NUTAN MAHARASHTRA VIDYA PRASARAK MANDAL'S

NUTAN COLLEGE OF ENGINEERING & RESEARCH (NCER)

Department of Computer Science & Engineering

BTCOC402

Operating System

Teaching Notes

Lecture Number	Topic to be covered (Unit 1)
	Introduction (7 Hrs)
1	**Introduction** Introduction and Operating system structures: Definition,
2	Types of Operating system, Real-Time operating system, System Components: System Services,
3	Systems Calls, System Programs, System structure
4	Systems Calls, System Programs, System structure
5	System Design and Implementation, System Generations

: Submitted by:

Prof. A. S. Aher

Nutan College Of Engineering & Research
Talegaon Dabhade, Pune- 410507

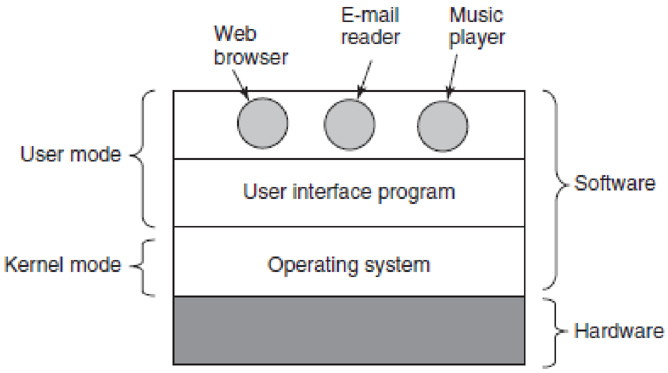
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Operating System

Unit 1:- Introduction

Introduction:

- A modern computer consists of one or more processors, some main memory, disks, printers, a keyboard, a mouse, a display, network interfaces, and various other input/output devices.
- All in all, a complex system. If every application programmer had to understand how all these things work in detail, no code would ever get written.
- Furthermore, managing all these components and using them optimally is an exceedingly challenging job.
- For this reason, computers are equipped with a layer of software called the operating system, whose job is to provide user programs with a better, simpler, cleaner, model of the computer and to handle managing all the resources just mentioned.
- The program that users interact with usually called the shell when it is text-based and the GUI (Graphical User Interface) when it uses icons, is actually not part of the operating system, although it uses the operating system to get its work done.



- A simple overview of the main components under discussion here is given in Fig. 1-1. Here we see the hardware at the bottom.
- The hardware consists of chips, boards, disks, a keyboard, a monitor, and similar physical objects.
- On top of the hardware is the software. Most computers have two modes of operation: kernel mode and user mode.
- The operating system, the most fundamental piece of software, runs in kernel mode (also called supervisor mode).
- In this mode, it has complete access to all the hardware and can execute any instruction the machine is capable of executing.
- The rest of the software runs in user mode, in which only a subset of the machine instructions is available. In particular, those instructions that affect control of the machine or do I/O) Input/Output" are forbidden to user-mode programs.
- The user interface program, shell or GUI, is the lowest level of user-mode software and allows the user to start other programs, such as a Web browser, email reader, or music player.
- These programs, too, make heavy use of the operating system.
- The placement of the operating system is shown in the Figure mentioned above.
- It runs on bare hardware and provides the base for all the other software.
- An important distinction between the operating system and normal (user mode) software is that if a user does not like a particular email reader, he is

free to get a different one or write his own if he so chooses; he is not free to write his own clock interrupt handler, which is part of the operating system and is protected by hardware against attempts by users to modify it.

- This distinction, however, is sometimes blurred in embedded systems (which may not have kernel mode) or interpreted systems (such as Java-based systems that use interpretation, not hardware, to separate the components).
- An operating system acts as an intermediary between the user of a computer and computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs conveniently and efficiently.
- An operating system is a software that manages computer hardware. The hardware must provide appropriate mechanisms to ensure the correct operation of the computer system and to prevent user programs from interfering with the proper operation of the system.
- An operating system is a program that controls the execution of application programs and acts as an interface between the user of a computer and the computer hardware.
- A more common definition is that the operating system is the one program running at all times on the computer (usually called the kernel), with all else being application programs.
- An operating system is concerned with the allocation of resources and services, such as memory, processors, devices, and information. The operating system correspondingly includes programs to manage these resources, such as a traffic controller, a scheduler, a memory management module, I/O programs, and a file system.
- OS is designed to serve two basic purposes:
- It controls the allocation and use of the computing System’s resources among the various user and tasks.

- It provides an interface between the computer hardware and the programmer that simplifies and makes it feasible for coding, creation, debugging of application programs.
- The Operating system must support the following tasks. The tasks are:
- Provides the facilities to create, modification of programs and data files using an editor.
- Access to the compiler for translating the user program from high-level language to machine language.
- Provide a loader program to move the compiled program code to the computer's memory for execution.
- Provide routines that handle the details of I/O programming

Introduction: What is an OS?

- An operating system acts as an intermediary between the user of a computer and computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs conveniently and efficiently.
- An operating system is a program that controls the execution of application programs and acts as an interface between the user of a computer and the computer hardware.
- A more common definition is that the operating system is the one program running at all times on the computer (usually called the kernel), with all else being application programs.
- An operating system is concerned with the allocation of resources and services, such as memory, processors, devices, and information. The operating system correspondingly includes programs to manage these resources, such as a traffic controller, a scheduler, a memory management module, I/O programs, and a file system.
- A computer system can be divided into four major components:

1. Hardware.
2. Operating System.
3. Application Programs.
4. Users.

- Every general-purpose computer consists of the hardware, operating system, system programs, and application programs. The hardware consists of memory, CPU, ALU, I/O devices, peripheral devices, and storage devices. System program consists of compilers, loaders, editors, OS, etc. The application program consists of business programs, database programs.

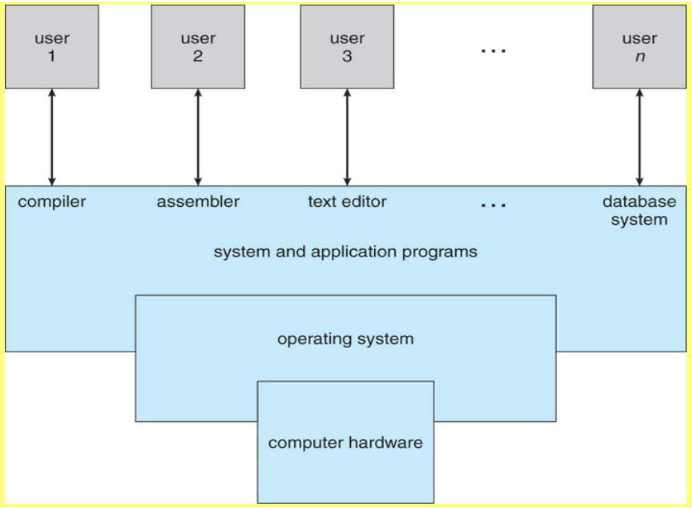


Fig: Conceptual View of an OS

- The hardware - the central processing unit (CPU), the memory and the i/o devices - provides the basic computing resources for the system.
- The application programs such as word processors/ spreadsheets/ compilers, and Web browsers define the ways in which these resources are used to solve users' computing problems. The operating system controls the hardware and coordinates its use among the various application programs for the various users.
- To understand more fully the operating system’s role, we next explore operating systems from two viewpoints: that of the user and that of the system.

1. User view

- The user's view of the computer varies according to the interface being used. Most computer users sit in front of a PC, consisting of a monitor/ keyboard/ mouse, and a system unit. Such a system is designed for one user to monopolize its resources.
- The goal is to maximize the work (or play) that the user is performing.
- In this case/ the operating system is designed mostly for ease of use, with some attention paid to performance and none paid to resource utilization – how various hardware and software resources are shared.
- Performance is, of course, important to the user; but such systems are optimized for the single-user experience rather than the requirements of multiple users.
- In other cases, a user sits at a terminal connected to a mainframe or a minicomputer.
- Other users are accessing the same computer through other terminals. These users share resources and may exchange information.
- The operating system in such cases is designed to maximize resource utilization – to assure that all available CPU time, memory, and I/0 are used efficiently and that no individual user takes more than her fair share.
- In still other cases, users sit at workstations connected to networks of other workstations and servers.
- These users have dedicated resources at their disposal, but they also share resources such as networking and servers-file compute, and print servers.
- Therefore, their operating system is designed to compromise individual usability and resource utilization.
- Recently, many varieties of handheld computers have come into fashion.
- Most of these devices are standalone units for individual users.
- Some are connected to networks, either directly by wire or (more often) through wireless modems and networking.

- Because of power, speed, and interface limitations, they perform relatively few remote operations.
- Their operating systems are designed mostly for individual usability, but performance per unit of battery life is important as well.
- Some computers have little or no user view. For example, embedded computers in-home devices and automobiles may have numeric keypads and may turn indicator lights on or off to show status, but they and their operating systems are designed primarily to run without user intervention.

2. System View

- From the computer's point of view, the operating system is the program
- most intimately involved with the hardware.
- In this context, we can view an operating system as a resource allocator.
- A computer system has many resources that may be required to solve a problem: CPU time, memory space, file-storage space, I/0 devices, and so on.
- The operating system acts as the manager of these resources. Facing numerous and possibly conflicting requests for resources, the operating system must decide how to allocate them to specific programs and users so that it can operate the computer system efficiently and fairly.
- As we have seen, resource allocation is especially important where many users access the same mainframe or minicomputer.
- A slightly different view of an operating system emphasizes the need to control the various I/0 devices and user programs.
- An operating system is a control program. A control program manages the execution of user programs to prevent errors and improper use of the computer. It is especially concerned with the operation and control of I/O devices.

Operating Systems Structure:

- One of the most important aspects of operating systems is the ability to multiprogramming.
- A single program cannot, in general, keep either the CPU or the I/O devices busy at all times.
- Single users frequently have multiple programs running. Multiprogramming increases CPU utilization by organizing jobs (code and data) so that the CPU always has one to execute.

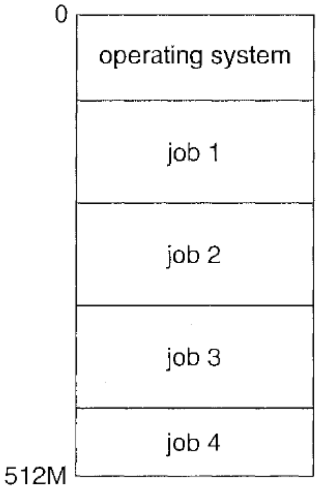


Fig: Memory layout for multiprogramming system

- The OS always have several jobs in memory simultaneously. This set of jobs can be a subset of jobs kept in the job pool which contains all jobs that enter into the system since the number of jobs that can be kept simultaneously in memory is usually smaller than the number of jobs that can be kept in the job pool.
- The OS picks and begins to execute one of the jobs in memory. In a non-multiprogramming system, the CPU would sit idle. In a multiprogramming system, OS simply switches to and executes another job. When that job needs to wait, the CPU switches to another job and so on.
- Eventually, the first job finishes waiting and gets the CPU back. As long as at least one job needs to execute, the CPU is never idle.

Objectives of Operating System

- The objectives of the operating system are –
- To make the computer system convenient to use in an efficient manner.
- To hide the details of the hardware resources from the users.
- To provide users a convenient interface to use the computer system.
- To act as an intermediary between the hardware and its users, making it easier for the users to access and use other resources.
- To manage the resources of a computer system.
- To keep track of who is using which resource, granting resource requests, and mediating conflicting requests from different programs and users.
- To provide efficient and fair sharing of resources among users and programs.

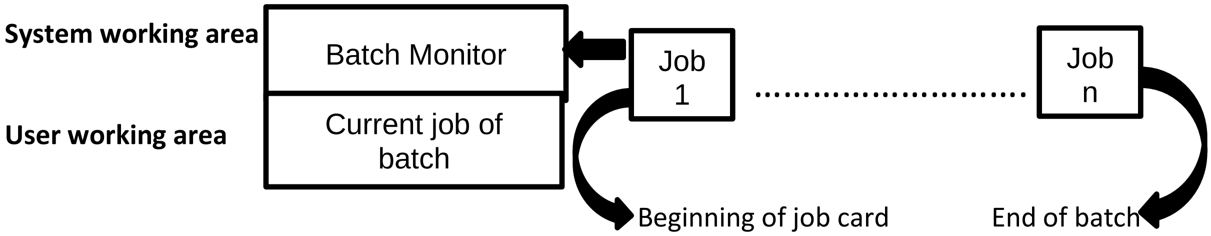
Types of Operating System:

- An Operating System performs all the basic tasks like managing files, processes, and memory. Thus operating system acts as the manager of all the resources, i.e. **resource manager**. Thus, the operating system becomes an interface between user and machine.
- Some widely used operating systems are as follows-

1. Batch Processing System

- In the early days' punch cards was used to record the user's job. But this was a manual and tedious job which was done by a human operator.
- To avoid the continuous human operators' devotion, the concept of batch processing grew up.
- To avoid the problems of early systems batch processing systems were introduced. The problem of early systems was more setup time. So the problem of more set-up time was reduced by processing the jobs in batches, known as batch processing system. In this approach, similar jobs were submitted to the CPU for processing and were run together.

- The system functioned by submitting jobs to the batch processing system (BPS) and the jobs/batches got executed automatically.
- The main function of a batch processing system is to automatically keep executing the jobs in a batch. This is the important task of a batch processing system i.e. performed by the 'Batch Monitor' residing in the low end of main memory.
- This technique was possible due to the invention of hard-disk drives and card readers.
- The jobs could be stored on the disk to create the pool of jobs for its execution as a batch. First, the pooled jobs are read and executed by the batch monitor, and then these jobs are grouped; placing the identical jobs (jobs with similar needs) in the same batch.
- So, in the batch processing system, the batched jobs were executed automatically one after another saving its time by performing the activities (like loading of a compiler) only once. It resulted in improved system utilization due to reduced turnaround time.
- In the batch processing system, earlier; the jobs were scheduled in the order of their arrival i.e. First Come First Served (FCFS).
- After execution of a job, it was sent to that particular user who owes it. Each job was an independent job in a batch.
- This was achieved by automating the transition from the execution of the job to that of the next job of the same batch.
- Batch monitor or supervisor was the major component which permanently placed in the part of the memory.
- Another part of memory is used to process the user's job. i.e. a current job in the batch.



- The left side of the diagram is a memory map in which half portion is occupied by batch monitor known as system area and the remaining memory contains the current job of the batch for execution.
- The major parts of the batch monitor were loader and control interpreter card which does the loading of jobs and control input and output respectively.

Batch Monitors Functions:

- The batch monitor keeps control of the processing environment.
- These functions can be categorized into three main tasks:
  1. Scheduling
  2. Memory Management
  3. Sharing and protections.

- The first two functions were done before the execution of the program and the third one was done after the job was executed.

1. Scheduling.

- Scheduling is the activity of determining which service request should be handled next by a server. Here service request means the user jobs and server is nothing but CPU.
- Batch processing uses FCFS – first come first serve scheduling criterion. i.e. the job which comes first is executed first i.e. as per the job sequence jobs are executed.

2. Memory Management.

- During batch processing operations, memory is divided into two parts called system area and user area. An issue of the partitioning of memory for the system area and user area is handled by an operating system area.

- Some part of the monitor i.e. some part of the code of the monitor is permanently stored in the memory. At the same time, other parts are loaded into the transient (temporary) area. This led to the system area as a residential area of monitor + transient area of the monitor.

3. Sharing and protection.

- These are the major function that OS provided for the purpose of a multi-user system. The sharing of the data can be done by choosing an appropriate operating system. This results in reducing the cost of hardware and processing time.
- During the job processing, the resources are allocated or de-allocated sequentially.
- The protection functions are more complicated than sharing.

In short Batch Systems:

- User does not interact with the systems
- Jobs with similar needs were grouped into Batches.
- Reads a stream of jobs operates on it and generated output.
- First come, first served i.e. job which comes first gets executed first.
- It can use **Spooling Technique**.

Advantages of Batch Operating System:

- It is very difficult to guess or know the time required for any job to complete. Processors of the batch systems know how long the job would be when it is in queue
- Multiple users can share the batch systems
- The idle time for the batch system is very less
- It is easy to manage large work repeatedly in batch systems

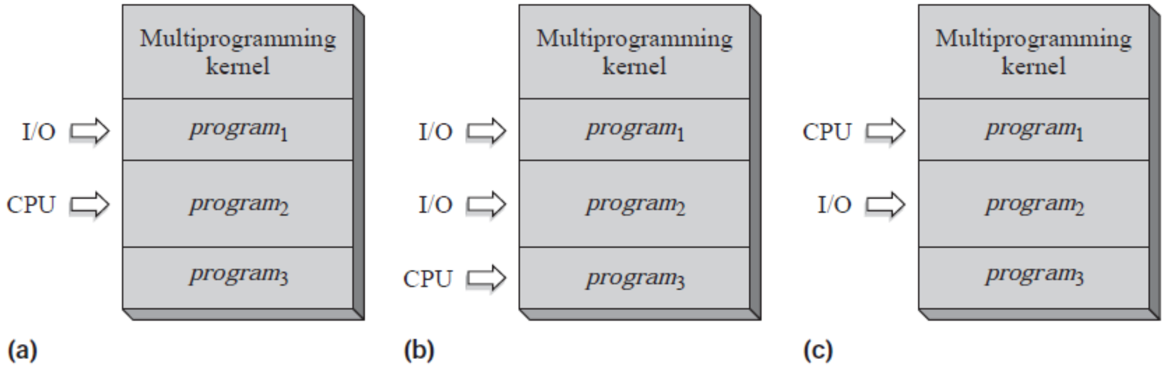
Disadvantages of Batch Operating System:

- The computer operators should be well known with batch systems

- Batch systems are hard to debug
- It is sometimes costly
- The other jobs will have to wait for an unknown time if any job fails
- Examples of Batch based Operating System: Payroll System, Bank Statements, etc.

2. Multiprogramming System

- Multiprogramming is the most important feature of an operating system.
- A single program cannot keep I/O devices as well as CPU busy. Using multiprogramming techniques, we can avoid CPU & I/O device idle time and maximum processing can be done.
- From the pool of jobs, operating system picks one job and begins executing it. While execution due to some reasons job may wait for some time e.g. may be waiting for storage or for any sort of input/output operation.
- In a non-multiprogramming system, the CPU would sit idle. In a multiprogramming system, the operating system simply switches to the next job, and executes, another job.
- When that job needs to wait, the CPU is switched to another job, and so on. Eventually, the first job finishes waiting and gets the CPU back. As long as at least one job needs to execute, the CPU is never idle.
- This way multiprogramming technique continuously keeps CPU and I/O devices in use.
- The other alternatives for keeping CPU and other devices continuously in use can use spooling and buffering techniques. But multiprogramming is more useful as compared to spooling and buffering.



- The figure illustrates the operation of a multiprogramming OS. The memory contains three programs. An I/O operation is in progress for program1, while the CPU is executing program2. The CPU is switched to program3 when program2 initiates an I/O operation, and it is switched to program1 when program1's I/O operation completes.
- The multiprogramming kernel performs scheduling, memory management and I/O management.
- It uses a simple scheduling policy and performs simply partitioned or pool-based allocation of memory and I/O devices.
- Since several programs are in memory at the same time, the instructions, data, and I/O operations of a program should be protected against interference by other programs.

<u>Advantages of multiprogramming systems</u>

- CPU is used most of the time and never become idle
- The system looks fast as all the tasks run in parallel
- Short time jobs are completed faster than long time jobs
- Multiprogramming systems support multiply users
- Resources are used nicely
- Total read time taken to execute program/job decreases
- Response time is shorter

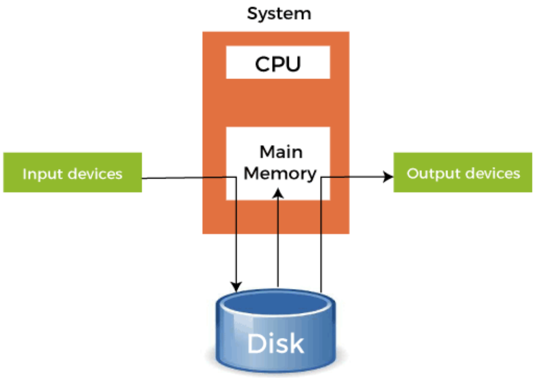
- In some applications multiple tasks are running and multiprogramming systems better handle these type of applications

<u>Disadvantages of multiprogramming systems</u>

- It is difficult to program a system because of complicated schedule handling
- Tracking all tasks/processes is sometimes difficult to handle
- Due to high load of tasks, long time jobs have to wait long

<u>What is Spooling?</u>

- Spooling is a process in which data is temporarily held to be used and executed by a device, program, or system. Data is sent to and stored in memory or other volatile storage until the program or computer requests it for execution.
- SPOOL is an acronym for Simultaneous Peripheral Operations Online. Generally, the spool is maintained on the computer's physical memory, buffers, or the I/O device-specific interrupts. The spool is processed in ascending order, working based on a FIFO (first-in, first-out) algorithm.
- In an operating system, spooling works in the following steps, such as:
- Spooling involves creating a buffer called SPOOL, which is used to hold off jobs and data till the device in which the SPOOL is created is ready to make use and execute that job or operate on the data.



- When a faster device sends data to a slower device to perform some operation, it uses any secondary memory attached as a SPOOL buffer. This data is kept in the SPOOL until the slower device is ready to operate on this data. When the slower device is ready, then the data in the SPOOL is loaded onto the main memory for the required operations.

- Spooling considers the entire secondary memory as a huge buffer that can store many jobs and data for many operations. The advantage of Spooling is that it can create a queue of jobs that execute in FIFO order to execute the jobs one by one.
- A device can connect to many input devices, which may require some operation on their data. So, all of these input devices may put their data onto the secondary memory (SPOOL), which can then be executed one by one by the device. This will make sure that the CPU is not idle at any time. So, we can say that Spooling is a combination of buffering and queuing.
- After the CPU generates some output, this output is first saved in the main memory. This output is transferred to the secondary memory from the main memory, and from there, the output is sent to the respective output devices.

What is Buffering?

- The buffer is an area in the main memory used to store or hold the data temporarily. In other words, buffer temporarily stores data transmitted from one place to another, either between two devices or an application. The act of storing data temporarily in the buffer is called buffering.
- A buffer may be used when moving data between processes within a computer. Buffers can be implemented in a fixed memory location in hardware or by using a virtual data buffer in software, pointing at a location in the physical memory. In all cases, the data in a data buffer are stored on a physical storage medium.
- Most buffers are implemented in software, which typically uses the faster RAM to store temporary data due to the much faster access time than hard disk drives. Buffers are typically used when there is a difference between

the rate of received data and the rate of processed data, for example, in a printer spooler or online video streaming.

- A buffer often adjusts timing by implementing a queue or FIFO algorithm in memory, simultaneously writing data into the queue at one rate and reading it at another rate.

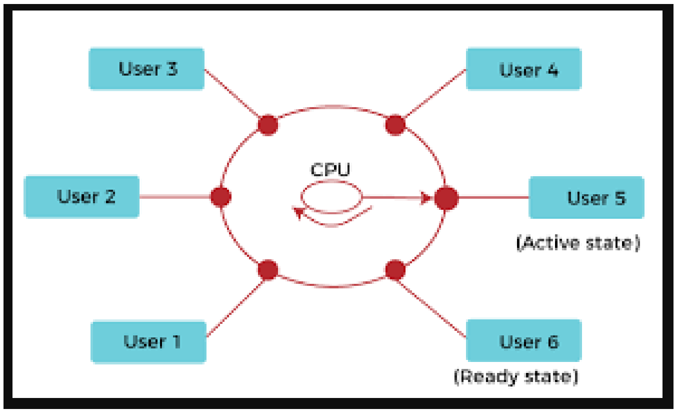
Spooling VS Buffering

Terms	Spooling	Buffering
Definition	Spooling, an acronym of Simultaneous Peripheral Operation Online (SPOOL), puts data into a temporary working area to be accessed and processed by another program or resource.	Buffering is an act of storing data temporarily in the buffer. It helps in matching the speed of the data stream between the sender and receiver.
Resource requirement	Spooling requires less resource management as different resources manage the process for specific jobs.	Buffering requires more resource management as the same resource manages the process of the same divided job.
Internal implementation	Spooling overlaps the input and output of one job with the computation of another job.	Buffering overlaps the input and output of one job with the computation of the same job.
Efficient	Spooling is more efficient than buffering.	Buffering is less efficient than spooling.
Processor	Spooling can also process data at remote sites. The spooler only has to notify when a process gets completed at the remote site to spool the next process to the remote side device.	Buffering does not support remote processing.
Size on memory	It considers the disk as a huge spool or buffer.	Buffer is a limited area in the main memory.

3. Time Sharing Operating System

- Cards and Tapes based system allowed only sequential access to programs and data. So only one application system could be used at a time.

- In the early days, computers were so user friendly that testing and development of programs was very easy for operators and programmers. But this arrangement resulted in more turnaround time.
- Due to wastage of CPU i.e., CPU’s idle time was more which was unreliable.
- In a time-sharing system, the time is managed properly for multiple users in an interactive manner and at a reasonable cost.
- A time-shared operating system allows multiple users to share computers simultaneously.
- Each action or order at a time the shared system becomes smaller, so only a little CPU time is required for each user.
- As the system rapidly switches from one user to another, each user is given the impression that the entire computer system is dedicated to its use, although it is being shared among multiple users.
- A time shared operating system uses CPU scheduling and multi-programming to provide each with a small portion of a shared computer at once.
- Each user has at least one separate program in memory.
- A program loaded into memory and executes, it performs a short period of time either before completion or to complete I/O.
- This short period of time during which the user gets the attention of the CPU is known as time slice, time slot or quantum.
- It is typically of the order of 10 to 100 milliseconds. Time shared operating systems are more complex than multiprogramming operating systems.
- In both, multiple jobs must be kept in memory simultaneously, so the system must have memory management and security.
- To achieve a good response time, jobs may have to swap in and out of disk from the main memory which now serves as a backing store for main memory.
- A common method to achieve this goal is virtual memory, a technique that allows the execution of a job that may not be completely in memory.



- In the above figure, the user 5 is active state but user 1, user 2, user 3, and user 4 are in waiting for state whereas user 6 is in the ready state.

1. **Active State –**

- The user's program is under the control of the CPU. Only one program is available in this state.

2. **Ready State –**

- The user program is ready to execute but it is waiting for its turn to get the CPU. More than one user can be in a ready state at a time.

3. **Waiting State –**

- The user's program is waiting for some input/output operation. More than one user can be in a waiting state at a time.

Advantages:

- Each task gets an equal opportunity.
- Fewer chances of duplication of software.
- CPU idle time can be reduced.

Disadvantages:

- Reliability problem.
- One must have to take off the security and integrity of user programs and data.

- Data communication problem.

4. Clustered Operating System.

- Clustered system is a group of computer systems connected with a high-speed communication link.
- Each computer system has its own memory and peripheral devices. Clustering provides high reliability.
- The clustered systems are a combination of hardware clusters and software clusters.
- Hardware clusters aid in the sharing of high-performance disks among all computer systems, while software clusters give a better environment for all systems to operate.
- A cluster system consists of various nodes, each of which contains its cluster software. The cluster software is installed on each node in the clustered system, and it monitors the cluster system and ensures that it is operating properly. If one of the clustered system's nodes fails, the other nodes take over its storage and resources and try to restart.
- Cluster components are generally linked via fast area networks, and each node executing its instance of an operating system. In most cases, all nodes share the same hardware and operating system, while different hardware or different operating systems could be used in other cases.
- The primary purpose of using a cluster system is to assist with weather forecasting, scientific computing, and supercomputing systems.

Types of Clustered Operating System

- There are mainly three types of clustered operating system:

1. **Asymmetric Clustering System**
2. **Symmetric Clustering System**
3. **Parallel Cluster System**

1. **Asymmetric Clustering System**

- In the asymmetric cluster system, one node out of all nodes is in hot standby mode, while the remaining nodes run the essential applications. Hot standby mode is completely fail-safe and also a component of the cluster system. The node monitors all server functions; the hot standby node swaps this position if it comes to a halt.

2. **Symmetric Clustering System**

- Multiple nodes help run all applications in this system, and it monitors all nodes simultaneously. Because it uses all hardware resources, this cluster system is more reliable than asymmetric cluster systems.

3. **Parallel Cluster System**

- A parallel cluster system enables several users to access similar data on the same shared storage system. The system is made possible by a particular software version and other apps.

Advantages

- Various advantages of Clustered Operating System are as follows:

1. High Availability

- Although every node in a cluster is a standalone computer, the failure of a single node doesn't mean a loss of service. A single node could be pulled down for maintenance while the remaining clusters take on a load of that single node.

2. Cost Efficiency

- When compared to highly reliable and larger storage mainframe computers, these types of cluster computing systems are thought to be more cost-effective and cheaper. Furthermore, most of these systems outperform mainframe computer systems in terms of performance.

3. Additional Scalability

- A cluster is set up in such a way that more systems could be added to it in minor increments. Clusters may add systems in a horizontal fashion. It means that additional systems could be added to clusters to improve their performance, fault tolerance, and redundancy.

4. Fault Tolerance

- Clustered systems are quite fault-tolerance, and the loss of a single node does not result in the system's failure. They might also have one or more nodes in hot standby mode, which allows them to replace failed nodes.

5. Processing Speed

- The processing speed is also similar to mainframe systems and other types of supercomputers on the market.

Disadvantages

- Various disadvantages of the Clustered Operating System are as follows:

1. Cost-Effective

- One major disadvantage of this design is that it is not cost-effective. The cost is high, and the cluster will be more expensive than a non-clustered server management design since it requires good hardware and a design.

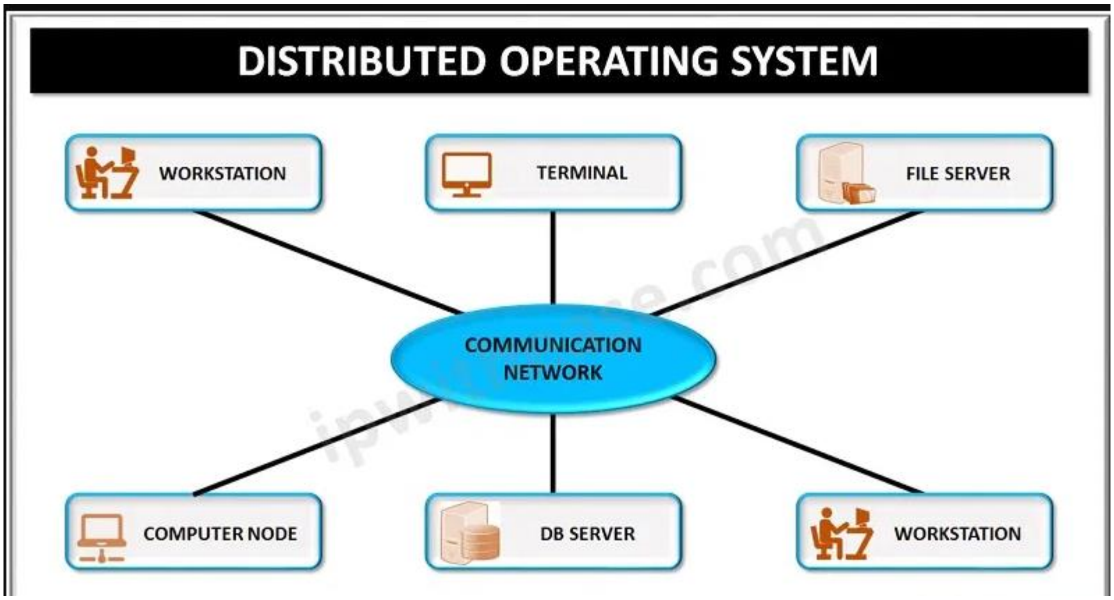
2. Required Resources

- Clustering necessitates the use of additional servers and hardware, making monitoring and maintenance difficult. As a result, infrastructure must be improved.

3. Maintenance

- It isn't easy to system establishment, monitor, and maintenance this system.

5. **Distributed Systems.**



- Distributed operating systems are loosely coupled systems. They are exactly opposite to tightly coupled.
- An example of the distributed operating system is Amoeba Operating System. In this system, the processor does not share memory or devices even clock. Each processor has got its own memory.
- The processor communicates with each other through high-speed buses or telephone lines.
- The two processors in this system may be different i.e. they may be microprocessors, workstations, minicomputers, etc.
- Their functions may be different. The main advantages of distributed systems are Resource Sharing. If a number of sites or machines are connected to each other, users can share the resources of other terminals or sites.
- Distributed systems use many central processors to serve multiple real-time applications and users. As a result, data processing jobs are distributed between the processors.

- It connects multiple computers via a single communication channel.
- This operating system consists of numerous computers, nodes, and sites joined together via **LAN/WAN** lines. It enables the distribution of full systems on a couple of center processors, and it supports many real-time products and different users. Distributed operating systems can share their computing resources and I/O files while providing users with virtual machine abstraction.

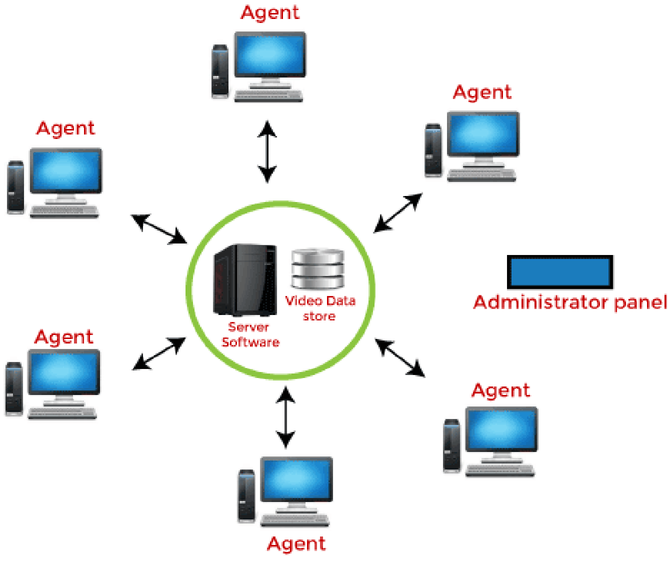
Advantages of Distributed Operating System:

- Failure of one will not affect the other network communication, as all systems are independent of each other
- Electronic mail increases the data exchange speed
- Since resources are being shared, computation is highly fast and durable
- Load on host computer reduces
- These systems are easily scalable as many systems can be easily added to the network
- Delay in data processing reduces

Disadvantages of Distributed Operating System:

- Failure of the main network will stop the entire communication
- To establish distributed systems, the language which is used are not well defined yet
- These types of systems are not readily available as they are very expensive. Not only that the underlying software is highly complex and not understood well yet
- Examples of Distributed Operating systems are- LOCUS, etc.

6. **Networked Operating System**.



Network Operating Systems

- A network operating system (NOS) is an operating system that manages network resources: essentially, an operating system that includes special functions for connecting computers and devices into a local area network (LAN). The NOS manages multiple requests (inputs) concurrently and provides the security necessary in a multiuser environment.
- A network operating system(NOS) is software that connects multiple devices and computers on the network and allows them to share resources on the network.
- Network Operating System runs on a server and gives the server the capability to manage data, users, groups, security, applications, and other networking functions.
- The composition of hardware that typically uses a NOS includes a number of personal computers, a printer, a server and a file server with a local network that connects them together. The role of the NOS is to then provide basic network services and features that support multiple input requests simultaneously in a multiuser environment.

The various Features of the Network Operating System are given below.

- Basic support for operating systems like protocol and processor support, hardware detection and multiprocessing.
- Printer and application sharing.

- Common file system and database sharing.
- Network security capabilities such as user authentication and access control.
- Directory
- Backup and web services.
- Internetworking.

Types of network operating systems

- There are two basic types of network operating systems, the peer-to-peer NOS and the client/server NOS:
- Peer-to-peer network operating systems allow users to share network resources saved in a common, accessible network location. In this architecture, all devices are treated equally in terms of functionality. Peer-to-peer usually works best for small to medium LANs and is cheaper to set up.
- Client/server network operating systems provide users with access to resources through a server. In this architecture, all functions and applications are unified under one file server that can be used to execute individual client actions regardless of physical location. Client/server tends to be most expensive to implement and requires a large amount of technical maintenance. An advantage to the client/server model is that the network is controlled centrally, makes changes or additions to technology easier to incorporate.

Advantages of Network operating systems:

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

Disadvantages of Network operating systems:

- Depend on the central location to perform the operations.
- High cost to buying server.

- Regular updating and maintenance are required.

Examples of Network Operating systems:

Following are the examples of network operating systems.

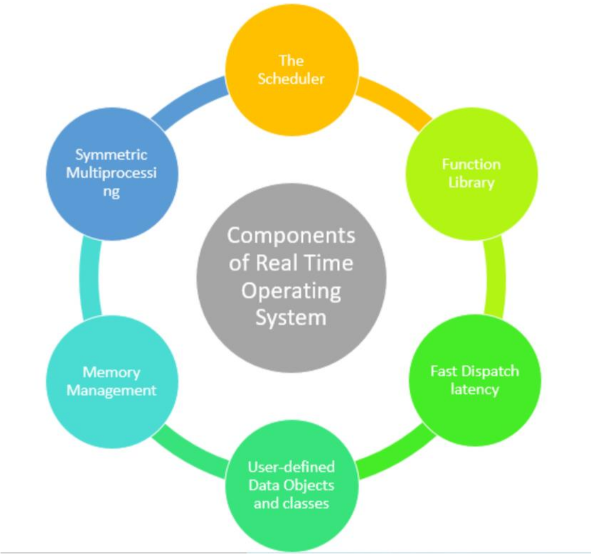
- Microsoft Windows Server
- UNIX/Linux

Real-Time Operating System

- Time constraints is the key parameter in a real-time system. It controls autonomous systems such as robots, satellites, air traffic control and hydroelectric dams.
- When a user gives an input to the system, it must process within the time limit and the result is sent back. The real-time system fails if it does not give results within the time limit.
- Real-time systems are of two types:
  1. Hard real-time
  2. Soft real-time
- Critical task is completed within the time limit in a hard real-time system. All the delay in the system is fixed and time-bounded. The existing general-purpose operating system does not support the hard real-time system functions. The real-time task cannot keep waiting for a longer time without allocating kernel.
- The soft real-time system is a less restricted type. Soft real-time cannot guarantee that it will be able to meet the deadline under all conditions. An example of a soft real-time system is digital audio.
- A soft real-time system is one where missing an occasional deadline, while not desirable, is acceptable and does not cause any permanent damage. Digital audio or multimedia systems fall in this category. Smartphones are also soft real-time systems.

- The real time-operating system uses a priority scheduling algorithm to meet the response requirement of a real-time application. General real-time applications with their examples are listed below:
- Transportation: Air traffic control and traffic light system
- Communication: Digital telephone.
- Process control: Papermill.
- Detecting: Burglar system and radar system.
- Flight simulation: Autopilot shuttle mission simulator.
- Real-time **operating systems (RTOS)** are used in environments where a large number of events, mostly external to the computer system, must be accepted and processed in a short time or within certain deadlines. such applications are industrial control, telephone switching equipment, flight control, and real-time simulations.

Components of RTOS:


Components of Real Time Operating System
The Scheduler
Function Library
Fast Dispatch latency
User-defined Data Objects and classes
Memory Management
Symmetric Multiprocessi ng

- The Scheduler: This component of RTOS tells that in which order, the tasks can be executed which is generally based on the priority.
- Symmetric Multiprocessing (SMP): It is a number of multiple different tasks that can be handled by the RTOS so that parallel processing can be done.

- Function Library: It is an important element of RTOS that acts as an interface that helps you to connect kernel and application code. This application allows you to send the requests to the Kernel using a function library so that the application can give the desired results.
- Memory Management: this element is needed in the system to allocate memory to every program, which is the most important element of the RTOS.
- Fast dispatch latency: It is an interval between the termination of the task that can be identified by the OS and the actual time taken by the thread, which is in the ready queue, that has started processing.
- User-defined data objects and classes: RTOS system makes use of programming languages like C or C++, which should be organized according to their operation.

Advantages:

- The advantages of real-time operating systems are as follows-
- **Maximum consumption –**
- Maximum utilization of devices and systems. Thus more output from all the resources.
- **Task Shifting –**
- The time assigned for shifting tasks in these systems is very less. For example, in older systems, it takes about 10 microseconds. Shifting one task to another and in the latest systems, it takes 3 microseconds.
- **Focus On Application** –
- Focus on running applications and less importance to applications that are in the queue.
- **Real-Time Operating System In Embedded System –**
- Since the size of programs is small, RTOS can also be embedded systems like in transport and others.

- **Error Free –**
- These types of systems are error-free.
- **Memory Allocation** –
- Memory allocation is best managed in these types of systems.

Disadvantages:

The disadvantages of real-time operating systems are as follows-

1. **Limited Tasks –**

- Very few tasks run simultaneously, and their concentration is very less on few applications to avoid errors.

2. **Use Heavy System Resources –**

- Sometimes the system resources are not so good and they are expensive as well.

3. **Complex Algorithms –**

- The algorithms are very complex and difficult for the designer to write on.

4. **Device Driver And Interrupt signals –**

- It needs specific device drivers and interrupts signals to respond earliest to interrupts.

5. **Thread Priority –**

- It is not good to set thread priority as these systems are very less prone to switching tasks.

6. Minimum Switching – RTOS performs minimal task switching**.**

System services:

- The operating system provides the programming environment in which a programmer works on a computer system. The user program requests various resources through the operating system. The operating system gives several services to utility programmers and users. Applications access these services through application programming interfaces or system calls. By invoking those interfaces, the application can request a service from the operating system, pass parameters, and acquire the operation outcomes.

Following are the services provided by an operating system -

- Program execution
- Control Input/output devices
- Error Detection and Response
- Security and Protection
- Resource Allocation
- File Management
- Communication

❖ Program execution

- To execute a program, several tasks need to be performed. Both the instructions and data must be loaded into the main memory.
- In addition, input-output devices and files should be initialized, and other resources must be prepared. The Operating structures handle these kinds of tasks.
- The user now no longer should fear the reminiscence allocation or multitasking or anything.
- Following are the major activities of an operating system with respect to program management −

- Loads a program into memory.
- Executes the program.
- Handles program's execution.
- Provides a mechanism for process synchronization.
- Provides a mechanism for process communication.
- Provides a mechanism for deadlock handling.

❖ Control Input/output devices

- As there are numerous types of I/O devices within the computer system, each I/O device calls for its own precise set of instructions for the operation. The Operating System hides that info with the aid of presenting a uniform interface. Thus, it is convenient for programmers to access such devices easily.

❖ Error Detection and Response

- An Error in a device may also cause malfunctioning of the entire device.
- These include hardware and software errors such as device failure, memory error, division by zero, attempts to access forbidden memory locations, etc.
- To avoid error, the operating system monitors the system for detecting errors and takes suitable action with at least impact on running applications.
- Errors may occur within CPU, memory hardware, I/O devices and in the user program. For each type of error, the OS takes adequate action for ensuring correct and consistent computing.
- The errors can be:

- **Input/ Output devices:** For example, connection failure in the network, lack of paper in the printer, etc.

- **User program:** For example: attempt to access illegal memory locations, divide by zero, use too much CPU time, etc.
- **Memory hardware:** For example, Memory error, the memory becomes full, etc.

❖ Security and 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 a computer system.
- Following are the major activities of an operating system with respect to protection −
- The OS ensures that all access to system resources is controlled.
- The OS ensures that external I/O devices are protected from invalid access attempts.
- The OS provides authentication features for each user by means of passwords.

❖ Resource Allocation

- In the case, of multi-user or multi-tasking environment, resources such as main memory, CPU cycles and files storage are to be allocated to each user or job.
- Following are the major activities of an operating system with respect to resource management −
- The OS manages all kinds of resources using schedulers.
- CPU scheduling algorithms are used for better utilization of CPU

❖ File Management

- A file represents a collection of related information. Computers can store files on the disk (secondary storage), for long-term storage purposes.
- Examples 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 data 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 the major activities of an operating system with respect to file management −
- Program needs to read a file or write a file.

- The operating system gives the permission to the program for operation on a file.
- Permission varies from read-only, read-write, denied and so on.
- Operating System provides an interface for the user to create/delete files.
- Operating System provides an interface for the user to create/delete directories.
- Operating System provides an interface to create the backup of the file system.

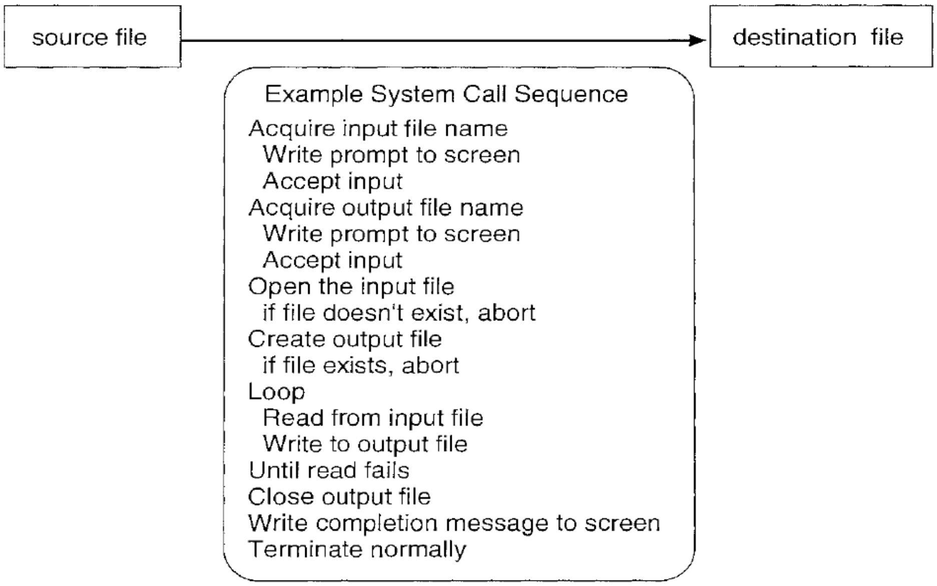
❖ <u>Communication</u>

- In the case of distributed systems which are a collection of processors that do not share memory, peripheral devices, or a clock, the operating system manages communications between all the processes.
- Multiple processes communicate with one another through communication lines in the network.
- The OS handles routing and connection strategies, and the problems of contention and security.
- Following are the major activities of an operating system 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.

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.

- A system call is a communication made by an operating system with the processes.
- System Calls provides an interface between the process and the operating system. System calls allow user-level processes to request some services from the operating system which the process itself is not allowed to do.
- In handling the trap, the operating system will enter in the kernel mode, where it has access to privileged instructions and can perform the desired service on the behalf of the user-level process.
- The kernel is a main part or component of an operating system.
- It is because of the critical nature of operations that the operating system itself does them every time they are needed.
- For example, for I/O a process involves a system call telling the operating system to read or write a particular area and this request is satisfied by the operating system.
- System programs provide basic functioning to users so that they do not need to write their own environment for program development. E.g. editors, compilers etc. and program execution e.g. shells. In short, they are bundles of useful system calls.

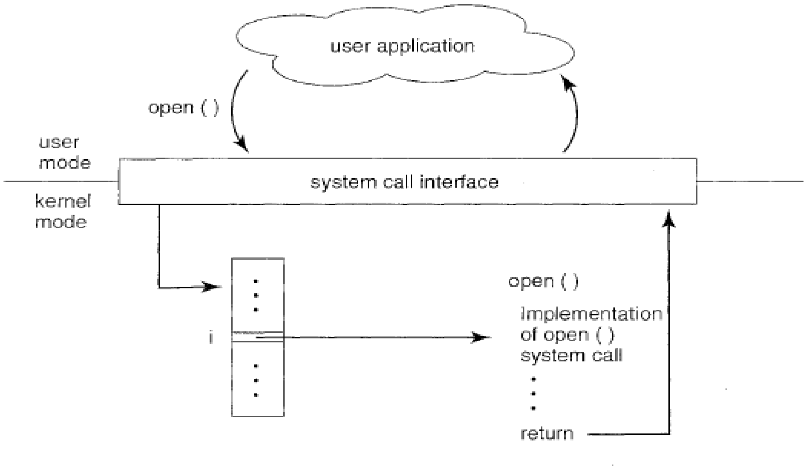


```
Example System Call Sequence
Acquire input file name
  Write prompt to screen
  Accept input
Acquire output file name
  Write prompt to screen
  Accept input
Open the input file
  if file doesn't exist, abort
Create output file
  if file exists, abort
Loop
  Read from input file
  Write to output file
Until read fails
Close output file
Write completion message to screen
Terminate normally
```

- The above diagram for system calls describes an example where one file is copied on another file. Consider two files one is the source and the other is the destination.
- If data is to be copied from one file to other, then the file names should be known. Either it should be initialized in the program or input should be taken interactively from the user when prompted.
- When files names are inputted, the system checks whether the file exists or do it requires access permission; if yes then contents are read from the respective files.
- Just observe the actions that are taken for copying the file, where every time system call is to be made by the operating system to do each and every action.
- After confirming details about a file either error messages are prompted on the file or further reading and writing actions are made. In this example system, calls are needed for taking input from one file and putting it into another file.

System Call Implementation:

- Generally, a number is associated with each system call. It is used to number the system calls. System – call interface maintains a table indexed according to these numbers.

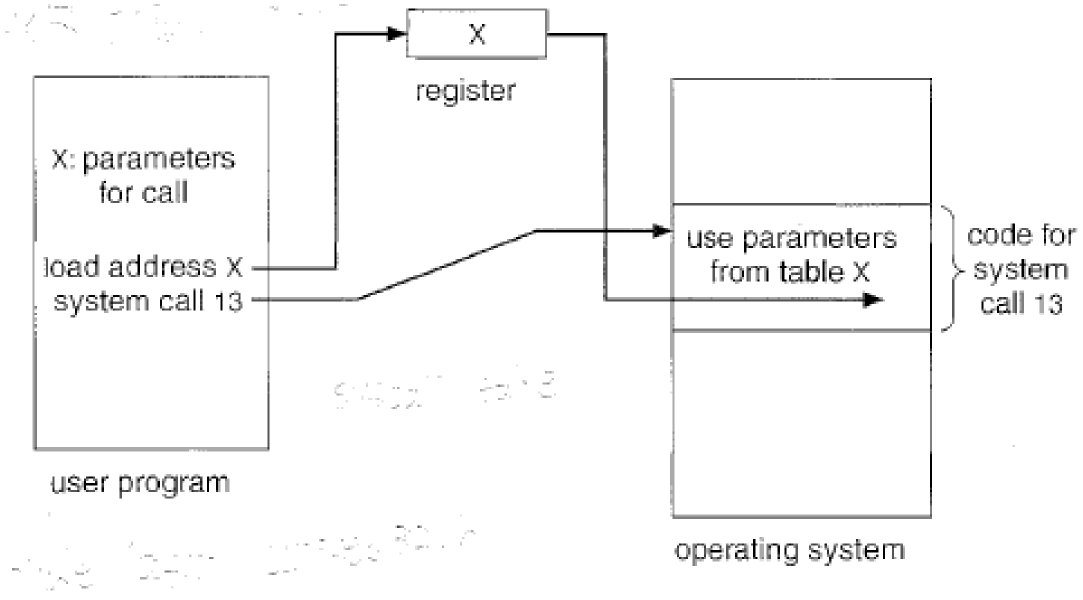


- The system call interface invokes the intended system call in OS kernel and returns status of the system call and return values.

- The caller needs to know nothing about how the system call is implemented. Just needs to use API (application interface) and understand what OS will do as a result of the call.
- Most of the details of the OS interface are hidden from programmer by API. It is managed a by run-time support library. Complier includes set of functions built into libraries.
- The relationship between an API, the system-call interface, and the operating system is shown in Figure. The figure illustrates how the operating system handles a user application invoking the open() system call.

System Call Parameter Passing:

- Three general methods are used to pass parameters to the operating system.
- The simplest approach is to pass the parameters in registers.
- In some cases, however, there may be more parameters than registers.



- In such cases, the parameters are generally stored in a block, or table, in memory, and the address of the block is passed as a parameter in a register as depicted in above given figure.

- This is the approach taken by Linux and Solaris. Parameters also can be placed, or pushed, onto the stack by the program and popped of the stack by the operating system.
- Some operating systems prefer the block or stack method because those approaches do not limit the number or length of parameters being passed.

Types of System Calls:

- There are 5 different categories of system calls –

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

System Calls	Windows	Unix

Process Control	CreateProcess()
	ExitProcess()
	WaitForSingleObject()
		fork()
		exit()
		wait()

File Manipulation	CreateFile()
	ReadFile()
	WriteFile()
	CloseHandle()
		open()
		read()
		write()
		close()

Device Manipulation	SetConsoleMode()
	ReadConsole()
	WriteConsole()
		ioctl()
		read()
		write()

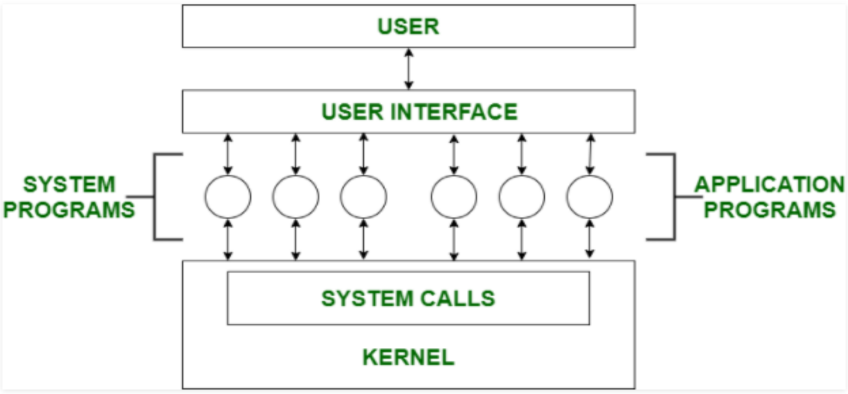
Information Maintenance	GetCurrentProcessID()
	SetTimer()
	Sleep()
		getpid()
		alarm()
		sleep()

Communication	CreatePipe()
	CreateFileMapping()
	MapViewOfFile()
		pipe()
		shmget()
		mmap()

Protection	SetFileSecurity()
	InitlializeSecurityDescriptor()
	SetSecurityDescriptorGroup()
		chmod()
		umask()
		chown()

System Programs in Operating System

- **System Programming** can be defined as the act of building Systems Software using System Programming Languages.
- According to Computer Hierarchy, one which comes at last is Hardware.
- Then it is Operating System, System Programs, and finally Application Programs. Program Development and Execution can be done conveniently in System Programs.
- Some of the System Programs are simply user interfaces, others are complex.
- It traditionally lays between the user interface and system calls.



- So here, the user can only view up-to-the System Programs he can't see System Calls.
- System Programs can be divided into these categories:

1. File Management –

- A file is a collection of specific information stored in the memory of a computer system. File management is defined as the process of manipulating files in the computer system; its management includes the process of creating, modifying and deleting files.
- It helps to create new files in the computer system and place them at specific locations.
- It helps in easily and quickly locating these files in the computer system.
- It makes the process of sharing files among different users very easy and user-friendly.
- It helps to store files in separate folders known as directories.
- These directories help users to search files quickly or to manage files according to their types of uses.
- It helps users to modify the data of files or to modify the name of files in directories.

2. Status Information –

- Information like date, time amount of available memory, or disk space is asked by some users.
- Others provide detailed performance, logging, and debugging information which is more complex.
- All this information is formatted and displayed on output devices or printed.
- Terminal or other output devices or files or a window of GUI is used for showing the output of programs.

3. File Modification –

- For modifying the contents of files we use this. For Files stored on disks or other storage devices, we used different types of editors.
- For searching contents of files or perform transformations of files we use special commands.

4. Programming-Language support –

- For common programming languages, we use Compilers, Assemblers, Debuggers, and interpreters which are already provided to users.
- It provides all support to users. We can run any programming language. All languages of importance are already provided.

5. Program Loading and Execution –

- When the program is ready after Assembling and compilation, it must be loaded into memory for execution.
- A loader is part of an operating system that is responsible for loading programs and libraries. It is one of the essential stages for starting a program.
- Loaders, relocatable loaders, linkage editors, and Overlay loaders are provided by the system.

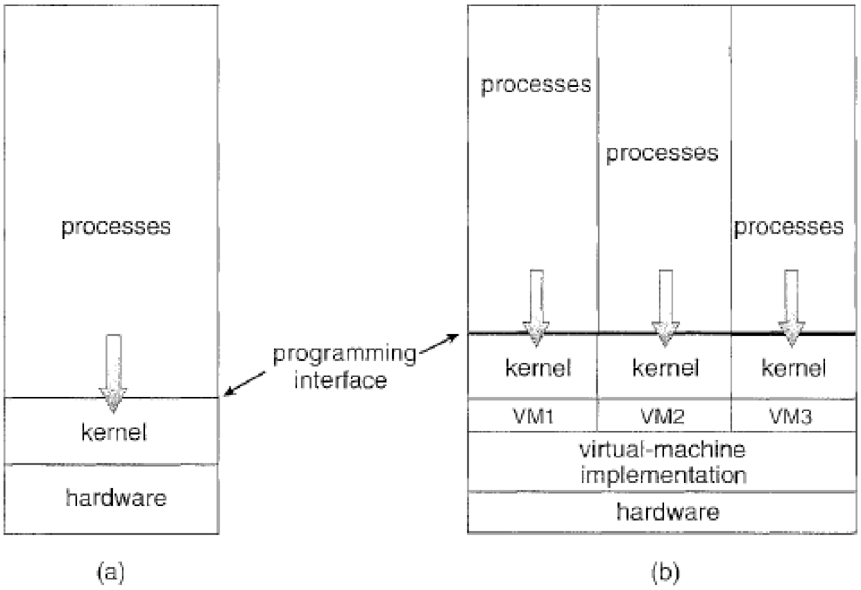
6. Communications –

- Virtual connections among processes, users, and computer systems are provided by programs.
- Users can send messages to another user on their screen; User can send e-mail, browsing on web pages, remote login, and the transformation of files from one user to another.

<u>Virtual Machines:</u>

- Different classes of users need different kinds of user service. Hence running a single OS on a computer system can disappoint many users. Operating the computer under different OS's during different periods is not a satisfactory solution because it would make accessible services offered under only one of the operating systems at any time.
- This problem is solved by using a virtual machine operating system (VM OS) to control the computer system.
- The VM OS creates several virtual machines.

- Each virtual machine is allocated to one user, who can use any OS of his own choice on the virtual machine and run his programs under this OS.
- This way user of the computer system can use different operating systems at the same time.
- We call each of these operating systems a guest OS and call the virtual machine OS the host OS.
- The computer used by the VM OS is called the host machine.



- A **virtual machine (VM)** is a virtual environment that functions as a virtual computer system with its own CPU, memory, network interface, and storage, created on a physical hardware system.
- VMs are isolated from the rest of the system, and multiple VMs can exist on a single piece of hardware, like a server. That means it is a simulated image of application software and operating system which is executed on a host computer or a server.
- It has its own operating system and software that will facilitate the resources to virtual computers.
- The diagram above diagram depicts virtual machine environment.
- Virtual machines first appeared commercially on IBM mainframes via the VM operating system in 1972. VM has evolved and is still available,

and many of the original concepts are found in other systems, making this facility worth exploring.

❖ <u>Characteristics of virtual machines</u>

- The characteristics of virtual machines are as follows −
- Multiple OS systems use the same hardware and partition resources between virtual computers.
- Separate Security and configuration identity.
- Ability to move the virtual computers between the physical host computers as holistically integrated files.
- Example of a Virtual Machine is VMWare.

- VMware is a virtualization and cloud computing software provider based in Palo Alto, Calif. Founded in 1998, VMware is a subsidiary of Dell Technologies.
- EMC Corporation originally acquired VMware in 2004; EMC was later acquired by Dell Technologies in 2016. VMware bases its virtualization technologies on its bare-metal hypervisor ESX/ESXi in x86 architecture.
- With VMware server virtualization, a hypervisor is installed on the physical server to allow for multiple virtual machines (VMs) to run on the same physical server.
- Each VM can run its own operating system (OS), which means multiple OSes can run on one physical server. All the VMs on the same physical server share resources, such as networking and RAM.
- In 2019, VMware added support to its hypervisor to run containerized workloads in a Kubernetes cluster in a similar way. These types of workloads can be managed by the infrastructure team in the same way as virtual machines and the DevOps teams can deploy containers as they were used to.

❖ <u>Advantages:</u>

1. There are no protection problems because each virtual machine is completely isolated from all other virtual machines.

2. Virtual machines can provide an instruction set architecture that differs from real computers.
3. Easy maintenance, availability and convenient recovery.

❖ Disadvantages:

1. When multiple virtual machines are simultaneously running on a host computer, one virtual machine can be affected by other running virtual machines, depending on the workload.
2. Virtual machines are not as efficient as a real one when accessing the hardware.

Operating system and Design Implementation:

Design Goals:

- The first problem in designing a system is to define goals and specifications.
- At the highest level, the design of the system will be affected by the choice of hardware and the type of system: batch, time shared, single user, multiuser, distributed, real time, or general purpose.
- Beyond this highest design level, the requirements may be much harder to specify.
- The requirements can, however, be divided into two basic groups: user goals and system goals.
- Users desire certain obvious properties in a system. The system should be convenient to use, easy to learn and to use, reliable, safe, and fast.
- Of course, these specifications are not particularly useful in the system design, since there is no general agreement on how to achieve them.
- A similar set of requirements can be defined by those people who must design, create, maintain, and operate the system. The system should be easy to design, implement, and maintain; and it should be flexible, reliable, error free, and efficient. Again, these requirements are vague and may be interpreted in various ways.
- There is, in short, no unique solution to the problem of defining the requirements for an operating system. The wide range of systems in

existence shows that different requirements can result in a large variety of solutions for different environments. For example, the requirements for VxWorks, a real-time operating system for embedded systems, must have been substantially different from those for MVS, a large multiuser, multi-access operating system for IBM mainframes.

- Specifying and designing an operating system is a highly creative task.

Mechanism and Policies:

- One important principle is the separation of policy from mechanism.
- The mechanism determines how to do something; policies determine what will be done. For example, the timer construct is a mechanism for ensuring CPU protection, but deciding how long the timer is to be set for a particular user is a policy decision.
- Policy decisions are important for all resource allocation. Whenever it is necessary to decide whether or not to allocate resources, a policy decision must be made. Whenever the question is how rather than what, it is the mechanism that must be determined.

Implementation:

- Once an operating system is designed, it must be implemented. Traditionally, operating systems have been written in assembly language.
- Now, however, they are most commonly written in higher-level languages such as C or C++.
- The Linux and Windows XP operating systems are written mostly in C, although there are some small sections of assembly code for device drivers and for saving and restoring the state of registers.
- The advantages of using a higher-level language, or at least a systems implementation language, for implementing operating systems are the same as those accrued when the language is used for application programs: the code can be written faster, is more compact, and is easier to understand and debug.

- For example, MS-DOS was written in Intel 8088 assembly language. Consequently, it runs natively only on the Intel X86 family of CPUs. (Although MS-DOS runs natively only on Intel X86, emulators of the X86 instruction set allow the operating system to run non-natively slower, with more resource use-on other CPUs. are programs that duplicate the functionality of one system with another system.)
- The Linux operating system, in contrast, is written mostly in C and is available natively on a number of different CPUs, including Intel X86, Sun SPARC, and IBM Power PC.
- The only possible disadvantages of implementing an operating system in a higher-level language are reduced speed and increased storage requirements.

System Generations:

- It is possible to design, code, and implement an operating system for one machine at one site. More commonly, however, operating systems are designed to run on any of a class of machines at a variety of sites which a variety of peripheral configurations.
- The system must then be configured or generated for each specific computer site, a process sometimes known as system generation SYSGEN.
- The operating system is normally distributed on disk, on CD-ROM or DVD-ROM or as an “ISO” image, which is a file in the format of a CD-ROM or DVD-ROM. To generate a system, we use a special program.
- This SYSGEN program reads from a given file, or asks the operator of the system for information concerning the specific configuration of the hardware system, or probes the hardware directly to determine what components are there. The following kinds of information must be determined.
- What CPU is to be used? What options (extended instruction sets, floating-point arithmetic, and so on) are installed? For multiple CPU systems, each CPU may be described.
- How will the boot disk be formatted? How many sections, or ‘partitions’, will it be separated into, and what will go into each partition?

- How much memory is available? Some systems will determine this value themselves by referencing memory location from memory location until an “illegal address” fault is generated. This procedure defines the final legal address and hence the amount of available memory.
- What devices are available? The system will need to know how to address each device (the device number), the device interrupt number, the device’s type and model, and any special device characteristics.
- Once this information is determined, it can be used in several ways. At one extreme, a System administrator can use it to modify a copy of the source code of the OS. The OS then is completely complied. Data declarations, initializations, and constants, along with conditional compilation, produce an output-object version of the OS that is tailored to the system described.
- System generation involves simply creating the appropriate tables to describe the system. The major differences among these approaches are the size and generality of the generated system and the ease of modifying it as the hardware configuration changes.
- Consider the cost of modifying the system to support the newly acquired graphics terminal or another disk drive. Balanced against that cost, of course, is the frequency of such changes.

Prof. A. S. Aher
(Subject In-charge)