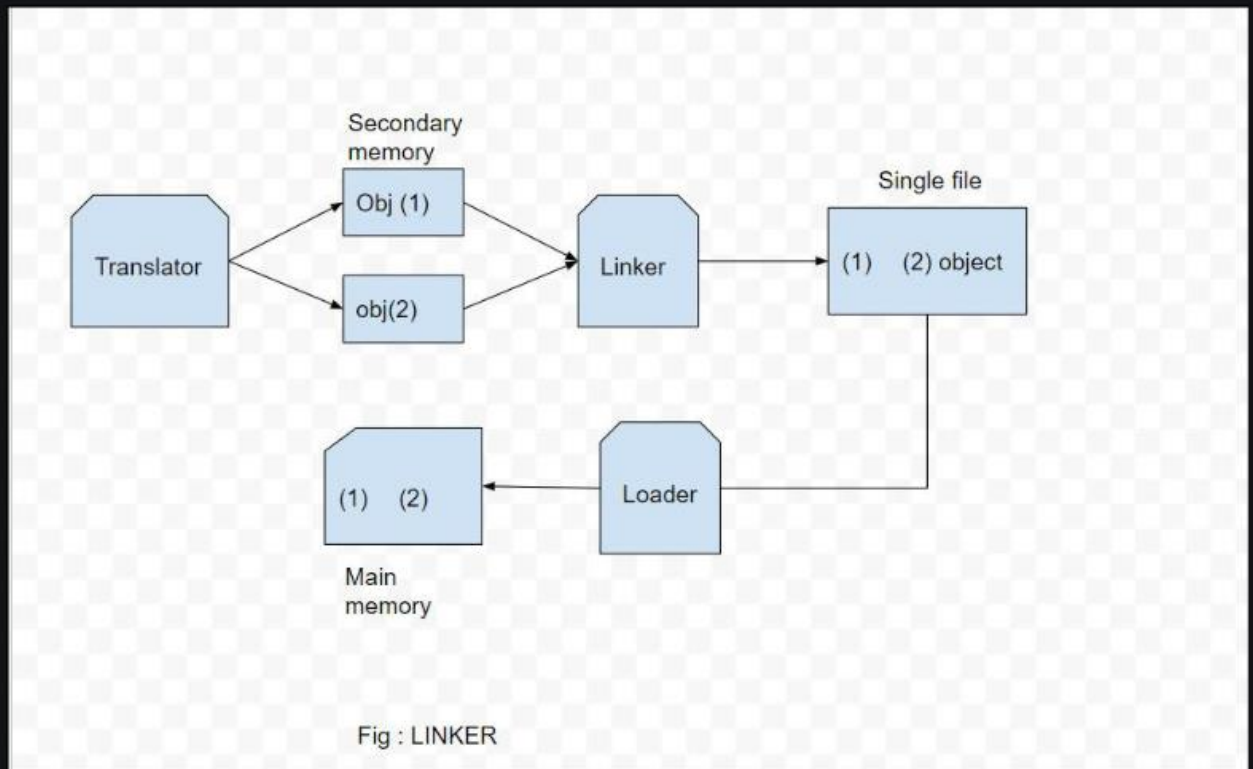
sol)

- Linker is a program in a system which helps to link object modules of a program into a single object file.
- It performs the process of linking.
- Linkers are also called as link editors.
- Linking is a process of collecting and maintaining piece of code and data into a single file
- Linker also links a particular module into system library.
- Linking is performed at both compile time and the program is loaded into memory by the loader.
- Linking is performed at the last step in compiling a program.

*Source code -> compiler -> Assembler -> Object code -> Linker ->  
Executable file -> Loader*



## LINKER DIAGRAM:



## Advantages of Linker:

- 1.Code Reuse
- 2.Reduced Disk Space
- 3.Improved Security
- 4.Easier to Update Libraries
- 5.Portability

## Disadvantages of Linker:

- 1.Complexity
- 2.Performance
- 3.Security

## Applications of Linker:

- 1.Combining object files
  - 2.Dynamic linking
  - 3.Library management
  - 4.Modularity
- 

**Q9) what is loaders?  
sol)**



- loader is a vital component of an operating system that is accountable for loading programs and libraries.
- Absolute, Direct Linking, Bootstrap and Relocating are the types of loaders.
- It is special program that takes input of executable files from linker, loads it to main memory, and prepares this code for execution by computer.
- Loader allocates memory space to program.

**Difference between Linker and Loader:**

## Difference between Linker and Loader

S.No.	Linker	Loader
1	A linker is an important utility program that takes the object files, produced by the assembler and compiler, and other code to join them into a single executable file.	A loader is a vital component of an operating system that is accountable for loading programs and libraries.
2	It uses an input of object code produced by the assembler and compiler.	It uses an input of executable files produced by the linker.
3	The foremost purpose of a linker is to produce executable files.	The foremost purpose of a loader is to load executable files to memory.
4	Linker is used to join all the modules.	Loader is used to allocate the address to executable files.
5	It is accountable for managing objects in the program's space.	It is accountable for setting up references that are utilized in the program.

LINKER	LOADER
The main function of Linker is to generate executable files.	Whereas main objective of Loader is to load executable files to main memory.
The linker takes input of object code generated by compiler/assembler.	And the loader takes input of executable files generated by linker.
Linking can be defined as process of combining various pieces of codes and source code to obtain executable code.	Loading can be defined as process of loading executable codes to main memory for further execution.
Linkers are of 2 types: Linkage Editor and Dynamic Linker.	Loaders are of 4 types: Absolute, Relocating, Direct Linking, Bootstrap.
Another use of linker is to combine all object modules.	It helps in allocating the address to executable codes/files.
Linker is also responsible for arranging objects in program's address space.	Loader is also responsible for adjusting references which are used within the

Another use of linker is to combine all object modules.	It helps in allocating the address to executable codes/files.
Linker is also responsible for arranging objects in program's address space.	Loader is also responsible for adjusting references which are used within the program.

.....

## **Q10) what is Booting in Operating System?**

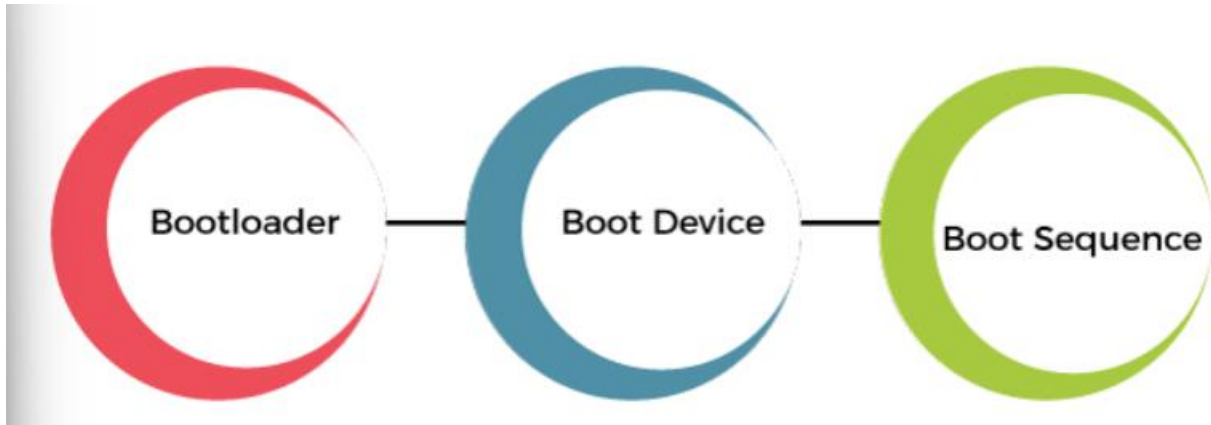
**sol)**

- Booting is the process of starting a computer.
- It can be initiated by hardware such as a button press or by a software command.
- After it is switched on, a CPU has no software in its main memory, so some processes must load software into memory before execution
- This may be done by hardware or firmware in the CPU or by a separate processor in the computer system.
- Restarting a computer also is called rebooting, which can be "hard" e.g., after electrical power to the CPU is switched from off to on, or "soft", where the power is not cut.

- Hard and soft booting can be initiated by hardware such as a button press or a software command.

## Sequencing of Booting:

1. Booting is a start-up sequence that starts the operating system of a computer when it is turned on.
2. A boot sequence is the initial set of operations that the computer performs when it is switched on.
3. Every computer has a boot sequence.



## Types of Booting:

- 1.Cold Booting
- 2.Warm Booting

## Booting Process in Operating System:

When our computer is switched on, it can be started by hardware such as a button press, or by software command



## Step 1:

- Once the computer system is turned on, BIOS (Basic Input /Output System) performs a series of activities or functionality tests on programs stored in ROM, called on POST (Power-on Self Test)

## Step 2:

- After the BIOS is done with pre-boot activities or functionality test, it read bootable sequence from CMOS (Common Metal Oxide Semiconductor)
- looks for master boot record in the first physical sector of the bootable disk as per boot device sequence specified in CMOS

For example, if the boot device sequence is:

1. Floppy Disk
2. Hard Disk
3. CDROM

### Step 3:

- After this, the master boot record will search first in a floppy disk drive.



- If not found, then the hard disk drive will search for the master boot record.

#### Step 4:

- The bootstrap loader first loads the IO.SYS file.
- After this, MSDOS.SYS file is loaded, which is the core file of the DOS operating system.

#### Step 5:

- After this, MSDOS.SYS file searches to find Command Interpreter in CONFIG.SYS file, and when it finds, it loads into memory.

#### Step 6:

- The last file is to be loaded and executed is the AUTOEXEC.BAT file that contains a sequence of DOS commands.
- 

## **Q11) resource management ? sol)**

- An operating system (OS) is basically a collection of software that manages computer hardware resources and provides common services for computer programs.
- The operating system is a crucial component of the system software in a computer system.

**types in resourcrce management:**

- 1.Process Management
- 2.Memory Management
- 3.Storage Management:
- 4.File-System Management
- 5.I/O Systems
- 6.Protection and Security

### 1. Process Management:

- A program in execution, as mentioned, is a process. A time-shared user program such as a compiler is a process.
- A process needs certain resources—including CPU time, memory, files, and I/O devices—to accomplish its task.
- Scheduling processes and threads on the CPUs

- Creating and deleting both user and system processes
- Suspending and resuming processes
- Providing mechanisms for process synchronization
- Providing mechanisms for process communication

## 2.Memory Management:

- The central processor reads instructions from main memory during the instruction-fetch cycle
- both reads and writes data from main memory during the data-fetch cycle.
- it accesses program instructions and data from memory by generating these absolute addresses.

- Allocating and de-allocating memory space as needed
- Deciding which processes and data to move into and out of memory
- Keeping track of which parts of memory are currently being used and who is using them

### 3.Storage Management:

- The operating system abstracts from the physical properties of its storage devices to define a logical storage unit, the file.
- The operating system maps files onto physical media and accesses these files via the storage devices
- The operating system is responsible for the following activities in connection with disk management:
  - 1.Free-space management
  - 2.Storage allocation

### 3.Disk scheduling

#### 4.File-System Management:

- Computers can store information on several different types of physical media. Magnetic disk, optical disk, and magnetic tape are the most common.
- A file is a collection of related information defined by its creator. Commonly, files represent programs and data.
- files are normally organized into directories to make them easier to use
- The operating system is responsible for the following activities in connection with file management:
  - 1.Creating and deleting files

2. Creating and deleting directories to organize files
3. Supporting primitives for manipulating files and directories
4. Mapping files onto secondary storage
5. Backing up files on stable (nonvolatile) storage media.

## 5.I/O Systems:

- The I/O subsystem consists of several components:
  - 1. A memory-management component that includes buffering, caching, and spooling
  - 2. A general device-driver interface
  - 3. Drivers for specific hardware devices