UNIT-I

Operating Systems

Introduction :

Computers are organized in the form of layers. As we know computers are capable of doing so many things.

However, a computer is useless without a software.

In association with software, computer can store, process and retrieve information.

Software can be divided into two kinds.

1. System software
2. Application software.

System software is designed to provide platform for other software. Its purpose is to improve the performance of the system and to ease the task of computing.

Examples : Operating systems, Compilers, Linkers, Interpreters etc.

Application software programs are developed for particular application to run and use.

Examples : MS-office for office work.

Operating system Def :

An Operating system is a program that acts as an intermediary between a user of a computer and the computer Hardware.

(or)

An Operating system is a System software which acts as an interface between the user and the Computer Hardware.

The purpose of an operating system is to provide an environment in which a user can execute programs.

The primary goal of an operating system is to make the computer system convenient to use.

The secondary goal is to use the computer hardware in an efficient mannar.

Every operating system must have one operating system to run other programs.

Applications like MS-office, Browser, Notepad, Games etc .... need some environment to run and perform its tasks.

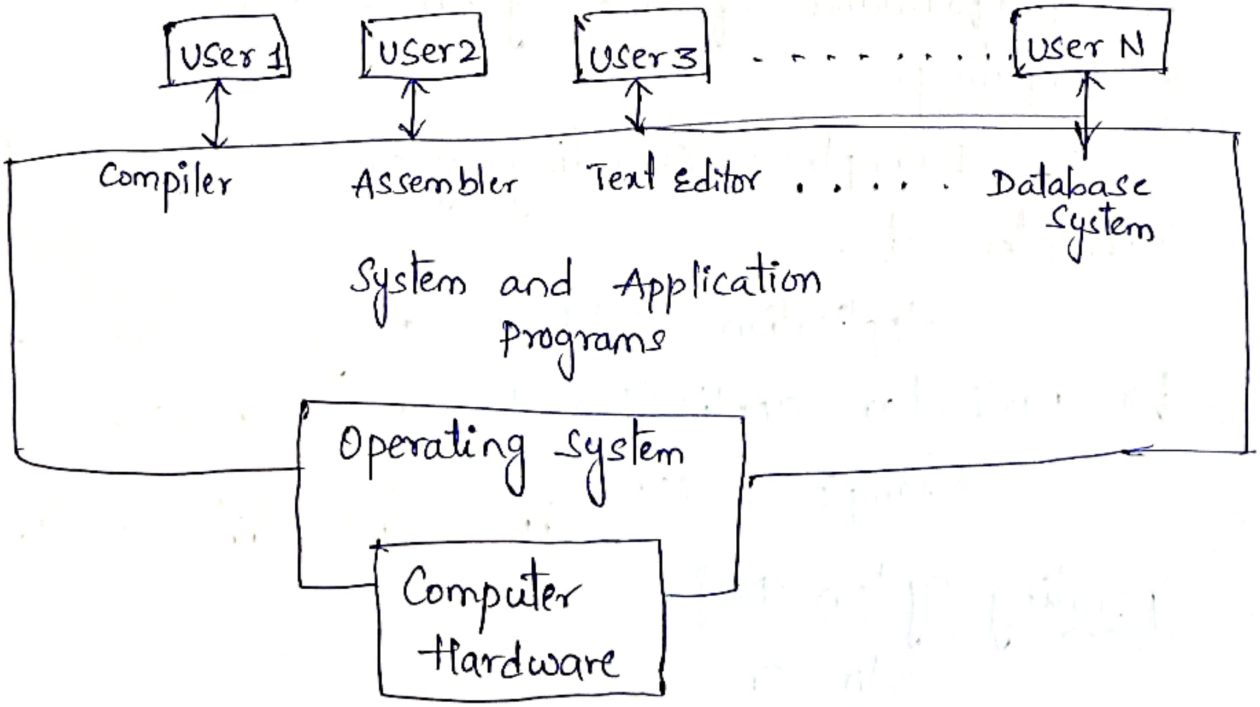


fig: Abstract view of the Components of a Computer System.

The operating system helps you to communicate with the computer without knowing how to speak the Computer's language.

It is not possible for the users to use any Computer or mobile device without having operating system.

Goals and objectives of an operating systems :-

- User friendly
- Convenience and easy to use.
- Throughput :- to improve the performance (or) to Speedup the process.
- Efficiency : The OS improves the efficiency of output. This is because the less time spent in Configuring the System.
- Abstraction of hardware details : It hides the details of hardware which are complex.
- Management of system resources.
- Keeps track of who is using which resource, granting resource requests, and mediating conflicting requests from different programs and users.
- provides efficient and fair sharing of resources among user and programs.

History of operating systems :-

The evolution of operating system is directly dependent on the development of computer systems and how users use them.

Operating systems first developed in the late 1950's to manage tape storage.

The General Motor research lab implemented the first OS in the early 1950's for their IBM 701.

In the mid 1960's, operating system started to use disks.

In the late 1960's, the first version of the Unix operating system was developed.

The first OS built by Microsoft was DOS (Disk operating system). It was built in 1981 by purchasing 86-DOS software from a seattle company.

The present day popular operating system Windows first came into existence in 1985. When the GUI (Graphical User Interface) was created and paired with MS-DOS.

Operating System structure :-

A system as large and complex as a modern OS must be engineered carefully if it is to function properly and be modified easily.

A common Approach is to partition the task into small components, or modules, rather than have one monolithic system.

Each of these modules should be a well defined portions of the system., with carefully defined inputs, outputs, and functions.

1. Simple structure :-

Many OS do not have well defined structures.

Frequently, such systems started as small, simple and limited systems and then grew beyond their original scope.

It was originally designed and implemented by a few people who had no idea that it would become so popular.

It was written to provide the most functionality in the least space, so it was not carefully divided into modules.

In MS-DOS, the interfaces and levels of functionality are not well separated. For instance, application programs are able to access the basic I/O routines to write directly to the display and disk drives. Such freedom leaves MS-DOS vulnerable to errant

Programs, causing entire system crashes when the user program fails.

MS-Dos was also limited by its hardware functionality.

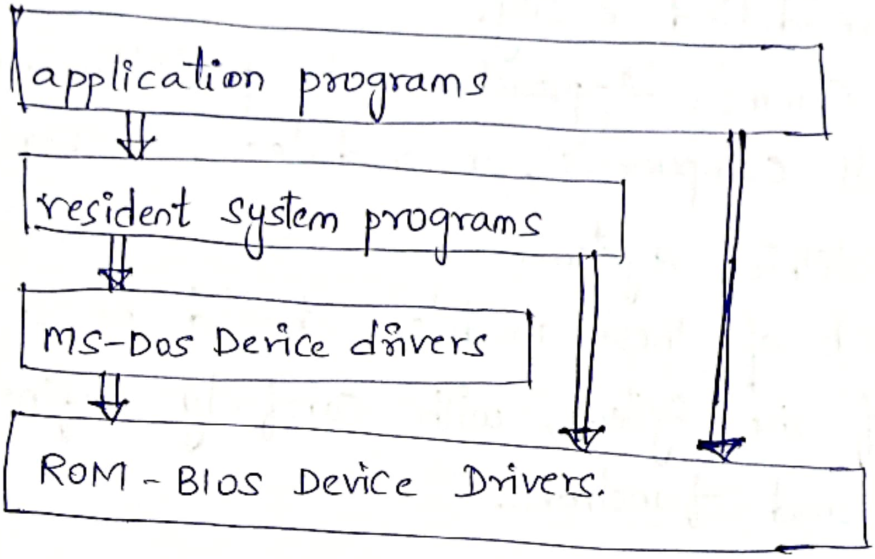


fig: MS-Dos Layer structure

2. Monolithic structure:

Another example of limited structuring is the original UNIX OS. (Monolithic structure)

Like MS-DOS, UNIX initially was limited by hardware functionality.

It consists of two separable parts: The kernel and the system programs.

The kernel is further separated into a series of interfaces and device drivers, which have been added and expanded over the years as UNIX has evolved.

We can view the traditional UNIX OS as being layered to some extent.

Everything below the system call interface and above the physical hardware is the kernel.

Implementers have more freedom in changing the inner workings of the system and in creating modular OS.

Under a top-down approach, the overall functionality and features are determined and are separated into components.

A system can be modular in many ways. One method is the layered approach, in which the OS is broken into a number of layers (levels). The bottom layer (level 0) is the hardware. The highest layer (layer N) is the user interface.

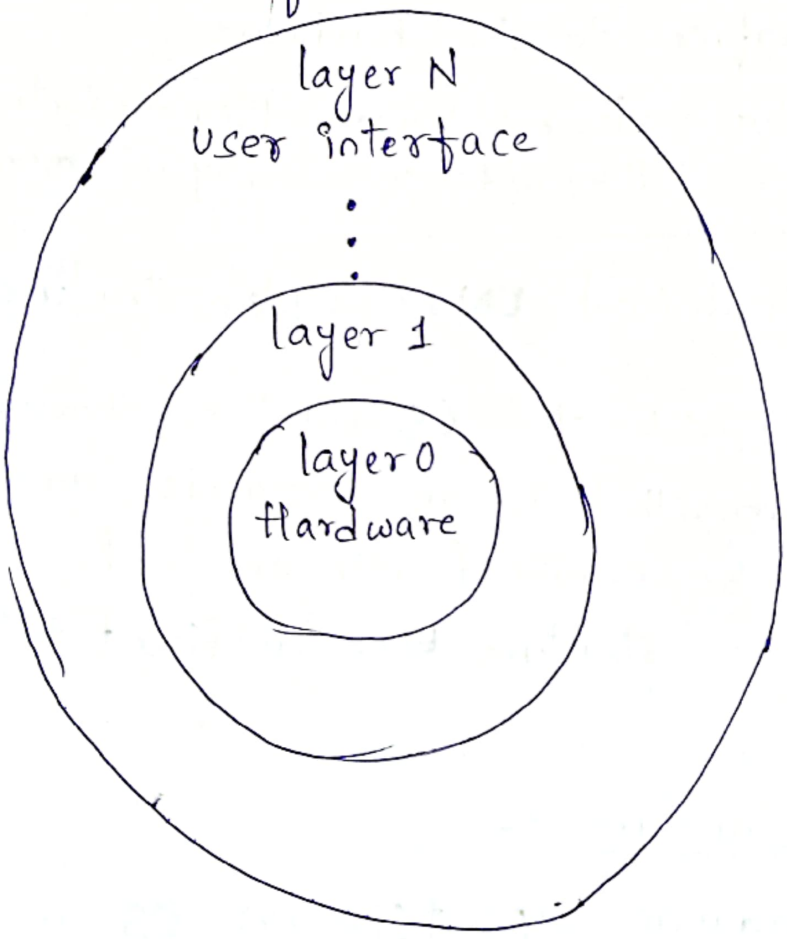


fig: A layered Operating system.

An OS layer is an implementation of an abstract object made up of data and the operations that can manipulate those data.

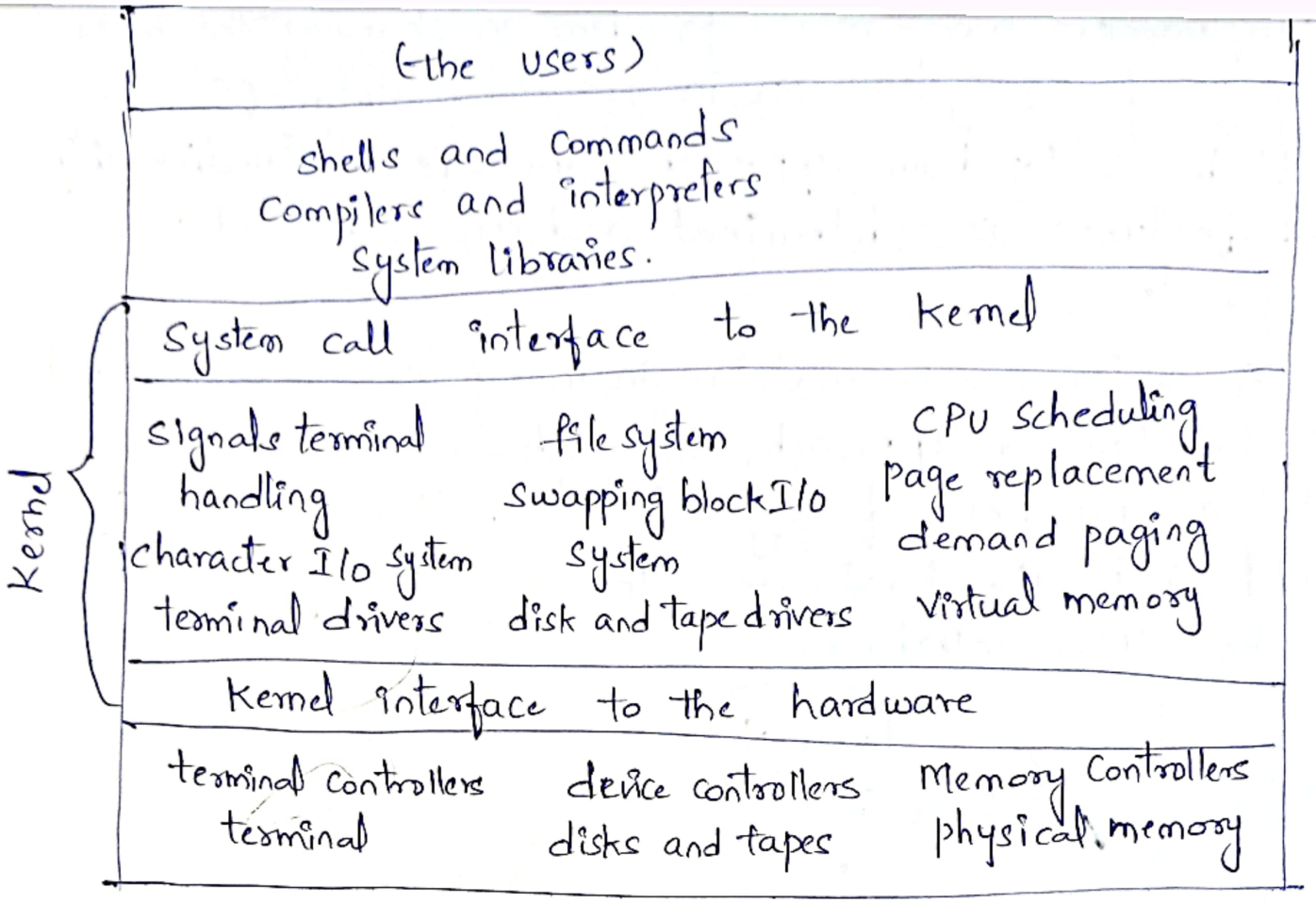


fig: Traditional UNIX system structure

The kernel provides the OS functions through systemcalls.

Taken in sum, that is an enormous amount of functionality to be combined into one level.

This monolithic structure was difficult to implement and maintain.

3. Layered structure :-

With hardware support (Proper), OS can be broken into pieces that are smaller and more appropriate than those allowed by the original MS-DOS and UNIX systems. The OS can retain much greater control over the computer and over the applications that make the use of the computer.

The main advantage of the layered approach is simplicity of construction and debugging.

The layers are selected so that each uses functions and services of only lower-level layers.

This approach simplifies debugging and system verification.

Each layer is implemented only with operations provided by lower-level layers.

A layer does not need to know how these operations are implemented; it needs to know only what these operations do.

Hence, each layer hides the existence of certain datastructures, operations and hardware from higher-level layers.

The major difficulty with the layered approach involves appropriately defining the various layers. Because a layer use only lower level layers, careful planning is necessary.

A final problem with the layered implementations is that they tend to be less efficient than other types. For instance when a user program executes an I/O operation, it executes a system call that is trapped to the I/O layer, which calls the memory management layer, which in turn calls the CPU scheduling layer, which is then passed to the hardware.

At each layer, the parameters may be modified, data may need to be passed and so on. Each layer adds overhead to the system call.

4. Microkernels :

In monolithic structure we have already seen that the kernel became large and difficult to manage.

In mid-1950's, researchers at Carnegie Mellon University developed an OS called Mach that modularized kernel using the microkernel approach.

This method structures the OS by removing all non essential components from the kernel and implementing them as system and user-level programs.

The result is a smaller kernel.

Typically however, microkernels provide minimal process and memory management in addition to a communication facility.

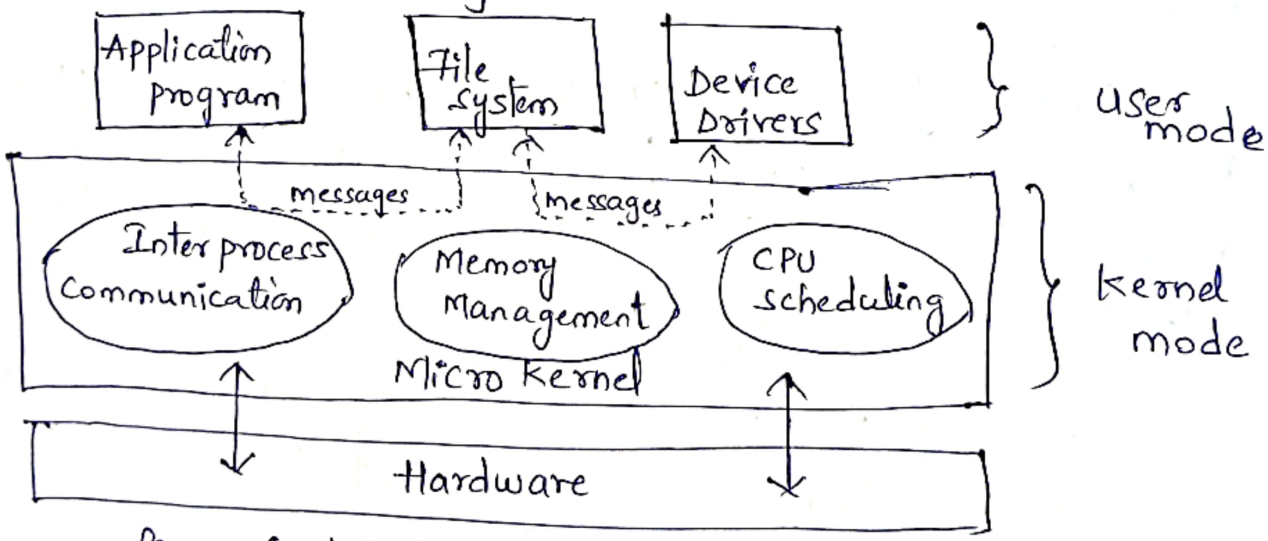


fig: Architechture of a typical Micro kernel.

The main function of a microkernel is to provide communication between the client program and the various services that are also running in user space.

Communication is provided through message passing.

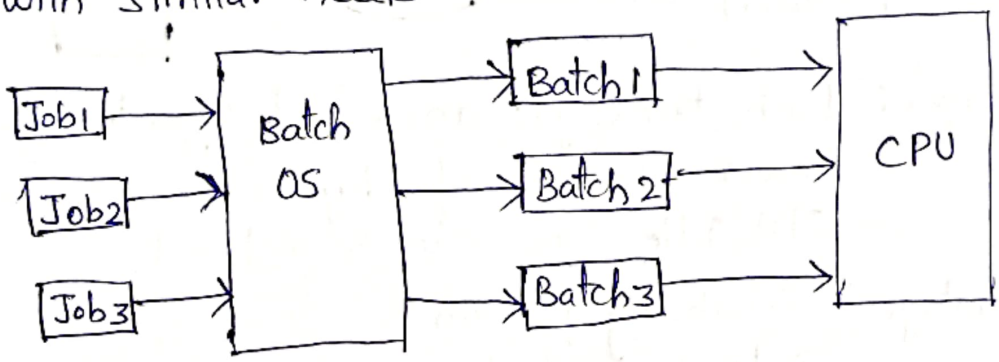
One benefit of microkernel approach is that it makes extending the os easier.

Disadvantage is that the performance of microkernel can suffer due to increased system function overhead.

Types of Operating Systems :-

1. Batch operating systems :- This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having same requirement and group them into batches.

It is the responsibility of the operator to start jobs with similar needs.



Advantages : Processors of the batch systems know how long the job would be when it is in queue.

Multiple users can share the same batch systems.

Idle time is less.

Easy to manage large work repeatedly.

Disadvantages : Operators should be well known with the batch systems because hard to debug.

Sometimes it is costly and other jobs have to wait for unknown period of time.

Examples of batch OS : payroll systems, Bank statements etc.

2. Time sharing operating systems :-

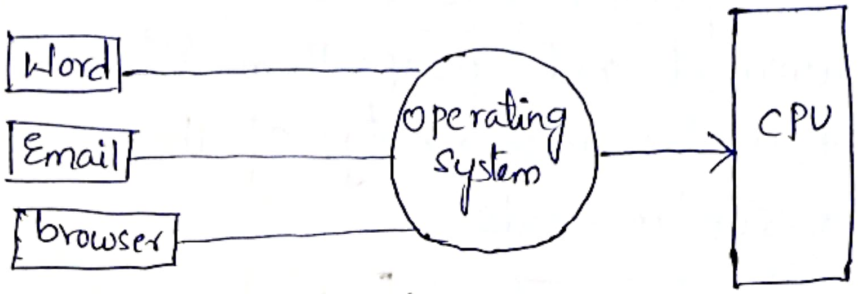
Each task is given some time to execute so that all the tasks works smoothly.

Each user gets the time of CPU as they use

a single system. These systems are also known as multi-tasking systems.

The time that each tasks gets to execute is called time quantum.

After this time interval is over OS switches to the next task.

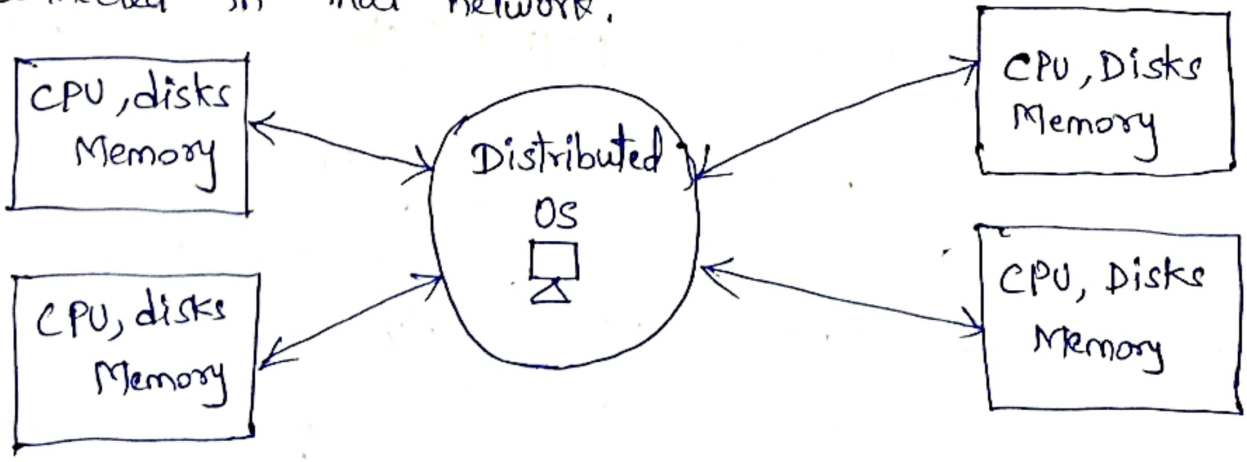


Advantages: Each tasks gets an equal oppurtunity.
Fewer chances of duplication of software.
CPU idle time can be reduced.

Disadvantages: Reliability problem.
Data communication problem.
One must have to take care of the security and integrity of the users programs and data.

3. Distributed operating system :-

These are referred to as loosely coupled systems. The major benefit of working with this type of OS is that it is always possible that one user can access the files or software which are not actually present on his system but some other system connected within this network i.e., remote access is enabled within the devices connected in that network.



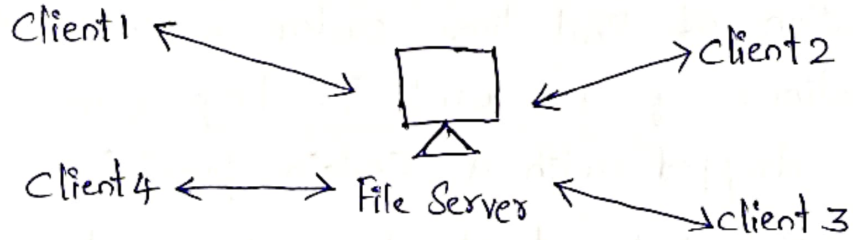
Advantages : Failure of one will not affect the other.
Less Delay.
Data exchange speed is high.
Load on host computer reduces.

Disadvantages : Failure of main network will stop the entire communication.
Expensive.

4. Network Operating system :-

These systems run on a server and provide the capability to manage data, users, groups, security applications and other networking functions.

These type of OS allow shared access of files, Printers, Security, applications and other networking functions over a small private network.



Advantages : Highly stable centralize servers.
Security Concerns are handled through servers.
New Technologies and the Hardware upgradation are easily integrated into the system.
Server access is possible remotely from different locations.

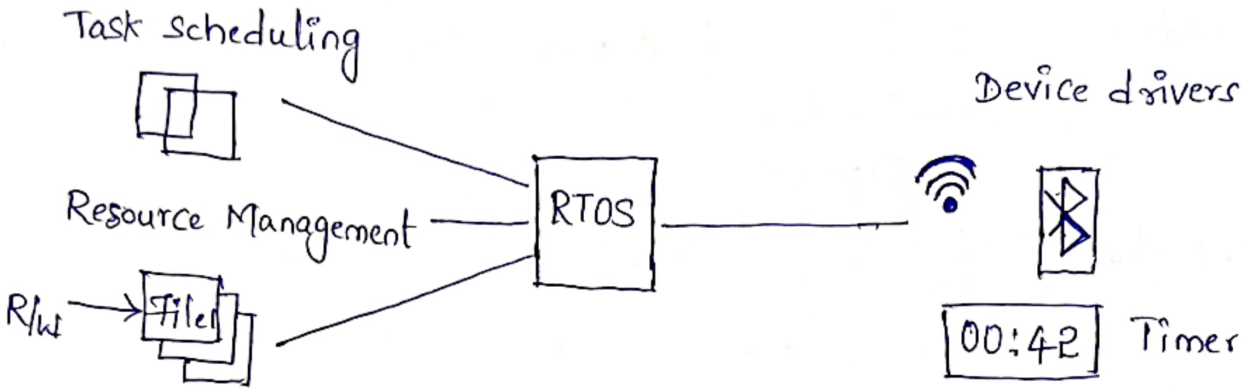
Disadvantages : Servers are costly.
Maintenance and updates are required regularly.

5. Real Time Operating systems :-

These type of operating systems serves real time systems. The time interval required to process and respond to inputs is very small. This time interval is response time.

Real time systems are used in Missile systems and Air traffic Control systems.

Two types of Real time OS: Hard Real time Systems and soft Real time systems.



In Real time systems each job carries a certain deadline within which the job is supposed to be completed; Otherwise, the huge loss will be there, over even if the result is produced, it will be completely useless.

Application of Real time system exists in the case of military applications if you want to drop a missile, then the missile to be dropped with a certain precision.

Advantages: Easy to layout, develop and execute real time applications under the real time OS.

Disadvantages: very costly.

6. Multiprogramming Operating systems :-

Multiprogramming is an extension to batch OS where the CPU is always kept busy.

Each process needs two types of system time. CPU time and I/O time.

In multiprogramming environment, when a process does its I/O, the CPU can start the execution of other processes. Therefore multiprogramming improves efficiency of the system.

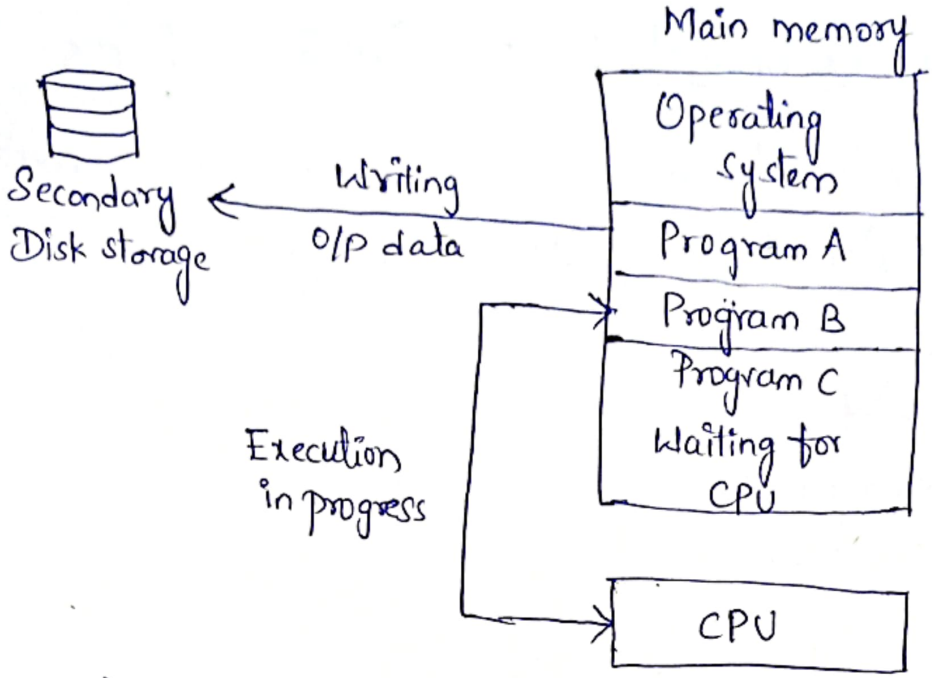


fig: Jobs in Multiprogramming system.

Advantages: CPU can't be idle.
Response time will be reduced.
Maximizes the CPU utilization.

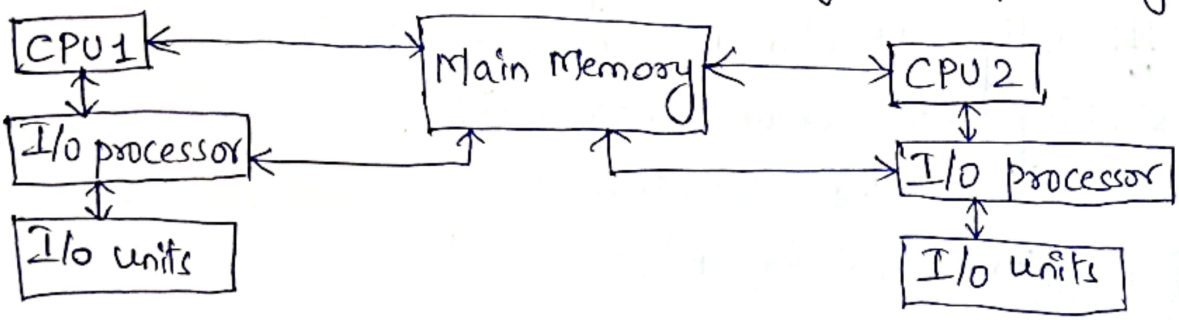
Disadvantages: They do not provide user interaction with the Computer system.

7. Multiprocessing Operating systems :-

In multiprocessing operating system, paralell computing is acheived.

There are more than one processor present in the system which can execute more than one process at the same time.

This will increase the throughput of the system.



Advantages: Increased Reliability and throughput.

Disadvantages: Multiprocessing OS is more Complex and Sophisticated as it takes care of multiple CPU's simultaneously.

Operating System Components :- (or) -Functions of OS

The components of an OS plays a keyrole to make a variety of Computer system parts work together.

1. Process Management
2. File Management
3. Network management
4. Main memory management
5. Secondary storage management
6. I/O Device management
7. Security management
8. Command Interpreter system.

1. Process Management :

The process management component is a procedure for managing many processes running simultaneously on the operating system.

Every running software application program has one or more processes associated with them.

For example : When you use a search engine like chrome, there is a process running for that browser program.

Process management keeps processes running efficiently. It also uses memory allocated to them and shutting them down when needed.

The execution of a process must be sequential So, atleast one instruction should be executed on behalf of the process.

Functions of Process Management :-

- process creation and deletion
- Suspension and resumption of process
- synchronization and communication process.

2. File Management :-

A file is a set of related information.
It commonly represents both the programs and data.
Data files can be alphabetic, numeric or alpha-numeric.

Functions of file management :

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

3. Network Management :-

Network management is the process of administering and managing Computer networks.

It includes performance management, provisioning of networks, fault analysis and maintaining the quality of service.

A distributed system is a collection of computers or processors that never share their memory and clock.

In this type of system, all the processors have their local memory and the processors communicate with each other using different communication cables, such as fibre optics or telephone lines.

Network management provides the following functions, such as :

- Distributed systems help you to provide various computing resources in size and function.

- It helps to access shared resources that help computation speed up or offers data availability and reliability.

4. Main Memory Management :-

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

It should be mapped to absolute addresses and loaded inside the memory to execute a program.

The selection of a memory management method depends on several factors.

However, it is mainly based on the hardware design of the system.

Each algorithm requires corresponding hardware support.

Functions of Memory Management :-

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

5. Secondary Storage Management :-

The most important task of a computer system is to execute programs.

These programs helps you to access the data from the main memory during execution.

This memory of the computer is very small to store all data and programs permanently.

The Computer system offers Secondary storage to backup the main memory.

Today's modern computers use hard-drives / SSD as the primary storage of both programs and data. However, the Secondary storage management works with storage devices, such as USB flash drives and CD/DVD drives.

Functions of secondary storage management :-

- Storage allocation.
- free space management.
- Disk scheduling.

6. I/O Device Management :-

One of the purpose of an OS is to hide the peculiarities of specific hardware devices from the user.

Functions of I/o Management :-

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

7. Security Management :-

The Various processes in an OS need to be secured from other activities.

Therefore various mechanisms can ensure those processes that want to operate, files, memory, CPU and other hardware resources should have authorization from the OS.

Security refers to a mechanisms for controlling the access of programs, processes or users to the resources

defined by computer controls to be imposed, together with some means of enforcement.

Security can improve reliability by detecting latent errors at the interfaces between component subsystems.

8. Command Interpreter systems :-

One of the most important components of an OS is its command interpreter.

The command interpreter is the primary interface between user and the rest of the system.

Many commands are given to OS by control statements.

Its function is quite simple, get the next command statement and execute it.

Operating System Services :-

An operating system provides an environment for the execution of programs. It provides services to programs and to the users of those programs.

These OS services are provided for the convenience of the programmer, to make programming task easier.

* User Interface :-

Almost all operating systems have a User Interface. This interface can take several forms. one is command line interface (CLI), which uses text commands and a method for entering them.

Another is a Batch interface, in which commands are directives to control those commands

are entered into files, and those files are executed.

Most commonly GUI (Graphical user Interface) will be used, GUI is user friendly desktop service interface.

* Program Execution and handling :-

Starting of programs, managing their execution and communicating their results.

Here the system must be able to load a program into memory and to run that program, end execution either normally or abnormally.

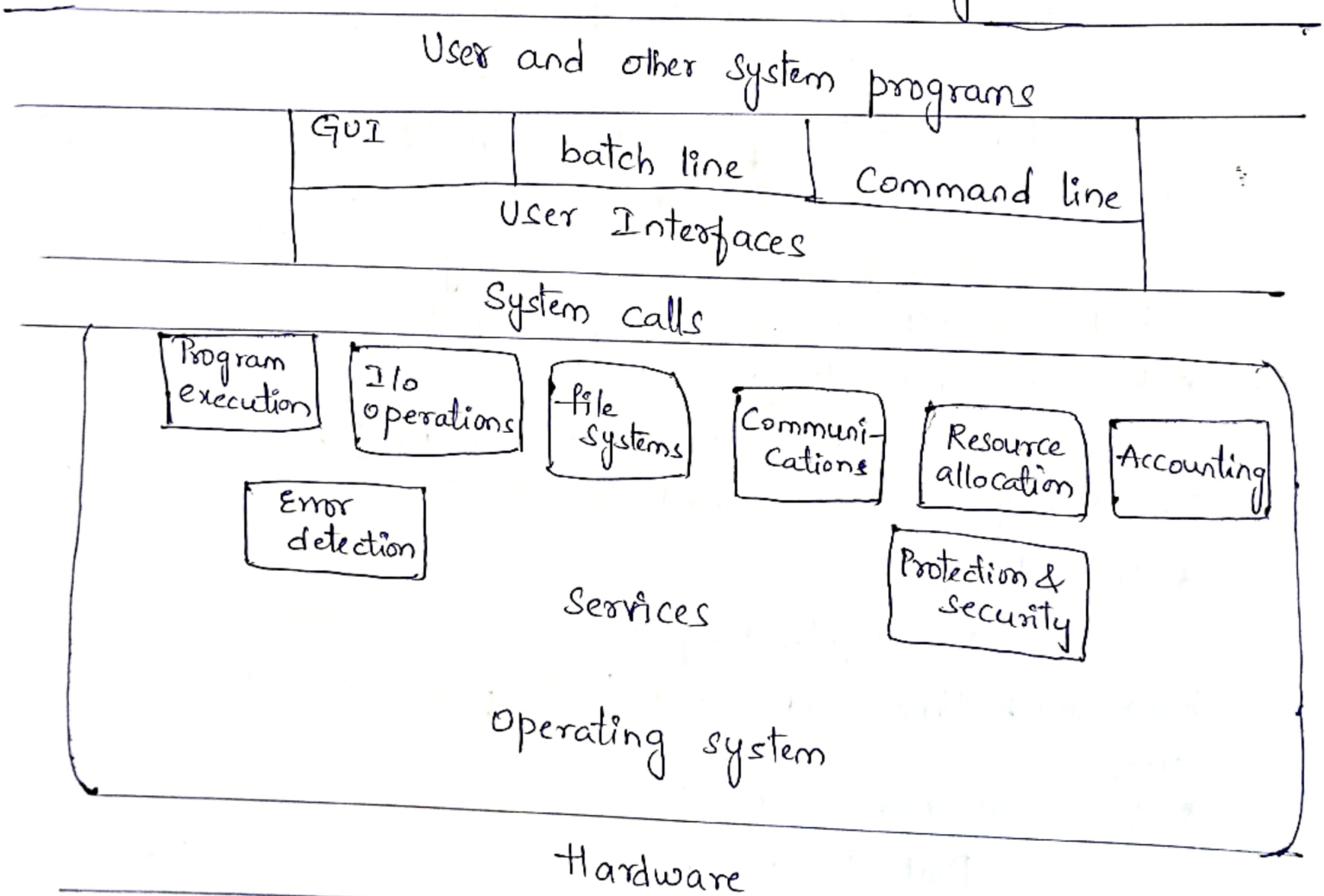


fig: View of operating system services.

* I/O Operations:

A running program may require I/o device or a file.

* File Systems:

Creating, managing and manipulating the data in files and directories.

* Communication:

Processes may exchange information on the same Computer or between the computers over a network.

Communications may be shared via a shared Memory or Message passing.

* Error Detection:

Operating system needs to be constantly aware of errors which may occur in the CPU and memory hardware or in I/O device or in user program.

For each type of error OS should take the appropriate action to ensure Correct and consistent computing.

Debugging facilities can greatly enhance the users and programmers ability to efficiently use the system.

* Resource Allocation:-

When multiple users or multiple jobs are running Concurrently, resources must be allocated to each of them.

* Accounting:

To keep track of which users use the resources and how much time and what kind of resources they will use.

* protection and Security:

Protection involves in ensuring that all the access to the system resources must be controlled.

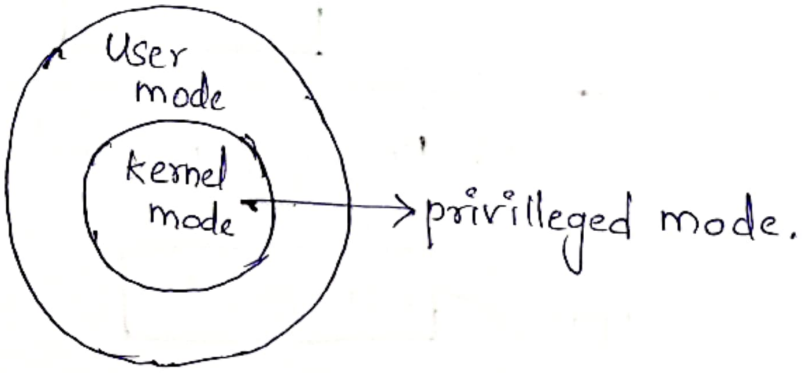
Security of the system from outsiders requires user authentication, extends to defend / secure external I/O devices from invalid access.

System Calls :-

System calls provide an interface to the services made available by an OS.

To understand system calls, first one needs to understand the difference between the kernel mode and the user mode of a CPU.

Every operating system supports two modes : User mode and Kernel mode.



Kernel Mode : When the CPU is in kernel mode, the code being executed can access any memory address and any hardware resource.

Hence kernel mode is a very privilleged mode.

If a program crashes in kernel mode then the entire system will be halted.

User mode : When the CPU is in user mode, the programs don't have direct access to memory and hardware resources.

In user mode, if any program crashes, only that particular program is halted.

That means the system will be in a safe state even if a program in user mode crashes.

Hence, most programs in OS run in user mode.

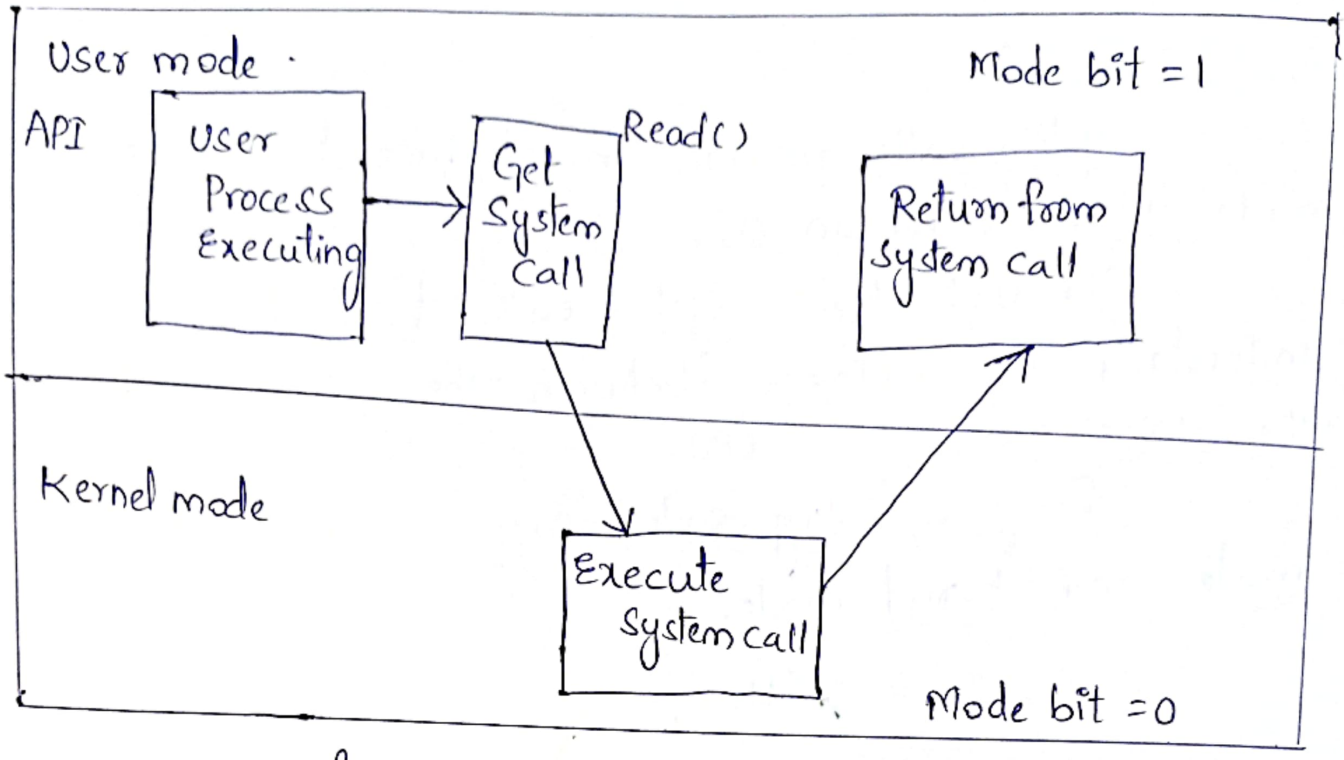


fig: Kernel Vs user mode

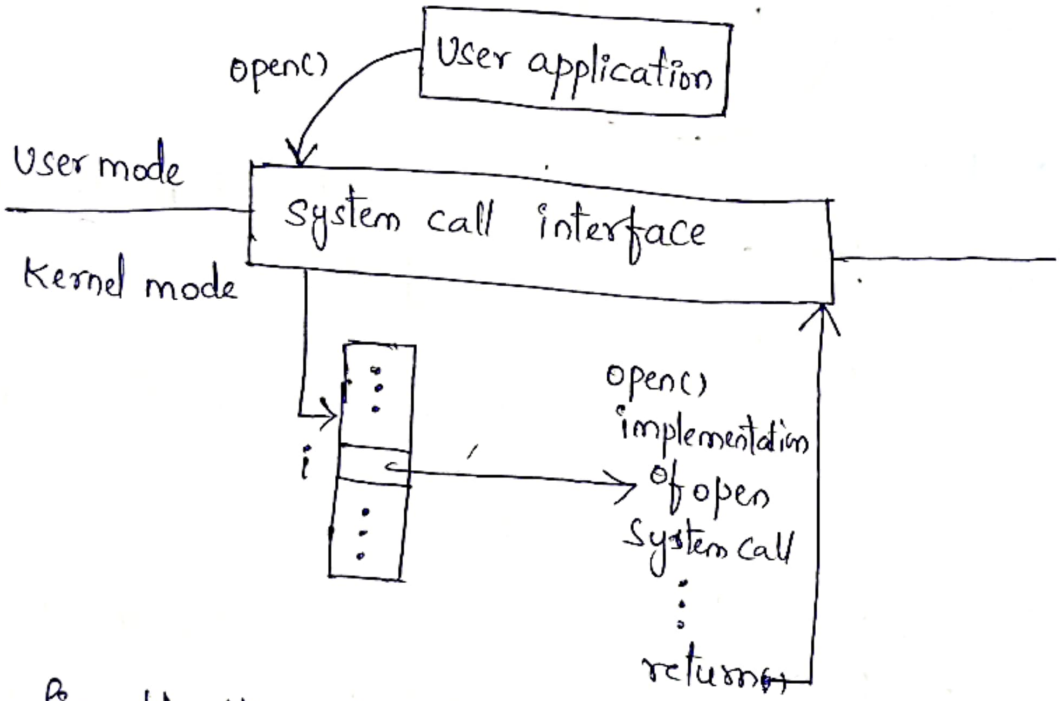


fig: Handling of user application invoking the open() system call.

When a program in user mode requires access to RAM or a hardware resource, it must ask the kernel to provide access to that resource. This is done via a system call.

When a program makes a system call, the mode is switched from user mode to kernel mode. This is called a context switch.

Then kernel provides the resource which the program requested. After that, another context switch happens which results in change of mode from kernel mode back to user mode.

System call is the programmatic way in which a computer program requeste a service from the kernel of the OS.

These calls are generally available as routines written in C and C++.

Types of System calls :-

System calls can be roughly grouped into five major categories.

1. Process control
2. File Management
3. Device Management
4. Information Maintenance
5. Communications.

1. Process Control :
   - end, abort.
   - load, execute.
   - Create process, terminate process
   - get process attributes, set process attributes.
   - Wait for time.
   - wait event, signal event.
   - allocate and free memory.
2. File Manipulation :
   - Create file, delete file
   - open, close.

- read, write, reposition.
- get file attributes, set file attributes.

3. Device Manipulation.:
   - request device, release device
   - read, write, reposition.
   - get device attributes, set device attributes.
   - logically attach or detach devices

4. 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.

5. Communications:
   - Create, delete communication connection.
   - Send, receive messages.
   - Transfer status information.
   - attach or detach remote devices.