15th July '2024

Sahil Biswas
16010422013

1. Overview of Operating systems

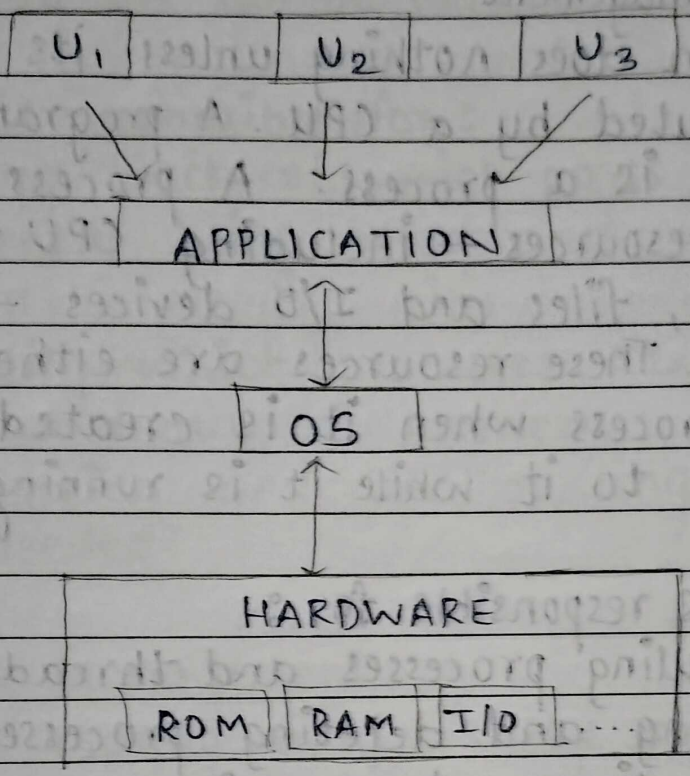


Fig: Abstract view of the components of a computer system.

An operating system is a program that manages a computer's hardware. It also provides a basis for application programs and acts as an intermediary between the computer user and the computer hardware.

The hardware - the central processing unit (CPU), the memory, and the input/output (I/O) devices - provides the basic computing resources for the system.

The operating system controls the hardware and coordinates its use among the various application programs for the various users.

Teacher's Signature:.....

Need for Operating Systems

Process Management

A program does nothing unless its instructions are executed by a CPU. A program in execution is a process. A process needs certain resources — including CPU time, memory, files and I/O devices — to accomplish its task. These resources are either given to the process when it is created or allocated to it while it is running.

The OS is responsible for :-

- Scheduling processes and threads on CPUs
- Creating and deleting processes
- Suspending and resuming processes
- Providing mechanisms for synchronization.
- Providing mechanisms for communication.

Memory Management

Main memory is a large array of bytes with each byte having its own address. Main memory is the repository of quickly accessible data shared by the CPU and the I/O devices. For a program to be executed, it must be mapped to absolute addresses and loaded into memory. As the program executes, it accesses program instructions and data from memory by generating these absolute addresses.

Eventually, the program terminates, its memory space is declared available, and the next program can be loaded and executed.

The OS is responsible for :-
- Keeping track of which parts of memory are currently being used and who is using them
- Deciding which processes and data move into and out of memory.
- Allocating and deallocating memory space as needed

3. Resource Management
Resource management is the process of allocating, controlling and optimizing the use of computer resources. It is important because computer resources are limited, and multiple processors or users may need access to the same resources at the same time.

The OS is responsible for :-
- Resource scheduling
- Resource Monitoring
- Resource Protection
- Resource Sharing
- Resource Accounting
- Performance optimization

Teacher's Signature:.....

4. Storage Management

(a) File system management

File management is one of the most visible components of an operating system. A file is a collection of related information defined by its creator. The operating system implements the abstract concept of a file by managing mass storage media. Finally when multiple users have access to files, it may be desirable to control which user may access a file and how the user may access it.

The OS is responsible for :-
- Creating and deleting files
- Creating and deleting directories
- Supporting primitives for manipulating files
- Mapping files onto secondary storage
- Backing up files on nonvolatile storage

(b) Mass-storage management

Main memory is too small to accomodate all data and programs and because the data that it holds are lost when power is lost, the computer system must provide secondary storage. Hence, proper management of disk storage is of central importance to a computer system.

The OS is responsible for :-
- Free space management
- Storage allocation
- Disk scheduling

Teacher's Signature:

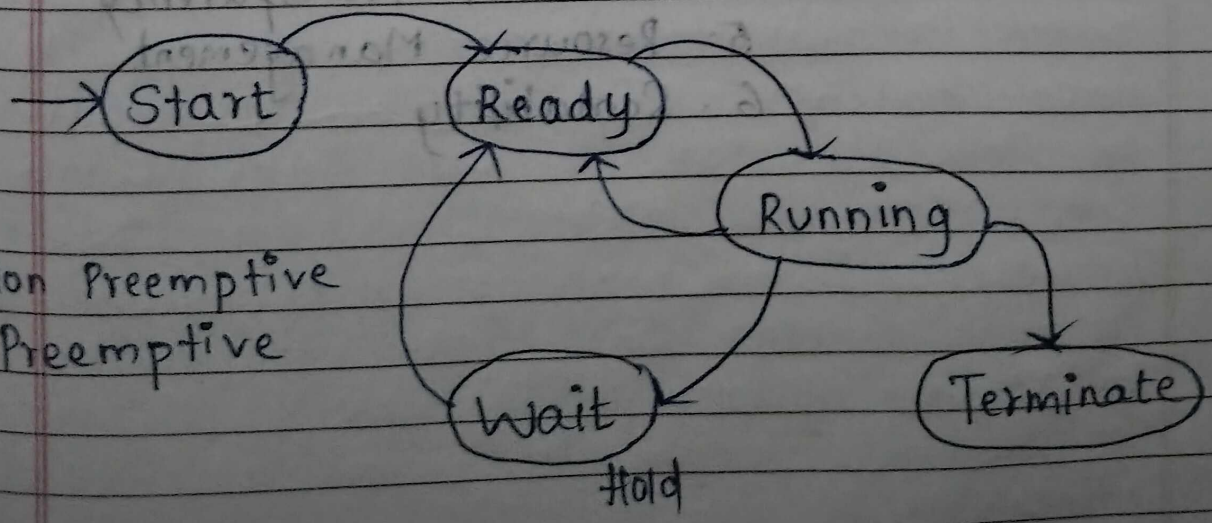
5. Security and Protection

If a computer system has multiple users and allows the concurrent execution of multiple processes, then access to data must be regulated. For that purpose, mechanisms ensure that files, memory segments, CPU and other resources can be operated on by only those processes that have gained proper authorization. Protection is any mechanism for controlling the access of processes or users to the resources defined by a computer system. It is the job of security to defend a system from external and internal attacks. Protection and security require the system to be able to distinguish among all its users.

19th July '2024

- Process

STATES

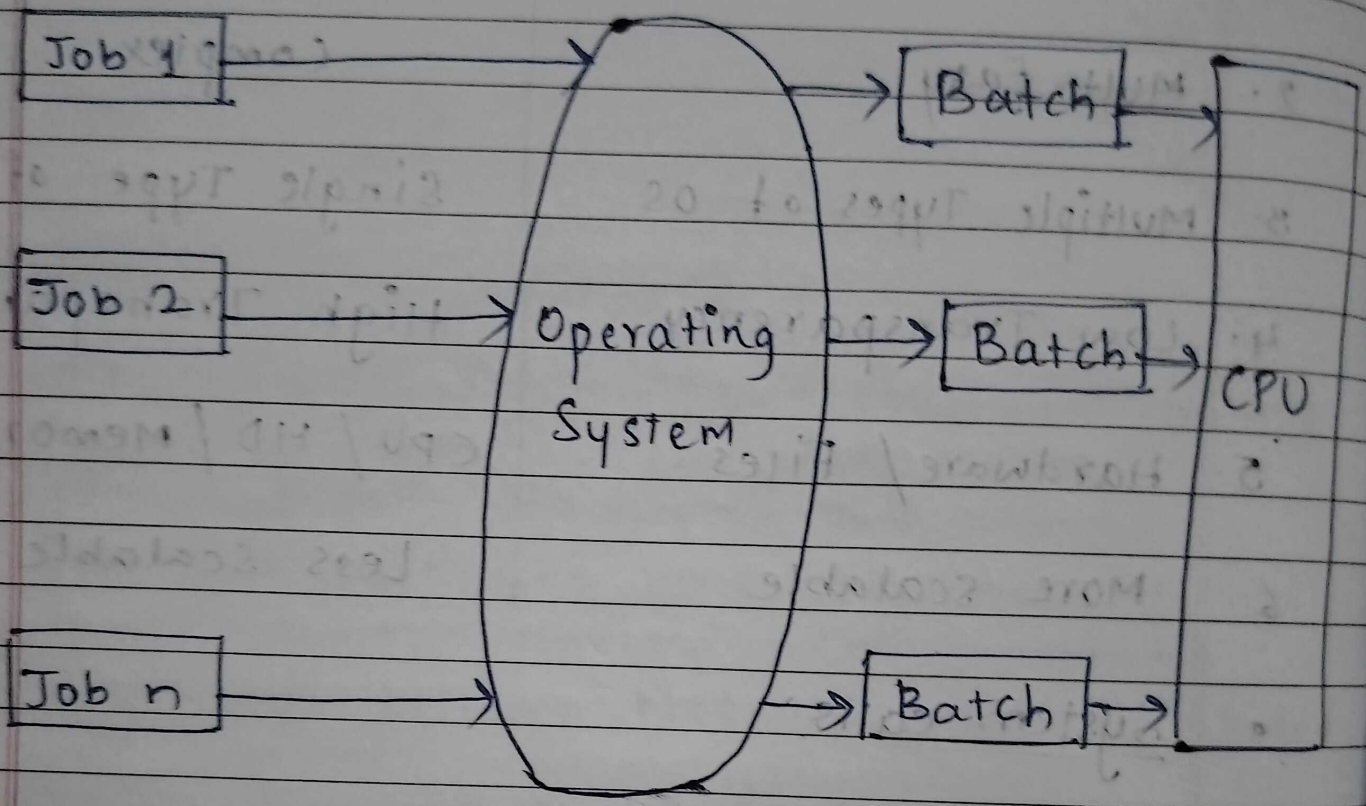


1. Non Preemptive
2. Preemptive

Teacher's Signature:.....

- Types of OS

1) Batch OS



Batch OS does not interact with the computer directly. There is an operator which takes similar jobs having the same requirements and groups them into batches. It is the responsibility of the operator to sort jobs with similar needs. Batch operating system is designed to manage and execute a large number of jobs efficiently by processing them in groups

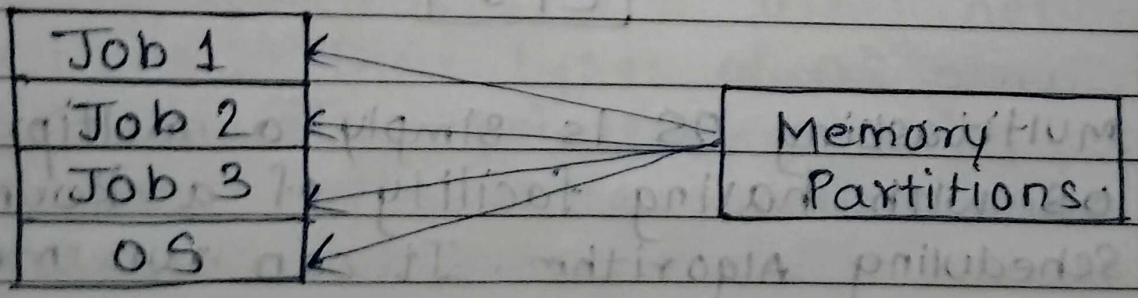
- Advantages

1. Multiple users can share the batch systems.
2. The idle time for the batch system is very less.
3. It is easy to manage large work repeatedly in batch systems.

- Disadvantages

1. Batch systems are hard to debug.
2. It is sometimes costly.
3. The other jobs will have to wait for an unknown time if any job fails.

2) Multi-programming OS



- ~~Adva~~ Multiprogramming OS can be simply illustrated as more than one program is present in the main memory and any one of them can kept in execution. This is basically used for better execution of resources.

- Advantages

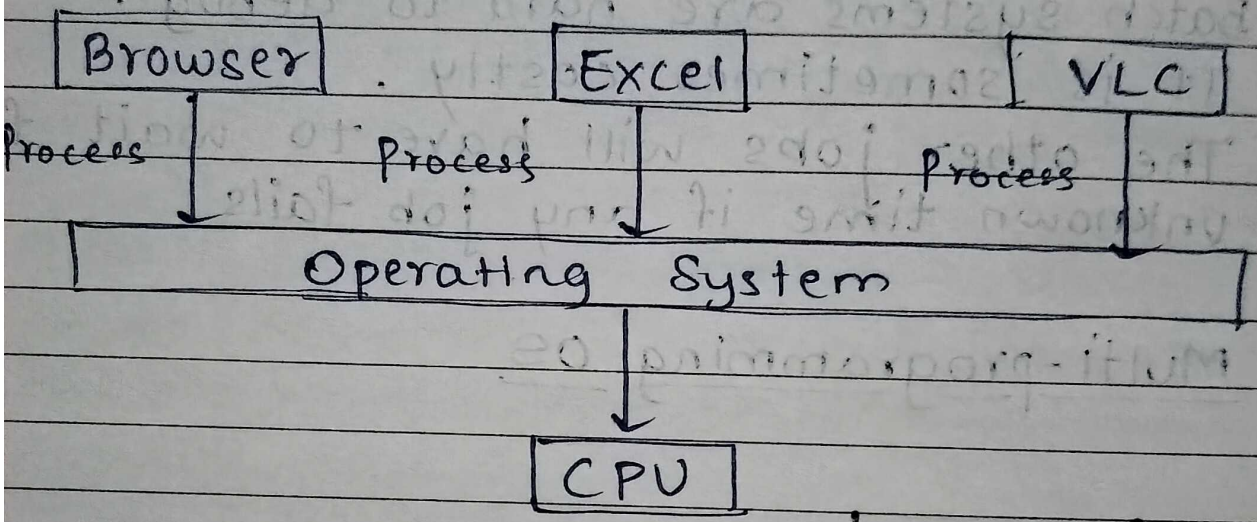
1. Multiprogramming increases the throughput of the system.

Teacher's Signature:.....

It helps in reducing the response time.

Disadvantage - There is no facility for user interaction of system resources with the system.

Multi-tasking OS



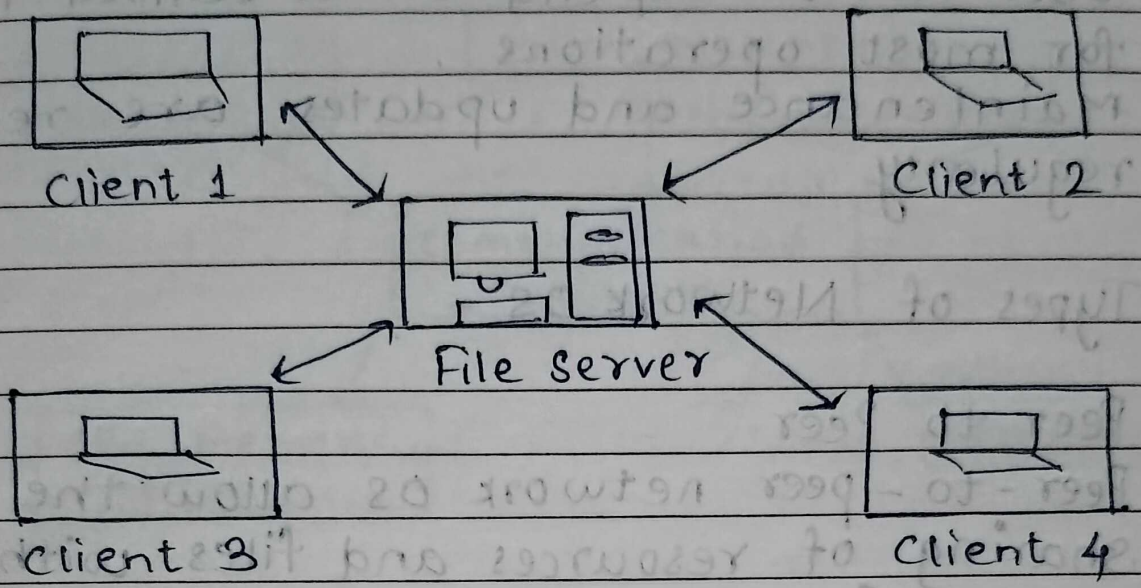
Multitasking OS is simply a multiprogramming OS with having facility of a Round-Robin Scheduling Algorithm. It can run multiple programs simultaneously.

Advantages -
Multiple Programs can be executed simultaneously in Multi-Tasking OS.
It comes with proper memory management

Disadvantage - The system gets heated in case of heavy programs multiple times.

Teacher's Signature:

4) Network OS



These systems run on a server and provide the capability to manage data, users, groups, security applications and other networking functions. These types of OS allow shared access to files, printers, security, applications and other networking functions over a small private network. One more important aspect of Network OS is that all the users are well aware of the underlying configuration of all other users within the network, their individual connections, etc.

- Advantages

1. Highly stable centralized servers.
2. Security concerns are handled through servers.
3. New technologies and hardware upgrade are easily integrated into the system.

Teacher's Signature:.....

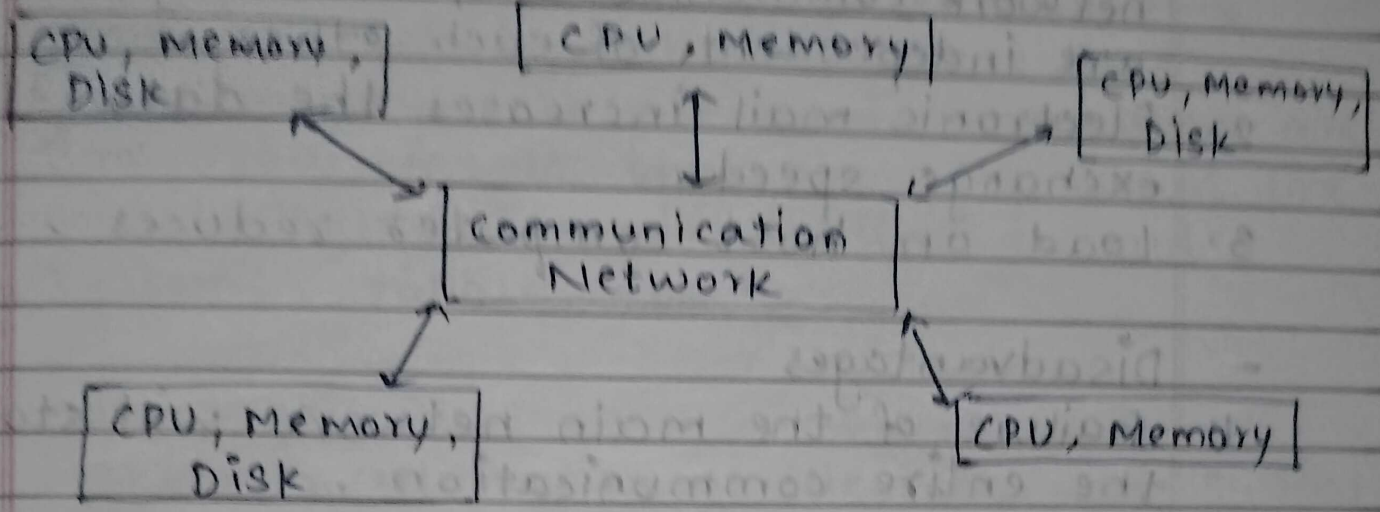
- Disadvantages
1. Servers are costly.
2. User has to depend on a central location for most operations
3. Maintenance and updates are required regularly.

5) Types of Network OS :-

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

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

5) Distributed OS



These types of OS is a recent advancement in the world of computer technology and are being widely accepted all over the world. Various autonomous interconnected computers communicate with each other using a shared communication network. Independent systems possess their own memory unit and CPU. These systems' processors differ in size and function. The major benefit of working with these types 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.

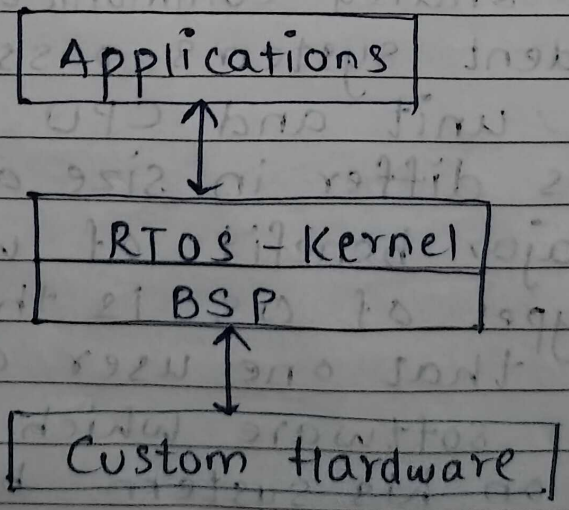
- Advantages

1. Failure of one will not affect the other network communication, as all systems are independent of each other.
2. Electronic mail increases the data exchange speed.
3. Load on host computer reduces.

- Disadvantages

1. Failure of the main network will stop the entire communication.
2. To establish distributed systems, the language is not well-defined yet.
3. These types of systems are not readily available as they are very expensive.

6) Real Time OS



These types of OS serve real-time systems. The time interval required to process and respond to inputs is very small. This time interval is called response time. Real-time systems are used when there are time requirements that are very strict like missile systems, air traffic control systems, robots, etc.

- Advantages

1. Maximum utilization of devices and systems, thus more output from all the resources.
2. These types of systems are error-free.
3. Memory allocation is best managed in these types of systems.

- Disadvantages

1. The algorithms are very complex and difficult for the designer to write on.
2. It needs specific device drivers and interrupts signal to respond earliest to interrupts.
3. It is not good to set thread priority as these systems are very less prone to switching tasks.

P.T.O

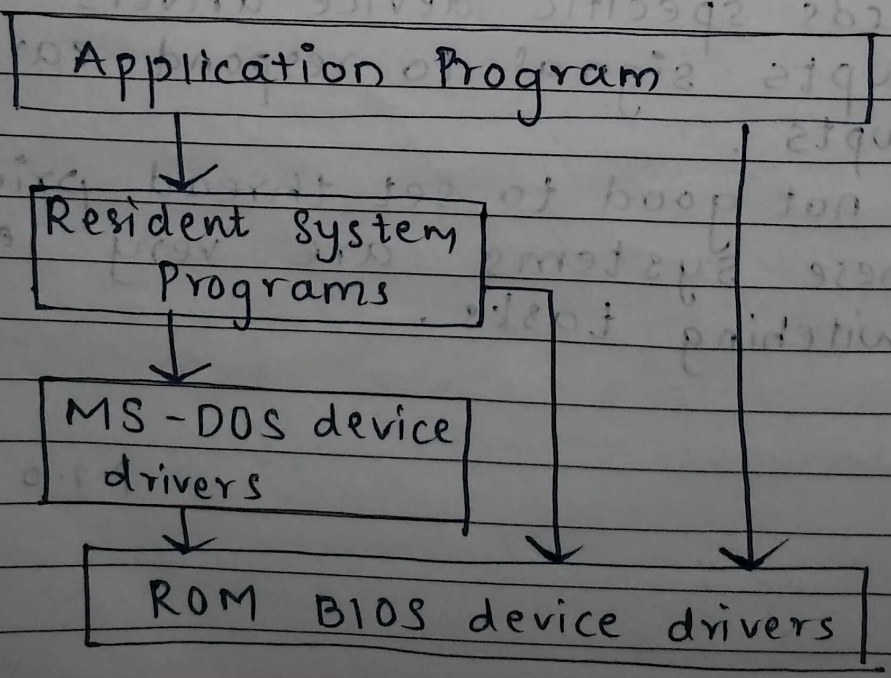
Types of Real Time OS

1. Hard RTOS
Hard RTOS are meant for applications where time constraints are very strict and even the shortest possible delay is not acceptable. These systems are built for saving life like airbags which are readily available in case of an accident. Virtual memory is rarely found in these systems.

2. Soft RTOS
These OS are for applications where time-constraint is less constraint.

Structures of OS

1) Monolithic Structure



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

- Advantages

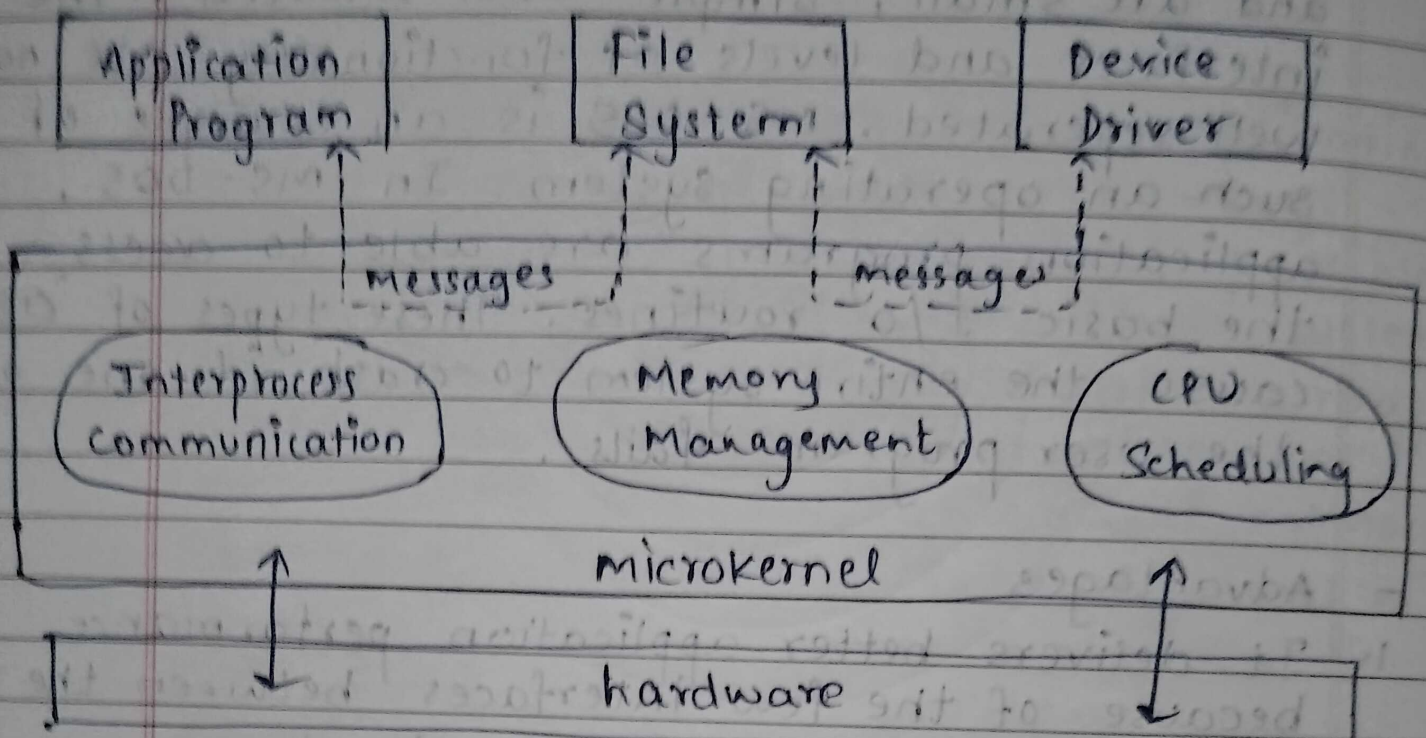
1. It delivers better application performance because of the few interfaces between the application program and the hardware.
2. It is easy for kernel developers to develop such an OS.

- Disadvantages

1. The structure is very complicated, as no clear boundaries exist between modules.
2. It does not enforce data hiding in the OS.

P.T.O

2) Microkernel structure



This structure designs the OS by removing all non-essential components from the kernel and implementing them as system and user programs. This results in a smaller kernel called the microkernel. Advantages of this structure are that all new services need to be added to user space and does not require the kernel to be modified. Thus it is more ~~sec~~ secure and reliable as if a service fails, then rest of the OS remains untouched.

→

Advantages -

1. It makes the operating system portable to various platforms.
2. As microkernels are small, so these can be tested effectively.

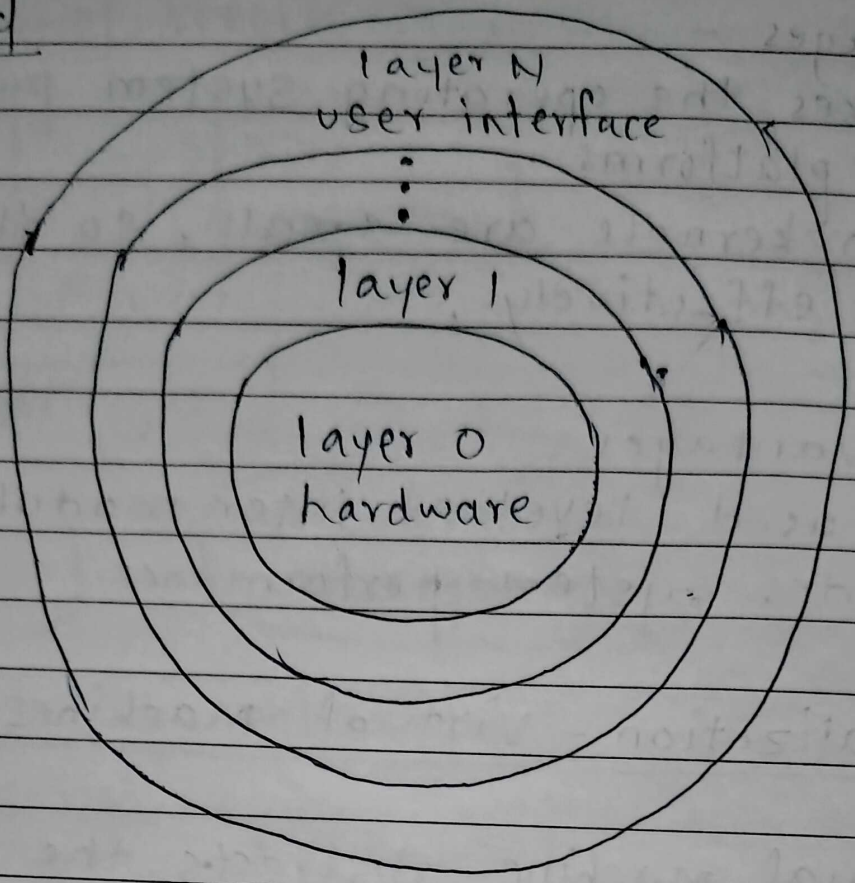
Disadvantages

1. Increased level of inter module communication degrades system performance.

3) Virtualization - Virtual Machines.

A virtual machine abstracts the hardware of our personal computer, including the CPU, disc drives, RAM, and NIC into a variety of different execution contexts, giving us the impression that each execution environment is a different computer. An illustration of it is a virtual box. An OS enables us to run multiple processes concurrently while making it appear as though each one is using a different processor and virtual memory by using CPU scheduling and virtual memory techniques. The program that creates virtual machines would need a significant amount of disc space in order to provide virtual memory & spooling. This is the fundamental issue with these structures.

4) Layered



An OS can be broken into pieces and retain much more control over the system. In this structure, the OS is broken into a number of layers (levels). The bottom layer is the hardware, and the topmost layer is the UI. These layers are so designed that each layer uses the functions of the lower-level layers. This simplifies the debugging process, if lower-level layers are debugged and an error occurs during debugging, then the error must be on that layer only.

Example: UNIX.

- Modern UNIX systems.

UNIX is an OS that is the base of all OS like Ubuntu, Solaris, POSIX, etc. UNIX is a family of multitasking, multiuser computer OS. It has a reputation for stability, security and scalability. The basic design philosophy of UNIX is to provide simple, powerful tools that can be combined to perform complex tasks.

Features:-

1. Multiuser Support: UNIX allows multiple users to simultaneously access the same system and share resources.

2. Multitasking: UNIX is capable of running multiple processes at the same time.

3. Shell Scripting: UNIX provides a powerful scripting language that allows users to automate tasks.

4. Security: UNIX has a robust security model that includes file permissions, user accounts, and network security features.

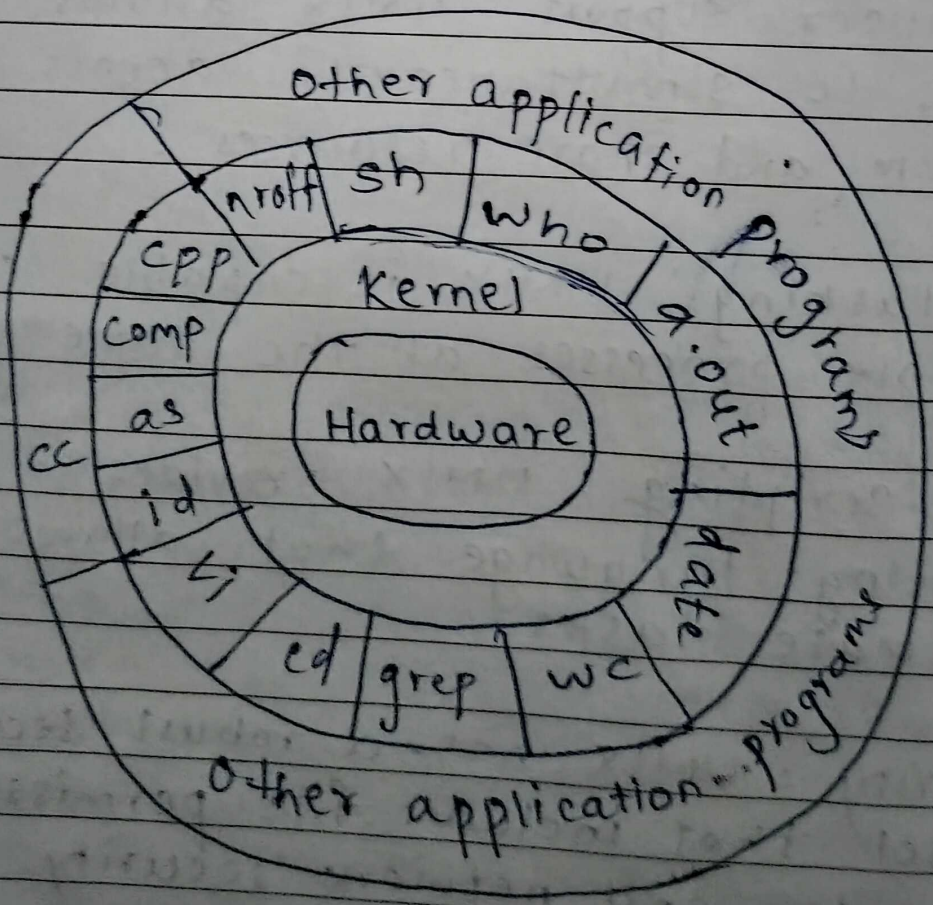
5. Portability: UNIX can run on a wide variety of hardware platforms, from small embedded systems to large mainframe computers.

Teacher's Signature:.....

6. Communication: UNIX supports communication methods using write command, mail command, etc.

7. Process Tracking: UNIX maintains a record of the jobs that the user creates. This function improves system performance by monitoring CPU usage. It allows you to keep track of how much disk space each user uses, and they use that information to regulate disk space.

Structure of UNIX:



Layer - 1 : Hardware : It consists of all hardware related information.

Layer - 2 : Kernel : This is the core of the OS. It is a software that acts as the interface between the hardware and the software. Most of the tasks like memory management, network management, process management, etc are done by the kernel.

Layer - 3 : Shell Commands : This is the interface between the user and the kernel. Shell is the utility that processes your requests. When you type in a command at the terminal, the shell interprets the command and calls the program that you want.

Layer - 4 : Application Layer : It is the outermost layer that executes the given external applications.

Goals of OS

1. Minimize idle time of system resources. Idle time refers to periods when system resources are not being utilized effectively. OS aims to keep the CPU and other resources as busy as possible to maximize efficiency. Techniques such as multitasking and scheduling are used to ensure this.

2. Minimize response time: Response Time is the time to receive first response when the request is made. The OS achieves this by prioritizing tasks that require user interaction and using preemptive scheduling to quickly respond to user inputs.

3. Maximize throughput, which is the number of processes completed in a given time frame. The OS aims to maximize throughput by efficiently managing processes balancing loads, and optimizing resource usage. Techniques such as batch processing, efficient process scheduling and load balancing are used to achieve high throughput.

System call

A system call is a mechanism used by programs to request services from the OS.

A user program can interact with the OS using a system call. A number of services are requested by the program, and the OS responds by launching a number of system calls to fulfill the request.

A system call is initiated by the program executing a specific instruction, which triggers a switch to kernel mode, allowing the program to request a service from the OS. The OS then handles the request, performs the necessary operations and returns the result back to the program.

System calls are essential for the proper functioning of an OS, as they provide a standardized way for programs to access system resources. Without system calls, each program would need to implement its methods for accessing hardware and system services, leading to inconsistent and error-prone behavior.

Teacher's Signature:.....

- Types of System Calls

Type	Functions	Windows Example	Unix Example
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.	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
2) File Management	create file, delete file, open, close, read, write, reposition, get file attributes, set file attributes.	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
3) Device Management	request device, release device, read, write, reposition, get device attributes, set device attributes, logically attach or detach devices.	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()

Type	Functions	Windows Example	UNIX Example
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	Get Current Process ID() Set Timer() Sleep()	getpid() alarm() sleep()
5) Communication	create, delete communication connection, send, receive messages, transfer status information, attach or detach remote devices	CreatePipe() CreateFile Mapping() MapViewOfFile()	pipe() shm_open() mmap()

P.T.O

Network OS v/s Distributed OS

	Network OS	Distributed OS
1.	Network OS's main objective is to provide the local services to remote client.	Distributed OS's main objective is to manage the hardware resources.
2.	In Network OS, communication takes place on the basis of files.	In Distributed OS, communication takes place on the basis of messages and shared memory.
3.	Network OS is more scalable than Distributed OS.	Distributed OS is less scalable than Network OS.
4.	In Network OS, fault tolerance is less.	While in Distributed OS, fault tolerance is high.
5.	Rate of autonomy in Network OS is high.	while the rate of autonomy is distributed OS is less.

6.	Ease of implementation in Network OS is also high.	while in distributed OS, ease of implementation is less.
7.	In Network OS, all nodes can have different OS.	while in distributed OS, all nodes have same OS.