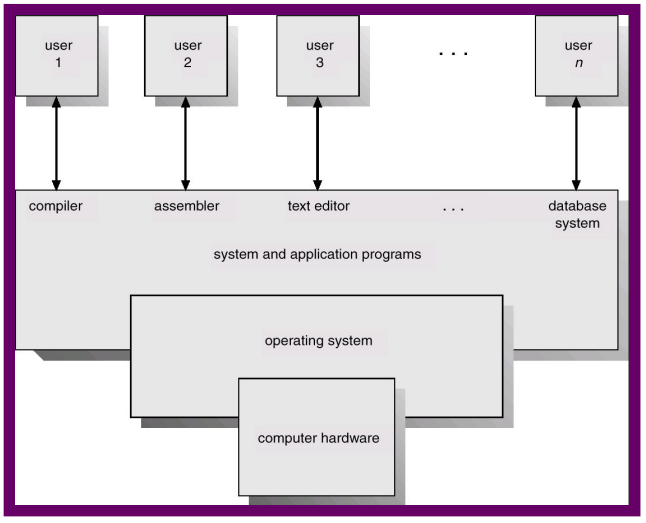
Chapter 1 Operating System Concepts

1.1 Evolution of Operating System & History

What is Operating System?

- An operating system is a program that acts as an intermediary between a user of a computer and the computer hardware.
- Purpose of operating system:
  - To provide an environment in which a user can execute programs and applications.
  - To allow user to access the computer resources.
  - To provide simple interface to the user so he can use the computer system efficiently.
- Goal of operating system:
  - To make the computer system convenient for user.
  - To use the computer hardware in an efficient manner.

How operating system is important part of the computer system? Or Explain components of the computer system?



The computer system can be divided into four main components:

1) User
2) Application program
3) Operating system
4) Hardware

1) **Hardware:** The hardware comes at the lowest level. It contains various kinds of physical devices. Like processor, memory, keyboard, mouse, monitor etc…

2) **Operating system:** The next level is for the operating system. It manages all the underlying hardware. It hides complex details of hardware from the user. Thus, it provides simple interface between application program and hardware. It also works as resource allocator. It controls input output devices and users.

3) **Application Programs:** These are the applications through which user can communicate with computer internal system. They use different kinds of functionalities provided by operating system to perform their tasks. Application programs include airline reservation system, web-browser, games, MS-Office, etc…

4) **Users:** Users are at the top level. Users interact with the system by using application program to perform particular tasks.

History of operating systems

Starting from beginning, computers and operating system have evolved a lot. Very early, Charles Babbage, he was English mathematician, has designed a true digital computer. He spends his most of life in making machine work properly but he remained unsuccessful. As an important outcome of these efforts, he realized the need of software. After that unsuccessful attempt of Babbage history was as follow:

1) **The First Generation (1945 - 1955):**
   - Hardware: Vacuum tubes and plug boards.
   - Neumann and other people help in building calculating engine.
   - No operating system, No Programming languages.
   - Introduction of punch cards.

2) **The Second Generation (1955 – 1965):**
   - Hardware: Transistors.
   - Clear separation between designers, builders, operators, programmers and maintenance personnel.
   - Machines were called Mainframes.
   - Batch operating system took birth.

3) **The Third Generation (1965 – 1980):**
   - Hardware: Integrated Circuits (ICs).
   - Multiprogramming operating system and variations of it such as Time sharing, Interactive, Multitasking operating system came in picture.

4) **The Fourth Generation (1980 – Present):**
   - Hardware: Large Scale Integration (LSI) circuits.
   - Personal Computers evolved.

In early 1980, IBM designed IBM PC. They wanted software so that their system can work. So, people from IBM contacted Bill Gates to get some help. Gates suggested them to meet Kildall.

At that time Kildall was running a company named Digital Research. Digital Research was the world's leading operating system company of that time.

But, unfortunately, kildall refused to meet IBM, and the deal between them failed. This decision of kildall is known as the worst business decision.

After then, IBM again contacted Gates. Now, Gates bought DOS (Disk Operating System) from a company named Settle Computer Products. Then, Gates hired a person, designer of DOS, Tim Peterson, to modify the DOS. This revised DOS is known as MS-DOS.

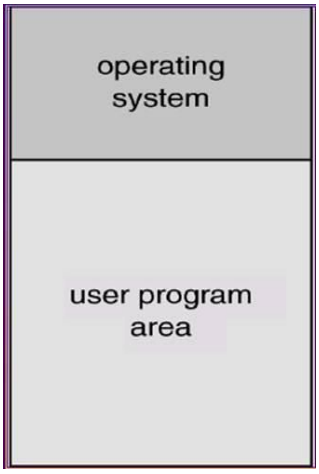
IBM used this MS-DOS and became popular in the market of PCs. And history knows very well about what happened with the minded man, Mr. Bill Gates…!!!

Later on Apple came with a system which provided user friendly GUI. And OS like Windows 95, 98, XP are invented.

In a parallel to this UNIX and Linux also got a good evolution.

Type of Operating System

Batch System



- Early computers were run from console. The card readers and tape drives were input devices. Line printers, tape drives and card punched were common output devices. The user did not interact directly with computer system. The user prepared a job that consisted of program, data and some control information (control cards). He then submitted it to the computer operator in the form of punched cards.
- The operating system was simple. Its basic job was to transfer control automatically from one job to the next. The operating system was always resident in memory. The operator .batched similar jobs together and then run in the computer to speed up the processing.
  The CPU is often idle in this environment as speed of I/O devices is much slower than CPU. After sometime, the introduction of disk instead of card reader resulted in faster I/O devices.

- In disk technology, the operating system keeps all jobs on a disk instead of card reader. The resources are utilized and jobs are performed more efficiently with the help of job scheduling. Job scheduling is possible because all jobs are present on the disk.
  The memory contains operating system in one part and user program in other part.

Processor point of view

The processor is executing instructions from the portion of main memory containing the monitor. These instructions cause the next job to be read into another portion of main memory.

Once a job has been read in, the processor will encounter a branch instruction in the monitor that instructs the processor to continue execution at the start of the user program.

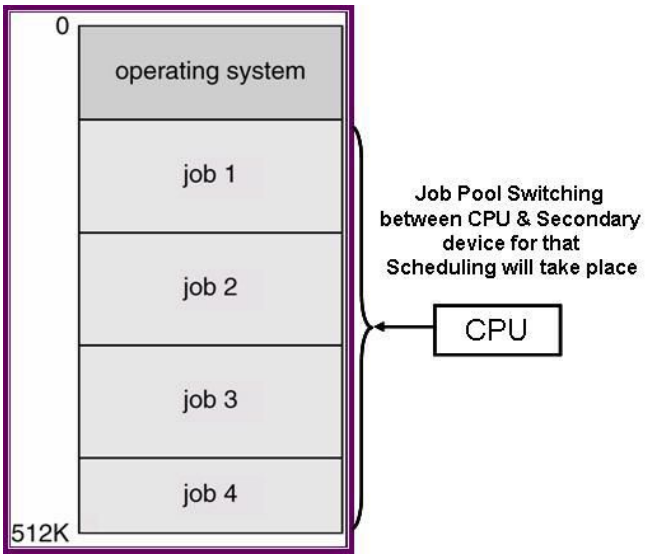
- The processor will then execute the instructions in the user program until it encounters an ending or error condition.
- Either event causes the processor to fetch its next instruction from the monitor program. Thus the phrase "control is passed to a job" simply means that the processor is now fetching and executing instructions in a user program and "control is returned to the monitor" means that the processor is now fetching and executing instructions from the monitor program.
- JCL (job Control Language): With each job, instructions are included in primitive form of JCL which is used to provide instructions to the monitor.

Disadvantages:

- The operator grouped all of the jobs into various batches with similar characteristics (all the quick jobs might run, then the slower ones, etc.) before running them as one at a time so user must wait for results until batch collected and submitted.
- The I/O used slow mechanical devices, the CPU was often idle waiting on a card to be read or some result to be printed, etc.

Multiprogrammed Systems

- With the automatic job sequencing provided by a simple batch operating system, the processor is often idle. The problem is that I/O devices are slow compared to the processor.
- Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute.
- The idea is as follows: The OS keeps several jobs in memory simultaneously. (Figure).



- This set of jobs can be a subset of the jobs kept in the job pool-which contains all jobs that enter the system.
- The operating system picks and begins to execute one of the jobs in the memory. Once this job need an I/O operation the O.S switches to another job (CPU or O.S always busy).
- If several jobs are ready to be brought into memory and there is not enough room for all of them, then the system chooses jobs among them (job Scheduling).
- If several jobs are ready to run at the same time, the system must choose among them (CPU Scheduling).
- Having several programs in memory at the same time requires memory management.
- Multiprogrammed systems provide an environment in which the various system resources (for example, CPU, memory and peripheral devices) are utilized effectively.
- It improves throughout the utilization.
- In a non-multiprogrammed system, the CPU would sit idle. In multiprogramming system, CPU will never be idle.

Time sharing systems (Multitasking)

- While multiprogrammed systems used resources more effectively i.e. minimized CPU idle time, a user could not interact with a program properly.
- Time sharing (or multitasking) is a logical extension of multiprogramming. In a time sharing systems, the CPU executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with each program while it is running.
- The user gives instruction to the OS or to a program directly, using an input device such as a keyboard or mouse, and waits for immediate results on an output device. Accordingly, the response time should be short-typically less than one second.

- As the system switches rapidly from one user to the next, each user is given the impression that the entire computer system is dedicated to his use, even though it is being shared among many users.
- A time-shared OS uses CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared computer.
- Time-sharing and multiprogramming require several jobs to be kept simultaneously in memory. Since in general main memory is too small to accommodate all jobs, the jobs are kept initially on the disk in job pool.
- Processes are swapped in and out of main memory to the disk. In effect, we are now "memory sharing" between competing users (programs). This idea leads to a mechanism called virtual memory.
- Virtual memory is a technique that allows the execution of a process that is not completely in memory. The main advantage of the virtual-memory scheme is that is enable user to run programs that are larger than actual physical memory.
- Time-sharing system must also provide a file system. The file system resides on a collection of disk; therefore, disk management must be provided.
- Also time-sharing systems provide a mechanism for protecting resources from inappropriate use.
- To ensure orderly execution, the system must provide mechanisms for job synchronization and communication, and it may ensure that jobs do not get stuck in a deadlock, forever waiting for one another.

Desktop systems

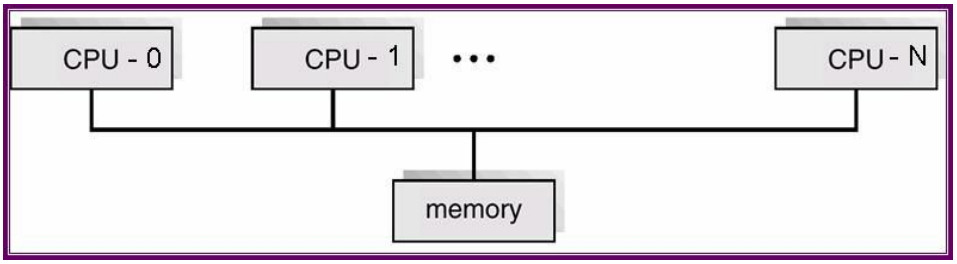
- Personal computers PCs appeared in 1970s
- Previously consideration as individuals have sole use of computer, do not need advance CPU utilization, protection features .therefore PC operating system neither multi-user nor multitasking.
- The goals of PC operating system were to maximize user convenience and responsiveness instead of maximizing CPU and I/O utilization
- PC may run several different types of OS (windows, Macos x, unix, linux) which offer multi tasking and virtual memory on PC hardware.
- Most system uses a single processor.
- On a single processor system, there is one main CPU capable of executing a general purpose instruction set, including instruction from user processes.
- Almost all systems have other special purpose processors as well. They may come in form of device specific processors, such as disk, keyboard and graphics controller, on main frames. They may come in the form of more general processors, such as I/O processors that move data rapidly among the components of the system.
- All of these special purpose processors run a limited instruction set and don't run user processes. Sometime they are managed by the OS, in that the OS sends information about their next task and monitor their status.

- For example, a disk controller micro processor receives a sequence of request from the main CPU and implements its own disk queue and scheduling algorithm. This arrangement relieves the main CPU of the overhead of disk scheduling.
- PCs contain a micro processor in the keyboard to convert the keystrokes into codes to be sent to the CPU.
- The use of special purpose micro processors is common and does not turn a single processor system in to a multi processor.

MULTIPROCESSOR SYSTEMS

- Multi processors systems there are more than one processor that works simultaneously and sharing the computer bus and sometime the clock, memory and peripheral device.
- Communication usually takes place through the shared memory.
- multi processors systems have 3 main advantages :

1. **Increase throughput:** by increasing the number of processors, we accept to get more work done in less time. When multi processors co-operate on a task, a certain amount of overhead is invited in keeping all the parts working correctly.



2. **Economy of scale:** multiprocessor systems' cost is less than equivalent multiple single processors systems, because they can share peripherals. Mass storage and power supplies.
3. **Increased reliability:** if functions can be distributed properly among several processors, then the failure of one processor will not halt the system, only slow it down.

- The multiple-processor systems in use today are of following types.

1. Some system use asymmetric multi-processing, in which each processor assigned a specific task. A master processor controls the system; processors either look to the master for instruction or have predefined tasks. This schema defines a master-slave relationship.
2. The most common system use symmetric multiprocessing (SMP), in which each processor performs all tasks within the OS. SMP means that all processors are peers; no master-slave relationship exists between processors.
   - The benefit of SMP model is that many processes can run simultaneously. However, we must carefully control I/O to ensure that the data reach the appropriate processor.

- Because of the CPU are separate, one may be sitting idle while another is overloaded, resulting in inefficiencies.
- Virtually all model OS including windows, windows XP, mac OSx and Linux-now provide support for SMP.

The difference between symmetric multiprocessing and asymmetric multiprocessing is based on the result from hardware or software. Special hardware can differentiate the multiple processors, or the software can be written to allow only one master and multiple slaves.

Distributed Systems

- A distributed system (Loosely coupled systems) consists of a collection of independent Computers, connected through a network, which enables computers to coordinate their activities and to share the resources of the system, so that users recognize the system as a single, integrated computing facility.
- They do not share memory and clock. Each processor has its own local memory.
- The processors communicate with one another through various communication lines, such as high-speed buses or telephone line.
- Access to a shared resource increases computation speed, functionality, data availability, and reliability.
- Network change by the protocols used the distances between nodes and the transport media.
- These networks also vary by their performance and reliability. May be either client-server or peer-to-peer systems.

Client-Server systems

- As PCs have become faster, more powerful, and cheaper, designers have shifted away from centralized system architecture. Terminals connected to centralized systems are now being supplanted by PCs.
- As a result, many of today's system act as server systems to satisfy requests generated by client system. This from of specialized distributed system, called client-server system, has the general structure depicted in figure.

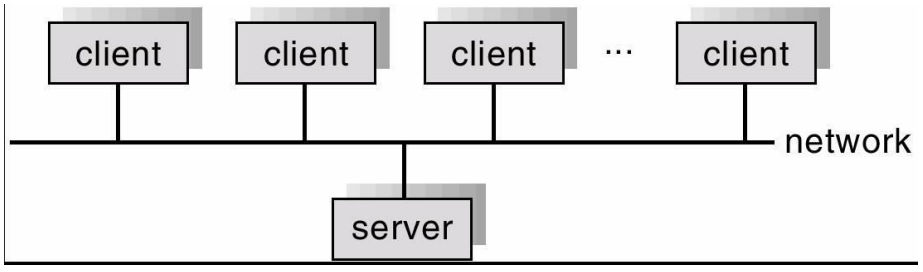
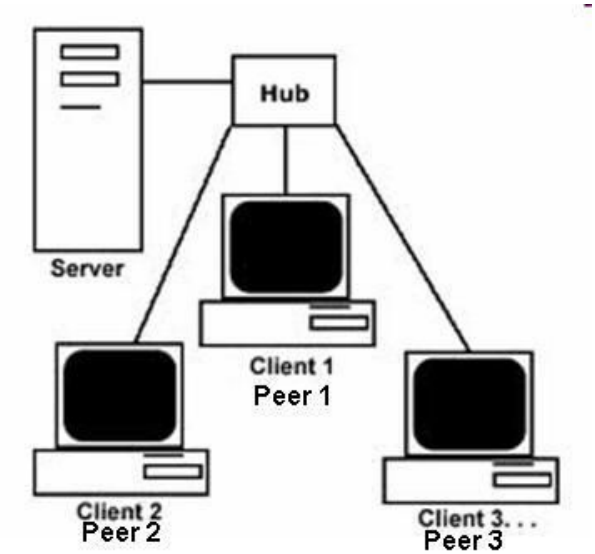


Figure shows general structure of a client-server system.

- Sever system can be broadly categorized as compute server and the file server.

- The **compute-server system** provides an interface to which a client can send a request to perform an action (for example, read data); in response, the server executes the action and sends back result to the client. A server running a database that responds to client requests for data is an example of such a system.
- Another example of a client-server system is the **file-server system** on campus. Here, a central server provides file access to authenticated users at client machines.

Peer-to-peer systems



- Peer-to-peer systems are another form of distributed system in which the participating computer systems network connectivity.
- Some OS have taken the concept of networks and distributed systems further than the notion of providing network connectivity.
- A network OS is an OS that provides features such as file sharing across the network and that includes a communication scheme that allows different process on different computers to exchange messages.
- A computer running a network OS acts autonomously from all other computer on the network, although it is aware of the network and is able to communicate with other networked computers.
- A distributed OS provides a less autonomous environment: the different OS communicate closely enough to provide the false impression that only a single OS controls the network.

Real – Time Systems (ATM)

- A real time system is used when rigid time requirements have been placed on the operation of a processor or the flow of data; thus, it is often used as control device in a dedicated application.

- Embedded computers are the most common form of computers in existence. These devices are found everywhere, from car engines and manufacturing robots to VCRs and microwave ovens. They tend to have very specific tasks.
- Usually, they have little or no user interface, preferring to spend their time monitoring and managing hardware devices, such as automobile engines and robotic arms.
- In a real-time system the correctness of the system behaviour depends not only on the logical results of the computations, but also on the physical instant at which these results are produced.
- A real-time system changes its state as a function of physical time, eg. a chemical reaction continues to change its state even after its controlling computer system has stopped.
- Based on this a real-time system can be decomposed into a set of subsystems i.e., the controlled object, the real-time computer system and the human operator.

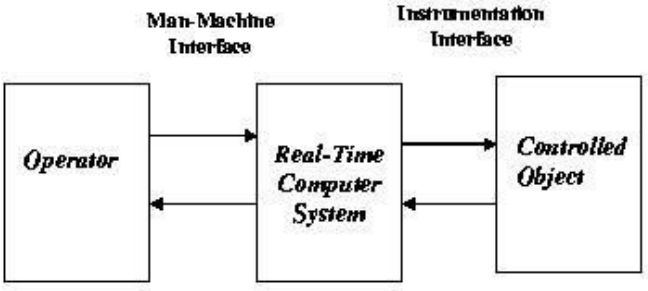


Figure 1: Real-Time System

- A real-time computer system must react to stimuli from the controlled object (or the operator) within time intervals dictated by its environment. The instant at which a result is produced is called a deadline.
- If the result has utility even after the deadline has passed, the deadline is classified as soft, otherwise it is firm. If a catastrophe could result if a firm deadline is missed, the deadline is hard.
- Real-time systems may be either;
- Hard (must react in time): A Hard real-time system guarantees that critical tasks be completed on time. This goal requires that all delays in the system be bounded, from the retrieval of stored data to the time that it takes the operating systems are examples for hard real-time systems.
- Soft (deal with failure to react in time): The soft real-time system can satisfy its performance criteria by running any critical task at a higher priority. Given their lack of deadline support, they are risky to use for industrial control and robotics. They are useful in applications (multimedia, virtual reality) requiring advanced

operating-system features. On-line transaction systems, airline reservation systems are soft real-time systems.

- **There are two types of real time operating system:**
  - ➢ **Hard real time system:** All critical tasks must get completed strictly within the specified time limits. Here task are guarantee to occur in time. **Example** system which controls the operation of an oil refinery.
  - ➢ **Soft real time System:** Less strict. Missing an occasional dead line is acceptable. **Example** Digital audio or multimedia system.

Handheld Systems

- Handheld systems include personal digital assistants (PDAs), such as palm and pocket-PCs, and cellular telephones, connectivity to a network such as the internet.
- Developers of handheld systems and applications face many challenges, most of which are due to the limited size of such devices. For example, a PDA is typically about 5 inches in height and 3 inches in width, and it weighs less than one-half pound. Due to this limited size, most handheld devices have a small amount of memory, include slow processors, and features small display screens.
- Many handheld devices have between 512 KB and 8 MB of memory.
- As a result, the operating system and applications must manager once the memory is no longer being used.
- Currently, many handheld devices do not use virtual memory techniques, thus forcing program developers to work within the confines of limited physical memory.
- A second issue of concern to developers of handheld devices is the speed of the processor used in the device. Processors for most handheld devices often run at a fraction of the speed of a processor in a PC.
- Faster processors require more power. To include a faster processor in a handheld device would require a larger battery that would have to be replaced (or recharged) more frequently.
- The last issue confronting program designers for handheld devices is the small display screens typically available.

Generally, the limitations in the functionality of PDAs are balanced by their convenience and portability. Their use continues to expand as network connections become more available and other options, such as cameras and MP3 players, expand their utility.

1.2 Needs of an Operating System

Why Operating system is required? Or what is the goal of operating system? Or what is the objective of Operating system? Or which are the services provided by operating system?

Services of OS:

- ***User point of view***

1) **Program execution**

The main purpose of OS is to provide an efficient and convenient environment for the execution of programs. So, an OS must provide various functions for loading a Program into main memory; execute it; and after execution, terminate it.

2) **Program Development**

The OS provides a variety of facilities and services such as editors and debuggers to assist the programmer in creating programs. Typically, these services are in the form of utility programs that, while not strictly part of the core OS and are referred to as application program development tools.

3) **Controlled access to files**

For file access, the OS must reflect a detailed understanding of not only the nature of the I/O device but also the structure of the data contained in the files on the storage medium. Further, in the case of a system with multiple users, the OS may provide protection mechanism to control access to the files.

4) **Access to I/O devices**

A running program needs I/O operations for reading input data, and for outputting the results. This I/O may be with a file or with a device. As user can not control I/O devices directly for security reasons, OS provides services for I/O operations.

5) **Error detection**

Error may be in user programs, CPU and memory hardware, or in I/O devices. Operating System detects such errors and makes from them. It also provides some error recovery mechanism.

- ***System Point of View***

6) **Resource Allocation**

When multiple users are sharing the same machine, or when multiple jobs are running simultaneously, there is a need of fair allocation of resources among them. Operating system does this.

7) **Accounting**

Accounting is the process of keeping information about which user uses which resources, and for what duration of time. Such information can be used to bill the users in multi-user environment, or to get usage statistics to make future planning.

8) **Protection**

Operating system ensures that all access to system resources is controlled. Also security from outsiders is important.

1.3 Single User & Multi User Operating System

- **Single-user operating system:**

A single task system is developed for use with a computer or electronic device that will only run one application at a time. This type of OS is typically used on devices such as wireless phones and two-way messaging devices. A single task, single-user operating system can only run one program or application at a time, and so is not as useful for a computer or other device intended to run multiple programs at once. The Palm OS for Palm handheld computers is a good example of a modern single-user, single-task operating system.

- **Multi user operating system:**

It allows multiple users to use the resources on a single computer simultaneously or at different times. It allows many different users to take advantage of the computer's resources simultaneously. Some operating systems permit hundreds or even thousands of concurrent users. Examples of multi user operating systems are Linux, UNIX, Windows 2000, VMS and mainframe operating systems etc…

- **What is the difference between a single-user and a multi-User Operating System?**

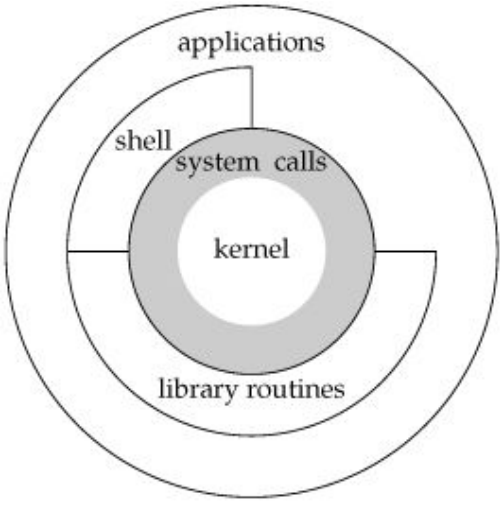
Single User OS	Multi User OS
One user uses the computer system at a time.	Multiple users use the computer system at a time by sharing resources.
Ex: DOS, WINDOWS 3X, WINDOWS 95/97/98 etc.	Ex: UNIX, LYNIX, WINDOWS VISTA etc.
Concept of individual computer.	Concept of client and server. One main computer which is called server and all other users have individual computers or terminals to work with server.

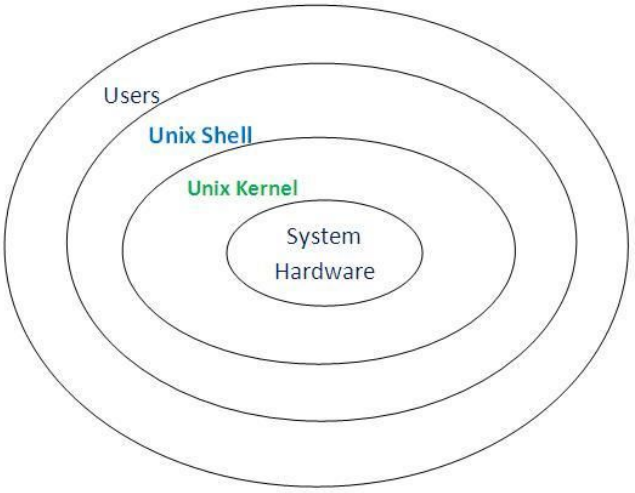
1.4 Elements of an Operating System

Users: System Administrators or anyone who uses the system

Unix Shell: It is a program that interprets commands and acts as interface between Users & Kernel

Unix Kernel: The kernel in an OS is the code and it acts as bridge between the activities by user and the hardware





Kernel:

- The kernel is the core of the UNIX operating system.
- UNIX uses microkernel approach, where code from the kernel in moved up in higher layers to keep it as small (thin) as possible.
- Kernel is a program, which is loaded in memory when system is turned on. It stays there and provides various services until the system is turned off.
- Kernel interacts with the hardware directly. When user program needs to use any hardware, it has to use services provided by the kernel. Special functions, called system calls, are used to request kernel, kernel performs the job on behalf of the user process.
- In addition to providing services to user programs, kernel also provides other services like process management, memory management, file system management so on.
- In short, kernel manages entire computer system.

Shell:

- The shell is an interface between the user program and the kernel.
- When user logs-in to the system, process for shell starts execution. It terminates when user logs-out from the system.
- Users can directly interact with the shell.
- It works as a command interpreter. It accepts commands from user and translates them into the form, which the kernel can understand easily.
- It is also a programming language. It provides various programming functionalities such as looping, branching and so on...

System Call:

- System calls are special functions.
- They are used to request kernel to provide various services, such as reading from a file stored on hard disk.
- They can be invoked via library procedures, or via commands provided by shell, or even directly from C programs in UNIX.
- System calls are similar to user-defined functions. Difference is that they execute in the kernel mode, having fully access to all the hardware; while user defined functions execute in user mode, having no direct access to the hardware.
- Various flavours of UNIX have one thing in common. They all use the same set of system calls provided by POSIX standard. If any operating system is using their system calls, then it will not be a UNIX operating system.

- **System call process:**
  - ➢ When user executes function then library procedure executes TRAP to change the User mode to Kernel mode.
  - ➢ Library procedure issues respective system calls.
  - ➢ System calls executes within kernel, can directly deal with hardware and generates output.
  - ➢ Output is given to the Library procedure.
  - ➢ Again TRAP is used to change the Kernel mode to User mode.
  - ➢ Library procedure gives the respective output to the user.

1.5 Functions of an OS?

1) Process Management:

It is important to note that a process is not a program. A process is only ONE instant of a program in execution.

There are many processes can be running the same program. The 5 major activities of an O.S in regard to process management are:

1. Creation and deletion of user and system processes.
2. Suspension and resumption of processes
3. A mechanism for process synchronization
4. A mechanism for process communication
5. A mechanism for deadlock handling.

2) Memory Management:

Memory management is the act of managing computer memory. The major activities of an operating in regard to memory-management are:

1. Keep track of which part of memory are currently being used and by whom.
2. Decide which processes are loaded into memory when memory space becomes available.
3. Allocate and reallocate memory space as needed.

3) File Management:

The five main major activities of an operating system in regard to file management are:

1. The creation and deletion of files and directory.
2. The support of primitives for manipulating files and directory.
3. The mapping of files onto secondary storage.
4. The backup of files on stable storage media.

4) Device Management:

As a device manager it allows user to view and control the hardware attached to the computer. When a piece of hardware is not working, it is highlighted for the user to deal with. The list of hardware can be sorted by various criteria. It handles the device cache, buffers and interrupts.

5) Security Management:

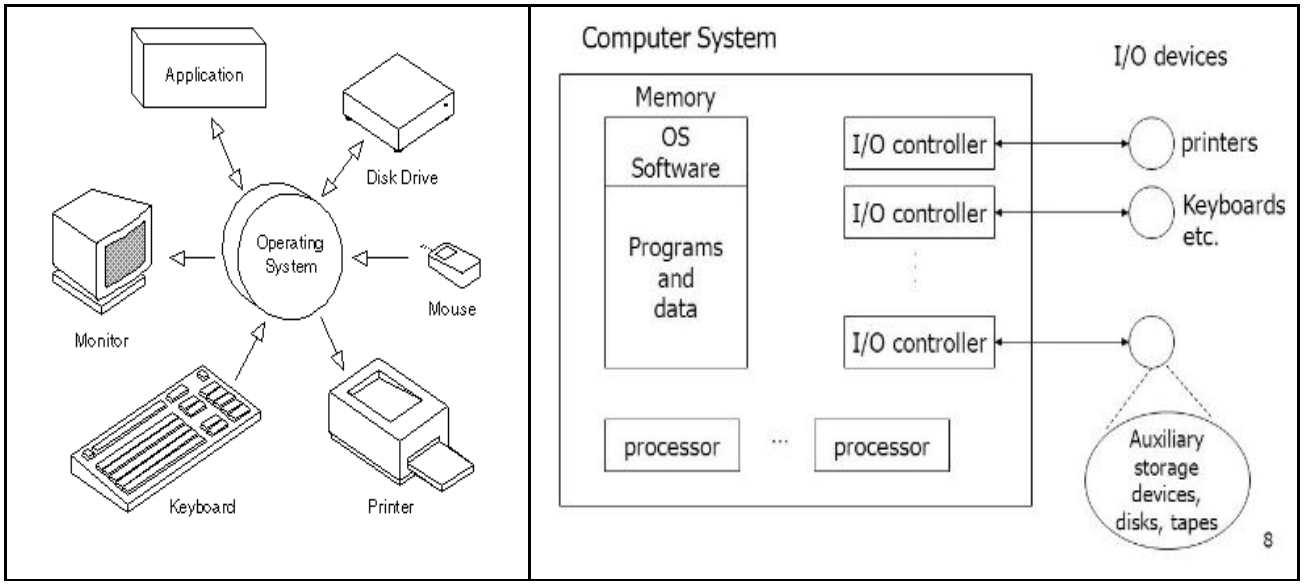
If a computer system has multiple users and allows the concurrent execution of multiple processes, then the various processes must be protected from one another's activities. Security refers to mechanism for controlling the access of programs, processes or uses to the resources defined by a computer system.

6) User Interface:

1. Operating system provides the interface between the user and hardware.
2. The user interface is the layer that actually interacts with the computer operator.
3. The interface consists of a set of commands or menus through which a user communicates with a program.

Operating System as Resource Manager:

- Computer System is the collection of multiple resources like Processor, input devices, Output Devices, storage devices etc.
- Internally the resources are also not able to work with each other. For this they need instructions which are given by OS.
- OS is responsible for Data Processing, Data Movement and Data Store.
- Following figures shown this concept.



- When user wants to access any program or application, then using GUI or CUI interface, he can execute the program.

- The operating system manages the resources like:
  - ➢ **Storage Device:**
    - OS manages the data communication between storage devices.
    - It gives respective instructions to transfer the data from Secondary storage devices to main memory for execution.
    - After the completion of task, it removes the data from main memory and store these information to its original place on secondary device.

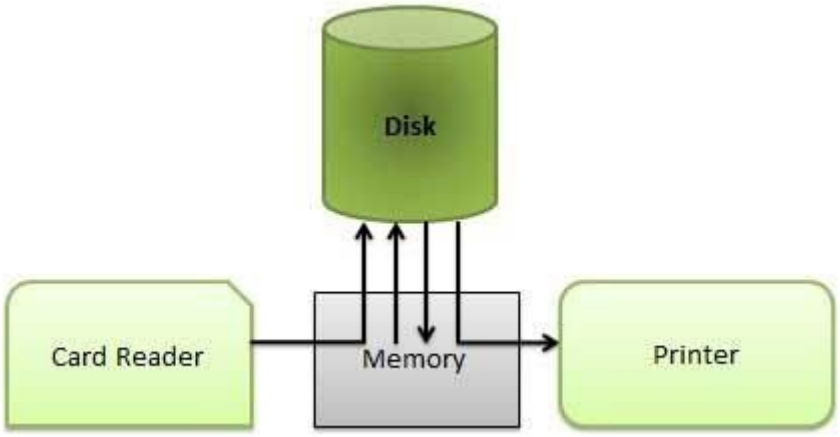
  - ➢ **Processor:**
    - OS is responsible to assign job to the processor.
    - It takes care about only one process remains there with processor for execution.
    - It handles the process scheduling techniques based on which it assigns job to CPU.
    - It tries to take maximum utilization of CPU.
    - Processor executes only those processes which are assigned by the OS. If OS doesn't assign any process to CPU then it remains free for that time.

  - ➢ **Input Devices:**
    - During program execution, when needs to take input from user then OS instruct the related Input Device for taking input from user.
    - When user gives inputs, instantly OS gives related instructions and transfers the respective data to the computer system for further process.

  - ➢ **Output Devices:**
    - During program execution, when system needs to give output to the user then OS instructs the output devices to display results.
    - OS sends the require information to the output device for printing.
    - It also maintains the print queue in which multiple results are stored for printing in First In First Out based.

Spooling:

- Spooling means Simultaneous Peripheral Operating On Line.
- Spooling is an acronym for simultaneous peripheral operations on line.
- Spooling refers to putting data of various I/O jobs in a buffer.
- This buffer is a special area in memory or hard disk which is accessible to I/O devices.



An operating system does the following activities related to distributed environment –

- Handles I/O device data spooling as devices have different data access rates.
- Maintains the spooling buffer which provides a waiting station where data can rest while the slower device catches up.
- Maintains parallel computation because of spooling process as a computer can perform I/O in parallel fashion. It becomes possible to have the computer read data from a tape, write data to disk and to write out to a tape printer while it is doing its computing task.

Example:

With printer device, when user gives print command on file then the file store in buffer and all stored files are printed by printer is FIFO manner.

Advantages:

- The spooling operation uses a disk as a very large buffer.
- Spooling is capable of overlapping I/O operation for one job with processor operations for another job.
- It reduces the idle time
- It reduces the overlapping I/O and CPU.

Disadvantage:

- Data needs to read from buffer every time before processing. CPU remains idle during this activity.