Book — Silberschatz, Galvin and Gagne, Operating System Concepts, Wiley

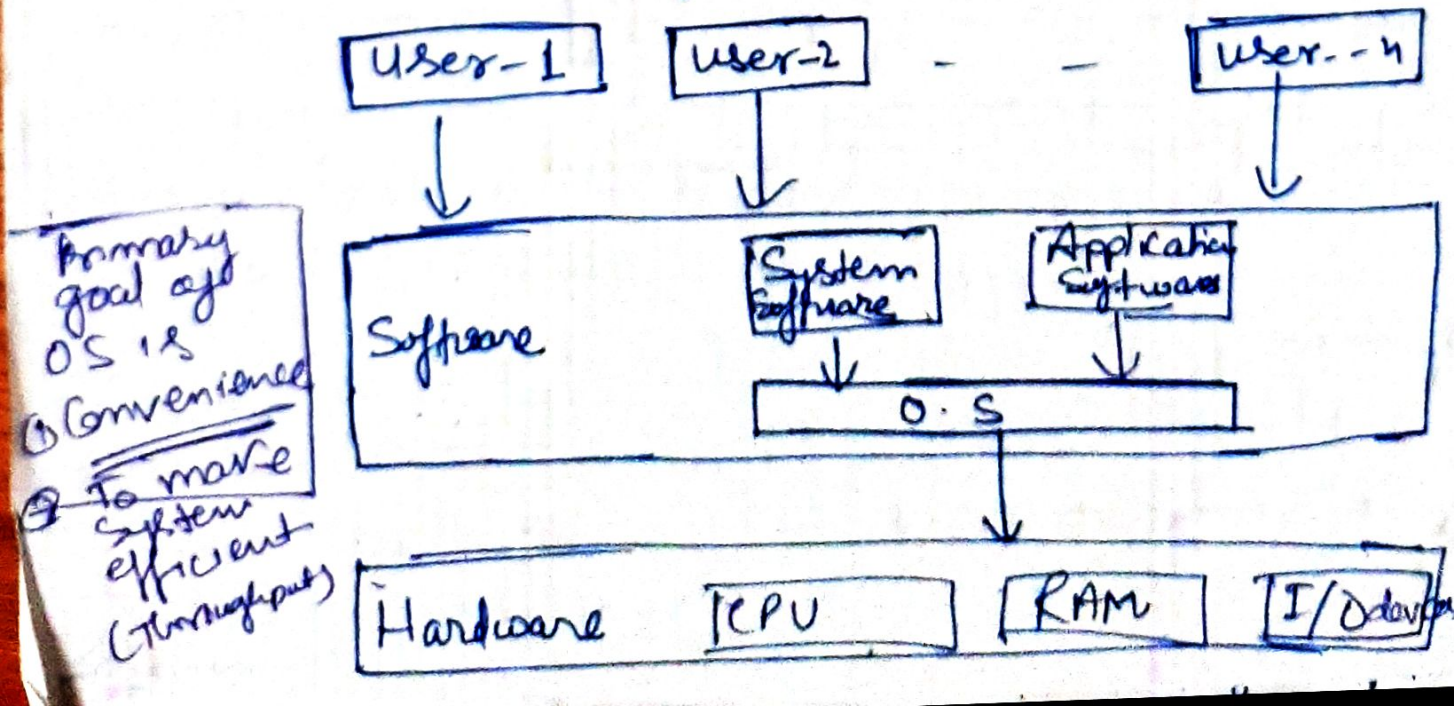
## Unit – I

→ An Operating System (OS) is an interface between a computer user and computer hardware.

→ An O.S is a software which performs all the basic tasks like file management, memory management, handling input and output and controlling peripheral devices such as disk drives and printers.

Ex – Linux O.S, Windows (Convenience), VMS, (Throughput) O/400, AIX, Z/OS etc.

Definition → "An O.S is a program that acts as an interface between the user and computer hardware and controls the execution of all kinds of programs.



following are some of important functions of an O.S

→ Resource Manager (In case Multiple users accessing the same Machine)
→ Memory Management (RAM)
→ Storage management (HDD) - File System
→ Processor Management
→ Device management
→ File management
→ Security
→ Control over System Performance
→ Job accounting
→ Error detecting aids
→ Coordination between other software and users.

**Memory Management** → Memory Management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.

Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must in the main memory. An O.S does the following activity

→ Keeps tracks of primary memory, i.e what part of it are in use by whom, what part are not in use.

→ In multiprogramming, the OS decides which process will get memory when and how much.

→ Allocates the memory when a process requests it to do so

→ De-allocates the memory when a process no longer needs it or has been terminated.

Processor Management → In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called (CPU) Process scheduling. An O.S does the following activities for processor management

→ Keeps tracks of processor and status of process. The program responsible for this task is known as traffic controller

→ Allocates the processor (CPU) to a process.

→ De-allocates processor when a process is no longer required

Device Management → An O.S manages device communication via their respective drivers. It does the following activities for device management:

→ Keeps tracks of all devices. The program responsible for this task is known as I/O Controller.

→ Decides which Process gets the device when and for how much time.

→ Allocates the device in the most efficient way.

→ De-allocates devices

File Management → A file system is normally organized into directories for easy navigation. These directories may contain files and other directions An O.S does the following activities for file management:

— Keeps track of information, location, uses, status etc. The collective facilities are often known as file system.

- Decides who gets the resources.
- Allocates the resources
- De-allocates the resources.

## Other Important Activities →

Following are some of the important activities that an O.S performs:

- Security - By means of Password and Similar other techniques, it prevents unauthorized access to Program and data.
- Control over System Performance → Recording delays between request for a service and response from the system.
- Job accounting - Keeping track of time and resources used by various Jobs and users.
- Error detecting aids → Production of dumps, traces, error messages, and other debugging and detecting aids.
- Coordination between other Software and users → Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of Computer System.

# Operating System - Types

**Batch Operating System** → The users of a batch O.S do not interact with computer directly.

→ Each user prepares his job on an off-line device like Punch Cards and submits it to the computer operator.

→ To speed up processing, jobs with similar needs are batched together and run as a group.

→ The programmer leave their programs with the operator and operator then sorts the programs with similar requirements into batches

**The Problem with Batch System are as follows.**

→ Lack of interaction between the user and the job.

→ CPU is often idle, because the speed of mechanical I/O devices is slower than CPU.

→ Difficult to Provide the desired priorities.

Time – Sharing O.S or Multitasking (Preemptive) → Time – Sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time.

→ Time – Sharing or multitasking is a logical extension of multiprogramming.

→ Processor's time which is shared among multiple users simultaneously is termed as time – sharing

→ The main difference between multiprogrammed Batch Systems and Time – Sharing Systems is that in case of multiprogrammed batch system, the objective is to maximize Processor use (ie minimize idleness), whereas in Time – Sharing systems, the objective is to minimize response time.

→ Multiple jobs are executed by the CPU by switching between them, but the switches occur so frequently. Thus, the user can receive an immediate response.

For example :- In a transaction Processing, the Processor executes each user Program in a short burst or quantum of computation. That is, if n users are Present, then each user can get a time quantum. When the user submits the command, the response time is few seconds at most

→ The O.S uses CPU scheduling and multiprogramming to provide each user with a small Portion of a time.

→ Computer systems that were designed Primarily as batch systems have been modified to time-sharing systems.

## Advantages of Time-Sharing O.S →

→ Provides the advantage of quick response.

~~Avoids duplicate software~~

→ Reduce CPU idle time.

## Disadvantages of Time-Sharing O.S

- Problem of reliability
- Question of Security and integrity of user Programs and data

- Problem of data Communication

Distributed O.S → Distributed Systems use multiple central Processors to serve multiple real-time applications and multiple users.

→ Data Process jobs are distributed among the Processors accordingly.

→ The Processors Communicate with one another through various Communication lines (such as high-speed buses or telephone lines). These are referred as loosely Coupled systems or distributed systems.

→ Processors in a distributed system may vary in size and functions. These Processors are referred as sites, nodes, computers, and so on

The advantages of distributed Systems are as follows —

→ With resource sharing facility, a user at one site may be able to use the resources available at another.

→ Speedup the exchange of data with one another via electronic mail.

→ If one site fails in a D.S, the remaining sites can Potentially Continue operating.

→ Better service to the Customers

→ Reduction of the load on the host Computer.

→ Reduction of delays in data Processing

Network Operating System → A N/w OS runs on a server and Provides the Server the capability to manage data, users, groups, security, applications, and other networking functions

→ The Primary Purpose of the N/w OS is to allow shared file and Printer access among multiple Computers in a network typically a LAN and, a Private N/w or to other N/w.

Examples → Microsoft Windows Server 2003, Microsoft windows Server 2008, UNIX, LINUX, Mac OS, Novell NetWare, and BSD

the advantage of H/w O.S are as follows:

→ Centralized Servers are highly stable
→ Security is server managed
→ Upgrades to new technologies and hardware Can be easily integrated into the System.
→ Remote access to servers is Possible from different locations and types of systems.

The disadvantages of H/w O.S are as follows:→

→ High Cost of buying and running a Server.
→ Dependency on a central location for most operations.
→ Regular maintenance and updates are required.

---

## Real-Time Operating System→

A real-time system is defined as a dataprocessing system in which the time interval required to process and respond to inputs is so small that it controls the environment

→ The time taken by the system to respond to an input and display of required updated information is termed as the response time.

→ The response time is very less as compared to online Processing

→ Real-time systems are used when there are rigid time requirements on the operation of a Processor or the flow of data and real-~~tag~~ time systems can be used to control device in a dedicated application.

→ A real-time O.S must have well-defined, fixed time constraints, otherwise the system will fail.

Ex- Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc. (Missile System, flight Simulator)

There are two types of real-time O.S

Hard Real-time Systems :-

Hard real-time systems guarantee that critical task complete on time. In hard

real-time systems, secondary storage is limited or missing and the data is stored in ROM.

→ In these systems, virtual memory is almost never found. (Missile System, Flight Simulator)

Soft Real-Time Systems — Soft real-time systems are restrictive.

→ A critical real-time task gets Priority over other tasks and retains the Priority untill it completes.

→ Soft real-time Systems have limited utility than hard real-time systems.

Ex— multimedia, virtually reality, Advanced Scientific Projects like under-sea exploration and planetary rovers etc. (youtub Streaming)

—X— —X—

# Operating System Services

An O.S provides services to both the user and to the programs

→ It provides programs an environment to execute

→ It provides users the services to execute the programs in a convenient manner

following are few common services provided by an O.S

→ Program execution
→ I/O operations
→ file system manipulation
→ Communication
→ Error Detection
→ Resource allocation
→ Protection

**Program Execution** → O.S handle may kind of activities from user programs to system programs like Printer spooler, name servers, file servers etc. each of these activities is encapsulated as a process.

→ A process includes the complete execution context (code to execute, data to manipulate, registers

9 OS resources in use). Following are the major activities of an O.S with respect to Program management:

→ Loads a Program into memory
→ Executes the Program
→ Handles Program execution.
→ Provides a mechanism for Process Synchronization.
→ Provides a mechanism for Process Communication.
→ Provides a mechanism for deadlock handling.

I/O operations → an I/O subsystem comprises of I/O devices and their corresponding driver software.

→ Drivers hide the ~~Peculiarities~~ Peculiarities of specific hardware devices from the users.

An O.S manages the Communication between user and device drivers

→ I/O operation means read or write operation with file or any specific I/O device.

→ OS provides the access to the requir I/O devices when required.

File System Manipulation → A file represents a collection of related information.

→ Computers can store file on the disk (Secondary Storage), for long-term storage Purpose.

→ Example of Storage media include magnetic tape, magnetic disk and optical disk drives like CD, DVD. Each of these media has its own Properties like speed, capacity, data transfer rate and dat access methods.

→ A file system is normally organized into directories for easy navigation and usage. These directories may Contain files and other directions.

Following are major activities of an O.S with respect to file management

→ Program needs to read a file or write a file

→ The O.S gives the Permission to the Program for operations on file

→ Permission varies from read-only, read-write, denied, and so on.

→ O.S Provides an interface to the user to create/delete files.

→ O.S Provides an interface to the user create/delete directories

→ O.S Provides an interface to create backup of file system.

Communication → In case of distributed systems which are collection of Processors that do not share memory, Peripheral devices, or a clock, the O.S manages Communications between all the Processes.

→ Multiple Processes Communicate with one another through communication lines in the network.

→ The O.S handles routing and Connection strategies, and the Problems of Contention and Security.

Following are the major activities of an O.S with respect to Communication:

→ Two Processes often require data to be transferred between them.

→ Both the Processes can be on one computer or on different computers, but are connected through a computer network.

→ Communication may be implemented by two methods, either by shared memory or by message Passing

Error Handling → Errors can occur anytime and anywhere.

→ An error may occur in CPU, in I/O devices or in the memory hardware.

Following are major activities of an O.S with respect to error handling:

→ The O.S constantly checks for possible errors

→ The O.S takes an appropriate action to ensure correct and consistent computing.

Resource Management → In case of multi-user or multi-tasking environment, resources such as main memory, CPU cycles and file storage are to be allocated to each user or job.

Following are the major activities of an O.S w.r.t resource management

→ The O.S manages all kind of resources using schedulers.

→ CPU scheduling algorithms are used for better utilization of CPU.

Protection : Considering a computer system having multiple users and concurrent execution of multiple processes, the various processes must be protected from each other's activities.

→ Protection refers to a mechanism or a way to control the access of programs, processes, or users to the resources defined by computer system.

Following are the major activities of an O.S with respect to protection.

→ The O.S ensures that all access to system resources is controlled.

→ The O.S ensures that external I/O devices are protected from invalid access attempts

→ the O.S provides authentication feature for each user by means of passwords.

---

## Operating System- Properties

**Batch Processing** → Batch Processing is a technique in which an O.S collects the programs and data together in a batch before processing starts.

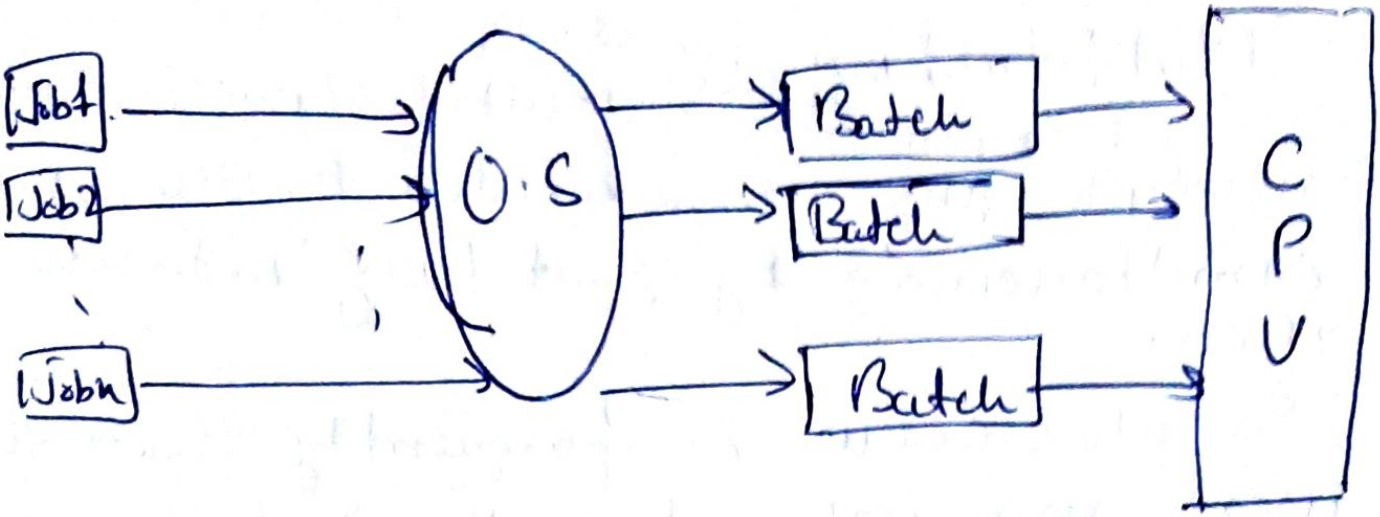
An O.S does the following activities related to batch processing:

→ The O.S defines a job ~~which~~ has predefined sequence of commands, programs and data as a single unit

→ The O.S keeps a number of jobs in memory and execute them without any manual information

→ Jobs are processed in the order of submission, i.e FCFS fashion.

→ when a job completes its execution, its memory is released and the output for job get copied into an output for later printing or processing




Advantages :

→ Batch Processing takes much of the work of the operator to the Computer

→ Increased Performance as a new job get started as soon as the previous Job is finished, without any manual intervention

Disadvantages :- Difficult to debug Programs

→ A job could enter an infinite loop

→ Due to lack of Protection scheme, one batch job can affect other Pending Jobs

## Multitasking / Timesharing (Preemptive)

→ Multitasking is when multiple jobs are executed by the CPU simultaneously by switching between them.

→ Switches occur so frequently that the users may interact with each program while it is running.

An O.S does the following activities related to multitasking:

→ The user gives instructions to the O.S or to a program directly, and receives immediate response.

→ the OS handles multitasking in the way that it can handle multiple operations/ executes multiple programs at a time.

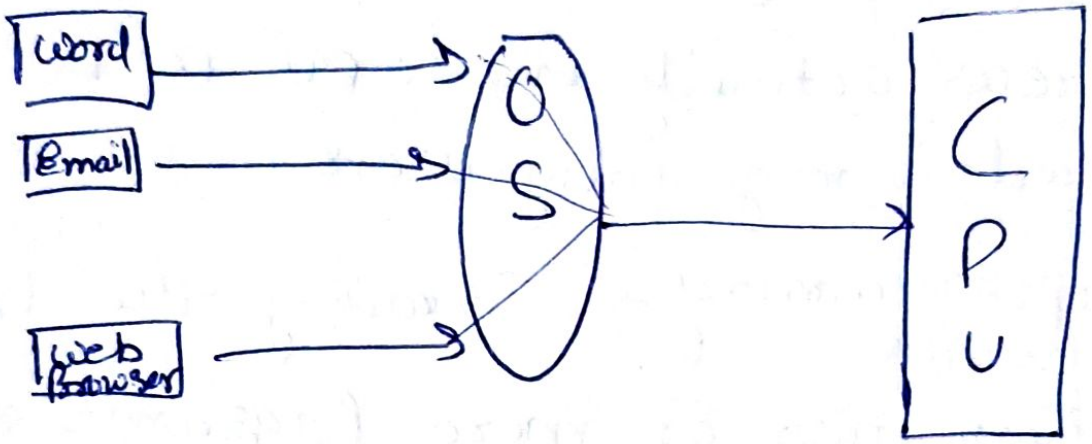
→ Multitasking O.S are known as time-sharing systems.

→ These O.S were developed to provide interactive use of a computer system at a reasonable cost

→ A time-shared O.S uses the concept of CPU scheduling and multiprogramming to provide each user with small portion of time

## Shared CPU

→ Each user has at least one separate program in memory.




⇒ Multitasking gives emphasis on Responsiveness

→ A program that is loaded into memory and is executing is commonly referred to as a Process

→ When a process executes, it typically executes for only a very short time before it either finishes or needs to perform I/O

→ Since interactive I/O typically runs at slower speeds, it may take a long time to complete. During this time, a CPU can be utilized by another process.

→ The O.S allows the users to share the computer simultaneously. Since each action or command in a time shared system tends to be short, only a little CPU time is needed for each user.

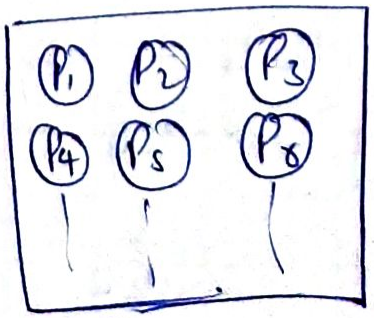
→ As the system switches CPU rapidly from one user/program to the next, each user is given the impression that he/she has his/her own CPU, whereas actually one CPU is being shared among many users.

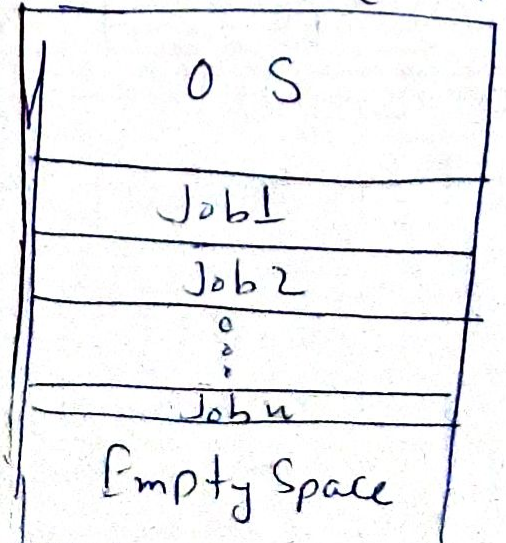
Multiprogramming (Non Preemptive) : Sharing the Processor, when two or more programs reside in memory at the same time, is referred as multiprogramming.

⇒ Multiprogramming assumes a single shared Processor.

⇒ Multiprogramming ~~assumes a single~~ increases CPU utilization by organizing jobs so that the CPU always has one to execute. (Non Preemptive)

⇒ Its aim is to engage the Process (not to remain idle)




→ Memory layout of Multiprogramming System

An O.S does the following activities related to multiprogramming.

→ The O.S keeps several jobs in memory at a time

→ This set of jobs is a subset of jobs kept in the job Pool

→ The O.S Picks and begins to execute one of the jobs in the memory

→ Multiprogramming O.S monitor the state of all active programs and system resources using memory management program to ensure that the CPU is never idle, unless there are no jobs to Process

Advantage :- High and efficient CPU utilization.

— User feels that many programs are allotted CPU almost simultaneously

Disadvantage :- → CPU Scheduling is required

→ To accommodate many jobs in memory, memory management is required.

Interactivity → Interactivity refers to the ability of users to interact with a computer system.

An O.S does the following activities related to interactivity:

→ Provides the user an interface to interact with the system.

→ Manages input devices to take inputs from the user. for ex- Keyboard

→ Manages o/p devices to show outputs to user. For ex- monitor

The Response time of the OS needs to be short, since the user submits and waits for result.

Real-Time Systems → Real-time systems are usually dedicated, embedded systems.

An O.S does the following activities related to real-time system activity.

→ In such systems, O.S typically read from and react to sensor data.

→ The O.S must guarantee response to events within fixed Periods of time to ensure correct Performance

**Distributed Environment** → A distributed environment refers to multiple independent CPUs or Processors in a computer system. An O.S does the following activities related to distributed environment

→ The OS distributes computation logics among several physical Processors.

→ The Processors do not share memory or clock. Instead, each Processor has its own local memory

→ The O.S manages the communications between the Processors. They communicate with each other through various communi-cation lines

**Spooling** → Spooling is an acronym for Simultaneous Peripheral operations on line.

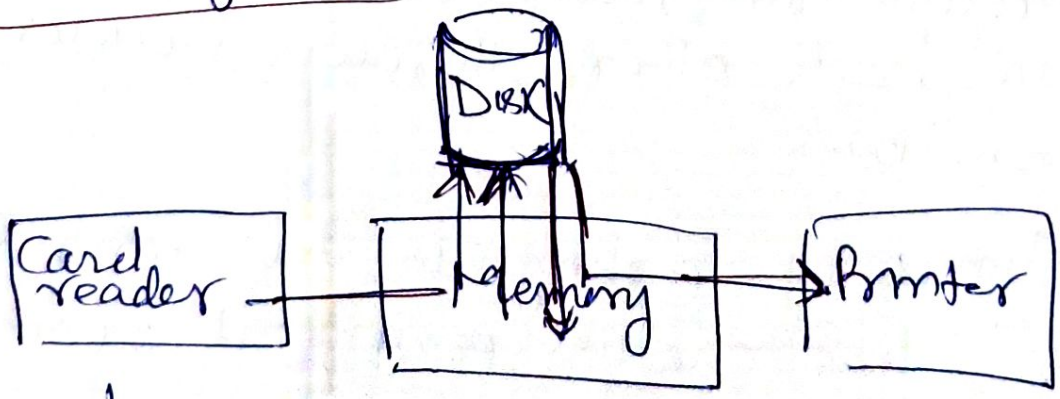
→ Spooling refers to Putting data of various I/O Jobs in a buffer. This buffer is a special area in memory or hard disk which is accessible to I/O devices

An O.S does the following activities related to Spooling –

→ Handles I/O device data spooling as devices have different data access rates.

→ Maintains the Spooling buffer which provides a waiting station where data can rest while the slower device catches up

→ Maintains Parallel Computation because of Spooling process as a computer can perform I/O in parallel fashion it becomes possible to have the computer read data from tape, write data to disk and to write out to a tape printer while it is doing its computing task.




Advantages →

→ The Spooling operation uses a disk as a very large buffer

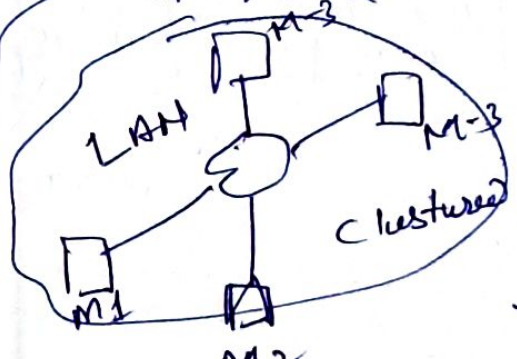
→ Spooling is capable of overlapping I/O operation for one job with Processor operation for another job.

# Types of OS :-

1. Batch
2. Multi programmed
3. Multitasking / Time sharing
4. Real time O.S
5. Distributed O.S
6. Clustered
7. Embedded.

Embedded — Microwave, AC, Washing Machine (Fixed functionality) the embedd functionality)

Clustered — Multiple machines connected together using LAN but acts like a single computer by sharing their all resources and hence make a cluster




Advantage -
- Availability
- Fault tolerance
- Scalability

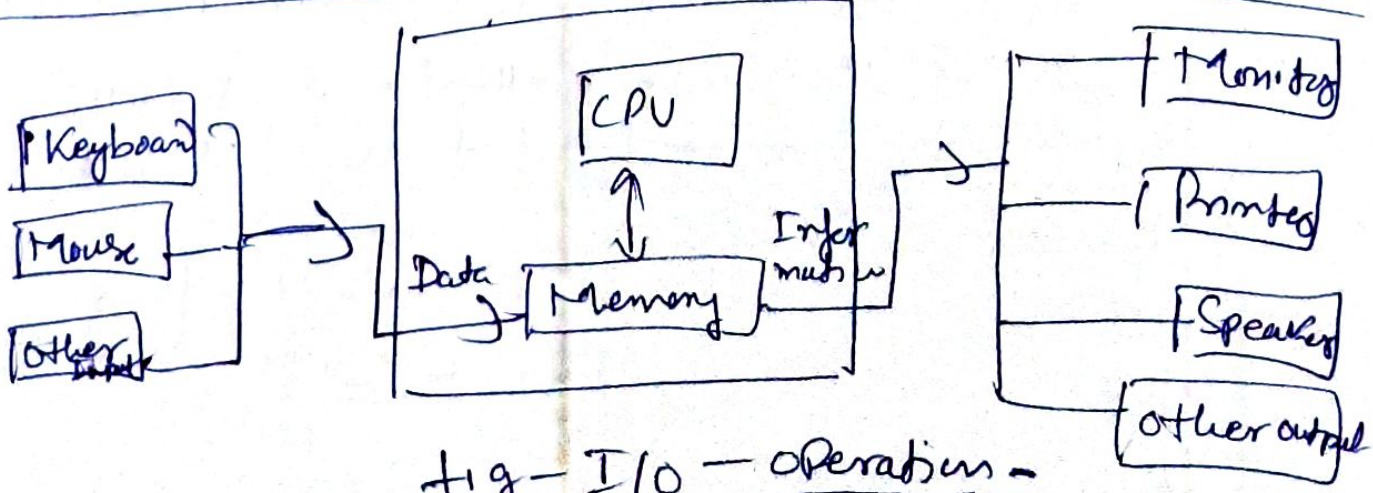
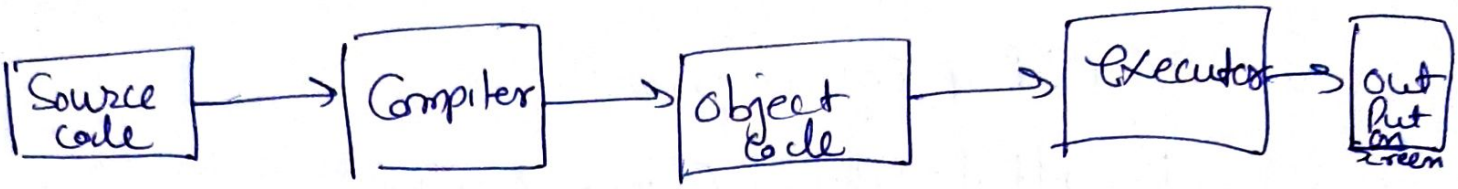



fig - I/O - operation -

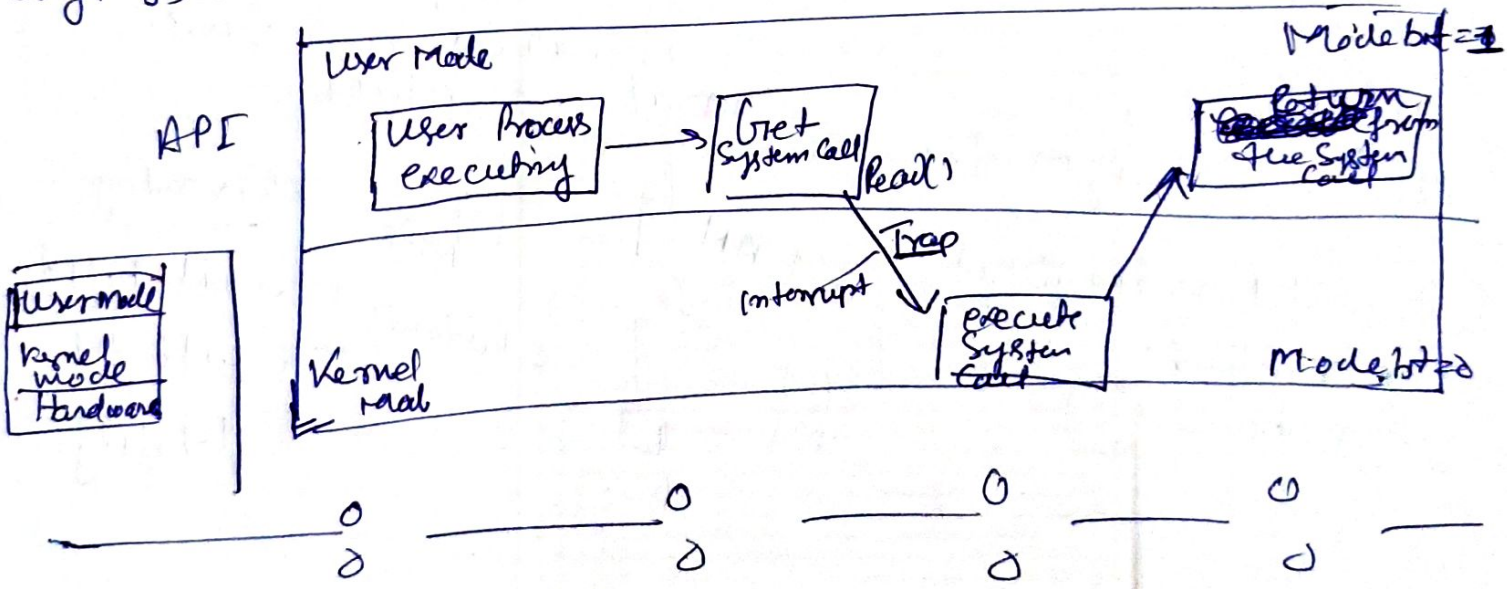
# fig- Program Execution




## User Mode Vs Kernel Mode

**Kernel Mode**
- Where OS runs
- Privileged (can do anything)

**User mode**
- Where process run
- Restricted access to resource
- Restricted capabilities




**System Call** ⇒ A System Call is the Programmatic way in which a computer Program requests a service from the Kernel of the O.S it is executed on.

⇒ A system call is a way for Programs to interact with O.S.

→ System Call Provides the services of the O.S to the user Program via Application Program Interface (API)

## Services Provided by System Call →

1 — Process creation and Management
2 — Main Memory Management
3 — File access, directory and File system management
4 — Device handling
5 — Protection
6 — Networking etc.

| Example → | | Windows | Unix |
|---|---|---|---|
| ① | Process Control System Call | CreateProcess()<br>ExitProcess()<br>WaitforSingleObject() | fork()<br>exit()<br>wait() |
| ② | File Manipulation System Call | CreateFile()<br>ReadFile()<br>WriteFile()<br>CloseHandle() | Open()<br>read()<br>write()<br>close() |
| ③ | Device Manipulation System Call | SetConsoleMode()<br>ReadConsole()<br>WriteConsole() | ioctl()<br>read()<br>write() |
| ④ | Information Maintenance System Call | GetCurrentProcessID() | getPid() |
| | | SetTimer()<br>Sleep() | Alarm()<br>Sleep() |
| ⑤ | Communication System Call | CreatePipe()<br>CreateFileMapping()<br>MapViewOfFile() | Pipe()<br>shmget()<br>mmap() |
| ⑥ | Protection System Call | SetFileSecurity()<br>InitializeSecurityDescriptor()<br>SetSecurityDescriptorGroup()<br>chown() | chmod()<br>umask() |

(Multiprocess or Parallel System) :-

Multiprocessing O.S :- An O.S, to improve the Performance of more than one CPU can be used within one computer system called Multiprocessor O.S

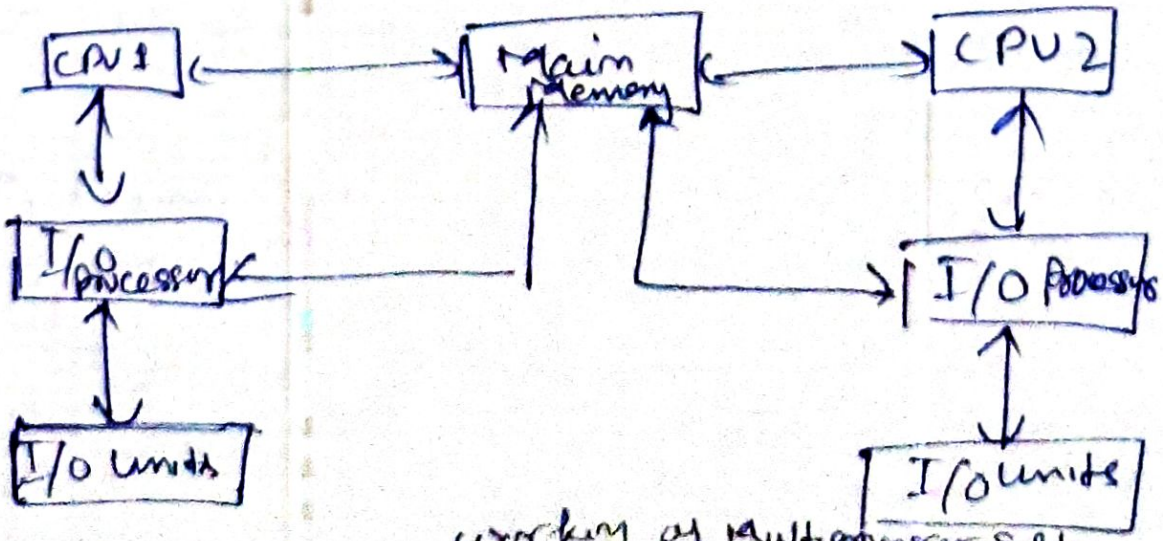
⇒ Multiple CPU are interconnected so that a job can be divided among them for faster execution. When a job finishes, results from all CPUs are collected and compiled to give the final output.

⇒ Job needed to share main memory and they may also share other O.S resources among themselves.

⇒ Multiple CPUs can also be used to run multiple jobs simultaneously.

Ex- Unix

O.S is one of the most widely used multiprocessing System.




working of Multiprocessor System

To employ a multiprocessing O.S effectively the Computer System must have the following things:

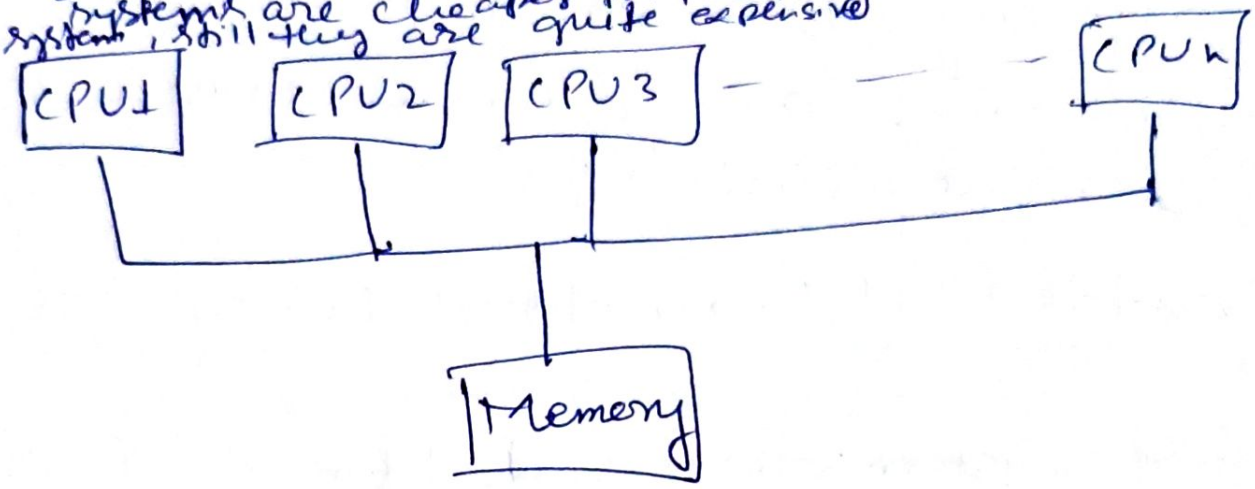
- A motherboard is capable of handling multiprocessor in a multiprocessing O.S
- Processor are also capable of being used in multiprocessing system.

Advantages of Multiprocessing O.S are

- Increased Reliability → Due to multiprocessing System, Processing task can be distributed among several Processors thus increases reliability as if one Processor fails; the task can be given to another processor for completion.
- Increased Throughput → As several Processor increase, more work can be done in less time
- The Economy View → As multiprocess - or System share Peripherals, Secondary Storage devices, and Power Supplies, they are relatively cheaper than single-Processor System.

## Disadvantages of Multiprocessing O.S

=> Large Memory Required.
=> O.S of multiprocessing is more complex and Soplusticated as it takes care of multiple CPUs at the same time
=> Increased Expenses — Even though Multiprocessor systems are cheaper in the long run than multicomputer systems, still they are quite expensive




## Types of Multiprocessing Systems:

### Symmetrical Multiprocessing O.S :-

In a Symmetrical Multiprocessing O.S, each processor executes the same copy of O.S and takes its own decision, and cooperate with other processes to smooth the entire functioning of the system.
=> The CPU scheduling Policies are very simple.
=> Any new job submitted by a user can be assigned to any processor that is least burdened
=> It also results in a system in which all processor are equally burdened at any time
=> The Symmetric Multiprocessing O.S is also known as "Shared every-thing" system, because

the Processor share memory and Input/output bus or data Path.

## Characteristic of Symmetrical Multiprocessing OS →

- In this system, any Processor can run any job or Process.
- In this, any Processor initiates an input/output operation

## Advantages of Symmetrical Multiprocessing O.S →

These systems are fault tolerant. Failure of a few Processor does not bring the entire system to a halt

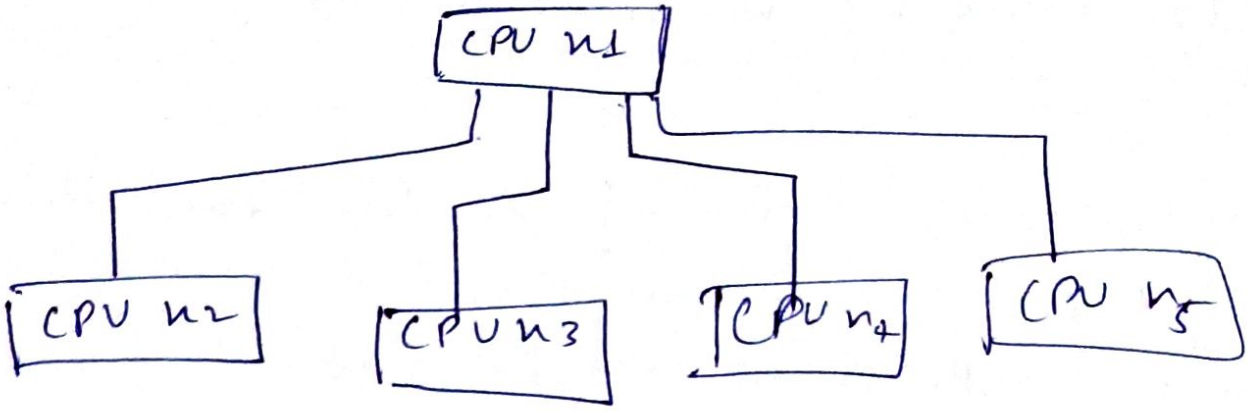
## Disadvantages of Symmetrical O.S →

- It is very difficult to balance the workload among Processor rationally.
- Specialized synchronization schemes are necessary for managing multiple Processor

## Asymmetric Multiprocessing O.S → In

Asymmetric multiprocessing system, there is a master slave relationship between the Processor.

⇒ further one Processor may act as a master Processor or supervisor Processor




⇒ In above fig, the asymmetric processing system shows that CPU n1 acts as a supervisor whose function control following processors.

⇒ In this type of system, each processor is assigned a specific task, and there is a designated master processor that controls the activities of other processors.

For example : We have a math co-processor that can handle mathematical job better than the main CPU

→ Similarly, we have an MMX processor that is built to handle multimedia-related jobs

→ Graphics processor etc.

⇒ When a user submits a new job, the O.S has to decide which processor can perform it better, and then that processor is assigned that newly arrived job. This processor acts as the master and controls the system. All other processor look for masters for instructions or have predefined tasks. It is the responsibility of master to allocate work to other processor

## Advantages of Asymmetric Multiprocessing O.S →

⇒ In this type of system execution of Input and output Operation or an application Program may be faster in some situations because of many Processors may be available for a single job

## Disadvantages of Asymmetric multiprocessing OS →

⇒ In this type of multiprocessing O.S, the Processors are unequally burdened. one Processor may be having a long job queue, while another one may be sitting idle.

⇒ In this system, if the Process handling a specific work fails, the entire system will go down.

The Best O.S in multiprocessor Parallel Computing environment is UNIX, because it has many advantages such as,

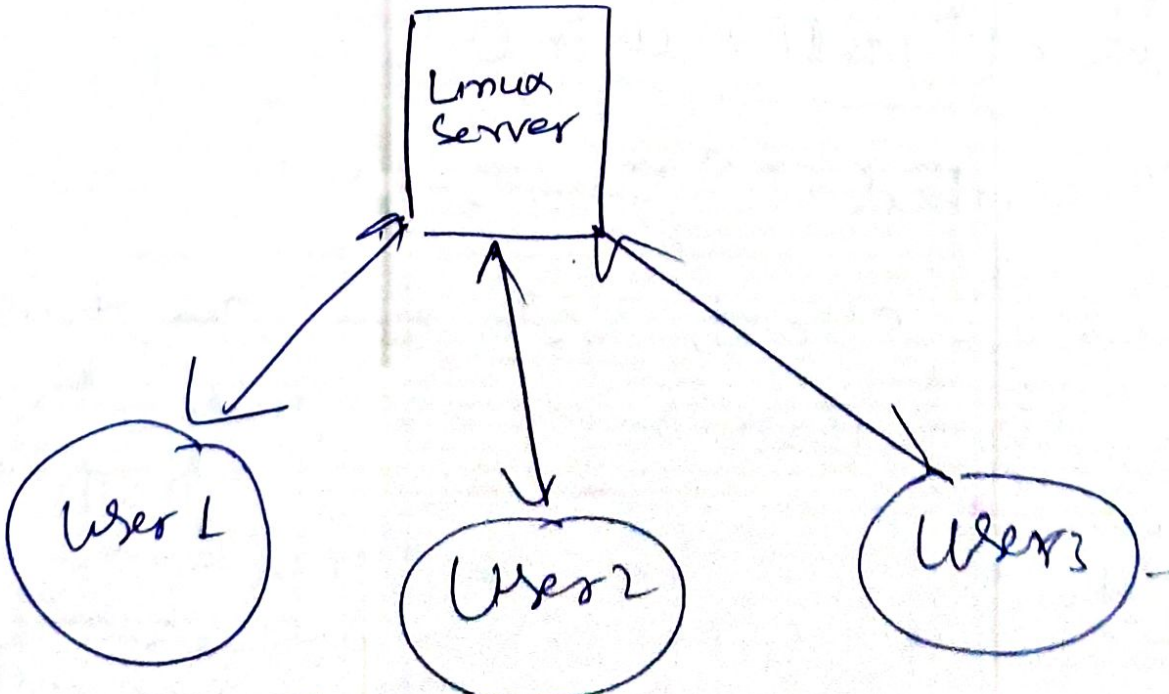
- It is a multiuser
- It is Portable
- It is good for multitasking
- It has organized file system
- It has device independence.
- Utilities are handy and operation commands can be combined in a single line.
- Unix provides various services, as it has built in administrative tools
- Unix can share files over electronic network with many various kinds of equipment

**Multiuser O.S** → "A multi-user O.S is a computer O.S which allows multiple users to access the single system with one O.S on it.

⇒ It is generally used on large mainframe computer.

ex- Linux, Unix, Windows 200, Ubuntu, Mac O.S etc.

⇒ In the multi-user O.S, different users connected at different terminals and we can access, these users through Network




**Features of the Multi-user OS →**

- **Multi-tasking** — Using multi-user O.S we can perform multiple tasks at a time. i.e we can run more than one program at a time.

- Resource Sharing - we can share different peripherals like printer, hard drives or we can share a file or data. For this, each user is given a small time slice of CPU time.

- Background Processing - It means that when commands are not processed first, then they are executed in the background while another programs are interacting with the system in the real time

- Time Sharing

---

Types of Multi-User O.S -

① Distributed Systems -

② Time-Sliced Systems - In this, a short period is assigned to each task, i.e each user is given a time slice of the CPU time. As we know these time slices are tiny, so it appears to the users that they all are using mainframe computer at the same time.

## Multiprocessor System →

Ex — Linux, Windows XP, Unix, Windows 2000, Windows 2003

## Advantages of Multiuser-O.S →

- A multi-user O.S can be used in Printing Process to allow multiple users to access the same Printer, which a normal O.S may not do.
- On a single computer system, several users can access the same copy of documents. For instance, if a PPT file is kept on one computer, other user can see it on other system
- If one computer fails in its own network system, the entire system does not come to halt.
- Airlines / Ticket reservation system uses multi user O.S

## Disadvantages of Multi-User O.S →

1- Virus attacks occur simultaneously on all of them as the computer are shared. As a result, if one machine is affected, the other will be as well.

2- All Computer information is Shared Publicly and your Personal information is accessible to everyone on the network

3- Multiple accounts on a single Computer may not be suitable for all users. Thus, better to have multiple PCs for each user.

# Unit - I

**Operating System Structure** → An O.S has a complex structures, so we need a well-defined structures to assist us in applying it to our unique requirement.

→ Just break down a big problem into smaller, easier to solve subproblems, designing an O.S in parts is a simpler approach to do it.

→ Each section is an O.S Component.

→ The approach of interconnecting and integrating multiple O.S Components into the Kernel can be described as an O.S structure.

---

* **Kernel** – is central component of an O.S that manages operations of computer and hardware.

→ It basically manages operations of memory and CPU time.

→ It is core component of an O.S

→ Kernel acts as a bridge between applications and data processing performed at hardware level using inter-Process Communication and System Calls.

→ Kernel loads first into memory when an O.S is loaded and remains into memory until O.S is shut down again.

→ It is responsible for various tasks such as disk management, task management, and memory management.

→ It decides which process should be allocated to processor to execute and which process should be kept in main memory to execute. Process scheduling

→ It basically acts as an interface between user applications and hardware.

→ The major aim of Kernel is to manage communication between software i.e user-level applications and hardware i.e CPU and disk memory

## Objective of Kernel:

→ To establish communication between user level application and hardware

→ To decide state of incoming processes

→ To control disk management

→ To control memory management

→ To control task management

## Types of Kernel:

① Monolithic Kernel: It is one of types of Kernel where all O.S services operate in kernel space. It has dependencies

between system components.

→ It has huge lines of code which is complete.

Ex— Unix, Linux, OpenVMS, XTS-400 etc.

Advantage→ It has good Performance

Disadvantage→ It has dependencies between system components and Lines of code in millions.

2- Micro Kernel→ It is Kernel types which has minimalist approach. It has virtual memory and thread scheduling. It is more stable with less services in Kernel space.

→ It Puts rest in user space.

Ex— QNX, Mach, L4, AmigaOS, Minix, K42 etc.

Advantage— It is more stable

Disadvantage— There are lots of system calls and context switches.

3— Hybrid Kernel— It is the combination of both monolithic Kernel and microkernel.

→ It has speed and design of monolithic Kernel and modularity and stability of microKernel

Ex— Windows NT, Netware, BeOS etc

Advantage - It combines both monolithic Kernel and microKernel.

Disadvantage → It is still similar to monolithic Kernel.

4- Exo Kernel → It is the type of Kernel which follows end to end Principle. It has fewest hardware abstraction as possible. It allocates physical resources to applications

Ex- Nemsis, ExOS etc.

Advantage — It has fewest hardware abstractions

Disadvantage: There is more work for application developers

5- Nano Kernel → It is the type of Kernel that offers hardware abstraction but without system services.

→ Micro Kernel also does not have system services therefore Micro Kernel and Nano Kernel have become analogous.

ex- EROS etc

Advantage — It offers hardware abstractions without system services

Disadvantage — It is quite same as MicroKernel hence it is less used

—x— —x— —x—

Simple Structure → It is the simplest O.S structure and is not well defined

→ It can only be used for small and limited systems.

– In this structure, the interfaces and levels of functionality are not well separated; hence programs can access I/O routines which can cause unauthorized access to I/O routines

→ This structure is implemented in MS-DOS O.S and original Unix

→ The MS-DOS OS is made up of various layers, each with its own set of functions

→ These layers are
- → Application Programs
- → System Program
- → MS-DOS device drivers
- → ROM BIOS device drivers

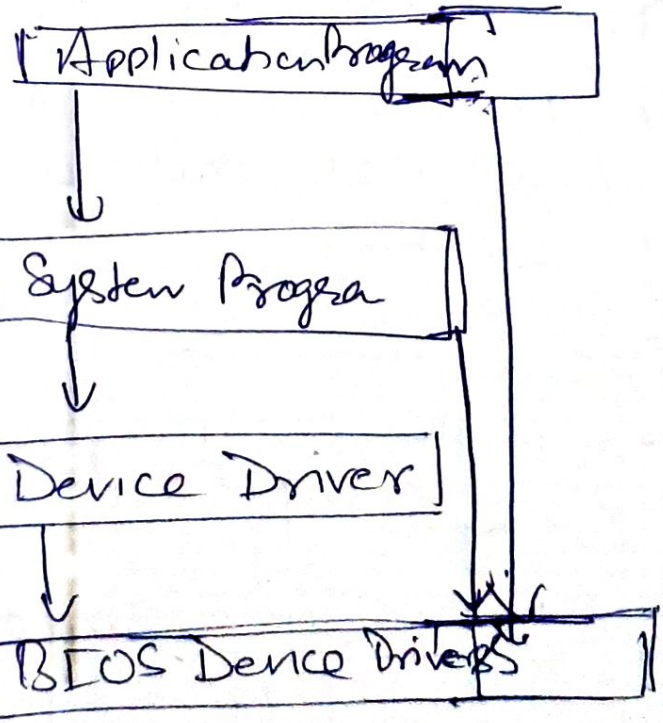
→ Layering has an advantage in the MS-DOS O.S since all the levels can be defined separately and can interact with each other when needed

→ It is easier to design, maintain, and update the system if it is made in layers. So, that's why limited systems with less complexity can be constructed easily using simple structure.

→ If one user program fails, the entire O.S gets crashed.

→ The abstraction level in MS-DOS system is low, so programs and I/O routines are visible to the end-user, so that user can have unauthorized access

Ex- MS-DOS




## Advantages of Simple Structure-

→ It is easy to develop because of the limited number of interfaces and layers

→ offers good Performance due to less layers between hardware and applications

Disadvantages of Simple Structure →

→ If one user Program fails, the entire O.S crashes.

→ Abstraction of data hiding is not Present as layers are connected and communicate with each other

→ Layers can access the Processes going in the O.S, which can lead to data modification and can cause System to Crash

Monolithic Structure → The monolithic O.S in which the Kernel acts as a manager by managing all things like file management, memory management, device management, and operational Processes of the O.S

→ In monolithic systems, Kernels can directly access all the resources of the O.S like Physical hardware (ex- Keyboard, mouse etc)..

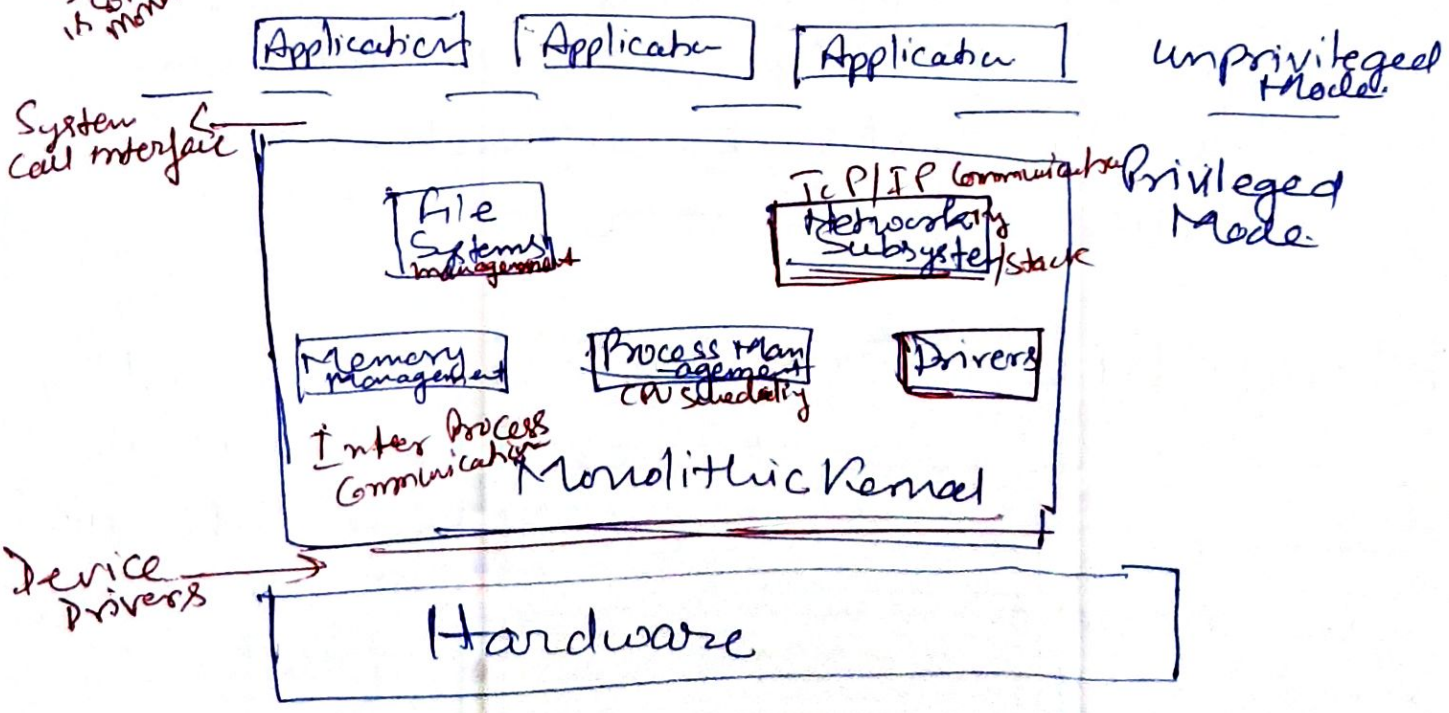
→ the monolithic Kernel is another name for the monolithic O.S.

Concept (System V M)

→ The monolithic Kernel functions as Virtual machine by working on top of the O.S and Controlling all hardware components.

→ This is an outdated O.S that was used in banks to accomplish minor activities such as batch processing and time sharing, which enables many people at various terminals to access the O.S

Ex- Linux is solely monolithic




## 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 monolithic Kernel is relatively fast as compared to normal systems. Using 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.

---

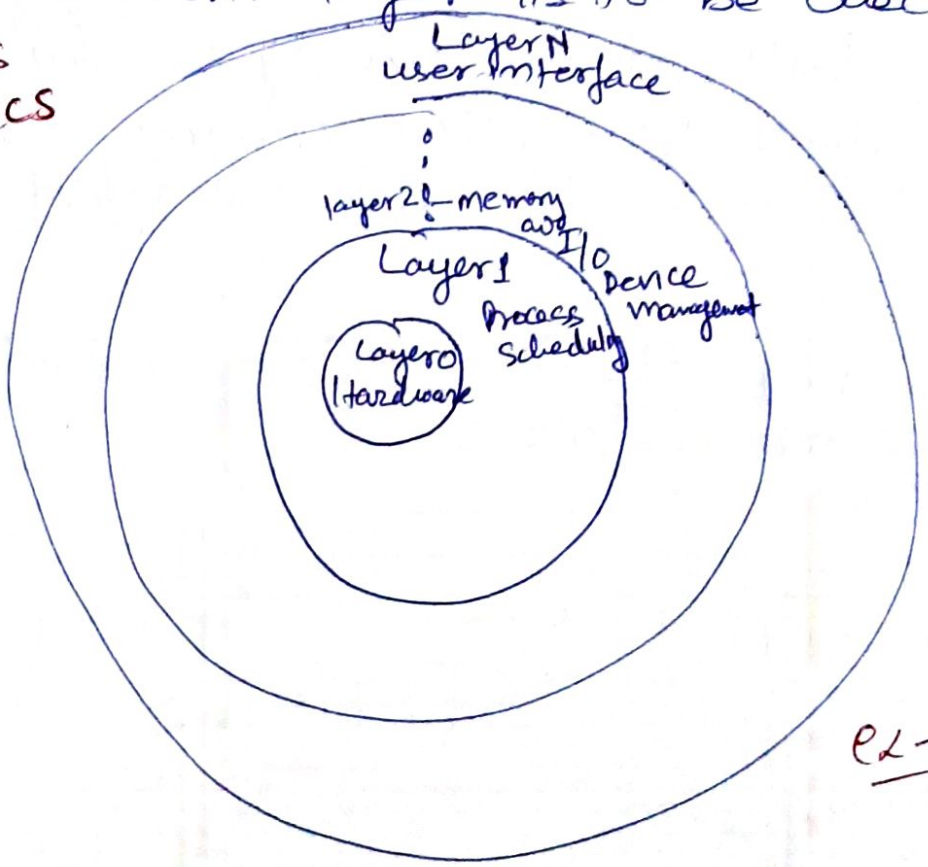
**Layered Approach →** In this type of structure, O.S is divided into layers or levels.

→ The hardware is on the bottom layer (layer 0), while the user interface is on top layer (layer N). These layers are arranged in hierarchical way in which the top layers use the functionalities of their lower-level layer.

→ In this approach, functionalities of each layer are isolated, and abstraction is also available.

→ In layered structure, debugging is easi as it is a hierarchical model, so all lower-level layered is debugged, and then the upper layer is checked. So, all the lower layers are already checked, and the current layer is to be checked only.

example
① THE OS
② MULTICS




Layer5 – User Progra
Layer4 – I/O Manageme
Layer3 – Operator Process Communic
Layer2 – Memory Manage
Layer1 – CPU Scheduli
Layer0 – H/W

Ex – Microsoft Windows NT

## Advantages of Layered Structures

→ 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 Structures

→ 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.

## Micro-Kernel → Micro-Kernel structures

design the O.S by removing all non-essential components of the Kernel.

→ These non-essential components of Kernels are implemented as systems and user Programs. Hence these implemented systems are called as Micro-Kernels

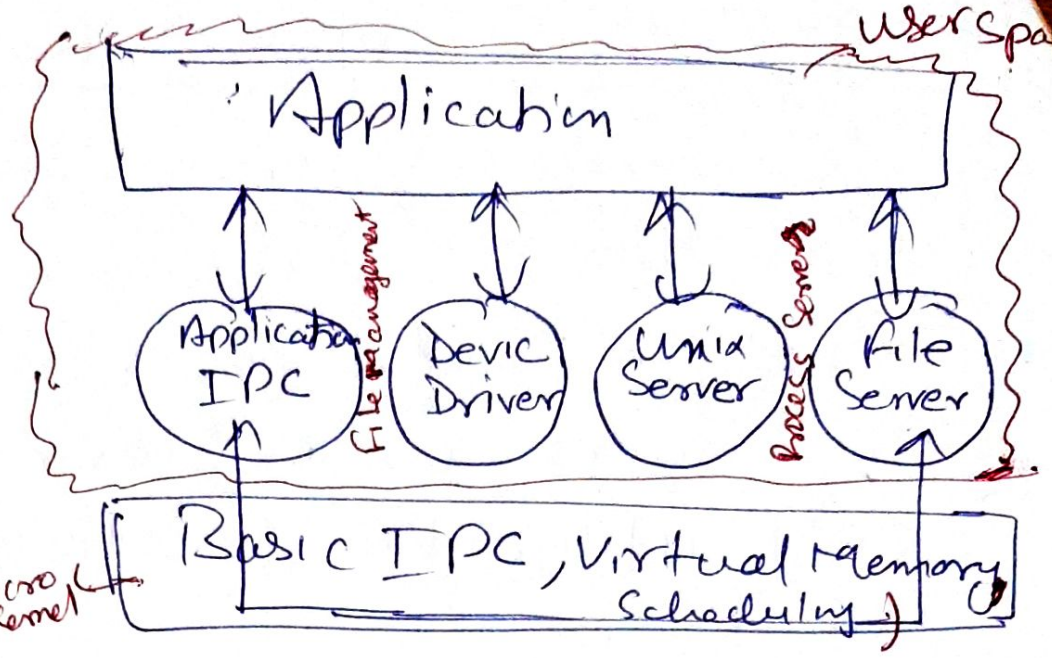
→ Each Micro-Kernel is made independently and is isolated from other Micro-Kernels. So, this makes the system more secure and reliable. If any Micro Kernel fails, then the remaining O.S remains untouched and works fine.

⇒ The interaction between the Client application and services running in user address space is established via message passing.

# MicroKernel O.S

Mac OS combines the feature of microkernel Model as a monolithic Kernel (BSD).

QNX, MINIX OS etc



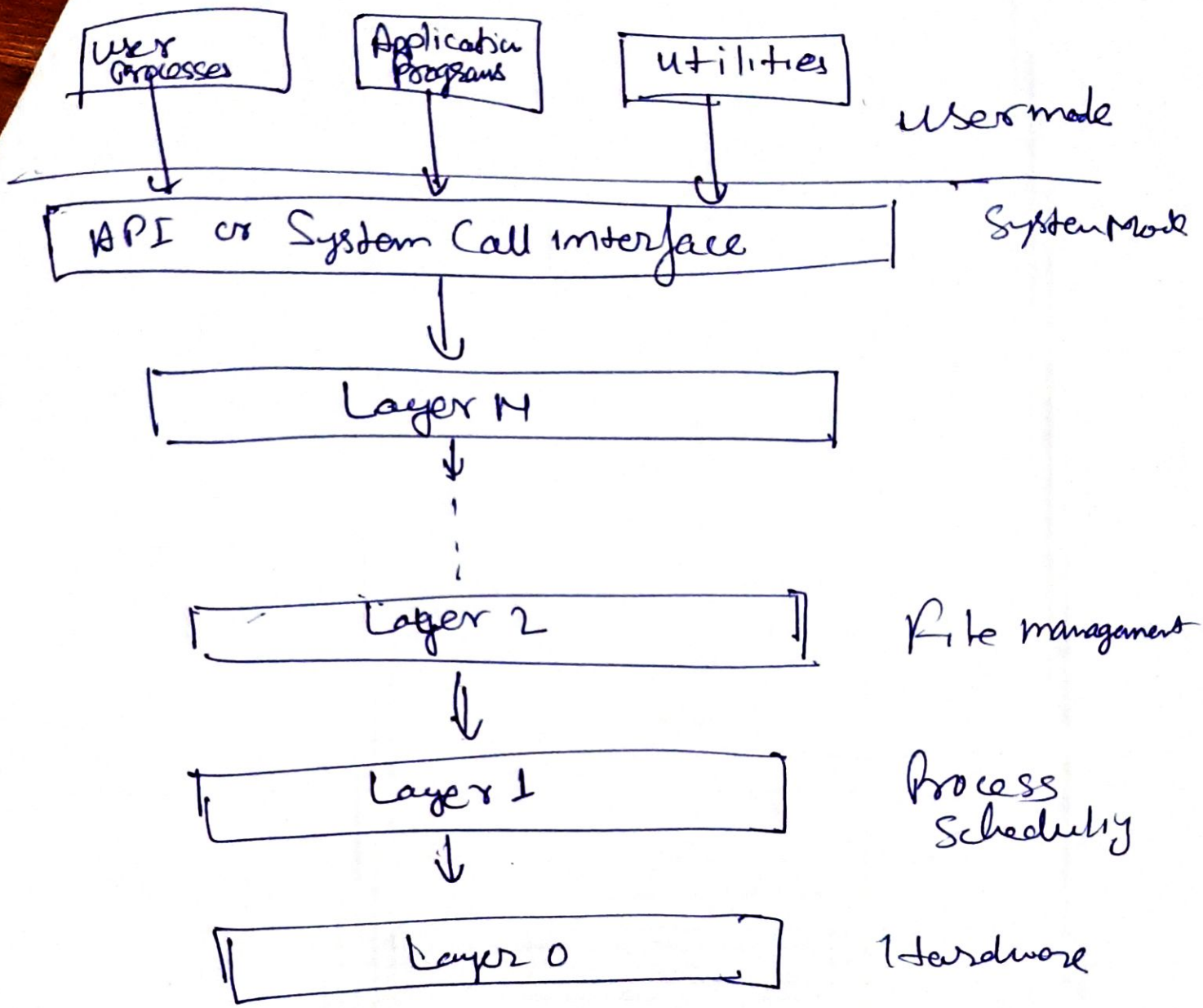

C# → Eclipse IDE is a good example of MicroKernel Architecture

## Advantages of Micro-Kernel Structure →

- → It allows the O.S 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 O.S is unaffected and continues to function normally.

## Disadvantages of Micro-Kernel Structure →

- → Increase inter module Communication reduces System Performance.
- → System is Complex to be Constructed


User Processes
Application Programs
Utilities
User mode
API or System Call interface
System Mode
Layer N
Layer 2
File management
Layer 1
Process Scheduling
Layer 0
Hardware

Layer O.S

# Reentrant Kernel (feature of OS)

⇒ If the Kernel is not re-entrant, a process can only be suspended while it is in user mode. Although it could be suspended in Kernel mode, that would still block Kernel mode execution on all other processes. The reason for this is that all Kernel threads share same memory. If execution would jump between them arbitrarily, corruption might occur.

⇒ A re-entrant Kernel enables processes (or, to be more precise, their corresponding Kernel threads) to give away the CPU while in Kernel mode. They do not hinder other processes from also entering Kernel mode. A typical use case is I/O wait. The process wants to read a file. It calls a Kernel function for this. Inside the Kernel function, the disk controller is asked for the data. Getting the data will take some time and the function is blocked during that time.
With a re-entrant Kernel, the scheduler will assign the CPU to another process (Kernel thread) until an interrupt from the disk

Controller indicates that the data is availa
and our thread can be resumed. Thus
Process can still access I/O (which needs
Kernel functions), like user input. The
System stays responsive and CPU time waste
due to I/O wait reduced.

All unix Kernels are Reentrant: Thus means
that several Processes may be executing
in Kernel mode at the same time. Of course,
on uniprocessor systems, only one Process can
progress, but many can be blocked in Kernel
mode when waiting for the CPU or the
Completion of some I/O operation for instance,
after issuing a read to a disk on behalf
of a Process, the Kernel lets the disk Controller
handle it and resumes executing other Processes.
An interrupt notifies the Kernel when the
device has satisfied the read, so the former
Process can resume the execution.

→ One way to provide reentrancy is to
write functions so that they modify only
local variables and do not alter
global data, structures. Such functions are
called reentrant functions.
But, a reentrant kernel is not limited only
to such reentrant functions (Although that is how
some real-time Kernels are implemented).

Instead, the Kernel can include nonreentrant functions and use locking mechanism to ensure that only one process can execute a nonreentrant function at a time.

→ If a hardware interrupt occurs, a reentrant Kernel is able to suspend the current running process even if that process is in Kernel mode. This capability is very important, because it improves the throughput of the device controll -ers that issue interrupt. Once a device has issued an interrupt, it waits until CPU acknowledges it. If the Kernel is able to answer quickly, the device controller will be able to perform other tasks while the CPU handles interrupt

---

Why Reentrant ? —→

Lets assume that a process is executing in Kernel mode and accessing a Kernel data structure and some global values associated with it.

— Suppose that process name is 'A'.

— Now 'A' accesses a global variable to see if the value is non zero (so that it can do some calculations etc) and just before it tries to use this value in some of its logic, a context switch to process 'B' happens.

- Now this Process 'B' tries to access the value of same global variable and decreaments it.

— Another Context Switch happens and Process 'A' comes back into execution

— Since 'A' does not know that 'B' has already decremented the value, it tries to use this value again.

— So, here is the catch, Process 'A' sees two different values of global variable as the value was changed by another Process 'B'.

So, now we know that why a Kernel needs to be reentrant

---

Following Points could be considered for making a Kernel reentrant:

— Write Kernel functions that modify only the local (stack) variables and do not alter the global variables or data structures. Such type of functions are known as reentrant functions.

— Strictly adhering to the use of only reentrant functions in Kernel is not a feasible solution. So another technique used

is 'locking' mechanism that ensure that only one process can use a non-reentrant function at a given time.

— from the above points it is clear that use of reentrant functions locking mechanism for non-reentrant functions is the core of making a Kernel reentrant. Since implementing reentrant functions is more related to good programming, the locking mechanism are related to concept of synchronization