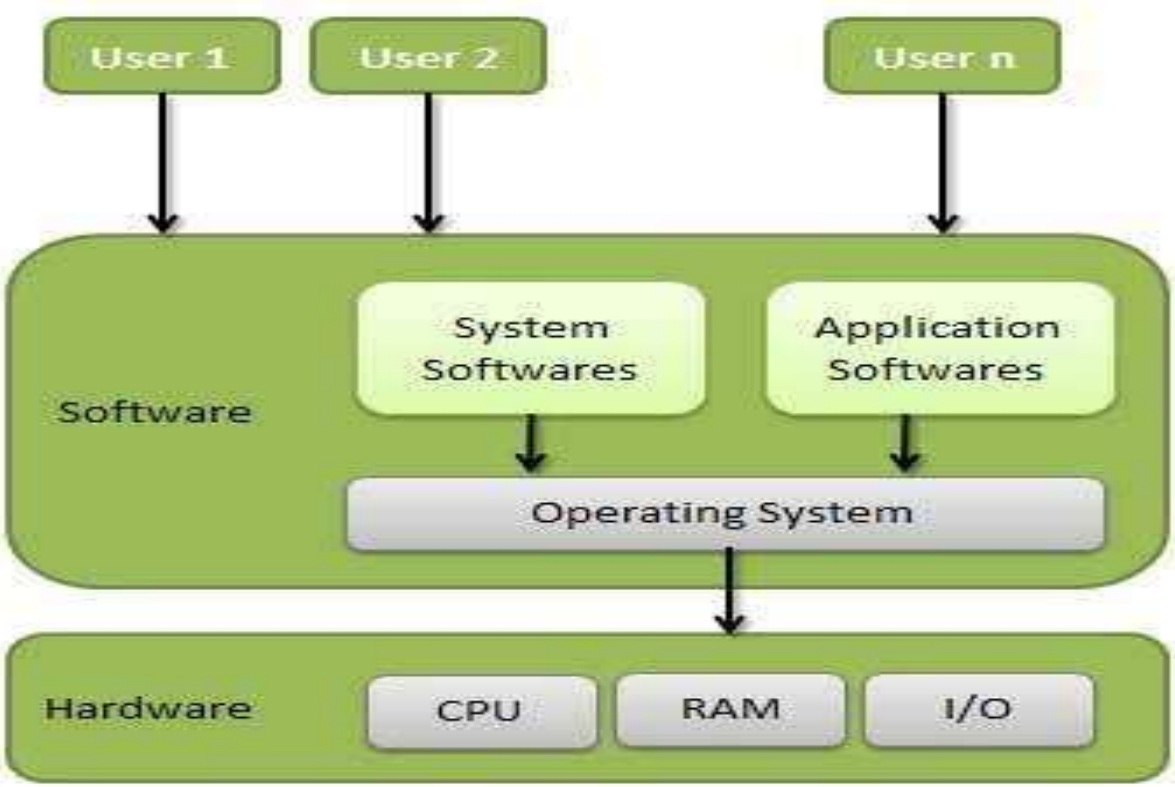
Computer software

Unit 6

Operating system

- An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
- Some popular Operating Systems include Linux, Windows, OS X, VMS, OS/400, AIX, z/OS, etc.


User 1
User 2
User n
Software
System Softwares
Application Softwares
Operating System
Hardware
CPU
RAM
I/O

Following are some of important functions of an operating System.

- Memory Management
- Process Management
- Device Management
- File Management
- Security
- User Interface
- Control over system performance
- Job accounting
- Error detecting aids
- Coordination between other software and users

Memory Management

- Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.
- Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must in the main memory. An Operating System does the following activities for memory management –
- Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.
- In multiprogramming, the OS decides which process will get memory when and how much.
- Allocates the memory when a process requests it to do so.
- De-allocates the memory when a process no longer needs it or has been terminated.

Process Management

- A program in execution is called a process. In order to accomplish its task, process needs the computer resources.
- There may exist more than one process in the system which may require the same resource at the same time. Therefore, the operating system has to manage all the processes and the resources in a convenient and efficient way.
- Some resources may need to be executed by one process at one time to maintain the consistency otherwise the system can become inconsistent and deadlock may occur.

The operating system is responsible for the following activities in connection with Process Management

- Scheduling processes and threads on the CPUs.
- Creating and deleting both user and system processes.
- Suspending and resuming processes.
- Providing mechanisms for process synchronization.
- Providing mechanisms for process communication.

Device Management

- An Operating System manages device communication via their respective drivers. It does the following activities for device management –
- Keeps tracks of all devices. Program responsible for this task is known as the **I/O controller**.
- Decides which process gets the device when and for how much time.
- Allocates the device in the efficient way.
- De-allocates devices.

File Management

- A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directories.
- An Operating System does the following activities for file management –
- Keeps track of information, location, uses, status etc. The collective facilities are often known as **file system**.
- Decides who gets the resources.
- Allocates the resources.
- De-allocates the resources.

Protection and security:

- Protection refers to a mechanism or a way to control the access of programs, processes, or users to the resources defined by a computer system. Following are the major activities of an operating system with respect to protection
- The OS ensures that all access to system resources is controlled.
- The OS ensures that external I/O devices are protected from invalid access attempts.
- The OS provides authentication features for each user by means of passwords.

User Interface:

- User interface helps to communicate between the user and the computer.
- Its types are:

1) Command line Interface(CLI)
2) Graphical User Interface(GUI)

- **Control over system performance** – Recording delays between request for a service and response from the system.
- **Job accounting** – Keeping track of time and resources used by various jobs and users.
- **Error detecting aids** – Production of dumps, traces, error messages, and other debugging and error detecting aids.
- **Coordination between other softwares and users** – Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

Types of Operating System

- **Batch operating system**
- **Time-sharing operating systems**
- **Distributed operating System**
- **Network operating System**
- **Real Time operating System**
  - **Hard real-time systems**
  - **Soft real-time systems**

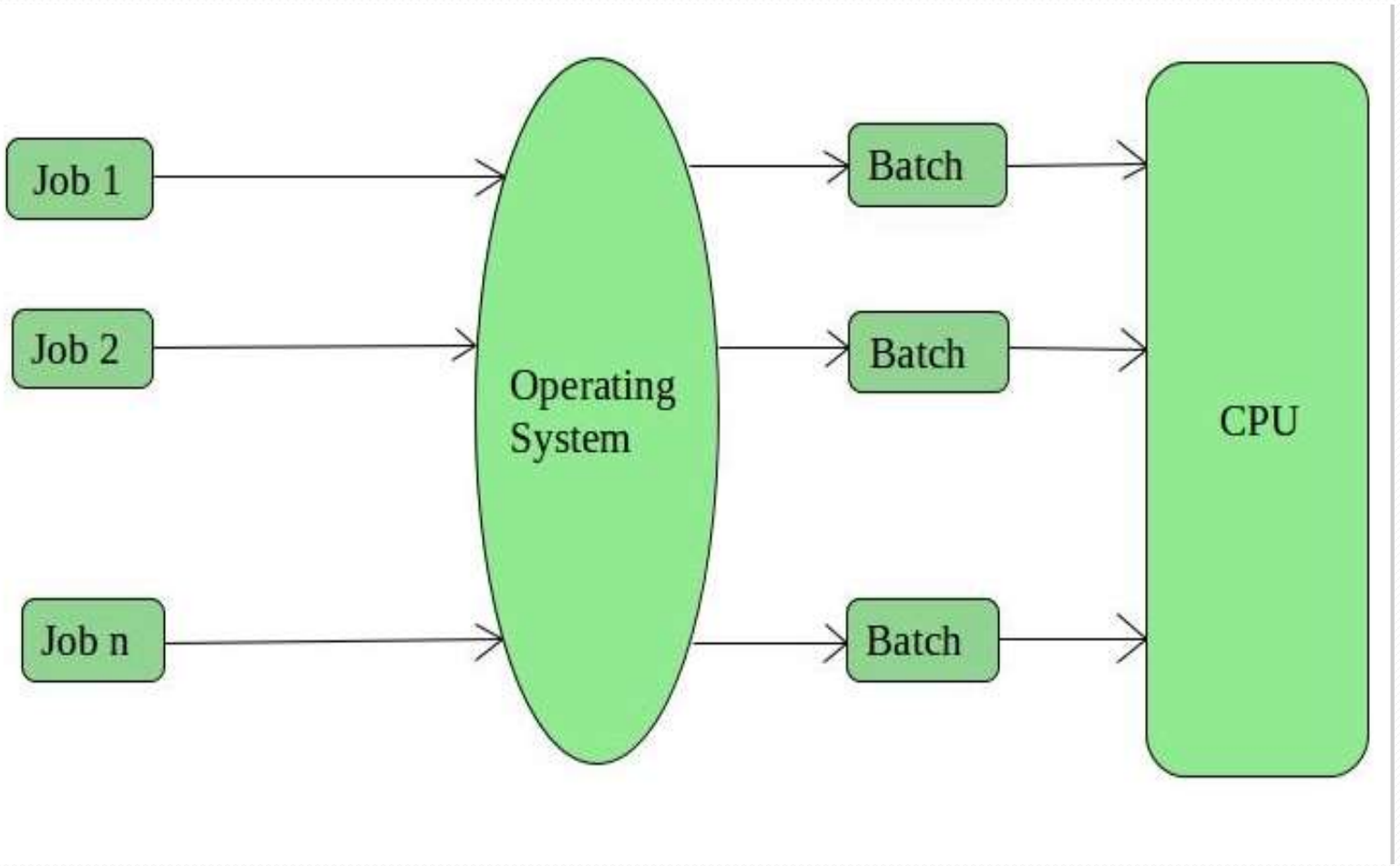
1) Batch operating system

- The users of a batch operating system do not interact with the computer directly. Each user prepares his job on an off-line device like punch cards and submits it to the computer operator. To speed up processing, jobs with similar needs are batched together and run as a group. The programmers leave their programs with the operator and the operator then sorts the programs with similar requirements into batches.
- **Examples of Batch based Operating System:** Payroll System, Bank Statements etc.

The problems with Batch Systems are as follows –

- The computer operators should be well known with batch systems
- Batch systems are hard to debug
- The other jobs will have to wait for an unknown time if any job fails.

Fig. Batch operating system



2) Time-Sharing Operating Systems

Each task has given some time to execute, so that all the tasks work smoothly. Each user gets time of CPU as they use single system. These systems are also known as Multitasking Systems. The task can be from single user or from different users also. The time that each task gets to execute is called quantum. After this time interval is over OS switches over to next task.

Cont.

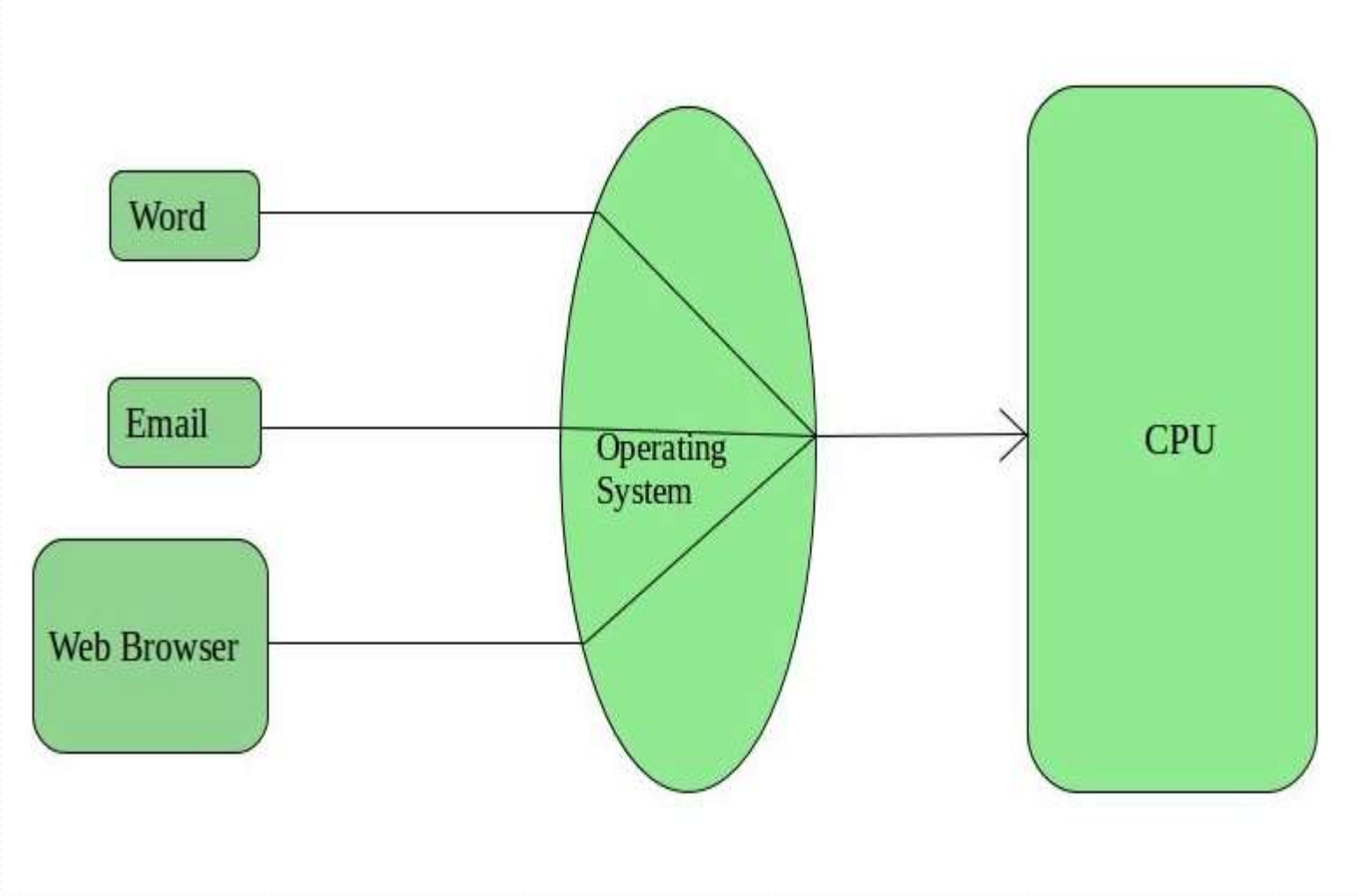
Advantages of Time-Sharing OS:

- Each task gets an equal opportunity
- Less chances of duplication of software
- CPU idle time can be reduced

Disadvantages of Time-Sharing OS:

- Reliability problem
- One must have to take care of security and integrity of user programs and data
- Data communication problem
- **Examples of Time-Sharing OSs are:** Multics, Unix etc.

Cont.



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(speed). Various autonomous(independent) interconnected computers communicate each other using a shared communication network. Independent systems possess their own memory unit and CPU. These systems processors differ in sizes and functions.
- 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.

Cont.

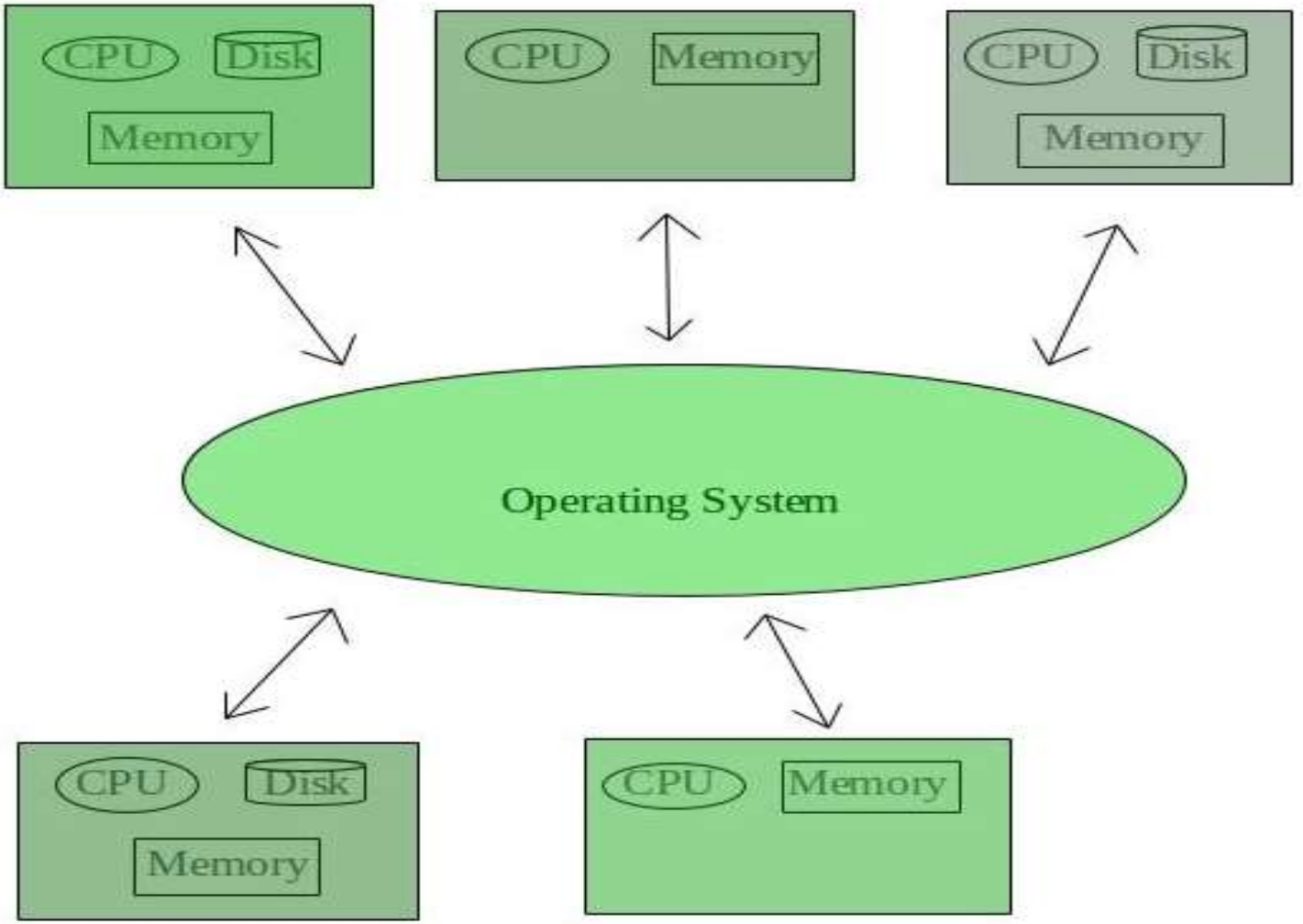
Advantages of Distributed Operating System:

- Failure of one will not affect the other network communication, as all systems are independent from each other
- 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
- **Examples of Distributed Operating System are-** LOCUS etc.

Fig. Distributed Operating System



4) Network Operating System

- These systems runs on a server and provides the capability to manage data, users, groups, security, applications, and other networking functions. These type of operating systems allows shared access of files, printers, security, applications, and other networking functions over a small private network.

Cont.

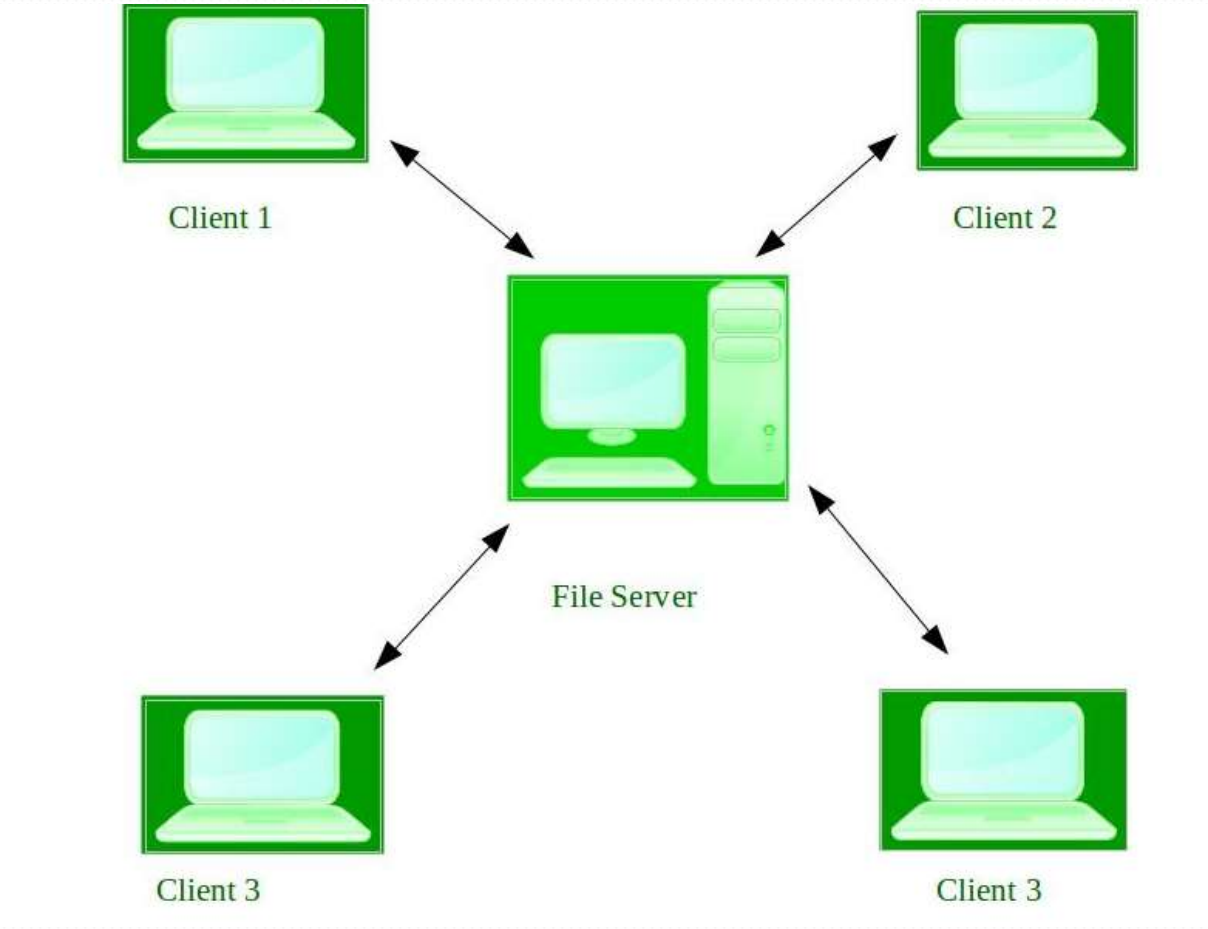
Advantages of Network Operating System:

- Highly stable centralized servers
- Security concerns are handled through servers
- 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.

Fig. Network Operating System –



5. Real-Time Operating System –

- These types of OSs serves the real-time systems. The time interval required to process and respond to inputs is very small. This time interval is called **response time**.
- **Real-time systems** are used when there are time requirements are very strict like missile systems, air traffic control systems, robots etc.
- **Two types of Real-Time Operating System which are as follows:**
- **Hard Real-Time Systems:**
  These OSs are meant for the applications where time constraints are very strict and even the shortest possible delay is not acceptable. These systems are built for saving life like automatic parachutes or air bags which are required to be readily available in case of any accident.
- **Soft Real-Time Systems:**
  These OSs are for applications where for time-constraint is less strict. Eg: web.

Advantages of RTOS:

- **Maximum Consumption:** Maximum utilization of devices and system,thus more output from all the resources
- **Task Shifting:** Time assigned for shifting tasks in these systems are very less. For example in older systems it takes about 10 micro seconds in shifting one task to another and in latest systems it takes 3 micro seconds.
- **Focus on Application:** Focus on running applications and less importance to applications which are in queue.
- **Real time operating system in embedded system:** Since size of programs are small, RTOS can also be used in embedded systems like in transport and others.
- **Error Free:** These types of systems are error free.

Disadvantages of RTOS:

- **Complex Algorithms:** The algorithms are very complex and difficult for the designer to write on.
- **Device driver and interrupt signals:** It needs specific device drivers and interrupt signals to response earliest to interrupts.
- **Examples of Real-Time Operating Systems are:** Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

6) Multiprogramming OS

- In a multiprogramming system there are one or more programs loaded in main memory which are ready to execute. Only one program at a time is able to get the CPU for executing its instructions while all the others are waiting their turn.
  The main idea of multiprogramming is to maximize the use of CPU time. Indeed, suppose the currently running process is performing an I/O task . Then, the OS may interrupt that process and give the control to one of the other in-main-memory programs that are ready to execute (i.e. *process context switching*). In this way, no CPU time is wasted by the system waiting for the I/O task to be completed, and a running process keeps executing until either it voluntarily releases the CPU or when it blocks for an I/O operation. Therefore, the ultimate goal of multiprogramming is to keep the CPU busy as long as there are processes ready to execute.

7) Multiprocessing/Parallel Processing OS

- Multiprocessing sometimes refers to executing multiple processes (programs) at the same time. Multiprocessing refers to the *hardware* (i.e., the CPU units) rather than the *software* (i.e., running processes). If the underlying hardware provides more than one processor then that is multiprocessing.

Process

- A process is a program in execution. The execution of a process must progress in a sequential fashion. Definition of process is following.
- A process is defined as an entity which represents the basic unit of work to be implemented in the system.

Components of process are following.

S.N.	Component & Description
1	**Object Program** Code to be executed.
2	**Data** Data to be used for executing the program.
3	**Resources** While executing the program, it may require some resources.
4	**Status** Verifies the status of the process execution.A process can run to completion only when all requested resources have been allocated to the process. Two or more processes could be executing the same program, each using their own data and resources.

Program

- A program by itself is not a process. It is a static entity made up of program statement while process is a dynamic entity. Program contains the instructions to be executed by processor.
- A program takes a space at single place in main memory and continues to stay there. A program does not perform any action by itself.

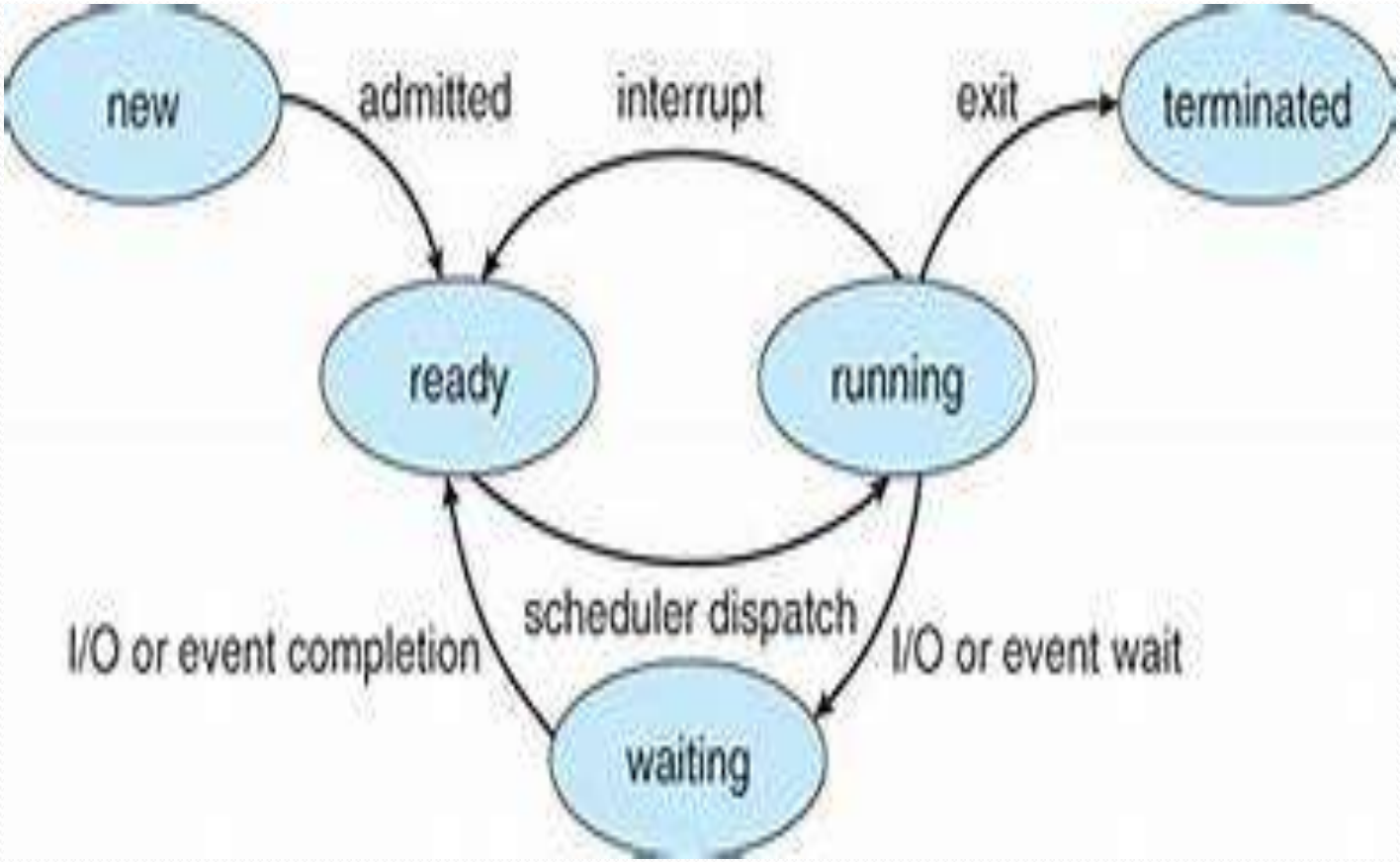
Process States

- As a process executes, it changes state. The state of a process is defined as the current activity of the process.
- Process can have one of the following five states at a time which are given below:

1. **New**
2. **Ready**
3. **Running**
4. **Waiting**
5. **Terminated**

S.N.	State & Description
1	**New** The process is being created.
2	**Ready** The process is waiting to be assigned to a processor. Ready processes are waiting to have the processor allocated to them by the operating system so that they can run.
3	**Running** Process instructions are being executed (i.e. The process that is currently being executed).
4	**Waiting** The process is waiting for some event to occur (such as the completion of an I/O operation).
5	**Terminated**

Fig: Process State diagram



CPU Scheduling

- **In Multiprogramming systems**, the Operating system schedules the processes on the CPU to have the maximum utilization of it and this procedure is called **CPU scheduling**.
- The Operating System uses various scheduling algorithm to schedule the processes. The different CPU scheduling algorithm are given below:

1) First come First served(FCFS) Scheduling
2) Shortest Job First(SJF) Scheduling
3) Round Robin(RR) Scheduling.
4) Shortest remaining time first Scheduling
5) Priority based Scheduling

1) First come First served(FCFS) Scheduling

- It is the simplest algorithm to implement. The process with the minimal arrival time will get the CPU first. The lesser the arrival time, the sooner will the process gets the CPU. It is the non-preemptive type of scheduling.

2) Shortest Job First

- The job with the shortest burst time will get the CPU first. The lesser the burst time, the sooner will the process get the CPU. It is the non-preemptive type of scheduling.

3) **Round Robin**

- In the Round Robin scheduling algorithm, the OS defines a time quantum (slice). All the processes will get executed in the cyclic way. Each of the process will get the CPU for a small amount of time (called time quantum) and then get back to the ready queue to wait for its next turn. It is a preemptive type of scheduling.

4) Shortest remaining time first

- It is the preemptive form of SJF. In this algorithm, the OS schedules the Job according to the remaining time of the execution.

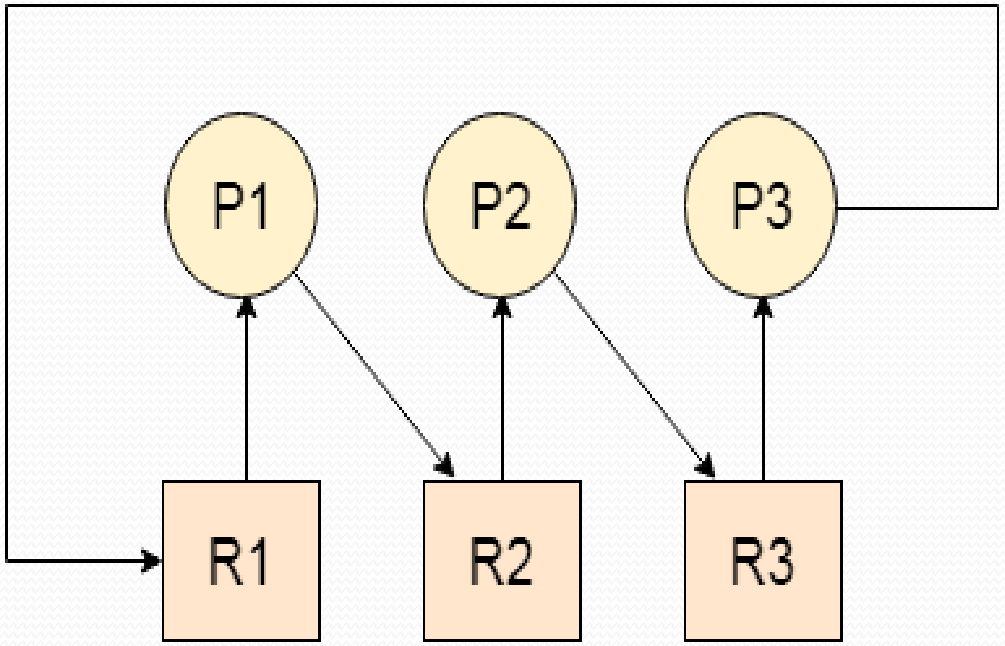
5) Priority based scheduling

- In this algorithm, the priority will be assigned to each of the processes. The higher the priority, the sooner will the process get the CPU. If the priority of the two processes is same then they will be scheduled according to their arrival time.

Deadlock:

- A Deadlock is a situation where each of the computer process waits for a resource which is being assigned to some another process. In this situation, none of the process gets executed since the resource it needs, is held by some other process which is also waiting for some other resource to be released.
- Let us assume that there are three processes P1, P2 and P3. There are three different resources R1, R2 and R3. R1 is assigned to P1, R2 is assigned to P2 and R3 is assigned to P3.
- After some time, P1 demands for R2 which is being used by P2. P1 halts its execution since it can't complete without R2. P2 also demands for R3 which is being used by P3. P2 also stops its execution because it can't continue without R3. P3 also demands for R1 which is being used by P1 therefore P3 also stops its execution.
- In this scenario, a cycle is being formed among the three processes. None of the process is progressing and they are all waiting. The computer becomes unresponsive since all the processes got blocked.

Fig. Deadlock



Necessary conditions for Deadlocks

1)Mutual Exclusion

- A resource can only be shared in mutually exclusive manner. It implies, if two process cannot use the same resource at the same time.

2) Hold and Wait

- A process waits for some resources while holding another resource at the same time.

3) No preemption

- The process which once scheduled will be executed till the completion. No other process can be scheduled by the scheduler meanwhile.

4) Circular Wait

- All the processes must be waiting for the resources in a cyclic manner so that the last process is waiting for the resource which is being held by the first process.

Strategies for handling Deadlock :

1. Deadlock Ignorance
2. Deadlock prevention
3. Deadlock avoidance
4. Deadlock detection and recovery

1) Ignore deadlock:

- Deadlock Ignorance is the most widely used approach among all the mechanism. This approach is best suitable for a single end user system where User uses the system only for browsing and all other normal stuff.
- The operating systems like Windows and Linux mainly focus upon performance. However, the performance of the system decreases if it uses deadlock handling mechanism all the time if deadlock happens 1 out of 100 times then it is completely unnecessary to use the deadlock handling mechanism all the time.
- In these types of systems, the user has to simply restart the computer in the case of deadlock. Windows and Linux are mainly using this approach.

2) Deadlock prevention

- Deadlock happens only when Mutual Exclusion, hold and wait, No preemption and circular wait holds simultaneously. If it is possible to violate one of the four conditions at any time then the deadlock can never occur in the system.

3) Deadlock avoidance

- In deadlock avoidance, the operating system checks whether the system is in safe state or in unsafe state at every step which the operating system performs. The process continues until the system is in safe state. Once the system moves to unsafe state, the OS has to backtrack one step.

4) Deadlock detection and recovery

- This approach let the processes fall in deadlock and then periodically check whether deadlock occur in the system or not. If it occurs then it applies some of the recovery methods to the system to get rid of deadlock.

Paging:

- In Operating Systems, Paging is a storage mechanism used to retrieve processes from the secondary storage into the main memory in the form of pages.
- The main idea behind the paging is to divide each process in the form of pages. The main memory will also be divided in the form of frames.
- One page of the process is to be stored in one of the frames of the memory. The pages can be stored at the different locations of the memory but the priority is always to find the contiguous frames or holes.
- Pages of the process are brought into the main memory only when they are required otherwise they reside in the secondary storage.

Virtual Memory

- Virtual Memory is a storage scheme that provides user an illusion of having a very big main memory. This is done by treating a part of secondary memory as the main memory.
- In this scheme, User can load the bigger size processes than the available main memory by having the illusion that the memory is available to load the process.
- Instead of loading one big process in the main memory, the Operating System loads the different parts of more than one process in the main memory.
- By doing this, the degree of multiprogramming will be increased and therefore, the CPU utilization will also be increased.

User Interface:

- User interface helps to communicate between the user and the computer.
- Its types are:

1) Command line Interface(CLI)
2) Graphical User Interface(GUI)

Command line Interface(CLI)

- CLI requires the user to interact with OS in the form of text keyed in from the keyboard.
- User has to remember all the commands to perform the task.
- Eg are MS-DOS and Linux shell .

Graphical User Interface(GUI)

- It uses graphics to give the commands.
- The interface consists of icons, menus, windows and pointers.
- The user do not need to remember the commands.
- Eg windows 7, windows 10

Example of os:

- MS-DOS
- Windows Family OS.
- Linux OS.

Linux OS

- It was developed by Linus Torvalds in 1992. Linux is copyright under the GNU("**GNU's Not Unix**!") public License.
- Linux is free operating system that is easily available.
- Linux is command line user interface OS, Linux has GUI interaces called desktop environments like GNOME(GNU Object Model Environment).
- Its vendor are Red Hat, Mandrake etc
- It is reliable and secure OS.
- It supports multi-tasking, multi processing.
- Eg Ubuntu, Redhat linux, kali linux etc.