Operating System Concepts

Introduction to operating System

- ❖ Definition of OS
- ❖ Structure of OS
- ❖ Objective of OS
- ❖ Operating System Functions
- ❖ Characteristics of operating System

Operating System

- An *operating system* acts as an intermediary between the user of a computer and the computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs in a *convenient* and *efficient* manner.
- An OS is a program that manages the computer hardware. It also provides a basis for application programs and acts as an intermediary between the computer user and the computer hardware.
- "An operating system (Os) is a set of program that manages computer hardware resources and provides common services for application software".
- "An operating System (Os) is a set of programs that provide interface between user and computer, creates user friendly environment".
- E.g. MAC OS, Microsoft windows 95,98 windows NT/ME/2000, Windows Xp, Windows vista, windows 7,8,8.1,10,unix Linux
- "An Operating system works as a mediator between hardware and user, for the efficient use of hardware and create user friendly environment".
- An operating system is a program designed to run other program on computers.
- Os is a collection of program and utilities.
- It acts as the interface between computer user and computer hardware.
- The operating system is most important type of system software in a computer system.
- A user cannot run application program on the computer without an operating system.

Structure Of Operating System

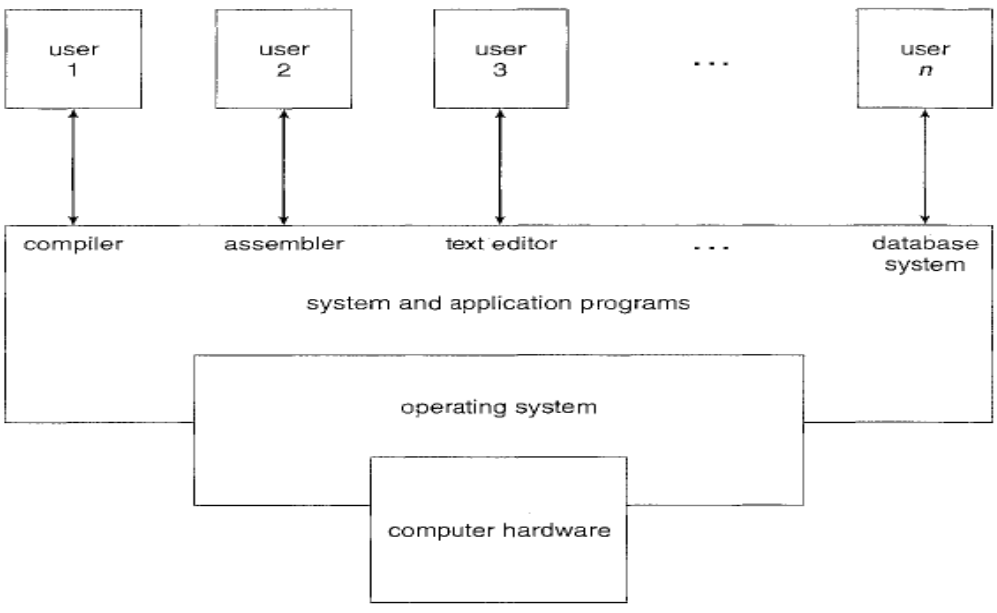


Figure 1.1 Abstract view of the components of a computer system.

- **The User**
  - Users are at the top. Users interact with the system by using application programs to perform particular task
  - Users convenience is operating systems main objective.
- **The application program**
  - It performs a particular task. They use different kinds of functionalities provided by Os to perform their task.
  - It consist compilers, assemblers ,linkers etc depend on the user
- **The Operating System**
  - The next level is for the operating system. It manages all the underlying hardware.
  - It includes process managements routines, Memory management routines,I/O control routines, file management routines etc.
  - In simple terms an interface between application program and the computer hardware
- **The Hardware**
  - It Comes at lower level, It contains various types of physical devices.
  - Eg. CPU, Main Memory, I/o Devices, Secondary Storage etc..

Objective of Operating System

- **Convenience**
  - Make computer user friendly.
  - Make a computer more convenient to use
- **Efficiency**
  - Allows computer to use resources efficiently
- **Ability to grow**
  - Constructed in a way to permit effective development, testing and introduction of new function without interfering with services.
- **To hide the details of the hardware from the user**
- **To act as an intermediary between the hardware and its user and making it easier for the user to access and use other resources.**
- **To manage the computer Resources**

Functions of Operating System

- Program Creation
- Program Execution
- Input/output operations
- Error detection
- Resource allocation
- Accounting resources
- security

- **Program Creation**
  - The operating system provides editors, debuggers, to assist the programmer in creating programs.
- **Program Execution**
  - A number of tasks required to execute a program, the task includes instruction and data must be loaded in main memory for execution, I/o operating system devices, files must be initialized, and other resources must be prepared, the os handles these tasks for the user.
  - The system must be able to load a program into memory and to run that program. The program must be able to end its execution, either normally or abnormally (indicating error).
- **Input/output operations**
  - A running program may require I/0, which may involve a file or an I/0 device.
  - For specific devices, special functions may be desired (such as recording to a CD or DVD drive or blanking a display screen).
  - For efficiency and protection, users usually cannot control I/0 devices directly.
  - The Os handles the task.
- **Error detection**
  - The operating system needs to be constantly aware of possible errors.
  - Errors may occur in the CPU and memory hardware (such as a memory error or a power failure), in I/0 devices (such as a parity error on tape, a connection failure on a network, or lack of paper in the printer), and in the user program (such as an arithmetic overflow, an attempt to access an illegal memory location, or a too-great use of CPU time).
  - For each type of error, the operating system should take the appropriate action to ensure correct and consistent computing.
- **Resource allocation**
  - When there are Multiple users or multiple jobs running at the same time, resources must be allocated to each of them.
  - Many different -types of resources are managed by the operating system
  - Some (such as CPU cycles, main memory, and file storage) may have special allocation code, whereas others (such as I/0 devices) may have much more general request and release code.
- **Accounting resources**
  - we want to keep track of which users use how much and what kinds of computer resources.
  - This record keeping may be used for accounting (so that users can be billed) or simply for accumulating usage statistics.
- **Security**
  - The owners of information stored in a multiuser or networked computer system may want to control use of that information.

- Protection involves ensuring that all access to system resources is controlled.
- Security of the system from outsiders is also important.

Characteristics of Operating System

- **Memory Management**
  - Keeps tracks of primary memory i.e. what part of it is in use by whom, what part is not in use etc. and allocates the memory when a process or program requests.

- **Process Management**
  - Allocates the processor (CPU) to a process and De-allocates processor when it is no longer required.

- **Device Management**
  - Keeps track of all devices. This is also called i/o controller that decides which process gets the device, when, and for how much time.

- **File Management**
  - Allocate and de-allocates the resources and decides who gets the resources
- **Security**
  - Prevents unauthorized access to programs and data by means of password and similar other techniques.
- **Job Accounting**
  - Keeps track of time and resources used by various jobs and/or user.
- **Control over system performance**
  - Records delays between request for a service and from the system.
- **Interaction with operators**
  - The interaction may take place via the console of the computer in the form of instructions.
  - Operating system acknowledge the same, does the corresponding action and informs the operation by a display screen
- **Error detecting aids**
  - Production of dumps, traces error messages and other debugging and error detecting methods
- **Co-ordination between other software and user**
  - Coordination and assignment of compilers, interpreters, assemblers and other software to the various user of the computer system.

Evolution of Operating System

- **hardware upgrades or new types of hardware:**

  - With hardware technologies development, the OS also needs to upgrade so as to utilize the new mechanisms introduced by new hardware. For example, Pentium IV extended instruction set of Pentium III for multimedia applications and internet transmission. An OS designed for the previous versions of Intel x86 series will have to be upgraded to be able to accommodate these new instructions.

- **New services:**
    - An OS may also expand to include more services in response to user demand.
- **Fixes:**
    - No software is perfect, and any program may contain more or less bugs or defects, thus fixes should be made from time to time. Microsoft Windows is a vivid example of this kind.
    - These situations all require OS designers to build an OS in the way that the system can be maintained and upgraded easily.

Evolution of Operating System

- Serial processing
- Batch Operating System
- Multiprogramming operating System
- Time Sharing Operating system /Multitasking OS
- Real time operating System
- Multi-user Operating System
- Multiprocessing operating system/parallel system or tightly coupled
- Distributed operating system
- Client-server operating system
- Clustered system
- Hand held operating system
- Network operating System

Serial Processing

- The earliest computer, from the late 1940s to the mid-1950s, the programmer interacted directly with the computer hardware; there was no operating system.
- These machine were run from a console, consisting of display lights, toggle switches, some form of input devices, and a printer.
- Programs in machine code were loaded via the input device(e.g. card reader)
- If error halted the program the error condition was indicated by the lights.
- If the program proceed to a normal completion, the output appeared on the printer.

- **Serial Processing presented two problems**
- **Scheduling**
  - if a user sign up for an hour and finish in 45 min this would result in wasted of computer idle time.
  - On the other hand , the user might run into problems, not finishing in the allotted time, and be forced to stop before resolving the program.
- **Setup time**
  - A single program called a job, could involve loading the compiler plus the high level language program into memory, saving the compiled program and then loading and linking together the object program and common function.
  - if an error occurred, the hapless user typically had to go to beginning of the setup sequence.
- **Batch systems**
  - Early machines were expensive, and therefore it was important to maximize machine utilization. The wasted time due to scheduling and setup time was unacceptable.
  - The common input device were card readers and tape drivers.
  - The common output devices were line printer, tape drivers and punch cards.
  - The user of such system did not directly interact with the computer systems.
  - The jobs were in form of punch card.

- The os major task was to transfer control automatically from one job to another
- The Os always resides in memory.
- To speed up processing jobs with similar needs were batched together.
- Programmer would prepare a job and submit it to the operator.
- Operator would sort them in batches with similar requirement, and as computer became available run them batch wise.
- At some later time, output appeared. The output consisted of the result of the program and error information. Programmer need to wait at this time and then collect the output from the operator.
- The output from each job would be sent back to the appropriate programmer by the operator.
- Here, memory is divided in two parts as given in figure; it is shared between the Os and the job.

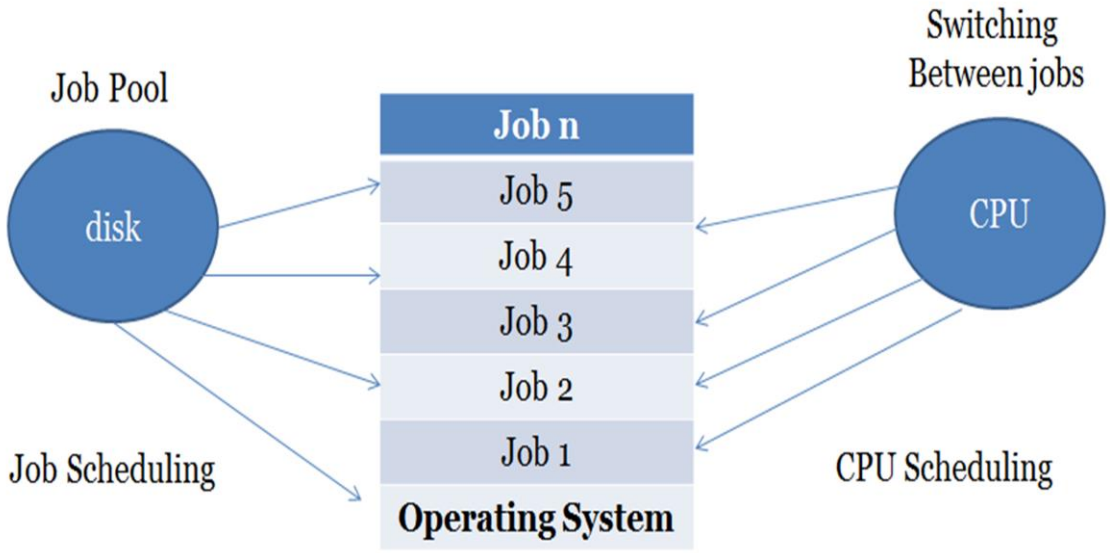
- At a time one job is selected out of the batch of jobs, and is loaded into memory for execution.
- Once its execution completes, another job will be selected and loaded into memory for execution. This process will continue untill all jobs in a batch get executed.

- **Turnaround time:-**
  - The delay between job submission and job completion called as turnaround time.
- **Disadvantage**
  - Turnaround time can be large from user standpoint
  - Difficult to debug program
  - Low throughput: because of, Cpu remains idle when I/O is going on
  - Programmers do not have direct interaction with job

- Debugging is possible only offline, after output appears
- Operation were too much time consuming
- A job could corrupt the monitors, thus affecting pending jobs
- Due to lack of protection scheme, one batch job can affect pending jobs.

- **Multiprogramming operating System**
  - Several jobs are kept in main memory at the same time, and the cpu is multiplexed among them which require memory management and protection.
  - The Os picks and begins to execute one job from memory. Once this job needs an operation the Os switches to another job(Cpu or Os is always busy).
  - A multiprogramming Os provides the ability to run more than one program concurrently by a single processor.
  - When two or more programs are in memory at the same time, sharing the processor is referred to the multiprogramming Os.
  - In multiprogramming number of process are residing in main memory at a time.

- Diagram:-



- The number of jobs in memory is less than the number of jobs in disk(job pool).

- If several jobs are ready to be brought into memory and there is not enough room for all of them, then system chooses job among them (job scheduling).
- If several jobs are ready to run at the same time, the system must choose among them (CPU Scheduling).

Implementation of Multiprogramming Os

- **Multiprogramming can be implemented as;-**
- Non Preemptive
- Preemtive
- **Non pre-emptive**
  - A program is allowed to execute until it voluntarily gives up the CPU
  - A program voluntarily gives up the CPU when it waits for some event, such as I/O operation, or when it terminates.
  - Once a CPU becomes free, it can be allocated to some other program.
- **pre-emptive**
  - A program is allowed to execute only for some maximum time duration.
  - After this time duration, A CPU is taken away from the program.
  - Hence one smaller program need not to wait for other large program to finish its execution.
- **Disadvantages**
  - User Cannot interact with their jobs, while executing.
  - A programmer cannot modify a program as it executes to study its behaviour.
- **Advantages**
  - Can give effective memory utilization
  - CPU is never idle, so the performance of CPU will increase
  - Throughput of the CPU may also increase.
  - Waiting time is limited for processes.

- **Time Sharing Operating system /Multitasking OS**
    - Logical extension of multiprogramming.
    - The CPU executes multiple jobs by switching among them, but the switches occur so frequently that the user can interact with each program while it is running.
    - An interactive computer system provides direct communication between the user and system.
    - A time shared Os allows many user to share the computer simultaneously.
    - A time shared Os uses CPU scheduling and multiprogramming to provide each user with small portion of a time-shared computer.
    - Each user has at least one separate program in memory.
    - A program loaded into memory and execution is commonly referred to as process.
    - Time sharing Os are even more complex than multi programmed Os.
    - The time slot is defined by the Os.
    - Time sharing system can run several programs at the same time, so it is also multiprogramming system. But multiprogramming is not a time sharing system.
    - The idea of time sharing was demonstrated as early as 1960 but since time shared systems are difficult and expensive to build they did not become common until the early 1970s
    - E.g. CTSS, Multics, Cal, Unix
- **Advantages**
    - Efficient CPU utilization
    - User can interact with the job when it is executing, that was not possible in batch Os.
- **Real time Operating System**
    - A System is said to be real time if it is required to complete it's work & deliver it's services on time.

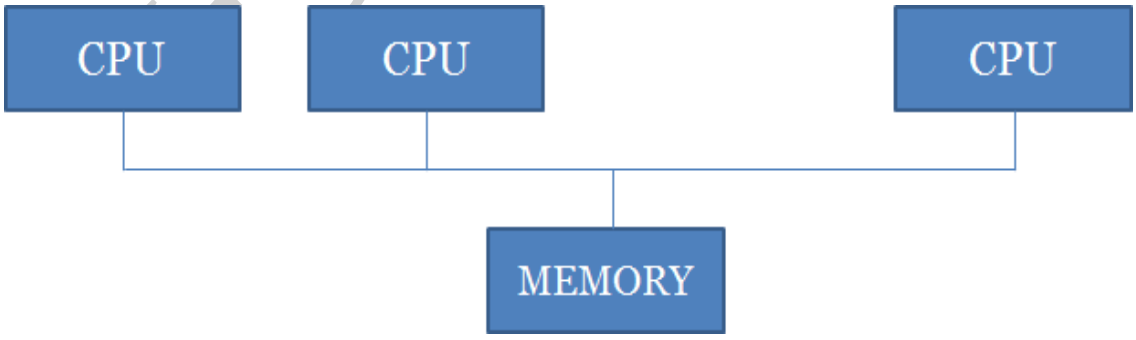
- A real time system is used when rigid time reqirements have been placed on the operation of processor or the flow of data;
- A real time system has well defined, fixed time constraints
- Process must be done with in the defined constraint, or the system will fail.
- Real time Os is one that must react to inputs and respond to them quickly. A real time system cannot afford to be late with a response to an event.
- E.g. flight control system; all tasks in that system must execute on time.
- **Hard Real time**
    - All critical task must get completed strictly within the specified time limits.
    - Tasks are guaranteed to occur in time.
    - This goal requires all delays in the system be bounded, from the retrieval of stored data to the time that it takes the operating system to finish any request made of it.
- **Soft Real time**
    - It is less restrictive type, missing an occasional dead line is acceptable.
    - A critical real time task gets priority over other task, and retains that priority until it completes.

Real Time Application	Examples
Detection	Radar System, Burglar System
Process monitoring and control	Petroleum refinery, paper mill
Communication	Telephone switching system
Flight simulation and control	Auto pilot shuttle mission simulator
Transportation	Traffic light system, Air traffic Control

- **Multi-user Operating System**
  - Single user Os allow single user to access computer system i.e. multiple user can't use single at a time.
  - Contrast to this, multi-user Os allow multiple user to access computer system at a same time.
  - Access to computer system is normally provided via network, so that user to access computer remotely using terminal or other computer.
  - A terminal contains only I/O devices such as keyboard and monitor. It is used to provide interaction between user and computer system
  - A touch screen help desk provided at railway station or an ATM machine provided at a bank is an example of such kind of terminal.
  - This can be implemented through time sharing and multiprogramming.
  - CPU is time sliced at a regular intervals among various users.
  - This kind of Os is much more complex then single user operating system.
- **Advantage**
  - Expensive hardware can be shared among several users using multi user operating system makes better utilization.
- **Disadvantage**
  - As more user access it, the performance becomes lower and slower.
  - Cost of hardware and software more expensive than single user Os.

Multiprocessing operating system/parallel system or tightly coupled

- Multiprocessing system have more than one processor in close communication, sharing the computer bus, clock, and some times memory and peripherals devices.
- All program running in parallel manner.
- Multiprogramming operating system allows more than one program to run concurrently on a uni-processes system. But a single processor can't execute more than one instruction at a time.
- **Multiprogramming is of two types**
    - Symmetric multiprocessing
    - Asymmetric multiprocessing
- **Symmetric multiprocessing**
    - Each processor runs an identical copy of operating system and they communicate with one another as needed.
    - All the CPU Shared the common memory.
    - Many processes run at once
    - SMP means that all processors are peer; no master slave Relationship exist between processors.
    - Each processor concurrently runs a copy of the operating system.
    - Virtually all modern operating systems support SMP, including Windows XP, Windows 2000, Solaris, Linux, and Mac OS X



- **Asymmetric multiprocessing**
    - Each processor has an assigned task

- Master processor controls the system schedule and allocate work for slave processors
- ASMP uses Master slave relationship.
- Master performs I/O and computations
- Only master may execute the Os. Slave can execute only user programs

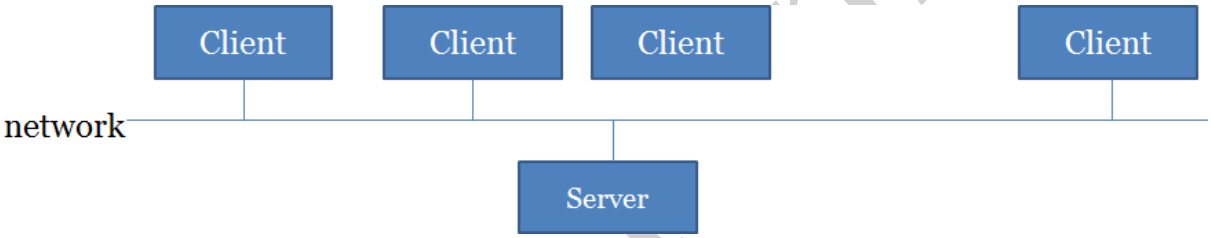
- **Features of multiprocessing Os**
  - If one processor fails, then the another processor should retrieve the interrupted process state so that execution of the process run continue.
  - The processor should support efficient context switching operation
  - Multiprocessor system support large physical address space and large virtual address space.
  - The IPC mechanism should be provided and implemented in hardware as it becomes efficient and easy.

- **Advantage of Multiprocessor**
  - Increased throughput
  - Economy of scale
  - Increased reliability

- **Distributed Os**
  - Distributed system depends on networking for their function ability.
  - By being able to communicate, distributed systems are able to share computational task.
  - Distributed Os Distribute the computation among several physical processors.
  - When there is higher load on one processor, this Os automatically distributes the computation on other machine, where the processor is idle or having low load.

- **Advantage of Multiprocessor**
  - Resource Sharing
  - Higher Reliability
  - Better price performance ration
  - Incremental growth

- **Client-server operating system**
    - Centralized system today act as server system to satisfy requests generated by client system.
    - Server system can be broadly categorized as compute server and file server.
- **Computer-server system**
    - Provides an interface to which client can send requests to perform an action, in response to which they execute the action and send back result to the client.
- **File-Server System**
    - Provide a file system interface where clients can create update, read, and delete files.



- **Peer to peer System**
    - Each computer may act as client or server depending on request & respond.
    - Service provided by several nodes distributed on network.
    - These systems are usually referred as loosely coupled system
- **Clustered system**
    - Clustered system gather together multiple CPUs to accomplish computational work.
    - Clustering allows two or more system to share storage
    - Provides high reliability
    - **Asymmetric clustering:-** one server runs the application while other servers are at standby
    - **Symmetric clustering:-** all N host are running the application or applications
    - Like multiprocessor system, clustered system gather multiple CPUs to accomplish computational work

- Clustered system differ from multiprocessor system , how ever, in that they are composed two or more individual system coupled together.

- **Asymmetric mode**
  - In asymmetric clustering, one machine is in hot-standby mode while the other is running the applications
  - The hot standby host machine does nothing but monitor the active server
  - If that server fails, the hot-standby host becomes active server.
- **Symmetric mode**
  - In symmetric mode, two or more hosts are running applications, and are monitoring each other
  - This mode is obviously more efficient, as it uses all of the available hardware
  - It does require that more than one application be available to run.
- **Handheld System**
  - **Personal Digital Assistants (PDAs)**
  - **Cellular Telephones**
  - **Issues:**
    - **Limited memory**
    - **Slow processors**
    - **Small display Screens**
- Handheld System include personal digital assistants (PDAs), Such as palm and pocket-Pcs, and cellular telephones, many of which use special-purpose embedded operating system.
- Developers of handheld system and applications face many challenges, most of which are due to the limited size of such devices.
- Because of their size, most handheld devices have a small amount of memory, show processors, and small display screens

Needs of Operating system

- Management of the process
- Management of Random Access Memory
- Management of Input/output
- Management of execution of Application
- Management of Authorization
- File management
- Information Management
- Convenient & efficient use of computer hardware
- **Management of the Processor**
    - The operating system is responsible for managing allocation of the processor between the different programs using a scheduling algorithm.
    - The type of scheduler is totally dependent on the operating system, according to desired objective.
- **Management of Random Access Memory**
    - The operating system is responsible for managing the memory space allocated to each application and where relevant, to each user.
    - If there is insufficient physical memory, the operating system can create a memory zone on the hard drive known as “Virtual memory”.
    - The virtual memory lets you run applications requiring more memory than there is available RAM on the System.
- **Management of Input/ Output**
    - The operating system allows unification and control of access of programs to material resources via drivers (also known as peripheral administrators or input/ output administrators).
- **Management of execution of application**
    - The operating system is responsible for smooth execution of application by allocating the resources required for them to operate.
    - This means an application that is not responding correctly can be “killed”.

- **Management of authorization**
    - The operating system is responsible for security relating to execution of programs by guaranteeing that the resources are used only by programs and user with the relevant authorization.
- **File management**
    - The operating system manages reading and writing in the file system and the user and application fill access authorizations.
- **Information Management**
    - The operating system provides a certain number of indicators that can be used to diagnose the correct operation of the machine.
- **Elements of an Operating System**
    - Kernel
    - Shell
    - File System
    - Memory management
- **Kernel**
    - The kernel is heart of the operating system.
    - The kernel, which represents the operating system's basic function such as management of memory, processes, files, main inputs/ outputs and communication functionalities.
    - Among its responsibilities are ensuring that each running process is given a fair amount of time to execute while a controlling the amount of resources each process can use.
- **Shell**
    - The Shell, allowing communication with the operating system Via a control language letting the user control the peripherals without knowing the characteristics of the hardware used, management of physical addresses, etc.
    - A shell is a user interface for access to an operating system's service.
    - In General, OS shell use either a command line interface(CLI) Or Graphical user interface(GUI)

- **File System**
  - The file system , allowing files to be recorded in a tree structure.
- **Memory management**
  - The name of this layer gives you a good idea what it is all about.
  - It is the responsibility of this layer to share your computers physical memory among the processes which want to use it.
  - It also has to manage such situation where there may not be enough physical memory to share out.
- **Operating System as Resource manager**
  - Resource – "Something Valuable". E.g. The Os is responsible for managing these resources. CPU, Memory Space(RAM), File Storage Space, I/O Devices.
  - A computer is a set of resources for the movement, storage and processing of data and for control of these function
  - A portion of Os is in main memory. This includes The kernel, which contains most frequently used function in Os
  - The allocation of this resource is controlled jointly by the Os and memory management hardware in the processor.
  - Os directs the processor in the use of the other system resources and in the timing of its execution of the other program.
  - A portion of Os is in main memory. This includes the kernel which contains the most frequently used function in the Os and at a given time, other portions of the Os currently in use.
- **Operation performed by resource manger**
  - Resource allocation– Os allocates resources to multiple user and multiple jobs running at same time.
  - Operation control program:- Os is a program that controls the execution of user programs and operations of I/O devices.
  - System Access:- Os ensures that all access to resource is protected, including authorization etc.
  - Accounting and usage Statistics:- Monitoring System Data and Resource usage.

- **The Os function in two ways:**
    - The Os Function in the same way as ordinary computer software , that is, it is program or suit of program executed by the processor.
    - The Os frequently relinquishes control and must depend on the processor to allow it to regain control.

History of Operating System

Generation	Year	Electronic Devices Used	Types of Os and Devices
First	1945-55	Vacuum Tubes	Plug boards
Second	1955-65	Transistors	Batch System
Third	1965-80	Integrated Circuit(IC)	Multiprogramming
Fourth	Since 1980	Large Scale Integration	PC

- **First Generation**
    - The earliest electronic digital computers had no operating systems. Machines of the time were so primitive that programs were often entered one bit at time on rows of mechanical switches.
    - Programming languages were unknown.
- **Second Generation**
    - By the early 1950's the routine had improved somewhat with the introduction of punch cards. The General motors research laboratories implemented the first operating system in early 1950's for IBM701.
    - The system of the 50's generally run one job at a time. These were known as single stream batch operating system
- **Third Generation**
    - The system of 1960's were also batch processing systems, but they were able to take better advantages of the computer's resource by running several jobs at once. So operating systems designers developed the concept of multiprogramming in which several jobs are in main memory at once
    - Operating system designers developed the concept of multiprogramming in which several jobs are in main memory at once; a processor is switched from one job to another job as needed to keep several jobs advancing while keeping the peripheral devices in use

- A major featured in third generation Os was the technique called Spooling(**simultaneous peripheral operation on line**). In spooling a high speed device like a disk interposed between a running program and a low speed device involved with the program in input/ output.
- Instead of writing directly to a printer outputs were written to a disk.
- Programs can run to completion faster, and other programs can be initiated sooner when the printer becomes available, the outputs may be printed
- Spooling technique is much like thread being spun to a spool so that it may be later be unwound as needed.

- **Fourth generation**
  - With the development of LSI (Large Scale integration) Circuits, Chips, OS entered in the system entered in the personal computer and the workstation age.
  - Microprocessor technology evolved to the point that it become possible to built desktop computers as powerful as the mainframes of the 1970s.
  - Two operating systems have dominated the personal computer scene: MS-Dos, Written by Microsoft, Inc. for the IBM PC and other machines using the Intel 8088 CPU and its successors, and UNIX, which is dominant on the large personal computers using the Motorola 6899 CPU family.