1 Operating System Concepts

1. Evolution of Operating System & History	4. Elements of an operating system.
2. Need of an Operating system	5. Operating System as a Resource Manager
3. Single user & Multiuser Operating system	* Exercise

DEFINITION

"Operating system is an integrated set of programs that is used to manage the various resources and over all operation of the computer system."

- ✓ In computing, an operating system (OS) is an interface between hardware and user, which is responsible for the management and coordination of activities and the sharing of the resources of a computer, that acts as a host for computing applications run on the machine
- ✓ *"Operating System is a program that acts as an intermediary between a user of a computer and the computer hardware. It enables effective communication in effective manner."*
- ✓ It is an S/W that controls the execution of programs and that provides services such as resource allocation, scheduling, I/O control, and data management.
- ✓ Essential system S/W which co-ordinates & controls the scheduling of work on system & the use of systems resources is called operating system.

OBJECTIVE OF OPERATING SYSTEM

There are mainly following objective of Operating System :

1) Convenience : An Operating System makes computer System more convenient to use.

2) Efficiency : An Operating System allows the computer system resources to be used in an efficient manner.

3) Ability to evolve : Permit effective development, testing and introduction of new system functions without interfering with service.

COMPONENTS OF COMPUTER SYSTEM

A Computer System can be divided into four 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).

✓ The Operating System as a User Computer Interface.

The H/W & S/W used in providing applications to user can be viewed in layered or hierarchical fashion as

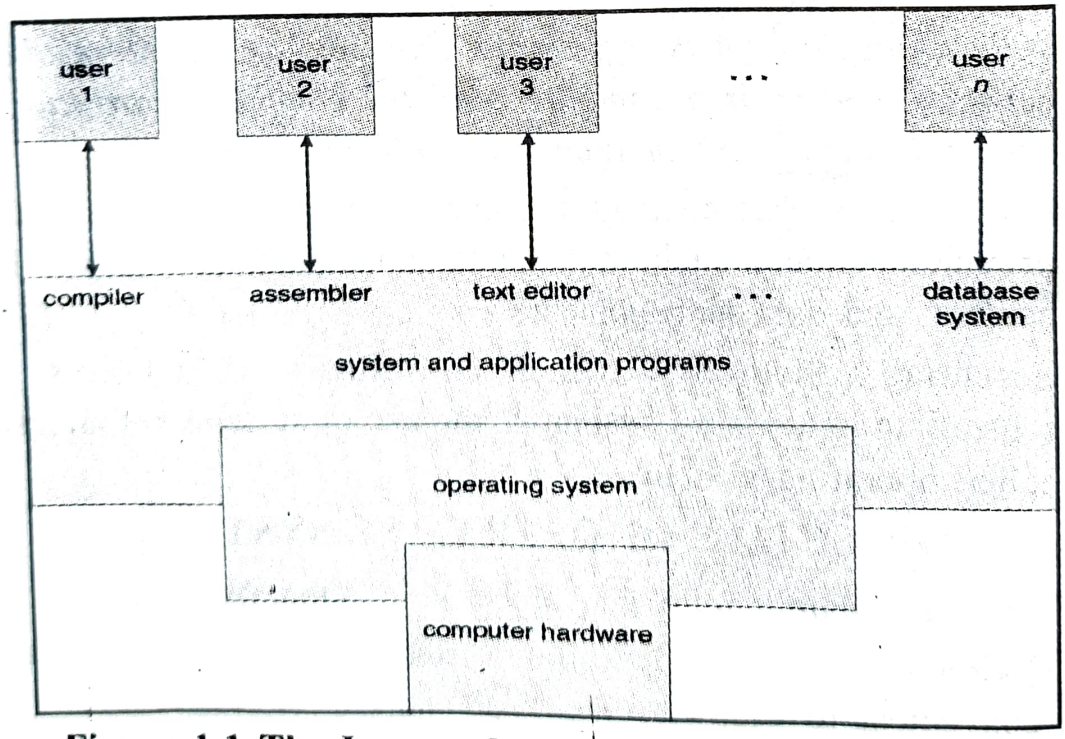

Figure 1.1 The Layers & views of a computer system

- ✓ The end users, generally is not concerned with details of the computers H/W, thus the end user views a computer system in terms of a set of applications.
- ✓ An application can be expressed in a programming language and is developed by application programmer.
- ✓ If one were to develop an application programmer as a set of machine instructions that is completely responsible for controlling the computer H/W, one would be faced with a complex task.
- ✓ To ease this task a set of systems programs are provided. The most important system program is Operating System.
- ✓ The Operating System makes the details of the H/W from the programmer & provides the programmer with convenient interface for using the system.
- ✓ It acts as mediator making it easier for the programmer & application to access and use those facilities and services.

OPERATING SYSTEM PROVIDES FACILITIES IN FOLLOWING AREAS

Program Development

- ✓ The Operating System 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 of the Operating System are supplied with the Operating System and are referred to as application program Development tools.

Program Execution

- ✓ A number of tasks need to be performed to execute a program. Instruction and data must be loaded into main memory, I/O Devices & file must be initialized, and other resources must be prepared. The Operating System handles these scheduling duties for the user.

Access to I/O Devices

✓ Each I/O devices requires its own particular set of instruction or control signals for operation. The Operating System provides uniform interface that hides these details so that the programmer can access such devices using simple reads and writes.

Controlled Access to Files

✓ In the case of files, for the operating system control must include a detailed understanding of not only the nature of I/O devices (Disk Drive, Tape Drive) but also the structure of data contained in the files on the storage medium. Further in the ease of a system with multiple users, the Operating System may provide protection mechanism to control access to the files.

Error detection & Response

✓ A variety of errors can occur while computer system is running. This include internal & external H/W errors, such as a memory error, or device failure; and various S/W errors, such as arithmetic overflow, attempt to access forbidden memory location and inability of O.S to grant request of an application.

✓ In such case, the Operating System must provide a response that clears the error condition with least impact on running applications. The response may range from ending the program that caused the error, to retrying the operation, to simply reporting the error to application.

Accounting

✓ We want to keep track of which users use how much and what kinds of computer resources. This record keeping may be for accounting (so that users can be billed) or simply for accumulating usage statistics usage statistics may be variable tool for researchers who wish to re-configure the system to improve computing services.

1.3 History of Operating Systems

Starting from beginning, computers and Operating Systems have evolved a lot. Very early, Charles Babbage, English mathematician, designed a true digital computer. He spent most of his life to get his machine properly worked. He remained unsuccessful. But, as an important outcome of his efforts, he realized the need of Software.

There are drastic changes in this field after unsuccessful attempts of Babbage. In the following sections, a "walk-through" of the history is given.

1. **The First Generation (1945 – 1955)**
   - Hardware: Vacuum tubes and plug boards.
   - Neumann and others succeeded in building Calculating engine.
   - No Operating Systems, 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 OS and variations of it such as Time sharing, Interactive, Multitasking OS came in picture.
   - SPOOLing (Simultaneous Peripheral Operation On Line) began during this time duration.
4. **The Forth Generation (1980 – Present)**
   - Hardware: LSI (Large Scale Integration) circuits.
   - Personal Computers evolved.

- Worst business decision in a record history :

  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 dominant operating systems 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 Seattle Computer Products. (Ya, you are absolutely right. He purchased it. Allegedly in $50,000.) 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 the dominant 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
- Microsoft also has the chain of Operating Systems, such as Windows 95, 98, NT, ME, XP, and at last Vista.
- In a parallel to this, UNIX and Linux also got a good evolution, which will be described later in the chapter of the UNIX Operating System.

1.4 Types of Operating Systems

Starting from the beginning, Operating Systems have evolved a lot. In this section the details of different types of operating systems are explained. There are six important operating systems as per the syllabus. They are:

(i) Batch Operating System
(ii) Multiprogramming Operating System
(iii) Multi-user Operating System
(iv) Multitasking / Time-sharing Operating System
(v) Network Operating System
(vi) Distributed Operating System
(vii) Multithreading Operating System

Other important OS, known as Real Time OS is given in supplementary topics.

1. Serial Processing System :

- ✓ With the earliest computers, from the late 1940s to the mid 1950s, the programmers interacted directly with the computer H/W, there was no Operating System. The machines were run from a console consisting of display lights, toggles switches, same from of input devices, and a printer.
- ✓ Programs in machines code were loaded via input devices (Card reader). If an error halted the program, the error condition was indicated by the lights. The programmer could proceed to examine processor registers and main memory to determine the cause of error. If the program proceeded to a normal completion, the O/P appeared on the printer.

Disadvantages :

Scheduling :

- ✓ Hardcopy sign-up sheet to reserve machine time. User signup for a block of time and may not finish within that limit. It might west computer's time of may not finish in allotted time and forced to stop before resolving the problem.

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 functions. Each of these steps could involve mounting/dismounting tapes or setting up card checks. If an error occurred the helpless users typically had to go back to the beginning of the setup sequence. Thus a considerable amount of time was spent just in setting up the program to run.
- ✓ This mode of operation could be formed serial processing, reflecting the fact that users have access to the computer in

1.4.1 Batch Operating System

In earlier days, computers were large machines. The common input devices were card readers, tap drives. The common output devices were line printers, punch cards and tap drives.

The entire system worked in following manner:

- The OS was very simple and always resident in main memory.
- Programmers would prepare a job and submit it to the operator. Job was consisted of program, data and some control information.
- Operator would sort them in batches with similar requirements, and as computer became available, run them batch wise.
- At some later time (may be after some hours or even after some days), output appeared. The output consisted of the result of the program and error information. Programmers needed to wait during this time, and then collect output from the operator.

Here, memory is divided in two parts as given in Figure 1.2. It is shared between the **operating system** 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 procedure will continue until all the jobs in a batch get executed.

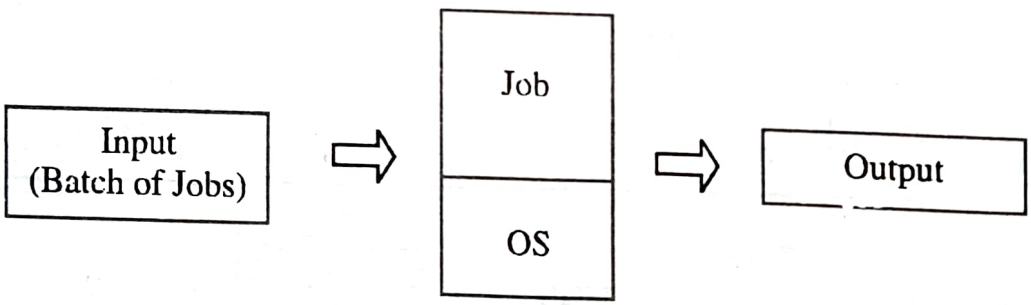

FIGURE-1.2 : MEMORY STRUCTURE IN BATCH OPERATING SYSTEM

Introduction to disk technology made it possible to direct access to several jobs and job scheduling.

- **Disadvantages :**
  - 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.
  - Operations were too much time consuming.

Solutions :

1. Off-line Operation :

Speed up computation by loading jobs into memory from tapes while card reading and line printing is done off-line using smaller machines.

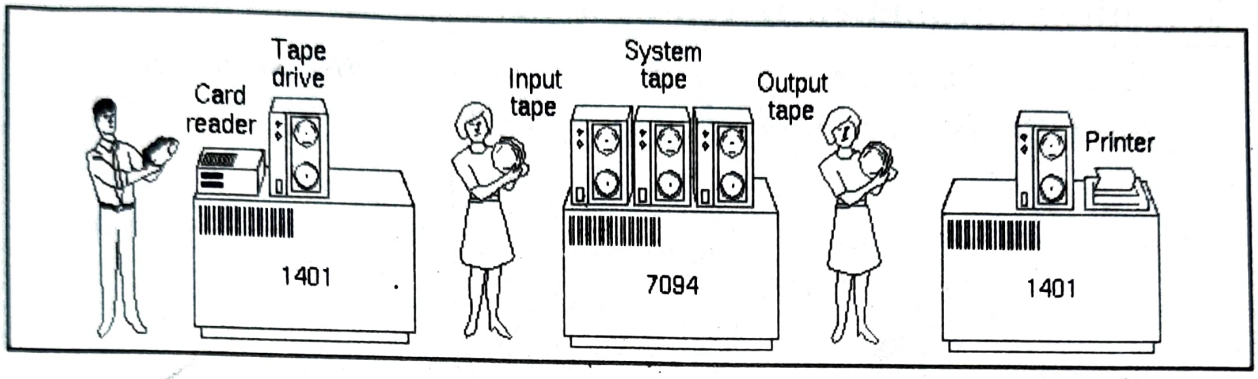

Figure 1.3 Off-line Operation

2. Spooling - (Simultaneous peripheral operation online) :

- It is a disk technology which is overlapping of input of one job, execution of other job and O/P of some other job.
- Spooling in essence users the disk as a huge buffer, for reading as far ahead as possible on input devices and for starting O/P files until the O/P devices able to accept them.

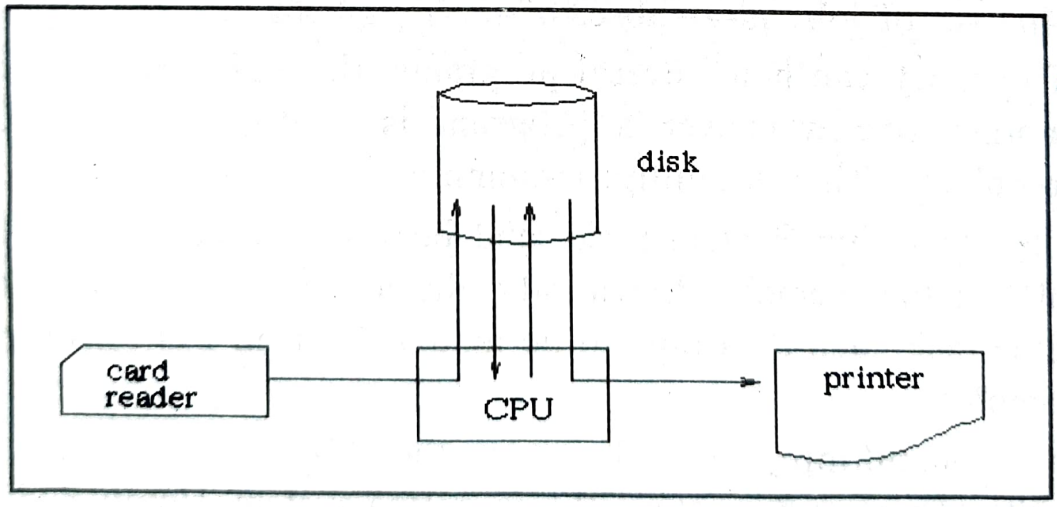

Figure 1.4 Spooling

- Spooling provides an important data structure: a job pool. Spooling will generally result in several jobs that have already been read waiting on disk, ready to run. A pool of jobs on disk allows the operating system to select which job to run next, to

1.4.2 Multiprogramming Operating System

A multiprogramming Operating System provides the ability to run more than one program **concurrently**.

Contrast to Batch Operating System, here more than one program can be loaded in main memory simultaneously. These programs can be executed concurrently (or say, simultaneously). Memory is shared between OS and such kind of programs.

Multiprogramming is also referred as **Multitasking**. User can execute more than one program simultaneously. This also means, user can perform more than one task simultaneously. For example, a user can execute program like MS-Word to prepare a document, as well as can run some other program (like Windows Media Player) to play songs. Both of these tasks can be performed simultaneously.

Multiprogramming significantly improves **system throughput** and **resource utilization**. Various resources, such as CPU, can be utilized as much as possible among various simultaneously executing programs. For example, when one program requests for I/O operation, the CPU will be free; and it can be allocated to other job.

The following figure 1.3 shows the concept of multiprogramming with three concurrently running programs.

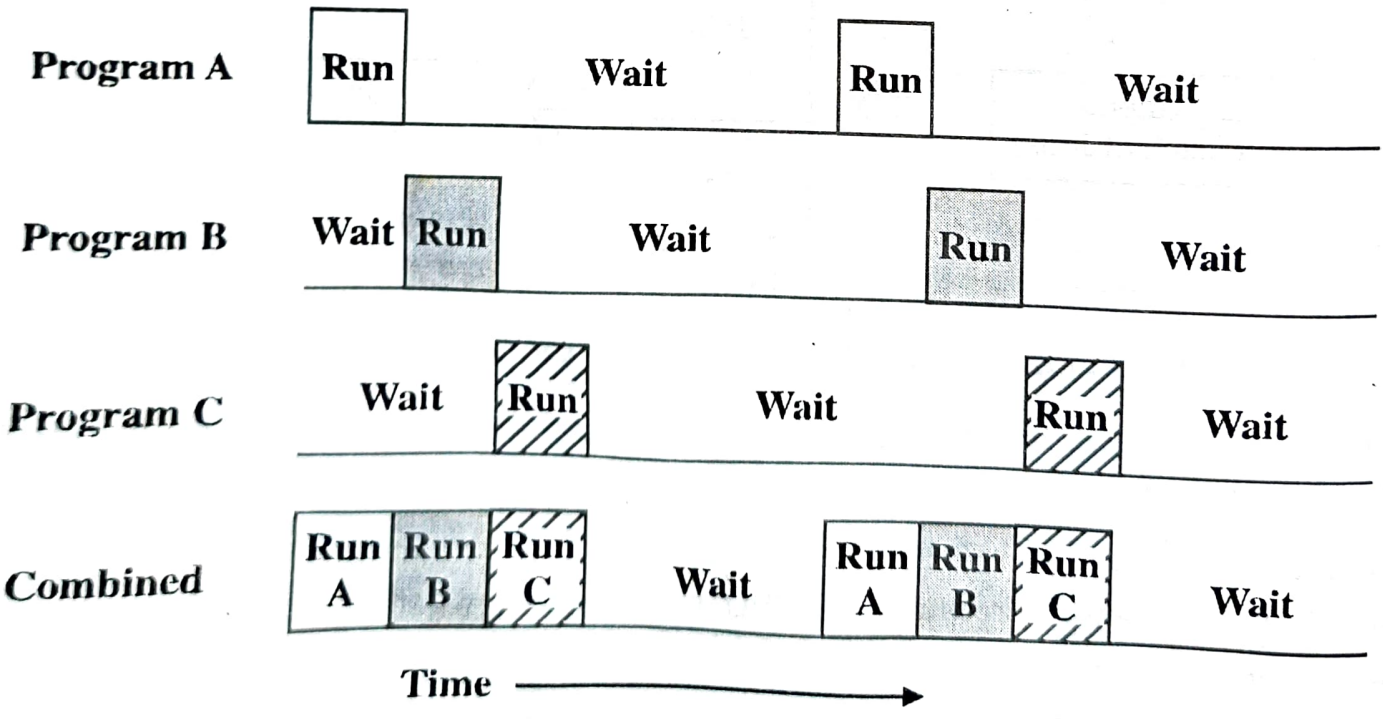

FIGURE-1.3 : MULTIPROGRAMMING WITH THREE PROGRAMS

Implementation :

Multiprogramming can be implemented in one of the two following ways:

1. **Non-preemptive :**
   - 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 as shown in figure 1.3.
2. **Preemptive :**
   - A program is allowed to execute only for some maximum time duration.
   - After this time duration, a CPU is forcibly taken away from the program.
   - This kind of OS is also referred as Time Sharing Operating System.
   - Here, one smaller programs need not to wait for other large program to finish its execution. So, this kind of implementation minimizes the response time for user.
   - And so, they are suitable for interactive programs in which user can directly interact with the program.
   - Today most of the modern Operating Systems running on personal computers belong to this category.

■ **Required Features :**

Here, as multiple programs reside simultaneously in main memory, OS requires following features :

- To store several programs in memory simultaneously, better **memory management** & **protection** are required (Chapter 3).
- As memory is limited, concepts like **swapping** and **virtual memory** are used to accommodate more than one program in memory. (Don't worry. These topics are explained in chapter 3.)
- CPU **scheduling** (chapter 2) should be sophisticated, to provide fairness to all programs.
- **Disk** management and **file** system (chapter 4) is complex now.
- **Synchronization** & **Communication** among running programs and problems like **deadlock** should be overcome (chapter 2).

1.3.3 Time Sharing System

- Time sharing is also called multitasking operating system. It is logical extension of the multiprogramming operating systems. User interaction with program is possible in time sharing operating system. During execution of the program, user interacts directly with the program, supplying information to the program.
- Multi-tasking means that the computer can work with more than one program at a time. For example, user could be working with information from one database on the screen analyzing data, while the computer is sorting information from another database, while a excel sheet is performing calculations on a separate worksheet.
- Many users share the computer system simultaneously in time sharing operating system. Time sharing system uses multiprogramming and CPU scheduling. Each user has at least one separate program in memory.
- In time sharing system, each user is given a time slice for

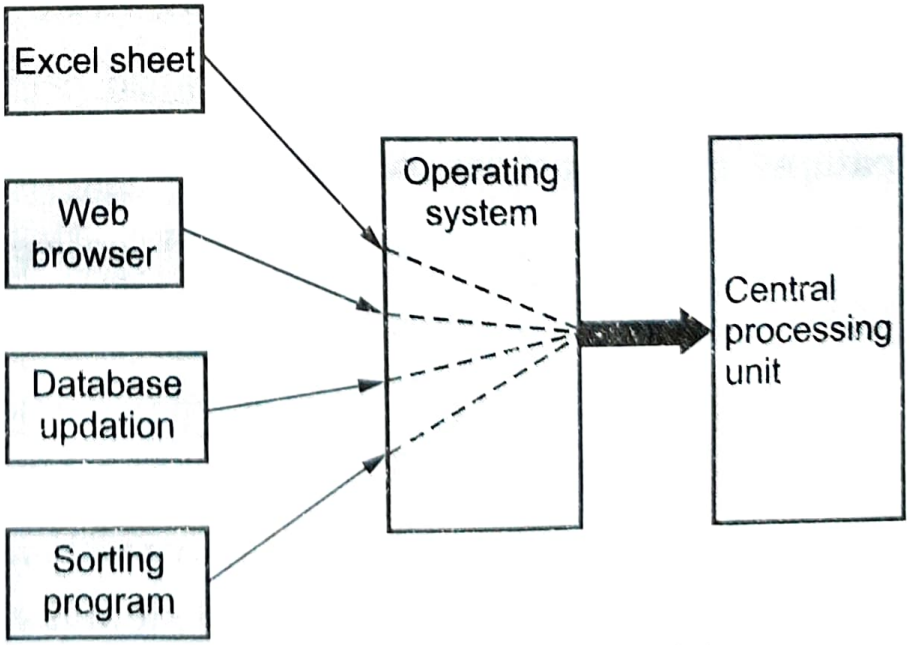

Fig. 1.3.6 Multitasking OS

executing his/her job in round robin fashion. Job continues until the time slice ends.

- Fig. 1.3.6 shows multitasking OS.
- Concept of virtual machine is used in time sharing system. It creates virtual machine one per user. User interaction with system by using virtual machine. User enters the command for virtual machine and result will received back to user.
- Time sharing system is more complex than multiprogramming operating system. It also takes help of file system. File system is stored on the disk so disk management is also required.
- Major problem with time sharing system is protection and security of data.
- Time sharing system uses medium term scheduling such as round robin for the foreground. Background process uses can use a different scheduling method.
- Difference between multiprogramming and multitasking operating system is context switching. In multiprogramming system a context switching occurs only when the currently executing process stalls for some reasons. Time sharing system gives each user the impression that the entire system is dedicated to his use. Context switching simply allows several applications to be open, but only one is working at a time.
- Truly speaking, even in true multi-tasking, only one application program is ever running at anyone instant. Because the computer automatically switches from one program to the next program so quickly, all the programs seem to run simultaneously.

1.3.4 Multiprocessor System

- Multiprocessor system means more than one processor in close communication. All the processor share common bus, clock, memory and peripheral devices.
- Multiprocessor system is also called parallel system or tightly coupled systems.

atures of Multiprocessor Systems :

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

- Multiprocessor systems are of two types : **symmetric multiprocessing and asymmetric multiprocessing.**

Symmetric multiprocessing

- In symmetric multiprocessing, number of homogeneous processor are running independently without affecting other programs. Each processor uses different data and program but sharing some common resources like memory, I/O device etc.
- Fig. 1.3.7 shows symmetric multiprogramming system.

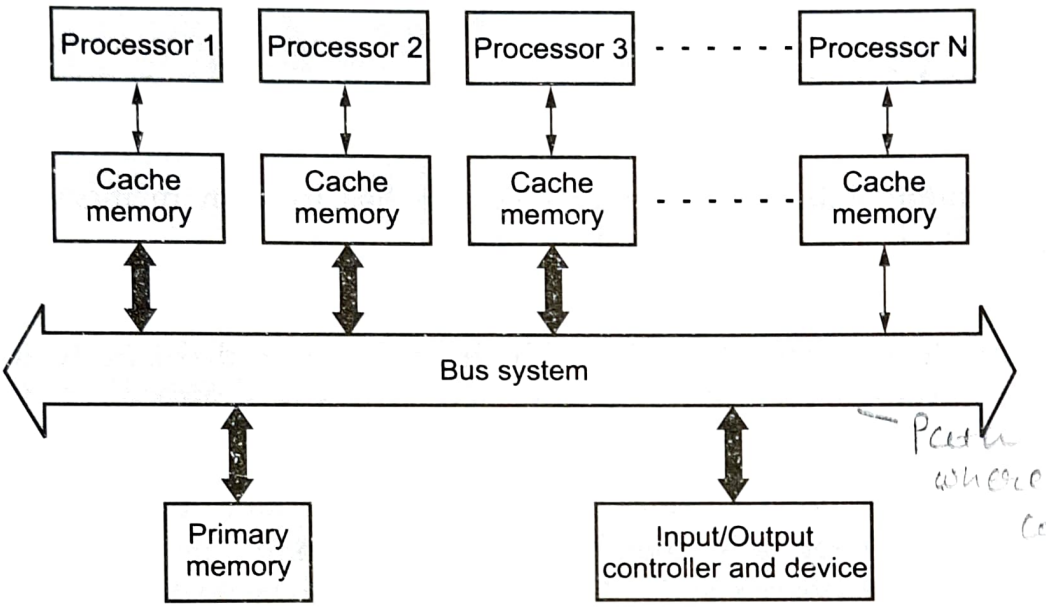

Fig. 1.3.7 Symmetric multiprogramming system

- In SMP, all processor are at the same level. They work in peers. There is no master-slave relationship in between the processor.
- Each processor shares a single copy of operating system.
- Symmetric multiprocessing is easier to implement in operating systems. Examples of symmetric multiprocessing operating systems are Windows NT 4.0, Novell Netware 2.1 etc.

Asymmetric multiprocessor

- All CPUs are not equal. There is one master processor and remaining is the slave processor. Each processor has its own memory address space.
- A master processor controls the whole operations. It gives the instruction to the slave processor or slave processor uses some predefined set of tasks.
- Each processor has own task which assign by master processor. Asymmetric multiprocessing is difficult to implement.

Advantages of multiprocessor system

1. **Throughput :** Throughput increases because number of processor is increases.

2. **Cost :** Multiprocessor system is cheaper than the multiple single processor system.
3. **Reliability :** If one processor fails, it has no effect on whole system operation.
4. **Response time :** Response time is less because of increased number of processor.

Difference between symmetric and asymmetric multiprocessing

- Symmetric multiprocessing treats all processors are equals, an I/O can be processed on any CPU. Asymmetric multiprocessing has one master CPU and the remainder CPUs are slaves. The Master distributes takes among the slaves and I/O is usually done by the master only.

1.3.6 Distributed Systems

- Distributed operating systems depend on networking for their operation. Distributed OS runs on and controls the resources of multiple machines. It provides resource sharing across the boundaries of a single computer system. It looks to users like a single machine OS. Distributing OS owns the whole network and makes it look like a virtual uniprocessor or may be a virtual multiprocessor.
- **Definition :** A distributed operating system is one that looks to its users like an ordinary operating system but runs on multiple, independent CPU.

Advantages of distributed OS :

1. **Resource sharing :** Sharing of software resources such as software libraries, database and hardware resources such as hard disks, printers and CDROM can also be done in a very effective way among all the computers and the users.
2. **Higher reliability :** Reliability refers to the degree of tolerance against errors and component failures. Availability is one of the important aspect of reliability. Availability refers to the fraction of time for which a system is available for use. Availability of a hard disk can be increased by having multiple hard disks located

at different sites. If one hard disk fails or is unavailable, the program can use some other hard disk.

3. **Better price performance ratio :** Reduction in the price of microprocessor and increasing computing power gives good price-performance ratio.
4. Shorter **responses times** and higher **throughput**.
5. **Incremental growth :** To extend power and functionality of a system by simply adding additional resources to the system.

- Fig. 1.3.9 shows the distributed system.

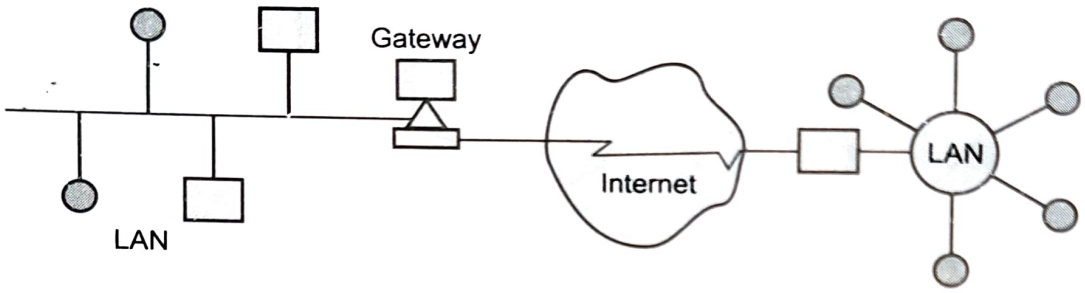

Fig. 1.3.9 Distributed system

- Examples of distributed operating system are amoeba, chrous, mach and v-system.

Difficulties in distributed OS are

1. There are no current commercially successful examples.
2. Protocol overhead can dominate computation costs.
3. Hard to build well.
4. Probably impossible to build at the scale of the internet.

1.3.7 Network Operating System

- Network operating system is designed by using existing operating system.
- In network operating system, user can login to the remote resources and then access that resource. There are multiple machines are connected by communication link. User can transfer data from their own machine to remote machine and vice versa.
- Telnet and remote file transfer are the example of network operating system.
- NOS supports multiple user accounts at the same time and enables concurrent access to shared resources by multiple clients.
- A function of network operating system is distributed over the network computers. It adds new functions whenever required. Server machine support multitasking. Operating system must be capable of executing multiple tasks at the same time.

- User knows the location of the resources before using that resource. Each machine has its own network operating system.

1.3.8 Real Time Operating Systems

- Time constraints is the key parameter is real time systems. It controls autonomous system such as robots, satellites, air traffic control and hydroelectric dams.
- When user gives an input to the system, it must process within the time limit and result is sent back. Real time system fails if it does not give result within the time limits.
- Real time systems are of two types : **Hard real time and soft real time**.
- Critical task is completed within the time limit in hard real time system. All the delay in the system is fixed and time bounded. Existing general purpose operating system does not support the hard real time system functions. Real time task cannot keep waiting for longer time without allocating kernel.
- Soft real time system is less restrictive type. Soft real time cannot guarantee that it will be able to meet deadline under all condition. Example of soft real time system is digital telephone and digital audio.
- Real time operating system uses priority scheduling algorithm to meet the response requirement of a real time application. General real time applications with their example are listed below :
  1. Transportation : Air traffic control and traffic light system
  2. Communication : Digital telephone
  3. Process control : Petroleum and paper mill
  4. Detection : Burglar system and radar system
  5. Flight simulation : Auto pilot shuttle mission simulator.

Memory management in real time system is less demanding than in other type of multiprogramming operating systems. Time critical device management is one of the main characteristics of the real time systems.

- A Real-Time Operating System (RTOS) is an operating system intended to serve real-time application process data as it comes in, typically without buffering delays.
- In general, an operating system is responsible for managing the hardware resources of a computer and hosting applications that run on the computer. An RTOS performs these tasks, but is also specially designed to run applications with very precise timing and a high degree of reliability. This can be especially

1.3.8.1 Difference between Hard and Soft Real Time System

Hard real time	Soft real time
Hard response time is required.	Soft response time is required.
Data integrity is short term.	Data integrity is long term.
Size of data file is small or medium.	Size of data file is large.
Peak load performance is predictable.	Peak load performance is degraded.
Hard real-time systems have little laxity and generally provide full deadline compliance.	Soft real-time systems are more flexible. They have greater laxity and can tolerate certain amounts of deadline misses.
Safety critical systems are typically hard real time systems.	Linux and many OS provide soft real time system.

1.3.9 Interactive System

1.4 ELEMENTS OF AN OPERATING SYSTEM :

1. The Kernel :

The kernel is the heart of the operating system. Amongst 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.

2. Memory Management :

The name of this gives you a good idea what it is all about. It is the responsibility of memory management to share your computers physical memory among the processes which want to use it. It also has to manage such situations where there may not be enough physical memory to share out.

3. Input/Output :

On this layer all the physical communication between your computers hardware, such as disk drive, keyboard, mouse, and screen and so on, takes place.

4. File Management :

Again the name of file management may give you a clue as to what it does. It is the job of this layer to control how the files on your computers hard drive are stored and accessed by any application seeking to use them.

5. The User Interface :

The last element as we have been calling them, of an operating system is the User Interface. This layer is probably the easiest of all to understand since it is the first thing you see when your operating system has logged you in. It is the job of this layer to provide a means for the user to actually interact with the rest of the layers and as such the system as a whole.

Fig. 1.1.1 Computer system conceptual view

1.1.2 Operating System as Resource Manager

- A computer is a set of resources. These resource provides various functions to t user. Functions like data movement, storing of data and program, operation data are control by an operating system.
- Fig. 1.1.2 shows OS as a resource manager.

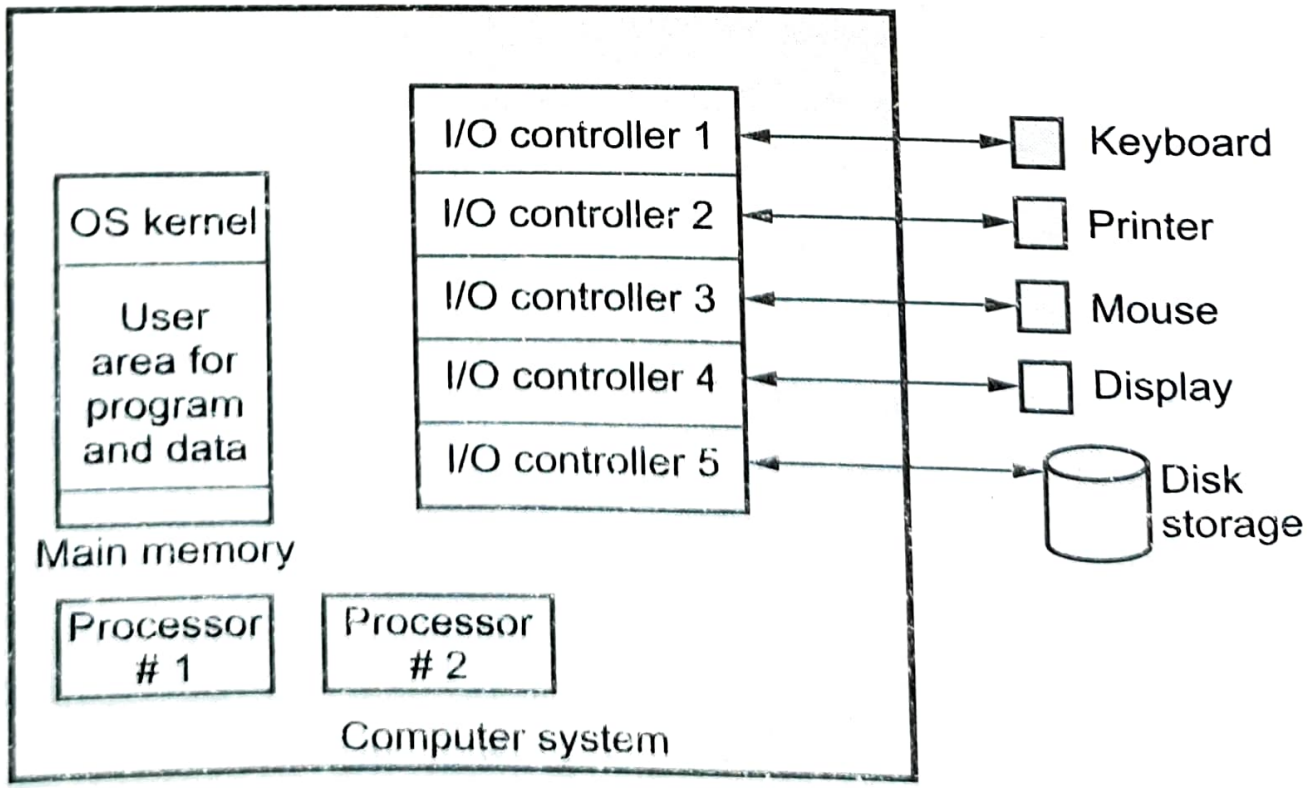

Fig. 1.1.2 OS as a resource manager

- The operating system is responsible for managing the all resources. A portion of the OS is in main memory. This portion of the OS is called kernel.
- User program and data is also stored in remaining parts of the memory. Allocation of main memory is controlled by operating system with the help of memory management hardware.
- I/O device is controlled by OS and it decides when an I/O device can be used by program in execution. Processor is one type of resource and OS control the execution of user program on the processor.
- Modern OS allows multiple programs to run at the same time. If multiple users are using computer then there is need of managing and protecting the memory, I/O devices and other resources.
- Resource management includes sharing resources in different ways. Time and space are the two concept for resource sharing.
  1. **Time :** Time slot is allocated to each program first one gets to use the resource then another and so on.
  2. **Space :** Consider the example of main memory. Main memory is normally divided up among several running programs, so each one can be resident at the same time.

University Questions

1. *What is operating system ? Give functions of operating system.* **GTU : Winter-13, Marks 4**
2. *Explain the objectives and functions of operating systems.* **GTU : Summer-14, Marks 7**

Looks over and protects the computer.

FUNCTIONS OF AN OPERATING SYSTEM :

(1) Process management :

A process is a program in execution. It is the job, which is currently being executed by the processor. During its execution a process would require certain system resources such as processor time, main memory, files etc.

OS supports multiple processes simultaneously. The process management module of the OS takes care of the creation and termination of the processes, assigning resources to the processes, scheduling processor time to different processes and communication among processes.

(2) Memory management module :

It takes care of the allocation and deallocation of the main memory to the various processes. It allocates main and secondary memory to the system/user program and data. To execute a program, its binary image must be loaded into the main memory.

Operating System decides :

(a) Which part of memory are being currently used and by whom.

(b) Which process to be allocated memory ?

(c) Allocation and de allocation of memory space.

(3) I/O management :

This module of the OS co-ordinates and assigns different I/O devices namely terminals, printers, disk drives, tape drives etc. It controls all I/O devices, keeps track of I/O request, issues command to these devices.

I/O subsystem consists of :

(i) Memory management component that includes buffering, caching and spooling.

(ii) Device driver interface

(iii) Device drivers specific to hardware devices.

(4) File management :

Data is stored in a computer system as files. The file management module of the OS would manage files held on various storage devices and transfer of files from one device to another. This module takes care of creation, organization, storage, naming, sharing, backup and protection of different files.

(5) Scheduling :

The OS also establishes and enforces process priority. That is, it determines and maintains the order in which the jobs are to be executed by the computer system. This is so because the most important job must be executed first followed by less important jobs.

(6) Security management :

This module of the OS ensures data security and integrity. That is, it protects data and program from destruction and unauthorized access. It keeps different programs and data which are executing concurrently in the memory in such a manner that they do not interfere with each other.

(7) Processor management :

OS assigns processor to the different task that must be performed by the computer system. If the computer has more than one processor idle, one of the processes waiting to be executed is assigned to the idle processor. OS maintains internal time clock and log of system usage for all the users. It also creates error message and their debugging and error detecting codes for correcting programs.

System Calls :

System call is an interface between the operating system and th user programs. All Operating Systems provide some set of system calls

Example :

Consider a situation in which executing program requires some system service, such as to read data from a file. Suppose file resides on the hard disk. As user programs run in user mode, they directly cannot access the hardware, here, the hard disk, so program will request the operating system to provide service. Operating System runs in Kernel mode and can access all the features of the hardware. Here, Operating System will run on behalf of the user program and provide the require service, such as reading data from a file.

EXERCISE

Q-1. Short Questions :

1. What is Operating System ?
2. Write an Objective of Operating System.
3. List out Operating System which is multiuser and multitasking ?
4. DOS is multiuser & multitasking operating system. State true or false.
5. Differentiate Multiuser and Multitasking Operating System.
6. What is Multiprocessor System ?
7. What is Multithreading Operating System ?
8. What is GUI ?
9. What is Spooling ?
10. What is System Call ?

Q-2. Long Questions :

1. Explain needs of Operating System.
2. Explain Operating System as a Resource Manager.
3. Explain Evolution of Operating System.
4. Explain Elements of Operating System.
5. Explain Functions of Operating System.

■ ■ ■