Operating System

Definition :- It is an interface between user and computer hardware.

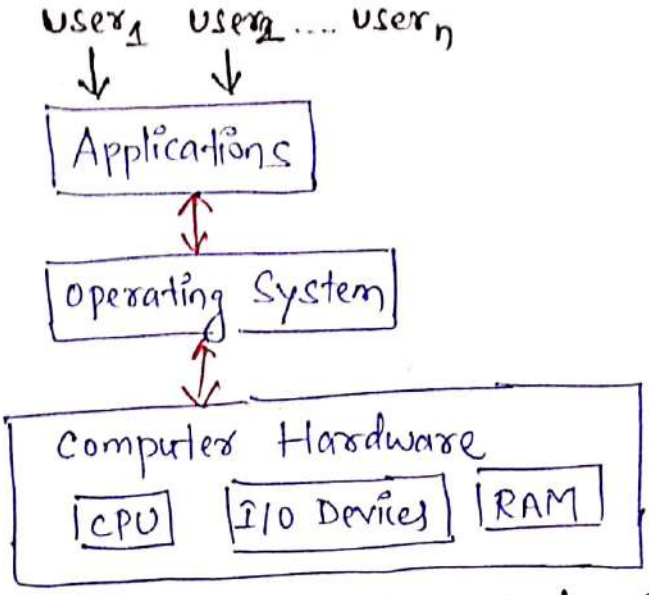


Figure : Computer system conceptual view

Operating system is a system software, that manages the computer hardware.

An operating system is also Known as Resource Allocator.

Resources

- Hardware type — e.g. Devices, Memory
- Software type — eg Files

Goal of an Operating System

(convenience)

(1) Primary Goal of O.S is convience ∧ means easy to use.

(2) Secondary Goal of O.S is Efficiency or Throughput (number of task executed per unit time)

(2)

Windows

Primary Goal : convenience

Secondary Goal : Efficiency

Linux

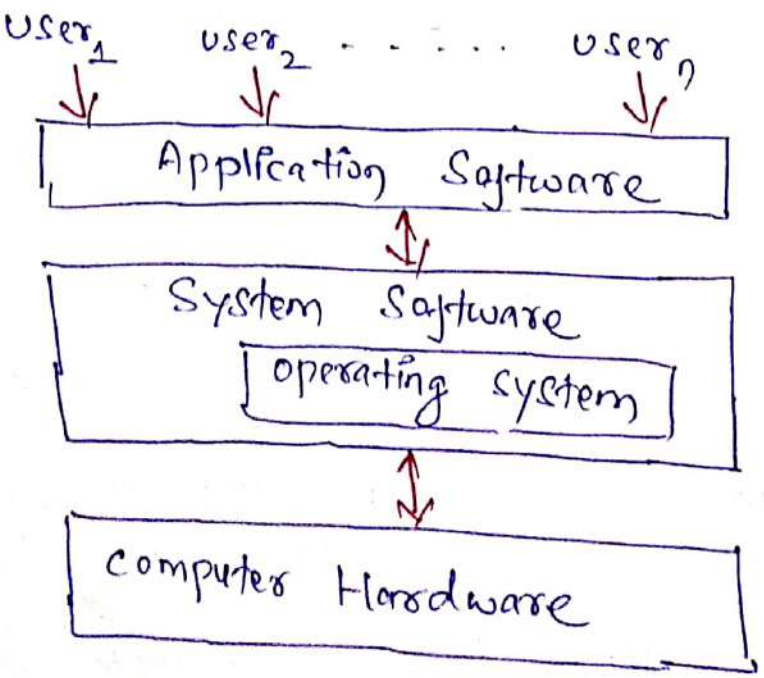
Primary Goal : Efficiency

Secondary Goal : convenience

∴ Linux is more Efficient than windows.

Operating System as a User Interface

- Computer system consists of software and Hardware. to solve specific problems.
- User, Application program, operating system and the Hardware are the components of the computer systems.



The purpose of OS :->

(i) OS provide an interface between the computer hardware and computer user.

(ii) Allocation and use of computer resources among the computer user is controlled by OS.

Functions of Operating System

(1) Resource Management (or) Device Management
(2) Process Management
(3) Storage Management (or) File System Management
(4) Memory Management

Resource Management

- Resource Management includes sharing resources in different ways. Time and space are the two concepts for resource sharing.

  Time :-> Time slot is allocated to each program so that the request for the resources are managed properly.

  Space :-> Consider the main memory where the main memory is divided into several execution area for all running programs so that each one can be executed in parallel.

- A computer communicates information through its input and output devices. Processes access those devices

(4) through operating system system call.

System call

It is a request made by the user program to the operating system for any kind of service.

e.g. write, read, open, close etc..

System call is also known as supervisor call (SVC).

Thus operating system attempts to manage those divices in a manner that allows them to be efficiently shared among the processes requiring them.

Process Management

- A program under execution is called process.
  OS manage execution of multiple processes in computer system so that resource allocation is also managed.
- Various CPU scheduling algorithms such as FCFS, SJF, SRTF, RR, LJF, LRTF, HRRN, Priority Based scheduling

FCFS : First Come First Serve
SJF : Shortest Job First
SRTF : Shortest Remaining Time First
RR : Round Robin
LJF : Longest Job First
LRTF : Longest Remaining Time First
HRRN : Highest Response Ratio Next

Storage Management (or) Hard disk

- Computer process information. The information must be transmitted, processed and stored in secondary storage device i.e Hard disk.
- Storage Management is done by File system.

File system is the way in which files are processed and stored in proper organized manner.

- A file system stores and organizes data in storage device.

Major File systems are :=>

(i) File allocation Table (FAT) :=> For Windows OS

(ii) Global File system (GFS) :=> For Linux OS

(iii) Hierarchical file system (HFS) :=> For Macintosh OS

NTFS (New Technology File System) is the default file system for windows

Linux OS uses open-source NTFS for file system.

Mac OS have read-only support for NTFS.

Memory Management

- Memory is shared by a number of processes.
- OS must manage the allocation of memory to processes and control the memory management hardware that determines which memory locations a process may access.

(c)

- The memory management is for RAM (Random Access Memory). Due to limitation of RAM size, it is the task of OS to manage RAM in more effective manner.
- Allocation and de-allocation of memory (RAM) is done by OS.

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 Operating System does the following activities for memory management –

- 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 / Process Management

In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called **process scheduling**. An Operating System 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 / Resource Management

An Operating System manages device communication via their respective drivers. It does the following activities for device management −

- Keeps tracks of all devices. Program responsible for this task is known as the **I/O controller**.
- Decides which process gets the device when and for how much time.
- Allocates the device in the efficient way.
- De-allocates devices.

File Management / Storage Management

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

An Operating System does the following activities for file management −

- Keeps track of information, location, uses, status etc. The collective facilities are often known as **file system**. FS is the way in which files are processed & stored in proper organized manner.
- Decides who gets the resources.
- Allocates the resources.
- De-allocates the resources.

A FS stores and organizes data in storage device.

Types of operating system

Operating systems are there from the very first computer generation and they keep evolving with time. In this chapter, we will discuss some of the important types of operating systems which are most commonly used.

Batch operating system

The users of a batch operating system do not interact with the 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 programmers leave their programs with the operator and the operator then sorts the programs with similar requirements into batches.

The problems with Batch Systems are as follows –

- Lack of interaction between the user and the job.
- CPU is often idle, because the speed of the mechanical I/O devices is slower than the CPU.
- Difficult to provide the desired priority.

Classification of Operating Systems

- Batch O.S
- Multi-programming O.S
- Multi-tasking O.S
- Multi-processor O.S
- real-time O.S

(1) Batch Operating System

- This type of operating system was used in 1970-80.
- In batch operating system, the user do not interact with the computer directly.
- Collection of jobs is called Batch.
- Each user prepares his/her job on an off-line device like punch cards and submit it to the computer operator.
- To speed up processing, jobs with similar needs are batched together and run as a group.
- The programmers leave their programs with the operator and the operator then sort the program with similar requirements into batches.

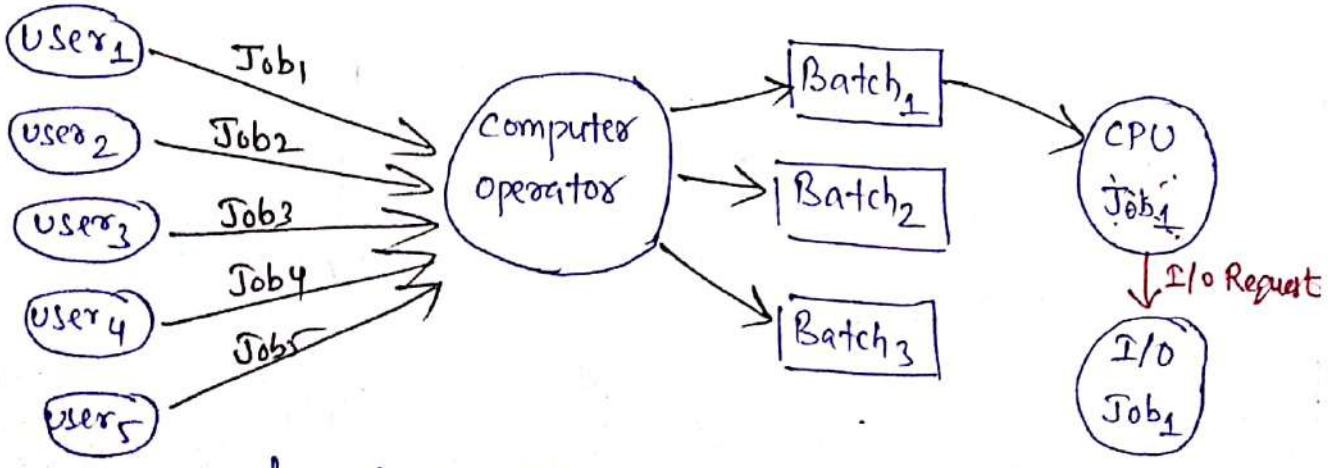


fig. Batch O.S

- Every Job has two time :

Time
- CPU time
- I/O time

Process
- CPU Bound
- I/O Bound

- In Batch OS, if one job is executed completely, then only another job will be scheduled on the CPU.

  Thus increase CPU ideal time.

  ⇒ throughput of system will decrease.

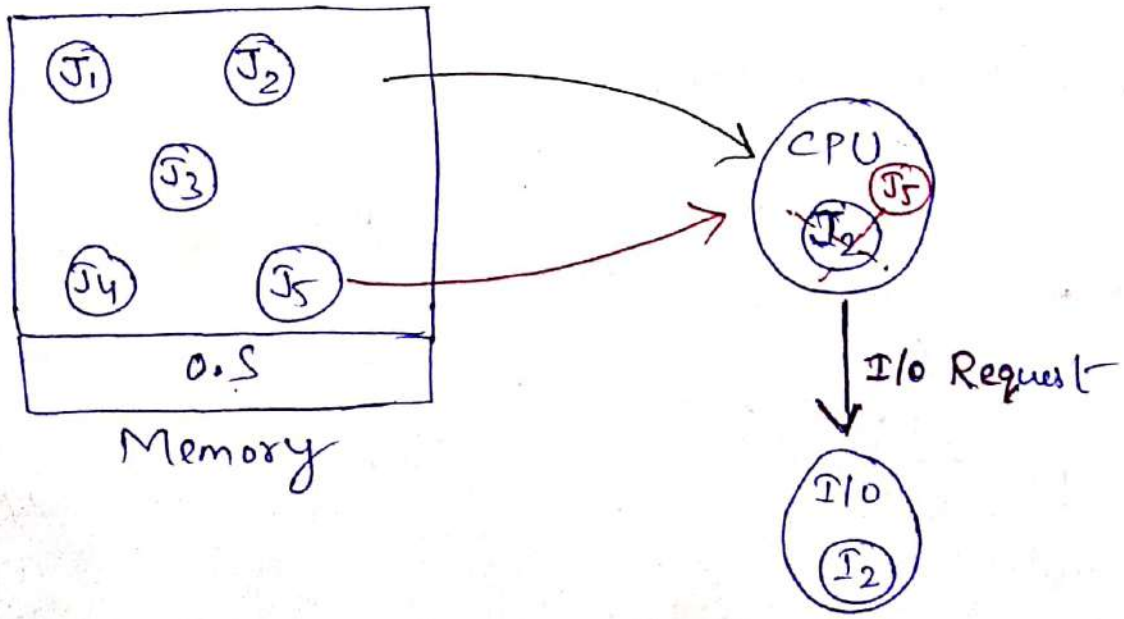
Throughput :⇒ Number of jobs completed per unit time.

- Example of Batch O.S : IBM OS/2

(2) Multi-programming Operating System

- CPU remains idle in batch O.S. At any time, either CPU or I/O divice was idle in batch system.

  To Keep CPU busy, more than one program / Job must be loaded for execution. It increases CPU utilizations.



Multi-programming O.S increases → CPU utilization
→ throughput of system

- If the job is leaving CPU to perform I/O operation then another job which is ready for execution will be scheduled onto CPU.
- System choose one job to run through CPU scheduling algorithms.
- In Multi-programming O.S, CPU never sit idle.
- The main focus of multi-programming O.S is to maximize CPU usage.

Example of Multi-programming O.S are Windows, Linux, macintosh.

Advantages

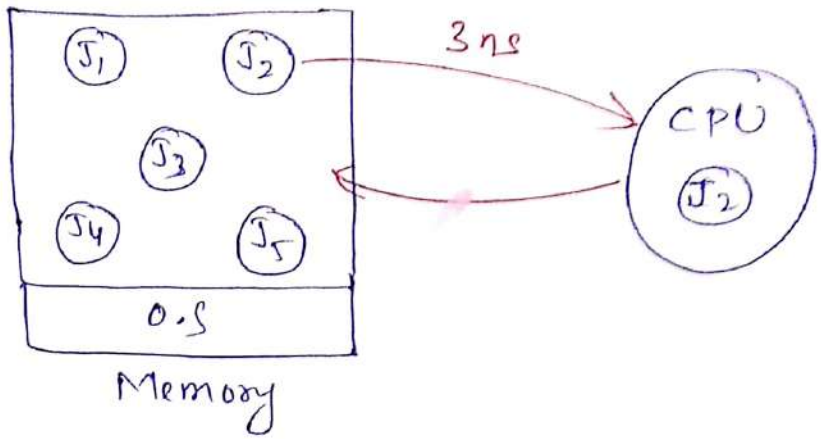
(i) CPU utilization is high
(ii) It increased degree of multiprogramming.

Disadvantages

(i) CPU scheduling algorithm is required.
(ii) Memory Management is also required.

(3) Multi-tasking O.S (or) Time sharing O.S

- Multi-tasking is an extension to multi-programming O.S
- The jobs will be executed on CPU in time-sharing mode.



Example : Windows, Linux, Macintosh

Time-sharing operating systems

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 systems, the objective is to maximize processor use, 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 in few seconds at most.

The operating system 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 Timesharing operating systems are as follows –

- Provides the advantage of quick response.
- Avoids duplication of software.
- Reduces CPU idle time.

Disadvantages of Time-sharing operating systems are as follows –

- Problem of reliability.
- Question of security and integrity of user programs and data.
- Problem of data communication.

Multiprocessor System / Parallel Systems.

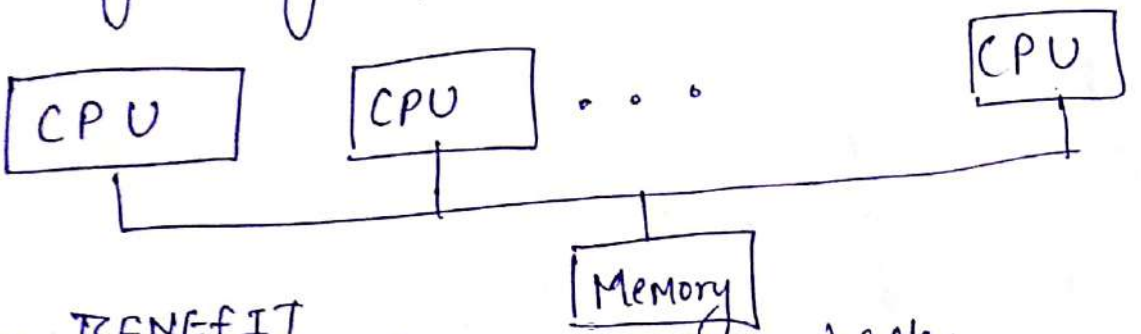
⇒ System which have two or more processors in close communication, sharing the computer bus and sometimes the clock, memory and peripheral devices.

The multiprocessor systems are of 2 types.

1. Asymmetric multiprocessing in which each processor is assigned a specific task. A master processor controls the system; the other processors either look to the master for instruction or have pre defined. tasks. This scheme defines a master slave relationship. The master processor schedules and allocates work to the slave processors

2. Symmetric Multiprocessing (SMP) in which each processor performs all tasks within the operating system.
   - SMP means all processors are peers.

   Ex: Solaris, a commercial version of Unix designed by SUN Microsystems.



The BENEFIT of this model is that many processes can run simultaneously

1. Increased reliability
2. Increased throughput
3. Economy of scale.

Drawbacks.

1. There should be proper control of I/O to ensure that the data reach the appropriate processor
2. CPU's are separate, one may be sitting idle while another is overloaded.

Network operating System

A Network Operating System 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 network operating system is to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks.

Examples of network operating systems include Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD.

The advantages of network operating systems 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 network operating systems 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 data processing 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**. So in this method, 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-time systems can be used as a control device in a dedicated application. A real-time operating system must have well-defined, fixed time constraints, otherwise the system will fail. For example, Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

There are two types of real-time operating systems.

Hard real-time systems

Hard real-time systems guarantee that critical tasks 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.

Soft real-time systems

Soft real-time systems are less restrictive. A critical real-time task gets priority over other tasks and retains the priority until it completes. Soft real-time systems have limited utility than hard real-time systems. For example, multimedia, virtual reality, Advanced Scientific Projects like undersea exploration and planetary rovers, etc.

Operating system services

An Operating System provides services to both the users and to the programs.

- It provides programs an environment to execute.
- It provides users the services to execute the programs in a convenient manner.

Operating System Services

Operating System provide the following services :

(1) Graphical User Interface
(2) Program Execution
(3) File Management
(4) Communication
(5) Error Detection
(6) Resource Management and Allocation
(7) Protection & Security
(8) Input / output Operating

(1) User Interface

- User interacts with operating system by using user interface.
- It converts high level language into machine language and vice versa
- Types :
  - (i) GUI (Graphical User Interface)
    - includes buttons, graphs, icons, folders, menus
    - Mouse, Keyboard, monitor for I/O
  - (ii) CLI (Command Line Interface)
    - user enters text command for performing operating.
    - Fastest way to interact with machine
      Linux, Unix

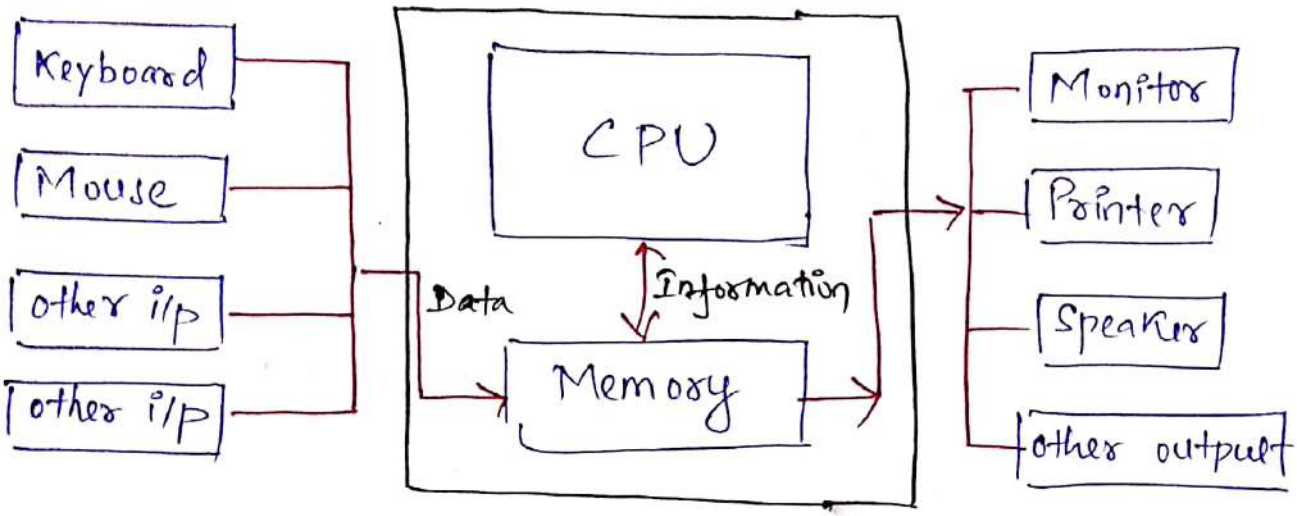
(iii) Batch Interface

- Commands are entered into file and executed.
- Non interactive interface

(2) Program Execution

- One of the main purpose of a computer systems is to allow the user to execute programs. So OS provides an environment where the user can conveniently run programs. The user does not have to be concerned with the memory allocation proces. Such services are provided by the OS.
- It includes
  - Loading program into memory
  - Execution
  - Run program
  - Handle errors
- Before executing the program, it is loaded into the memory by O.S.
  once program loads into memory, its start execution.
  Program finishes its execution with error or without error.

(3) Input / output Operations



- Each program requires an input and produces output. This involves the use of I/O.
- The OS hides the details of hardware for I/O to the user.
- All the user sees is that I/o has been performed without any details.
- So O.S. by providing I/O makes it convenient for the users to run programs.

(4) File Management-

- Computer can store files on the hard disk (secondary storage) for long term storage purpose.
  Each of these media has its own properties like speed, capacity, data transfer rate and data access methods.
- O.S organizes the file system into directories for easy navigation and usage.

These directories may contain files and other directions.

- Following are the major activities of OS with respect to File management.
  - OS programs needs to read a file or write a file
  - OS controls the permission of a given file
  - OS provides creating, deleting and editing of files
  - OS provides an management of directories.

- Create and delete by name
  Read and write files and directories
  Copy and paste
  Rename
  Print or search for a given file.

(5) Communication :->

- In the computer system, processes needs to communicate and exchange data. This is Known as Inter Process communication.
  Other components of O.S also have to communicate and access shared resources at times. called Synchronization Mechanism.
  This communication is handled by O.S.

- With respect the communication, the OS provides the following services
  - two processes may require data to be exchanged between them.
  - communication may be implemented either by Shared Memory or Message Passing.

⑥ Error Detection

- Errors may occur in CPU, in I/O devices or in memory hardware. or in the user program.
- Errors in I/O devices can be connection failure on a network, or lack of papers in the printer.

"No audio / video output is installed."

"Location is not available, I/O device error"

"audio / sound problems on windows 10"

- Errors in memory hardware can be

"corrupted pendrive / memory card"

↳ Message is: The disk is write Protected

"DDR3 Memory type not supported"

"Parity check / Memory parity Error"

↳ The system has halted

"Blue Screen Error".

- Errors in user programs can be

Software installation Error in Windows

"The application has failed to start because its side-by-side configuration is incorrect".

"Stack overflow error"

illegal memory location access error.

- Following are the major activities of an operating system with respect to error handling :->

  > OS constantly remains aware of possible errors.
  > OS takes the appropriate action to ensure correct and consistent computing.
  > Debugging facilities can greatly enhance the user's and programmers abilities to use the system efficiently

(7) Resource Management & Allocation

- When there are multiple jobs running at the same time, the resources must be allocated to each of them.
- Many different types of resources are managed by the operating system.

Example :-> ✓CPU cycles and Speed of CPU.
  ↳ for this, OS have CPU scheduling routines

✓I/O devices and file storage
  ↳ there may also be routines to allocate printers, modems, USB storage drives.

✓Main Memory allocation and deallocation.

(8) Protection And Security

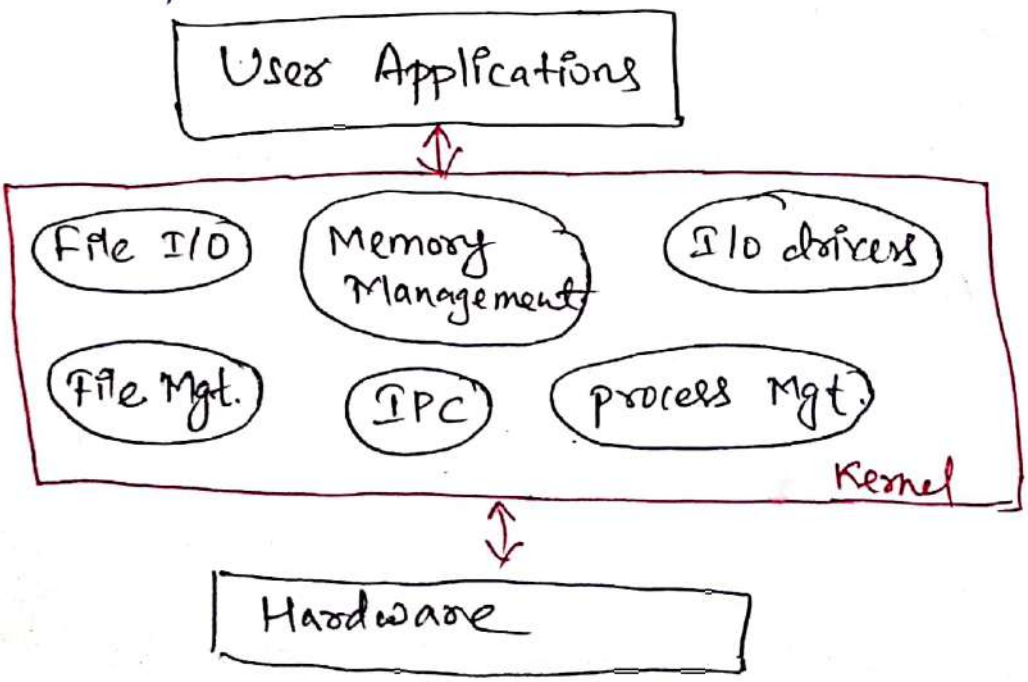
- Protection involves ensuring that all access to system resources is controlled.
- Following are the major activities of an operating system with respect to protection.
  - (i) OS ensures that all access to system resources is controlled.
  - (ii) OS ensures that external devices are protected from invalid access attempt.
  - (iii) OS provides authentication feature for each user by means of a password.
- Security of system starts with requiring each user to authenticate himself or herself to the system usually by means of password to gain access to the system resources.

NOTE :→ An operating system provides an environment for the execution of programs. It provides certain services to programs and to the user of those programs

→ Operating System Services : Introduction.

Monolithic Architecture

- Monolithic operating system is one of the simple operating system architecture. and was used in very early times.
- Traditional UNIX operating system uses monolithic Kernel architecture. The entire OS runs as a single program in Kernel mode.
- The programs contains OS core functions and device drivers.
- In monolithic Kernel design, all the functions of OS like process management, memory management, device management, file management are all integrated under a single unit in the Kernel address space.



Example of operating systems that uses monolithic Kernel are :→ Linux, BSD (BerKeley Software Distribution was unix-based (1971–1995, University of California), Free BSD, Open BSD, Net BSD, Solaris, OS-9, HP-UX, Windows (95, 98, Me), openVMS etc.

Advantages

1. Simple to design and implement
2. Simplicity provides speed on simple hardware.
3. Monolithic Kernels are fast. The code is accessible to all the components of the operating system and this code can be executed without much restriction making the overall system fast.
4. Monolithic Kernel provide direct communication between the components of the operating system that makes system more efficient to work.

Disadvantages

1. Monolithic Kernels are more prone to errors and bugs as user process runs in same address space as that of Kernel.
2. Fault tolerance is low
3. Harder to port because of dependency on code.

(iv) Adding / removing any feature or functionality in a monolithic system are quite difficult and often require rewriting and recompiling of the whole code again.

Note :- In monolithic Kernel architecture, there is one code (program) consisting of routines, subroutines and data is written and stored in memory. The components of OS use this code and data to provide necessary functionality.

* A monolithic Kernel is an OS architecture where the entire operating system is working in Kernel space.

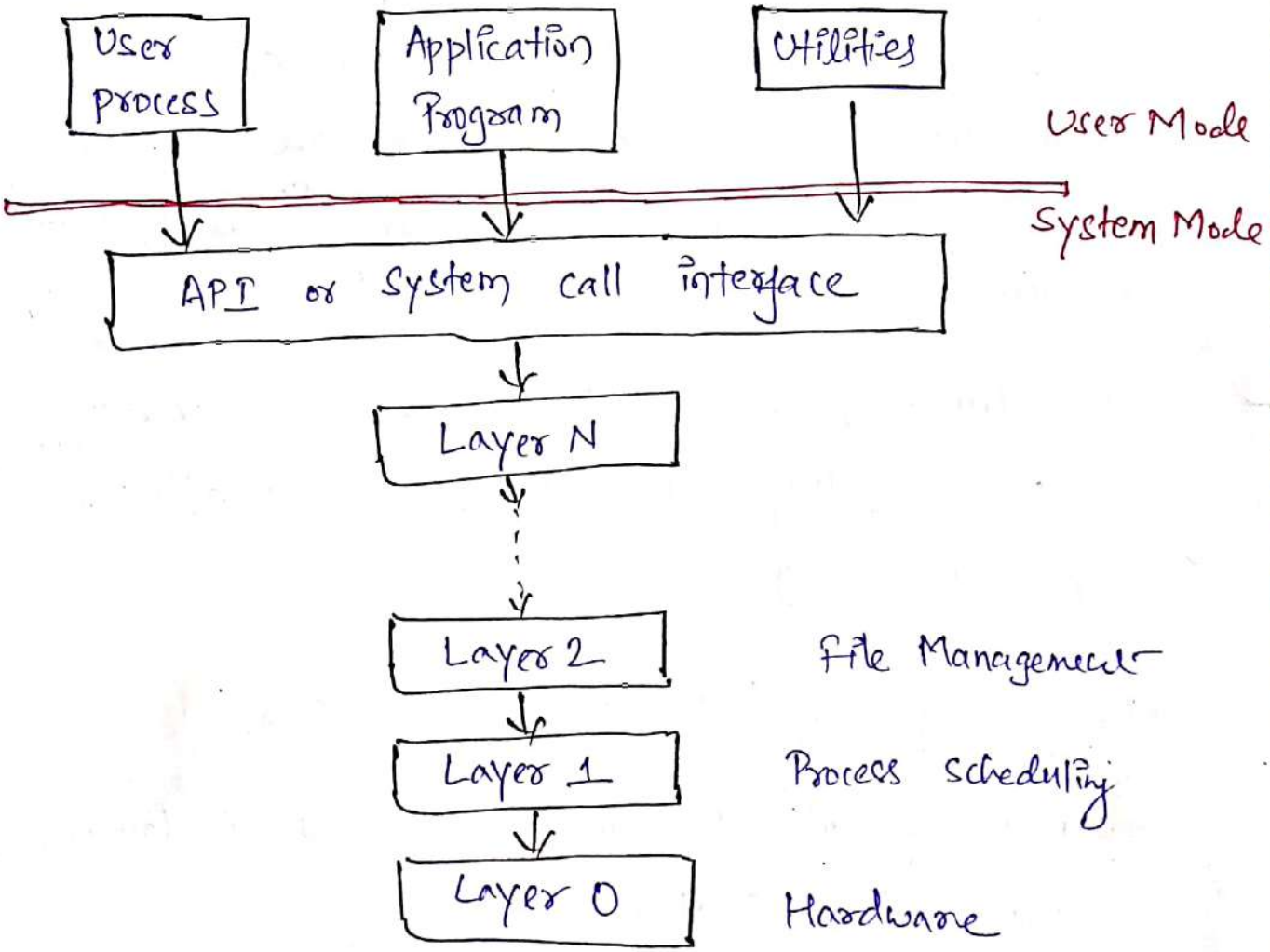
* Monolithic Kernel has the following :

File I/O
Memory Mgt
Process Mgt
I/O Drivers
Memory Drivers
Interrupt Drivers
Device Drivers
File System
Schedulers
SysCall Handler
IPC

Operating System Structure

(1) Layered Structure

- In layered approach, the operating system consists of several layers where each layer has a well defined functionality and each layer is designed, coded and tested independently.



- The layers are arranged in increasing order of abstraction means that the lowest layer (0) interacts and deals with the underlying hardware

and the topmost layer (N) provides an interface to the application programs and user processes.

- Each layer relies on the service of layers below it.
- The communication takes place only between adjacent layers.

for example : Layer 3 can request a service from layer 2 immediately below it and it can provide service only to the layer 4 immediately above it.

- Each layer knows what services are provided by the layer above it but the details as how these services are provided are hidden.
- The different layers can be file management layer, memory ~~mange~~ management layer, communication management layer, user program layer etc.

Example of Layered operating system is

THE operating system : consists of 6 layers

MULTICS system

- The Layered architecture of operating system was developed in 1960's.
- In this approach, the operating system is broken into number of layers.
- The bottom layer (Layer 0) is the hardware layer and the highest layer (Layer N) is the user interface layer.
- Example of layered architecture of operating system is OS/2 by IBM.

Another example is earlier version of windows NT. (Windows New Technology) (1993 - 1996)

Windows NT two versions → Windows NT (acts as server in networks)

→ Windows NT workstation for stand-alone (client workstation).

Advantages of Layered Structure

(1) In the layered approach, it is easier to add any new features or make changes in one layer without affecting the other layers of the operating system.

(2) Any bug or problem in a particular layer is restricted to that layer only. That layer can

be debugged, corrected or redesigned without affecting the functionality of other layer.

(3) Easy to add new layer as and when required

DS's - advantages of Layered Structure

(1) The more the layers, the more is the overhead incurred to maintain them.

(2) If the functionality is not properly divided among layers, it might be possible that one layer has two much functionality. If this happens that layer may be overburden leading to overall low system performance.

(3) Appropriate definition of various layers and careful planning of the proper placement of the layer.

Kernel

- Kernel is a software code that resides in the central core of a operating system.
- It is called the Heart of an OS.
- It has complete control over the system.

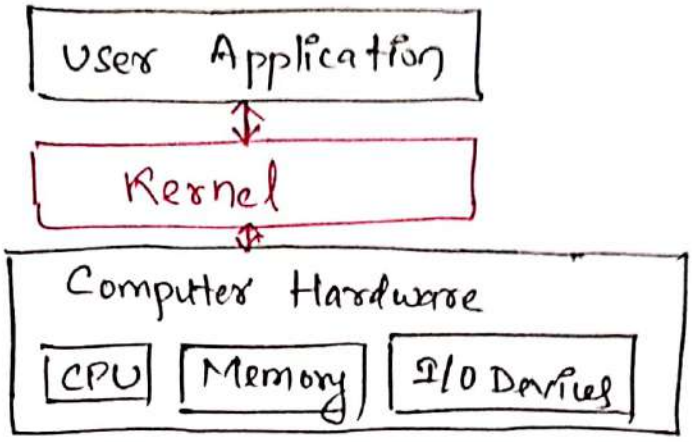


fig Position of Kernel

- When operating system boots, Kernel is the first part of the operating system to load into memory.
  Kernel remains in memory for the entire duration of the computer session.
- The Kernel code is loaded into a protected area of memory.
- Any program has → User space
  → Kernel space

User space
⇒ Less privileges
↳ Limited Access
⇒ If it crash, affects single process.

Kernel space

⇒ high privilege

↳ all access : user / Kernel space

⇒ If exception occurs, crash whole OS.

- The Kernel performs its tasks in Kernel space. Executing processes and handling interrupts are performed by Kernel in Kernel space.
- User performs its task in user area of memory.
- Memory is divided into system area and user area. This memory separation is made in order to prevent user data and Kernel data from interfering with each other.
- The Kernel provides services :
  - (i) Process Management
  - (ii) Memory Management
  - (iii) File Management
  - (iv) I/O Management

  System calls are used for providing this type of service.
- Kernel includes :
  - (i) Scheduler
  - (ii) Supervisor
  - (iii) Interrupt handler
  - (iv) Memory Manager.

~~Supervisor~~ Scheduler

~~Supervisor~~ allocates Kernel's processing time to various processes.

Supervisor :⇒ Supervisor grants the permission to use computer system resources to each process.

Interrupt handler :⇒ Interrupt handler handles all requests from the various hardware divices which compete for the Kernel's services.

Memory Manager ⇒ Memory Manager allocates space in memory for all users of the Kernel services.

Types of Kernel

1) Monolithic Kernel
2) MicroKernel
3) Reentrant Kernel

Functions of Kernel

(1) Abstraction
Hardware access, Schedaling, IPC

(2) Arbitration
Access privileges, preventing system crash.

- Kernel can be classified further into two categories:

(i) MicroKernel (ii) Monolithic Kernel

MicroKernel is the one in which user services and Kernel services are kept in separate address space.

In Monolithic Kernel, user services and Kernel services both are kept in same address space.

Dual Mode of Operation

mode bit

0 — current inst^{n} is executing in Kernel mode

1 — current inst^{n} is executing in user mode

- In hardware level, the inst^{n} are executed in two different modes: user mode and Kernel mode
- Dual mode of operation is used to provide the protection and security to the user programs and the operating system from errant user (false/ hackers).

- O.S always runs in Kernel mode.
- Privileged instructions are executed in Kernel mode and non-privileged instructions are executed in user mode.
- In which particular mode, the currener instruction is executing, is identified by using "Mode bit".
- privileged instruction means the instructions which require more security and protection. thus executed in Kernel mode.

Some of Known Privileged Instructions

(1) context switching
(2) Disabling interrupts
(3) set the value of timer
(4) clearing the memory to the process.
(5) changing the memory map (moving of process from one location to another).
(6) I/O operation (Reading the data from file of the Hard disk).

Some of Known Non-Privileged Instructions

(1) Reading the time of the clock.
(2) Reading the status of processor
(3) Sending final print output to the printer.

Definition of Monolithic Kernel

- The monolithic Kernel manages the system resources between application and hardware of the system.
- But unlike microKernel, the user services and Kernel services are implemented under same address space. This increases the size of Kernel and the size of operating system.
- Monolithic Kernel provides CPU scheduling, memory management, file management and other operating system functions through system calls.
- As user services and Kernel services both reside in same address space, this results in the fast executing operating system.
- One of the drawbacks of the monolithic Kernel is if any one service fails entire system is crashed.
- If a new service is to be added in monolithic Kernel, the entire operating system is to be modified.

Definition of MicroKernel

- In microKernel, the user services and the Kernel services are implemented in different address space.
- The user services are kept in user address space and Kernel services are kept under Kernel address space.

This reduces the size of the Kernel and further reduces the size of operating system.

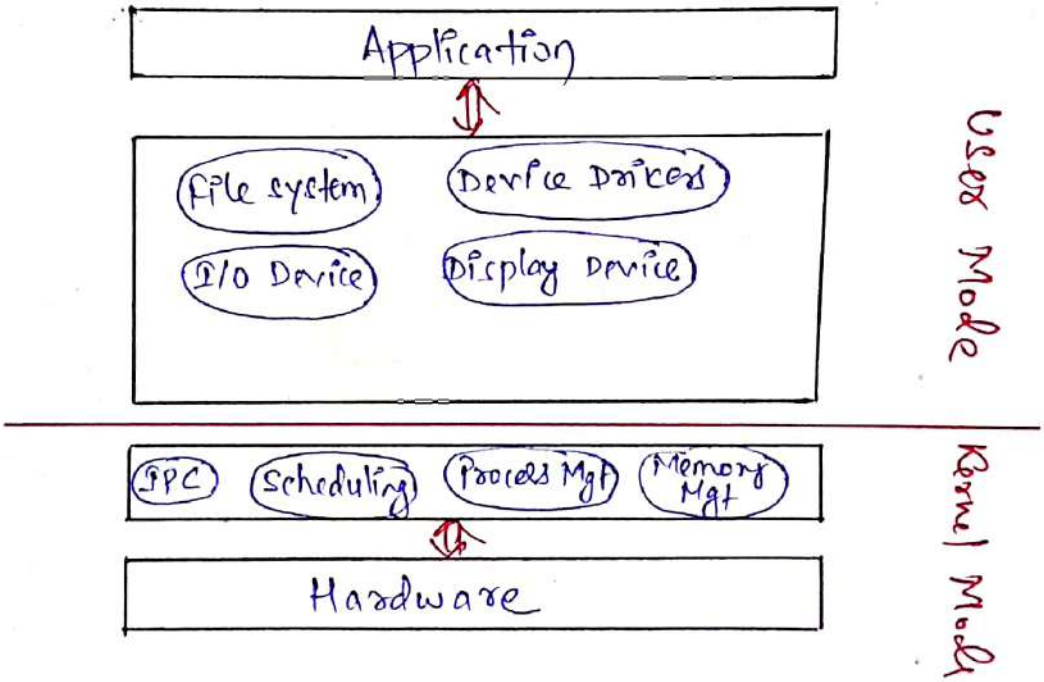
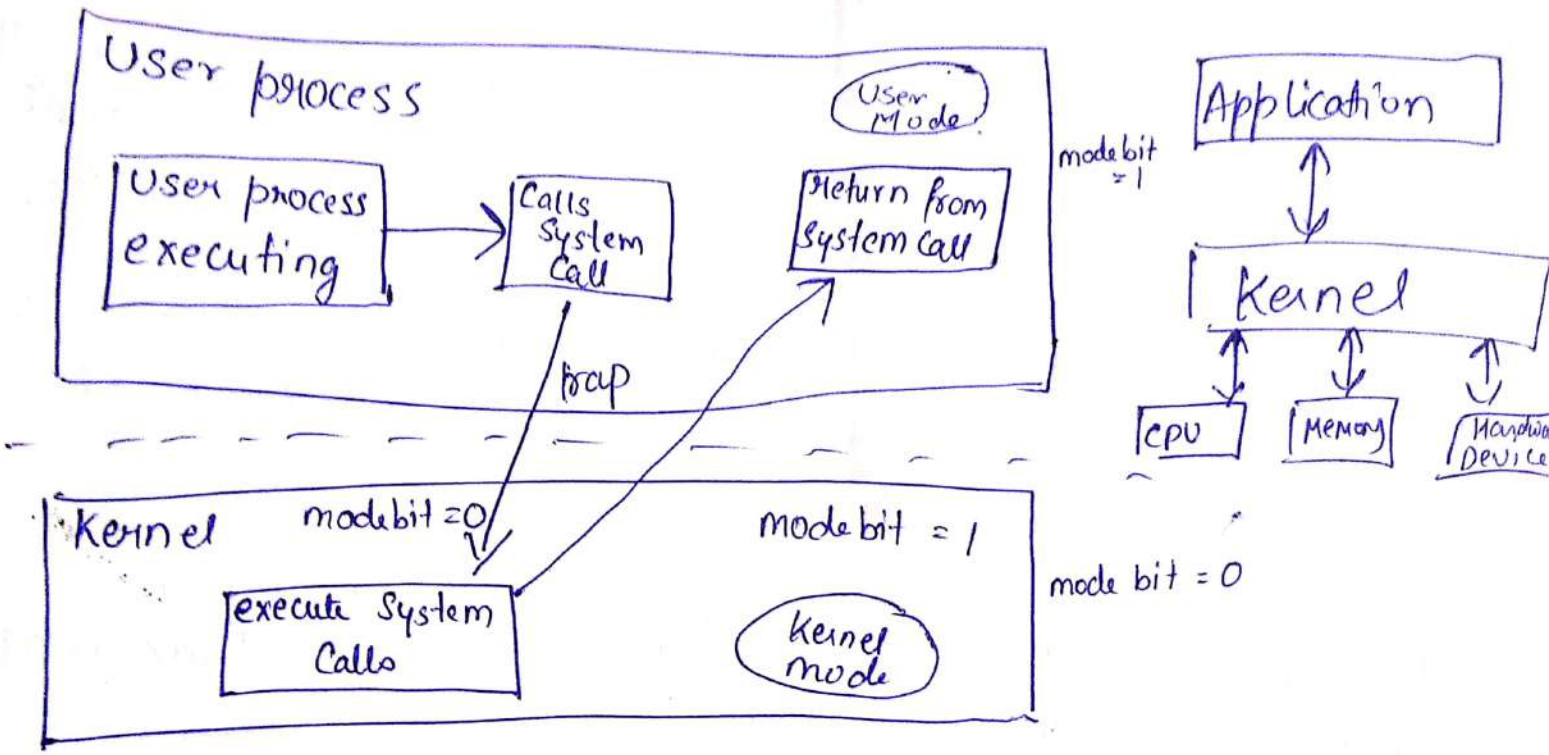


fig MicroKernel Architecture.

- The communication between the client program / application and services running in user address space is established through message passing. They never interact directly. This reduces the speed of execution of microKernel.
- In a microKernel, the user services are isolated from Kernel services so if any user service fails it does not affect the Kernel services.

Hence the operating system remain unaffected.

- The microKernel is easily extendable. means if the new services are to be added, they are added to user address space and hence, the Kernel space do not require any modification.



Basis For comparison	MicroKernel	Monolithic Kernel
Basic	In microKernel, user services and Kernel services are kept in separate address space.	In monolithic Kernel, both user service and Kernel service are kept in same address space.
Size	MicroKernel are smaller in size	Monolithic Kernel is larger than microKernel.
Execution	Slow Execution	Fast Execution
Extendible	MicroKernel is easily extendible.	The monolithic Kernel is hard to extend.
Security	If a service crashes, it does not effect on working of microKernel.	If a service crashes, the whole system crashes in monolithic Kernel.
Code	To write a microkernel, less ~~more~~ code is required	To write a monolithic Kernel, More code is reqd.
Debugging	Easy debugging and management	debugging is difficult
Examples	Mac OS, Windows NT, Symbian, HURD, Minix, Coyotos, QNS, L4Linux.	Linux, BSD (FreeBSD, OpenBSD, NetBSD), Windows (95, 98, Me), Solaris, DOS, OpenVMS, HP-UX, AIX.

⇒ The main function of the micro kernel is to provide a communication facility between the client program & the various services that are running in user space.

Reentrant Kernels

- A reentrant kernel is one that consists of executable code (programs and data) stored in memory such that several processes can use this code at the same time without hindering the working of other processes.
  Thus in reentrant kernel several processes may be executing in kernel mode at a time.

For example:
a process 'A' wants to perform read operation on disc, then the kernel will send this request to disc controller. The disc controller will handle this read operation request and kernel will assign CPU to some other process 'B'. When the process 'A' read operation is completed, the interrupt will notify the kernel so that process 'A' can resume the execution.

- The different concurrently processes executing in Kernel mode also share data. The data modified by one process will affect the functioning of other processes using that data, so to avoid this condition a reentrant Kernel uses ~~enen~~ reentrant functions and locking mechanism.
- Reentrant functions are the one which allow process to modify only the local variables and do not affect the global data variables.
- In locking mechanism, several processes can execute reentrant function concurrently but only one process at a time can execute non-reentrant function.

This will ensure that global data is modified by only one process at a time. and thus each process has same copy of global data being shared among them. In other words, by using locking mechanism there will be no chance of data corruption or multiple different copies of global data.

This is how reentrant Kernel works, UNIX is a good example of reentrant Kernel.

Components of OS:

1. Kernel
2. process Execution
3. Interrupt
4. Memory Management
5. Multitasking
6. Security
7. User Intuface
8. Networking

System Calls

- System Calls provide an interface to the services made available by an operating system. These calls are generally available as routines written in C and C++, although certain low-level tasks (for example, tasks where hardware must be accessed directly), may need to be written using assembly-language instructions.

Types of System Calls:

- Process Management / Process Control:
  1. end / abort
  2. load / execute
  3. create process / terminate process
  4. get process attributes, set process attributes
  5. wait for time
  6. wait event, signal event
  7. allocate and free memory
- File Management
  1. create file, delete file
  2. open, close
  3. read, write, reposition
  4. get file attributes, set file attributes.

- Device Management
  1. request device, release device
  2. read, write, reposition
  3. get device attributes, set device attributes
  4. logically attach or detach devices.

- Information Maintenance
  1. get time or date, set time or date
  2. get system data, set system data
  3. get process, file, or device attributes
  4. set process file

- Communications!
  1. create, delete communication connection
  2. send, receive messages
  3. transfer status information
  4. attach or detach remote devices.

What is Shell ?

- Shell is the interface between user and Kernel.
- It is a command line interpreter.
- The user can enter commands to the shell. Then it interprets the commands to perform the required task.

Further, it executes programs and shell scripts.

- A shell script is a set of commands.
- The user should follow the standard syntax to write commands to the shell.

Types of shell

1. Bourne shell (sh)
2. Korn shell (ksh)
3. Bourne Again Shell (bash)
4. C shell (csh)

Bourne shell

- most popular Unix shell
- default prompt is $

Korn shell

- backward compatible with Bourne shell
- includes many features of C shell.

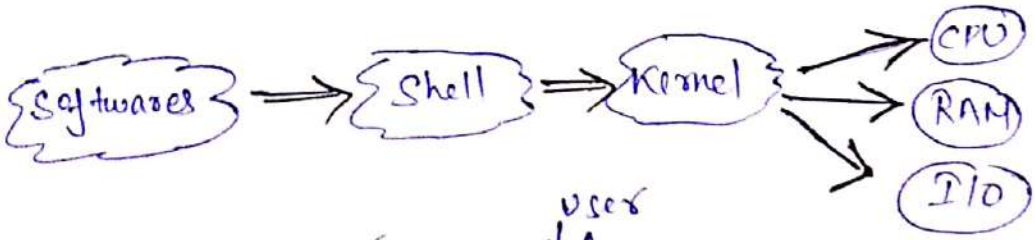
Bourne Again Shell

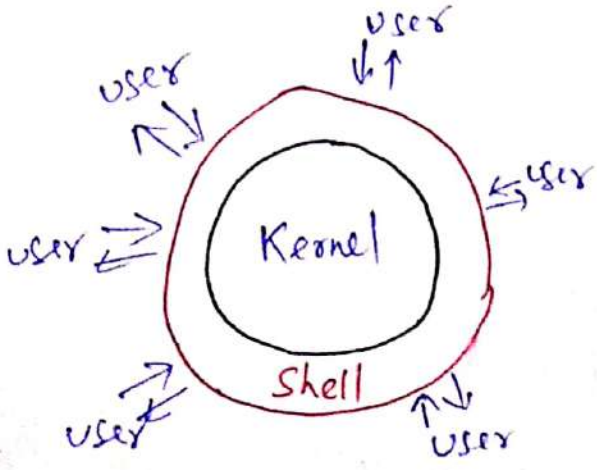
- Free shell replacement for the Bourne shell.

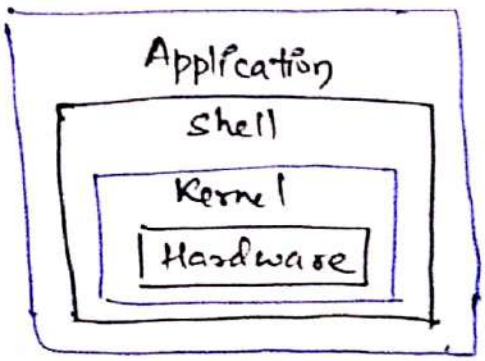
C shell

- provides history features and aliasing of commands.
- default prompt is %.

Comparison	Kernel	Shell
Definition	A computer program which acts as the core of the operating system and has control over everything in the system.	A computer program which works as the interface to access the services provided by operating system.
Usage	Kernel is the core of the system that controls all the tasks of the system.	Shell is the interface between the Kernel and user.
Types	Kernel does not have types.	Shell has different types such as Bourne shell, C shell, Korn shell, Bourne Again shell





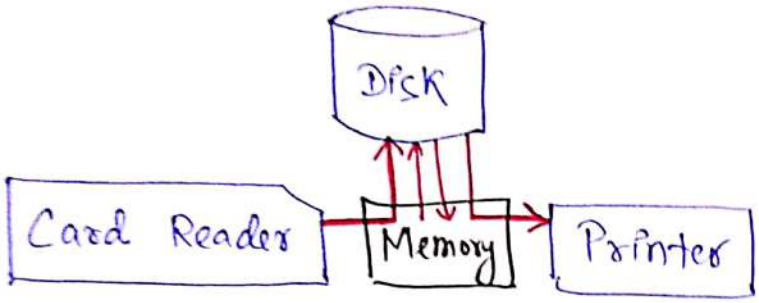


Interactive Operating System

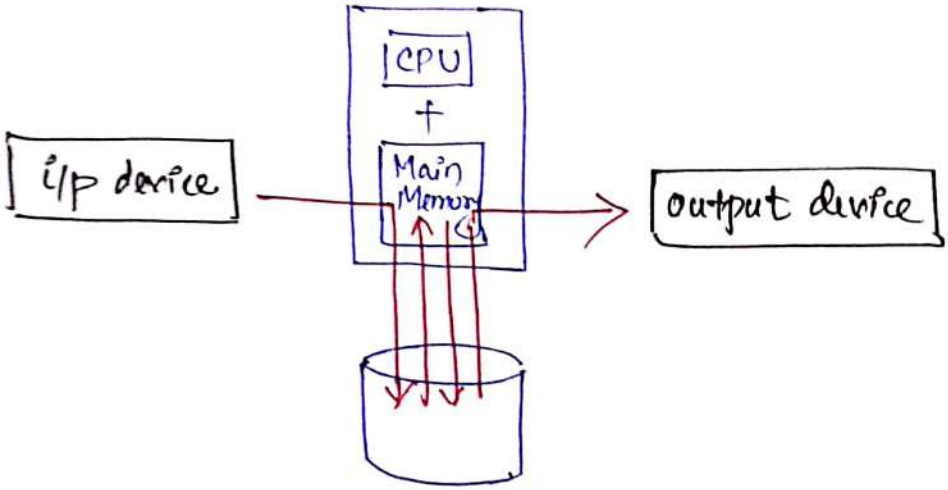
- In an interactive operating system, the user interacts directly with the operating system by typing commands.
- The interaction can be
  (1) Graphical User Interface (GUI)
  (2) Command Line Interface (CLI).
- Interactive media plays an important role in designing interface. Interactive media include text, animation, video, animated image and buttons, video games etc.

Spooling

Spooling stands for simultaneous peripheral operations on line.



(Or)



- Input and output are relatively slow compare to CPU (digital).
- In Spooling, data is stored first onto the disk and then CPU interact with disk (digital) via Main Memory.
- Spooling is capable of overlapping I/O operations for one job with CPU operations of other jobs.

Advantages

(1) No interaction of Input and output device with CPU.
(2) CPU utilization is more as CPU is busy most of the time.

Disadvantages

In Earlier times, spooling was uniprogramming.

Note:→ Dispatch Latency

- The time taken by dispatcher to stop one process and start running another process is called dispatch latency.
- Dispatch latency describes the amount of time it takes for a system to respond to a request for a process to begin operation.