Unit1. Operating System Concepts

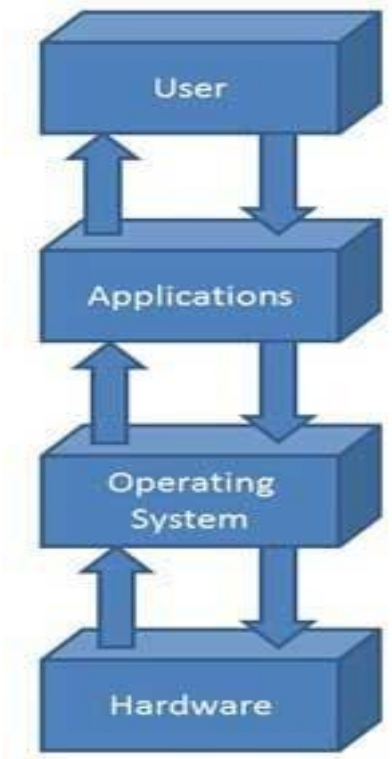
- Evolution of Operating System & History
  - Need of an Operating System
  - Single User & Multi User Operating System
- Elements of an Operating System
  - Operating System as a Resource Manager

❖ What is Operating System?

- ➢ An Operating System (OS) is software that manages computer hardware and software resources and provides common services for computer programs.
- ➢ The operating system is an essential component of the system software in a computer system.
- ➢ Application programs usually require an operating system to function.
- ➢ An Operating system acts as an interface between user and hardware of the computer system.
- ➢ An Operating system also acts as a manager of resources of the computer system such as processor, memory files and I/O devices.

The Operating System is a program with the following features –

- ➢ An operating system is a program that acts as an interface between the software and the computer hardware.
- ➢ It is an integrated set of specialized programs used to manage overall resources and operations of the computer.
- ➢ It is specialized software that controls and monitors the execution of all other programs that reside in the computer, including application programs and other system software.



Objectives of Operating System

The objectives of the operating system are –

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

❖ Characteristics of Operating System

Here is a list of some of the most prominent characteristic features of Operating Systems –

- ➢ **Memory Management** – Keeps track of the 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 it.

- **Processor Management** – Allocates the processor (CPU) to a process and deallocates the processor when it is no longer required.
- **Device Management** – Keeps track of all the devices. This is also called I/O controller that decides which process gets the device, when, and for how much time.
- **File Management** – Allocates and de-allocates the resources and decides who gets the resources.
- **Security** – Prevents unauthorized access to programs and data by means of passwords and other similar techniques.
- **Job Accounting** – Keeps track of time and resources used by various jobs and/or users.
- **Control Over System Performance** – Records delays between the request for a service and from the system.
- **Interaction with the Operators** – Interaction may take place via the console of the computer in the form of instructions. The Operating System acknowledges 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.
- **Coordination between Other Software and Users** – Coordination and assignment of compilers, interpreters, assemblers, and other software to the various users of the computer systems.

Evolution of Operating System & History

❖ Serial Processing

- Early computer from late 1940 to the mid 1950.
- The programmer interacted directly with the computer hardware.
- These machines are called bare machine as they don't have OS.
- Every computer system is programmed in its machine language.
- Uses Punch Card, paper tapes and language translator

These systems presented two major problems.

1. Scheduling
2. Set up time

1. Scheduling

- Used signup sheet to reserve machine time.
- A user may sign up for an hour but finishes his job in 45 minutes.
- This would result in wasted computer idle time, also the user might run into the problem not finish his job in allotted time.

2. Set up Time

A single program involves:

- Loading compiler and source program in memory
- Saving the compiled program (object code)
- Loading and linking together object program and common function
- Each of these steps involves the mounting or dismounting tapes on setting up punch cards.
- If an error occur user had to go the beginning of the set up sequence. Thus, a considerable amount of time is spent in setting up the program to run.

This mode of operation is turned as serial processing, reflecting the fact that users access the computer in series.

❖ Batch Processing

- Early computers were very expensive, and therefore it was important to maximize processor utilization.

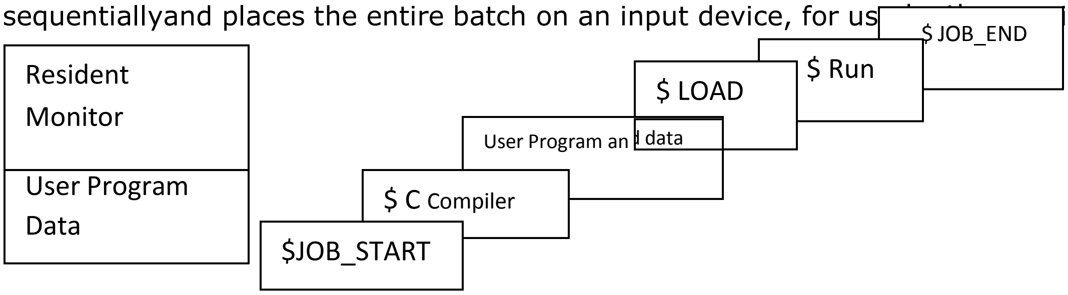
- The wasted time due to scheduling and setup time in Serial Processing was unacceptable.
- To improve utilization, the concept of a batch operating system was developed.
- In this type of system, there is no direct interaction between user and the computer.
- Batch is defined as a group of jobs with similar needs.
- The operating system allows users to form batches.
- The user has to submit a job (written on cards or tape) to a computer operator.
- Then computer operator places a batch of several jobs on an input device.
- Jobs are batched together by type of languages and requirement.
- Then a special program, the monitor, manages the execution of each program in the batch.
- The monitor is always in the main memory and available for execution.
- Computer executes each batch sequentially, processing all jobs of a batch considering them as a single process called batch processing.

<u>Working of batch processing</u>

- For example by batching several " C " compilation job s together, the C compiler can be loaded only once to process all of them in row then put another batching that contains "Java " compilation jobs together, and java compiler also loaded only once to execute all job in batch
- A mounted batch of job must be executed automatically.
- For that, some instruction must be provided to instruct operating system how to treat each individual job.
- These instructions are usually supplied as operating system commands which are written in Job Control Language (JCL) and embedded in batch stream.
- JCL command include job start and end commands, commands for loading and execution of program and command to announce resource need such as expected execution time and memory requirements .
- These commands are embedded in job stream with user program and data.

1. Batch Monitor

- The central idea behind the batch-processing scheme is the use of a piece of software known as the monitor.
- User submits the job on cards or tape to a computer operator, who batches the jobs together sequentiallyand places the entire batch on an input device, for use by the monitor.



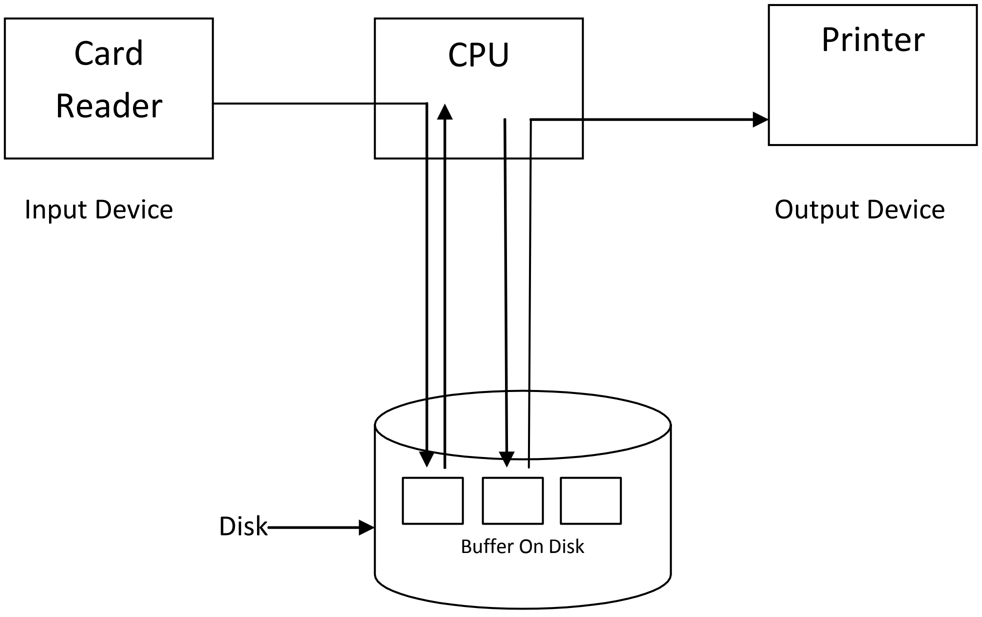
Batch System Memory Layout

- Batch monitor – read, interpret and execute commands and batch jobs are executed one at a time.
- A job may consist of compilation and execution of user program where each particular step to be performed is indicated by monitor with appropriate command.
- When JOB_END command encounter monitor may look for another job which may be identified by JOB_START command.
- With the sequencing of program execution automated by batch processing emerging performance decrease due to speed gap between fast processor and slow I/O devices such as card reader and printers.

- To improve performance and reduce speed gap between processing and I/O device, buffering and spooling is used.

2. Buffering

- I/O subsystems have allowed overlapping of program execution with input/output operations of next job.
- For example, a portion of memory may be set for I/O buffering.
- During execution of program, I/O section concurrently read input cards of next job into a buffer. When that job is schedule for execution, the processor can obtain its input from fast – memory (buffer).
- Similarly, output of job may be buffered and subsequently printed output concurrently with the execution of next job.



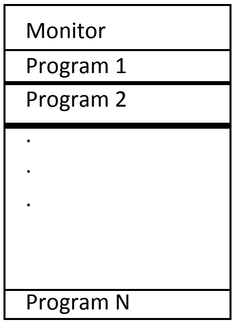
Spooling

3. SPOOL (Simultaneously Peripheral Operation OnLine)

- ✓ It is most sophisticated forms of I/O buffering.
- ✓ It uses disk to temporarily store input and output of jobs.
- ✓ Card – to disk and disk – to printer operation for subsequent and previous programs respectively are performed concurrently by spooling monitor with execution of current program.
- ✓ Both Input and output of several jobs may be queued on disk as necessary to obtain high processor utilization rate.

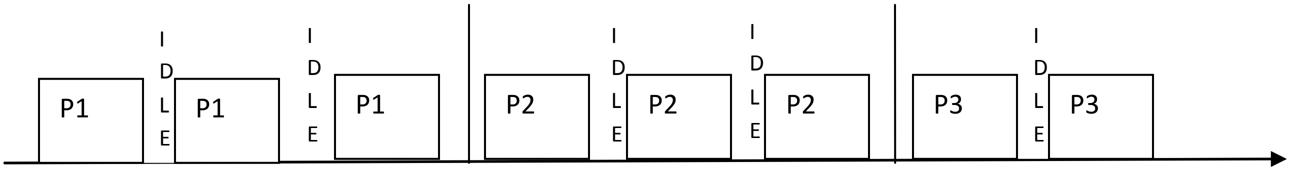
Multiprogramming

- Buffering and spooling improve system performance by overlapping the input, output and computation of a single job, but both of them have their limitations.
- A single user cannot always keep CPU or I/0 devices busy at all times.
- Multiprogramming offers a more efficient approach to increase system performance. In order to increase the resource utilization, systems supporting multiprogramming approach allow more than one job (program) to utilize CPU time at any moment. More number of programs competing for system resources, better will be resource utilization.
- The idea is implemented as follows. The main memory of a system contains more than one program.



Memory Layout for multiprogramming system

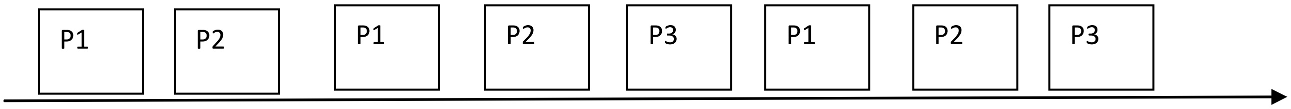
- ➢ In Single programming, the operating system picks one of the programs and start executing. During execution process program 1 may need some 110 operation to complete. In a sequential execution environment the CPU would sit idle.



Processing activity of P1, P2 and P3

Sequential execution of program 1 2 and 3

- ➢ In a multiprogramming system, operating system will simply switch over to the next program (Program2).
- ➢ When that program needs to wait for some I/0 operation, it switches over to Program 3 and so on. If there is no other new program left in the main memory, the CPU will pass its control back to the previous programs.



Concurrent execution of program 1 2 and 3

- ➢ Multiprogramming has traditionally been employed to increase the resource utilization of a computer system and to support multiple simultaneously interactive users (terminals).

Need of an Operating System

The OS typically provides services (Functions) in the following areas:

I. **Program Creation**

- ➢ The operating system provide variety of facility and services such as editors and debuggers through which programmer create program.
- ➢ Typically these types of service provider software is called "Utility programs or software" that are not actually part of operating system but to access these software required operating system.

II. **Program Execution**

- ➢ A number of task need to be perform to execute program instructions and data must be load into memory.
- ➢ I/O devices and file must be initialized and other resourced must be prepared.

- Operating system handles such type of task for the user.

III. **Access to I/O device**

- Each I/O device require its own particular set of instruction or control signals for operations
- Operating system takes care of that so programmer can read and write.

IV. **Control access to files**

- It maintains Structure of file format on storage medium.
- In multiuser, operating system can provide protection mechanism to prevent unauthorized file access.

V. **System access**

- In the case of share and public system, operating system has control on resource and whole system.

VI. **Error Detection and response**

- A variety of errors occur while computer system is in use
- Operating system can detect internal or external error such as hardware failure, memory error, device failure error or software error or inability of operating system to grant request of an application program, etc…

VII. **Accounting**

- Operating system monitorsall process performance such as response time, waiting time CPU utilization time turn –around time and it is useful for another system or future enhancement to improve performance.

Single User & Multi User Operating System

- Operating System is system software. This is software acts as an interface between the user and the computer. It also controls and coordinates different operations of computer. These operating systems are categorized of basis of users and their tasks.
- Base on User and Task

1. **Single user, Single task**

- This operating system is designed to manage the computer so that one user can effectively do one thing at a time.
- The Palm OS for Palm handheld computers is a good example of a modern single-user, single-task operating system.
- When you are using MS-DOS it is a single user single task operating system.

2. **Single user, Multi-Tasking**

- This is the type of operating system most people use on their desktop and laptop computers today.
- Microsoft's Windows and Apple's MacOS platforms are both examples of operating systems that will let a single user have several programs in operation at the same time. For example, it's entirely possible for a Windows user to write a letter and at the same time the printing of another letter can be done or at the same Internet browsing can be done. So in Single user operating systems there is one keyboard and one monitor that you interact with.
- Consider a typical home computer. There is a single keyboard and mouse that accept input commands, and a single monitor to display information output. There may also be a printer for the printing of documents and images.
- In essence, a single-user operating system provides access to the computer system by a single user at a time. If another user needs access to the computer system, they must wait till the current user finishes what they are doing and leaves.

3. **Multi-User**

- A multi-user operating system allows many different users to take advantage of the computer's resources simultaneously.

- The operating system must make sure that the requirements of the various users are balanced, and that each of the programs they are using has sufficient and separate resources so that a problem with one user doesn't affect the entire community of users.
- UNIX, VMS and mainframe operating systems are examples of multi-user operating systems.
- A multi-user operating system lets more than one user access the computer system at one time. Access to the computer system is normally provided via a network, so that users access the computer remotely using a terminal or other computer.
- These terminals now a day are generally personal computers and use a network to send and receiveinformation to the multi-user computer system.
- Examples of multi-user operating systems are UNIX, Linux and mainframes such as the IBM AS400.
- The multi user operating systems must manage and run all user requests, ensuring they do not interfere with each other. Devices which can only be used by one user at a time, like printers and disks must be shared amongst all those requesting them so that all the output documents are not mixed up.
- If each user tried to send their document to the printer at the same time, the end result would be garbage. Instead, documents sent are placed in a queue, and each document is printed in its entirety before the next document to be printed is retrieved from the queue.
- It is similar to a situation where in you are waiting for your turn in a ticket counter to get a ticket. The ticket issuer issues the ticket when your turn comes. Here also all the printing jobs wait in a queue and jobs are printed one after the other. Some priority can also be set to some jobs so that they can be taken up early according to some priority.

Elements of an Operating System

An operating system is broken down into five layers.

1. **Kernel**

- The kernel is the heart of the operating system.
- 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**

- Its responsibility is 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**

- All the physical communication between your computer's hardware, such as disk drives, keyboards, mousse, screens and so on, takes place.

4. **File Management**

- To control how the files on your computer's hard drive are stored and accessed by any application seeking to use them.

5. **User Interface**

- The first thing you see when your operating system has logged you in it is user interface.
- Its jobis to provide a means for the user to actually interact with the rest of the component and as well whole system.
- There are two different types of User interfaces.
- The first one is you are most familiar with, the graphical user interface, which is where you see windows and icons for each of your files and so on.
- The second is a command line interface, or text based interface where a user would interact with the system using text based commands.

- The second is a command line interface, or text based interface where a user would interact with the system using text based commands. E.g. DOS os

Operating System as a Resource Manager

- Primary goal of operating system is to providing convenient interface to user.
- For that, Job of operating system is to provide an orderly and controlled allocation of processors, memories and I/O devices among various programs to complete them.
- For example, if 3 programs running on some computer all tired to print their output simultaneously o the same printer. The first few linesmight be from program1; next few from program 2 then some program 3 and so on. The result would be messy.
- Operating system can bring order by buffering all output for printer on disk.
- When 1st program is finished, the operating system can then copy its output from disk to where it has been stored to printer, while at the same time other program can continue generating more output.
- When computer has multiuser, the need for managing and protecting memory, I/O devices and other resources is even greater.
- In addition, in multiuser environment user not only share hardware, it also share information such as file, database, data, etc… as well.
- Where operating system is to keep track of who is using which resource, to grant resource request, to account for usage and to mediate conflicting request from different program and users.
- Resource management includes multiplexing (sharing) resources in 2 ways:

1. **In time**

- When a resource is time multiplexed, different programs or user takes turn using it.
- 1st one of them to use resource then another and so on.
- For example, one CPU and multiple program want to run on it, operating system first allocate CPU to one program, the after it has run long enough, another one gets to use CPU, then another and then eventually first one again.
- Determine how the resource is time multiplexed , who goes next, and for how long is the task of operating system.

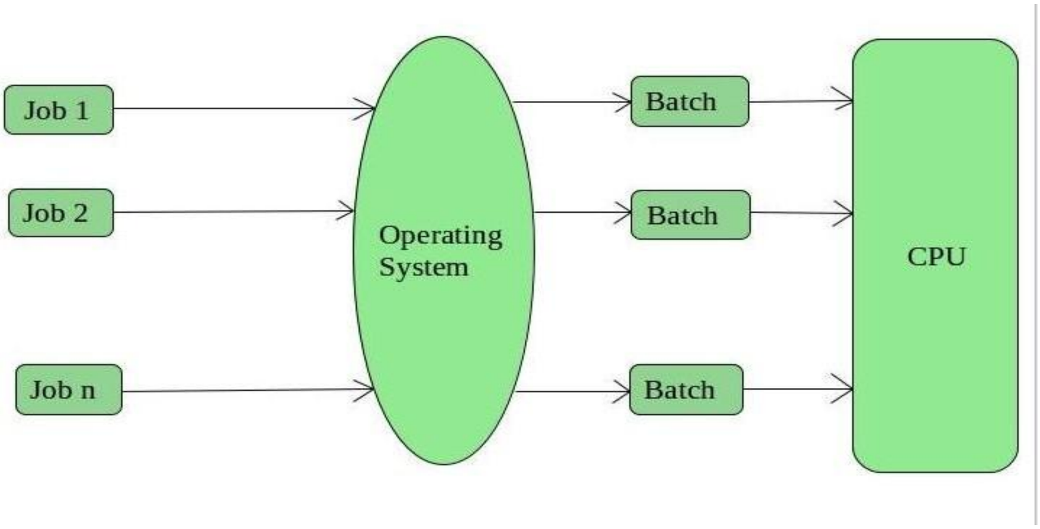
2. **In space**

- Instead of program taking turns, each one gets part of the resource.
- For example, main memory is normally divided up among several running programs, so each can be reside at the same time.
- It is more efficient to hold several programs in memory at once rather than giving one of them all, these raise issues of fairness, protection and so no and it is up to operating system to solve them.
- In many systems, single disk can hold files from many user at a time. Allocating disk space and keeping track of who is using which disk block is a typical operating system resource management task

Types of Operating System

1. **Batch Operating system:**

- This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having same requirement and group them into batches. It is the responsibility of operator to sort the jobs with similar needs.



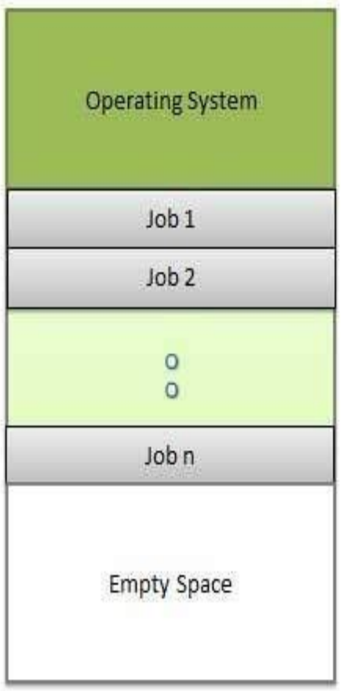
- ❖ **Advantages of Batch Operating System:**
  - ➢ It is very difficult to guess or know the time required by any job to complete. Processors of the batch systems know how long the job would be when it is in queue
  - ➢ Multiple users can share the batch systems
  - ➢ The idle time for batch system is very less
  - ➢ It is easy to manage large work repeatedly in batch systems
- ❖ **Disadvantages of Batch Operating System:**
  - ➢ The computer operators should be well known with batch systems
  - ➢ Batch systems are hard to debug
  - ➢ It is sometime costly
  - ➢ The other jobs will have to wait for an unknown time if any job fails

Examples of Batch based Operating System: Payroll System, Bank Statements etc.

2. Multiprogramming operating System

- ➢ Sharing the processor, when two or more programs reside in memory at the same time, is referred as multiprogramming.
- ➢ Multiprogramming assumes a single shared processor.
- ➢ Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute.

The following figure shows the memory layout for a multiprogramming system.



An OS does the following activities related to multiprogramming.

- ➢ The operating system keeps several jobs in memory at a time.
- ➢ This set of jobs is a subset of the jobs kept in the job pool.
- ➢ The operating system picks and begins to execute one of the jobs in the memory.
- ➢ Multiprogramming operating systems monitor the state of all active programs and system resources using memory management programs to ensures that the CPU is never idle, unless there are no jobs to process.

Advantages

- ➢ High and efficient CPU utilization.
- ➢ User feels that many programs are allotted CPU almost simultaneously.

Disadvantages

- ➢ CPU scheduling is required.
- ➢ To accommodate many jobs in memory, memory management is required.

Time-Sharing Operating System

- It is multiprogramming and multi-user system which operates in interactive mode with quick terminal response time.
- Time sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time.
- User send request to computer through keyboard then computer process that request and response is display on user's terminal.
- Processor's time which is shared among multiple users simultaneously is termed as time-sharing.
- The main difference between Multiprogramming Batch Systems and Time-Sharing Systems is that in case of Multiprogramming batch systems, objective is to maximize processor use, whereas in Time-Sharing Systems objective is to minimize response time.
- Multiple jobs are executed by the CPU by switching between them, but the switches occur so frequently. Thus, the user can receive an immediate response.
- For example, in a transaction processing, processor execute each user program in a short burst or quantum of computation. That is if n users are present, each user can get time quantum. When the user submits the command, the response time is in few seconds at most.
- In order to prevent a program from monopolizing the processor, a program executing longer than system define time quantum is incorporated by operating system and place at the end of queue of waiting of program for the process.
- Memory management in time- sharing system provides mechanism for protection and separation of address space of user processes.
- I/O management feature in time-sharing system must be able to handle multi-user terminals.
- As required by most multi user environment allocation and deallocation of devices must be performed in a manner that preserve system integrity and provide good performance.

Advantages of Timesharing operating systems are following

- Provide advantage of quick response.
- Avoids duplication of software.
- Reduces CPU idle time.

Disadvantages of Timesharing operating systems are following.

- Problem of reliability.
- Question of security and integrity of user programs and data.
- Problem of data communication.

Real- Time Operating System(RTOS)

- Real-Time operating systems are used in environment where a large number of events (mostly external) occurred and must be accepted and processes in a short time or within certain deadline to computer system.
- For example Industrial control, telephone switching equipment, flight control, etc… are RTOS.
- In real-time system Time is a key parameter.
- Primary objective of real time system is to provide quick response time and thus meet scheduling deadline.
- User convenience and resource utilization is secondary concern to real time system designers.
- Real time processing is always on line whereas on line system need not be real time.
- The time taken by the system to respond to an input and display of required updated information is termed as response time. So in this method response time is very less.

- Real-time operating system has well-defined, fixed time constraints otherwise system will fail. For example Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, and home-appliance controllers, Air traffic control system etc.
- Programmer –defined and controlled processes are commonly encountered in real –time system.
- Separate process is charged with handling a single external event which is often signaled by interrupt.
- Each process assigned a certain level of priority that corresponds to the relative importance of event that it services.
- Processor normally allocated to the higher priority process among those that are ready to execute.
- Higher priority processes usually preempt execution of the lower- priority processes this form of scheduling is called priority- based preemptive scheduling, is used by majority of real-time system
- Memory management is comparatively less demanding than in other types of multiprogramming systems the primary reason for that many processes permanently reside in memory to provide quick response time.
- Critical device management is one of the main characteristic of real time system.
- To providing sophisticated forms of interrupt management and I/O buffering, real-time system often provide system calls to allow user programs to connect themselves to interrupt vectors and to service event directly.
- File management is usually found only in larger installation of real-time system.

There are two types of real-time operating systems.

1. **Hard real-time systems**

- Hard real-time systems guarantee that critical tasks complete on time.
- In hard real-time systems secondary storage is limited or missing with data stored in ROM.
- In these systems virtual memory is almost never found.

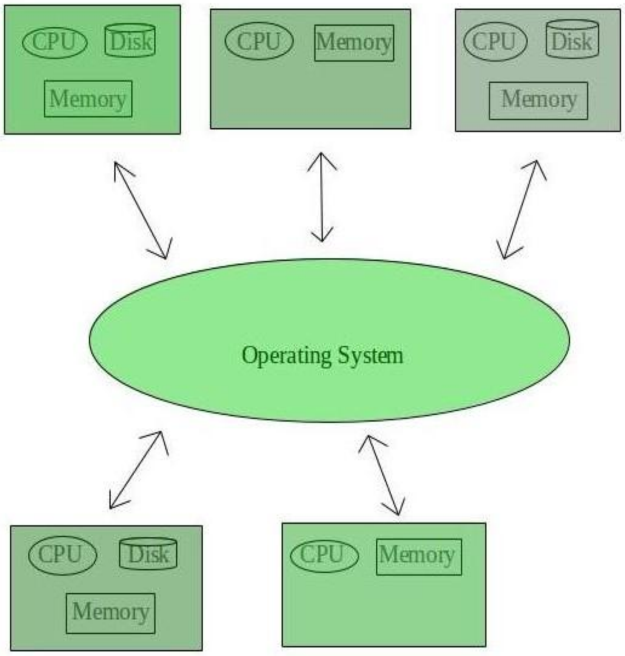
2. **Soft real-time systems**

- Soft real time systems are less restrictive. Critical real-time task gets 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.

Combination Operating System

- Different types of operating systems are optimized or at least largely prepare for serving the needs of specific environments.
- For e.g. both interactive program development and lengthy simulations are often counter in university computing centers.
- For this reason some commercial operating system provides a combination of described services.
- For. E.g. A time sharing system may support interactive users and also incorporate full fledge of batch monitor.
- This allows computationally intensive non-interactive programs to be run concurrently with interactive programs.

3. Distributed Operating System



- These types of operating system is a recent advancement in the world of computer technology and are being widely accepted all-over the world and, that too, with a great pace.
- Various autonomous interconnected computers communicate each other using a shared communication network.
- Independent systems possess their own memory unit and CPU.
- These are referred as **loosely coupled systems** or **distributed systems**.
- These system's processors differ in size and function.
- The major benefit of working with these types of operating system is that it is always possible that one user can access the files or software which are not actually present on his system but on some other system connected within this network i.e., remote access is enabled within the devices connected in that network.

Advantages of Distributed Operating System:

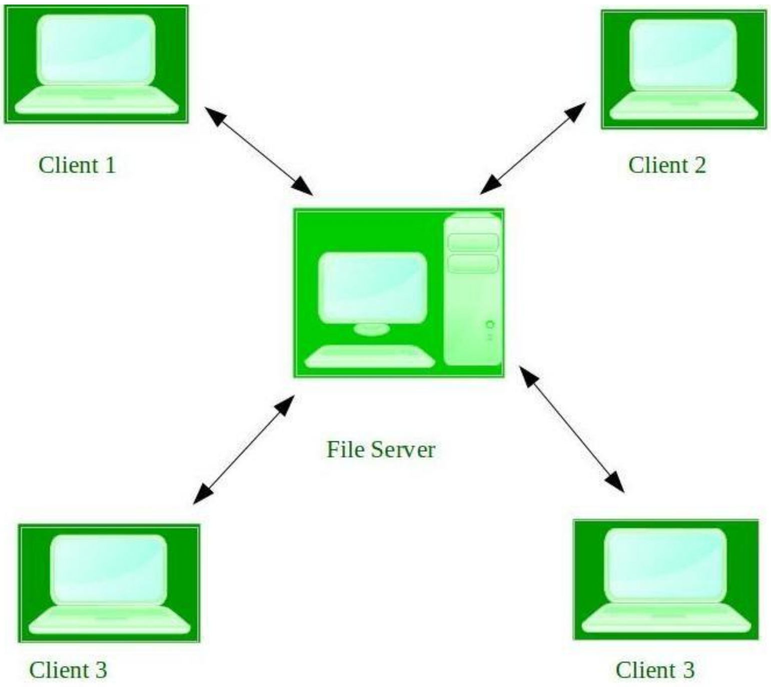
- Failure of one will not affect the other network communication, as all systems are independent from each other
- Electronic mail increases the data exchange speed
- Since resources are being shared, computation is highly fast and durable
- Load on host computer reduces
- These systems are easily scalable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

Disadvantages of Distributed Operating System:

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

4. Network Operating System –

- These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions.
- These types of operating systems allow shared access of files, printers, security, applications, and other networking functions over a small private network.
- One more important aspect of Network Operating Systems is that all the users are well aware of the underlying configuration, of all other users within the network, their individual connections etc. and that's why these computers are popularly known as **tightly coupled systems**.



Advantages of Network Operating System:

- Highly stable centralized servers
- Security concerns are handled through servers
- New technologies and hardware up-gradation are easily integrated to the system
- Server access are possible remotely from different locations and types of systems

Disadvantages of Network Operating System:

- Servers are costly
- User has to depend on central location for most operations
- Maintenance and updates are required regularly
- Examples of Network Operating System are: Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD etc.