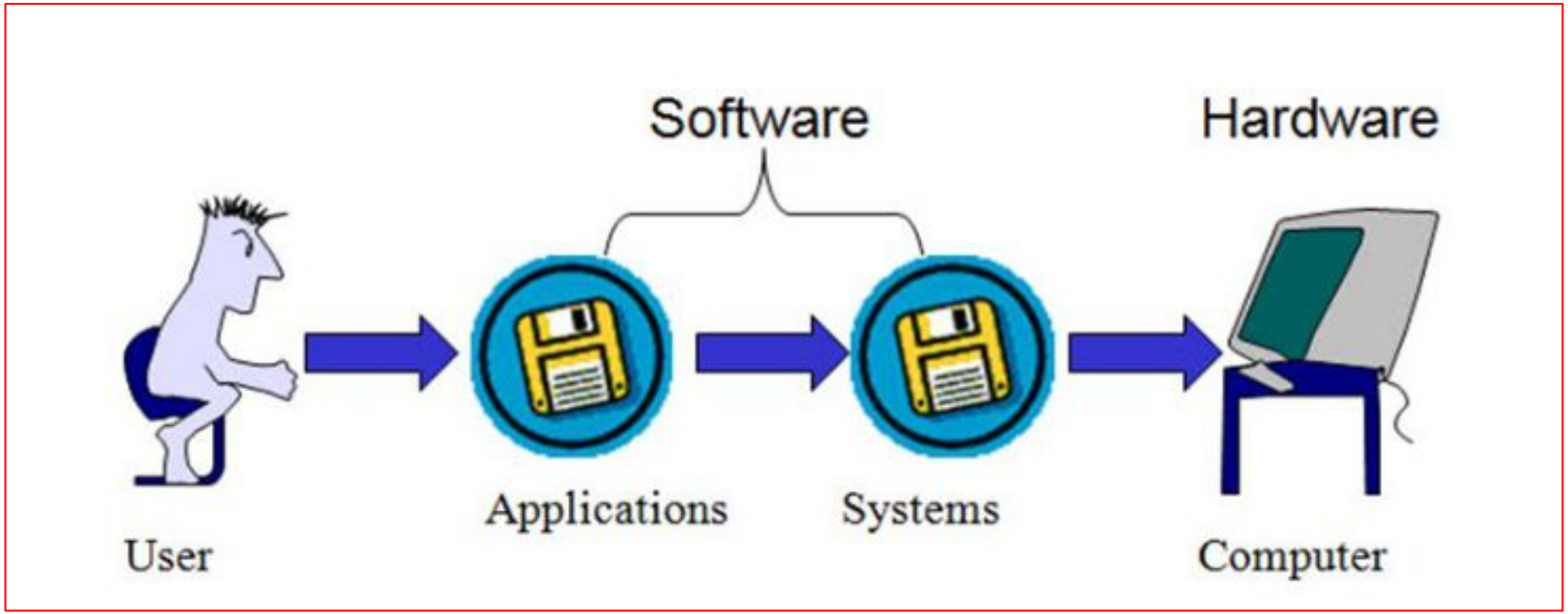
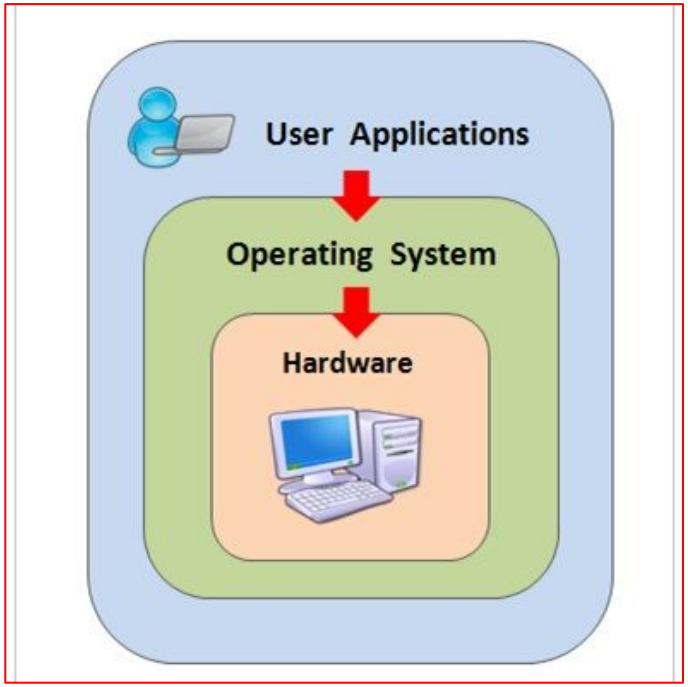
Operating System

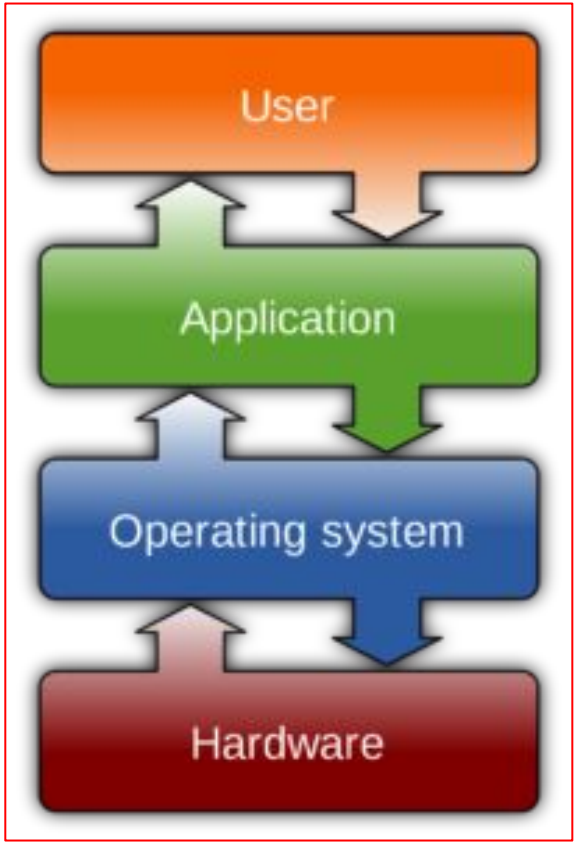
What is an Operating System?

What is an Operating System?

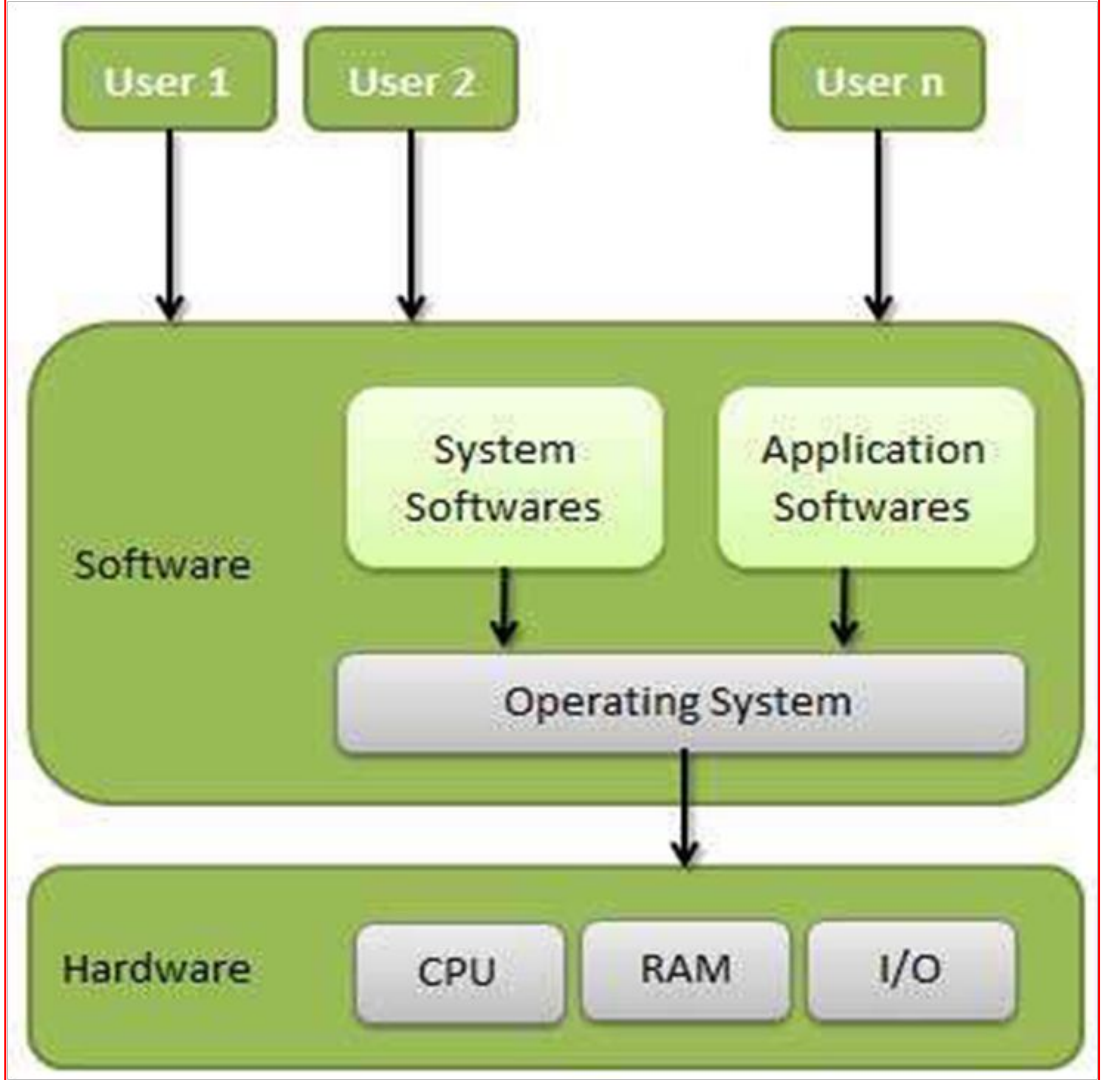


What is an Operating System?





Conceptual view of a Computer System :



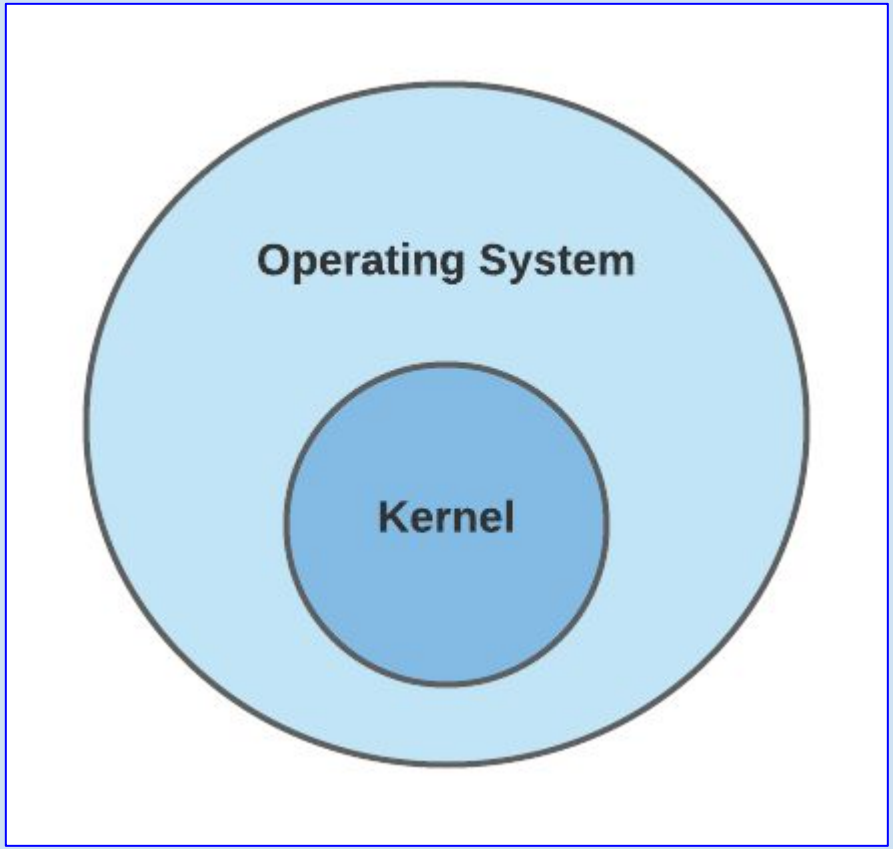
What is an Operating System?

- ★ An operating system (OS) is a program that acts as **an interface between the system hardware and the user.**
- ★ Moreover, it handles all the interactions between the software and the hardware.
- ★ All the working of a computer system depends on the OS at the base level. Further, it performs all the functions like handling memory, processes,files,device management etc.
- ★ Applications like Browsers, MS Office,Games,Anti-virus,Audio & Video players etc., need some environment **to run and perform its tasks**. OS provides that environment.

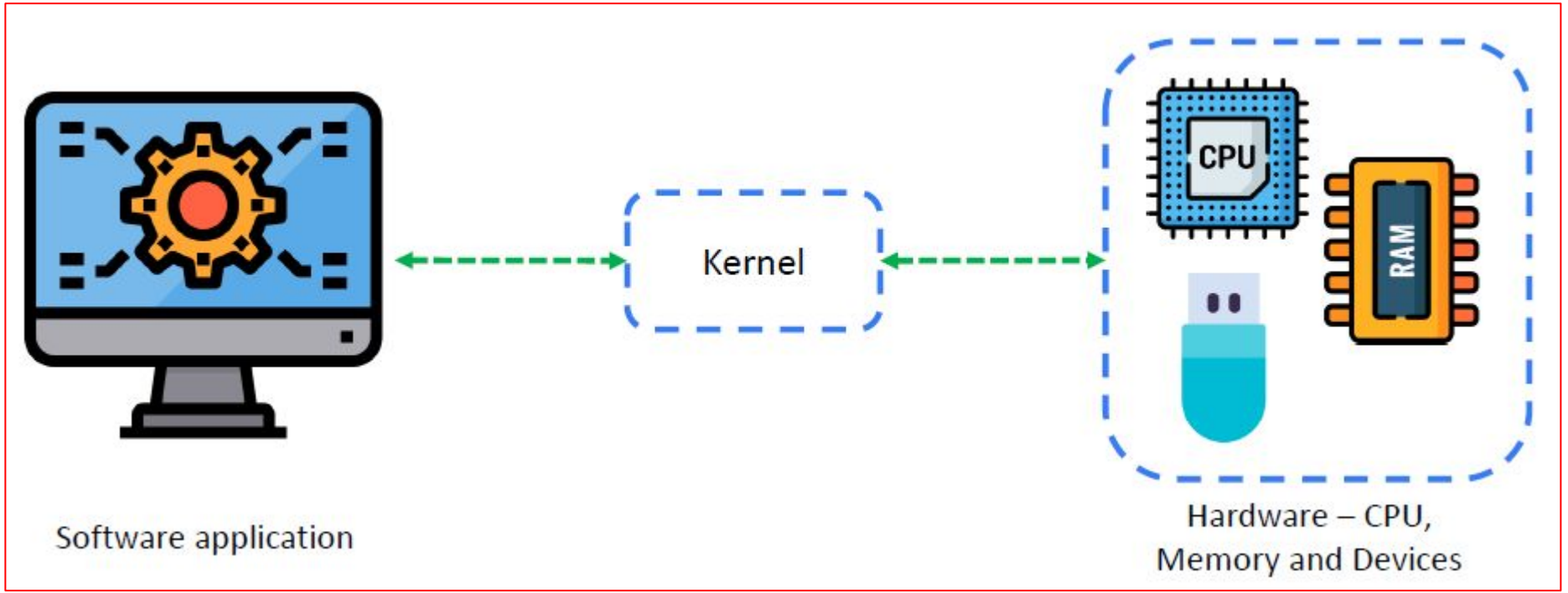
Eg:


Microsoft Windows
Android
iOS
macOS
Chrome OS
UNIX
Linux
Linux
Ubuntu
Fedora
BlackBerry OS
Examples of Operating System

What is kernel?



What is Kernel?



What is Kernel?

- ★ The kernel is **a computer program** which is the heart of a computer's operating system and generally has complete control over everything in the system.
- ★ This kernel is relatively **a small piece of code** that is embedded on the **hardware.**
- ★ Actually it is a collection of programs that are mostly written in c.
- ★ **Every Operating system has a kernel** (just one) that gets automatically loaded on to the memory as soon as the system is **booted**.
- ★ AS the kernel sits on the hardware it can directly communicate with the hardware.

Objectives of OS:

Convenience – An operating system improves the use of a machine. Operating systems enable users to get started on the things they wish to complete quickly without having to cope with the stress of first configuring the system.

Efficiency – An operating system enables the efficient use of resources. This is due to less time spent configuring the system.

Ability to evolve – An operating system should be designed in such a way that it allows for the effective development, testing, and introduction of new features without interfering with service.

Management of system resources – It guarantees that resources are shared fairly among various processes and users.

Functions of Operating System / :

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

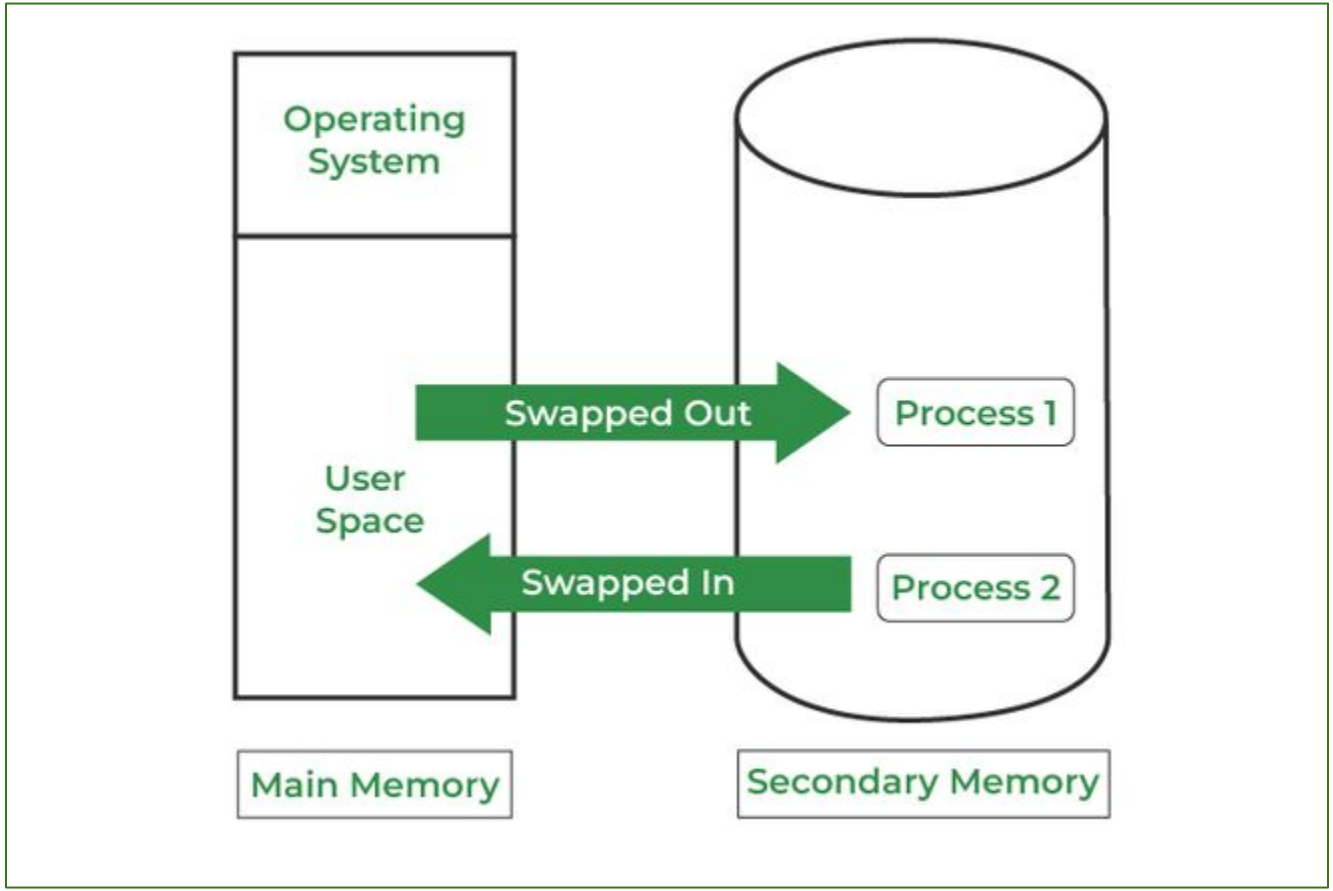
1. Memory Management

- ➔ It is the management of the **main or primary memory**.
- ➔ Whatever program is executed, it has to be present in the **main memory**.
- ➔ Main memory is a quick storage area that may be accessed directly by the CPU.
- ➔ When the program is completed, the memory region is released and can be used by other programs.
- ➔ Therefore, there can be more than one program present at a time. Hence, it is required **to manage the memory**.

The operating system:

1. Allocates and deallocates the memory.
2. Keeps a record of which part of primary memory is used by whom and how much.
3. Distributes the memory while multiprocessing.
4. In **multiprogramming**, the operating system selects which processes acquire memory when and how much memory they get

1. Memory Management



2.Processor Management/Scheduling

- A Program does nothing unless its instructions are executed by a CPU.
- A program in execution is called a process.
- In order to accomplish its task, process needs the computer resources.
- There may exist more than one process in the system which may require the **same resource at the same time.**
- Therefore, the operating system has to manage all the processes and the resources in a convenient and efficient way.
- When more than one process runs on the system the OS decides how and when a process will use the CPU. Hence, the name is also **CPU Scheduling**.
- The OS:
    - Allocates and deallocates processor to the processes.
    - Keeps record of CPU status.
    - Providing mechanisms for process synchronization.

Algorithms used for CPU scheduling are as follows:

1. First Come First Serve (FCFS)
2. Shortest Job First (SJF)
3. Round-Robin Scheduling
4. Priority-based scheduling etc.

3.Device Management

- Device management in an operating system means controlling the Input/Output devices like disk, microphone, keyboard, printer, magnetic tape, USB ports, camcorder, scanner, other accessories.
- The processes may require devices for their use. This management is done by the OS.
- The OS:
    - Allocates and deallocates devices to different processes.
    - Keeps records of the devices.
    - Decides which process can use which device for how much time.

4. File Management

- File management in operating system is nothing but software that handles or manages the files (binary, text, pdf, docs, audio, video, etc.) present in computer software.
- The file system in the operating system is capable of managing individual as well as groups of files present in the computer system.
- The file system in operating system tells us about the location, owner, time of creation and modification, type, and state of a file present on the computer system.
- The OS:
    - Keeps records of the status and locations of files.
    - Allocates and deallocates resources.
    - Decides who gets the resources.

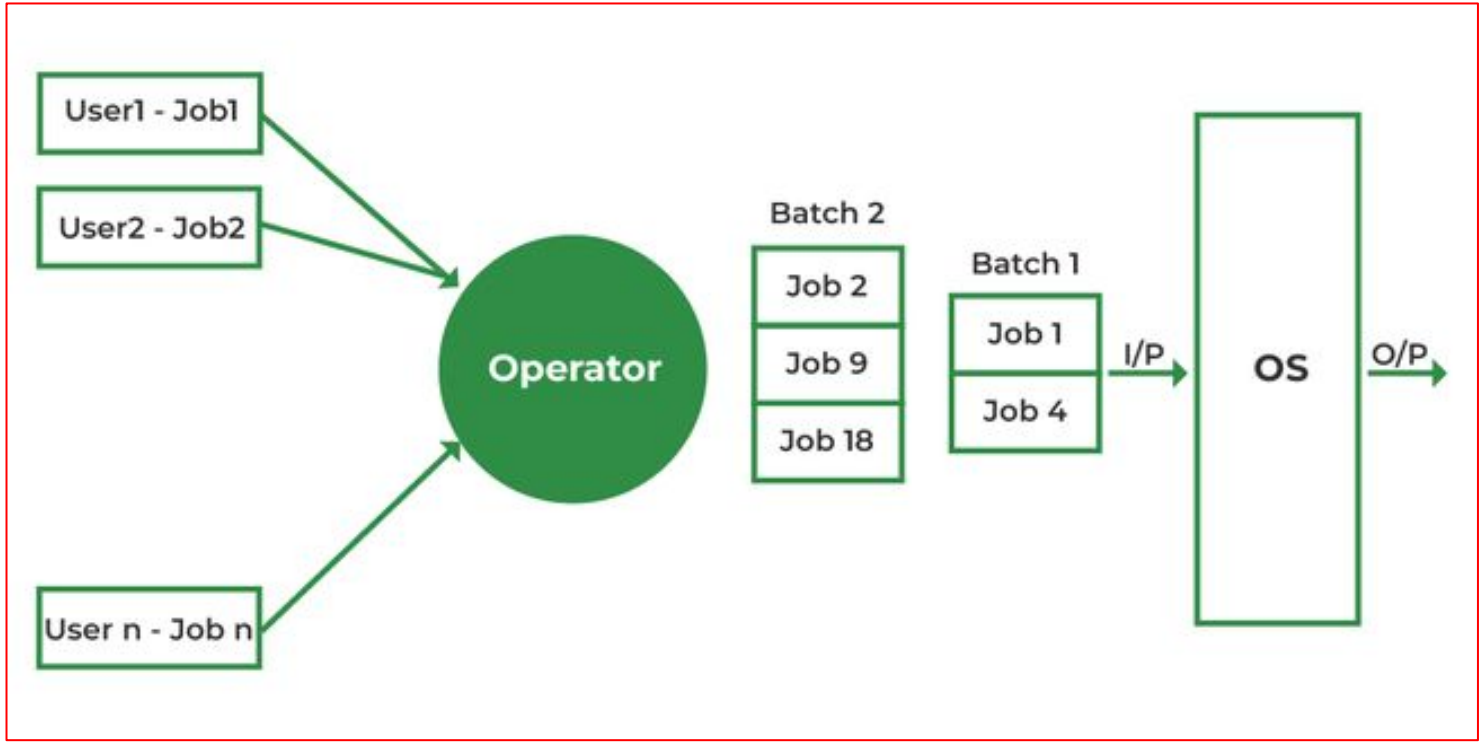
5. Security Management:

- ➔ The process of ensuring OS **availability, confidentiality, integrity** is known as operating system security.
- ➔ OS security refers to the processes or measures taken to protect the operating system from dangers, including **viruses, worms, malware, and remote hacker intrusions**.
- ➔ Operating system security comprises all preventive-control procedures that protect any system assets that could be stolen, modified, or deleted if OS security is breached.

Types of Operating Systems:

1. Batch operating system
2. Time-sharing operating systems
3. Distributed operating System
4. Network operating System
5. Real Time operating System

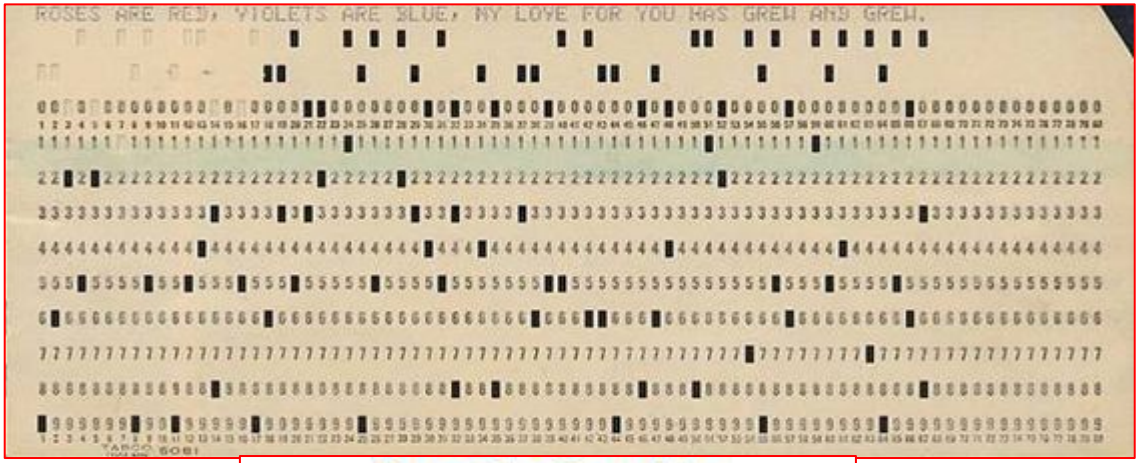
1.Batch operating system

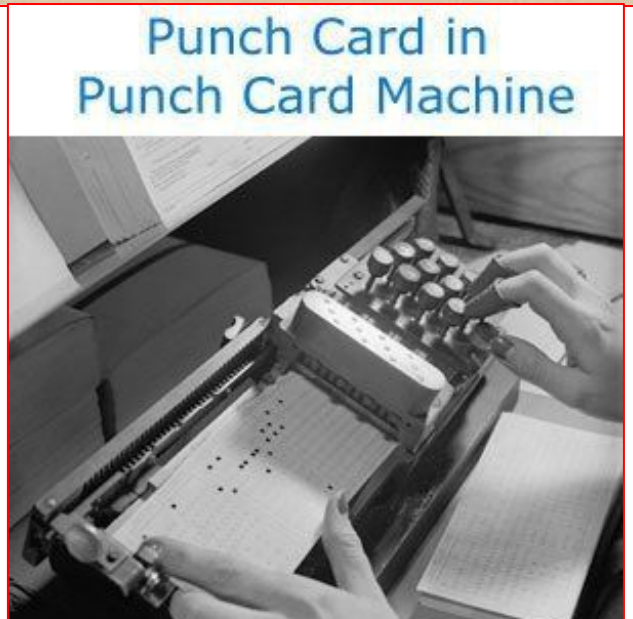


1.Batch operating system

- Batch processing was very popular in the **1970**s. The **jobs** were executed in **batches**.
- Users using batch operating systems do not interact directly with the computer.
- Each user prepares their job using an offline device like **a punch card** and submitting it to the computer operator.
- Jobs with similar requirements are grouped and executed as a group to speed up processing.
- Once the programmers have left their programs with the operator, they sort the programs with similar needs into batches.

Punch Card:

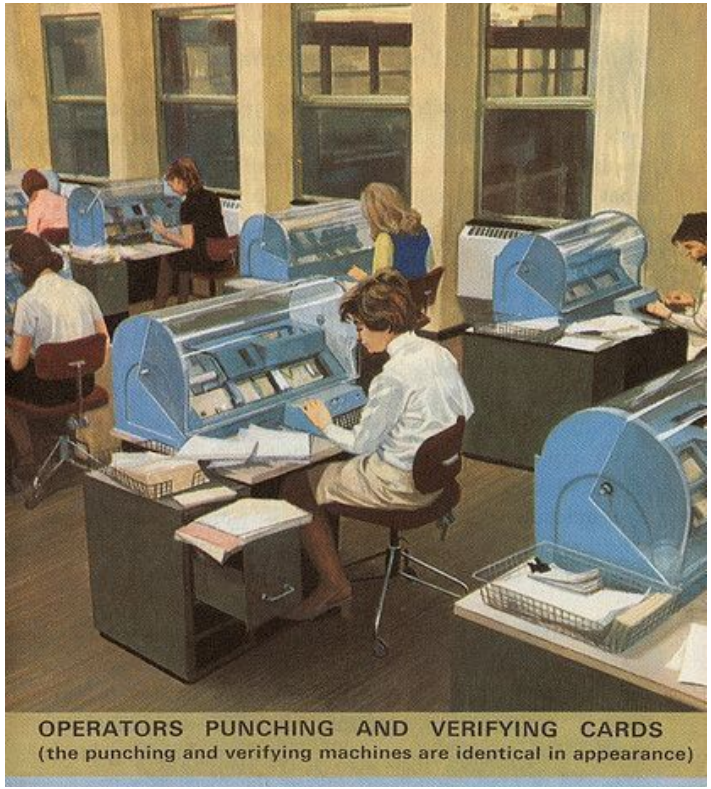

ROSES ARE RED, VIOLETS ARE BLUE, MY LOVE FOR YOU HAS GREW AND GREW.


Punch Card in
Punch Card Machine

How Programmers write program using Punch Card.(click the below link)

https://vishnulearning.com/mod/resource/view.php?id=69939

https://vishnulearning.com/mod/resource/view.php?id=69941



OPERATORS PUNCHING AND VERIFYING CARDS
(the punching and verifying machines are identical in appearance)

Advantages:

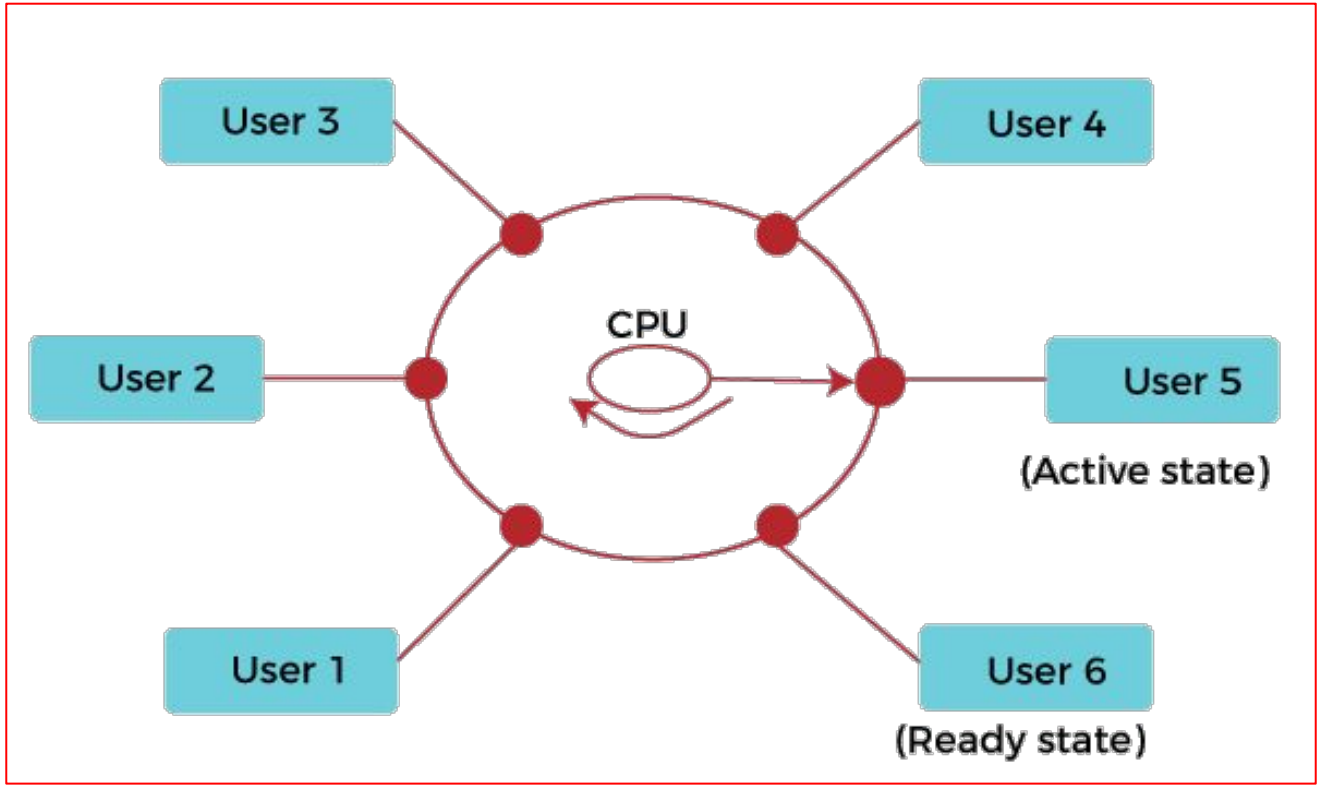
- This system can easily **manage large jobs** again and again.
- The batch process can be divided into several stages to **increase processing speed.**
- When a process is finished, the next job from the job spool is run without any user interaction.
- **CPU utilization gets improved.**

Disadvantages:

- When a job fails once, it must be scheduled to be completed, and it may take a long time to complete the task.
- Computer operators must have full knowledge of batch systems.
- The computer system and the user have no direct interaction.
- If a job enters an infinite loop, other jobs must wait for an unknown period of time.

Applications: Payroll systems, Bank statements, Telephone bill generation

2.Time-sharing operating systems



2.Time-sharing operating systems

- Time-sharing is a technique that enables many people located at various terminals to use a particular computer system simultaneously.
- Time-sharing operating systems or **Multitasking** is a logical extension of Multiprogramming.
- In this time-sharing Operating system, many processes are allocated with computer resources in respective time slots.
- In this, the processor's time is shared with multiple users. That's why it is called a time-sharing operating system.
- It has a fixed time slice for the different processes. Its main purpose is interactive response time.
- This time duration is known as time slot, time slice, or quantum.

- only one process runs at a time but, the switching between the processes is very fast. Hence, the user feels that all the processes are running at the same time.

Advantages:

- Response time of CPU reduces.
- Idle time of CPU reduces.
- Each task/process gets an equal time slot to execute.

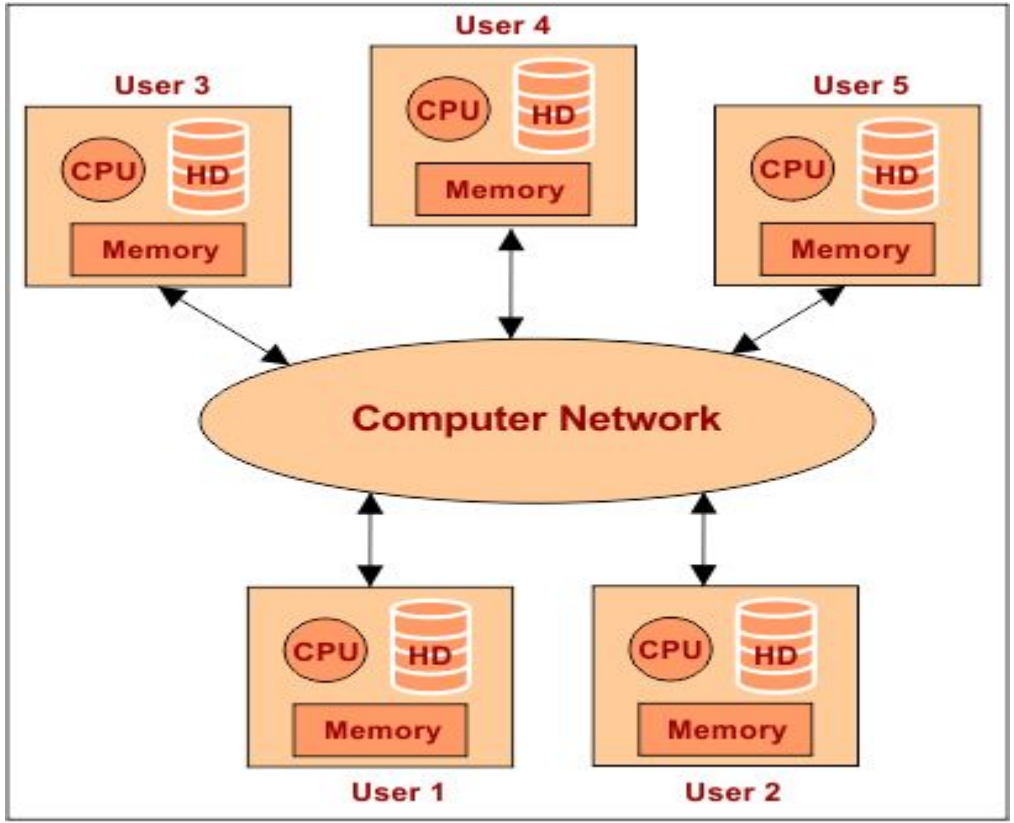
Disadvantages:

- Communication is very important to maintain. Lack of communication can affect the whole working.
- Requires high specification of hardware
- It has a problem of reliability

Some examples of Time-sharing operating systems are:

- ★ UNIX
- ★ Multics
- ★ Linux
- ★ Windows 2000 server
- ★ Windows NT server

3.Distributed Operating System:



Distributed OS

3. Distributed Operating System

- A distributed operating system is one in which several computer systems connected through a single communication channel.
- Moreover, these systems have their individual processors and memory.
- These processors communicate through high-speed buses or telephone lines.
- These individual systems that connect through a single channel are considered as a single unit.
- We can also call them loosely coupled systems. The individual components or systems of the network are nodes.

Types of Distributed Operating System

1. Client/Server Systems

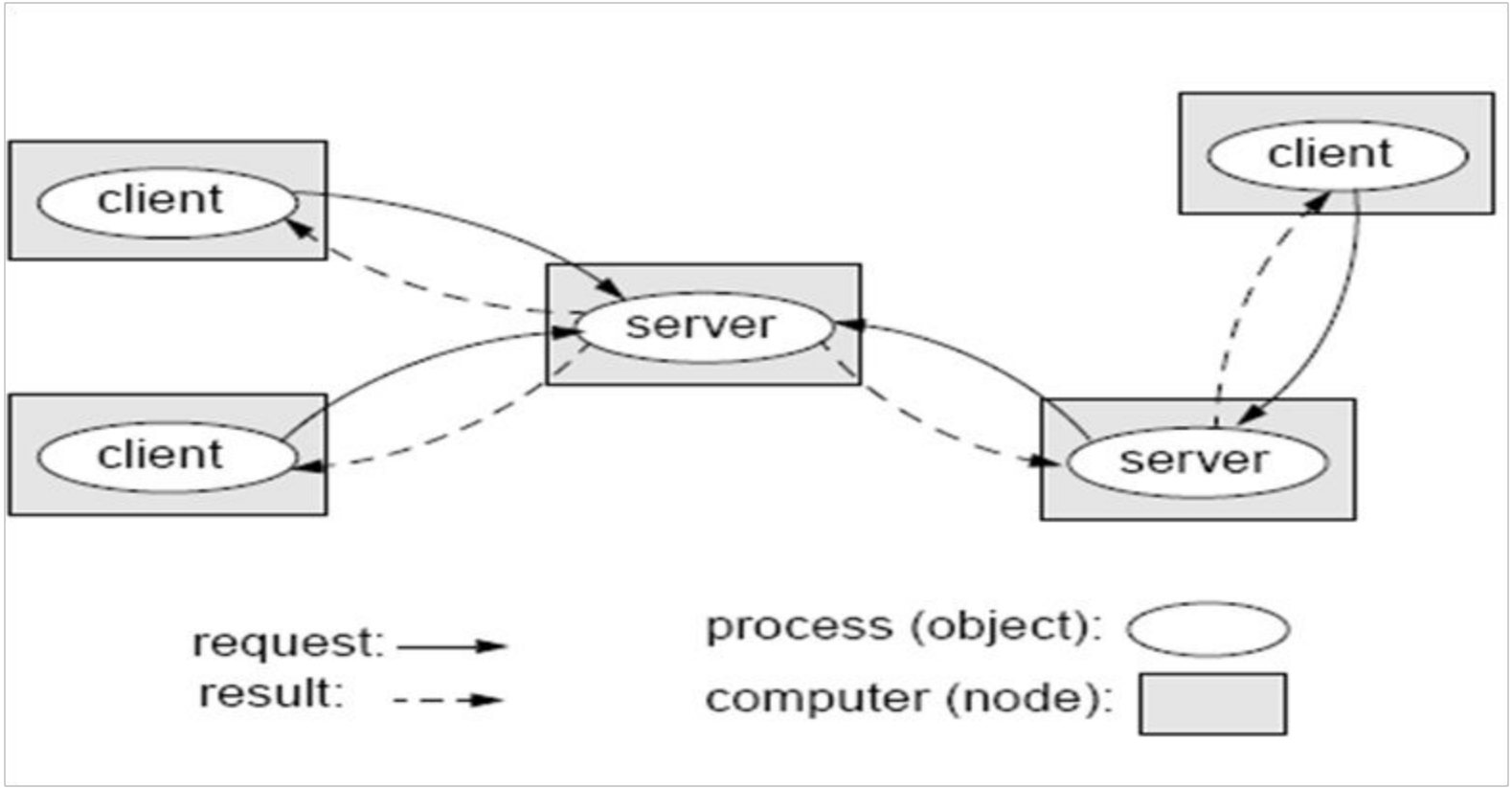
★ In this system, the client requests the server for a resource. On the other hand, the server provides this resource to the client.

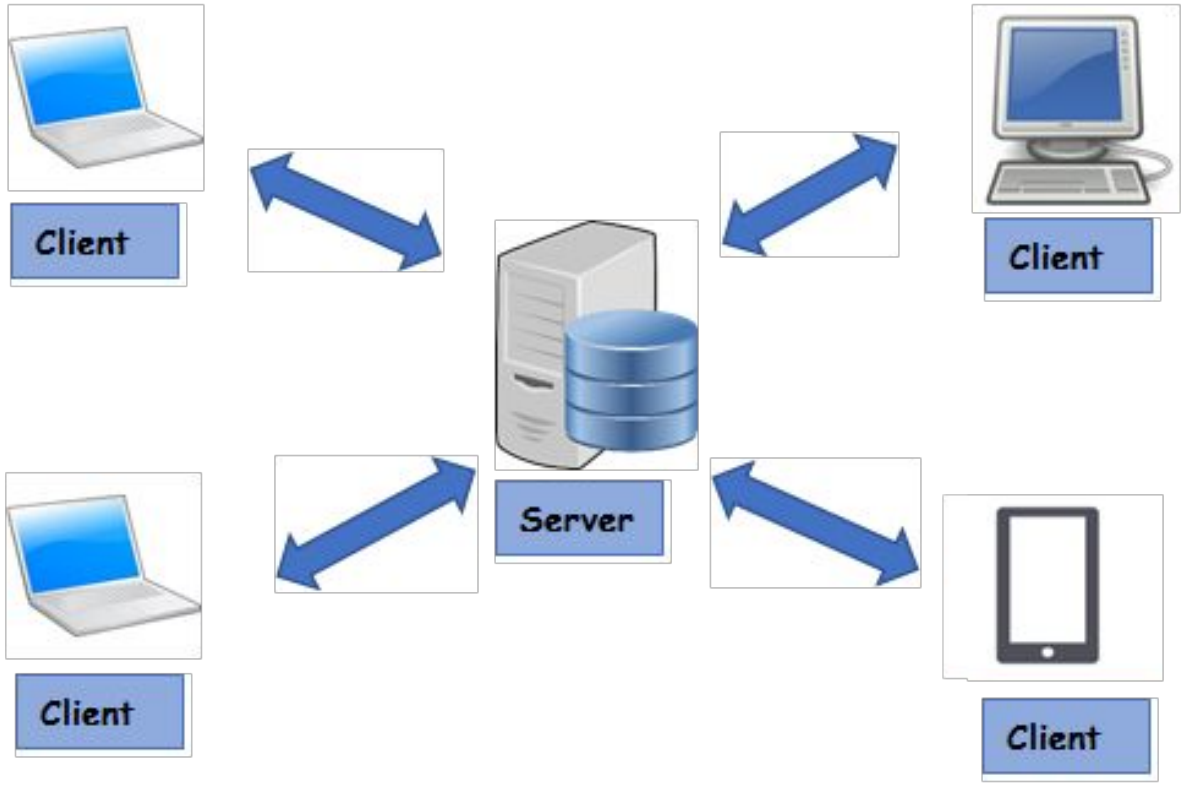
★ One client contacts only a single server at a time. Whereas a single server can deal with multiple clients simultaneously.

★ The clients and servers connect through a computer network in the system.

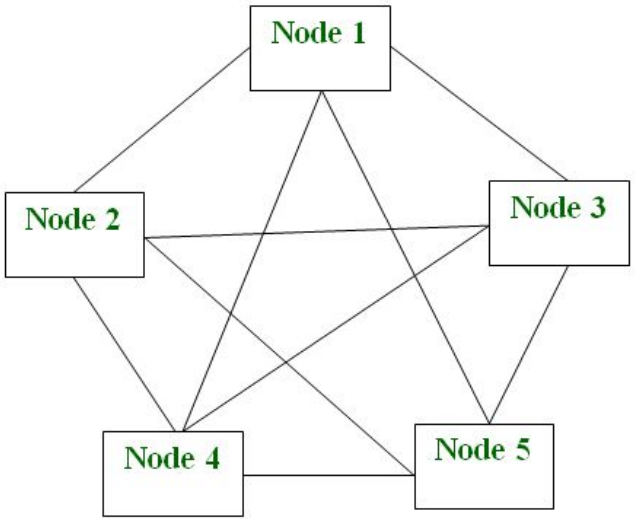
Examples of computer applications that use the client–server model are **email, network printing, and the World Wide Web**.

Client/Server Systems

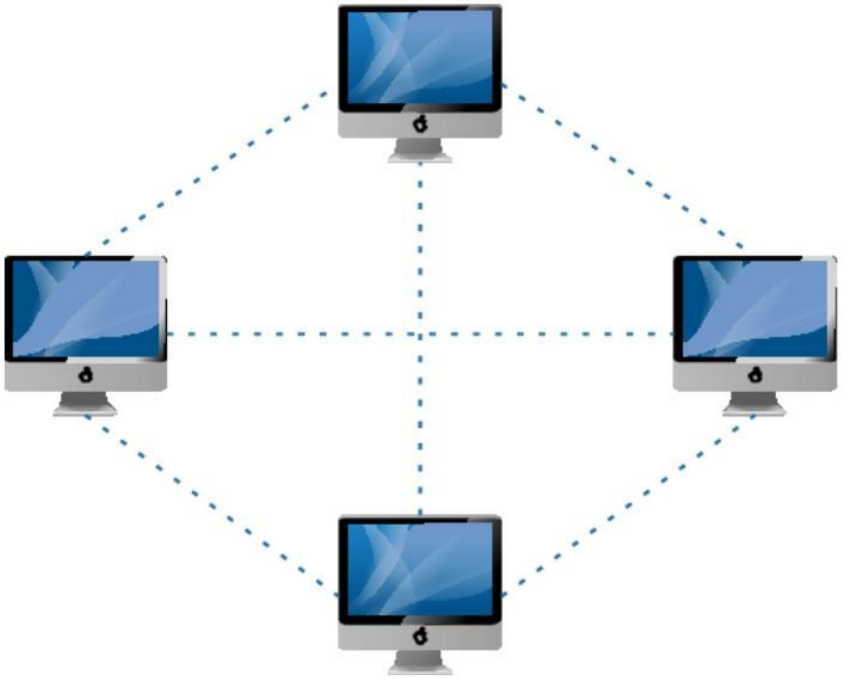



Client
Client
Server
Client
Client

2. Peer to Peer Systems



P2P Architecture



2. Peer to Peer Systems

- ➔ In this system, the nodes play an important role.
- ➔ All the work equally divides among the nodes.
- ➔ Furthermore, these nodes can share data or resources as per the requirement. Again, they require a network to connect.
- ➔ Here **each node acts as a server** and thus **there is no central server in the network**. This allows the sharing of a huge amount of data

Examples of computer applications that use the Peer to Peer model are

- **File Sharing**
- **IP Telephony (Skype is one good example of a P2P application in VoIP)**

Examples of Distributed Operating System:

- **AIX operating system** for **IBM RS/6000** computers.
- **Solaris operating system** for **SUN multiprocessor workstations.**
- **Mach/OS** is a multitasking and multithreading **UNIX compatible operating system.**
- **Windows**

Advantages:

★ The load on the system decreases.

★ If one system stops it will not affect the other.

★ The system shares a workload that makes calculations easy.

Disadvantages:

★ The cost for set up is more.

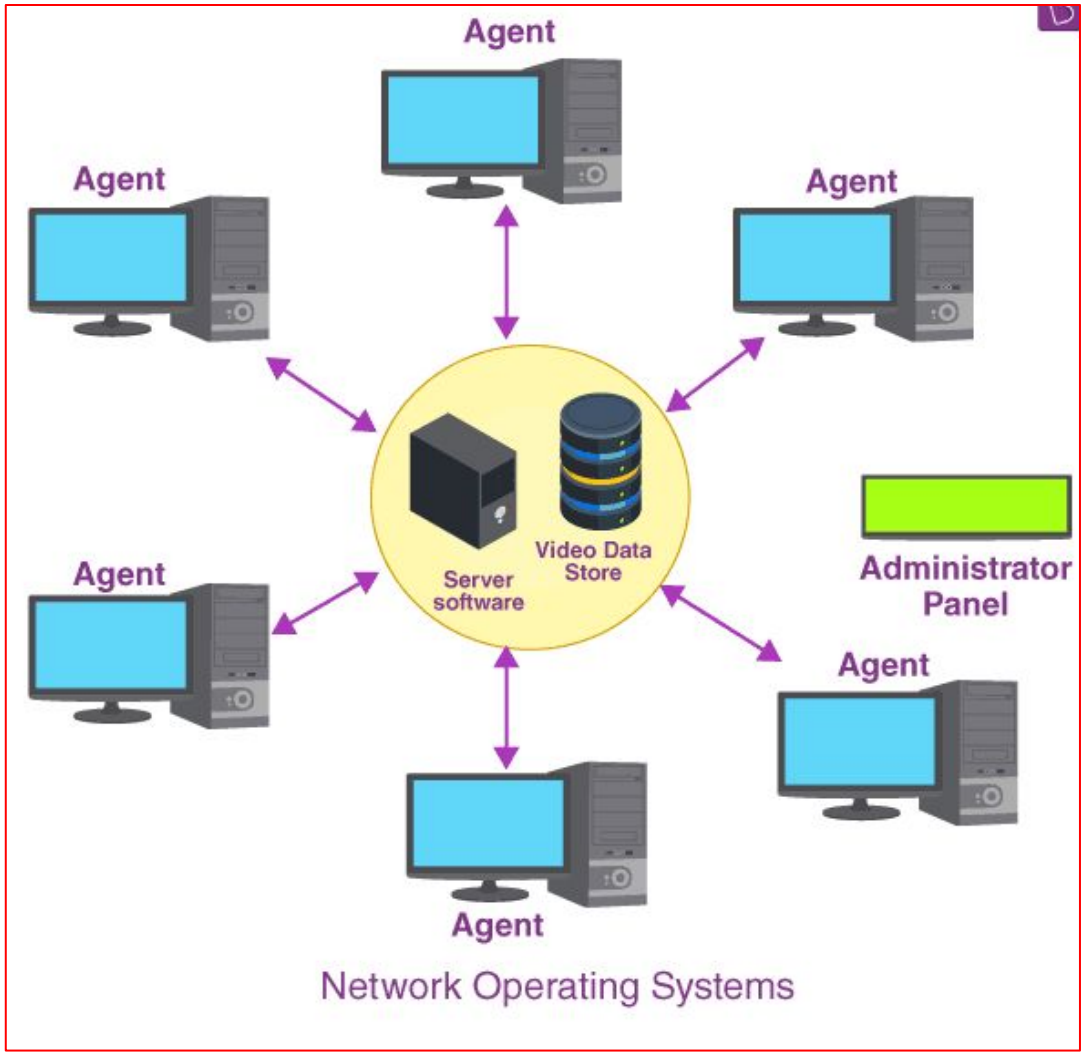
★ Failure of **the main system(Server)** will affect the whole system.

★ Programming is complex.

4.Network operating system (NOS) :

Examples of Network Operating systems :

1. **Microsoft Windows Server**
2. **UNIX/Linux**
3. **Artisoft's LANtastic**



4.Network operating system (NOS)

- a network operating system(NOS) is software that connects multiple devices and computers on the network and allows them to share resources on the network. or
- An Operating system, which includes software and associated protocols to communicate with other computers via a network conveniently and cost-effectively, is called Network Operating System..

Following are the main functions of NOS :

★ Printers and application sharing on the network.

★ File systems and database sharing.

★ Provide good security by using functionality like user authentication and access control.

★ Create backups of data.

Types of Network operating systems :

Peer to Peer :

Peer-to-peer network operating systems allow sharing resources and files with small-sized networks and having fewer resources. In general, peer-to-peer network operating systems are used on LAN.

Client/server :

Client-server network operating systems provide users access to resources through the central server. This NOS is too expensive to implement and maintain. This operating system is good for the big networks which provide many services.

Advantages of Network operating systems :

- ➔ Highly stable due to central server.
- ➔ Provide good security.
- ➔ Upgradation of new technology and hardware can be easily implemented in the network.
- ➔ Provide remote access to servers from different locations.

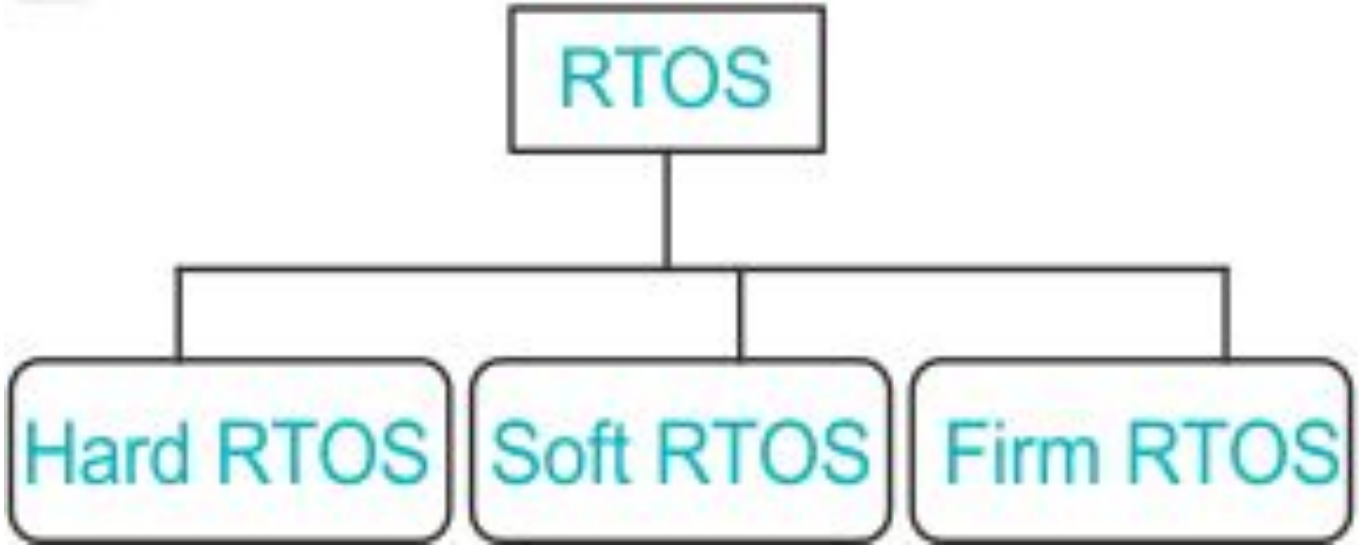
Disadvantages of Network operating systems :

- ➔ Depend on the central location to perform the operations.
- ➔ High cost to buying server.
- ➔ Regular updating and maintenance are required.

5.Real Time operating System

- The real-time operating systems are used in real-time systems where the time constraints are fixed and followed strictly.
- This means that the time for processing and responding is **very small.**
- Moreover, the system should perform the given task in a fixed time otherwise, it results in a system failure.
- **Response Time** is the time within which the system takes the input, processes the data, and gives the results. Moreover, they are used in systems like **robots, missile launches, airplanes, etc.**

Types of RTOS:



1. Hard Real-Time Systems:

In this, the time constraint is very short and strict. Even seconds of delay is not acceptable. Therefore, it is compulsory to complete the task within the given time only.

Examples : **Airplanes systems, Medical treatment systems**, etc.

2. Soft Real-Time Systems:

As the name suggests, the system handles the deadlines softly. This means that if there are small delays in the system, it is acceptable.

Examples: **Online Transaction systems, Web browsing, Computer games**, etc

3.Firm Real-Time Systems:

- Firm real-time tasks are such type of real-time tasks which are associated with time bound and the task need to produce the result within the deadline.
- Although firm real-time task is different from hard real-time task as in hard real-time once deadline is crossed and task is not completed, system fails but in case of firm real-time task even after the passing of deadline, system does not fail.

Examples :**1. Video conferencing 2. Satellite based tracking**

Advantages of RTOS

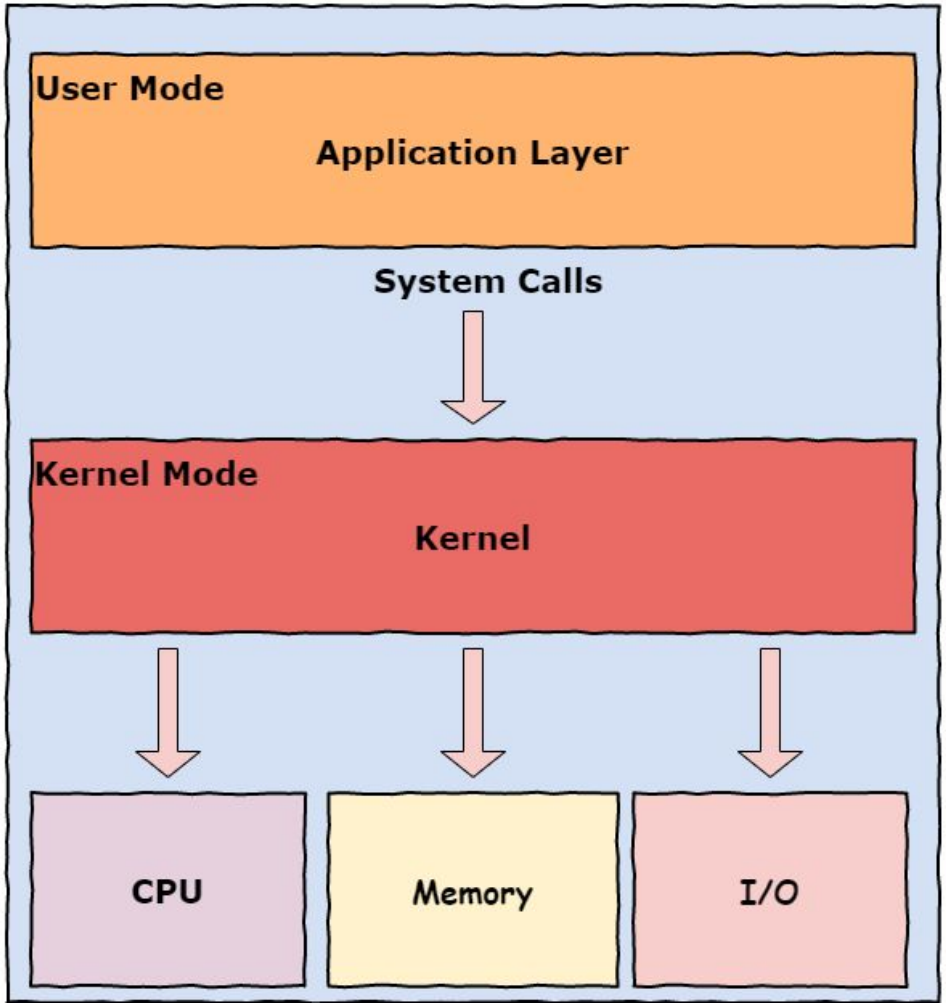
- They have no errors.
- Due to their small size, they can be easily added to other systems.

Disadvantages of RTOS

- Algorithms are complex.
- System resources are expensive.

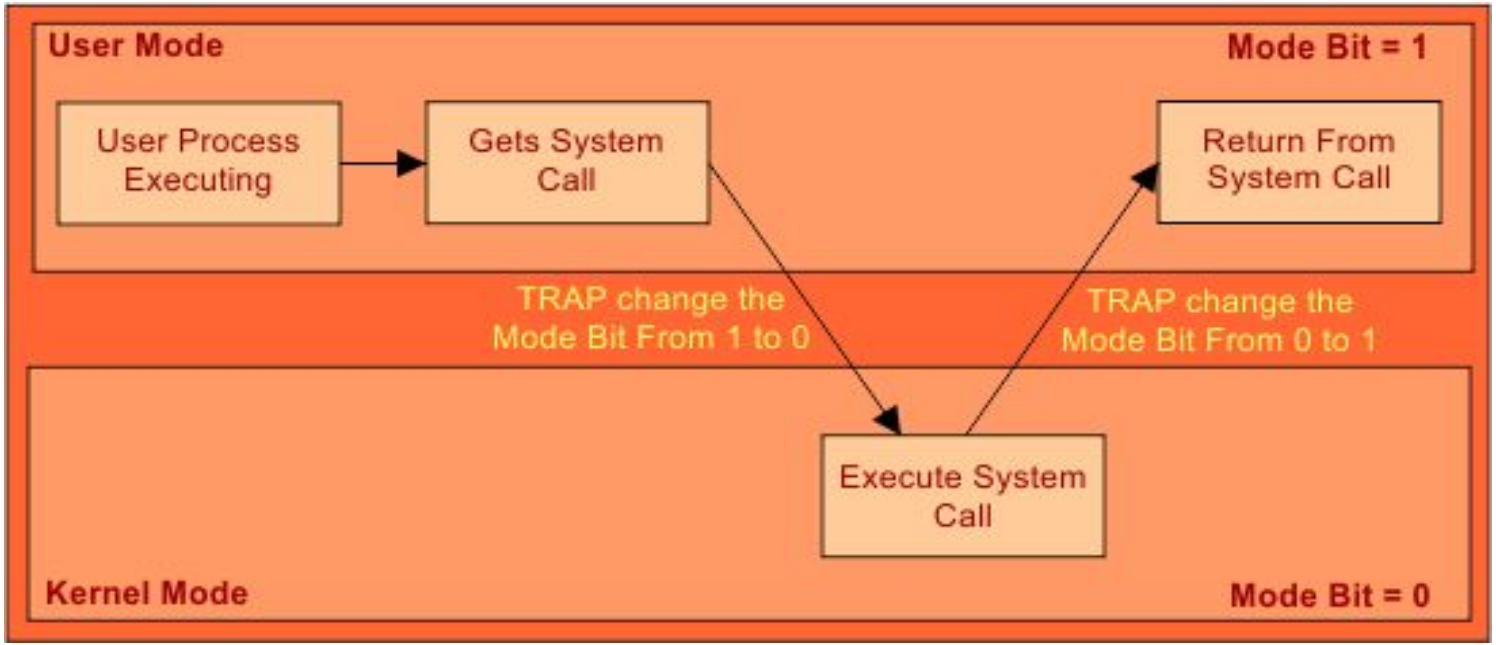
Operating-System Operations (or)
Dual Mode Operations in Operating System

Operating-System Operations [Dual Mode Operations in Operating System]



- The dual-mode operations in the operating system protect the operating system from illegal users.
- To ensure proper operating system execution, we must differentiate between machine code execution and user-defined code.
- The processor switches between the two modes depending on what type of code is running on the processor.
- **Applications** run in **user mode**, and **core operating system components** run in **kernel mode.**
- Modern operating systems have **two basic modes** in which they can execute a certain program:
    1. **The user-mode [Mode bit is 1]**
    2. **The kernel-mode. [Mode bit is 0]**

Mode bit is required to identify in which particular mode the current instruction is executing. If the mode bit is 1, it operates user mode, and if the mode bit is 0, it operates in kernel mode.



- When the computer system executes on behalf of a user application, the system is in user mode.
- However, when a user application requests a service from the operating system via a system call, it must transition from user to kernel mode to fulfill the request. As we can say, this architectural enhancement is useful for many other aspects of system operation.
- At system boot time, the hardware starts in kernel mode.
- The operating system is then loaded and starts user applications in user mode.
- Whenever a trap or interrupt occurs, the hardware switches from user mode to kernel mode, changing the mode bit's state to 0.
- Thus, whenever the operating system gains control of the computer, it is in kernel mode.
- The system always switches to user mode by setting the mode bit to 1 before passing control to a user program.

The User-Mode:

- ➔ Every user process operates under the user mode.
- ➔ In this mode, processes do not have direct access to the RAM or other hardware resources and have to make system calls to the underlying APIs to access these resources.

2. Kernel-Mode:

- ➔ The system starts in the kernel mode when it **boots up.**
- ➔ The kernel mode has **direct access** to all the underlying **hardware resources**.
- ➔ In the kernel mode, **all memory addresses are accessible and all CPU instructions are executable.**
- ➔ Kernel mode is usually reserved for drivers which need finer control over the hardware they are operating on.

System call and Types of system calls.

- A System call is the programmatic way in which a computer program requests a service from the kernel of the operating system.
- System call provides the services of the operating system to the user programs via Application Program Interface(API).
- 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.
- All programs needing resources must use system calls.

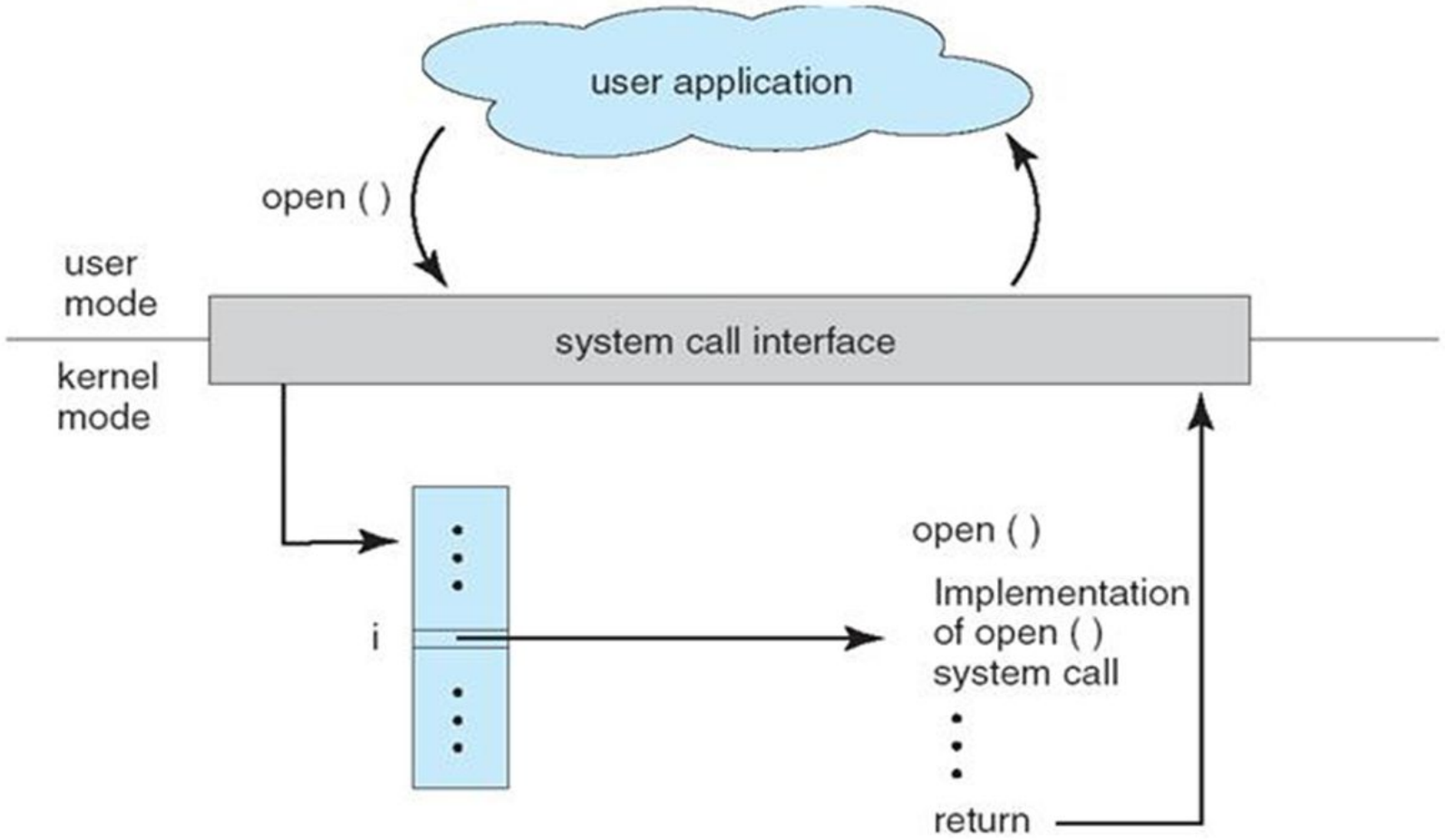
Services Provided by System Calls :

- Process creation and management
- Main memory management
- File Access, Directory and File system management
- Device handling(I/O)
- Protection
- Networking, etc.

Three most common APIs are:

1. Win32 API for Windows,
2. POSIX API (all versions of UNIX, Linux, and Mac OS X), and
3. Java API for the Java virtual machine (JVM)

The handling of a user application invoking the open() system call



System Call Implementation

- Typically, a number associated with each system call
- System-call interface maintains a table indexed according to these Numbers
- The system call interface invokes intended system call in OS kernel and returns status of the system call and any return values.
- The caller need know nothing about how the system call is implemented.Just needs to obey API and understand what OS will do as a result Call.
- Most details of OS interface hidden from programmer by API

Types of System Calls

1. Process control
2. File management
3. Device management
4. Information maintenance
5. Communication

1. **Process control:**
   - Create process, Terminate process
   - End, Abort
   - Load, Execute
   - Get process attributes, Set process attributes
   - Wait for time
   - Wait event, Signal event
   - Allocate memory, Free memory

2. **File management:**
   - Create file, Delete file
   - Open, Close
   - Read, Write, Reposition
   - Get file attributes, Set file attributes

- **Device management:**
  - Request device, Release device
  - Read, Write, Reposition
  - Get device attributes, Set device attributes
  - Logically attach devices, Logically detach devices
- **Information maintenance:**
  - Get time or date, Set time or date
  - Get system data, Set system data
  - Get process, file, or device attributes, Set process, file, or device attributes

- **Communications:**
  - Create communication connection, Delete communication connection
  - Send messages, Receive messages
  - Transfer status information
  - Attach remote devices, Detach remote devices

Eg:

Windows and Unix System Calls :

	Windows	Unix
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Manipulation	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device Manipulation	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()

Eg:

Windows and Unix System Calls :

Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()
Protection	SetFileSecurity() InitlializeSecurityDescriptor() SetSecurityDescriptorGroup()	chmod() umask() chown()

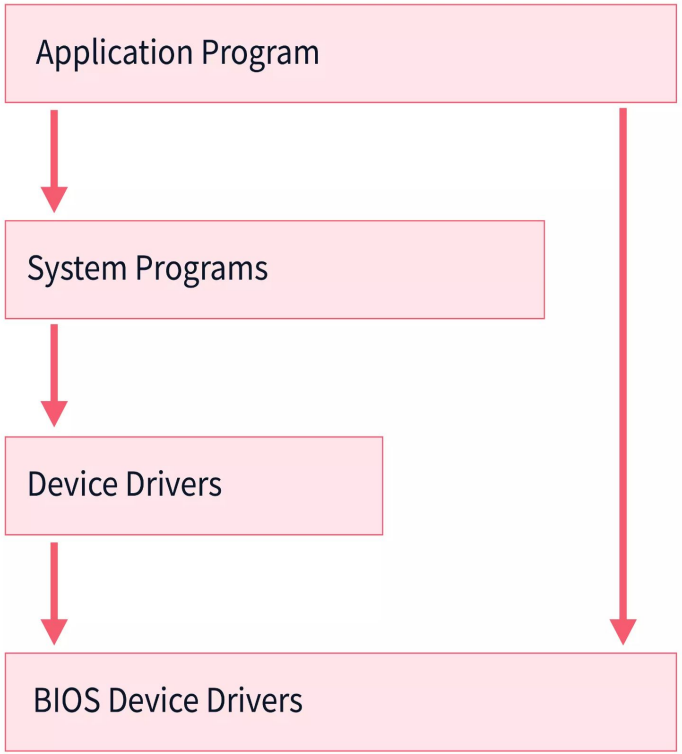
Different approaches or Structures of Operating Systems

Simple structure:

- ★ Such operating systems do not have well defined structure and are small, simple and limited systems.
- ★ The interfaces and levels of functionality are not well separated. MS-DOS is an example of such operating system.
- ★ In MS-DOS application programs are able to access the basic I/O routines.
- ★ These types of operating system cause the entire system to crash if one of the user programs fails.

Simple structure:

Eg: MS-DOS



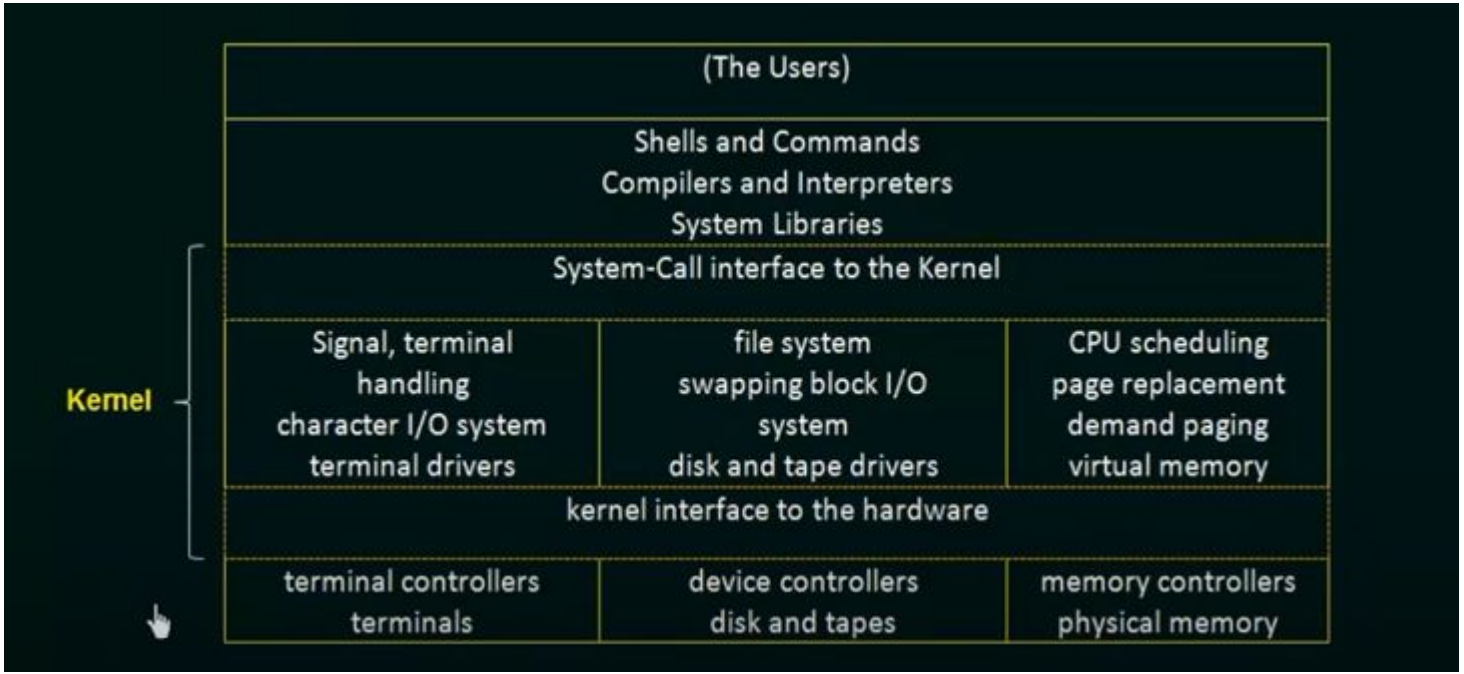
Advantages of Simple structure:

- It delivers better application performance because of the few interfaces between the application program and the hardware.
- Easy for kernel developers to develop such an operating system.

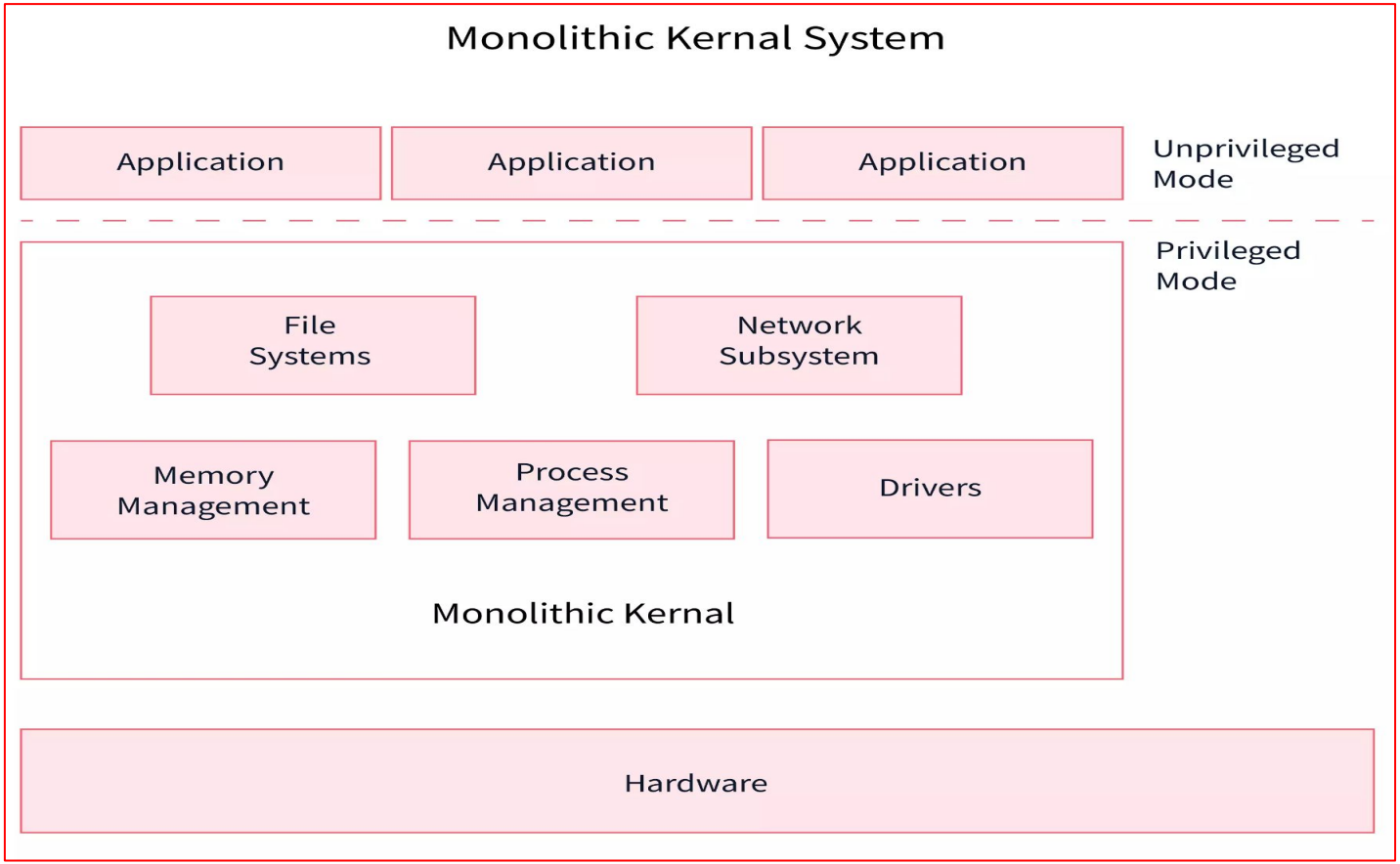
Disadvantages of Simple structure:

- The structure is very complicated as no clear boundaries exists between modules.
- It does not enforce data hiding in the operating system.

2.Monolithic Structure



Eg: UNIX OS


Monolithic Kernal System
Application
Application
Application
Unprivileged Mode
Privileged Mode
File Systems
Network Subsystem
Memory Management
Process Management
Drivers
Monolithic Kernal
Hardware

- 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.
- The kernel is the heart of a computer operating system (OS). Kernel delivers basic services to all other elements of the System.
- Kernel serves as the primary interface between the Operating System and the hardware.
- In monolithic systems, kernels can directly access all the resources of the operating System like physical hardware, exp Keyboard, Mouse etc.
- This is an **outdated operating system** that was used in **banks** to accomplish minor activities.

Advantages of Monolithic structure:

- It is **simple to design and implement** because all operations are managed by kernel only, and layering is not needed.
- As services such as memory management, file management, process scheduling, etc., are implemented in the same address space, the execution of the monolithic kernel is relatively fast as compared to normal systems. **Using the same address** saves time for address allocation for new processes and makes it faster.

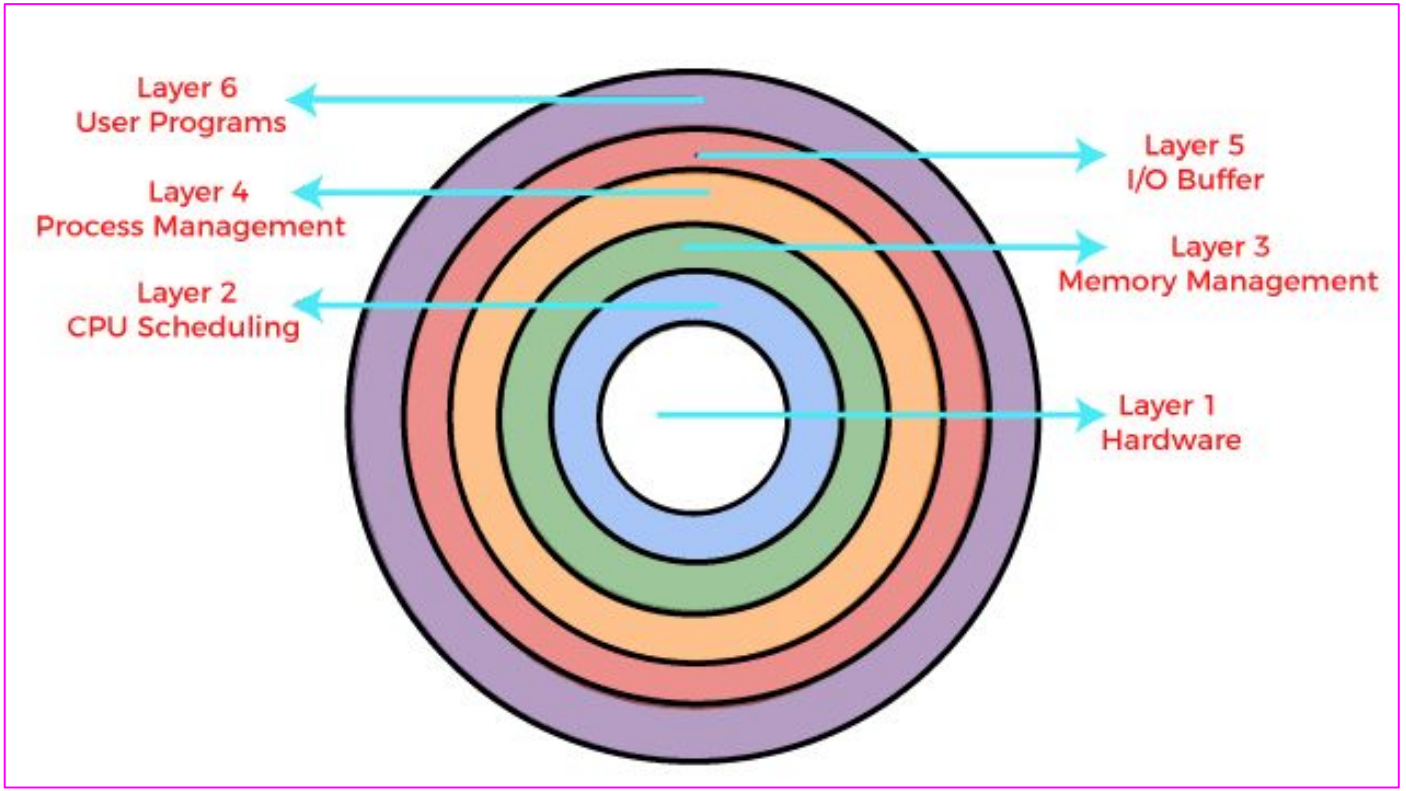
Disadvantages of Monolithic structure:

- If any service in the monolithic kernel fails, the entire System fails because, in address space, the services are connected to each other and affect each other.
- It is not flexible, and to introduce a new service

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 only.
- In this approach, functionalities of each layer are **isolated**
- This simplifies the debugging process as if lower level layers are debugged and an error occurs during debugging then the error must be on that layer only as the lower level layers have already been debugged.
- so all lower-level layered is debugged, and then the upper layer is checked.

3.Layered structure:



Advantages of Layered Structure:

- Each layer has its functionalities, so work tasks are **isolated**, and abstraction is present up to some level.
- **Debugging is easier** as lower layers are debugged, and then upper layers are checked.

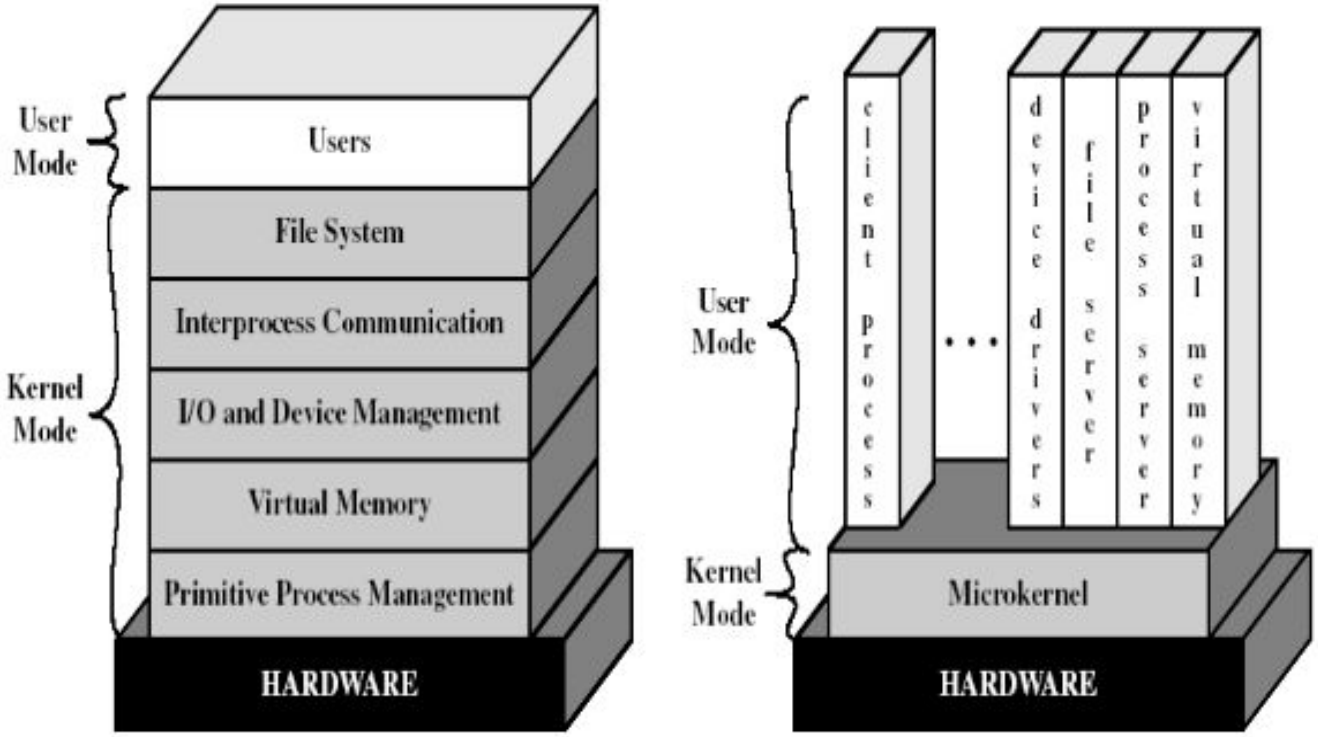
Disadvantages of Layered Structure:

- In Layered Structure, layering causes **degradation in performance.**
- It takes careful planning to construct the layers since higher layers only utilize the functions of lower 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.
- Advantages of this structure are that all new services need to be added to user space and does not require the kernel to be modified.
- Thus it is **more secure and reliable** as if any service fails then rest of the operating system remains untouched.
- **Mac OS** is an example of this type of OS.

Layered vs. Microkernel Architecture



(a) Layered kernel

(b) Microkernel

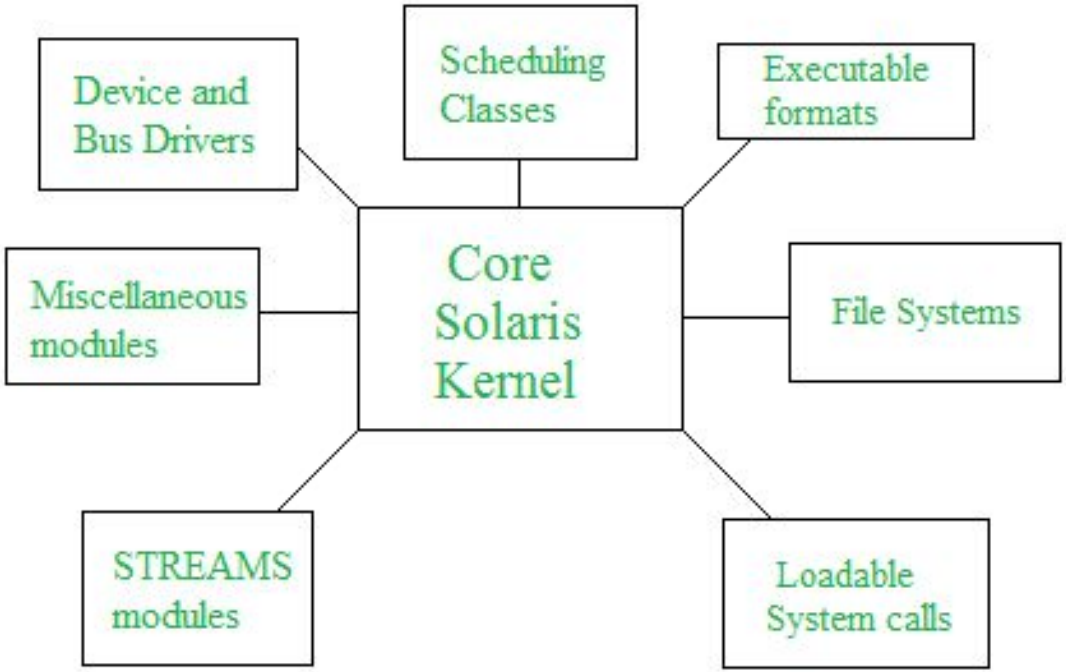
Advantages of Micro-kernel structure:

- It allows the operating system to be portable between platforms.
- As each Micro-Kernel is isolated, it is safe and trustworthy.
- Because Micro-Kernels are smaller, they can be successfully tested.
- If any component or Micro-Kernel fails, the remaining operating System is unaffected and continues to function normally.

Disadvantages of Micro-kernel structure:

- Increased inter-module communication **reduces system performance**.
- System is complex to be constructed.

5.Modular structure :



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(seem like)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.
- For example **Solaris OS** is organized as shown in the figure