UNIT-2

What is an Operating System?

An **Operating System** is a System software that manages all the resources of the computing device.

- Acts as an interface between the software and different parts of the computer or the computer hardware.
- Manages the overall resources and operations of the computer.
- Controls and monitors the execution of all other programs that reside in the computer, which also includes application programs and other system software of the computer.
- Examples of Operating Systems are Windows, Linux, macOS, Android, iOS, etc.

Operating System Goals:

- Run user applications and simplify problem-solving for users.
- Ensure the computer system is user-friendly and easy to navigate.
- Optimize the use of computer hardware for efficiency.

What is an Operating System Used for?

- **As a platform for Application programs:** It provides a platform, on top of which, other programs, called application programs can run.
- **Managing Input-Output unit: It** also allows the computer to manage its own resources such as memory, monitor, keyboard, printer, etc. Management of these resources is required for effective and fair utilization.
- **Multitasking:** It manages memory and allows multiple programs to run in their own space and even communicate with each other through shared memory.
- **Manages memory and Files:** It manages the computer's main memory and second storage. Additionally, it allows and deallocates memory to all tasks and applications.
- **Provides Security:** It helps to maintain the system and applications safe through the authorization process. Thus, the OS provides security to the system.

Objectives of Operating Systems

Let us now see some of the objectives of the operating system, which are mentioned below.

- **Convenient to use:** One of the objectives is to make the computer system more convenient to use in an efficient manner.

- **User Friendly:** To make the computer system more interactive with a more convenient interface for the users.
- **Easy Access:** To provide easy access to users for using resources by acting as an intermediary between the hardware and its users**.**
- **Management of Resources:** For managing the resources of a computer in a better and faster way.
- **Controls and Monitoring:** By keeping track of who is using which resource, granting resource requests, and mediating conflicting requests from different programs and users.
- **Fair Sharing of Resources:** Providing efficient and fair sharing of resources between the users and programs.

Examples of Operating Systems

- **Windows** (GUI-based, PC)
- **GNU/Linux** (Personal, Workstations, ISP, File, and print server, Three-tier client/Server)
- **macOS** (Macintosh), used for Apple's personal computers and workstations (MacBook, iMac).
- **Android** (Google's Operating System for smartphones/tablets/smartwatches)
- **iOS** (Apple's OS for iPhone, iPad, and iPod Touch)

Functions of Operating System:

An Operating System acts as a communication interface between the user and computer hardware. Its purpose is to provide a platform on which a user can execute programs conveniently and efficiently. The main goal of an operating system is to make the computer environment more convenient to use and to utilize resources most efficiently.

Operating System handles the following responsibilities:

- Controls all the computer resources.
- Provides valuable services to user programs.
- Provides resources for user programs.
- Provides an interface (virtual machine) to the user.
- Supports multiple execution modes.
- Monitors the execution of user programs to prevent errors.

Fig: Functions of OS

1. Process Management

Process management in operating system is about managing processes. A Process is a running program. The life cycle of process is from the moment program start until it finishes. Operating system makes sure each process:

- gets its turn to use the CPU
- synchronized when needed
- has access to the resources it needs, like memory, files, and input/output devices.

It also handles issues like process coordination and communication, while preventing conflicts such as deadlocks. This way, the OS ensures smooth multitasking and efficient resource use.

2. Memory Management

Memory management is an essential task of the operating system that handles the storage and organization of data in both main (primary) memory and secondary storage. The OS ensures that memory is allocated and deallocated properly to keep programs running smoothly. It also manages the interaction between volatile main memory and non-volatile secondary storage.

Key Activities in Memory Management:

Main Memory Management

- **Memory Allocation:** Assigns memory to processes using techniques like paging and segmentation.
- **Memory Deallocation:** Frees memory when no longer needed.
- **Memory Protection:** Prevents processes from accessing each other's memory.
- **Virtual Memory:** Uses disk space as extra memory to run larger processes.
- **Fragmentation:** Manages wasted memory space (internal/external) through compaction.

Secondary Memory Management

- **Disk Space Allocation:** Organizes how files are stored on the disk (contiguous, linked, indexed).
- **File System Management:** Manages files and directories for efficient data access.
- **Free Space Management:** Tracks available space on the disk.
- **Disk Scheduling:** Organizes the order of disk read/write requests.
- **Backup and Recovery:** Ensures data is backed up and can be restored after failure.

3. File System Management

File management in the operating system ensures the organized storage, access and control of files. The OS abstracts the physical storage details to present a logical view of files, making it easier for users to work with data. It manages how files are stored on different types of storage devices (like hard drives or SSDs) and ensures smooth access through directories and permissions.

File System Management includes managing of:

File Attributes

- **File Name:** Identifies the file with a name and extension (e.g., .txt, .jpg).
- **File Type:** Defines the format of the file (e.g., text, image, executable).
- **Size:** The amount of storage the file occupies.
- **Permissions:** Determines who can read, write, or execute the file.

File Types

- **Text Files:** Contain human-readable content (e.g., .txt, .md).
- **Binary Files:** Store data in binary format (e.g., .jpg, .mp3).
- **Executable Files:** Contain program code (e.g., .exe, .out).

Operations on Files

- **Create:** Allows users to create new files.
- **Read:** Opens files to read their contents.
- **Write:** Modifies the contents of a file.
- **Delete:** Removes a file from the system.

Access Methods

- **Sequential Access:** Reads data in order, from start to finish.
- **Direct Access:** Jumps to a specific part of the file.
- **Indexed Access:** Uses an index for quick data retrieval.

4. Device Management (I/O System)

Device management of an operating system handles the communication between the system and its hardware devices, like printers, disks or network interfaces. The OS provides device drivers to control these devices, using techniques like Direct Memory Access (DMA) for efficient data transfer and strategies like buffering and spooling to ensure smooth operation

Major components in Device Management-

Device Drivers: The operating system uses **device drivers** to interact with hardware devices. There are two types of device drivers:

- **Kernel-space drivers** run in the OS kernel, offering direct access to hardware.
- **User-space drivers** run outside the kernel and are more isolated, providing safety but less performance.

Buffering & Caching:

- **Buffering** temporarily stores data in memory to manage differences in device speeds. Block devices (e.g., hard drives) use larger blocks of data for buffering, while character device**s** (e.g., keyboards, mice) use smaller, byte-by-byte buffering.
- **Caching** improves access speed by storing frequently accessed data in a faster storage medium (like RAM).

5. Protection and Security

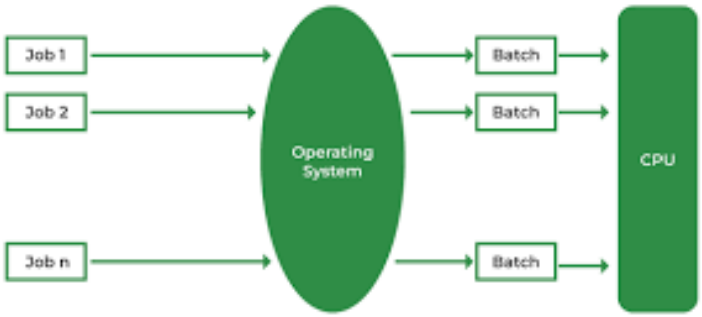
Protection and security mechanisms in an operating system are designed to safeguard system resources from unauthorized access or misuse. These mechanisms control which processes or users can access specific resources (such as memory, files, and CPU time) and ensure that only authorized users can perform specific actions. While protection ensures proper access control, security focuses on defending the system against external and internal attacks.

Types of Operating Systems:

Different types of OS serve different needs some handle one task at a time, while others manage multiple users or real-time processes.

1. Batch Operating System

A Batch Operating System is designed to handle large groups of similar jobs efficiently. It does not interact with the computer directly but instead processes jobs that are grouped by an operator. These jobs are queued and executed one after the other, without user interaction during the process.



Advantages of Batch Operating System

- **Efficient Job Management**: Multiple users can efficiently share the system, making it cost-effective.
- **Minimal Idle Time**: The system minimizes idle time by processing jobs in a continuous sequence without human intervention.
- **Handling Repetitive Tasks**: Ideal for managing large, repetitive tasks, such as payroll and billing, with minimal effort.
- **Improved Throughput**: Batch systems can handle high volumes of jobs at once, improving overall system throughput.

Disadvantages of Batch Operating System

- **Inefficient CPU Utilization**: When a job is waiting for input/output (I/O), the CPU remains idle, leading to poor utilization of resources.
- **Unpredictable Job Completion**: If one job fails, others may be delayed indefinitely, making job completion time unpredictable.
- **Increased Response Time**: The time between job submission and output can be high as all jobs are processed sequentially.

Examples:

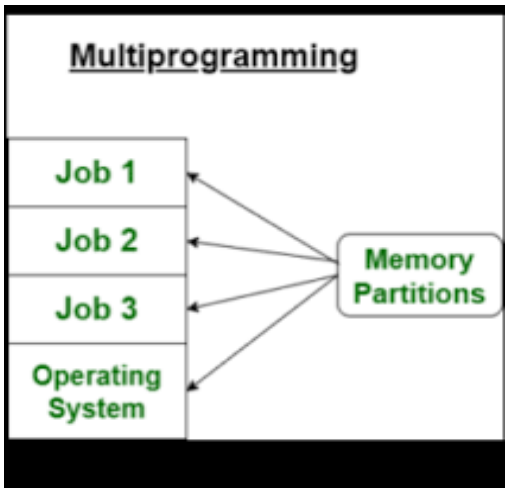
Payroll Systems
Bank Statements

2. Multi-Programming Operating System

In a Multi-Programming Operating System, multiple programs run in memory at the same time. The CPU switches between programs, utilizing its resources more effectively and improving overall system performance.

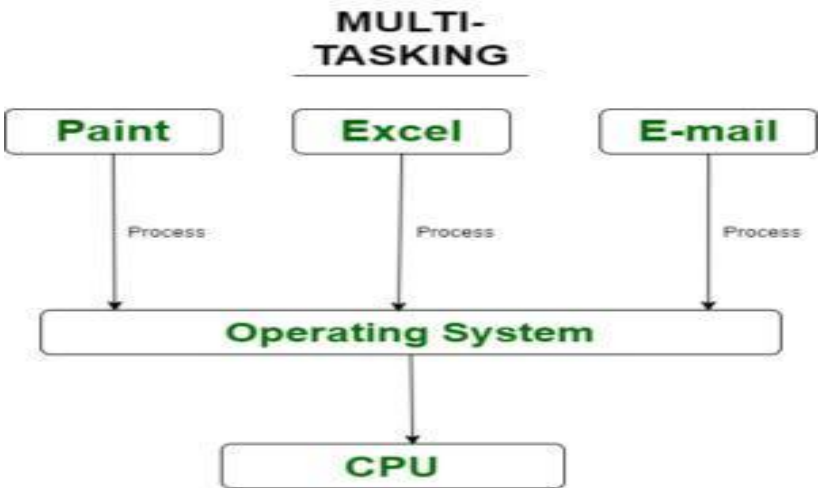
Advantages of Multi-Programming Operating System

- CPU is better utilized and the overall performance of the system improves.
- It helps in reducing the response time.



3. Multi-tasking/Time-sharing Operating systems

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



Advantages of Time-Sharing OS

- Each task gets an equal opportunity.
- Fewer chances of duplication of software.
- CPU idle time can be reduced.
- Resource Sharing: Time-sharing systems allow multiple users to share hardware resources such as the CPU, memory and peripherals, reducing the cost of hardware and increasing efficiency.

- Improved Productivity: Time-sharing allows users to work concurrently, thereby reducing the waiting time for their turn to use the computer. This increased productivity translates to more work getting done in less time.
- Improved User Experience: Time-sharing provides an interactive environment that allows users to communicate with the computer in real time, providing a better user experience than batch processing.

Disadvantages of Time-Sharing OS

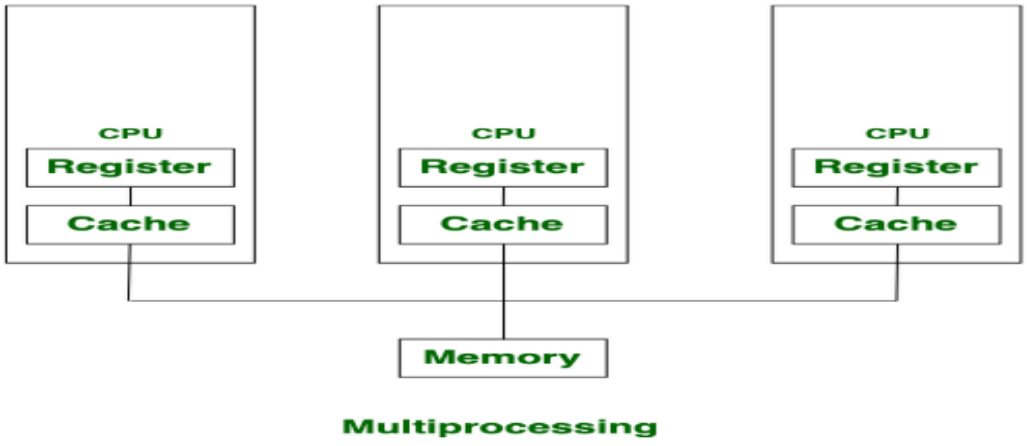
- Reliability problem.
- One must take care of the security and integrity of user programs and data.
- Data communication problem.
- High Overhead: Time-sharing systems have a higher overhead than other operating systems due to the need for scheduling, context switching and other overheads that come with supporting multiple users.
- Security Risks: With multiple users sharing resources, the risk of security breaches increases. Time-sharing systems require careful management of user access, authentication and authorization to ensure the security of data and software.

Examples:

UNIX OS
TSO (Time Sharing Option
Windows Terminal Services

4. Multi-Processing Operating System

A Multi-Processing Operating System is a type of Operating System in which more than one CPU is used for the execution of resources. It betters the throughput of the System.



Advantages of a Multi-Processing Operating System

- It increases the throughput of the system as processes can be parallelized.
- As it has several processors, so, if one processor fails, we can proceed with another processor.

5. Distributed Operating System

Distributed operating systems are a recent advancement in the world of computer technology and are being widely accepted all over the world and, that too, at a great pace. Various

autonomous interconnected computers communicate with each other using a shared communication network. Independent systems possess their own memory unit and CPU.

The major benefit of working with these types of operating systems is that it is always possible that one user can access the files or software which are not present on his system but on some other system connected within this network, i.e., remote access is enabled within the devices connected to that network.

Advantages of Distributed Operating System

- Failure of one will not affect the other network communication, as all systems are independent of 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 is not yet well-defined.
- 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.

6. Real-Time Operating System

These types of OSs serve 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 that are very strict like missile systems, air traffic control systems, robots, etc.

Types of Real-Time Operating Systems

- **Hard Real-Time Systems:** Hard Real-Time OSs are meant for applications where time constraints are very strict and even the shortest possible delay is not acceptable. These systems are built for saving lives like automatic parachutes or airbags which are required to be readily available in case of an accident. Virtual memory is rarely found in these systems.

- **Soft Real-Time Systems:** These OSs are for applications where time is less strict.

Advantages of RTOS

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

Disadvantages of RTOS

- **Limited Tasks:** Very few tasks run at the same time and their concentration is very less on a few applications to avoid errors.
- **Use heavy system resources:** Sometimes the system resources are not so good and they are expensive as well.
- **Complex Algorithms:** The algorithms are very complex and difficult for the designer to write.
- **Device driver and interrupt signals:** It needs specific device drivers and interrupt signals to respond earliest to interrupts.

.

Examples:

Scientific experiments
Medical imaging systems
Industrial control systems
Weapon systems
Robots
Air traffic control systems

7. Mobile Operating Systems

Mobile operating systems are designed specifically for mobile devices such as smartphones and tablets. Examples of such operating systems are Android and iOS. These operating systems manage the hardware and software resources of the device, providing a platform for running applications and ensuring a seamless user experience.

Advantages of Mobile Operating Systems

- **User-Friendly Interfaces**: Mobile operating systems are designed to be intuitive and easy to use, making them accessible to a wide range of users.
- **Extensive App Ecosystems**: The availability of a vast number of applications allows users to customize their devices to meet their specific needs.
- **Connectivity Options**: Mobile operating systems support multiple connectivity options, enabling users to stay connected wherever they go.
- **Regular Updates**: Mobile operating systems receive regular updates, including new features, security patches and performance improvements.

Disadvantages Mobile Operating Systems

- **Battery Life Constraints**: Despite advancements in power management, battery life remains a challenge for mobile devices, especially with heavy usage.
- **Security Risks**: Mobile devices are susceptible to various security threats, such as malware and phishing attacks, which can compromise user data.
- **Fragmentation**: In the case of Android, the wide range of devices and customizations can lead to fragmentation, making it difficult for developers to ensure compatibility across all devices.
- **Limited Hardware Resources**: Mobile devices have limited processing power, memory and storage compared to desktop computers, which can affect the performance of resource-intensive applications.

Elements of command based and GUI based operating system:
A Command Line Interface (CLI) and a Graphical User Interface (GUI) are two distinct methods for interacting with computer systems and software. CLI allows users to execute commands by typing them into a terminal or console providing precise control and efficiency but requiring knowledge of command syntax. In contrast, GUI offers a visual interface with elements like windows, icons, and buttons making it more intuitive and user-friendly by allowing users to interact through graphical representations. While CLI excels in environments where precision and speed are critical GUI enhances usability and accessibility through its visual and interactive design.

Difference Between CLI & GUI-

CLI	GUI
CLI is difficult to use.	Whereas it is easy to use.
It consumes low memory.	While consuming more memory.
In CLI we can obtain high precision.	While in it, low precision is obtained.
CLI is faster than GUI.	The speed of GUI is slower than CLI.
CLI operating system needs only a keyboard.	While GUI operating system needs both a mouse and keyboard.
CLI's appearance can not be modified or changed.	While its appearance can be modified or changed.
In CLI, input is entered only at a command prompt.	While in GUI, the input can be entered anywhere on the screen.
In CLI, the information is shown or presented to the	While in GUI, the information is

CLI	GUI
user in plain text and files.	presented to the user in plain text, videos, images, etc.
In CLI, there are no menus provided.	While in GUI, menus are provided.
There are no graphics in CLI.	While in GUI, graphics are used.
CLI do not use any pointing devices.	While it uses pointing devices for selecting and choosing items.
In CLI, spelling mistakes and typing errors are not avoided.	Whereas in GUI, spelling mistakes and typing errors are avoided.
Some command-line environments provide multitasking but it is complicated to see several things on one screen.	GUI enables a user to easily observe and operate various things at once.
CLI enables a user to simply script a series of instructions to carry out a task or execute a program.	GUI does not provide the facility to script a sequence of commands.

Computer Network:

A Computer Network is a system where two or more devices are linked together to share data, resources and information**.** These networks can range from simple setups, like connecting two devices in your home, to massive global systems, like the Internet. Below are some uses of computer networks

- **Sharing devices such as printers and scanners**: Multiple systems can access the same hardware, reducing the need for duplicate devices and lowering costs.
- **Sharing Data**: Teams can work on shared documents, applications or systems, which boosts efficiency.
- **Communicating using web, email, video and instant messaging**: Networks enable both real-time and delayed communication. Users can access information, send messages, participate in video calls and chat.

- **Data management**: Networks allow organizations to store data in a central or distributed location, making it easier to manage, secure and back up critical information.
- **Remote access** : Users can log into computers, servers or cloud platforms from different locations, supporting remote work and 24/7 access.

Types of Computer Networks:

Computer networks are classified based on several factors, such as geographical area, ownership, architecture, topology, and transmission technology. They are as follows-

Local Area Network (LAN)

LAN is the most frequently used network. It is a computer network that connects computers through a common communication path, contained within a limited area, that is, locally. A LAN encompasses two or more computers connected over a server. The two important technologies involved in this network are Ethernet and Wi-fi. It ranges up to 2km & transmission speed is very high with easy maintenance and low cost.

Examples of LAN are Wi-Fi in a home or school, wired LAN in a company's office.

Metropolitan Area Network (MAN)

A MAN is larger than a LAN but smaller than a WAN. This is the type of computer network that connects computers over a geographical distance through a shared communication path over a city, town, or metropolitan area. This network mainly uses FDDI, CDDI, and ATM as the technology with a range from 5km to 50km. Its transmission speed is average. It is difficult to maintain and it comes with a high cost.

Examples of MAN are networking in towns, cities, a single large city, a large area within multiple buildings, etc.

Wide Area Network (WAN)

WAN is a type of computer network that connects computers over a large geographical distance through a shared communication path. It is not restrained to a single location but extends over many locations. WAN can also be defined as a group of local area networks that communicate with each other with a range above 50km. Here we use Leased-Line & Dial-up technology. Its transmission speed is very low and it comes with very high maintenance and very high cost.

Examples of WAN are the Internet (largest WAN), Banking networks linking global branches

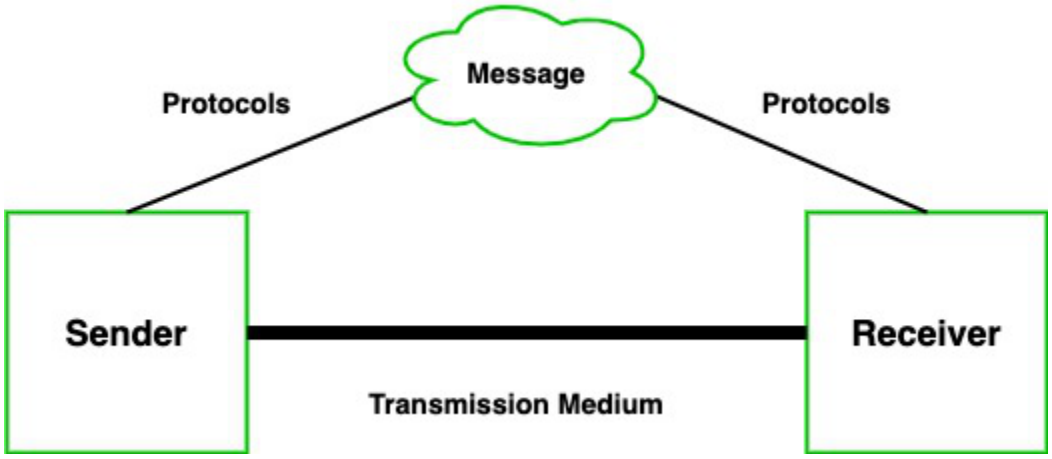
Data Communication:

Transferring data over a transmission medium between two or more devices, systems, or places is known as data communication. Nowadays, computing and telecommunications depend heavily on this data transmission, which makes a variety of applications conceivable, including email, video chatting, the Internet, and many more things.

Components of Data Communication

A communication system is made up of the following components:

1. **Message:** A message is a piece of information that is to be transmitted from one person to another. It could be a text file, an audio file, a video file, etc.
2. **Sender:** It is simply a device that sends data messages. It can be a computer, mobile, telephone, laptop, video camera, or workstation, etc.
3. **Receiver:** It is a device that receives messages. It can be a computer, telephone mobile, workstation, etc.
4. **Transmission Medium / Communication Channels:** Communication channels are the medium that connect two or more workstations. Workstations can be connected by either wired media or wireless media.
5. **Set of rules (Protocol):** When someone sends the data (The sender), it should be understandable to the receiver also otherwise it is meaningless. For example, Sonali sends a message to Chetan. If Sonali writes in Hindi and Chetan cannot understand Hindi, it is a meaningless conversation.



Type of data communication

As we know that data communication is communication in which we can send or receive data from one device to another. The data communication is divided into three types:

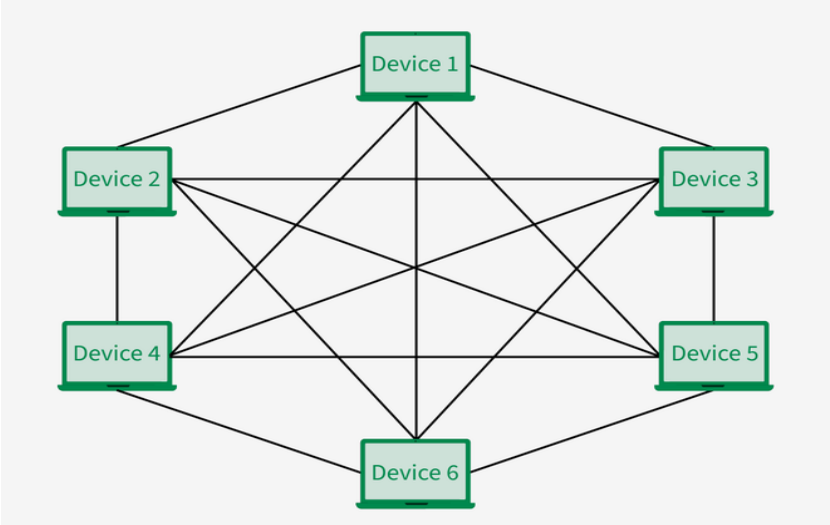
1. **Simplex Communication:** It is one-way communication or we can say that unidirectional communication in which one device only receives and another device only sends data and devices uses their entire capacity in transmission. For example, IoT, entering data using a keyboard, listening music using a speaker, etc.
2. **Half Duplex communication:** It is a two-way communication, or we can say that it is a bidirectional communication in which both the devices can send and receive data but not at the same time. When one device is sending data then another device is only receiving and vice-versa. For example, walkie-talkie.
3. **Full-duplex communication:** It is a two-way communication or we can say that it is a bidirectional communication in which both the devices can send and receive data at the same time. For example, mobile phones, landlines, etc.

Network Topology:

Network topology refers to the arrangement of different elements like nodes, links, or devices in a computer network. Common types of network topology include bus, star, ring, and mesh topologies.

Mesh Topology

In a mesh topology, every device is connected to another device via a particular channel. Every device is connected to another via dedicated channels. These channels are known as links.



- Suppose, the N number of devices are connected with each other in a mesh topology, the total number of ports that are required by each device is N-1. In Figure , there are 6 devices connected to each other, hence the total number of ports required by each device is 5. The total number of ports required = N * (N-1).
- Suppose, N number of devices are connected with each other in a mesh topology, then the total number of dedicated links required to connect them is N C 2 i.e. N(N-1)/2. In Figure, there are 6 devices connected to each other, hence the total number of links required is 6*5/2 = 15.

Advantages of Mesh Topology

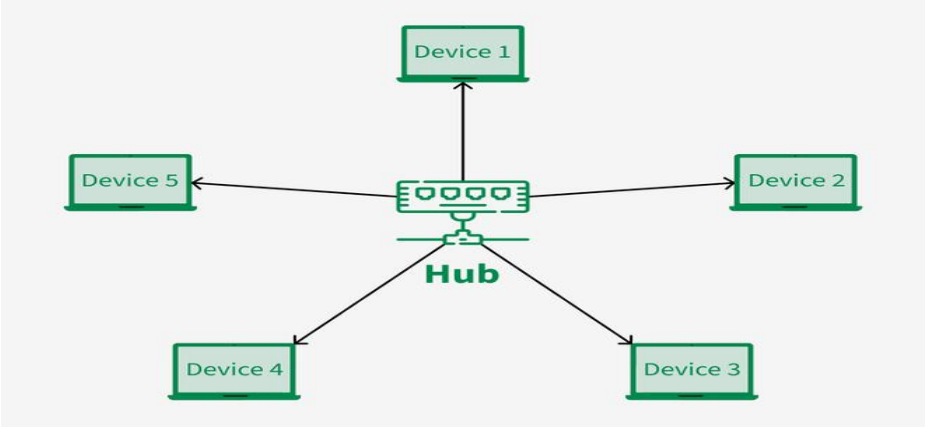
- Communication is very fast between the nodes.
- Mesh Topology is robust.
- The fault is diagnosed easily. Data is reliable because data is transferred among the devices through dedicated channels or links.
- Provides security and privacy.

Disadvantages of Mesh Topology

- Installation and configuration are difficult.
- The cost of cables is high as bulk wiring is required, hence suitable for less number of devices.
- The cost of maintenance is high.

Star Topology

In Star Topology, all the devices are connected to a single hub through a cable. This hub is the central node and all other nodes are connected to the central node. The hub can be passive in nature i.e., not an intelligent hub such as broadcasting devices, at the same time the hub can be intelligent known as an active hub. Active hubs have repeaters in them.



Advantages of Star Topology

- If N devices are connected to each other in a star topology, then the number of cables required to connect them is N. So, it is easy to set up.
- Each device requires only 1 port i.e. to connect to the hub, therefore the total number of ports required is N.
- It is Robust. If one link fails only that link will affect and not other than that.
- Easy to fault identification and fault isolation.
- Star topology is cost-effective as it uses inexpensive coaxial cable.

Disadvantages of Star Topology

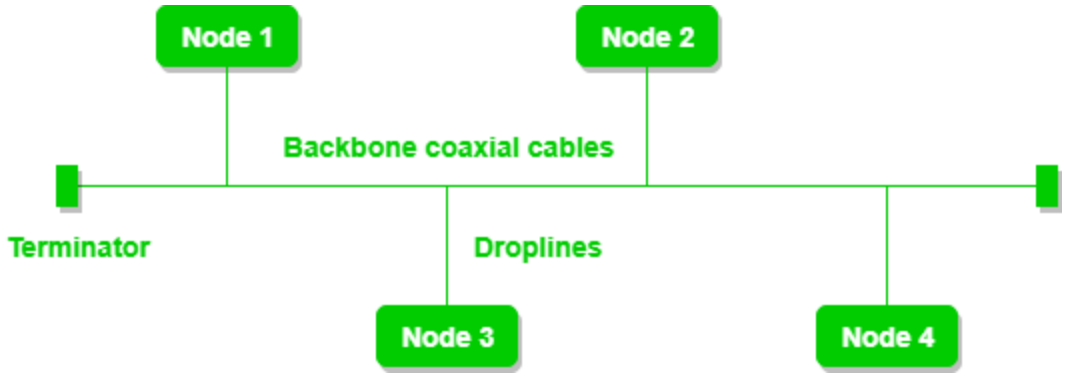
- If the concentrator (hub) on which the whole topology relies fails, the whole system will crash down.
- The cost of installation is high.
- Performance is based on the single concentrator i.e. hub.

A common example of star topology is a **local area network (LAN)** in an office where all computers are connected to a central hub. This topology is also used in wireless networks where all devices are connected to a wireless access point.

Bus Topology

Bus Topology is a network type in which every computer and network device is connected to a single cable. It is bi-directional. It is a multi-point connection and a non-robust topology because if the backbone fails the topology crashes.

A common example of bus topology is the Ethernet LAN, where all devices are connected to a single coaxial cable or twisted pair cable. This topology is also used in cable television networks



Advantages of Bus Topology

- If N devices are connected to each other in a bus topology, then the number of cables required to connect them is 1, known as backbone cable, and N drop lines are required.
- Coaxial or twisted pair cables are mainly used in bus-based networks that support up to 10 Mbps.
- The cost of the cable is less compared to other topologies, but it is used to build small networks.
- Bus topology is familiar technology as installation and troubleshooting techniques are well known.
- CSMA is the most common method for this type of topology.

Disadvantages of Bus Topology

- A bus topology is quite simpler, but still, it requires a lot of cabling.
- If the common cable fails, then the whole system will crash down.
- If the network traffic is heavy, it increases collisions in the network. To avoid this, various protocols are used in the MAC layer known as Pure Aloha, Slotted Aloha, CSMA/CD, etc.
- Adding new devices to the network would slow down networks.
- Security is very low.

Ring Topology

In a Ring Topology, it forms a ring connecting devices with exactly two neighboring devices. A number of repeaters are used for Ring topology with a large number of nodes, because if someone wants to send some data to the last node in the ring topology with 100 nodes, then the data will have to pass through 99 nodes to reach the 100th node. Hence to prevent data loss repeaters are used in the network.

The data flows in one direction, i.e. it is unidirectional, but it can be made bidirectional by having 2 connections between each Network Node, it is called Dual Ring Topology. In-Ring Topology, the Token Ring Passing protocol is used by the workstations to transmit the data.

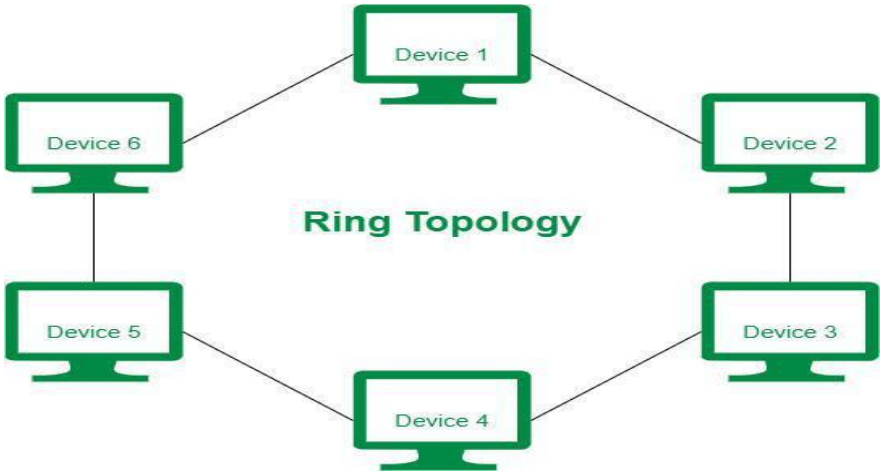
The most common access method of ring topology is token passing.

- **Token passing:** It is a network access method in which a token is passed from one node to another node.
- **Token:** It is a frame that circulates around the network.

Operations of Ring Topology

- One station is known as a **monitor** station which takes all the responsibility for performing the operations.

- To transmit the data, the station has to hold the token. After the transmission is done, the token is to be released for other stations to use.
- When no station is transmitting the data, then the token will circulate in the ring.
- There are two types of token release techniques: **Early token release** releases the token just after transmitting the data and **Delayed token release** releases the token after the acknowledgment is received from the receiver.



Advantages of Ring Topology

- The data transmission is high-speed.
- The possibility of collision is minimum in this type of topology.
- Cheap to install and expand.
- It is less costly than a star topology.

Disadvantages of Ring Topology

- The failure of a single node in the network can cause the entire network to fail.
- Troubleshooting is difficult in this topology.
- The addition of stations in between or the removal of stations can disturb the whole topology.
- Less secure.

Each topology, whether it's bus, star, ring, or mesh, offers unique benefits and potential drawbacks. By understanding these different arrangements, network designers can choose the most appropriate topology to meet the specific needs of their systems, ensuring optimal performance and connectivity.