Unit 1 Introduction to Operating System

Definition of OS: - A program that acts as an intermediary between a user of a computer and the computer hardware.

An operating system (commonly abbreviated OS and O/S) is the software component of a computer system that is responsible for the management and coordination of activities and the sharing of the limited resources of the computer.

Operating system goals:

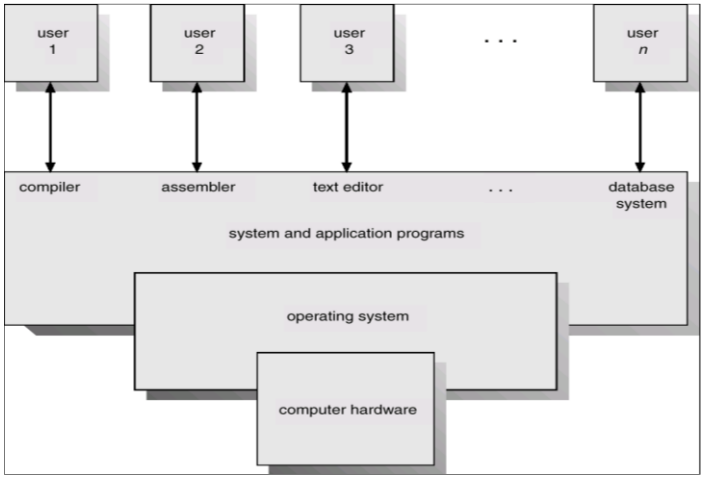
i. Execute user programs and make solving user problems easier.
   - Make the computer system convenient to use.
   - Use the computer hardware in an efficient manner.

i. Resource allocator – manages and allocates resources.

ii. Control program – controls the execution of user programs and operations of I/O devices .

iii. Kernel – the one program running at all times (all else being application programs).

Computer System Components:



1. Hardware – provides basic computing resources (CPU, memory, I/O devices).
2. Operating system – controls and coordinates the use of the hardware among the various application programs for the various users.
3. Applications programs – define the ways in which the system resources are used to solve the computing problems of the users (compilers, database systems, video games, business programs).
4. Users (people, machines, other computers).

What OS does?

An operating system performs basic tasks such as,
1.controlling and allocating memory,
2.prioritizing system requests,
3.controlling input and output devices,
4.facilitating networking and
5.Managing file systems.

Characteristics of Operating system:

1. Whether multiple programs can run on it simultaneously i.e. multi-tasking.
2. Whether it can take advantage of multiple processors i.e. multi-processing.
3. Whether multiple users can run program on it simultaneously i.e. multi-user.
4. Whether it can reliable prevent application programs from directly accessing hardware devices i.e. protected.
5. Whether it has built-in support for graphics.
6. Whether it has built-in support for networks.

Basic function of operating system:

1. A task scheduler: The task scheduler is able to allocate the execution of the CPU to a number of different tasks. Some of those are the different applications that are the user is running and some of them are operating system tasks. The tasks scheduler is the part of the operating system that lets you

print a document from your word processor in one window while you are downloading a file in another window and recalculating a spreadsheet is a third window.

2. A memory manager: The memory manager controls the system's Ram and normally creates a larger virtual memory space using a file on the hard disk.
3. A disk manger: The disk manger creates and maintains the directories and files on the disk. When you request a file, the disk manger brings it in from the disk.
4. A network manager: The network manger controls all the data moving between the computer and the network.
5. Other I/O Services manager: The operating system manages the keyboard, mouse, video, display, printers etc.
6. Security manager: The operating system maintains the security of the information in the computer's files and controls that can access the computer.

Desires of an Operating system:

1. **Usability:**
    a. Robustness: OS accept all valid inputs without error, and gracefully handle all invalid inputs. OS would not be crashed in any circumstances, and could be recovered in the case that we suddenly remove hardware while they are running, or loss of power supplied
    b. Consistency: For example. If "-" means options flags in one place, it means it in another. The key idea is conventions. We should base on the concept: The Principle of Least Astonishment. This helps us easy to understand and adapt with the new system.
    c. Proportionality: Simple, cheap and frequent things are easy. Also, expensive and disastrous things are hard.
    d. Forgiving: Errors can be recovered from. Reasonable error messages. Example from "rm"; UNIX vs TOPS.
    e. Convenient: Not necessary to repeat things, or do awkward procedure to accomplish things. Example copying a file took a batch job.
    f. Powerful: It has high level facilities.
2. **Facilities**:
    a. The system should supply sufficient for intended use.
    b. The facilities is complete, do not leave out any part of a facility.
    c. Appropriate, e.g. do not use fixed-width field input from terminals.
3. **Cost**:
    a. Want low cost and efficient services.
    b. Good algorithms. Make use of space/time tradeoffs, special hardware.
    c. Low overhead. Cost of doing nothing should be low. For example, idle time at a terminal.
    d. Low maintenance cost. System should not require constant attention.
4. **Adaptability**:
    a. Tailored to the environment: Support necessary activities. Do not impose the unnecessary restrictions. What are the things people do most make them easy?
    b. Changeable over time: Adapt as needs and resources change.
    c. Extendible-Extensible: Adding new facilities and features- which look like the old ones.

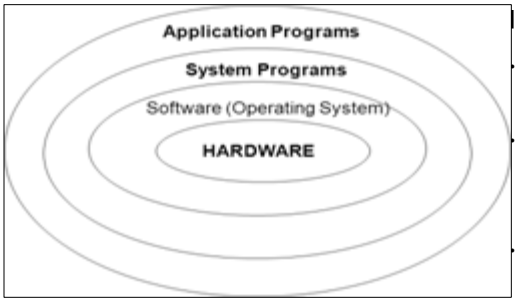
Need Of OS:-

1. The main feature of operating system is to execute multiple programs in interleaved fashion or different time cycle is called multiple programming systems.
2. Some of the important reasons why we need an operating systems are as follows:-
    a. User interacts with the computer through operating system in order to accomplish his/her task since it is his primary interface with a computer.
    b. It helps the user in understand the inner functions of a computer very closely.

c. Many concepts and techniques found in operating system have general applicability in other applications.

3. An operating system is an essential component of a computer system. The primary objectives of an operating system are to make computer system convenient to use and utilizes computer hardware in an efficient manner.
4. An operating system is large collection of software which manages resources of the computer system, such as memory, processor, rite system and input/output devices.
5. Operating system keeps track of the status of each resource and decides who will have a control over computer resources, for how long and when.

Structure of Operating System:



The structure of OS consists of 4 layers:

1. **Hardware**:- Hardware consists of CPU, Main memory, I/O Devices, etc.,
2. **Software (Operating System)**:- Software includes process management routines, memory management routines, I/O control routines, file management routines.
3. **System programs**: - This layer consists of compilers, Assemblers, linker etc.
4. **Application programs**: - This is dependent on users need. Ex. Railway reservation system, Bank database management etc.,

Examples of system:-

1. Before 1950 the programmers directly interact with hardware, there is no operating system at that time.
2. If the programmer wish to execute on those days, the following serial steps are necessary.
   a. Type the program on punched card to card reader.
   b. Submit to the computing machine, if there are any errors, the error condition was indicated by light.
   c. The programmer examines the registers and main memory to identify the cause of the error.
   d. Take the output on the printers.
   e. Then the programmer is ready for the next program.
3. This mode of operating could be termed processing, this type of processing is difficult for users, it takes much time and next program should wait for the completion of the previous one.
4. The programmer are submitted to the machine one after another, therefore this method is said to be serial processing.

A. MS-DOS

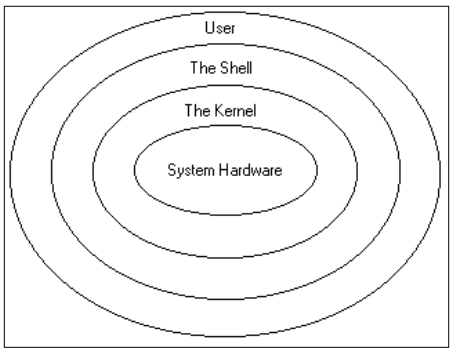
1. When IBM came out with their personal computer in the early 1980's, they also arranged for Microsoft to provide the MS-DOS operating system.
2. MS-DOS was developed for very small personal computers and did not have many of the features expected of operating systems for large computers. But the PC and its clones became much more successful than most people had imagined and MS-DOS has grown to be the most widely used operating system in the world.
3. As the power and capabilities of the PC increased, features were added to MS-DOS to deal with the new hardware. Although MS-DOS is quite primitive by modern operating system standards, it is an open system, (due to the fact that there is no protection at all) and so it could be adapted and modified by add-on programs, MS-DOS has gone through six major versions over the year since the first PC was introduced.
4. The MS-DOS system call interface is modeled after UNIX and has a similar set of system calls. One major difference is that it does not have the UNIX fork or exec system calls, but has a system call more like the create process.

5. One add-on to MS-DOS is the Microsoft Window windowing environment. Windows adds significant functionality to MS-DOS and could be considered an operating system on its own.

B. Unix:

1. The first version of UNIX was developed in 1969 by Ken Thompson of the research group of Bell laboratories to use for idea PDP-7.
2. During the past 25 year the UNIX operating system has evolved into a powerful, flexible and versatile operating system.
3. It serves as the operating system, for all types of computers including single-user personal computers, workstations, microcomputers, minicomputers and super computers as well as special purpose device.
4. The member of computer running by UNIX and more than 100 million people using these systems.
5. This rapid growth is expected to continue. This success of UNIX is due to many factors, including its portability to a wide range of machines, its adaptability and simplicity, the wide range of tasks it can perform, its multi user and multi-tasking nature, and its suitability for networking, etc.
6. The main features of UNIX are:
   a. Ability to support multiuser and multi-tasking.
   b. Portability.
   c. Excellent network environment
   d. Adaptability and simplicity
   e. It provides the better security
   f. Flexible file system.

UNIX Architecture:



1. The kernel of UNIX is the hub of the operating system: it allocates time and memory to programs and handles the file store and communications in response to system calls.
2. As an illustration of the way that the shell and the kernel work together, suppose a user type's rm my file (which has the effect of removing the file **my file**).
3. The shell searches the file store for the file containing the program rm, and then requests the kernel, through system calls, to execute the program rm on my file.
4. When the process rm my file has finished running, the shell then returns the UNIX prompt % to the user, indicating that it is waiting for further commands.
5. Amongst the functions performed by the kernel are:
   a. Managing the machine's memory and allocating it to each process.
   b. Scheduling the work done by the CPU so that the work of each user is carried out as efficiently as is possible.
   c. Organizing the transfer of data from one part of the machine to another.
   d. Accepting instructions from the shell and carrying them out.
   e. Enforcing the access permissions that are in force on the file system.
6. The shell
   a. The shell acts as an interface between the user and the kernel.
   b. When a user logs in, the login program checks the username and password, and then starts another program called the shell. The shell is a command line interpreter (CLI).
   c. It interprets the commands the user types in and arranges for them to be carried out.
   d. The commands are themselves programs: when they terminate, the shell gives the user another prompt (% on our systems).
   e. The user can customize his/her own shell, and users can use different shells on the same machine.
      The shell keeps a list of the commands you have typed in.

f. If you need to repeat a command, use the cursor keys to scroll up and down the list or type history for a list of previous commands.
g. You can use any one of these shells if they are available on your system. And you can switch between the different shells once you have found out if they are available.

C. Linux

1. Linux is the UNIX based operating system originally developed as for Intel-Compatible PC's. It is now available for the most of hardware platform ranging from PDA's to main frames. Linux is "Modern Operating System", meaning it has such features as virtual memory, memory protection and preemptive multitasking.
2. Linux is built and supported by a large international community of developers and users dedicated to free, open source software.
3. This community sees Linux as an alternative to such proprietary systems are windows and Solaris, and a platform for alternatives to such proprietary applications, developer's tools, system utilities, business applications, document publishing tools, network client and server applications.
4. The best part of this community is that all code is open. This means there is no barriers to entry for any given problem; there are generally several applications that solve the problem. These applications can also borrow the best parts from each other to become even better. An excellent example of this is Galeon.
5. Galeon is a web browser which took Mozilla's web page rendering engine and integrated it with a GTK front end.
6. Linux specifically refers to the Linux kernel. However, the kernel is useless without a set of tools and application to run on the kernel Linux is most commonly distributed with this tool set and a collection in what is called a "distribution".
7. The most common are Redhat, Mandrake, Suse and Debain. Distributions differs in three basic ways the process for installing the distribution, the application available and the process for installing and managing these applications.
8. Reasons for installing Linux :
    a. Configurability
    b. Convenience
    c. Stability
    d. Community
    e. Freedom.

D. Window 98:

Microsoft launched an advanced Window in 1995. It had 32 bit file system and could run programs that were in MS-DOS.

E. Windows 98:

An enhanced and highly developed version of Windows 95 was released in1998 which is known as Window 98.

F. Window NT:

Window New Technology is powerful system with multitasking functionality. Windows NT was specially designed for computer networking and also called client/server operating system.

G. **Window 2000**:

Window 2000 has four products each with specific functionality

a. Windows 2000 professional
b. Windows 2000 Server.
c. Windows 2000 Advanced server.
d. Window 2000 Data Center.

H. Window XP:

Window XP is more advance and high functionality window that is used in homes, office and business places.

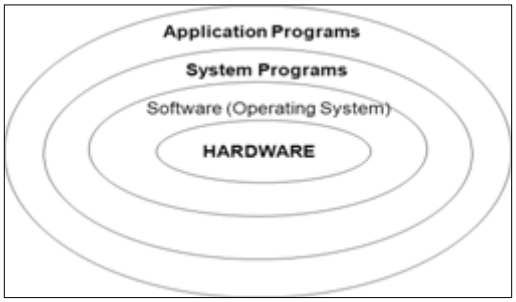
I. Window Vista:

Latest windows operating system introduced in 2006. Windows vista required high level of hardware compatibility.

J. Window 7:

Window 7 is the latest release of Microsoft Windows, a series of operating systems produced by Microsoft for use on personal computers, including home and business desktops, laptops, notebooks, tablet PCs, and media center PCs. Window 7 was released to manufacturing on July 22, 2009, and predecessors, Windows vista. Windows 7's server counterpart, Windows Server 2008 R2, was released at the same time.

Operating System Layered:



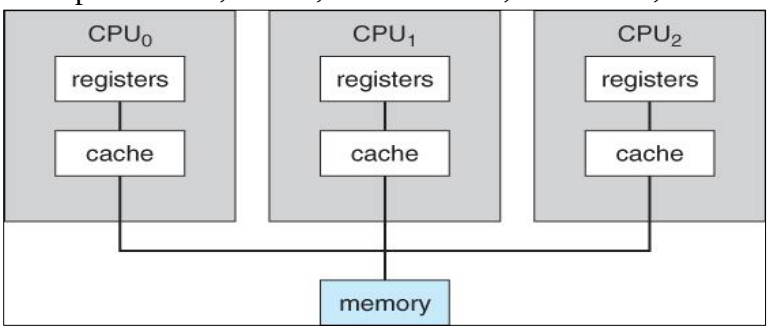
1. The lowest layer consists of Physical devices such as integrated circuits, lines, cables, buses, the CPU, caches, hard-disks and so on.
2. The layer above is primitive software that directly controls the physical device and provides a cleaner interface to the next layer. This layer of software is called Microprogramming, which is used to fetch machine language instructions and is usually stored in the physical device Rom.
3. Above the Microprogramming layer is the Machine language layer. Machine language mainly used to move data around the computer machine itself. The machine language layer controls I/O devices.
4. The operating system resides above the Machine language layer. A major function of an operating system is to hide the above complexities and to provide the user with a convenient set of instructions to work with.
5. Above the operating system are the rest of the system programs, such as the command interpreter (shell), compilers, and editors and so on. These programs are distinct from the operating system, even though a system manufacturer usually provides them.
6. The operating system is protected from users tampering with it by the hardware. Compilers and interpreters run in user mode.
7. For example, you can write your own Object Oriented Compiler for Eiffel code if you want to.
8. The top layer of this hierarchy is the application program. These programs are written by users to solve problems, such as data processing, games and so on.

Computer system architecture:

Computer System are categories into single processor, dual processor, multiprocessor, parallel processor according to number of CPU used.

1. **Single- Processor System:**
   a. Single-processor system contains only one CPU.
   b. The one CPU is capable of executing instruction set.
   c. Some device specific processors are used to reduce overhead of CPU.
   e.g. Disk-controller microprocessor receive a sequence of request from main CPU, creates its own disk queue and scheduling algorithm is implemented.
2. **Multiprocessor System**:
   a. Multiprocessor system is also called parallel system or tightly coupled system.
   b. The computer bus is shared by two or more processors. Also memory and peripheral device can be shared.
   c. The Fig 1.7 shows the multiprocessing architecture.
   The multiprogramming systems are of two types:
   - Asymmetric multiprocessing: Each processor performs all tasks within the operating system. All processors are peer, no master-slave relationship. All processors share the physical memory.

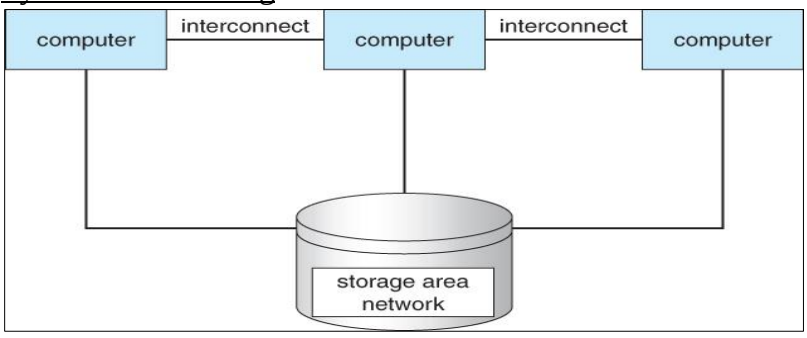
- Symmetric multiprocessing (SMP): Each processor performs all tasks within the operating system. All processors are peer, no master-slave relationship. All processors share the physical memory.
  Example: Solaris, UNIX, Windows XP, mac OS X, LINUX are SMP systems.



d. The recent CPU design includes multiple computing cores on a single chip. This is suitable for server systems such as database server and web server.
e. In dual-core system, two cores are present on the single chip, each core has its own register set own local cache.
f. Advantages of Multiprocessor system:
   - Increased throughput: As the number of processor increases, the result or output is faster. The time required is less to computer the task.
   - Economy of scale: The cost of multiprocessor system is less than that of multiple single-processor system, because peripheral devices, memory, bus is shared. If the multiple user wants to share the same data, and that can be shared by all processor.
   - Increased reliability: In multiprocessor system, if any of the processor fails, the work of that processor is shared by remaining processor and system will not halt.

3. Clustered System:

1. Clustered system keeps multiple CPUs together like parallel systems. Clustered system uses multiple CPU's like multiprocessor system, but they are composed of two or more individual systems or nodes joined together. Fig shows the clustered system structure. This system is highly reliable to provide service to the user. The service will continue even if one or more systems in the cluster fail.
2. Clustering are of two types:
   a. Asymmetric clustering: In this, one machine is in standby mode and other is running the applications. The standby machine is monitoring the task of others.
   b. Symmetric clustering is more efficient.



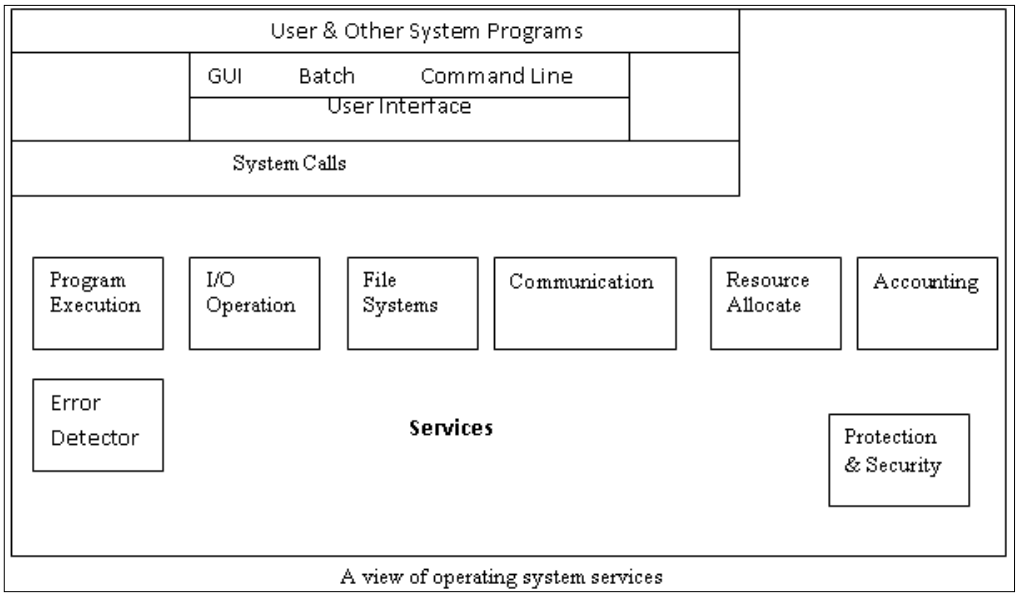
- A cluster consisting of many computers are connected via a network (LAN or WAN).
- Such a system is used to run application concurrently on all computers in the cluster.
- Many computers can share the same database.
- Increasing performance and throughput.
- More reliable.
- Designed for solving high-performance computing tasks.

- Example, oracle parallel server is version of oracle's database which runs on parallel clusters.

Operating System Services:-

Following are the six services provided by operating systems to the convenience of the users.

1.User interface:- All operating system have a User Interface (UI). This interface can take several forms.



A view of operating system services

2.Program Execution:- The purpose of computer systems is to allow the user to execute programs. So the operating system provides an environment where the user can conveniently run programs. The user does not have to worry about the memory allocation or multitasking or anything. These things are taken care of by the operating systems.

Running a program involves the allocating and deallocating memory, CPU scheduling in case of multiprocess. These functions cannot be given to the user-level programs. So user-level programs cannot help the user to run programs independently without the help from operating systems.

3.**I/O Operations:-** Each program requires an input and produces output. This involves the use of I/O. The operating systems hides the user the details of underlying hardware for the I/O. All the user sees is that the I/O has been performed without any details. So the operating systems by providing I/O make it convenient for the users to run programs.

For efficiently and protection users cannot control I/O so this service cannot be provided by user-level programs

4.**File System Manipulation:-** The output of a program may need to be written into new files or input taken from some files. The operating systems provide this service. The user does not have to worry about secondary storage management. User gives a command for reading or writing to a file and sees his /her task accomplished. Thus operating systems make it easier for user programs to accomplish their task. This service involves secondary storage management. The speed of I/O that depends on secondary storage management is critical to the speed of many programs and hence I think it is best relegated to the operating systems to manage it than giving individual users the control of it. It is not difficult for the user-level programs to provide these services but for above mentioned reasons it is best if this service s left with operating system.

5.**Communications:-**There are instances where processes need to communicate with each other to exchange information. It may be between processes running on the same computer or running on the different computers. By providing this service the operating system relieves the user of the worry of passing messages between processes. In case where the messages need to be passed to processes on the other computers through a network it can be done by the user programs. The user program may be

customized to the specifics of the hardware through which the message transits and provides the service interface to the operating system.

6. **Error Detection**:- An error is one part of the system may cause malfunctioning of the complete system. To avoid such a situation the operating system constantly monitors the system for detecting the errors. This relieves the user of the worry of errors propagating to various part of the system and causing malfunctioning.

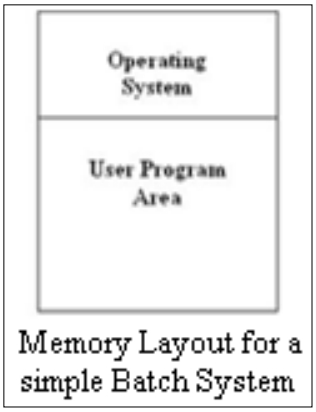
This service cannot allow to be handled by user programs because it involves monitoring and in cases altering area of memory or deallocation of memory for a faulty process. Or may be relinquishing the CPU of a process that goes into an infinite loop. These tasks are too critical to be handed over to the user programs. A user program if given these privileges can interfere with the correct (normal) operation of the operating systems.

Types Of operating System:-

- Operating systems depend on hardware and software requirements
- Results in different types of O/S's
- Examples:
  - Batch operating system
  - Multiprocessor computers
  - Multiprogramming operating system
  - Distributed system
  - Real-time & embedded systems

I. Batch Operating System

1. Early computers were (physically) enormously large machines run from a console. The common input devices were card readers and tape drives. The common output devices were line printers, tape drives, and card punches.
2. The users of such systems did not interact directly with the computer systems. Rather, the user prepared a job—which consisted of the program, the data, and some control information about the nature of the job (control cards)—and submitted it to the computer operator.



Memory Layout for a simple Batch System

3. The job would usually be in the form of punch cards. At some later time (perhaps minutes, hours, or days), the output appeared. The output consisted of the result of the program, as well as a dump of memory and registers in case of program error.
4. The operating system in these early computers was fairly simple. Its major task was to transfer control automatically from one job to the next. The operating system was always (resident) in memory (**Figure 1.1**).
5. To speed up processing, jobs with similar needs were batched together and were run through the computer as a group
6. The operator would sort programs into batches with similar requirements and, as the computer became available, would run each batch.
7. The output from each job would be sent back to the appropriate programmer.
8. A batch operating system, thus, normally reads a stream of separate jobs-(from a card reader, for example), each with its own control cards that predefine what the job does.
9. When the job is complete, its output is usually printed (on a line printer, for example).

II. Spooling operating System.

1. In this execution environment, the CPU is often idle. This idleness occurs because the speeds of the mechanical I/O devices are intrinsically slower than those of electronic devices.

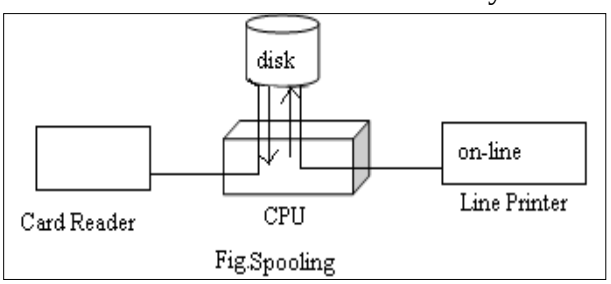


Fig.Spooling

2. Over time, improvements in technology resulted in faster I/O devices. Unfortunately, CPU

speeds increased even faster, so that the problem was not only unresolved, but also exacerbated.

3.The introduction of disk technology has helped in this regard. Rather than the cards being read from the card reader directly into memory, and then the job being processed, cards are read directly from the card reader onto the disk.

4. The location of card images is recorded in a table kept by the operating system. When a job is executed, the operating system satisfies its requests for card-reader input by reading from the disk.

5.Similarly, when the job requests the printer to output a line, that line is copied into a system buffer and is written to the disk. When the job is completed, the output is actually printed. This form of processing is called spooling; as show in fig.

6.Spooling is also used for processing data at remote sites. The CPU sends the data via communications paths to a remote printer (or accepts an entire input job from a remote card reader). The remote processing is done at its own speed, with no CPU intervention. The CPU just needs to be notified when the processing is completed, so that it can spool the next batch of data.

Advantage:-

1. Spooling overlaps the I/O of one job with the computation of other jobs.
2. Spooling has a direct beneficial effect on the performance of the system.
3. Spooling can keep both the CPU and the I/O devices working at much higher rates
4. Spooling Keep both the CPU and I/O devices working at much higher rates.

III. Multiprogramming Operating System:-

1.Spooling provides an important data structure: a job pool. A pool of jobs on disk allows the operating system to select which job to run next, to increase CPU utilization.

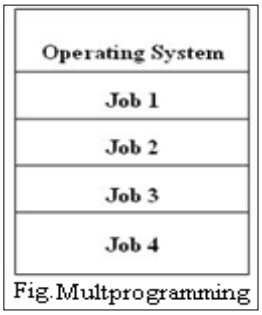


Fig. Multprogramming

2.Thus, the most important aspect of job scheduling is the ability to multiprogram.

The idea is as follows.

3. The operating system keeps several jobs in memory at a time (**Figure)**.

4.This set of jobs is a subset of the jobs kept in the job pool (since the number of jobs that can be kept simultaneously in memory is usually much smaller than the number of jobs that can be in the job pool.)

5.The operating system picks and begins to execute one of the jobs in the memory.

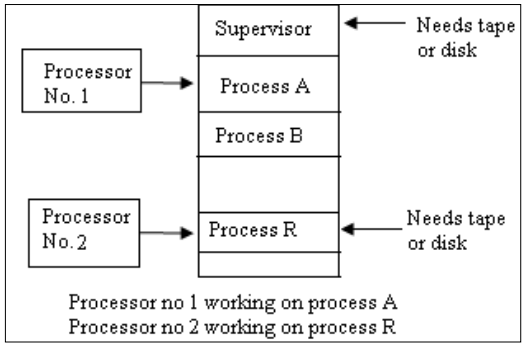
6.Eventually, the job may have to wait for some task, such as a tape to be mounted, or an I/O operation to complete. In a non-multiprogrammed system, the CPU would sit idle.

7.In a multiprogramming system, the operating system simply switches to 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 there is always some job to execute, the CPU will never be idle.

Advantage:-

1. Efficient memory utilization.
2. CPU never sits idle, so it increase the CPU performance.
3. Throughput of the CPU increase.
4. In non-multiprogramming environment the user/program has to wait for CPU much time. But waiting time is limited in multiprogramming.

IV. Multiprocessing system:-



1. This system is similar to multiprogramming system, except that there is more than one CPU available. In most multiprocessor systems, the processors share a common memory. Thus, the user can view the system as if it were powerful single processor.

2.Fig. depicts the manner in which the multiple processors can be used for a multiprogramming. Usually, we visualize several separate processes as being in memory.
3.In actually, a process is often paged so that only part of it is in memory at one time; this allows the number of processes active in the system to be very large.
4.A processor is assigned to a task and operates on it until it is blocked. When task is blocked, the processor selects another task and continues processing.
5.After the blocking condition has been satisfied, a processor will eventually be assigned to the process; it need not be the same physical processor as before.

Advantages of multiprocessor systems

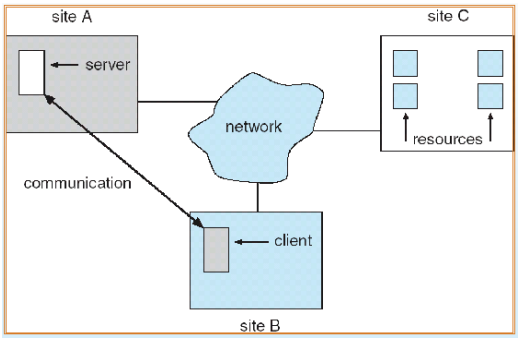
1. If one processor fails, then other processors should retrieve the interrupted process state so that execution of the process can continue.
2. The processors should support efficient context switching operation.
3. Multiprocessor system supports large physical address space & large virtual address space.
4. The IPC mechanism should be provided & implemented in hardware as it becomes efficient & easy.

V. Time Sharing Systems:-

1.Time sharing, or multitasking, is a logical extension of multiprogramming. Multiple jobs are executed by switching the CPU between them, but the switches occur so frequently that the users may interact with each program while running.
2.These systems were developed to provide interactive use of a computer at reasonable cost.
3.A time shared operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared computer. Each user has at least one separate program in memory.
4.A program that is loaded into memory and is begin executed is commonly referred to as a process.
5.A time shared operating system allows many uses to share the computer simultaneously. Since each action or command in, in a time-shared system tends to be short, only a little CPU time is needed for each user.
6.In this, the CPU time is shared by different processes, so it is called as "Time sharing Systems".
7.Time slice is defined by the OS, for sharing CPU time between processes.
8.Examples: Multics, Unix, etc.,

VI. Distribution Systems:-

1.Distributed system is collection of loosely coupled processors interconnected by a communications network.
2.Processors variously called nodes, computers, machines, hosts



3. Site is location of the processor
4. **Types of Distribution Systems**
   a. Network--Operating Systems
      i. Users are aware of multiplicity of machines.

Access to resources of various machines is done explicitly by:
Remote logging into the appropriate remote machine (telnet, ssh)
Remote Desktop (Microsoft Windows)

ii. Transferring data from remote machines to local machines, via the File Transfer Protocol (FTP) mechanism

   b. Distributed Operating Systems

i. Users not aware of multiplicity of machines
ii. Access to remote resources similar to access to local resources
iii. Data Migration – transfer data by transferring entire file, or transferring only those portions of the file necessary for the immediate task.
iv. Computation Migration – transfer the computation, rather than the data, across the system
v. Process Migration – execute an entire process, or parts of it, at different site
vi. Load balancing – distribute processes across network to eve the workload

vii. Computation speedup – sub processes can run concurrently on different sites
viii. Hardware preference – process execution may require specialized processor
ix. Software preference – required software may be available at only a particular site
x. Data access – run process remotely, rather than transfer all data locally

5. **Reasons for distributed systems**
   a. Resource sharing:- sharing and printing files at remote sites Processing information in a distributed database using remote specialized hardware devices
   b. Computation speedup – load sharing
   c. Reliability – detect and recover from site failure, function transfer, reintegrate failed site
   d. Communication – message passing

VII. Real-time Systems

1.Real time systems which were originally used to control autonomous systems such as satellites, robots and hydroelectric dams. A real time operating system is one that must react to inputs and responds to them quickly. A real time system cannot afford to be late with a response to an event.

2.A real time system has well defined, fixed time constraints. Deterministic scheduling algorithms are used in real time systems.

3. Real time systems are divided into two groups: Hard real time system and soft real

4.Real time operating system uses priority scheduling algorithm to meet the response requirement of a real time application.
   a. **Hard real-time systems:** Hard real-time systems guarantee that critical tasks complete on time. In hard real-time systems secondary storage e is limited or missing with data stored in ROM. In these systems virtual memory is almost never found.
   b. **Soft real-time systems:** Soft real time systems are less restrictive. Critical real-time task g ets priority over other tasks and retains the priority until it completes. Soft real-time systems have limited utility than hard real-time systems. For example, Multimedia, virtual reality, Advanced Scientific Projects like undersea exploration and planetary rovers etc.

5.General real time applications with some examples are listed Below

Sr.No.	Real Time	Application Examples
1.	Detection Radar system	Burglar alarm
2.	Process monitoring and control	Petroleum, Paper mill.
3.	Communication	Telephone switching system
4.	Flight simulation and control	Auto pilot shuttle mission simulator
5.	transportation	Traffic light system, Air traffic control

6.Memory management in real time system is comparatively less demanding than in other types of multiprogramming systems. Time-critical device management is one of the main characteristics of real time systems. The primary objective of file management in real time system is usually speeding of access, rather than efficient utilization of secondary storage.