

Draft Study Material

JOB ROLE

JUNIOR CLOUD COMPUTING ASSOCIATE

Qualification Pack

QG-04-IT-00354-2023-V1-NIELIT

SECTOR: IT-ITeS

Grades XI



विद्यया ऽ मृतमश्नुते



एन सी ई आर टी
NCERT

PSS CENTRAL INSTITUTE OF VOCATIONAL EDUCATION

(a constituent unit of National Council of Educational Research and Training (NCERT)
under Ministry of Education, Government of India)

www.psscive.ac.in

Junior Cloud Computing Associate Grade – XI

**Qualifications Pack
QG-04-IT-00354-2023-V1-NIELIT**



PSS Central Institute of Vocational Education

[A constituent unit of NCERT, Under Ministry of Education,
Government of India]

Shyamla Hills, Bhopal - 462 002, Madhya Pradesh, India
www.psscive.ac.in

DISCLAIMER

This material is only a reference study material and has been prepared by experts. Care has been taken to acknowledge the information with suitable references.

January, 2025

© PSSCIVE, 2025

All rights reserved

Published by:

Joint Director
PSS Central Institute of Vocational
Education, NCERT,
Shyamla Hills, Bhopal

**CHIEF PATRON**

Prof. Dinesh Prasad Saklani

Director
National Council of Educational
Research and Training (NCERT),
New Delhi

PATRON

Dr. Deepak Paliwal

Joint Director
PSS Central Institute of Vocational
Education, Bhopal

PROGRAMME COORDINATOR

Dr. Munesh Chandra

Professor (CSE), Head, ICT Centre
Department of
Engineering and Technology,
PSS Central Institute
of Vocational Education, Bhopal

PREFACE

The National Education Policy, 2020 emphasizes upon removing hard distinction between arts, science and commerce; and between curricular, co-curricular and extracurricular activities; and between vocational and general education. NEP focuses on flexible curricular structure and multidisciplinary learning. The secondary stage is for students aged between 14 and 18 and is divided into two phases: Phase 1 — Grades 9 and 10 and Phase 2 – Grades 11 and 12. The secondary stage for students aged 14-18 is divided into two phases, with the guidelines presented for grades 11 and 12. The National Curriculum Framework for School Education (NCFSE) 2023 advocates for choice-based courses, aiming to provide flexibility, remove separations between disciplines, and align with industry needs. Vocational education in the secondary stage will be an integral part of the educational system designed to provide students with practical skills and knowledge that directly prepare them for specific careers or trades. The focus should be on the holistic development of each child, addressing not only vocational skills but also social, emotional, and life skills. In schools, vocational courses are expected to align with the National Skill Qualifications Framework (NSQF) falling within NSQF levels 3 and 4. The NSQF is a quality assurance framework that organizes qualifications in a series of eight levels, in increasing order of complexity and competency. These levels are defined in terms of learning outcomes which are an explicit description of what a learner should know, understand and be able to do as a result of learning, regardless of whether these competencies were acquired through formal, non-formal or informal learning.

The NEP underscores the importance of vocational education, preparing students with practical skills aligned with industry needs. In the context of web development, this translates to equipping learners with not just theoretical knowledge but also hands-on experience. We should strive to provide comprehensive training that empowers individuals to tackle real-world challenges in the digital landscape.

Just as the NEP emphasizes the holistic development of students, our approach to web development should extend beyond technical skills. Let's prioritize the development of essential soft skills such as problem-solving, collaboration, and adaptability, ensuring that learners are well-rounded professionals capable of thriving in diverse environments.

I thank all other members for completing this task on time and in such an admirable way. I am also thankful to all the institutions and organisations which have generously extended their help and assistance in making this possible. As an organisation committed to reforming school education in Bharat and continuously improving the quality of all learning and teaching material that it develops, NCERT looks forward to critical comments and suggestions from all its stakeholders to further improve upon this textbook.

Professor Dinesh Prasad Saklani

Director

National Council of Educational Research and Training

New Delhi

April 2024

Foreword

Vocational Education and Training (VET) plays a significant role in willing youth for relevant occupation and meeting the skill demand of the changing labour market. This is even more relevant, at India is witnessing accelerated youth population and the need for preparing skilled workforce for the growing economy. The strong partnership with the industry partners characterises India's National Skill Qualification Framework (NSQF), The Vocationalisation of Education in Schools under *Samagra Shiksha* by the Ministry of Education, Government of India is spearheading and catalysing the role of vocational education and training in equipping young people with skills.

The recent reforms through National Educational Policy (NEP) 2020 have focused on making VET system more coherent and flexible to both the needs of the labour market and social challenges. Improving the learning pathways and bridging the gap between vocational and general education and avoiding dead ends is another goal. The ultimate goal is to ensure flexibility and responsiveness to the needs through education and training and to provide a strong framework for lifelong learning.

Reflecting on vocational education and training priorities, and recent developments in the system, priority has to be placed on developing vocational teachers or trainers to act as the link between education and training and employment. Preparing a cadre of professionally trained vocational teachers is vital for imparting quality vocational education and developing skilled workforce in different sectors. In this perspective, the PSS Central Institute of Vocational Education (PSSCIVE), Bhopal has introduced a 'Diploma in Vocational Education and Training' through distance mode, with the aim to develop a pool of trained vocational teachers or resource persons in spearheading the effective implementation of the scheme on vocationalisation of education in schools across India. The Diploma in VET is a one year programme, which will be taught in four blocks of tri-semester. It aims at providing the learners with the latest knowledge, skills and competencies in the field of vocational education and training. Among others, the programme will also enable the learners to appreciate the ethical dimension of teacher professionalism in Vocational Education. The goal is to equip the learners with a strong theoretical and practical understanding of VET while integrating ICT in their teaching.

I acknowledge the contributions of the material development team, reviewers and the support team for their contributions in the development of this self-learning material. We would welcome suggestions, which would help us to improve further the quality of this programme.

Wish you all the very best in this endeavour.

Dr. Deepak Paliwal

Joint Director

PSSCIVE, Bhopal

About the Textbook

“Junior Cloud Computing Associate for class 11” is a concise yet comprehensive textbook designed to equip students with essential cloud computing. Through clear explanations and practical exercises, students learn the foundational concepts of Windows and Linux OS, switches and routers, IP addressing and basics of cloud computing. By engaging in hands-on projects, students gain practical experience in managing operating systems, configuring networks, implementing virtualization, and understanding cloud environments, building confidence and problem-solving skills in real-world IT scenarios.

This book provides a structured pathway for students to acquire valuable skills that are increasingly in demand. Whether students aim to pursue further education or enter the workforce, this textbook serves as a stepping stone to success in the dynamic field of cloud computing. The book is divided into three units, the book meticulously covers every aspect of Operating Systems (Windows & Linux), Network Management, Virtualization, and the Basics of Cloud Computing.

Throughout the book, emphasis is placed on developing critical thinking and problem-solving skills, enabling students to manage and troubleshoot system-level and network-level challenges effectively. Practical exercises and case studies accompany each chapter, providing students with hands-on experience and reinforcing theoretical concepts.

Dr. Munesh Chandra

Professor

Department of Engineering and Technology
PSSCIVE, NCERT, Bhopal

Textbook Development Team

1. Dr. Kanika Taneja, ABES Institute of Technology, Ghaziabad
2. Dr. Nandita Goyal, AKG Engineering College, Ghaziabad
3. Mr. Mayank Bhushan, Directorate of Education, Delhi, Lecturer
4. Ms. Soumya Trivedi, AKG Engineering College, Ghaziabad

MEMBER-COORDINATOR

Dr. Munesh Chandra,

Professor & Head ICTC & Computer Centre

Department of Engineering and Technology

PSSCIVE, NCERT, Bhopal

Acknowledgement

On behalf of team at the PSS Central Institute of Vocational Education (PSSCIVE), Bhopal we are grateful to the officials of the Department of School Education and Literacy, Ministry of Education, Government of India for the guidance and support at all times and levels.

We are obliged to the Director, NCERT for his care and leadership. We are indebted to the PAC NCERT for financial support.

We acknowledge the contributions of our colleagues at PSSCIVE and other experts for their academic support, untiring efforts and contributions in development of this material. The names of all the experts are acknowledged in the list of contributors.

The contributions made by the Administration and the supporting staffs of PSSCIVE are duly acknowledged.

TEAM PSSCIVE

Table of Content

Particular		Page No.
Unit – 1 Introduction to Operating Systems (Windows & Linux)		
Chapter -1	Basic Concept of Operating System	11
Chapter -2	Operating System Structure	50
Chapter -3	Explore Windows and Linux OS Environment	72
Unit – 2 Introduction to Network Management		
Chapter -1	Introduction to Computer Network	90
Chapter -2	Protocol Models and Ip addressing	106
Chapter -3	Network Management and Cloud-based Networking	126
Chapter -4	Switching, Routing, and Network Components	156
Unit – 3 Implementing Virtualization and Basics of Cloud Computing		
Chapter -1	Understanding the Cloud Computing Fundamentals	172
Chapter -2	Cloud Infrastructure as a Service (IaaS)	186
Chapter -3	Platform as a Service (PaaS)	199
Chapter -4	Software as a Service (SaaS) and Cloud Applications	210
Chapter -5	Cloud Security and Governance	221

Unit-1.

Introduction of Operating Systems (Windows & Linux)

Chapter-1**Basic Concept of Operating System**

Kiara loved playing on Computer. Once while she was playing on computer, she thought that this a machine then how it understands our instructions. She explored and learnt about operating system. She realized that Windows she sees on her terminal is an operating system.

**1.1 Operating System**

An operating system (OS) serves as a mediator between a computer's user and its hardware. In simple terms, it acts as a bridge that connects user commands to the hardware components. The primary function of an operating system is to enable the efficient and seamless execution of user-defined programs.

The operating system is the software responsible for managing both hardware and software resources on a computer. To ensure smooth operation and to prevent user programs from interfering with system functions, the hardware must incorporate appropriate control mechanisms. Typically, only one operating system runs at a time, serving as the foundation for all other applications and programs. It manages the allocation of resources such as memory, the CPU, and input/output devices to various processes. Fair and secure resource allocation is essential for maintaining optimal system performance.

1.2 Functions of Operating system

Resource Management: The operating system (OS) acts as a resource manager by allocating hardware resources to users. While numerous processes run simultaneously on the system, each requires its own set of resources, such as memory and input/output devices. Efficient resource management reduces the overall system load.

Process Management: Process management includes a variety of tasks such as scheduling and synchronization. Central Processing Unit (CPU) scheduling algorithms are used to determine the order in which processes are executed. Synchronization is essential because processes often need to communicate with one another. For example, two processes might attempt to update the same memory location in the wrong sequence, leading to potential errors—this is just one of many issues that can arise during inter-process communication.

I/O Management:

- The primary function of an operating system is to shield users from the complexities of interacting directly with individual hardware components.
- Every operating system includes device drivers, which are software routines that manage input/output devices through their respective controllers.

Managing Files:

- Every computer has its own unique set of files, folders, and directory structure.

- The operating system is responsible for creating and deleting these files and directories. In addition to managing file and directory operations, the operating system also handles file backups to external media and prevents unauthorized access by ensuring that certain users cannot open specific files.

Storage Management: The file system is the method by which secondary storage—such as a hard drive is managed. Operating systems employ a variety of file systems to control storage, including NIFS, CIFS, CFS, NFS, and others. Data is stored across multiple tracks on the hard drive, all of which are managed by the storage manager.

Memory management refers to the handling of primary memory, including the allocation and deallocation of memory to processes. The operating system must monitor memory usage, identify which processes are using memory, and determine how much memory each process requires.

Management of Security and Privacy: The operating system also ensures privacy by using passwords to prevent unauthorized applications from accessing programs or data. For example, Windows protects user data from unauthorized access by using Kerberos authentication.

A User Interface:

Person using the system: Interacts with the system.

Applications and system programs: User-facing software.

System software: Controls and coordinates the use of IT hardware and software.

Physical components used for computing: Hardware, operating system(s), system programs, and application programs are the four main components of every general-purpose computer. The hardware includes storage devices, peripherals, memory, the central processing unit (CPU), and address lookup units. Compilers, loaders, editors, operating systems, and similar tools are all part of system programs.

A conceptual view of a computer system is shown in Figure 1.1.

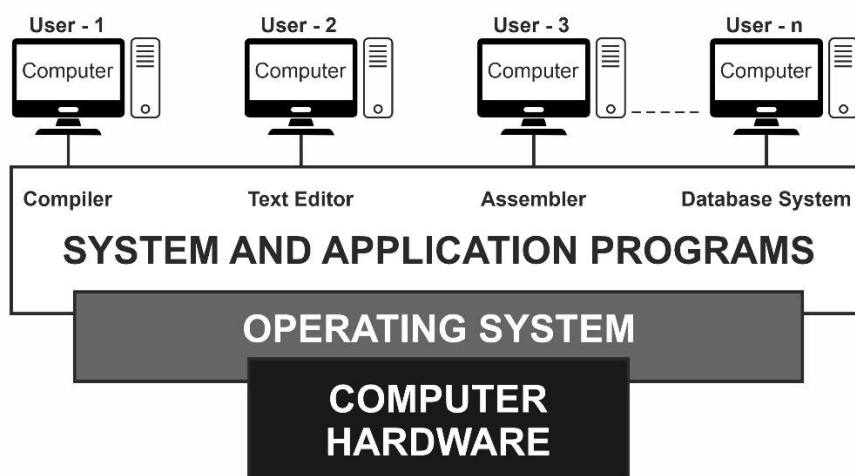


Fig 1.1. Conceptual View of Computer System

An operating system is required for other applications to run on any computer. It is responsible for coordinating the use of hardware among various user-specific application programs and system programs. Its primary role is to lay the groundwork for other programs to perform their tasks effectively.

Operating systems are collections of programs that enable computers to function efficiently. Basic tasks handled by the operating system include recognizing keystrokes, organizing disk files and directories, displaying output on the screen, and controlling peripheral devices. The main functions of an operating system are shown in Figure 1.2.

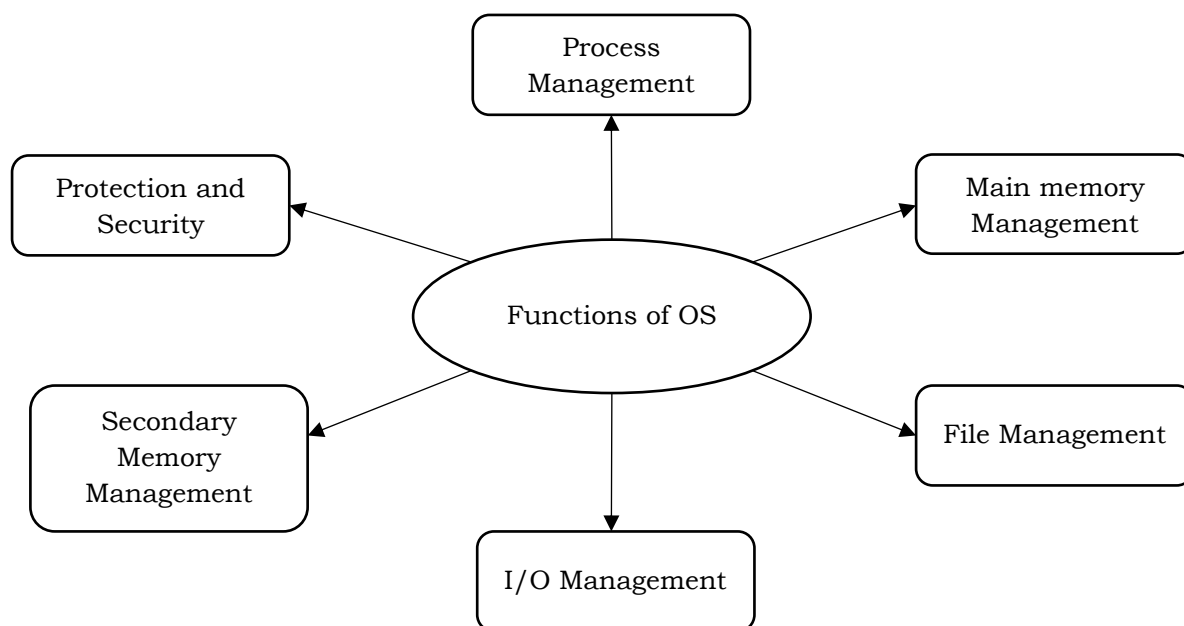


Fig 1.2. Functions of Operating System

Assignment 1.1

1. List the role of operating system.
2. List down the main functions of operating system.

1.3 Goals of Operating Systems

Goals of Operating Systems provide a high-level, generalized view of the final destination, while functions offer more specific, detailed guidance for achieving it. Let us consider our own country, India, to better understand this concept. All of India's resources—such as water, forests, minerals, food, and more—are available to us as citizens. We elect the Government of India to act as a mediator between the people and the nation's resources, ensuring that everyone benefits fairly and efficiently. This is similar to how an operating system functions as a mediator between computer users and hardware, as illustrated in Figure 1.3.

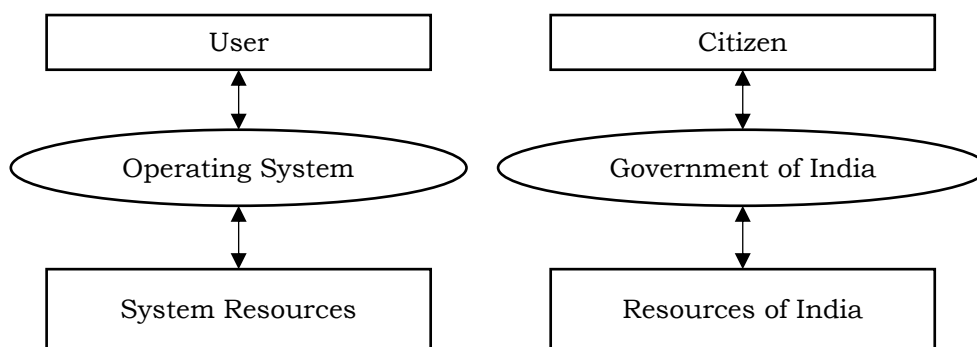


Fig 1.3. Comparison of Operating system role as resource manager with Government of India.

One of the many goals of our government is to guarantee the basic rights of all citizens, enabling national progress. As a result, not only does the country develop, but every citizen also gains access to quality healthcare and basic education. Ministries and departments within the government—such as those responsible for health, education, defense, and others—work together to achieve these objectives.

Similarly, there are primarily two types of objectives for an operating system: primary and secondary objectives.

- **Primary Purpose:** An operating system's fundamental role is to provide a user-friendly and convenient environment. While an OS is not strictly necessary, having one significantly simplifies the user experience. Without an OS, users would have to handle tedious tasks like process scheduling and translating user code into machine code—distractions that take focus away from core logic. The operating system assists the user

by managing these underlying processes. Once the user specifies what needs to be done, the operating system takes care of the rest.

- **Secondary Purpose:** In addition to being user-friendly, operating systems also aim for efficiency. It is the OS's responsibility to ensure that all resources are used effectively and that no resource remains idle when there is an active request for it.

The operating system handles numerous tasks to fulfill both the primary and secondary objectives described above. The goals of operating system is shown in Figure 1.4.

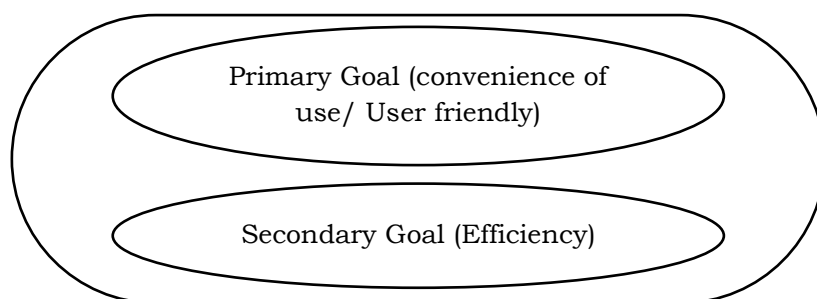


Fig 1.4. Goals of Operating System

1.4 Types of Operating Systems

Batch Operating System

There is no direct interaction between the user and the operating system. An operator collects tasks that are similar in nature and requirements, and then processes them in batches. It is the operator's responsibility to sort tasks with similar requirements. The purpose of a Batch Operating System is to efficiently organize and execute multiple jobs at once through group processing. The structure of batch operating system is shown in Figure 1.5.

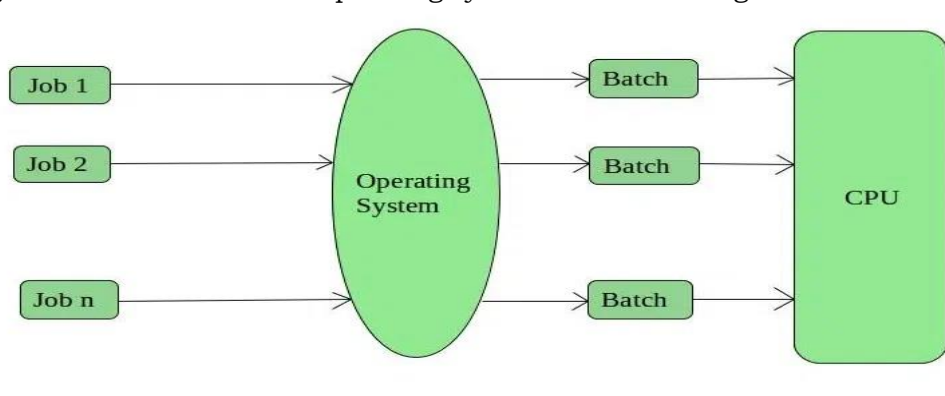


Fig 1.5. Batch Operating System

Advantages of batch operating system

- The batch system has minimal idle time and can be shared by multiple users.
- It is easy to handle large tasks repeatedly using batch systems.
- However, it is not efficient in utilizing the CPU. While the current process is performing input/output operations, the CPU remains idle, even though other processes could use it.

Disadvantages of batch operating system

- If one job fails, the remaining jobs are delayed for an unpredictable amount of time.
- The average response time increases, as each process is executed one after another.
- Examples include payroll systems, bank statement generation, etc.

Multi-Programming Operating System

Multiprogramming operating systems can be explained simply by noting that they allow multiple programs to reside in the main memory simultaneously. Any of these programs can be executed at a given time, which allows for more efficient use of available resources. The structure of multi-programming operating system is shown in Figure 1.6.

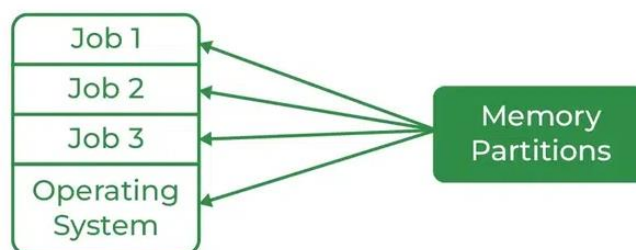


Fig 1.6. Multi Programming

Advantages of, Multi-Programming Operating System

- It helps minimize response time.
- It improves the overall performance of the system by utilizing the central processing unit (CPU) more effectively.

Multi-tasking/Time-sharing Operating systems

Each process in a multiprogramming system runs in a round-robin fashion. To ensure smooth operation, each task is assigned a specific amount of time to execute. Since all users share a single system, they also share CPU time. These types of systems are also known as multitasking systems. Tasks can be submitted by either a single user or multiple users. The term quantum refers to the amount of time allocated to each job. Once this time interval ends, the operating system moves on to the next task. The structure of multitasking operating system is shown in Figure 1.7.

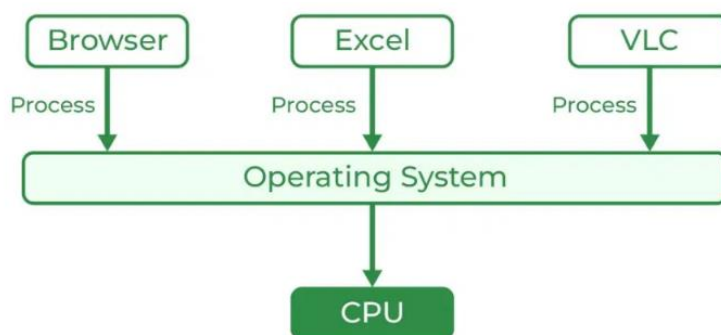


Fig 1.7. Multitasking

Advantages of Multitasking Operating System

- Lower likelihood of software duplication.
- Reduced CPU idle time.
- Equal opportunity for each task.
- Resource Sharing: Time-sharing systems optimize hardware usage by allowing multiple users to share resources such as the CPU, memory, and peripherals. This not only reduces hardware costs but also improves efficiency.
- Greater Efficiency: Time-sharing enables users to work simultaneously, reducing wait times for computer access. This boost in productivity allows more to be accomplished in less time.
- Enhanced User Experience: Time-sharing offers a more interactive and user-friendly environment compared to batch processing, allowing real-time communication between users and the computer.

Disadvantages of Multitasking Operating System

- Reliability issues.
- User programs and data must be protected.
- Data Exchange Challenges: Scheduling, context switching, and other overheads associated with supporting multiple users increase the system's complexity. Time-sharing systems have higher overhead compared to other operating systems and require advanced software to manage multiple users simultaneously. This level of complexity raises the chances of errors and system issues.

- **Security Concerns:** When multiple users share resources, the risk of security breaches increases. Ensuring data and software security in time-sharing systems requires strict control of user access, authentication, and authorization.

Examples of Time-Sharing Operating Systems

- **VM/CMS (Virtual Machine/Conversational Monitor System):** Developed by IBM in 1972, VM/CMS is a time-sharing operating system that allows multiple users to run their own operating systems and applications within a virtual machine environment. It is still in use today.
- **TSO (Time Sharing Option):** Developed by IBM in the 1960s for the System/360 mainframe, TSO was designed to facilitate time-sharing among users. It allowed multiple users to simultaneously launch and run their own programs on the same machine, which was a significant advancement at the time.
- **Windows Terminal Services:** A time-sharing operating system that enables multiple users to remotely access a Windows server. Users can run their applications simultaneously and access shared resources such as printers and network storage in real time.

Multi-Processing Operating System

A Multi-Processing Operating System is an operating system that uses more than one central processing unit (CPU) to execute tasks. This results in improved system throughput. The structure of multi-processing operating system is shown in Figure 1.8.

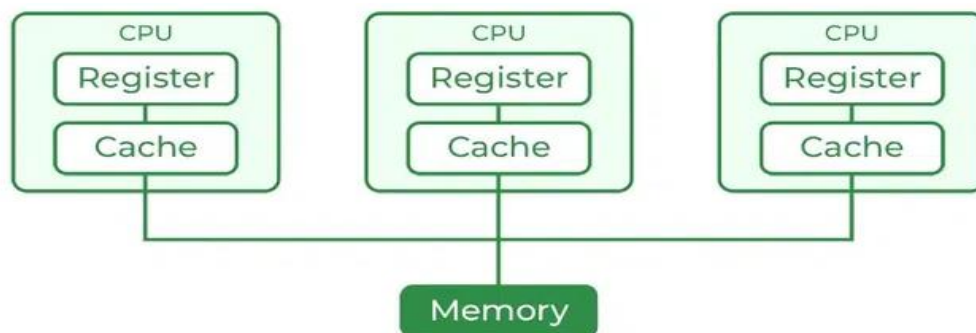


Fig 1.8 Multiprocessing Operating System

Advantages of a Multi-User Operating System

- Operations can be parallelized, resulting in improved system throughput.
- Additionally, having multiple processors allows the system to continue functioning using another processor in the event of a failure.

Multi-User Operating Systems

These technologies enable multiple users to engage in activities concurrently. Depending on the configuration, the system may be a multiprocessor or a single processor with interleaving. The structure of multi-user operating system is shown in Figure 1.9.

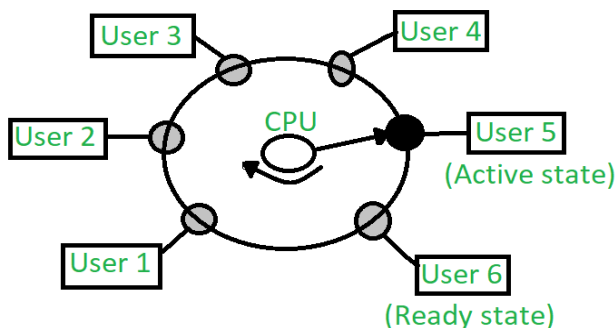


Fig 1.9 Multiuser OS

Distributed Operating System

Modern computer technology has recently seen the emergence of these operating systems, which are rapidly gaining acceptance around the globe. A shared communication network allows multiple autonomously connected computers to communicate with one another. Each

system has its own central processing unit and memory. The processors in these systems vary in size and purpose. The most appealing aspect of these operating systems is their ability to provide users with remote access—allowing them to access files and programs that are not installed locally but are available on another device within the same network. The Architecture of distributed operating system is shown in Figure 1.10.

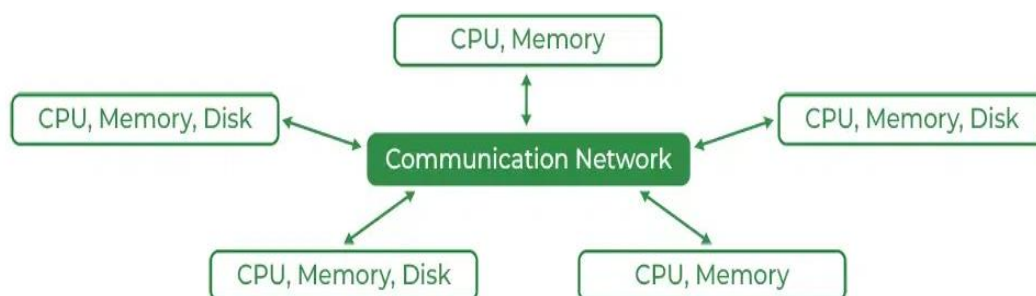


Fig 1.10 Architecture of Distributed OS

Advantages of Distributed Operating System

- All systems operate independently, so if one fails, it does not affect the network connectivity of the others.
- Data interchange speeds are improved through electronic mail.
- Resource sharing makes computing both efficient and sustainable, reducing the load on the host computer.
- These systems are highly scalable, as more systems can easily be added to the network.
- Data processing delays are minimized.

Disadvantages of Distributed Operating System

- If the main network fails, all communication comes to a halt.
- The programming languages and tools for developing distributed systems still require significant improvement.
- These systems are relatively expensive and not easily accessible. Additionally, the underlying software is highly complex and not yet fully understood.
- Example: LOCUS is one example of a distributed operating system.

Issues with Distributed Operating Systems

- **Network Latency:** Data transfers between nodes can be slowed down by networking delays. These lags can make it difficult to determine the correct sequence of events and may cause data inconsistencies across nodes.
- **Distributed Control:** To achieve computational speedup and maintain stability in the event of hardware or network failures, control functions such as scheduling, resource allocation, and deadlock detection must be performed across multiple nodes.
- **Security Risks:** Data exchanged between processes on different nodes may traverse unmanaged systems and public networks. Attackers can exploit this by intercepting or fabricating messages to bypass authentication and impersonate legitimate users.

Network Operating System

This server-based program enables the management of a wide variety of network tasks, including data, users, groups, security, and applications. Using these operating systems on a small private network makes it easy to share files, printers, security settings, applications, and other network features. Another key characteristic that contributes to the perception of these systems being closely integrated is the fact that all users of a network operating system have access to each other's configuration settings, connections, and other information. The structure of network operating system is shown in Figure 1.11.

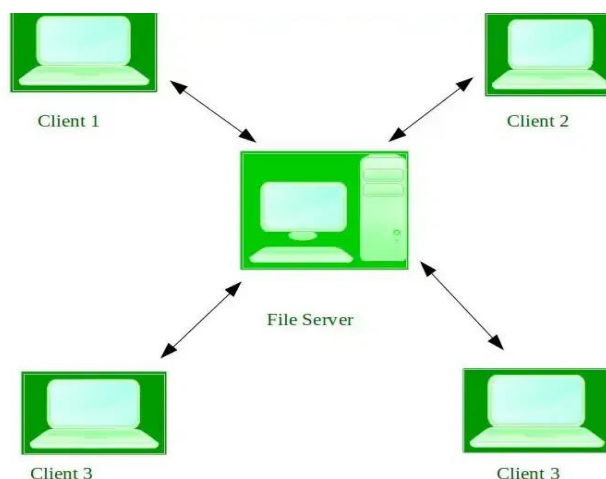


Fig 1.11 Network Operating System

Advantages of Network Operating System

- Highly reliable, centralized servers.
- Responsibility for handling security concerns lies with the servers.
- The system can easily adapt to new technologies and hardware changes.
- Remote access to the server is possible from multiple locations and on multiple computer systems.

Disadvantages of Network Operating System

- Servers can be quite expensive.
- Most operations require users to rely on a central location.
- Regular maintenance and upgrades are necessary.
- Examples of network operating systems include Linux, Novell NetWare, BSD, macOS X, Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, and many more.

Real-Time Operating System

Real-time operating systems are used in servers or systems that must process and respond to inputs with minimal delay. The response time refers to this duration. These systems are essential in applications where precise timing is critical, such as missile systems, air traffic control systems, and robotics. The structure of real-time operating system is shown in Figure 1.13.

Types of Real-Time Operating Systems

Hard Real-Time Systems: These operating systems are designed for applications with extremely strict timing requirements, where even the slightest delay is unacceptable. Examples include automated parachute systems and airbags, which are life-saving technologies that must activate instantly in the event of an accident. In most cases, these systems do not use virtual memory.

Soft Real-Time Systems: These operating systems are designed for applications where timely execution is important but not absolutely critical. A minor delay is tolerable and does not lead to system failure.

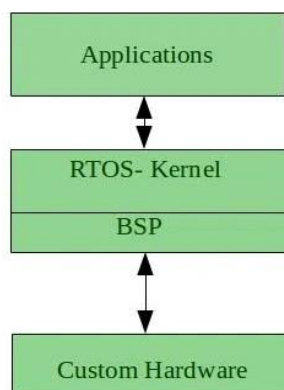


Fig 1.12 Real Time OS

Advantages of RTOS

- **Maximum Utilization:** Allows optimal use of all available resources by utilizing devices and systems to their fullest potential.
- **Fast Job Switching:** These systems offer very short switching times between tasks. In modern systems, task switching takes only about three microseconds, whereas in older systems, it used to take approximately ten microseconds.
- **Application Focus:** Gives more attention to currently running applications and less to those waiting in the queue.
- **Compact Program Size:** Due to their small overall program size, real-time operating systems (RTOS) are well-suited for embedded systems used in transportation and other industries.
- **High Reliability:** These systems are designed to operate with minimal errors. Memory allocation is handled in an extremely efficient manner.

Disadvantages of RTOS

- **Limited Task Handling:** To minimize errors, RTOS executes a relatively small number of tasks simultaneously and gives limited attention to background or queued applications.
- **High Resource Usage:** These systems make extensive use of system resources. While system resources are generally powerful, they can sometimes be expensive and unreliable.
- **Complex Algorithms:** The algorithms used are often very complex, making them difficult for designers to implement.
- **Interrupt Signals and Device Drivers:** To respond quickly to interruptions, these systems require efficient interrupt handling and reliable device drivers.
- **Thread Priority:** Setting thread priority is not advisable, as these systems are not typically designed for frequent task switching.
- **Real-time operating systems** are used in a variety of applications, including but not limited to scientific research, medical imaging, industrial control, defense systems, robotics, and air traffic management, among others.

Assignment 1.2

1. List down types of operating systems.
2. Compare Hard real time OS with Soft real time OS

1.5 Evolution of Operating Systems**1940s-1950s: Early Beginnings**

- System software was not necessary for computers to function.
- Each program had to be loaded and executed manually.
- The first operating system debuted in 1956.
- The GM-NAA I/O batch processing system (1956) automated job management.

1960s: Multiprogramming and Timesharing

- Multiprogramming was introduced to maximize the use of the central processing unit (CPU).
- Multiple users could interact with a single system through time-sharing systems such as CTSS (1961) and Multics (1969).

1970s: Unix and Personal Computers

- Unix (1971) dramatically changed OS design by introducing multitasking, portability, and simplicity.
- CP/M (1974) and PC-DOS (1981) were among the earliest personal computer operating systems.

1980s: GUI and Networking

- Graphical User Interfaces (GUIs) became widely adopted after the release of the Macintosh (1984) and Microsoft Windows (1985).
- Networking capabilities, including Unix's support for TCP/IP, became increasingly important.

1990s: Linux and Advanced GUIs

- Linux brought open-source development into the mainstream in 1991.
- Windows and OS X introduced advanced graphical user interfaces and gained widespread popularity.

2000s-Present: Mobility and Cloud

- Android (2008) and iOS (2007) became the dominant mobile operating systems.
- Operating systems like Linux and Windows Server have led advancements in cloud computing and virtualization, reshaping the computing landscape.

AI Integration - (Ongoing)

- As technology evolved, artificial intelligence became a core component of modern operating systems.
- OSs have improved significantly by integrating AI technologies such as Google Assistant, Alexa, and Siri.
- Features like voice commands, predictive text, and personalized recommendations are made possible through AI integration.

1.6 An Introduction to Windows

The Windows operating system is a computer program that provides services to applications running on top of it and manages all computer resources. Windows has undergone continuous development since its inception and is now one of the most widely used operating systems in the world. It is compatible with a wide range of hardware, including desktops, laptops, servers, and even mobile devices.

Unlike some other operating systems, Windows relies on a graphical user interface (GUI), allowing users to interact with the computer through visual menus, buttons, and icons. Among its many features are the ability to run multiple applications simultaneously and the convenience of connecting peripherals—such as printers, scanners, and cameras—through its plug-and-play functionality.

Advantages

- **Wide Compatibility:** Windows offers broad interoperability with a wide variety of hardware and software, allowing users to operate the OS on many different devices.
- **User-Friendly Interface:** Windows features an intuitive and widely recognized user interface, making it easy to learn and use—especially for those who are new to computers.
- **Built-in Multimedia Tools:** Tools like Windows Media Player make it easy to play a wide range of audio and video files.
- **Extensive Developer Support:** A large community of developers creates a vast range of applications and games for Windows, providing users with numerous options.
- **Long-Term Support:** Microsoft offers extended support for its Windows versions, including regular updates and security patches for several years after release.

Disadvantages

- **Vulnerability to Malware:** As the most widely used operating system globally, Windows is a frequent target for viruses and malware. Users must exercise extra caution.
- **High System Requirements:** The latest versions of Windows often require relatively high hardware specifications, which may not be supported by older PCs or laptops.
- **Licensing Costs:** Windows is not free; users must purchase a license to use the OS, which can be costly, especially for multiple devices.
- **Complex Feature Set:** At times, Windows can be overly complex, requiring users to invest time in learning and understanding all of its functionalities.
- **Security Concerns:** Like any operating system, Windows has its pros and cons. While its widespread compatibility and strong developer support are major advantages, users must take extra precautions to protect their systems from malware and security threats.

1.7 Setting up and installing Windows

Installing Windows is a straightforward process that typically takes less than an hour.

Configuration refers to setting up all the necessary system settings to ensure the installed operating system works effectively with hardware and other software components. This contrasts with installation, which involves placing the operating system onto the device.

Some common boot options include:

- **USB:** To boot from a USB drive, you first need to create a bootable USB using specific software. Then, copy the operating system onto the flash drive to begin the installation.

The ability to update data stored on the flash drive is a significant advantage—especially when installing the OS on multiple machines.

- **PXE:** PXE (Preboot Execution Environment) allows you to automatically install an operating system from a network drive. This method eliminates the need for physical media.

Type of installations

- **Image creation:** Before beginning, it's important to understand that the installer will need to clone the existing Windows operating system by creating sector-by-sector copies of all the computer's hardware and software data.

One common installation issue in large enterprises is the need for unattended installations. Due to the high volume of installation prompts, it is often necessary for an IT specialist or installer to be physically present during such installations. However, an `unattend.xml` file can be created and used in this situation, as it contains predefined answers to all the setup questions and prompts. By editing the XML file on the Windows installation disc, the setup can proceed automatically without user intervention.

- **Upgrade:** An in-place upgrade allows the existing operating system to continue running during the installation process. As soon as the installer detects an existing OS, it upgrades to the most recent version without modifying any existing data or applications.

There is a significant difference between an in-place upgrade and a clean installation, the two common types of operating system installations. A clean installation involves deleting all files on the machine, which means creating a backup of important data is essential to avoid loss. For this purpose, a cloud backup or external hard drive can be used. All documents and personalized settings are erased during a clean installation. Therefore, it is strongly recommended to copy these files to an external drive beforehand to ensure they are not lost.

- **Repair installation:** If the operating system files become corrupted or damaged—possibly due to a virus, Trojan horse, or a faulty hard drive sector—a repair installation is typically performed. A repair installation is usually one of the options that appear when Windows starts up. It allows you to reinstall the OS files without altering any settings or data on the hard drive. This type of installation is often sufficient to restore the integrity of the Windows operating system files.

The ability to boot into multiple operating systems is useful when you want to upgrade your OS without completely removing your current one. In such cases, you can set up a multiboot configuration, also known as a dual-boot setup. This setup requires two separate operating systems installed on different hard disk partitions. When configured correctly, you will see a prompt at startup asking you to choose between the old and new operating systems. For example, Windows 7 and Windows 8 can coexist on the same PC. Simply restarting the machine allows you to switch to the other operating system.

Another effective method for installing an operating system is through remote network installation. This approach is particularly useful when physical installation media is not required. Windows setup files and other configuration data can be stored on a network drive. Using the Preboot Execution Environment (PXE), a computer can connect to the network and retrieve all installation files stored on the shared drive. For this method, it is recommended to have a central server that manages the installation process.

If a machine needs to be deployed, image deployment is a simple process involving cloning the OS image to another computer. Since the new machine inherits the system information from the original setup, there is no need to run the full installation process. This becomes apparent during system startup.

If you're installing Windows for the first time, you will likely be prompted to partition your hard drive. Most modern computers use the Master Boot Record (MBR) partition table, which supports up to four primary partitions on a single hard drive. More recent systems use the GUID Partition Table (GPT), which supports up to 128 partitions.

Dynamic disk partitions allow data to span from a basic partition to newly added drives and prevent unauthorized drive additions. This feature is supported by several versions of Windows.

Microsoft introduced the concept of a basic disk partition, which can host an operating system like Windows or DOS. Basic disks support both physical and virtual partitions, including logical drives.

A partition from which an operating system can boot is called a primary partition. One of these can be designated as the active partition, which initiates the boot process when the machine powers on. This enables multiboot functionality.

When more than four partitions are required on a disk, extended partitions are used. These allow the creation of additional partitions but cannot be used to boot an operating system, making them less versatile than primary partitions. If you create a partition within an extended partition, it is referred to as a logical partition. Logical partitions also cannot be used to boot operating systems.

Various file system formats and types

To write data to a specific drive, file systems must first be established. It is essential to ensure that the chosen file system is compatible with the operating system. The OS should be able to read from and write to the file system efficiently. Some operating systems are more versatile than others in their ability to read various file systems.

FAT

The File Allocation Table (FAT) is one of the earliest and most continuously updated file systems for personal computers. FAT32, a newer version of FAT, was introduced in Windows 2000 and subsequent versions. It supports volumes up to nearly two terabytes and individual files up to four gigabytes in size.

NTFS

Compared to FAT32, the New Technology File System (NTFS) is a more advanced file system. It includes features such as built-in file compression, encryption, improved security, and data recovery capabilities—all without requiring third-party software. NTFS first appeared as Version 1.0 and has since undergone several improvements.

CDFS

The Compact Disc File System (CDFS) is included in Windows and other operating systems to support the reading of CDs and DVDs. It conforms to ISO standard 9660 and is compatible with Windows, UNIX, Linux, and macOS. CDFS simplifies the process of burning and reading data on optical discs.

Quick Format vs. Full Format

Quick formatting is a fast method of preparing a drive for use. It deletes all existing data and formats the drive with the selected file system, but it does not check for bad sectors. Note that Windows Vista and Windows 7 do not offer the quick format option during OS installation.

Full formatting, on the other hand, installs a new file system, erases all data, and scans the drive for bad sectors. This ensures that data can be reliably written to the storage device, making it a more thorough but time-consuming process.

Loading Third-Party Drivers

During OS installation, you may need to load third-party drivers if the setup program cannot detect certain disk controllers. In such cases, it is essential to have the required drivers available. These are especially critical for accessing internal storage devices when default drivers are missing or unsupported. Many OS installers allow users to manually load third-party drivers during setup.

Workgroup vs. Domain Setup

Configuring network settings is a necessary part of OS installation. You will typically be prompted to join either a domain or a workgroup. Joining a domain is common in enterprise environments, where systems are part of an Active Directory network. A workgroup, on the other hand, is typically used in home or small office settings, allowing local peer-to-peer networking without centralized administration.

Date, Time, Location, and Language Preferences

During installation, the system will prompt you to confirm the current date, time, region, and language preferences. These settings are important for ensuring accuracy across time zones and regions. While they may not affect a home system significantly, it is essential to verify them when setting up business machines, especially those used across multiple locations. The Windows OS offers a wide variety of language options—users who are not fluent in English should choose a language they are comfortable with.

Software, Driver, and Windows Update Installations

For the Windows setup to be considered complete, additional drivers often need to be installed. These may include graphics, chipset, wireless LAN, and other hardware-specific drivers. Windows updates should also be properly configured to keep the system up to date. Microsoft recommends enabling automatic updates so the OS can regularly check for and install the latest security patches and drivers. This helps maintain the system's stability, security, and performance.

Factory Recovery Disk

To ensure that the operating system includes a recovery option, a factory recovery partition is often created during installation. If it is not automatically created, users can manually set one up. This recovery partition allows users to restore the system to its original factory state if problems occur. It is especially useful for troubleshooting and system recovery without needing separate installation media.

Practical activity 1.1 Installation of Windows OS and configuring it.

Supplies Required for this Installation:

- Windows 11 Key
- USB Drive
- Computer
- Mouse, Keyboard and other essential peripherals

Step 1: Check That Your Computer Is Properly Installed.

1. Take the computer out of its packaging and discard the packing materials.
2. Insert the power cord into the computer's rear port.
3. Insert the opposite end of the power cord into an electrical socket.

Step 2: power on your computer and insert the DVD or USB drive containing the operating system installation software into the available slot or port.

Step 3: Set Up Boot Settings and Place Order

1. Give the computer a few moments to start up.
2. To access the settings in the BIOS or UEFI, press the corresponding key (often F2, F10, or Delete).
3. Get to the Boot tab by utilizing the arrow keys.
4. Fourth, go to the boot order list and use the arrow keys to put the OS installation disk or USB drive at the top.
5. To quit the UEFI or BIOS settings, press F10.

Step 4: Install the Operating System



1. Insert the operating system installation disc or USB drive into the computer and wait for it to boot.
2. To install the operating system, follow the instructions that are displayed on the screen.

3. Set the language and keyboard layout that you prefer.
4. Choose the partition or hard drive on which you would like to install the OS.
5. The installation process can be finished by following the remaining on-screen instructions.

Step 5: Once the operating system is installed, connect your computer to the internet.

1. Download and install any necessary drivers for your computer's hardware.
2. Install any additional software you need.

1.8 Architecture for Windows

The kernel of an operating system is its most essential and dependable component. While many general-purpose systems only make use of two protection rings, the hardware may allow for additional CPU modes. The Multics operating system was groundbreaking because it incorporated multiple rings of protection. Earlier versions of Windows were created to operate on processors with only two security levels.

Consequently, both Windows 7 and Windows Server 2008, along with their earlier versions, utilize just two rings: Ring 0 represents kernel mode, while Ring 3 signifies user mode. Similar to Unix, Windows NT does not fully leverage ring-based protection, even though numerous modern CPU architectures, such as the widely used Intel x86, support several rings.

In real-mode operating systems, minimal protection is provided as the system operates. For example, memory managers like EMM386 function at Ring 0 in the 386 architectures. In DOS, the kernel, drivers, and applications generally run in Ring 3, although this occurs only when protected-mode drivers and DOS extenders are utilized.

The x86 architecture specifies four different privilege levels, referred to as 'rings,' with each ring granting a varying degree of access and protection. Code executing in Ring 3 is designed to fail or crash without impacting the entire system, while code operating in Ring 0 enjoys unrestricted access to system resources. Rings 1 and 2 are seldom utilized in practice but can be set up with different levels of access when necessary.

In most modern systems, transitioning from user mode (Ring 3) to kernel mode (Ring 0) results in a significant performance overhead, which is an important consideration in system design and optimization. The Windows Operating System Architecture (Layered Structure with Protection Rings) is shown in Figure 1.13.

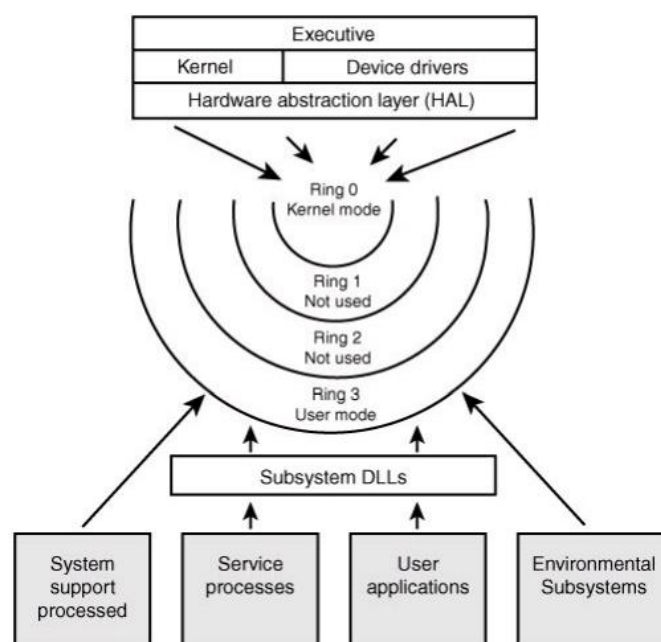


Fig 1.13 Windows Operating System Architecture (Layered Structure with Protection Rings)

Windows Memory

Starting processes on x86 Windows systems use a flat memory model with a range from 0x00000000 to 0xFFFFFFFF. The lower half of this memory range, 0x00000000 to

0x7FFFFFFF, is reserved for user-space code, while the upper half, 0x80000000 to 0xFFFFFFFF, is designated for kernel code.

Although Windows does not actively use segmentation in the traditional sense, it still relies on it at a structural level due to hardware requirements. The segment table includes descriptors that span the entire linear address space. Each execution mode (user and kernel) has its own data and code segments—two for user mode and two for kernel mode. However, all segment descriptors map to the same linear address space, from 0x00000000 to 0xFFFFFFFF, which confirms that Windows operates on a flat memory model and largely bypasses segmentation.

In this context, 'virtual address space' and 'linear address space' are used interchangeably. The user-space code resides in the linear address range from 0x00000000 to 0x7FFFFFFF. Once addresses are generated, they are passed to the paging unit, which translates them into physical addresses.

Although the system uses a 4 GB linear address space, individual processes can access only the lower 2 GB. The upper 2 GB is reserved for the kernel. This is why, even within a flat memory model, processes are effectively limited in the amount of addressable memory they can use.

Each process has its own page directory, pointed to by the CR3 register. Because of this, two processes can have the same linear address mapped to different physical addresses. While each process has its own private address space, all processes must access the same kernel. The kernel is loaded into memory only once, and each process has page directory entries (PDEs) that reference the same kernel page table, ensuring a shared and consistent kernel mapping.

Kernel memory is strictly protected. User-mode processes cannot directly access kernel memory; instead, they must go through special gateways—known as system calls—to request services from the kernel. Since the calling code resides in user space, it cannot transition arbitrarily into kernel mode. Programs can only invoke predefined kernel actions.

Running custom code in kernel space requires a method of safely transferring control to the kernel. This is typically achieved by installing a driver. Drivers run with kernel privileges because they interact directly with hardware. Once installed, drivers expose controlled interfaces that user-mode processes can access. However, the actions allowed are limited to the functionality the driver explicitly implements.

In conclusion, only code loaded into kernel mode can execute with kernel privileges. User-mode code cannot be granted such privileges directly, ensuring strict separation and security between user applications and critical system functions. The windows architecture is shown in Figure 1.14.

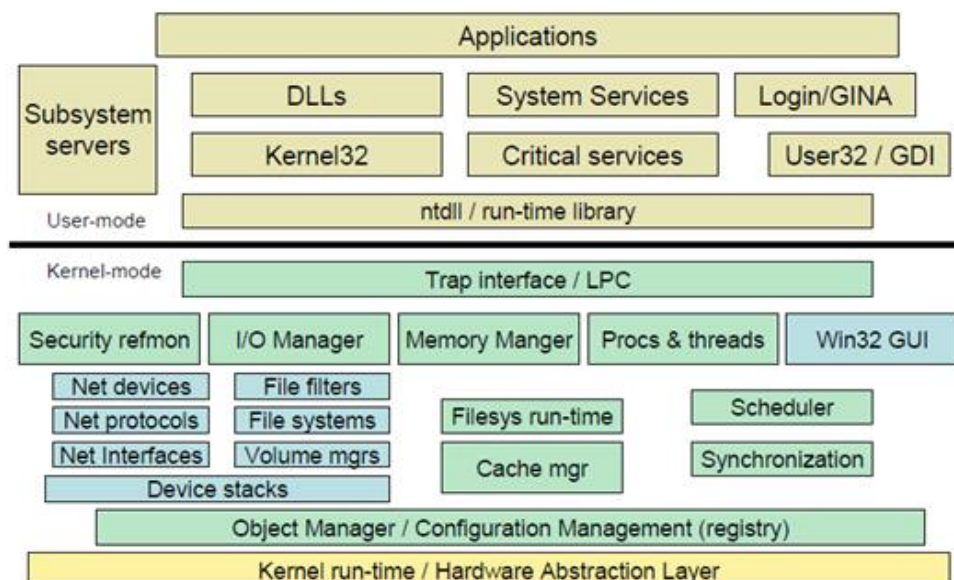


Fig 1.14 Windows Architecture

The Hardware Abstraction Layer (HAL), as shown above, serves as the foundational abstraction layer that shields the operating system (OS) from hardware-specific details. With HAL in place, the OS can utilize the same API functions regardless of the underlying hardware, while HAL handles their execution. No driver—not even those installed in kernel mode—interacts directly with the hardware; instead, all drivers rely on HAL for hardware access. Although kernel-mode drivers technically have the capability to communicate directly

with hardware, they typically prefer using HAL's APIs to ensure compatibility and system stability. The hal.dll library file, located in the C:\WINDOWS\system32\ directory, contains the implementation of the HAL API.

Operating System Security and Protection

Operating systems use different methods to protect both the software and hardware parts of a computer. Two important methods are called security and protection. Although these words are often used as if they mean the same thing, they actually have different meanings.

Security

Security is focused on protecting system resources (such as data, memory, and disks) from malicious activities, unauthorized access, corruption, and inconsistencies. It employs techniques such as authentication, authorization, and encryption to validate users before granting access to sensitive resources.

For example, while a company employee may be authorized to access certain internal data, an external or unauthorized user from another organization would be denied access. Therefore, implementing robust security is critical for any organization to protect confidential and sensitive information from breaches.

Protection

Protection refers to internal controls within the system—defining who can access what. It specifies user permissions and access rights for different resources. System protection mechanisms ensure that only authorized users and processes are permitted to interact with specific components such as the CPU, memory, and storage.

These mechanisms help prevent unauthorized use or manipulation of resources by validating users and processes against predefined access policies. For instance, in a typical organization structured with multiple departments, each unit may share general data while restricting access to sensitive information. Based on roles and responsibilities, different users will have different levels of access privileges.

An effective protection system not only restricts access but also outlines which controls must be in place and how they should be enforced, thereby supporting operational integrity and compliance.

Differences Between the Security and Protection in Operating Systems

Operating system security and protection are not the same. The difference between them can be explained through the following key points:

1. Security helps the operating system handle external threats and ensures the system runs properly. On the other hand, protection is used to control access within the system and manage internal risks.
2. Security decides whether a user is allowed to use the system. In contrast, protection defines what resources a user can access and how they can use them.
3. Examples of security measures include adding or removing users, checking their permissions, using antivirus software, and more.

Various OS User Interfaces

Operating systems (OS) rely on user interfaces (UIs) to let people control and interact with their machines. Many different kinds of interfaces, each tailored to specific needs and preferences, allow this interaction to take place. A complete overview of the most common OS user interface types is as follows:

1. Command-Line Interface (CLI)

A method of interacting with a computer that relies on typing commands into a window or terminal. You can find it in the Terminal on macOS, the Command Prompt in Windows, and PowerShell or Bash on Linux and Unix systems.

Advantages:

- **Efficient and Fast:** Skilled users can perform complex tasks easily.

- **Low Resource Usage:** CLI uses far fewer system resources compared to graphical interfaces.
- **Scripting:** Works well with scripts (like shell scripts) that automate tasks.

Disadvantages:

Learning Curve: New or casual users may find it difficult to learn.

Memorization Required: Commands and syntax need to be remembered.

2. Graphical User Interface (GUI)

Uses windows, icons, menus, and pointers to allow users to interact with the operating system. It is standard in most modern operating systems, including Windows, macOS, and many Linux distributions (such as GNOME and KDE).

Advantages:

- **Easy to Use:** Simple and intuitive, even for beginners.
- **Accessibility:** Provides visual elements and features to help all users.
- **Multitasking:** Allows multiple applications to run and be switched between easily.

Disadvantages:

- **High Resource Usage:** Consumes more RAM and CPU than CLI.
- **Slower Performance:** May be slower for some tasks compared to CLI.

3. Touch User Interface (TUI)

Uses touch controls on smartphones, tablets, and touch-enabled laptops. Common in mobile operating systems (like iOS and Android) and desktop systems with touch support (Windows 10 and 11).

Advantages:

- **Intuitive Gestures:** Natural interaction using gestures like swipe, tap, and pinch.
- **Convenient:** Ideal for portable devices.
- **Accessibility:** Useful for people with mobility challenges.

Disadvantages:

- **Less Precise:** Not as accurate as using a mouse for some tasks.
- **Fingerprints:** Screens get smudged, affecting visibility and usage.

4. Voice User Interface (VUI)

Allows users to control the operating system through voice commands. Found in digital assistants like Alexa, Siri, Google Assistant, and Cortana.

Advantages:

- **Hands-Free Use:** Helps with multitasking and improves accessibility.
- **Natural Interaction:** Some tasks are easier using spoken commands.

Disadvantages:

- **Accuracy Issues:** Background noise can affect voice recognition.
- **Privacy Concerns:** Security and privacy of voice data are important concerns.

5. Gesture-Based Interface

Uses sensors and cameras to detect user gestures for interaction. Used in virtual reality systems, gaming consoles (like Microsoft Kinect), and advanced touch devices.

Advantages:

- **Immersive Experience:** Makes gaming and virtual environments more engaging.
- **Helpful for Accessibility:** Useful for users with limited physical movement.

Disadvantages:

- **Difficult to Master:** May be complex and hard to control accurately.
- **Environmental Requirements:** Needs proper lighting and space for accurate detection.

Practical 1.2 Creation of disk partitions and configure Active directory.

One way to arrange your data in Windows 10 is to create partitions. This will allow you to isolate your operating system files from your personal files. Disk partitioning in Windows 10 is possible in a few different ways, allowing you to better organize your data. If you want

to know how to optimize your computer and partition a hard drive in Windows 10, this is the instruction for you.

Before starting:

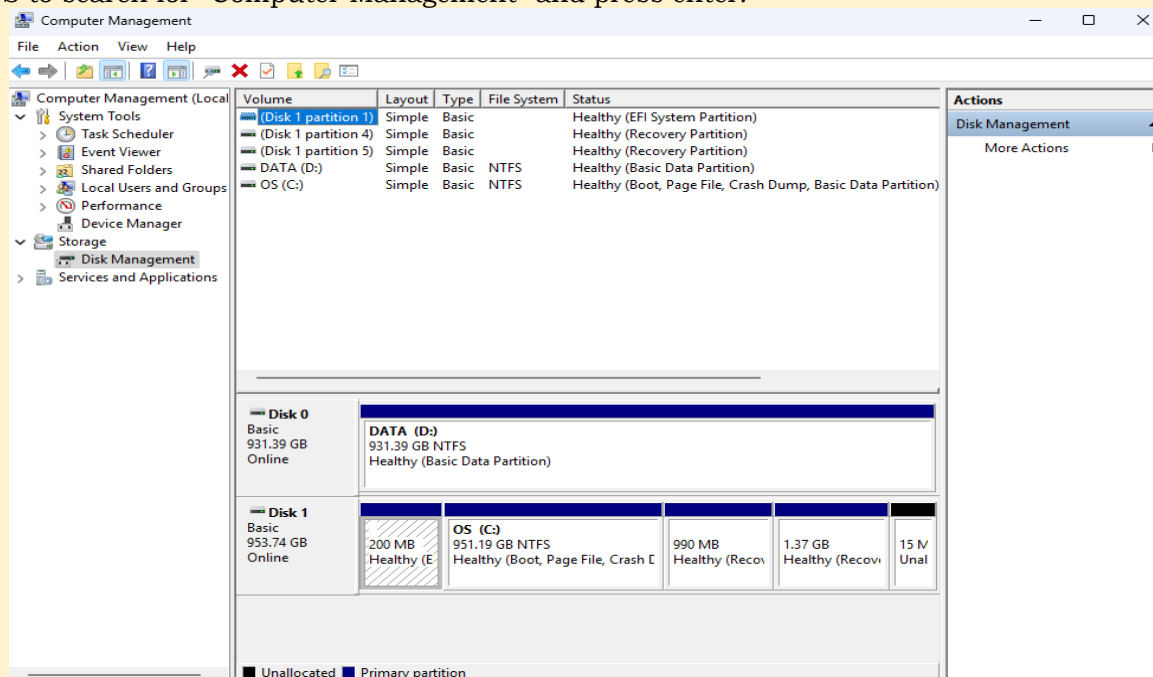
1. Make sure there is enough room on your hard disc.
2. Get an administrator to back up your data.

1. Using Disk Management

You can manage system drives (including partitions) with the help of this built-in utility that Windows OS provides. Here are a few simple steps to help you achieve it:

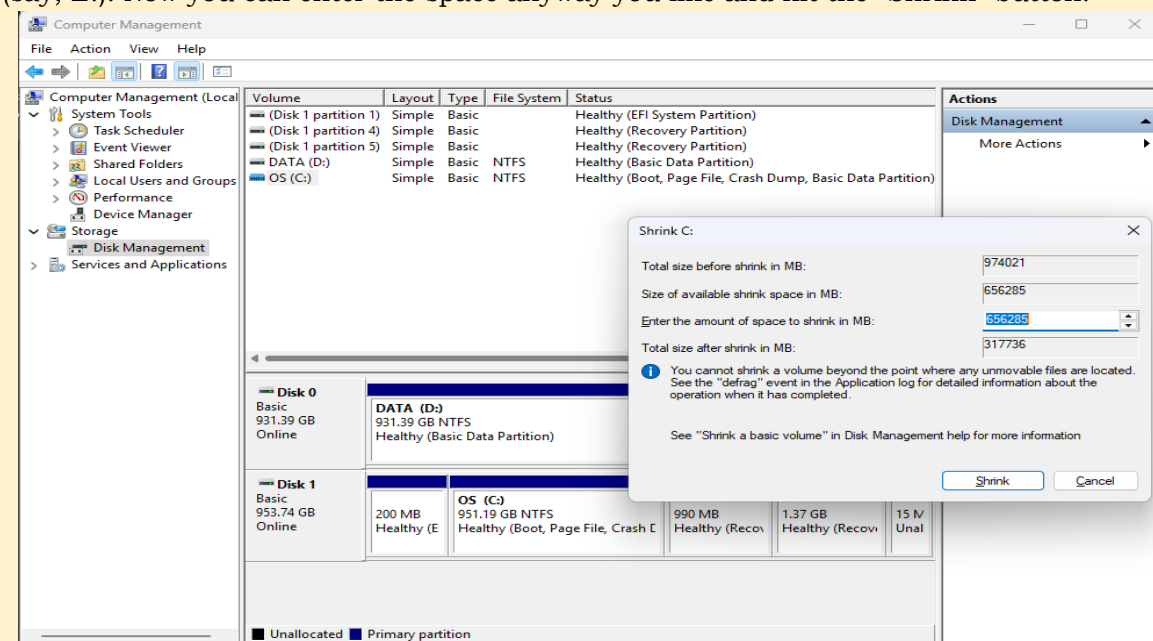
Step 1: Go to Windows Disk Management Settings and open it.

Choose "Disk Management" from the left-hand menu after pressing Win + X, or use Win + S to search for "Computer Management" and press enter.



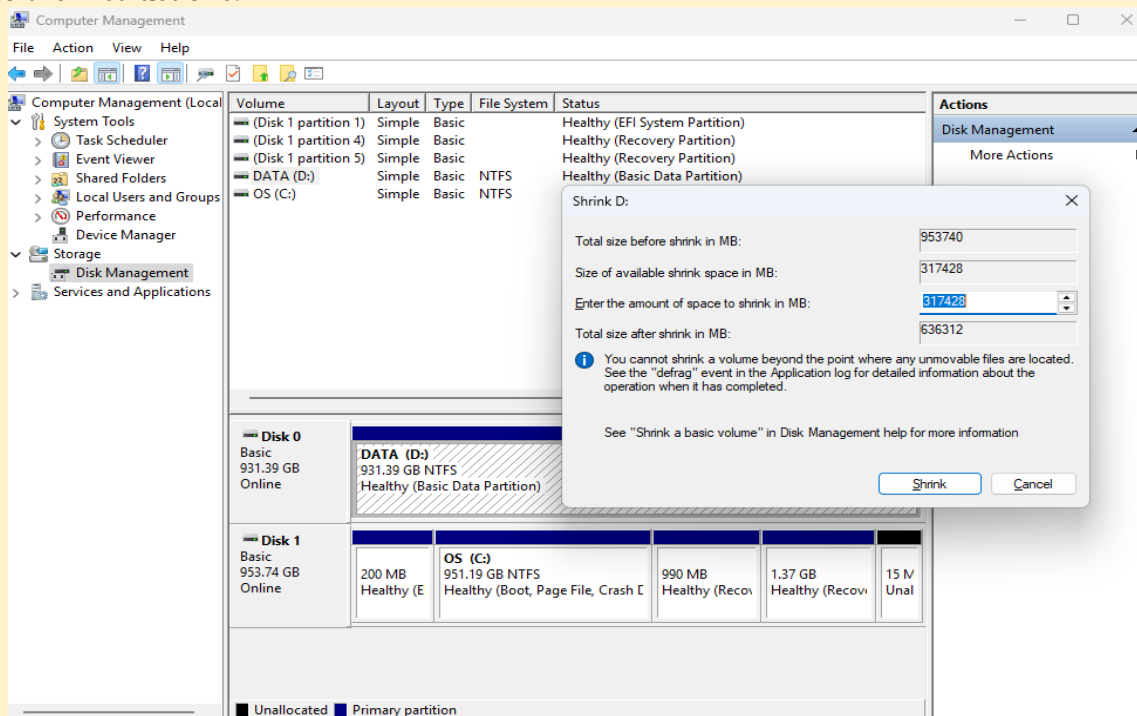
Step 2: Shrink Volume of the Selected Drive

Select "Shrink Volume" from the context menu after right-clicking the current partition (say, E:). Now you can enter the space anyway you like and hit the "Shrink" button.



Step 3: Create a New Volume

Choose "New Simple Volume" from the context menu after right-clicking the empty area. Assign the new drive letters (A, B, D, G, etc.) and select NTFS as the file system according to the instructions.

**Step 4: Format and Complete the Process**

Press "Finish" after selecting "Format" in Windows 10 to create a new disk partition.

2. Using DiskPart (Command-Line Interface)

DiskPart provides powerful command-line control over disk, partition, and volume management, suitable for scripting and advanced scenarios.

Step 1: Accessing DiskPart: Type cmd in search, right-click > Run as Administrator or Open an elevated PowerShell window (Right-click Start button > Windows PowerShell (Admin)). Press Enter and type diskpart. The prompt will turn to DISKPART>

Step 2: Select the Disk you want to Format

Now type the following command to select the desired disk:

```
select disk A
```

(A : disk name)



Select Disk

Step 3: Type the following command to assign the disk size, disk name and disk format

```
create partition primary size=20480
```

```
assign letter=H
```

```
format fs=ntfs quick
```

Disk 0 is now the selected disk.

```
DISKPART> create partition primary size=20480
```

DiskPart succeeded in creating the specified partition.

```
DISKPART> assign letter=H
```

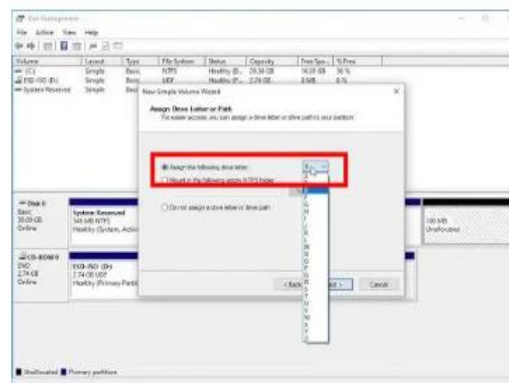
DiskPart successfully assigned the drive letter or mount point.

```
DISKPART> format fs=ntfs quick
```

Complete the partition

select disk A

(A : disk name)



Select Disk

Step 4: Type the following command to assign the disk size, disk name and disk format

```
create partition primary size=20480
```

```
assign letter=H
```

```
format fs=ntfs quick
```

Disk 0 is now the selected disk.

```
DISKPART> create partition primary size=20480
```

DiskPart succeeded in creating the specified partition.

```
DISKPART> assign letter=H
```

DiskPart successfully assigned the drive letter or mount point.

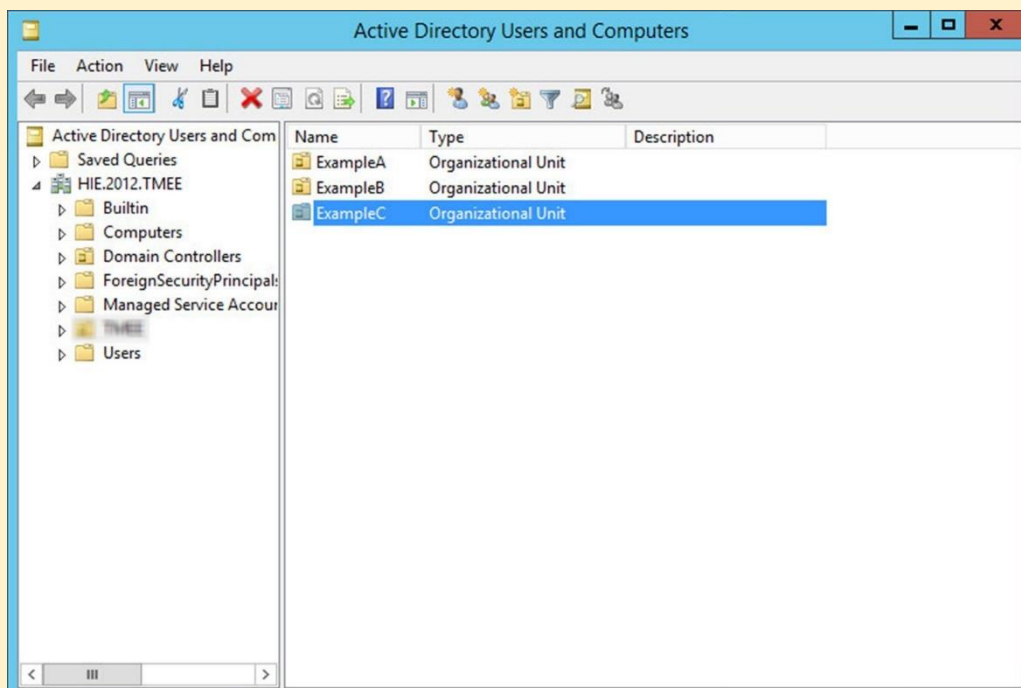
```
DISKPART> format fs=ntfs quick
```

Complete the partition

- **Listing Disk and Partition Details:** List disk to see all available disks. Whether the disk is GPT, its status, total size, free space, and disk number are all displayed by this command.
- **See partitions** on the chosen disk by listing partition. Beginning from the disk's start, this command returns the Partition Number, Type, Size, and Offset. While Recovery, System, and Reserved partitions are system-created and should not be changed, the Primary partition normally houses user and operating system data and should not be changed.

Configuring Active Directory

Step 1: Go to Start > Administrative Tools > Active Directory Users and Computers. The Active Directory Users and Computer screen appears.



Step 2: Create your organizational units (OUs).

For each OU you intend to create, perform the following steps:

- Right-click the new domain created during AD installation and then select **New**.
- Select **Organizational Unit**.
- From the New Object - Organizational Unit screen, specify the new name and click **OK**.

The new group appears in the left navigation under the domain name. Perform this step for as many organizational units you intend to use with PolicyServer.

The new groups will be used to synchronize with a PolicyServer group. Before synchronization, users must be added to the groups.

Step 3: Add new users to your OUs.

For each user you intend to create, perform the following steps:

- Right-click the intended OU and go to New > User.
- From the New Object - User screen, specify the new user's account information and click Next.
- Specify and confirm the new user's domain password and click Next.
- When prompted to complete, click Finish.

The domain controller is configured with a new OU and a user in that group. To synchronize that group with PolicyServer, install PolicyServer and create a group for synchronization.

1.9 The Difference Between Windows and Linux

One widely used operating system standard is Linux, which is both free and open-source. It provides a programming interface that is compatible with various software applications and other operating systems. Additionally, it supports a wide range of applications. The UNIX operating system is also interoperable and does not contain proprietary code, as it is made up of different components developed by various teams.

Windows: On Windows systems, an ASCII text file may not always be accessible. Windows is mainly designed for commercial and industrial use, as well as for users who may not have programming knowledge. It is easy to use and user-friendly.

Linux is free and open-source, whereas Windows is a commercial and paid operating system. An operating system is software that acts as a bridge between the user and the computer hardware. Linux users have access to ASCII text files and can modify the system's code to make improvements. On the other hand, Windows is a proprietary system that does not allow users to access or modify ASCII text files.

We can compare Linux and Windows by examining their differences:

Practical Activity 1.3: Demonstration of the difference between Windows OS and Linux OS (Through simulation).

Go through the simulation below:

Differences Between Windows and Linux – A Guide for Windows Users by FOSS and LINUX journal

Link for simulation: <https://www.youtube.com/watch?v=GM2chYafL3g>

Let us discuss the difference between Windows and Linux using a few simulation scenarios:

Simulation Scenario: Onboarding a New User with Specific Access Needs

Consider the task of onboarding a new employee who requires specific, nuanced access to various system resources.

Windows: While explicitly denying access to the "HR Records" folder, an IT administrator must onboard a new project manager who requires full control over their personal files, read/write access to the "Project X" shared folder, and read-only access to the "Company Policies" folder. The administrator can create the user account, add the user to relevant groups (e.g., "Project X Team"), and then apply specific ACL entries to each folder using Active Directory and NTFS ACLs. The "deny" entry for "HR Records" would take precedence, ensuring that the project manager cannot access it, even if they belong to a broader group that otherwise has access. The granular nature of ACLs and centralized administration tools simplifies this process.

Linux: Using traditional POSIX permissions, the IT administrator attempts to achieve the same level of granular access for the new project manager. They can easily assign ownership and basic read/write/execute rights for personal files. However, providing varying levels of access to shared folders for different groups—or denying access to a specific user within a group—proves more difficult without affecting others. If remote users are involved, the administrator would need to enable and configure POSIX ACLs on the filesystem and possibly create more complex group structures or integrate with an external directory service (such as LDAP) to achieve the desired granularity. This typically requires a deeper understanding of Linux's permission system and additional manual steps.

This comparison highlights the differences between fine-grained, system-level access control and centralized enterprise identity management. Windows, with its tight integration with Active Directory, provides a powerful, centralized solution for managing users, groups, and permissions across large enterprise networks. This ensures consistent enforcement of security policies and reduces administrative overhead. While Linux offers strong control through POSIX permissions and ACLs, seamless integration with enterprise-grade directory services for managing remote users and groups often requires more manual configuration and specialized knowledge. Although Linux can achieve similar levels of security and access control, large-scale deployments typically demand a greater investment in expertise and setup.

Simulation Scenario: Setting up a New Peripheral Device

Consider the experience of setting up a new peripheral device, such as a high-end gaming mouse or a specialized scientific instrument.

Windows: A user connects a new high-end gaming mouse to their Windows PC. Windows' plug-and-play feature recognizes the device immediately, and typically, a generic driver is automatically loaded to provide basic functionality. Usually, the user visits the manufacturer's website to download a proprietary driver and accompanying software suite to enable enhanced features (e.g., custom macros, RGB lighting). Generally, this process is straightforward, as manufacturers prioritize Windows compatibility due to its dominant market share.

Linux: A user connects the same high-end gaming mouse to a Linux PC. The Linux kernel likely includes a generic driver that provides basic functionality, allowing the mouse to work immediately. However, accessing advanced features is a different scenario. If the manufacturer does not provide a Linux driver, the user may need to search for a community-developed open-source driver—if one exists—or may have to manually compile and install it. Performance may not match that of the Windows version, and in some cases, there may be no support at all. For a specialized scientific instrument, the setup process can be even more complex, as the chances of a ready-made Linux driver from the manufacturer are lower, often requiring custom development or reliance on community support.

This comparison highlights a fundamental difference in hardware support: vendor-driven integration versus community-driven adaptation. Through strong manufacturer partnerships and its Plug and Play system, Windows prioritizes broad, out-of-the-box compatibility for mainstream devices. This approach is driven by market incentives, as manufacturers typically prioritize building drivers for the leading desktop OS. Conversely, Linux largely relies on kernel-included and community-developed open-source drivers. This offers greater freedom and often better performance for older hardware but can result in slower support for the newest or most niche devices, particularly when manufacturers do not offer official Linux support. As a result, hardware lifespan may be extended under Linux, as community-driven support for older devices often continues long after manufacturers discontinue Windows drivers.

1.10 Ubuntu vs. Windows

The following Table 1.1 highlights the differences between Linux and Windows operating system.

Table 1.1 Comparison between Linux and Windows

S. No	Linux	Windows
1.	Linux is an open-source operating system	Windows is not an open-source operating system.
2.	Linux is free of cost.	Windows is paid and requires a license.
3.	File names are case-sensitive, meaning file.txt and File.txt are different.	File names are case-insensitive, meaning file.txt and File.txt are treated the same.
4.	Uses a monolithic kernel.	Uses a hybrid kernel.
5.	More efficient and stable, especially for servers and developers.	Less efficient due to resource intensive processes.
6.	Uses forward slash (/) for directory separation.	Uses backslash (\) for directory separation.
7.	More secure with better user control and fewer vulnerabilities.	Less secure due to higher susceptibility to malware and viruses.

8.	Preferred by hackers and security experts due to its open- source nature and control.	Not widely used for hacking as it lacks built-in security tools.
----	---	--

1.11 LINUX

Operating System Linux (Linux) is an open-source, free, UNIX-based OS. The programming interface it offers is compatible with systems that rely on operating systems. It provides a wide range of applications and can be easily customized. It is widely used in servers, development environments, cybersecurity, and embedded devices.

A UNIX-based operating system that is both compatible and devoid of proprietary code is the result of various components developed by different contributors. Developers, system administrators, and ethical hackers prefer Linux because it allows them to access, modify, and distribute its source code.

Linux distributions (distros) like Ubuntu, Debian, Fedora, and Arch Linux provide users with complete access to the source code, enabling them to optimize and modify their systems—unlike Windows. This is one of Linux’s key features. Compared to Windows, Linux offers better protection against malware and viruses, fewer vulnerabilities, and built-in user privilege restrictions, making it a highly secure operating system. Linux distributions such as Arch Linux and Linux Mint are known for their efficiency and lightweight nature, allowing them to outperform Windows, especially on older hardware.

A Wide Variety of Software Support: Linux and its ecosystem support many different kinds of applications, including both proprietary and open-source options. Users can install programs using package managers like Pacman (for Arch Linux), APT (for Debian/Ubuntu), and DNF (for Fedora).

Linux vs. Windows for Developers: Linux is ideal for data scientists, DevOps engineers, and developers due to its robust tools, such as terminal-based programming, shell scripting, and support for numerous programming languages.

Rolling Release Model: Arch Linux and some other Linux distributions use a rolling release strategy, meaning users always have the latest updates without needing to install a new version of the OS.

What are the Common Uses of Linux?

Linux distributions such as CentOS, Ubuntu Server, and Debian power the majority of the internet and enterprise cloud infrastructure, making Linux the dominant language in web servers.

- Ethical hacking, network security, and penetration testing are all areas where Linux shines thanks to tools like Kali and Parrot OS.
- Steam Proton, Lutris, and Wine are just a few of the recent advancements that have greatly enhanced Linux gaming, making it possible to play games developed for Windows with much less hassle.

Practical Activity 1.4: Demonstration of different commands on CMD on Windows OS & familiarity of these commands with LINUX.

- **Windows CMD:**

- **The cd command (Change Directory)** is used to move between folders.
- Typing `cd <directory_name>` changes to a specified subdirectory, while `cd..` navigates up one level in the directory hierarchy. To return to the root of the current drive, `cd \` is employed.

```
cd [directory]
```

- For example, to change the location to a directory named *Public*

```
cd Public
```

- **The dir command** lists files and subdirectories within the current directory. It provides details such as file size, modification date, and attributes.

```
dir [drive][path][filename] [options]
```

```
dir C:\
```

```
C:\Users>dir C:\
Volume in drive C is Windows
Volume Serial Number is 0887-04E0

Directory of C:\

11/26/2013  08:53 PM           678,584 CMTrace.exe
12/21/2022  06:02 PM    <DIR>          EFI
01/15/2021  06:30 PM    <DIR>          Intel
12/07/2019  10:14 AM    <DIR>          PerfLogs
12/21/2022  06:36 AM    <DIR>          Program Files
11/21/2022  11:44 AM    <DIR>          Program Files (x86)
05/23/2022  08:09 AM    <DIR>          Snagit
05/09/2022  01:35 PM    <DIR>          Users
12/31/2022  01:07 PM    <DIR>          Windows
               1 File(s)        678,584 bytes
               8 Dir(s)  172,534,497,280 bytes free
```

- **Linux:** Similarly, `cd <directory_name>` changes the current directory. `cd..` moves up one level, and `cd/` navigates to the root directory of the entire file system.
- The `ls` command lists directory contents. For a detailed list, `ls -l` displays permissions, ownership, size, and modification date. Hidden files, which typically start with a dot, can be shown using `ls -a`.

```
ls -l
```

```
ls -l -a
```

- **Creating and Removing directories**
- Commands for creating new directories and removing existing ones are core to file system management.

Windows CMD:

`mkdir <directory_name>` creates a new directory.

`rmdir <directory_name>` removes an empty directory.

```
md [path]
```

```
mkdir [path]
```

To remove a directory and all its contents without prompting for confirmation, `rmdir /s /q <directory_name>` is used.

```
rd [path]
```

```
rmdir [path]
```

Linux:

- * `mkdir <directory_name>` creates a new directory.
- * `rmdir <directory_name>` removes an empty directory.
- * To recursively remove a directory and its contents, `rm -r <directory_name>` is employed.
- * For forceful, recursive removal without prompts, `rm -rf <directory_name>` is the command.

```
mkdir Invoices
```

```
rm -r Invoices/
```

- **Copying, Moving, and Renaming**

These commands are essential for duplicating, relocating, or renaming files and directories.

- **Windows CMD:**

copy <source> <destination> copies one or more files to a specified location.

```
copy [options] [source] [destination]
```

xcopy <source> <destination> /E /I /H /K /Y is a more robust command for copying files and directories, including subdirectories (even empty ones), hidden/system files, and overwriting existing files without prompting. It is more suitable for complex directory structures than copy.

- For example, use the following command to copy contents from one location to another, including subdirectories (even if empty):

```
xcopy [source] [destination] /s /e
```

robocopy <source> <destination> [options] (Robust File Copy) is an advanced utility designed for mirroring directories, resuming interrupted transfers, and handling intricate synchronization tasks, making it ideal for backup scenarios.

```
robocopy [source] [destination] [file] [options]
```

rename <old_name> <new_name> (or its shorthand ren) renames a file.

```
ren [path][old_name] [new_name]
```

```
rename [path][old_name] [new_name]
```

move <source> <destination> relocates files or directories.

```
move [options] [source] [destination]
```

- **Linux:**

cp <source> <destination> copies files or directories. For recursive directory copying, cp -r is used.

```
cp fstab /home/alex
```

- Wherever you put ~, it's the same as typing the full path to your home directory, **/home/alex**.
- **mv <source> <destination>** moves files or directories. Significantly, mv <old_name> <new_name>

```
mv Invoice.pdf Invoices/
```

- **Advanced Network Settings**

These commands configure advanced network settings, manage the ARP cache, and identify the host.

- **Windows CMD:**

arp -a displays current ARP (Address Resolution Protocol) entries, showing IP addresses and their corresponding MAC addresses.

```
arp [options] [address]
```

hostname displays the computer's network name.

```
hostname
```

```
C:\Users\Public\Documents>hostname
DESKTOP-M9TP6I7
```

- **Linux:**
iptables configures firewall rules, serving as a powerful and complex tool for packet filtering.

```
iptables --table TABLE -A/-C/-D... CHAIN rule --jump Target
```

arp manages the ARP cache.

```
arp [-v] [-i if] [-H type] -a [hostname]
```

Screenshot of hosts before adding

```
root@kali:~# more /etc/hosts
127.0.0.1    localhost
127.0.1.1    kali
```

Addition of host

```
root@kali:~# arp -a
_gateway (10.0.2.1) at 52:54:00:12:35:00 [ether] on eth0
root@kali:~# ping 10.0.2.4
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data:
64 bytes from 10.0.2.4: icmp_seq=1 ttl=64 time=0.541 ms
64 bytes from 10.0.2.4: icmp_seq=2 ttl=64 time=0.280 ms
^C
--- 10.0.2.4 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1031ms
rtt min/avg/max/mdev = 0.280/0.410/0.541/0.132 ms
root@kali:~# arp -a
? (10.0.2.4) at 08:00:27:ad:87:b3 [ether] on eth0
_gateway (10.0.2.1) at 52:54:00:12:35:00 [ether] on eth0
root@kali:~#
```

Hosts file after adding machines

```
root@kali:~# more /etc/hosts
127.0.0.1    localhost
127.0.1.1    kali
10.0.2.4     machine1
10.0.2.5     machine2
```

Now checking arp for all

```
root@kali:~# arp -a
machine1 (10.0.2.4) at 08:00:27:ad:87:b3 [ether] on eth0
machine2 (10.0.2.5) at 08:00:27:27:d6:c7 [ether] on eth0
_gateway (10.0.2.1) at 52:54:00:12:35:00 [ether] on eth0
```

hostname displays or sets the system's hostname.

```
hostname -[option] [file]
```

1.12 Windows

Designed with simplicity, software compatibility, and enterprise applications in mind, Microsoft's Windows is a proprietary operating system. Unlike Linux's open-source nature, Windows is a closed-source OS that discourages users from making changes to its source code.

The broad software ecosystem, user-friendly Graphical User Interface (GUI), and plug-and-play capabilities of Windows make it a popular choice for personal computers, gaming, business solutions, and workplace settings. With its straightforward point-and-click interface, it is ideal for users who do not have coding knowledge. One of the key differences between the two operating systems is that Windows requires a paid license for installation and use, whereas Linux is free and open-source. A product key or subscription model (such

as Windows 365) is required to activate any edition of Windows 10, including Home, Pro, and Enterprise, each catering to different user needs.

Windows Essentials Interface for Graphical User Experience (GUI): A visually pleasing and intuitive setup that includes the Start menu, taskbar, and File Explorer.

- **Software Compatibility:** Runs Adobe Photoshop, Microsoft Office, and many third-party applications that are often unavailable or limited on Linux. It is the best option for gamers due to its gaming performance, compatibility with AAA titles, and support for high-end GPU drivers (NVIDIA, AMD).
- **Hardware Support:** Compatible with a wide range of USB peripherals, printers, and scanners; typically, no additional drivers are required.
- **Frequent Updates:** Receives bug fixes and new features via Windows Update. However, major version upgrades (e.g., Windows 10 to Windows 11) usually require reinstallation.

While Windows may be the most widely used operating system, Linux distributions offer superior flexibility, security, and customization options. Since Windows is not open-source, users cannot modify its code, making it less customizable than Linux-based systems like Fedora, Arch Linux, or Ubuntu.

1.13 Linux or Windows:

Whenever you must decide whether to use Windows or Linux keep following points in your mind. Your requirements, level of technical knowledge, and personal preferences will determine whether you should use Linux or Windows. Because of their unique strengths, the two operating systems are best suited to distinct user bases. Let me break it down for you so you can choose the OS that works best for you.

Choose Linux if:

- Linux has fewer vulnerabilities and greater access control, making it inherently more secure. If you value security and privacy, Linux is a good choice.
- You're looking for an open-source operating system (OS) that is free, allows extensive customization, and doesn't require licensing.
- Distributions like Ubuntu, Arch Linux, and Fedora enable fine-tuning, which is perfect if you value customization and control.
- Linux's powerful command-line tools, scripting capabilities, and server optimizations make it ideal for developers and system administrators.
- If you're a penetration tester or work in ethical hacking, you're likely familiar with Kali Linux and Parrot OS.
- Linux runs efficiently on low-end devices with minimal resource usage, making it a great option for older hardware needing a lightweight OS.
- If you prefer a rolling release model, distributions like Arch Linux provide continuous updates instead of large version upgrades.

Choose Windows if:

- You're looking for a user-friendly operating system with plug-and-play support—Windows is designed for users who are not highly tech-savvy.
- You need proprietary software like Adobe Photoshop or Microsoft Office. Windows also offers better gaming support due to DirectX, NVIDIA/AMD drivers, and broad compatibility with major game titles.
- Windows automatically detects and works with most peripherals, printers, and USB devices, reducing concerns about hardware compatibility.
- You value access to official customer support—unlike Linux, which primarily relies on community forums, Microsoft provides dedicated support for its users.
- Most businesses rely on Windows-based software for daily operations, so if you work in a corporate setting or are a business user, Windows is likely the better fit.

Create/manage users and permissions.

Types of User Accounts

Windows-based operating systems feature three primary local account types that dictate a user's capabilities on a single machine. These include **Standard User accounts**, designed for everyday computing, which allow users to perform routine tasks such as browsing the internet, running applications, and creating documents. However, these accounts have limited privileges and require Administrator credentials to execute system-wide changes,

including installing software, modifying the registry, or altering critical system files. This restriction is enforced by User Account Control (UAC), which prompts for elevated rights when necessary.

Administrator accounts provide the most extensive control over a computer. Users with Administrator privileges can access all system files, modify settings, manage other user accounts, install and uninstall applications, and configure network and firewall settings. Essentially, an Administrator account holder can perform any configuration-related task on the computer.

Guest accounts are intended for temporary use by individuals who do not have a permanent account on the system. They offer highly restricted privileges, preventing guest users from changing PC settings, installing applications, or accessing private files belonging to other users. It is important to note that traditional Guest accounts have been deprecated in Windows 10 and 11. Microsoft now recommends creating a restricted local account to emulate this functionality, providing a more controlled temporary access solution.

Beyond these local account types, Windows integrates with different account identities that extend functionality and management scope.

Microsoft Accounts, often referred to as Personal Accounts, are linked to an email address and password used across a wide array of Microsoft products and services, such as Outlook.com, Xbox, Skype, Microsoft 365, OneDrive, and the Microsoft Store. When a user signs in to a PC with a Microsoft account, it enables seamless integration with Microsoft cloud services, enhanced security features like Windows Hello, and synchronization of settings and files across multiple devices.

Microsoft Entra Work or School Accounts are distinct identities created and managed by an administrator within an organization. These accounts are typically associated with Microsoft 365 for business or Microsoft 365 Education products and are specifically designed for organizational use, offering advanced security and compliance tools to protect sensitive corporate data. Management of these accounts often occurs via the Microsoft Entra admin center and command-line tools like PowerShell.

In contrast to Microsoft accounts, **Local Accounts** are specific to a single device and do not require internet connectivity for sign-in. They operate independently of cloud services, meaning that user settings, files, and applications are confined to that particular device. While Microsoft generally recommends using a Microsoft account for its integrated benefits, local accounts can be preferred for privacy or when custom control over updates and personalization is desired.

Finally, if a computer is part of a larger network domain, such as an Active Directory environment, **Domain Accounts** come into play. These accounts introduce additional account types and permissions that are typically managed by a central network administrator, allowing for centralized control over users and resources across an entire organizational network. The Table 1.2 below shows the different account types and their specifications.

Table 1.2. Different account types and their specifications.

Account Type	Primary Purpose	Key Characteristics/ Privileges	Recommended Use Case	Security Implications
Standard User	Everyday computing	Limited privileges; requires admin credentials for system changes (UAC prompts)	Daily tasks (browsing, email, apps)	More secure; limits malware damage, accidental changes
Administrator	Full system control	Unrestricted access to system files, settings, user management, software installation	System setup, maintenance, managing other users	High risk; prime target for attackers; use sparingly

Guest/Restricted Local	Temporary access	Very limited privileges; no access to private files or system changes	Visitors, shared kiosks (Windows 10/11 requires manual creation of restricted local account)	Enhances privacy for primary user; system protection
Microsoft Account	Personal use, cloud services access	Syncs settings/files across devices; access to Outlook, Xbox, OneDrive, Microsoft 365, etc.	Personal devices to leverage Microsoft ecosystem	Enhanced security features (Windows Hello); cloud integration
Work or School Account	Organizational use	Managed by IT; associated with Microsoft 365 Business/Education; advanced security/compliance	Corporate environments, accessing sensitive organizational data	Strong security and compliance tools; centralized management
Local Account	Device-specific use	Independent of cloud services; settings/files limited to one device	Privacy-conscious users, specific single-device scenarios, custom restricted accounts	Device isolation; no cloud sync; no internet required for login
Domain Account	Network-wide management	Managed by network administrator (e.g., Active Directory); applies to networked computers	Enterprise networks, centralized user and resource management	Centralized control; complex permissions; requires network administrator oversight

2. Managing User Accounts

Managing user accounts involves a lifecycle of creation, modification, password handling, and eventual deletion or disabling. Windows offers a variety of tools, from intuitive graphical interfaces to powerful command-line utilities, to perform these tasks, providing flexibility for different administrative preferences and automation needs.

Creating New User Accounts

Windows provides several methods to create new user accounts, catering to different user preferences, system editions, and automation requirements.

Graphical User Interface (GUI) Methods:

- **Settings App:** This is generally the most common and user-friendly method for Windows 10 and 11. To access, navigate to Settings > Accounts > Other users (or Family & other users in some Windows versions). From there, select Add account or Add someone else to this PC. Users can choose to add a Microsoft account by entering an existing email address or creating a new one. Alternatively, a local account can be created by selecting I don't have this person's sign-in information, followed by Add a user without a Microsoft account. For local accounts, the process requires setting a username, password, and password hint or security questions. Family members can also be added and managed via Settings > Accounts > Family, which integrates with Microsoft Family Safety features.
- **Control Panel:** While the Settings app is increasingly the primary interface, the Control Panel remains an option for user account management, particularly for Windows 10. The path is typically Control Panel > User Accounts > Manage another account > Add a new user in PC settings. However, in newer Windows versions, this option often redirects directly to the Settings app.
- **Computer Management (Windows Pro/Enterprise Editions):** This advanced tool is available in Windows 10 Pro/Enterprise and Windows 11 Pro/Enterprise editions, offering more granular local user and group management capabilities. It can be accessed via

Windows + X > Computer Management > Local Users and Groups > Users. Within the Users pane, an administrator can right-click anywhere blank, select New User, fill in the required details (username, full name, description, password, and password options), and then click Create.

- **Netplwiz Utility:** Netplwiz is a convenient utility that provides a quick graphical interface to view and manage user accounts and their group memberships. To open it, press Win + R to open the Run dialog, type netplwiz or control userpasswords2, and press Enter. In the User Accounts window, an administrator can click Add, then follow the on-screen instructions to create either a local user or a Microsoft account.

Command Line Methods (Administrator Privileges Required):

- **Command Prompt (CMD):** The Command Prompt offers a fast and scriptable method for user creation, ideal for automation or remote management. First, it is essential to open Command Prompt as Administrator. To create a user, the command `net user <Username> <Password> /add` is used. If <Password> is left blank, an account without a password will be created. By default, `net user /add` creates a standard user account.
- **PowerShell:** PowerShell provides more robust scripting capabilities and cmdlets specifically designed for local account management. As with CMD, PowerShell must be opened as Administrator. To create a local user with a password, the command `New-LocalUser -Name "Username" -Password (ConvertTo-SecureString "Password" -AsPlainText -Force)` is used. The `-NoPassword` parameter can be used to create an account without a password. While `New-LocalUser` is primarily for local system accounts, it can also be used to create local user accounts connected to a Microsoft account by specifying the username in a specific format, for example, `MicrosoftAccount\<username>@Outlook.com`.

Modifying Account Types

Changing an account type, typically between Standard User and Administrator, is a common administrative task that always requires existing administrative privileges on the system.

Graphical User Interface (GUI) Methods:

- **Settings App:** For both Windows 10 and 11, the Settings app is the primary method. Navigate to Settings > Accounts > Other users (or Family & other users). Select the specific user account to modify, then click Change account type. From the dropdown menu, choose either Administrator or Standard User and click OK to apply the change.
- **Control Panel:** This traditional method is still available. Access the Control Panel, then go to User Accounts > Change account type. Select the user account, click Change the account type, choose the new type, and click Save.
- **Computer Management (Windows Pro/Enterprise Editions):** Open Computer Management > Local Users and Groups > Users. Right-click the specific account, select Properties, navigate to the Member Of tab, and then add or remove the user from relevant groups (e.g., the Administrators group) to change their privileges.

Command Line Methods (Administrator Privileges Required):

- **Command Prompt (CMD):** To promote a user to the Administrator group, the command `net localgroup administrators "AccountName" /add` is used. To demote an Administrator back to a Standard user, the command `net localgroup administrators "AccountName" /delete` is executed.
- **PowerShell:** To promote a user to the Administrator group, use the cmdlet `Add-LocalGroupMember -Group "Administrators" -Member "AccountName"`. To demote an Administrator back to a Standard user, the cmdlet `Remove-LocalGroupMember -Group "Administrators" -Member "AccountName"` is used.

It is generally recommended to restart the system after changing an account type to ensure all new permissions and settings take full effect.

Password Management

Effective password management for user accounts is a critical component of overall system security. This includes changing passwords regularly and knowing how to reset them if forgotten.

Changing/Resetting Your Own Password (When Already Logged In):

- **Microsoft Account:** To change a Microsoft account password, navigate to Settings > Accounts > Sign-in options > Password, then click Change. If the current password has been forgotten, the "Forgot my current password" link can be used to initiate the account verification and new password creation process, which typically involves security questions or alternative verification methods.
- **Local Account:** Similarly, for a local account, go to Settings > Accounts > Sign-in options > Password, and click Change. The current password, followed by the new password and a password hint, must be entered. If a local account password is forgotten, it can be reset using the security questions set up during account creation or a pre-created password reset disk. Without these, resetting the entire Windows device might be necessary, which results in data loss.

Resetting Another User's Password (as an Administrator):

- **Graphical User Interface (GUI):**
 - **Control Panel:** An administrator can open Control Panel > User Accounts > Manage another account. Select the target user, choose Change the password, and then enter and confirm the new password. This method is primarily effective for local accounts.
 - **Computer Management (Windows Pro/Enterprise Editions):** Access Computer Management > Local Users and Groups > Users. Right-click on the desired user, select Set Password, and then enter and confirm the new password.
- **Command Line (Administrator Privileges Required):**
 - **Command Prompt (CMD):** To be prompted to enter a new password for a user, the command `net user <Username> *` is used. The password will not be displayed as it is typed, ensuring privacy. Alternatively, to set a password directly, use `net user <Username> <NewPassword>`.
 - **PowerShell:** To change a local user's password using PowerShell, the `Set-LocalUser` cmdlet is employed. An example sequence involves securely prompting for the new password and then applying it: `$Password = Read-Host -AsSecureString; Set-LocalUser -Name "Username" -Password $Password`. This method ensures the password is handled securely.

Disabling User Accounts

Disabling a user account temporarily revokes access without deleting the account or its associated data. This is useful for temporary suspensions or when an account is not needed for a period but may be reactivated later. A disabled account disappears from the login screen and the Start menu's user switching options.

Graphical User Interface (GUI) Methods:

- **Local Users and Groups (Windows Pro/Enterprise Editions):** This is the most direct GUI method for disabling local accounts. Open the Local Users and Groups snap-in by typing `lusrmgr.msc` in the Run dialog (Win + R). Navigate to Users, right-click the desired user account, select Properties, and then check the Account is disabled box in the General tab. Unchecking the box re-enables the account. This method is not available in Windows Home editions.
- **Computer Management:** Similar to Local Users and Groups, this tool provides access to the same functionality. Right-click the Start button, select Computer Management, expand Local Users and Groups, click Users, right-click the target account, select Properties, and check Account is disabled.
- **Settings App (for Family Accounts):** For accounts part of a Microsoft family group, an administrator can disable sign-in via Settings > Accounts > Family by selecting the account and clicking Block sign in.

Command Line Methods (Administrator Privileges Required):

- **Command Prompt (CMD):** To disable a local account, open an elevated Command Prompt and use the command: `net user "AccountName" /active:no`. To re-enable it, use `/active:yes`.
- **PowerShell:** PowerShell offers specific cmdlets for this purpose. To disable a local user account, use `Disable-LocalUser -Name "AccountName"`. To enable the account, use `Enable-LocalUser -Name "AccountName"`.

Deleting User Accounts

Deleting a user account permanently removes the user's access, profile, and associated permissions from the system. It is crucial to back up any important data from the user's profile before proceeding with deletion, as this action is irreversible and results in data loss. An administrator account is required to perform this action.

Graphical User Interface (GUI) Methods:

- **Settings App:** This is a common method for Windows 10 and 11. Navigate to Settings > Accounts > Other users (or Family & other users). Under the "Other users" section, select the account to remove, then click Remove. Confirm by selecting Delete account and data. Note that this action does not delete the person's Microsoft account, only its local profile on the device.
- **Control Panel:** The Control Panel can also be used. Open Control Panel > User Accounts > Remove user accounts. Select the account to delete, then click Delete the account. You will be prompted to choose whether to keep or delete the user's files.
- **Computer Management (Windows Pro/Enterprise Editions):** For more granular control, especially in Pro or Enterprise editions, open Computer Management > Local Users and Groups > Users. Right-click the user account, and select Delete. Confirm the deletion when prompted.

Command Line Methods (Administrator Privileges Required):

- **Command Prompt (CMD):** To delete a user account, open an elevated Command Prompt and use the command: `net user <Username> /delete`. After deleting the account, the user's profile folder (e.g., `C:\Users\Username`) may still exist and can be manually removed using `rd /s /q "C:\Users\USERNAME"`.
- **PowerShell:** PowerShell provides the `Remove-LocalUser` cmdlet for deleting local user accounts. The command `Remove-LocalUser -Name "Username"` will delete the account. To ensure complete removal, including the user's profile folder, additional commands may be necessary, such as `Remove-item "C:\Users\USERNAME" -Recurse -Force`. It is also recommended to remove the corresponding registry entry under `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Profile List` to prevent temporary profiles from being created upon re-login.

3. Managing User Groups

User groups are fundamental to efficient and secure permission management in Windows environments. They provide a structured way to manage access for multiple users simultaneously, rather than assigning permissions individually.

Purpose and Benefits of User Groups

The primary purpose of user groups is to collect user accounts, computer accounts, and other groups into manageable units. This approach significantly simplifies permission management, enhances security, and streamlines administrative tasks. Instead of assigning permissions to each individual user, administrators can assign permissions to a group. Any user added to that group automatically inherits the permissions assigned to it.

This method offers several key benefits:

- **Simplified Management:** When a user's role changes, or a new user joins, administrators simply add or remove them from the relevant groups, rather than modifying permissions across numerous individual files and folders. This reduces manual tasks and the likelihood of configuration errors.
- **Enhanced Security:** Assigning permissions to groups rather than individual users helps enforce the Principle of Least Privilege (PoLP) by ensuring consistent access levels. It also prevents the accumulation of orphaned Security Identifiers (SIDs) when user accounts are deleted, which can clutter Access Control Lists (ACLs) and complicate system auditing.
- **Consistency and Scalability:** Groups ensure that all members have the same access rights to shared resources, promoting a consistent security posture across the organization. This approach scales effectively in larger environments, making it easier to manage permissions for a growing number of users and resources.

- **Delegation of Administration:** Group-based management allows for the delegation of specific administrative tasks. For example, a group can be given rights to manage certain types of accounts without having full administrative control over the entire domain.

Types of Groups (Local vs. Domain)

Windows environments utilize different types of groups based on their scope and management context:

- **Local Groups:** These groups are managed on individual Windows machines using the local Security Account Manager (SAM) database. Members of a local group can only access resources on that specific local computer. Local groups can include local users, as well as domain users and groups if the computer is part of a domain. They are useful for managing permissions on standalone computers or for specific machine-level access control within a larger network.
- **Domain Groups:** In environments integrated with Active Directory (AD), domain groups provide centralized management of permissions across multiple computers and resources within a domain or forest. AD contains two primary types of groups: Security Groups (used to assign permissions to shared resources) and Distribution Groups (used for email distribution lists). Both security and distribution groups have four different scopes that determine where their permissions can be enforced:
 - **Universal Groups:** Can include users and groups from any domain in the forest and can be used to access resources in any domain in the forest. They are typically used in multi-domain environments.
 - **Global Groups:** Can include users and other global groups only from their own domain. However, members of a global group can access resources in any trusting domain within the same forest. They are often used to categorize users based on roles (e.g., HR, Sales).
 - **Domain Local Groups:** Can include users and groups from any domain in the forest but grant access to resources only within the domain where the domain local group is defined. They are commonly used to assign permissions to specific resources like printers or file shares.
 - **Local Groups (within a domain context):** These are the same as the local groups described above, managed on individual member servers or workstations, but they can include domain users and domain groups as members. It is important to note that domain BUILTIN groups (e.g., Domain Admins, Account Operators) primarily apply to Domain Controllers by default and do not automatically confer access to all domain members unless configured via Group Policy.

Basic Elements and Hierarchical Organization

Built on numerous fundamental components working in concert to arrange and control data, the Windows file system provides a hierarchical structure that enables effective retrieval and storage.

Files: The Elementary Data Storage Units

Fundamentally, a file is a distinct collection of related data kept on a physical storage device—such as a hard drive or Solid State Drive (SSD). From basic text documents and spreadsheets to sophisticated images, high-definition videos, executable programs, and more, these incredibly flexible files can store many kinds of information. A "file object" within the Windows operating system logically represents a resource. This resource can be either a physical device itself—such as a printer or a network adapter—or, more commonly, a specific resource found on a physical device, such as a data file on a hard disk.

This abstraction allows the Windows I/O (Input/Output) system to uniformly control and communicate with many resources. The constant emphasis on "logical" groupings and "abstraction" indicates that the main function of the file system is to hide the complexity of sectors, tracks, or NAND cells from both end-users and software developers. In doing so, it creates a user-friendly and programmable interface over the raw physical storage.

By removing the requirement for every application to understand and control low-level physical features, the file system greatly simplifies software development and makes systems significantly more versatile. This abstraction is therefore absolutely essential for usability and modern software development. The pictorial representation of desktop icons in windows operating system is shown in Figure 1.15.

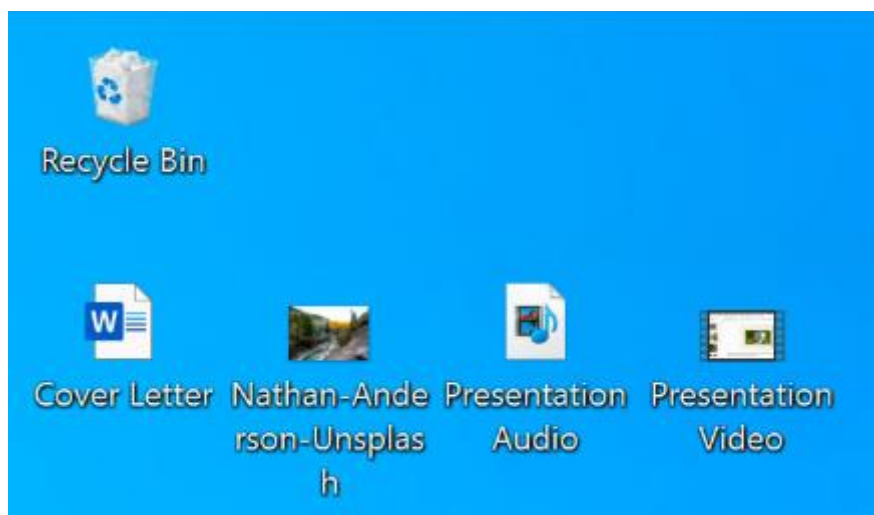


Fig 1.15. Desktop Icons in Windows Operating System

Directories (Folders): Organizing Files Hierarchically

Files are not stored haphazardly; they are systematically organized into containers known as directories, or more commonly in Windows, "folders". Directories are designed to hold both individual files and other directories, thereby creating a nested, hierarchical arrangement. This hierarchical organization is often conceptualized as a "directory tree," where the main storage location branches out into sub-branches, allowing for a logical and intuitive method of organizing and navigating vast quantities of files. This structure significantly simplifies file management for users. From a practical standpoint, the primary constraint on the number of files or subdirectories that can be contained within a single directory is the physical storage capacity of the disk on which that directory resides. This means that, theoretically, a directory can hold an immense number of items as long as there is sufficient disk space. The pictorial representation of desktop folders in windows operating system is shown in Figure 1.16.

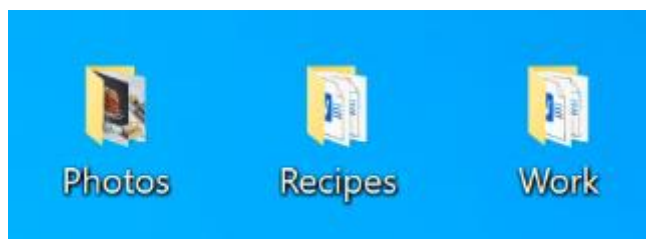


Fig 1.16. Folders in Windows Operating System

File Paths: Navigating the Directory Tree

To precisely identify the location of any file or directory within the hierarchical file system, Windows assigns a unique identifier known as a "file path." This path serves as an address, specifying the exact route from a starting point to the desired item. In Windows, file paths are typically constructed starting with a drive letter (e.g., C:), followed by a series of directory names, with each level of the hierarchy separated by a backslash (\). For example, C:\Users\Documents\MyReport.docx specifies a file named MyReport.docx located within the Documents folder, which is itself within the Users folder on the C: drive. The "root directory" of any given drive is denoted by a single backslash (\) immediately following the drive letter, such as C:\. This represents the top-level directory from which all other directories and files on that specific drive branch.

Windows supports several distinct file path formats, each designed to address different access scenarios. The pictorial representation of File Explorer Window and Quick Access View in Windows operating system as shown in Figure 1.17 and Figure 1.18 respectively.

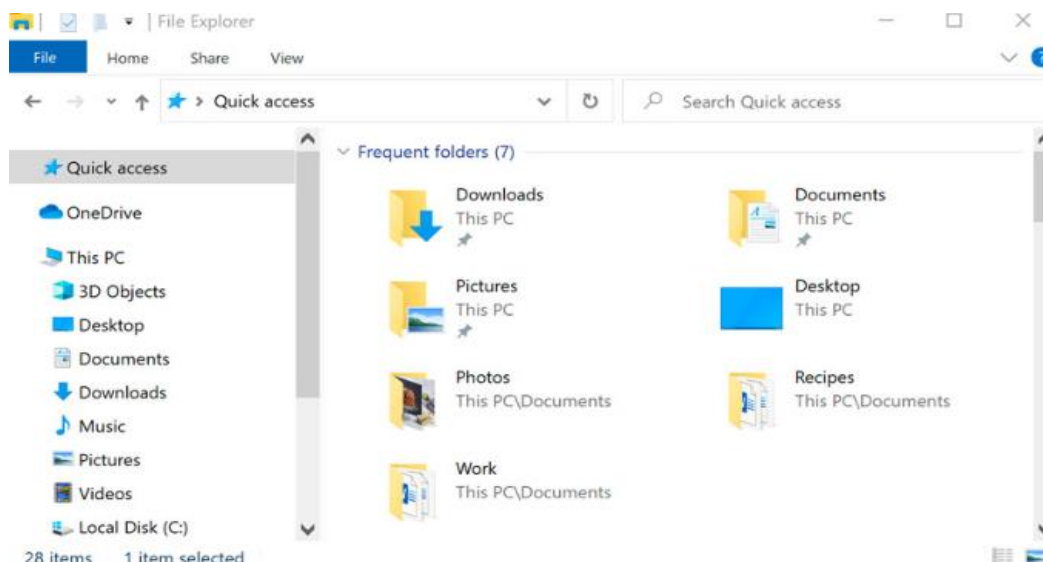


Fig 1.17. File Explorer Window in Windows OS

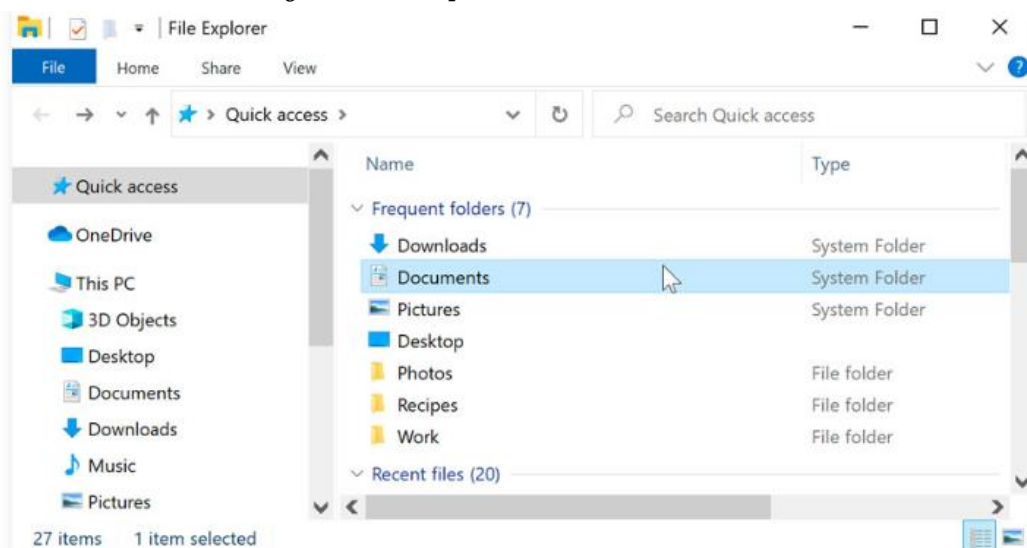


Fig 1.18. File Explorer – Quick Access View in Windows OS

Summary

- An Operating System (OS) is software that acts as an interface between computer hardware components and the user.
- The operating system is responsible for tasks such as process management, memory management, file system management, I/O management, networking, security, and providing a user interface.
- Operating systems come in various types like batch OS, multitasking, multiprocessing, multiprogramming, distributed OS, and RTOS—each used for specific needs. Understanding these types helps in choosing the right operating system for the right job, ensuring efficiency and effectiveness.
- Windows is a graphical operating system developed by Microsoft. Since 1985, Microsoft has been producing distinctive versions of the Windows operating system, providing updates and new features in every subsequent version.
- Every Linux-based OS includes the Linux kernel—which manages hardware resources—and a set of software packages that make up the rest of the operating system.

- A few important differences between Windows and Linux include: Linux is open-source and free, whereas Windows is a paid operating system. Linux has machine-friendly features and requires users to learn how to use it. Windows uses a Graphical User Interface, making it accessible to non-technical users without requiring deep technical knowledge.

Check your progress**A. Multiple Choice Questions**

1. The _____ connects applications to the actual processing of data and manages all communications between software and hardware components. It is the core of any OS.
 - a. Central Processing Unit
 - b. Main Memory
 - c. Kernel
 - d. Systematic Multi-processing
 - e. Cluster Unit
2. Which of the following options is not an operating system?
 - a. MS-DOS
 - b. BlackBerry OS
 - c. Windows
 - d. Oracle
 - e. UNIX
3. Which of these is not a type of operating system?
 - a. Network Operating System
 - b. Distributed Operating System
 - c. Embedded Operating System
 - d. Batch Operating System
 - e. All are a type of operating system
4. The first operating system was developed by _____.
 - a. IBM
 - b. Microsoft
 - c. BlackBerry
 - d. Mac
 - e. General Motors
5. A program that automates the translation of assembly language into machine language is called _____.
 - a. Assembler
 - b. Interpreter
 - c. Compiler
 - d. Processor
 - e. None of the above
6. What is the main function of the command interpreter?
 - a. To provide the interface between the API and application program.
 - b. To handle the files in the operating system.
 - c. To get and execute the next user-specified command.
 - d. None of the mentioned
7. In operating systems, which of the following is/are CPU scheduling algorithms?
 - a) Priority
 - b) Round Robin
 - c) Shortest Job First
 - d) All of the mentioned

8. To access the services of the operating system, the interface is provided by the _____
- Library
 - System calls
 - Assembly instructions
 - API
9. CPU scheduling is the basis of _____
- Multiprogramming operating systems
 - Larger memory-sized systems
 - Multiprocessor systems
 - None of the mentioned
10. Which one of the following is not true?
- The kernel remains in the memory during the entire computer session.
 - The kernel is made of various modules which cannot be loaded in a running operating system.
 - The kernel is the first part of the operating system to load into memory during booting.
 - The kernel is the program that constitutes the central core of the operating system.

B. Fill up the blanks

- _____ encompasses a wide range of activities, such as process scheduling and synchronization.
- The operating system also ensures _____ by utilizing passwords to prevent unauthorized apps from accessing programs or data.
- The purpose of the _____ is to efficiently organize and execute multiple jobs at once using group processing.
- It is possible to explain _____ in a straightforward manner by observing that these systems have more than one program in the main memory, and any one of these programs can be kept in execution.
- With _____, users can work at the same time, cutting down on waiting for their computer turn.
- _____ operating systems are designed for applications with extremely tight time requirements, where even the slightest delay is unacceptable.
- Originally published in 1985 as Windows 1.0, this operating system was created and released by _____.
- A method of interacting with a computer that relies on the input of commands into a window or terminal is called _____.
- _____ is an open-source, free, and UNIX-based OS.
- The _____ operating system is both interoperable and free of proprietary code since it is composed of multiple portions that have been created by different teams.

C. True or False

- An Operating System (OS) is software that acts as an interface between computer hardware components and the user.
- An operating system is not an example of application software; it's system software.
- An operating system is essential for running a computer. It manages hardware and software resources, providing a platform for applications to run.
- Multiprogramming doesn't use multiple CPUs; it allows multiple programs to run seemingly concurrently on a single CPU.
- An operating system provides a basic user interface, allowing users to interact with the computer.
- It is the responsibility of the operating system to ensure that all resources are adequately utilized and that no resource remains idle if there is an active request for it at a given moment.

7. Hard real-time systems are designed for applications with extremely tight time requirements, where even the slightest delay is unacceptable.
8. Linux is a free and open-source operating system, in contrast to Windows, which is commercially available and costly.
9. Compared to Windows, Linux offers superior protection against malware and viruses, fewer vulnerabilities, and built-in user privilege restrictions, making it a highly secure operating system.
10. A computer can have multiple operating systems, but only one OS can be active at a time.

D. Short Answer Questions

1. What is the primary function of an operating system?
2. Name any two main functions of the operating system.
3. How does a batch operating system handle multiple jobs?
4. What is the role of CPU scheduling in process management?
5. Why is memory management important in an operating system?
6. What is the difference between installation and configuration of an operating system?
7. Name any two types of installation methods in Windows.
8. What is the purpose of using an unattend.xml file during Windows installation?
9. What is the main difference between a quick format and a full format?
10. Which partition type is required for booting an operating system?

Answer Key**A. Multiple Choice Questions**

1.c, 2.d, 3.e, 4.e, 5.a, 6.c, 7.d, 8.b, 9.a, 10.b

B. Fill in the blanks:

1. Operating system, 2. Security, 3. Job scheduler, 4. Multiprogramming, 5. Time-sharing, 6. Real-time, 7. Microsoft, 8. Command-line interface, 9. Linux, 10. Linux

C. True or False

1. True, 2. True, 3. True, 4. True, 5. True, 6. True, 7. True, 8. True, 9. True, 10. True

Chapter-2**Operating System Structure**

Ria asked her friend Arun to explain the Operating System. Arun said, “The OS is like a manager. The Kernel makes big decisions, while helpers like the Memory Manager, File Manager, and Device Drivers keep things organized. Different structures—monolithic, layered, or microkernel—still work to keep programs running smoothly.” Ria smiled, “So it’s teamwork!” “Exactly,” Arun replied. Ria and Arun learning about operating system.

**2.1. Operating System Services**

The operating system is a piece of software that bridges the gap between the user and the hardware of a computer. We can execute many programs with the help of this software. It is the only application that is continuously running. For other programs to run successfully on any computer, an operating system is essential.

The OS ensures that all hardware and software components work together. It provides a platform for other application programs to function. A computer cannot operate correctly without an operating system, which is a collection of specialized applications. It oversees the operation of programs, file management, and input/output devices.

Services of Operating System

- Program execution
- Input Output Operations
- Communication between Process
- File Management
- Memory Management
- Process Management
- Security and Privacy
- Resource Management
- User Interface
- Networking
- Error handling
- Time Management

Program Execution

The operating system controls the execution of programs. It loads a program into memory and executes it. The execution order is determined using CPU scheduling algorithms such as FCFS (First-Come, First-Served) and SJF (Shortest Job First). The OS also handles deadlocks—situations where no two processes can proceed because each is waiting for the other. It ensures that all user and system programs run smoothly. To efficiently perform all functionalities, the OS utilizes various system resources.

Input Output Operations

The operating system coordinates the flow of data between the user's devices and system hardware. Device drivers are used to handle hardware and ensure synchronization between devices. The OS also enables programs to access I/O devices when needed.

Communication Between Processes

The OS controls how processes communicate with each other. Processes can communicate with one another by exchanging data. Even when processes aren't physically located on the same machine but rather communicate with one another over a network, the OS is responsible for coordinating this communication.

File Management

The OS facilitates file management by authorizing programs to access files with permissions such as read-only or read-write. Users can create and delete files. The OS also determines where and how data is stored—whether on a hard disk, floppy disk, or pen drive—and how it is accessed or manipulated.

Memory Management

First, let's break down the OS's memory management in a simple method. Envision a cricket squad that is severely understaffed. The forthcoming player's performance determines whether he is included in the starting lineup, the starting fifteen, or not at all, according to the team manager (OS). Similarly, before loading an upcoming program into memory, the operating system verifies that it meets all requirements for memory space. If everything checks out, the system then determines how much memory space is needed for the program. The program is thereby prevented from utilizing RAM that is not essential.

Process Management

Imagine the CPU as a kitchen stove and the OS as a chef preparing multiple dishes (programs). The chef ensures each dish gets cooking time without overuse of resources. Likewise, the OS schedules and manages multiple processes efficiently, ensuring balanced and fair CPU usage.

Security and Privacy

OS adds a security layer to our computer, keeping it safe from unwanted users. Security is merely an additional safeguard that prevents malicious actors, such as hackers and viruses, from gaining access to a computer system. Security measures provided by operating systems, such as firewalls and anti-virus software, guarantee the utmost protection for both computers and personal data.

OS provides the ability to conceal important information, like to a lock on a door that only you can open; others are not permitted to enter. In essence, it protects our privacy and gives us the means to do so.

Resource Management

All of the processes running on the system share the same system resources. The sharing of these resources is controlled by the operating system. It uses CPU scheduling algorithms to manage CPU time among tasks. Additionally, it aids in memory management and handles I/O devices. The OS also determines which resources are to be used by particular users, thereby ensuring that all available resources are utilized appropriately.

User Interface

An OS's user interface is a must-have feature. Both the command line and the graphical user interface (GUI) are used by users to interact with the operating system. The next command that the user specifies is carried out by the command interpreter.

A graphical user interface (GUI) provides a window and menu system that can be navigated with a mouse.

Networking

The ability to manage connections, send and receive data packets, and establish connections to the internet are all functions made possible by this service, which allows devices on a network to communicate with one another.

Error Handling

Any problems with the central processing unit, I/O devices, etc. are also handled by the operating system. By fixing faults and ensuring they do not occur frequently, it does more than that. Additionally, it stops the process from getting stuck. In addition, it searches for any mistakes or issues that may arise during the course of any activity. When it comes to protecting computer systems from outside threats, a well-secured operating system can occasionally double as a defense mechanism.

Time Management

Think of the traffic light as an operating system (OS), which tells all the cars (programs) whether they should be stopped (red) (in a simple queue), started (yellow) (in a ready queue), or moved (green) (under execution). The light changes at regular intervals on both sides of the road (the computer system) to ensure that the cars (programs) on both sides can move freely without any hitches.

2.2. System Calls in OS

The kernel is the central component of an OS; it manages all the hardware and the services provided by the OS. User-level applications are able to interact with low-level hardware components through system calls, which are provided by the OS.

In order for processes to communicate with the kernel and make appropriate use of the services it offers; several system calls are required. A process can communicate with the operating system through system calls. Moreover, kernel mode can only be accessed through these system calls.

Types of System Calls

Any operation that a user program is capable of performing, such as creation, termination, forking, moving, communication, etc., is usually covered by the services offered by an OS. A single system call category is used for similar sorts of tasks. There are several ways in which system calls are categorized as shown in Figure 2.1.

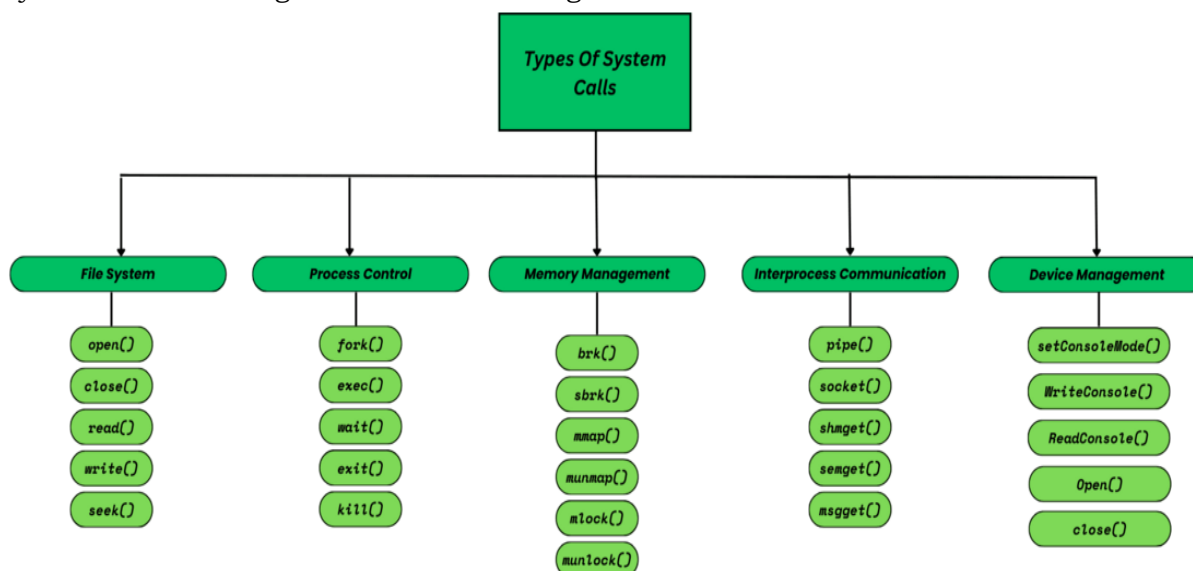


Fig 2.1. Types of System Calls

1. File System Operations

While performing operations on files within the operating system, such as creating, deleting, terminating, etc., these system calls are generated.

- `open()`: Reads from and writes to a file. Any kind of data can be stored in a file, including text, audio, etc.
- `read()`: Reads from a text file. If a process wishes to read data from a file, it will issue a `read()` system call immediately after opening the file using the `open()` system function.
- `write()`: Writes information to a file. This occurs whenever the user saves a file after making any kind of alteration to it.

- **close():** Closes a previously opened file.
- **seek():** Repositions the file pointer inside a file. When the user attempts to read data from a certain place in a file, this call is usually made.

2. Process Control

The creation, termination, allocation, and deallocation of processes are all handled by these system calls. Controls and oversees the various OS processes.

- **fork():** Duplicates the parent process to create a new one, called a child process. This call is made whenever a process creates a duplicate of itself, and it temporarily suspends the parent process until the child process completes its execution.
- **exec():** Switches processes by loading and executing a new application inside the existing one. A new process takes over and replaces all the old ones, including the data in the stack, registers, heap memory, and so on. This is called an overlay. Take the case of running a byte code program in Java with the command `java "filename"` as an example. Later on, the JVM and the java file will be executed in the background through an `exec()` call.
- **wait():** This call's principal objective is to prevent the parent process from continuing execution until all of its child processes have completed their execution. When a fork occurs in a process, this call is executed.
- **exit():** It simply terminates the current process.
- **kill():** This call sends a signal to a specific process and has various purposes, including requesting it to quit voluntarily, force quitting it, or reloading its configuration.

3. Memory Management

Allocating and deallocating process memory, as well as dynamically altering its size, are handled by these system calls. A quick rundown of what these system calls do: manage memory.

- The `"brk"` function modifies the size of a process's HEAP Memory data segments. To specify the heap's endpoint and to manually set its size, it accepts an address as an input.
- To manage heap memory, you can use the `sbrk()` function, which accepts an integer argument (positive or negative) to define the size to increase or decrease, accordingly.
- Memory Map—`mmap()` essentially maps a device or file into main memory and then into the address space of a process so that the latter can execute operations on the former. There will be a direct correlation between the revision history and the updated file content.
- To remove a memory-mapped file from the address space of a process and from main memory, you can use the `unmap()` function.
- The methods `mlock()` and `unlock()`: A "memory lock" is a method that blocks the copying of particular memory pages to the disk's swap space. Applying this is one approach to avoid page errors. Memory unlocks are the antithesis of locks; they free the captive audience from the grip of previously established information.

4. Interprocess Communication (IPC)

If more than one process needs to communicate with another, the operating system uses one of many interprocess communication (IPC) techniques, which involve several system calls. Some examples are as follows:

- **pipe():** Creates a one-way communication channel between processes using the `pipe()` function. As an example, a parent process can act as an input source for its child process by communicating with it over a pipe.
- **socket():** opens a channel for data transfer over a network. With the appropriate network permissions, processes on the same or different networks can use this socket to communicate with each other.
- **shellget():** "shared-memory-get" is abbreviated as `shellget()`. It enables interprocess communication and memory sharing among several processes.
- **semget():** Semaphore-get is abbreviated as `semget()`. It is common practice for this function to handle the coordination of numerous processes that share the critical section as a resource.

- **msgget():** This is an abbreviation for "message-get": msgget(). A message queue, a data structure inside memory that allows different processes to communicate with each other, is one of the basic principles in an IPC system. This call allots a message queue, which other processes can use to communicate data in a systematic manner.

5. Device Management

To manage the present device or communicate with other peripherals connected to the computer, you can utilize the calls provided by the device management system.

- **SetConsoleMode():** The method to change the console's mode (from input to output) is called SetConsoleMode(). A process can use it to control different console modes. When using Windows, it's for command line behavior control.
- **WriteConsole():** One such method is WriteConsole(), which lets us display data on the console.
- **ReadConsole():** If passed an argument, it reads the contents of the console screen. When you open a file or a device, this call is executed. In order to keep track of who has access to the opened file or device, a distinct file descriptor is generated.
- **close():** When the file or device is closed, either by the system or the user, this function is executed.

Importance of System Calls

- **Better Use of System Resources:** System Calls allow your computer to better use its resources. To keep programs running smoothly and efficiently without consuming excessive resources, they manage and allocate memory. The ability to multitask and general performance are both affected by this.
- **Isolation and Security:** System Calls prevent one program from accessing or interfering with another program's memory. The safety and stability of your device are both improved by this.
- **Capability to Run Multiple Programs Simultaneously:** System Calls enable multitasking. Because of this, productivity goes up, and navigating between apps is a breeze.
- **System Calls allow input and output devices like your screen, mouse, and keyboard to be communicated with, which is known as input/output (I/O) operations.** They make sure these devices function properly.
- **System Calls make it easy for programs to talk to each other and network with one another.** They facilitate online activities such as data transfer, web browsing, emailing, and connecting.

2.3 What is the purpose of System Calls in OS?

A program running in the background communicates with the operating system (OS) using system calls. Programmers who work with assembly language typically find these instructions outlined in manuals. A user-mode program will make a system call whenever it wants to access a resource. The operating system kernel receives this request and performs the necessary acquisition. A number of operations make use of system calls, including:

- Making new files or running existing ones in a file system
- File operations (reading and writing)
- Working on program processes and overseeing their execution
- Establishing links to networks, which involves transmitting and receiving data packets
- Using physical devices, such as scanners and printers.

Assignment 2.1

- a) Name the types of system calls.
- b) Write the purpose of System calls in OS.

Categories and Examples of Windows System Calls

System calls are generally grouped into several major categories based on the type of service they provide to user applications. These categories help in understanding the broad functionalities offered by the operating system.

- **Process Control:** These system calls are vital for managing the lifecycle of processes and threads within the operating system. Examples include CreateProcess (creates a new

process and its primary thread), `ExitProcess` (terminates the current process), `TerminateProcess` (forcibly terminates a specified process), `WaitForSingleObject` (allows a process to wait until a specified object, such as a process or thread, is in a signaled state), and `CreateThread` (creates a new thread within the calling process). These functions are crucial for enabling multitasking, allowing the OS to control execution, suspend or end processes, and manage system resources effectively.

- **File Management:** This category encompasses system calls necessary for performing operations on files and directories. Examples include `CreateFile` (opens or creates a file or I/O device), `ReadFile` (reads data from a specified file or device), `WriteFile` (writes data to a specified file or device), and `CloseHandle` (closes an open object handle, releasing the associated resources). These operations are fundamental for applications to access, manipulate, and persist data on disk.
- **Device Management:** These system calls facilitate interaction with various hardware resources and I/O devices. Examples include `SetConsoleMode`, `ReadConsole`, `WriteConsole` (specifically for console input/output operations). Functionalities include requesting and releasing devices, reading from or writing to device buffers, and acquiring or altering specific device attributes.
- **Information Maintenance:** This category involves system calls used for retrieving and setting various types of system-wide or process-specific information. Examples include `GetCurrentProcessID` (retrieves the process ID of the calling process), `SetTimer` (creates a timer event), and `Sleep` (suspends the execution of the current thread for a specified interval). These calls are used to get or set system-wide information like time and date, computer name, or metadata related to processes, files, or devices.
- **Inter-Process Communication (IPC) and Network Communication:** These system calls enable different processes to communicate with each other, whether on the same system or across a network. Examples include `CreatePipe`, `CreateFileMapping`, `MapViewOfFile` (for local IPC mechanisms), and a suite of socket-related functions for networking, including `socket`, `bind`, `listen`, `accept`, `send`, `recv`, `sendto`, and `recvfrom`. These are crucial for applications to interact cooperatively and to participate in network-based services.
- **Memory Management:** This category deals with the allocation and management of virtual memory for processes. Examples include `VirtualAlloc` (reserves, commits, or releases a region of memory within the virtual address space of a process), `VirtualProtect` (changes the protection on a region of pages in the virtual address space), and `VirtualQuery` (retrieves information about a range of pages in the virtual address space). These functions allow applications to dynamically request and manage memory resources as needed during execution.
- **Protection:** System calls related to protection manage access permissions and security attributes for system resources. An example is `SetFileSecurity` (used to set the security descriptor for a file or directory).

While the categories of system calls are conceptually similar across different operating systems, the specific function names, parameters, and underlying invocation mechanisms vary significantly. For instance, Windows utilizes `CreateProcess()` to initiate a new process, whereas Unix-like systems typically employ a combination of `fork()` and `exec()`. This fundamental difference highlights a significant challenge for developers aiming to write portable low-level system code. To achieve cross-OS compatibility, developers often must rely on higher-level standard libraries or employ dedicated abstraction layers that encapsulate and translate platform-specific system calls. Directly invoking Windows-specific Win32 APIs inherently ties an application tightly to the Windows environment, limiting its portability to other operating systems. Table 2.1 shows the Common Windows System Calls and Table 2.2 shows Common UNIX System Calls.

Table 2.1: Common Windows System Calls

System Call	Description
CreateProcess()	A new process is created using this command
ExitProcess()	This system call is used to exit a process.
CreateFile()	A file is created or opened using this system call.
ReadFile()	Data is read from the file using this system call.
WriteFile()	Data is written into the file using this system call.
CloseHandle()	This system call closes the file currently in use.
SetTimer()	This system call sets the alarm or the timer of a process
CreatePipe()	A pipe is created using this system call
SetFileSecurity()	This system call sets the security for a particular process
SetConsoleMode()	This sets the input mode or output mode of the console's input buffer or output screen buffer respectively.
ReadConsole()	This reads the characters from the console input buffer.
WriteConsole()	This writes the characters into the console output buffer.

Table 2.2: Common UNIX System Calls

System Call	Description
access()	This checks if a calling process has access to the required file
chdir()	The chdir command changes the current directory of the system
chmod()	The mode of a file can be changed using this command
chown()	This changes the ownership of a particular file
kill()	This system call sends kill signal to one or more processes
link()	A new file name is linked to an existing file using link system call.
open()	This opens a file for the reading or writing process
pause()	The pause call suspends a file until a particular signal occurs.
stime()	This system call sets the correct time.
times()	Gets the parent and child process times
alarm()	The alarm system call sets the alarm clock of a process
fork()	A new process is created using this command
chroot()	This changes the root directory of a file.
exit()	The exit system call is used to exit a process.

2.4. System Programs

The system program is responsible for managing the performance of the computer's application software, and there are various kinds of system programs that users can access within an operating system. System calls allow users to access the different types of system programs designed for specific tasks, which in turn enables these system programs to create and execute applications.

- **File Management:** These programs are used for managing files and directories. They can create, delete, copy, rename, print files, and terminate the program.

- **Information Status:** This includes details on input, output, storage, CPU utilization time, the amount of memory needed to complete a task, and how the process is calculated.
- **Support Tools:** The operating system utilizes programming language support tools such as compilers, assemblers, and interrupts for specific computing purposes.
- **Programming, Loading, and Execution:** After a program has been loaded, it must execute its output. System calls and system programs assist with this process.
- **Communication:** The user is responsible for utilizing these services, as the operating system relies on a network of interconnected devices that can communicate with each other through wireless means.
- **Background Services:** The operating system offers many communication services, one of which is background services. These manage tasks such as changing the desktop background and scanning for and detecting computer viruses.

Purpose of using system program

A system's hardware and software are controlled and coordinated by its system programs, which also communicate with one another. One example of system software is an operating system. A computer's operating system is like a bridge between the many applications and the hardware that runs them.

Types of System programs

The types of system programs are as follows –

Utility program

A number of computer resources are managed, maintained, and controlled by it. The use of utility applications requires a certain level of technical expertise and is typically reserved for those who possess that expertise.

Some examples of utility programs include disk tools, backup programs, and antivirus software.

Device drivers

It manages the peripherals that are connected to a computer. To put it simply, device drivers mediate communication between the operating system and any attached hardware.

Examples of drivers include those for printers, scanners, storage devices, and so on.

Tools for Reporting on Directories:

These tools are essential to an operating system in order to run software that makes navigating the computer system easier.

Examples include dir, ls, Windows Explorer, and so forth.

2.5. Structure of Operating System

A user's application programs are able to communicate with the system hardware through an operating system. The operating system is a very intricate structure; thus, it needs to be carefully designed to be easy to use and modify. Splitting the OS development process into smaller components is a straightforward approach. The functions, inputs, and outputs of each of these components should be clearly defined.

The following are various popular implementations of operating system structures.

- Simple Structure
- Monolith Structure
- Micro-Kernel Structure
- Exo-Kernel Structure
- Layered Structure
- Modular Structure
- Virtual Machines

Simple Structure

A lot of operating systems have a very straightforward architecture. Although they began as modest systems, their scope quickly expanded. Windows is a typical example of this—it was initially developed with a specific group of users in mind. That it would gain such a large following was completely unexpected. The pictorial representation of simple structure of Operating System is shown in Figure 2.2.

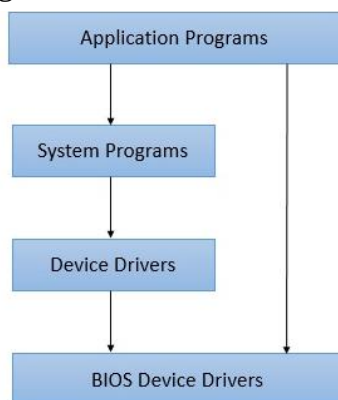


Fig 2.2. Simple Structure of Operating System

Operating systems, in contrast to MS-DOS, should have a modular structure. That would provide more control over the machine and its programs. Because of the modular design, developers could modify the external specifications as they saw fit while still hiding data and implementing internal procedures.

The Benefits

The benefits of an OS with a straightforward architecture are as follows:

- Simple operating systems allow for easier development due to the low number of interfaces required to provide even the most basic features.
- Superior Performance: Compared to other OS types, this one can deliver higher performance thanks to its minimal layers and direct interaction with hardware.

Negative aspects

The following are some drawbacks of an OS with a straightforward architecture.

- Because of its shoddy design, such a system is prone to frequent breakdowns. The entire operating system will crash if even a single program fails. This means that basic operating systems are prone to system breakdowns.
- Changes made to one layer of an operating system might have far-reaching effects on other layers, rendering the code unmanageable over time due to poor maintainability.

Monolith Structure

In an OS with a monolithic structure, the kernel is the fundamental chunk of code that controls the OS's main functions. File, memory, and device management are all examples of such tasks. Application and system programs rely on the kernel—the central component of an OS—for all of the OS's services. Interfacing between application programs and the underlying hardware, the kernel has access to all system resources. The use of a monolithic kernel structure in older banking systems encouraged timesharing and a multiprogramming paradigm. The monolithic structure is shown in Figure 2.3.

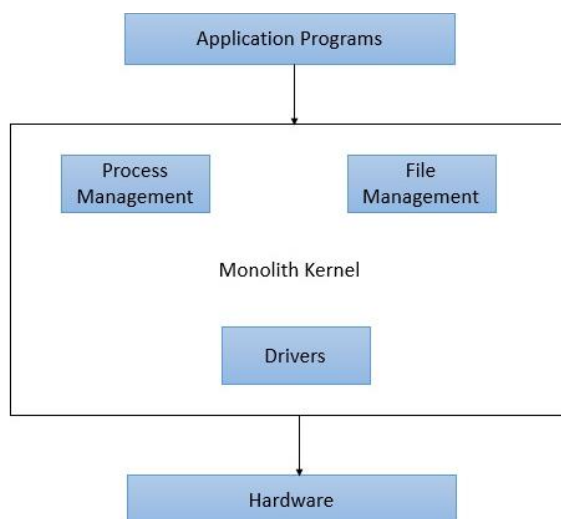


Fig 2.3. Monolithic Structure

The Benefits

Among the many benefits of a unified OS architecture are the following:

- **Developer-Friendly Processes:** The kernel is the only layer that can be built with all the main features, making it developer-friendly.
- **Performance:** The kernel is more efficient since it manages memory, handles various tasks, and has direct access to the hardware.

Negative aspects

Some drawbacks of an OS with a single-layered structure are as follows.

- Due to the fact that the kernel is in charge of all operations, the entire operating system can crash if even a single function fails.
- Difficult to improve— Adding a new service to a monolith OS is a huge pain because it affects all of the existing services.

Micro-Kernel Structure

In a microkernel architecture, rather than using a single large kernel as in a monolithic design, the system uses a minimal kernel that handles only essential functions such as communication between processes and basic I/O operations. Other services, such as device drivers, file systems, and network protocols, run in user space as separate processes. The system becomes more stable because these components are isolated—if one fails, it does not necessarily bring down the entire system, allowing the OS to continue running. The Micro-kernel Architecture is shown in Figure 2.4.

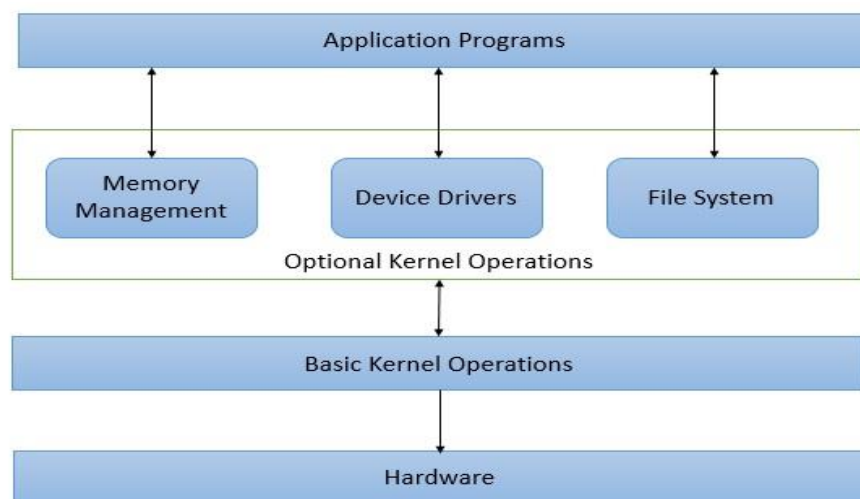


Fig 2.4. Micro-kernel Architecture

Advantages

The following are advantages of a microkernel operating system structure:

- **Reliability and Stability:** Because the kernel is minimal and most services run in user space, the operating system has a lower probability of total failure. If one service fails, others can continue functioning independently, improving overall system stability.
- **Manageable Code Size:** Since the kernel is small, its codebase is easier to maintain. Improving the microkernel does not require changing all system components.

Disadvantages

Following are disadvantages of a microkernel operating system structure.

- It is challenging to build such an architecture that relies on microkernel technology.
- Compared to a monolithic design, multi-kernel, multi-modular communication may reduce performance.

Exo-Kernel Structure

Researchers and developers at MIT created the Exo-Kernel Structured OS. Minimizing the kernel size while empowering application programs to directly manage hardware resources was the goal of this architecture. Application programmers can now create high-performance code while the exo-kernel takes care of other activities, thanks to the removal of operating system abstraction for hardware resources.

The Benefits

Using an exo-kernel operating system architecture has the following benefits:

- A well-designed program can make the most efficient use of the memory it is allotted, leading to high performance.
- Application Control — Since the operating system does not restrict resource management, application programs have more access to system resources and can create their own operations on those resources.

Negative aspects

Disadvantages of an exo-kernel OS architecture include the following:

- Poorly written code can compromise the system because security is managed at the application program level.
- Unreliable and unsafe.
- Exo-kernel design is intricate and requires significant consideration.

Layered Structure

The use of layers is one strategy for making an OS more modular. The hardware is at the bottom, and the user interface is at the top, of this stack.

An image demonstrating the layered approach is as shown in Figure 2.5.

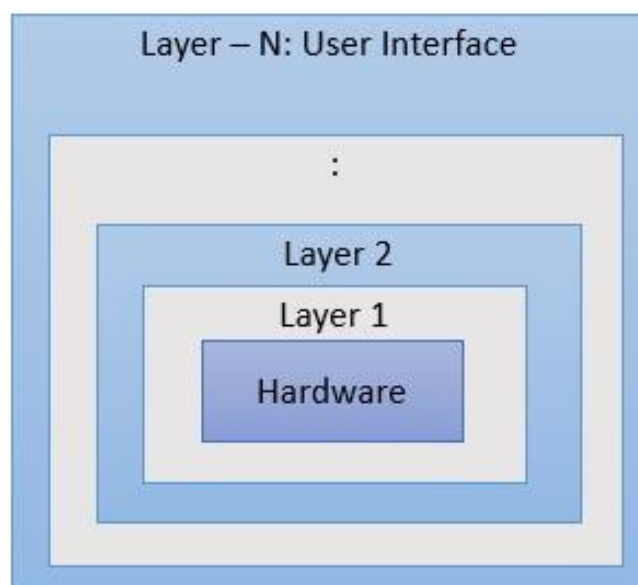


Fig 2.5. Layered Structure

Each successive layer is constructed upon the one below it, as can be seen in the image. Every layer conceals some structures, actions, etc. from the layers above it.

The necessity for meticulous layer definitions is one drawback of the layered structure. The reason this is essential is that higher-level layers can only make use of the features provided by lower-level layers.

Advantages

Following are advantages of a layered operating system structure.

- Very customizable—the fact that it is layered makes it easy to alter the implementation of each layer. The addition of new functionality can be done independently of other modules.
- Verifiable — The modular design makes it easy to verify and debug each layer.

Disadvantages

Following are disadvantages of a layered operating system structure.

- A layered-structured OS is less efficient than a basic-structured OS in terms of performance.
- The creation of a layered operating system necessitates a good design approach and meticulous planning at each layer because each layer communicates with just the lower layers.

Modular Structure

Operating systems with a modular structure are comparable to monoliths in essence but have superior design. All of an operating system's main functions are handled by a central kernel. The kernel dynamically loads additional services as modules either during boot time or during runtime, in addition to its basic functions. Among the many examples of modular organized OSes, Sun Solaris OS stands out.

Advantages: Following are advantages of a modular operating system structure.

- **Highly Modifiable:** The system's modular design allows for easy customization of each module's implementation. The addition of new functionality can be done independently of other modules.
- **Verifiable:** The modular design makes it easy to verify and debug each layer.

Disadvantages: Following are disadvantages of a modular operating system structure:

- **Less Performant:** A modular structured operating system is less performant compared to a basic structured operating system.
- **Complex designing:** Each module must be carefully planned, as each module communicates with the kernel. A communication API must be devised to facilitate this communication.

Virtual Machine Structure

Virtual machines are used to abstract the underlying hardware, such as the central processing unit, RAM, and hard disks, in this type of setup. Through the use of execution contexts, users can configure them. Disk space is required for the provisioning of virtual machines, which can be quite substantial. On a single physical computer, you can run numerous virtual machines.

Advantages: Following are advantages of a virtual machine-based operating system structure:

- **High Customizable:** Being virtual, functionality is easily accessible and can be customized on a need basis.
- **Secure:** As there is no direct hardware access, such systems are highly secure.

Disadvantages: Following are disadvantages of a virtual machine-based operating system structure:

- **Less Performant:** A virtual machine-based operating system is less performant compared to a modular structured operating system.
- **Complex designing:** Each virtual component of the machine must be carefully planned, as each component is responsible for abstracting the underlying hardware.

Assignment 2.2

- a) Explain Micro-Kernel Structure and Exo-Kernel Structure
- b) Explain Virtual Machine Structure

2.6. Comparison of UNIX and Windows

Because of this, understanding how UNIX and Windows differ from one another is crucial for our professional, academic, and everyday lives. UNIX is the operating system of choice in highly controlled, stable situations due to its exceptional performance and adaptability. However, Windows remains a favorite for most end and business applications because of its point-and-click graphical shell and improved hardware compatibility.

What is UNIX?

UNIX is an operating system that supports multiple users and a wide variety of tasks. It was developed in the late 1960s at Bell Laboratories, the research and development subsidiary of AT&T. Originally, it was intended for use in servers, workstations, and academic systems, with the goals of being robust, reliable, and flexible.

What is Windows?

Windows, created and sold by Microsoft, is among the fastest operating systems available. The first version of Windows was launched in 1985. Windows is more user-friendly for those who don't have much experience with computers because it is based on a graphical user interface.

Difference Between UNIX and Windows Operating System

Here we will discuss the differences between UNIX and Windows operating systems, based on various parameters.

- 1. Licensing:** UNIX was constructed using the C and Assembly languages as an open-source operating system. Linux, which is based on UNIX, is the most popular operating system in the world because it is open-source and has numerous distributions. The GPL makes Linux distributions freely available. Since Microsoft owns the source code of the Windows operating system, no one else can access it.
- 2. Graphical User Interfaces (GUIs):** Unix-based systems traditionally rely on command-line interfaces, making them harder for beginners. However, modern Linux distributions now include graphical user interfaces (GUIs) to improve usability. In contrast, Windows was designed with a user-friendly GUI from the beginning, making it easier to navigate for non-technical users.
- 3. Multiprocessing:** This allows for the execution of multiple processes simultaneously, is supported by the UNIX operating system. To increase processing power, multiprocessing allows for the addition of central processing units (CPUs), and each process has its own address space. Windows allows for multithreading, which increases processing power by spawning several threads from a single process. In multithreading, a process's numerous threads run in parallel, sharing a single address space.
- 4. File System:** The UNIX operating system utilizes the Unix File System (UFS), which treats all physical drives as one logical drive, and the STD.ERR and STD.IO file systems. The file system is strong and works well. Every level of the file system's structure is based on a tree with the same root. The File system structure is shown in Figure 2.6.

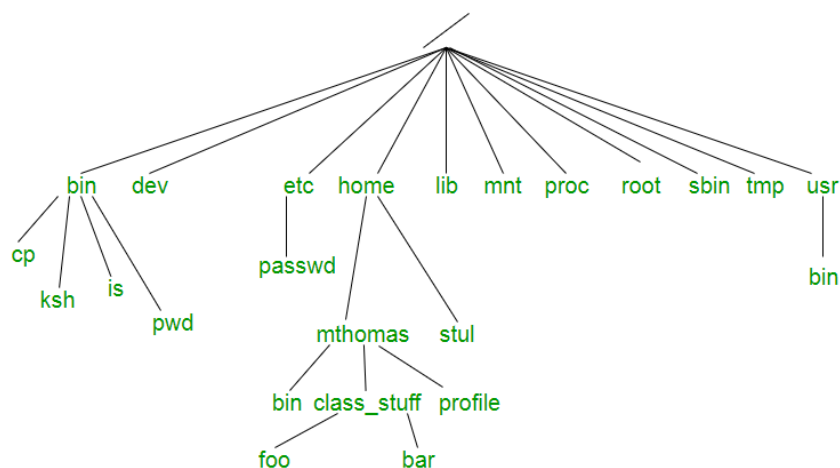


Fig 2.6. Structure of File System

Alternatively, Windows organizes files using the File Allocation Table (FAT32) and the New Technology File System (NTFS). Before executables can be run, the owner of the file must be prompted. Windows also employs folders on separate storage drives, such as C, D, and E, to hold files.

5. Users Accounts: In the Linux system, there are three types of user accounts:

- **Regular:** These accounts have moderate privileges intended for routine work. They can perform only those tasks and access only those files and services for which they are authorized.
- **Root:** This account is the main user account on a Linux system. It has the highest privileges and is used to perform administrative-level tasks and manage all services. It is also known as the superuser.
- **Service:** This account is generally used by services to run processes.

In Windows, there are three types of user accounts:

- **Administrator:** This account controls all the services and allows the user to make changes and monitor the activities of other users.
- **Child:** This account comes with Family Safety settings and is assigned by the administrator.
- **Guest:** This account is not a standard account and offers no privacy; it allows other people to use your computer without being able to change PC settings or install applications.

6. File Name Convention: In Linux, we can save two files with the same name. The Figure 2.7 shows a Linux file manager window where two files named Sample and SAMPLE exist in the same directory, demonstrating Linux's case-sensitive file handling.

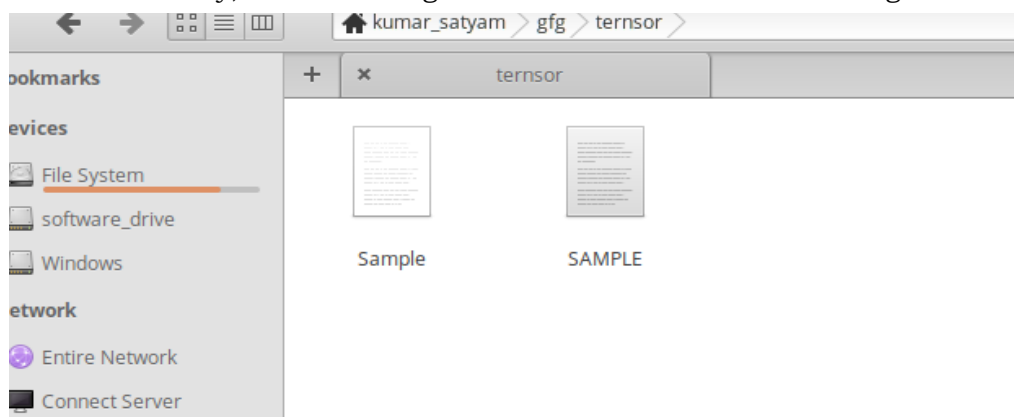


Fig 2.7. Case-Sensitive File System in Linux

But in windows, we cannot save with the same name. It generates errors. The Figure 2.8 shows that Windows does not allow multiple files with the same name differing only in case (e.g., Sample and SAMPLE).

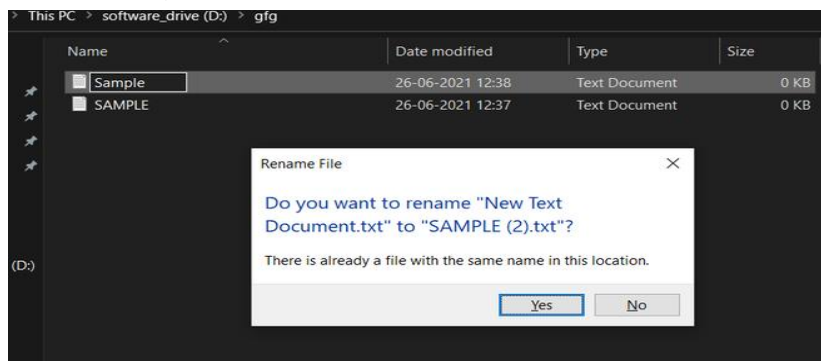


Fig 2.8. Case-Insensitive File System in Windows

7. Security: The design of UNIX makes it more secure. One of the main security features of a UNIX system is the significantly higher difficulty in running a “.exe” file. Because UNIX treats all processes the same, executables cannot be run without special authorization. However, this is beginning to change as UNIX distributions increasingly prioritize graphical user interfaces and application security.

The fact that viruses can only infect user accounts and not the root account makes UNIX systems ideal for virus prevention and removal. Windows lacks such security measures, as it allows users with minimal access to easily execute files—potentially installing malware on the machine without the user’s knowledge.

8. Data Backup & Recovery: When the device storing the data became inaccessible, it was not possible to salvage the data from the system in older versions of UNIX. However, data recovery from an inaccessible device is now possible with the help of a new feature in certain Linux distributions, such as Ubuntu, known as Live Distribution.

With Windows, creating backups and recovering lost data is a breeze, thanks to the operating system's built-in GUI. Data backup and recovery are also made easier through its integration with OneDrive.

9. Unix Case Sensitivity: Unix is case-sensitive, two files with the names "GFG.txt" and "gfg.txt" are treated as distinct files. In contrast, Windows offers the option to enable case sensitivity.

10. Hardware: UNIX systems have limited hardware support. It is possible that drivers are not developed for all devices. The vast majority of hardware drivers are pre-installed on Windows.

11. Reliability: Unix has a reputation for being extremely stable. Windows has improved significantly in terms of stability in recent years, but most Unix systems are still far more stable.

Table 2.3 shows the comparison of various parameters of Unix and Windows.

Parameter	UNIX	Windows
Basic	It is a command-based operating system.	It is a menu-based operating system.
Licensing	It is an open-source system which can be used under General Public License.	It is a proprietary software owned by Microsoft.
User Interface	It has a text-based interface, making it harder to grasp for newcomers.	It has a Graphical User Interface, making it simpler to use.
Processing	It supports multiprocessing.	It supports multithreading.

Parameter	UNIX	Windows
File System	It uses Unix File System (UFS) that comprises STD.ERR and STD.IO systems.	It uses File Allocation System (FAT32) and New Technology File System (NTFS).
Security	It is more secure as all changes to the system require explicit user permission.	It is less secure compared to UNIX.
Data Backup & Recovery	It is tedious to create a backup and recovery system in UNIX, but it is improving with new distributions.	It has an integrated backup and recovery system that makes it simpler to use.
Hardware	Hardware support is limited in UNIX system. Some hardware might not have drivers built for them.	Drivers are available for almost all the hardware.
Reliability	UNIX and its distributions are well known for being very stable to run.	Windows has improved, but still lacks the stability of UNIX systems.
Case Sensitivity	It is fully case-sensitive, and files can be considered separate files.	It has case sensitivity as an option.

Practical Activity 2.1 Demonstration of various UNIX system calls for: Process Management.

Materials required: A computer system with UNIX installed.

You can use the following UNIX process management system calls :

- A new process can be started using the fork() function.
- Use exec() to launch a new program.
- Use wait() to make the process wait.
- You can use the exit() function to end the process.
- Use getpid() to get the distinct process id.
- Use getppid() to get the parent process id.
- The nice() method is utilized to bias the presently executing process's attribute.

The use of all these calls is shown in Practical 2.2.

Practical Activity 2.2: Write simple shell scripts using system calls like fork(), exec(), wait(), file handling.

Materials required: A computer system with UNIX installed.

fork() and exit()

Only by utilizing the fork() system function can a process be created in classic Unix. A duplicate of the current program is passed on to the new process, along with its unique identifier (PID). In order to construct a process tree, the process ID of the parent process—the one that invoked fork()—is recorded as the PPID of the newly created process. The new process's PID is returned by the fork() function in the parent process. The fork() system function also returns in the new process (because the copy was created at that time), but this time the result is 0. The system function fork() is thus unique. The method returns twice, even though you only call it once: once in the parent process and once in the child process. The fork() function adds one process to the system's total count. At the very beginning of its lifecycle, every Unix process copies the code of its parent process and returns a zero from the fork() system call. Because the parent and child versions of the program get different results from the fork() system call, which causes execution to follow different if() branches, their destinies can diverge inside the code.

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

main(void) {
    pid_t pid = 0;

    pid = fork();
    if (pid == 0) {
        printf("I am the child.\n");
    }
    if (pid > 0) {
        printf("I am the parent, the child is %d.\n", pid);
    }
    if (pid < 0) {
        perror("In fork()");
    }

    exit(0);
}
```

Running this, we get:

```
kris@linux:/tmp/kris> make probe1
cc    probe1.c  -o probe1
kris@linux:/tmp/kris> ./probe1
I am the child.
I am the parent, the child is 16959.
```

Here, we define the type `pid_t` for the variable `pid`. This variable stores the output of the `fork()` function and is used to activate the "I am the child." or "I am the parent." branches of an if statement. After the program is run, two lines of output are displayed. An instance of the program can only be in one of the two branches of the code, since there is only one variable, and this variable can only have one state. Two lines of output indicate that the program was executed twice, each with a different pid value.

As an experiment, you can update the code to display two distinct pids to demonstrate this by calling `getpid()` and printing the result. The number of processes in the system is increased by one with the `fork()` system call, which is entered once but returns twice. There are still multiple processes in the system after our program completes. To reduce the total number of active processes, another system call is required. `exit()` is the name of this system call. The system call `exit()` is one-time-use only. As a result, the system's process count drops by one.

The parent process can receive (or perhaps must receive) an exit status as a parameter from `exit()`, which tells the parent what happened to the child. We are using both the parent and child processes to call `exit()`, as all of the program variations in our example do. So, two processes are terminated. The parent process is also a child process, and our shell is the parent of all children, so this is our only option. Our actions are mimicked by the shell precisely:

wait()

An `exit(0)` is the last step in our child procedure. We are able to pass 0 as the exit status of our program. We must implement a new system call to ensure that the parent process recognizes this value. You are calling the `wait()` method on the system. Inside the code:

```

#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

#include <sys/types.h>
#include <sys/wait.h>

main(void) {
    pid_t pid = 0;
    int status;

    pid = fork();
    if (pid == 0) {
        printf("I am the child.\n");
        sleep(10);
        printf("I am the child, 10 seconds later.\n");
    }
    if (pid > 0) {
        printf("I am the parent, the child is %d.\n", pid);
        pid = wait(&status);
        printf("End of process %d: ", pid);
        if (WIFEXITED(status)) {
            printf("The process ended with exit(%d).\n", WEXITSTATUS(status));
        }
        if (WIFSIGNALED(status)) {
            printf("The process ended with kill -%d.\n", WTERMSIG(status));
        }
    }
    if (pid < 0) {
        perror("In fork()");
    }

    exit(0);
}

```

After running this we get the following output:

```

kris@linux:/tmp/kris> make probe2
cc    probe2.c  -o probe2
kris@linux:/tmp/kris> ./probe2
I am the child.
I am the parent, the child is 17399.
I am the child, 10 seconds later.
End of process 17399: The process ended with exit(0).

```

As a reference parameter, the system call `wait()` will overwrite the variable `status`. In addition to the `exit` status, the value is a bitfield that provides further information on why the program terminated. C has a variety of macros with predicates like `WIFEXITED()` and `WIFSIGNALED()` that can decode this. We also receive extractors like `WTERMSIG()` and `WEXITSTATUS()`. As an additional result of the method, `wait()` also returns the process ID of the process that terminated. The parent process will not continue running until a signal is received or a child process exits, whichever comes first. You can limit the wait time by setting up a `SIGALRM` to notify you of an impending event.

exec()

`exec()` loads programs into processes that already exist, in contrast to `fork()`, which creates new processes. Inside the code:

```

#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

#include <sys/types.h>
#include <sys/wait.h>

main(void) {
    pid_t pid = 0;
    int status;

    pid = fork();
    if (pid == 0) {
        printf("I am the child.\n");
        execl("/bin/ls", "ls", "-l", "/tmp/kris", (char *) 0);
        perror("In exec(): ");
    }
    if (pid > 0) {
        printf("I am the parent, and the child is %d.\n", pid);
        pid = wait(&status);
        printf("End of process %d: ", pid);
        if (WIFEXITED(status)) {
            printf("The process ended with exit(%d).\n", WEXITSTATUS(status));
        }
        if (WIFSIGNALED(status)) {
            printf("The process ended with kill -%d.\n", WTERMSIG(status));
        }
    }
    if (pid < 0) {
        perror("In fork():");
    }

    exit(0);
}

```

Running this:

```

kris@linux:/tmp/kris> make probe3
cc      probe3.c  -o probe3

kris@linux:/tmp/kris> ./probe3
I am the child.
I am the parent, the child is 17690.
total 36
-rwxr-xr-x 1 kris users 6984 2007-01-05 13:29 probe1
-rw-r--r-- 1 kris users  303 2007-01-05 13:36 probe1.c
-rwxr-xr-x 1 kris users 7489 2007-01-05 13:37 probe2
-rw-r--r-- 1 kris users  719 2007-01-05 13:40 probe2.c
-rwxr-xr-x 1 kris users 7513 2007-01-05 13:42 probe3
-rw-r--r-- 1 kris users  728 2007-01-05 13:42 probe3.c
End of process 17690: The process ended with exit(0).

```

In this case, the child process discards the code of probe3 (the `perror("In exec():")` is not invoked). In its place, the supplied call to `ls` supersedes the currently running program. The protocol shows that while probe3 is waiting for the `exit()`, its parent instance is also waiting. It is not an `exit()` in our code because the `perror()` that follows `execl()` is never executed. In reality, the exit status that we get in our parent process's `wait()` call comes from `ls`, which terminates the process we started with an `exit()`.

Summary

1. Major services performed by an operating system include program execution, input/output operations, communication between processes, file management, memory management, process management, and ensuring security and privacy.
2. The kernel is the central component of an OS; it manages all the hardware and the services provided by the OS.
3. User-level applications are able to interact with low-level hardware components through system calls, which are provided by the OS.
4. If more than one process has to communicate with each other, the operating system uses one of many Inter-Process Communication (IPC) techniques.
5. A system's hardware and software are controlled and coordinated by its system programs, which also communicate with one another. One example of system software is an operating system.
6. In an OS with a monolithic structure, the kernel is the fundamental chunk of code that controls the OS's main functions. File, memory, and device management are all examples of such tasks.
7. In a microkernel architecture, rather than a single kernel as in a monolithic one, several modules or services optimized for specific tasks are used.
8. Virtual machines are used to abstract the underlying hardware—such as the central processing unit, RAM, and hard disks—in this type of setup.

Check your progress**A. Multiple Choice Questions**

1. In ____ mode, the kernel runs on behalf of the user.
 - a) user
 - b) kernel
 - c) real
 - d) all
2. All UNIX and LINUX systems have one thing in common, which is ____
 - a) set of system calls
 - b) set of commands
 - c) set of instructions
 - d) set of text editors
3. The chmod command invokes the ____ system call.
 - a) chmod
 - b) ch
 - c) read
 - d) change
4. If exec is called immediately after forking, ____
 - a) the program specified in the parameter to exec will replace the entire process
 - b) all the threads will be duplicated
 - c) all the threads may be duplicated
 - d) none of the mentioned
5. Which system call is used for opening a file?
 - a) read
 - b) write
 - c) open
 - d) close
6. Which system call creates a new process (child process)?
 - a) fork
 - b) exec
 - c) exit
 - d) wait

7. What happens when a process calls fork()?
 - a) The current process is terminated.
 - b) A new process is created that is a copy of the current process.
 - c) A new process is created with a completely fresh memory space.
 - d) The current process is suspended and waits for another process to complete.
8. What is the primary purpose of system calls?
 - a) To provide a user-friendly interface for interacting with the operating system.
 - b) To manage hardware resources efficiently.
 - c) To provide a programming interface for accessing kernel services.
 - d) To allow users to run programs more quickly.
9. How does a program invoke a system call?
 - a) By calling a user-level function that is provided by the operating system.
 - b) By sending a hardware interrupt to the kernel.
 - c) By using a special instruction that switches the CPU to kernel mode.
 - d) By using a specific library of functions provided by the system.
10. What is a system call?
 - a) A routine built into the kernel that performs a basic function.
 - b) A user-level function that provides access to operating system services.
 - c) A hardware interrupt that signals the kernel for attention.
 - d) A command-line interface used to interact with the operating system.

B. Fill up the blanks

1. It is the responsibility of the _____ to authorize programs to access files.
2. The _____ is the central component of an OS; it manages all the hardware and the services provided by the OS.
3. Any operation that a user program is capable of performing, such as creation, termination, forking, moving, communication, etc., is usually covered by the _____ offered by an OS.
4. _____ repositions the file pointer inside a file. When the user attempts to read data from a certain place in a file, this call is usually made.
5. _____ call is made whenever a process creates a duplicate of itself.
6. _____ sends a signal to a specific process and has various purposes, including requesting it to quit voluntarily, force quit, or reload configuration.
7. If more than one process has to talk to each other, the operating system will use one of many _____ techniques.
8. To communicate with other peripherals connected to the computer, you can utilize the calls provided by the _____.
9. The _____ is in charge of the computer's application software performance, and there are various kinds of system programs that users can access within an operating system.
10. _____ allow users to access the many sorts of system programs that are defined for certain tasks, which in turn allows the system programs to design and execute programs.

C. True or False

1. A system call is a routine built into the kernel and performs a basic function.
2. System calls are the interface between user programs and the kernel: A user program initiates a system call to request a specific OS service.
3. All system calls are directly accessible to application developers.
4. System calls can be invoked from library functions: A program can use library functions that, in turn, make system calls.
5. System calls are inherently faster than library function calls.
6. System calls handle process control, file management, and other OS services: System calls allow programs to create, terminate, and manage processes, as well as interact with files, devices, and other OS resources.
7. All programs written by a programmer are application software.

8. The fork() system call creates a new process: This is a fundamental system call used in multi-process environments.
9. System calls are essential for program execution: They provide the foundation for programs to interact with the underlying OS and access hardware and resources.
10. System software includes device drivers, which are programs that allow the computer to communicate with hardware devices.

D. Short Answers Questions

1. Name any four services provided by an operating system.
2. What is the purpose of the fork() system call in process management?
3. Which system call is used to create a file in an operating system?
4. Define the role of device drivers in system programs.
5. What is the main function of the kernel in a monolithic OS structure?
6. What is a system call and why is it used in an operating system?
7. Name two system calls used for file operations in Windows OS.
8. Which system call is used in Windows to create a new process?
9. What is the role of the VirtualAlloc system call in memory management?
10. Give one advantage of organizing system calls into categories.

Answer Key**A. Multiple Choice Questions**

1.b, 2.a, 3.a, 4.a, 5.c, 6.a, 7.b, 8.c, 9.c, 10.a

B. Fill in the blanks:

1. Operating system, 2. Kernel, 3. System calls, 4. lseek, 5. fork, 6. kill, 7. Inter-process communication (IPC), 8. Device drivers, 9. Operating system, 10. System calls

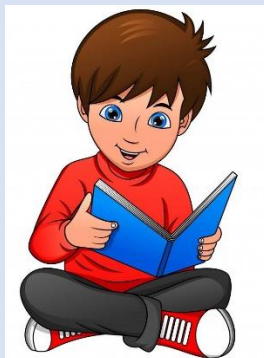
C. True or False

1. True, 2. True, 3. False, 4. True, 5. False, 6. True, 7. False, 8. True, 9. True, 10. True

Chapter-3

Explore Windows and Linux OS Environment

Rohan was an inquisitive boy and was always excited to learn new things. He lived in Faridabad. One day he thought to explore the details of Linux and Windows operating systems. He found a book in library about Linux and Windows OS. He studied the book and solved various problems on these topics. He mastered the concepts and everyone appreciated him. His learnings encouraged his classmates to learn about concepts on Linux and Windows.



3.1. Overview of Linux and Windows OS (GUI vs. CLI)

Windows and Linux are two of the most widely used OSes. Windows, developed by Microsoft, is a proprietary OS well-known for its graphical user interface and large program ecosystem. Contrarily, Linux, developed by Linus Torvalds, is an open-source OS similar to Unix that offers a strong command-line interface, security, and a lot of freedom.

CLI (Command Line Interface)

It is a textual representation of the Command Line Interface. The command-line interface (CLI) allows users to interact with an OS by entering detailed commands in a terminal or console window. Commonly used in computer systems, command-line interfaces (CLIs) enable users to type in a command or a series of commands in response to an on-screen prompt, which the system then uses to carry out the desired action. When precision of input is of the utmost importance, the CLI excels in high-cost computing environments.

GUI: (Graphical User Interface)

Graphical User Interface is abbreviated as GUI. Graphical user interfaces (GUIs) allow users to interact with operating systems visually. The graphical user interface provides access to various menus, windows, scrollbars, buttons, wizards, icons, drawing tools, and more. Because of its simplicity, it reduces the cognitive load. Many different types of information—such as plain text, images, videos, and more—are displayed to the user using a graphical user interface (GUI).

The difference between Command-Line Interface (CLI) and Graphical user interfaces (GUIs) is shown in Table 3.1.

Difference between GUI and CLI

Table 3.1. Difference between Command-Line Interface (CLI) and Graphical user interfaces (GUIs)

S.No	CLI (Command Line Interface)	GUI (Graphical User Interface)
1	CLI is challenging to use.	However, it is simple to use.
2	It uses little memory.	While using up more memory.
3	We can achieve high precision in CLI.	Low precision is achieved in it.
4	The CLI is quicker than the GUI.	Compared to CLI, GUI is slower.

5	All you need for the CLI operating system is a keyboard.	GUI operating system requires a keyboard and mouse.
6	The appearance of CLI cannot be altered.	It is possible to alter or change its appearance.
7	Only at a command prompt can input be entered in CLI.	Anywhere on the screen can be used to enter input.
8	Information is shown in files and plain text.	Information is shown in various formats—text, videos, photos, etc.
9	There are no menus available in CLI.	There are menus available.
10	CLI doesn't have any graphics.	Graphics are employed in GUIs.
11	No pointing devices are used.	Items are selected using pointing devices.
12	Typing and spelling errors are not eliminated.	Typing and spelling errors are prevented.

3.2. The LINUX operating system overview

An OS similar to Unix, Linux is based on the Linux kernel and is an operating system in its own right. The Linux kernel is the core of the operating system, controlling how the machine communicates with its components. It ensures that everything functions as intended. However, the Linux kernel alone is insufficient to build a complete OS. To create a fully functional system, Linux distributions combine the Linux kernel with a variety of software packages and tools.

These distributions configure the Linux operating system, enabling users to safely and efficiently use their applications and complete tasks on their computers. Many different Linux distributions are available, each designed to meet the specific needs and preferences of its users.

The History of Linux

The Linux kernel, created by Linus Torvalds in 1991, is a free and open-source operating system. Torvalds set out to create a flexible and free system for PCs, drawing inspiration from the UNIX and MINIX operating systems. Engineers and enthusiasts from all over the world responded to the Linux kernel's first release, fostering collaboration in its development process. By combining the Linux kernel with various open-source software packages, fully functional operating systems—often called Linux distributions—were produced.

Linux has grown to become a foundational component of modern computing, powering everything from servers and personal computers to supercomputers and mobile devices. Its adaptability, reliability, and strong community support make it a popular choice among developers, corporations, and educational institutions.

Assignment 3.1

1. List down the major differences between GUI and CLI.

Linux: What is it?

Linux is an open-source, free, and robust collection of operating systems. Anyone may examine and modify the system's source code, which is beneficial because it makes it easier to understand how the system works. Linux encourages global collaboration for its continuous development because it is open-source.

Operating systems like Linux have come a long way since their inception; they are now secure, versatile, and used on everything from desktop computers and smartphones to massive supercomputers. Efficiency, which allows for the speedy completion of various tasks, and cost-effectiveness, which indicates reduced expense in its usage, are commonly praised.

3.3. The Linux Filesystem:

All operating systems built on the Linux platform have the Linux filesystem as their backbone. From system performance to security and deployment processes, it controls file storage, organization, and access.

Objectives and Key Ideas

Fundamentally, the Linux filesystem offers a methodical way to arrange files and data on a storage medium. In contrast to Windows and other operating systems, Linux's "everything is a file" attitude is a distinguishing feature. With this method, directories, hardware devices, and even running processes are treated as files, which improves system flexibility and simplifies interactions.

Reduced complexity for users and applications alike is a direct result of this unified approach to resources, which presents various system components through a common "file" interface. A standardized set of file operations—such as opening, reading, writing, and closing—can be implemented globally instead of requiring separate APIs or interaction models for storage, networks, and hardware.

By standardizing permissions and access restrictions across all "file" types, this abstraction improves consistency and security while simplifying system design and development. The administrative burden is reduced since controlling a printer or hard drive is conceptually equivalent to controlling a text file.

Starting with a single root directory (/), all subsequent files and directories branch out in a tree-like layout, illustrating the hierarchical nature of the Linux filesystem. Logical arrangement and intuitive navigation are both ensured by this framework. Memory blocks and fragmentation procedures are under the purview of the filesystem, which also defines naming conventions such as maximum filename length and case sensitivity, and controls the underlying data organization.

Linux Filesystem Layers

In Linux, the filesystem is complex and multi-layered, with three main layers that play different but complementary roles:

- **Logical File System:** The logical file system is the layer that user applications employ to interact with the file system directly. It acts as the main interface between users and applications, handling high-level actions including opening, reading, and shutting files.
- **Virtual File System (VFS):** Multiple instances of physical file systems can run in tandem thanks to the VFS, an essential abstraction layer. It offers a standard interface that lets different file systems (such as Ext4, XFS, and NTFS) coexist and function together without applications needing to know which underlying file system is in use. The VFS is an essential piece of middleware because it decouples programs from the specifics of the storage type that actually holds the data. A file-opening program only needs to interact with the VFS; it doesn't need separate logic for Ext4, XFS, or NTFS. This architectural design is crucial to Linux's flexibility, allowing users to mount and access data from different storage media without recompiling or redesigning applications. This includes USB drives, network shares, and partitions from other operating systems. This capability is the backbone of Linux's strong compatibility in diverse computing environments.
- **Physical File System layer:** Managing and storing the actual memory blocks on the disk is the job of the Physical File System layer. It manages the nitty-gritty of data storage and retrieval, communicates directly with hardware components, and makes sure physical storage resources are used efficiently.

3.4. Common Filesystem Types

Linux supports a wide array of filesystem types, each optimized for specific use cases and offering distinct features:

- **Ext (Extended File System):** Introduced in 1992, this was the first filesystem specifically designed for Linux.
- **Ext2:** Developed in 1993, Ext2 is a non-journaling filesystem often favored for flash drives and SSDs due to its simplicity and reduced write overhead.
- **Ext3:** This is a journaling filesystem that enhances data integrity by logging changes before they are permanently committed to disk, which significantly aids in recovery after system crashes.
- **Ext4:** The most prevalent filesystem in Linux, developed in 2006. As a journaling filesystem, it offers backward compatibility with Ext2 and Ext3. It balances performance and reliability with features such as persistent pre-allocation, support for an unlimited number of subdirectories, and metadata checksumming. Ext4 is the default filesystem for many Linux distributions.

- **XFS:** A high-performance journaling filesystem, frequently deployed in enterprise environments for large filesystems and applications requiring high data throughput.
- **JFS (Journaled File System):** Another journaling filesystem, originally developed by IBM.
- **Btrfs (Better/Butter/B-tree FS):** Developed in 2007, Btrfs provides advanced features including snapshotting, drive pooling, data scrubbing, self-healing capabilities, and online defragmentation.
- **SquashFS:** A compressed, read-only filesystem commonly used for live CDs, embedded systems, and system recovery environments.

Detailed Breakdown of Key Directories and Their Purposes

Under the FHS, all files and directories are organized under the single root directory (/), regardless of whether they are stored on different physical or virtual devices.

- **/ (Root Directory):** This is the primary hierarchy root and the starting point of the entire filesystem. It is generally kept small and contains highly critical files, sufficient for the system to function in single-user mode and to provide tools for system repair.
- **/bin:** Contains essential command binaries, such as ls, cp, and mv, which are necessary for the system to operate, particularly in single-user mode, and are available to all users.
- **/boot:** Stores bootloader files, including the Linux kernel.
- **/dev:** Houses device files that represent hardware components. Examples include character devices like /dev/tty (for terminals) and block devices like /dev/sda (for hard drives). These special files enable users to interact with hardware as if they were ordinary data files.
- **/etc:** Often referred to as the "nerve center," this directory contains system-wide configuration files. Nearly every service or application on the system has its configuration file located here.
- **/home:** Contains personal directories for each user, where they store their personal files and directories. Separating /home onto its own partition significantly simplifies backup procedures.
- **/lib:** Stores essential shared libraries required by the binaries located in /bin and /sbin.
- **/mnt:** Traditionally used as a temporary mount point for filesystems, such as removable media.
- **/opt:** Reserved for optional software packages. Linux distributions typically do not install anything in this directory; it is designated for local administrators to install additional software without concerns of overwrites during system updates.
- **/proc:** A virtual filesystem that provides a dynamic interface to kernel data structures, process information, and kernel parameters. It is a "process information pseudo-file system" that does not contain "real" files but rather runtime system information.
- **/root:** The dedicated home directory for the root user.
- **/sbin:** Similar to /bin, but contains system binaries primarily used by the root user for administrative tasks.
- **/tmp:** Stores temporary files. Data in this directory is often cleared upon system reboot.
- **/usr:** A secondary hierarchy that holds user programs and data. It contains all commands, libraries, manual pages, and other unchanging files necessary for normal system operation. It is typically split into subdirectories such as /usr/bin, /usr/sbin, and /usr/lib.
- **/var:** Stores variable data files that frequently change, such as logs, databases, and email spools. This directory often grows in size over time. Historically, everything in /var was located under /usr, but separating it allows /usr to be mounted as read-only, enhancing security.

Commands to display file system in Linux

1. df command

The df command is a powerful tool for displaying information about disk space usage in Linux. It can also be used to view the file system type and other related information by running the following command:

```
df -Th
```

Columns such as Filesystem, Type, Size, Used, Available, and Use% are displayed in the output produced by the -Th option. For example:

2. lsblk Command

```
lsblk -f
```

Types of file systems and additional information, such as mount points, are shown by this command. This is one possible output:

```
(geeksforgeeks@kali)-[~]
$ lsblk -f
NAME FSTYPE FSVER LABEL UUID                                FSAVAIL FSUSE%
OUNTPOINTS
sda
├─sda1
│   ext4      1.0              174b531d-a947-4dde-97d2-674fb0d2d619    7.5G     63%
├─sda2
└─sda5
    swap       1              a70392d8-1539-41a0-91df-638b1905cc28
SWAP]
sr0
```

3. blkid Command

You can get a complete inventory of all block devices, along with their corresponding file system types and UUIDs, by using this command.

```
blkid
```

4. findmnt Command

You can view the types of mounted file systems using this command:

```
findmnt -o SOURCE,FSTYPE
```

```
(geeksforgeeks@kali)-[~]
$ findmnt -o SOURCE,FSTYPE
SOURCE      FSTYPE
/dev/sda1   ext4
sysfs       sysfs
securityfs  securityfs
cgroup2     cgroup2
pstore      pstore
bpf         bpf
debugfs     debugfs
tracefs     tracefs
fusectl     fusectl
configfs    configfs
proc        proc
systemd-1   autofs
binfmt_misc binfmt_misc
udev        devtmpfs
```

findmnt command to display file system type in linux

Practical Activity 3.1: Demonstrate the File system in Linux**Materials Required:**

- A computer with any Linux distribution installed (e.g., Ubuntu)
- Access to the Terminal

Steps:

Step 1: Open the Terminal: Press Ctrl + Alt + T to open the terminal.

Step 2: View the Root Directory

- Type: `ls /`
- Observe the main directories like `/bin`, `/etc`, `/home`, `/usr`, etc.

Step 3: Explore the Purpose of Each Directory Using Commands Below:

Command	Purpose
<code>ls /bin</code>	View essential user commands like <code>ls</code> , <code>cp</code> , <code>mv</code>
<code>ls /boot</code>	View files required for booting (like the kernel)
<code>ls /dev</code>	View device files (e.g., <code>/dev/sda</code> for hard disk)
<code>ls /etc</code>	View system configuration files
<code>ls /home</code>	View user directories (e.g., <code>/home/student</code>)
<code>ls /lib</code>	View shared libraries required by binaries
<code>ls /root</code>	View the home directory of the root user
<code>ls /sbin</code>	View system management commands for administrators
<code>ls /tmp</code>	View temporary files (auto-deleted on reboot)
<code>ls /usr</code>	View user-installed software and documentation
<code>ls /var</code>	View variable files like logs, spool data, etc.

Step 4: Optional: Try `cd` into each directory and explore contents, for example:

- `cd /etc`
- Then type `ls` to list files.

3.3. Overview of Linux operating system

Linux, created by Linus Torvalds in 1991, is an open-source software platform known for its robustness and flexibility. It powers a wide range of devices, including computers, servers, smartphones, and embedded systems. Linux is renowned for its reliability, security, and adaptability, allowing users to customize and modify their environments based on individual needs. Supported by a large and active community, Linux remains a popular choice among individuals and businesses due to its extensive resources and ongoing development.

A "distribution"—what is it?

Linux is an operating system that includes a collection of programs built on top of the Linux kernel. It comes with its own set of libraries and software. Alternatively, users can choose an operating system that is built on top of Linux. There are several Linux distributions available for download and use on a variety of devices, including embedded systems and personal computers. More than 600 Linux distributions exist, with some of the most well-known being:

- Linux MX
- The Manjaro
- Mint for Linux
- Basic
- Ubuntu
- Debian
- The Solus

- Fedora
- openSUSE
- Deepin

Why utilize Linux?

The open-source, free, and highly versatile nature of Linux makes it a popular choice. It is widely adopted by both server administrators and developers due to its exceptional security, stability, and performance. Among Linux's many advantages are its extensive software library, broad hardware compatibility, and an active community that provides support and regular updates. Whether for home or office use, Linux's adaptability allows users to customize the operating system to meet their specific needs.

Linux's architecture

The components of the Linux architecture are as shown in Figure 3.1.

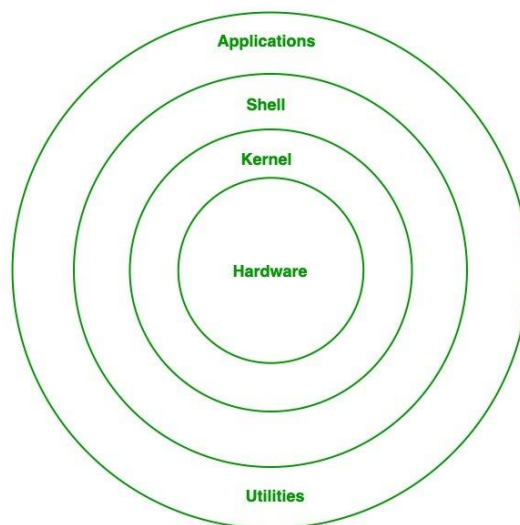


Fig 3.1. Components of the Linux architecture

Kernel: The kernel is the core component of any operating system based on Linux. It creates isolated virtual environments for each process by managing the computer's shared resources. This makes it appear to the user as if only one process is running at a time. The kernel is also responsible for minimizing and preventing conflicts between programs.

There are several varieties of kernels:

- Kernel Monolith
- Kernels that are hybrid
- Exo kernels
- little kernels

System Library: System libraries, commonly called shared libraries, are essential to Linux and enable many operating system capabilities. Applications can use these libraries and utilize the pre-written code to do various tasks. Developers save a ton of time by utilizing these libraries instead of writing the same code over and over again. The system libraries mediate interactions between the kernel and applications, allowing for efficient and consistent data transfer between the two.

Shell: To interact with the Linux operating system, one must use Shell. The shell is the interface between the user and the system that allows them to input commands, which are then deciphered and executed. Shells mediate between users and kernels by passing user requests on to them for processing. File management, system configuration, and program execution are just a few of the many useful functions it offers the user.

Hardware Layer: The central processor unit (CPU), input/output devices (I/O), random access memory (RAM), and hard disk drive (HDD) are all components of the computer that make up this layer. In addition to providing the resources needed for the system and its programs to execute successfully, this layer also manages communication with the Linux OS.

The smooth operation of various hardware components and system libraries is guaranteed by the communication and control provided by the Linux Kernel.

System Utility: System utilities are essential tools and applications designed for managing and configuring various system components in the Linux operating system. Using these applications, one may accomplish a great deal, including installing software, configuring networks, monitoring system performance, controlling users and permissions, and much more besides. System utilities simplify system administration tasks, allowing users to maintain their Linux systems more successfully.

3.4. File storage and performance across file systems

Unless there's a really compelling reason to do so, we advise against transferring files across different operating systems. When working in a Linux command-line environment (such as Ubuntu or OpenSUSE), it is recommended to save your files in the WSL file system for optimal efficiency. Use Windows' file system to save any files you work with in Windows' command-line interfaces (PowerShell, Command Prompt).

As an illustration, when you're storing your WSL project files:

Navigate to the root directory of the Linux file system: `/home/<user>/`.

File path: `/mnt/c/Users/<user>/`, not the Windows file system root directory.

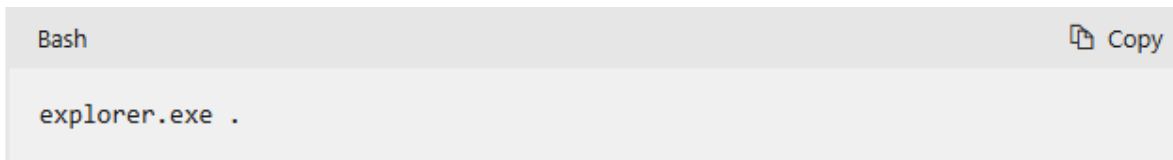
Incorrect file path: `Project$` or `C:\Users\<user>\Project`

In a WSL command line, working from a mounted drive is indicated by the presence of `/mnt/` in the file path. Once mounted in a WSL command line, the Windows file system `C:/` drive (`C:\Users\<user>\Project`) will appear as: `/mnt/c/Users/<user>/Project$`.

You can save your project files on a mounted drive if you want; however, storing them directly on the `\\wsl$` disk will boost your performance speed.

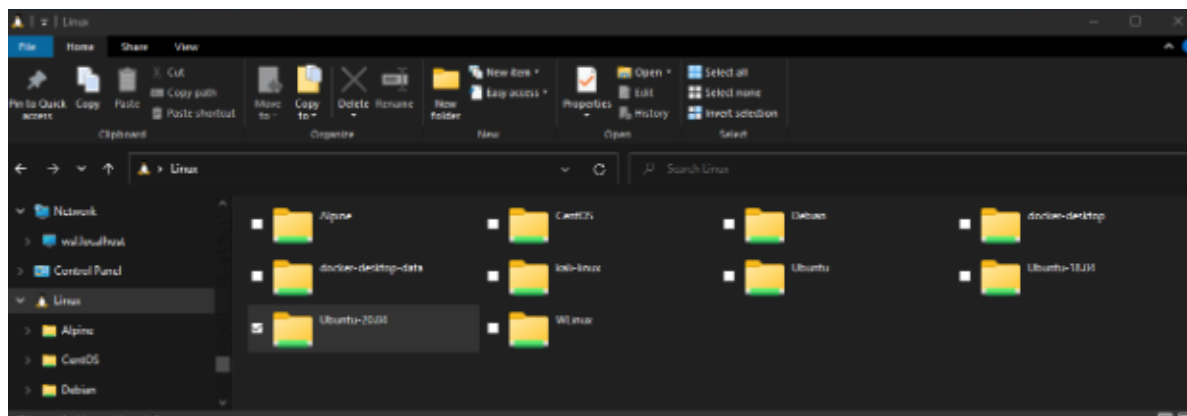
View your current directory in Windows File Explorer

Opening Windows File Explorer from the command line allows you to see the location of your files. Just type:



The command `powershell.exe /c start` is another option. To access the current directory, finish the command with a period.

Enter `\\wsl$` in the address bar of Windows File Explorer to see all the Linux distributions you have installed and their root file systems.



Interoperability between Windows and Linux commands

With WSL, you may use the same commands and tools that you would with Windows or Linux.

Use the Linux command line to run Windows programs (like notepad.exe) in an Ubuntu environment.

Utilize PowerShell on a Windows command line to access Linux tools, such as grep.

Allow Linux and Windows to share environment variables

Run Linux tools from a Windows command line

The Windows Command Prompt (CMD) or PowerShell can be used to run Linux binaries by utilizing the wsl command or wsl.exe command.

PowerShell

```
C:\temp> wsl ls -la
<- contents of C:\temp ->
```

Binaries invoked in this way:

After you run this program, you'll see your default Linux distribution username and be prompted to enter your password. Once you enter your password correctly, the distribution will begin downloading updates. Keep the current directory the same as in the command prompt (CMD or PowerShell). Start with the privileges of the WSL default user, which will have the same administrative privileges on Windows as the process and terminal that initiated the call.

Practical Activity 3.2: Create, Rename, and Delete Files and Directories in Linux and Windows

Materials Required:

- A computer with Windows OS
- A computer with Linux OS (or dual-boot/VM access)
- Basic understanding of using File Explorer and Terminal

Part A: Using File Explorer in Windows

Steps:

1. Open File Explorer.
2. Navigate to the Desktop or Documents folder.
3. Create a New Folder: Right-click → New → Folder → Name it MyFolder.
4. Create a New File inside the folder: Right-click → New → Text Document → Name it notes.txt.
5. Rename the File: Right-click on notes.txt → Rename → Change it to summary.txt.
6. Delete the File and Folder: Right-click → Delete.

Part B: Using Terminal in Linux

Steps:

1. Open the Terminal (Ctrl + Alt + T).
2. Navigate to your home directory:
 - cd ~
3. Create a Folder:
 - mkdir MyFolder
4. Create a File inside the folder:
 - cd MyFolder
 - touch notes.txt
5. Rename the File:
 - mv notes.txt summary.txt
6. Delete the File and Folder:
 - rm summary.txt
 - cd..
 - rmdir MyFolder

3.5. Linux Software Installation

Installing software on Linux is easy. Use package managers such as apt and sudo apt install package_name for Debian-based systems (like Ubuntu); use dnf and sudo dnf install package_name for Fedora-based systems. A graphical application installation and search interface can also be found in software centers. The linked resource contains comprehensive instructions for installing Python.

Benefits of Linux

- The principal advantage of Linux is its status as an open-source operating system. This indicates that the source code is openly accessible to anyone, allowing individuals to modify, enhance, and distribute it freely without the necessity of permission.
- In comparison to other operating systems, Linux is the most secure. While Linux is susceptible to certain viruses, it is less prone to vulnerabilities compared to other operating systems; however, this does not mean it is entirely secure. Consequently, antivirus software is often considered unnecessary. Linux software upgrades are routine and straightforward.
- Many Linux distributions are available for use based on your requirements or preferences. Linux is available for free use on the internet.
- It possesses substantial community backing.
- It provides a significant degree of stability. It seldom freezes or lags, and a restart is rarely necessary.
- It safeguards the user's privacy.
- The Linux operating system exhibits markedly superior performance relative to other operating systems. It manages resources efficiently and allows numerous users to collaborate concurrently.
- It functions effectively with networks.
- Linux has considerable adaptability. It is not necessary to install the complete Linux suite; you are only required to install the essential components.
- A multitude of file formats are compatible with Linux.
- Downloading it over the internet is efficient and uncomplicated. Furthermore, it may be installed on any hardware, including obsolete computer systems.
- Despite possessing limited hard disk capacity, it performs all tasks accurately.

The drawbacks of Linux

- It is not very easy to use; therefore, it can be perplexing for novices.
- Compared to Windows, it has fewer peripheral hardware drivers.

Commands for Linux

Linux commands allow the command line interface to communicate with the OS. pwd displays the path of the current directory, cd changes directories, and ls lists the contents of directories; these are among the most commonly used commands.

Common commands used to manipulate files include cp, mv, and rm. You can access system statistics and manage processes using commands like free for memory usage, df for disk space usage, and top for monitoring system processes. ping, ipconfig, and netstat are some of the networking tools that users can use to troubleshoot connections.

Different distributions utilize different package managers; for example, dnf is used by Fedora systems and apt is used by Debian-based systems. awk, sed, cat, and grep are essential commands for working with text files.

Simple Terminal Commands for Linux

1. In Linux, the ls command shows details about the files located in the current directory.
2. pwd is a Linux command that shows the currently selected working directory.
3. Linux's mkdir command creates a directory.
4. Linux's cd command is used to navigate between different folders.
5. Linux's rmdir command removes empty directories from the directory list.
6. Linux's cp command transfers files between directories.

7. Linux's mv command moves and renames files.
8. Linux's rm command deletes files.
9. Linux's uname command is used to obtain basic information about the operating system.
10. Linux's find command searches through the directory hierarchy for a file.
11. Linux's touch command creates empty files.
12. In Linux, the ln command creates shortcuts (symbolic or hard links) to other files.
13. Linux's cat command displays the contents of a file in the terminal.
14. The clear command in Linux clears the terminal screen.
15. Linux's ps command displays currently running processes in the terminal.
16. Linux's man command accesses the manual for any Linux command.
17. Linux's grep command searches for a specific string in the output.
18. Linux's echo command sends text or a string to the terminal.
19. Linux's wget command downloads files from the internet.
20. Linux's whoami command shows the name of the current user.
21. Linux's sort command sorts the contents of a file.
22. Linux's cal command displays a calendar in the terminal.
23. Linux's whereis command shows the exact location of any command provided.
24. In Linux, the df command displays file system disk space usage.
25. Linux's wc command provides the number of lines, words, and characters in a file.

3.6. Hard links Vs Soft links

In Linux, every file begins with a single hard link. This link represents the connection between the filename and the data stored on the filesystem.

Soft links, also called symbolic links, can connect both regular and non-regular files. Additionally, they can span multiple filesystems. A soft link is defined as a special type of file that refers to an existing one.

Linux file permissions form the foundation of its security model, determining who can access, modify, or execute files and directories. A thorough understanding of these permissions is crucial for maintaining system integrity and preventing unauthorized access.

Permissions for Files

- **Read (r):** This permission enables a user to view the contents of a file. Commands such as cat or less can be used to display the file's content, and text editors like Vi can open the file for viewing. The ability to make copies of a file also necessitates read permission, as accessing the file's data is a prerequisite for duplication.
- **Write (w):** Granting write permission allows a user to modify or change the file's contents. This includes the ability to alter the file using text editors or to redirect and append output to the file using shell operators like > or >>. Without write permission, any attempts to alter the file's contents are prohibited.
- **Execute (x):** Execute permission empowers a user to run the file as a program or script. This is typically applicable to commands, compiled binary applications, Bash shell scripts, or Python programs. It is important to note that while direct execution usually requires this permission, an interpreter that itself possesses execute permission can read and execute a script's instructions, even if the script file itself lacks execute permission (e.g., executing script.sh via bash script.sh, where bash is the executable interpreter).

Permissions for Directories

- **Read (r):** For directories, read permission allows a user to view the names of files and subdirectories stored within it. This permission is essential for commands like ls to successfully list the directory's contents.
- **Write (w):** Write permission on a directory permits the modification of its contents. This includes the ability to add new files (e.g., using touch), remove existing files (rm), move

files into or out of the directory (mv), or copy files into it (cp). Essentially, it controls the ability to alter the directory's structural composition.

- **Execute (x):** Execute permission for a directory allows a user to "enter" or "search" it. This means a user can use the cd command to navigate into the directory and access its contents. Even if a user cannot list the contents (lacking read permission), they can still access or run a file within the directory if its name is known. This permission is fundamental for navigating the filesystem hierarchy.

Viewing Permissions: The ls -l Command

The ls -l command is the fundamental utility for displaying file and directory permissions within the Linux terminal. This command provides a detailed, long listing of file attributes.

Interpreting ls -l Output (including special flags)

The output of ls -l typically begins with a 10-character string that represents the file's permissions and type, followed by other relevant metadata.

- **First Character:** This character indicates the type of file system entry:
 - -: Denotes a regular file.
 - d: Denotes a directory.
 - l: Denotes a symbolic link.
 - Other special flags, such as s for SetUID/SetGID or t for the Sticky Bit, can appear within the permission triplets, not in the file type indicator.
- **Next Nine Characters:** These characters are organized into three sets of three, each representing the read, write, and execute permissions for a specific ownership category:
 - Owner Permissions (first three): These characters define what the file's owner is permitted to do.
 - Group Permissions (second three): These characters specify the permissions for users who are members of the file's assigned group.
 - Other/World Permissions (third three): These characters indicate the permissions granted to all other users on the system.
 - Within each set, an r, w, or x signifies that the respective permission is enabled, while a hyphen (-) indicates that the permission is disabled.
- **Hardlinks Count:** Immediately following the permission string, an integer or number displays the total count of hard links pointing to the file.
- **Owner and Group Assignment:** The subsequent two columns clearly identify the file's owner and its assigned group, typically formatted as Owner:Group.
- **File Size, Date, and Name:** The remaining columns provide the file size in bytes, the date and time of its last modification, and finally, the name of the file or directory itself.

When inspecting directories, it is important to use the ls -ld command. The -d flag ensures that the permissions of the directory *itself* are displayed, rather than listing the contents within it.

While ls -l provides a comprehensive view of permissions and hard link counts, the ls -li command serves as a critical diagnostic tool for understanding the underlying filesystem structure. The ls -li command reveals the inode number associated with each file. This is particularly useful for verifying if two seemingly distinct files are, in fact, hard links to the same underlying data. The ability to correlate files by their inode number allows administrators to confirm the integrity of hard links and accurately assess storage utilization, as multiple hard links do not consume additional disk space since they all point to the same data block. This command moves beyond simple viewing to provide a deeper understanding of file relationships within the filesystem.

Changing Permissions: The chmod Command

The chmod (change mode) command is the primary utility used to modify file and directory permissions in Linux. It offers two main modes for specifying permission changes: symbolic and octal (also known as absolute) mode.

Symbolic Mode with Examples

Symbolic mode utilizes letters and symbols to represent permission changes, offering an intuitive approach for incremental adjustments to existing permissions.

- **Syntax:** The basic syntax for chmod in symbolic mode is `chmod [who][operator][permissions][filename]`.
- **Who (u, g, o, a):** This component specifies the target audience for the permission change: u for the user (owner), g for the group, o for others, or a for all (user, group, and others).
- **Operator (+, -, =):** This defines the type of modification to be made:
 - **+**: Adds the specified permissions to the existing ones.
 - **-**: Removes the specified permissions from the existing ones.
 - **=**: Sets the permissions *exclusively* to the specified values, effectively removing any other permissions not explicitly listed for that category.
- **Permissions (r, w, x):** These represent the specific read, write, or execute permissions to be modified.

Examples:

- `chmod o+x xyz.txt`: Adds execute permission for "others" on the file xyz.txt.
- `chmod ugo-rwx xyz.txt`: Removes all read, write, and execute permissions from the user, group, and others on xyz.txt.
- `chmod ug+rw,o-x abc.mp4`: Adds read and write permissions for the user and group, while simultaneously removing execute permission for others on abc.mp4.
- `chmod g=r myfile.txt`: Sets the group permissions to read-only, explicitly removing any other existing group permissions.
- `chmod go= [filename]`: Removes all permissions for both the group and "world" (others) categories.

When chmod is applied to a symbolic link, it dereferences the link and alters the permissions of the *target file* that the link points to, rather than changing the permissions of the symbolic link itself. To recursively apply permissions to all files and subdirectories within a given directory, the -R flag can be used; however, it is important to note that this flag does not traverse symbolic links.

Octal (Absolute) Mode with Examples

Octal mode, also known as absolute mode, employs a three-digit numerical system to represent permissions. Each digit corresponds to the sum of permissions for the owner, group, and others, respectively.

- **Numerical Mapping:**
 - Read (r) = 4
 - Write (w) = 2
 - Execute (x) = 1
 - No permission (-) = 0

To determine the octal value for each category, the numerical values of the desired permissions are summed. For example, `rwX` (read, write, execute) sums to $4+2+1=7$; `r-x` (read, execute) sums to $4+0+1=5$; and `rw-` (read, write) sums to $4+2+0=6$.

Examples:

- `chmod 777 [file_name]`: Grants full read, write, and execute permissions to the owner, group, and others. It is crucial to understand that this setting is generally **not recommended** for most files due to the significant security risk it poses.
- `chmod 755 [file_name]`: Assigns read, write, and execute (7) permissions to the owner, and read and execute (5) permissions to both the group and others. This is a common permission setting for executable scripts or directories.
- `chmod 644 [file_name]`: Grants read and write (6) permissions to the owner, and read-only (4) permissions to both the group and others. This is a typical permission setting for regular data files.

- `chmod 700 [filename]`: Provides read, write, and execute (7) permissions solely to the owner, while denying all permissions (0) to both the group and others. This setting ensures maximum privacy for the file's owner.
- `chmod 435 [file_name]`: Sets read-only (4) for the owner, write and execute (3) for the group, and read and execute (5) for others.

Absolute mode functions similarly to the `=` operator in symbolic mode, in that it exclusively sets the specified permissions, removing any other permissions not explicitly included in the numerical sum. The choice between symbolic and absolute mode is often a matter of personal preference; symbolic mode is typically easier for making small, incremental modifications, while absolute mode is often preferred for making large-scale permission changes.

The use of `chmod 777` to grant universal read, write, and execute access is widely recognized as a significant security vulnerability. This overly permissive setting can expose a Linux system to attackers by allowing unauthorized modification or deletion of files. A critical defense against such accidental exposures is the `umask` value, which proactively defines the default permissions for newly created files and directories. A properly configured `umask` ensures that files are created with more restrictive permissions by default, even if a user neglects to explicitly set them, thereby reducing the attack surface from the moment a file is created. This mechanism works in conjunction with `chmod` to enforce the Principle of Least Privilege, making it a fundamental security best practice.

The following table illustrates the numerical mapping for octal permissions:

Permission Type	Value	Common Combinations	Octal Sum
Read (r)	4	rw- (Read, Write, Execute)	7
Write (w)	2	rw- (Read, Write)	6
Execute (x)	1	r-x (Read, Execute)	5
No Permission (-)	0	--- (No Permissions)	0

Sudo Command

The `sudo` command is an indispensable utility in Linux for securely managing administrative tasks, allowing users to execute commands with elevated privileges without directly logging in as the root user.

The term `sudo` is an acronym for "substitute user do" or "super user do". Its primary function is to enable a permitted user to execute a command as the superuser (root) or any other specified user, according to the system's defined security policy. This mechanism provides temporary privilege elevation, which is essential for performing administrative tasks.

Any command executed in WSL is treated the same as the Linux command that follows `wsl` (or `wsl.exe`). Tools like pipelining, file redirection, and `sudo` are functional. One way to upgrade your default Linux distribution is by using the `sudo` command:

PowerShell

```
C:\temp> wsl sudo apt-get update
```

Practical Activity 3.3: Demonstrate and Configure Permissions, Use of Sudo Command, and Working with Hard & Soft Links in Linux.

Materials Required:

- Linux OS with terminal access
- Basic user account with `sudo` access

Steps:

Step 1: Create a File: "touch demo.txt"

Step 2: View Default Permissions: `ls -l demo.txt`

Step 3: Change Permissions

- Give read, write, and execute permission to the user:
"chmod u+rw demo.txt"

- Remove write permission for others:
“chmod o-w demo.txt”
- View changes:
“ls -l demo.txt”

Step 4: Use sudo to Create a File in a Protected Folder

“sudo touch /root/specialfile.txt”

Step 5: Create Hard and Soft Links

- Create a hard link:
“ln demo.txt demo_hard.txt”
- Create a soft link:
“ln -s demo.txt demo_soft.txt”
- View all:
“ls -l”

Summary

1. Both graphical user interfaces (GUIs) and command-line interfaces (CLIs) exist for interacting with software and computer systems.
2. The command-line interface (CLI) allows users to interact with an operating system by entering detailed commands in a terminal or console window.
3. GUI stands for Graphical User Interface. GUIs allow users to interact with operating systems visually.
4. The Linux kernel is the brain of the operating system, managing communication between the machine and its components.
5. The Linux kernel, created by Linus Torvalds in 1991, is a free and open-source operating system.
6. The kernel is the core of any Linux-based operating system. It creates an isolated virtual environment for each process by managing access to the system's shared resources.
7. To interact with the Linux operating system, one must use the Shell. The shell serves as the interface between the user and the system, allowing the input of commands that are then interpreted and executed.
8. System utilities are essential tools and applications designed to manage and configure various components of the Linux operating system.
9. A primary advantage of Linux is its status as an open-source operating system.
10. Linux commands enable the command-line interface to communicate directly with the operating system.

Check your progress**A. Multiple Choice Questions**

1. Which of these commands is used for removing files?
 - a. erase
 - b. delete
 - c. rm
 - d. dm
2. Which option of the rm command is used to remove a directory with all its subdirectories?
 - a. -b
 - b. -o
 - c. -p
 - d. -r
3. The command used to count the number of characters in a file is:
 - a. grep
 - b. wc

- c. count
 - d. cut
4. The total number of primary partitions that can exist on one drive is:
 - a. 1
 - b. 2
 - c. 16
 - d. 4
 5. What is the core of the Linux operating system?
 - a. Terminal
 - b. Kernel
 - c. Command
 - d. Shell
 6. What type of installation requires a special boot disk?
 - a. FTP
 - b. NFS
 - c. PCMCIA
 - d. All of the above
 7. Which command is used to print a file?
 - a. wc
 - b. pg
 - c. lp
 - d. print
 8. Which of the following commands is used to change the working directory?
 - a. changed
 - b. chdr
 - c. dr
 - d. cd
 9. Which of the following commands is used to see the services running on the NFS server?
 - a. serverinfo
 - b. rpcinfo
 - c. infserv
 - d. NFSinfo
 10. Which interface allows users to interact with an operating system by entering detailed commands in a terminal or console window?
 - a. Graphical User Interface (GUI)
 - b. Command-Line Interface (CLI)
 - c. Application Programming Interface (API)
 - d. Kernel Interface

B. Fill in the blanks:

1. _____, commonly called shared libraries, are essential to Linux and enable many operating system capabilities.
2. _____ mediate between users and the kernel by passing user requests on to it for processing.
3. _____ simplify system administration tasks, allowing users to maintain their Linux systems more effectively.
4. The Linux operating system is extensively employed across various areas due to its _____ nature, versatility, and security.
5. The _____ command in Linux displays the path of the current directory.
6. The _____ command is an indispensable utility in Linux for securely managing administrative tasks, allowing users to execute commands with elevated privileges without directly logging in as the root user.

7. Due to its ability to efficiently utilize system resources while maintaining functionality, _____ is frequently chosen by users who want lightweight operating systems or possess outdated hardware.
8. _____ interfaces let users interact with operating systems visually.
9. Linux, developed by _____, is an open-source OS similar to Unix that offers a strong command-line interface, robust security, and a high degree of freedom.
10. The _____ interface allows users to interact with an OS by entering detailed commands in a terminal or console window.

C. True or false

1. Linux is an operating system.
2. The Debian Package Manager (DPM) is the default package manager used by Ubuntu Linux.
3. The line that configures the hostname for the computer at boot time can be found in /etc/hostname.
4. UIDs and GIDs are unique to the system and, once used, can never be reused.
5. To create a container using the docker build command, you must first create a Dockerfile.
6. To install a new program from RPM software repositories on the Internet, you can use the dnf install programname command..
7. The pwd command in Linux prints the path of the current working directory.
8. You can clear a log file simply by redirecting nothing into it.
9. The Linux kernel, created by Linus Torvalds in 1991, is an open-source and free operating system.
10. The shell is the interface between the user and the system that allows them to input commands, which are then deciphered and executed.

D. Short Answer Questions

1. Who developed Linux and in which year was it released?
2. What is the main difference between GUI and CLI?
3. Name any two Linux distributions.
4. What is the function of the /home directory in the Linux filesystem?
5. What is the role of the kernel in the Linux operating system?
6. Which command is used to install software on Ubuntu or Debian-based systems?
7. State any two benefits of using Linux as an operating system.
8. What is the purpose of the chmod command in Linux?
9. Differentiate between a hard link and a soft (symbolic) link in Linux.
10. What permission is required to run a file as a program or script in Linux?

Answer Key**A. Multiple Choice Questions**

1. c., 2. d., 3. b., 4. d., 5. b., 6. d., 7. c., 8. d., 9. b. 10. b

B. Fill in the blanks:

1. Dynamic link libraries (DLLs), 2. Shells, 3. Utilities, 4. Open-source, 5. pwd, 6. sudo, 7. Linux, 8. Graphical user, 9. Linus Torvalds, 10. Command-line

C. True or False

1. True, 2. True, 3. True, 4. False, 5. True, 6. False, 7. True, 8. True, 9. False, 10. True

Unit-2.

Introduction to Network Management

Chapter-1**Introduction to Computer Network**

Aanya asked her teacher, “What is a computer network?” Mr. Raj replied, “It’s when computers are connected to share data. They connect in patterns called topologies—like bus, star, or ring.” He drew a star shape. “This is a star topology, where all systems connect to a central point.” Aanya smiled, “So, networks are like puzzles with different shapes!”

“Exactly,” said Mr. Raj. “Each shape helps computers work together.” Raj teaching Aanya about computer network.



In this session, you will learn about Computer network, its topologies, various network devices and transmission medias.

1.1 Introduction to Computer Network

A computer network can be described as a connected system of more than two computer devices that can exchange data and share resources. This is achieved through defined communication protocols such as TCP/IP, HTTP, and DNS, over specific transmission media. The system facilitates the transfer of information by encoding it into signals that propagate through physical or wireless media, guided by network architectures and controlled by networking hardware and software. The efficiency and reliability of this data exchange are governed by factors such as bandwidth, latency, and network topology. Furthermore, network management involves applying scientific principles to monitor, analyze, and optimize the performance and security of these interconnected systems.

Types of Computer Networks

PAN: Personal Area Networks (PANs) are small networks that connect electronic devices over very short ranges, typically within a single workspace. They facilitate data and resource exchange between devices such as computers, smartphones, and peripherals.

Assignment 1.1:

1. List some advantages of Computer Network
2. List down various components of Computer Network
3. List down the types of topologies in Computer Network
4. List the 2 types of network models in Computer Network

LAN: Locally Area Networks (LANs) are organized networks that typically cover small areas such as offices or campuses. They are commonly used to connect computers and other devices within a single building or a group of nearby buildings.

MAN: A Metropolitan Area Network (MAN) is a type of computer network that connects devices across a metropolitan area, such as a city or a large campus. It is larger than a Local Area Network (LAN) but smaller than a Wide Area Network (WAN). MANs are commonly used by organizations, municipalities, or institutions to connect various buildings, campuses, or entire city-wide infrastructures.

WAN: A Wide Area Network (WAN) is a network that covers a vast geographical region, such as a state, city, country, or even the entire world. It is designed to interconnect multiple LANs and is commonly used for long-distance communication.

Benefits of Computer Network

In our increasingly digital world, computer networks have become essential. They offer numerous advantages that impact how we learn, work, and interact. Let's explore some of these key benefits:

1. Sharing Resources: Networks allow many users to use hardware (like printers and scanners) and software (which is typically applications). For example, in a school lab, all the computers can use one printer.

2. Communication and Collaboration: Networks also allow you to send emails, attend online classes, or collaborate together on shared documents regardless of where each user is physically located.

3. Centralized Data and Easy Access: Central files can be stored on a server. This allows for backing up data in one location, ensuring more security and that everyone is working from the most current information. For example, students and teachers can access study materials from a central server.

4. Efficiency and Productivity: By providing immediate network access to information or resources, networking saves time and gives people the ability to work faster as groups do in schools, in businesses, or in office work.

5. Information, Business, and Entertainment: The internet, being the largest network, can provide knowledge (online libraries, educational sites), business opportunities (e-commerce, transactions), and entertainment (movies, games, music and social media).

1.2. Various Components & Topologies

Key components of a computer network include end devices, which users directly interact with to send or receive data (for example – computers, smartphones, printers, etc.); networking devices that facilitate communication and manage data flow within the network (for example – hubs, switches, routers, etc.); and transmission media that carry data signals between devices.

The key concepts and components of a Computer Network are detailed below:

Node: A device that is connected to a network, such as a computer, server, printer, etc.

Link: A communication pathway, which can be physical (e.g., fiber optic cables) or wireless (e.g., microwaves).

Network Interface Card (NIC): A hardware component within a device that enables it to connect to a network.

Topology: The physical and logical arrangement of links and nodes in a network (e.g., Bus, Star, Ring, Mesh, Tree).

Protocol: A set of rules and procedures that govern how data is transmitted over a network (e.g., TCP/IP, HTTP, DNS, Ethernet, Wi-Fi).

Packet: A small unit of data transmitted over a network, containing both addressing information and the actual data.

IP Address (Internet Protocol Address): A unique numerical identifier assigned to each device that participates in a network using the IP protocol.

MAC Address (Media Access Control Address): A unique hardware address assigned to a device's NIC.

Topologies in Computer Network

Topology in a computer network refers to the layout or arrangement of links, nodes, or devices. Different types of topologies include Bus, Star, Ring, Mesh, and Tree. It defines the structure in which nodes are connected and the manner in which data is transferred across the network.

1. Physical Network Topology: It describes the actual physical layout of cables, devices, and other hardware components used to transmit data in a network.

2. Logical Network Topology: It refers to the way data flows between nodes in a network, regardless of the physical layout or connections. The proper design and connection of a network are essential for its efficient and reliable operation.

1.3. Types of Network Topologies:

1. Bus Topology

It is a category of network in which each individual network device and computer is connected to a single cable, called the backbone cable, as shown in Figure 1.1. This topology is bi-directional in nature, allows multi-point connections, and is considered non-robust because if the backbone cable fails, the entire network will stop functioning.

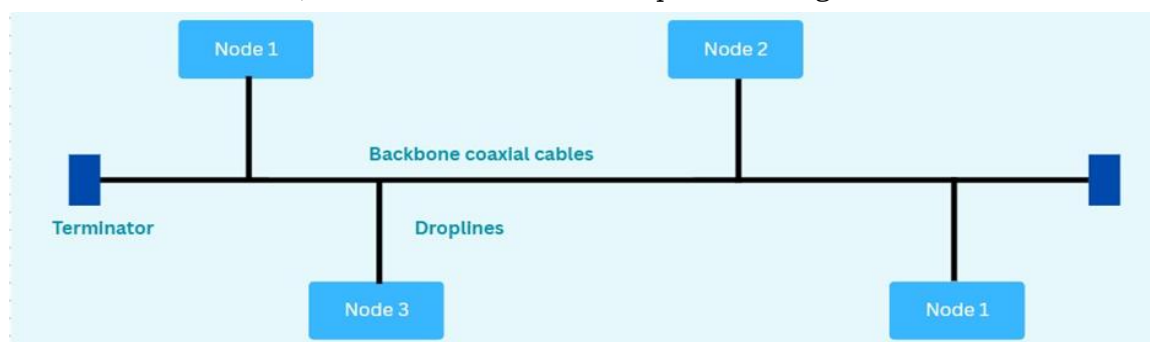


Fig 1.1. Bus topology

Pros of Bus Topology:

- If N number of devices are connected in a bus topology, only one cable, known as the backbone cable, is required, along with N drop lines to connect the devices to the backbone.
- The overall cable cost is lower compared to other network structures.
- Suitable for creating small networks.
- It is a well-known and easy-to-implement technology.

Cons of Bus Topology:

- Although relatively simple, it still requires a considerable amount of cabling.
- If the backbone cable fails, the entire network goes down.
- Adding new devices to the network can slow down performance.

2. Star Topology

In this topology, all nodes are interconnected via cables to a single central server, as shown in Figure 1.2. This central server can be a hub or a switch. The central server acts as the main computer, and various nodes are connected to it.

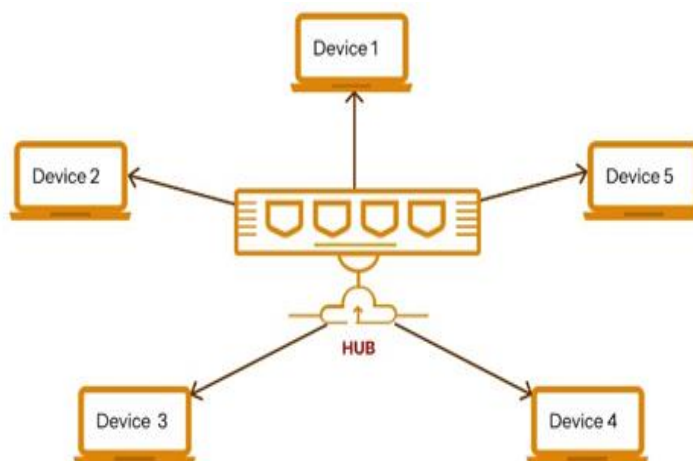


Fig 1.2: Star topology

Pros of Star Topology:

- If multiple devices (e.g., N) are linked to the system in this topology, then the number of cables needed to interlink them is N. That's why it is easy to maintain.
- Each node requires one port to link to the hub or switch; therefore, the total number of ports required is N.
- Robust in nature, i.e., if one link fails, the network will continue to function.

Cons of Star Topology:

- If the main server fails, the entire system will fail.
- The infrastructure cost of setting it up is also on the higher side.

3. Ring Topology

This forms a ring linking devices with just two neighboring devices, as shown in Figure 1.3. A lot of repeaters are used when there are a large number of nodes. The flow of data is in a single direction, i.e., it is unidirectional. The most prevalent method in this topology is token passing, in which a token is passed from one node to another. It is a frame that circulates throughout the entire setup.

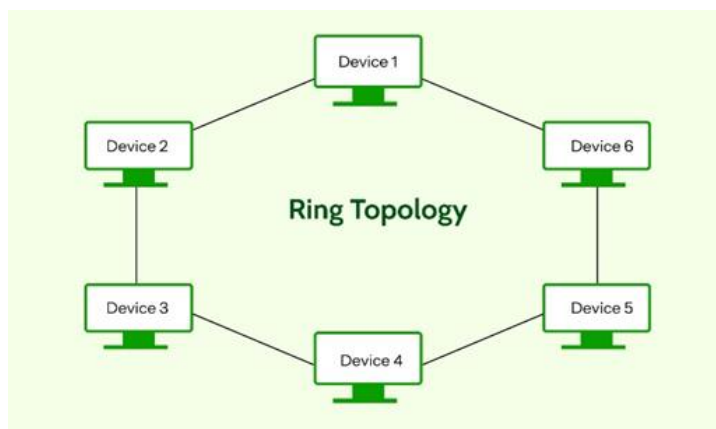


Fig 1.3. Ring topology

Pros of Ring Topology:

- High-speed data communication is possible.
- Expansion and installation are inexpensive.

Cons of Ring Topology:

- If one node fails, the entire network fails.
- Less secure.

4. Mesh Topology

In this, each individual node is linked to another via a specific channel, as shown in Figure 1.4. Each node is connected to the next node through communication channels, which are known as links.

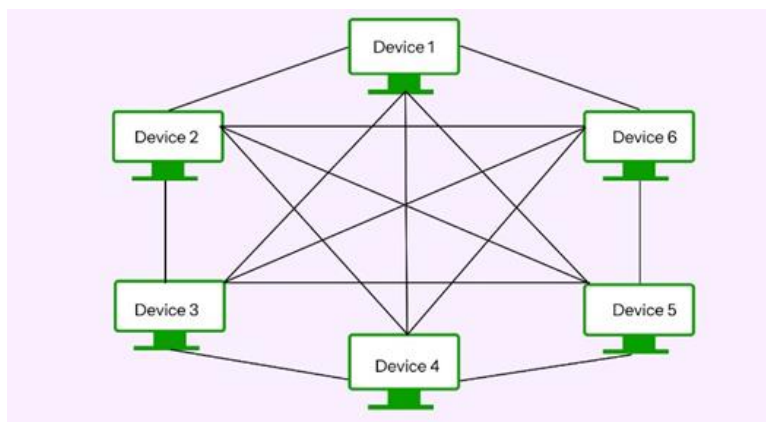


Fig 1.4. Mesh topology

Pros of Mesh Topology:

- Fast transmission between nodes.
- It is robust in nature.
- Faults can be easily detected. Data is transmitted among the devices over dedicated channels or links, which makes it reliable.
- This topology offers high security.

Cons of Mesh Topology:

- Setup and installation are complex.
- High cable costs are involved as a large amount of wiring is required; hence, it is better suited for a smaller number of devices.
- High maintenance is needed.

5. Tree Topology

Tree topology is a blend of Star topology and Bus topology, as shown in Figure 1.5. This topology features a hierarchical flow of information. Data moves from top to bottom, i.e., from the main server to the secondary server and then to the devices, or from bottom to top, i.e., from the devices to the secondary server and then to the main server. It is not a very reliable topology, as a failure in the main backbone (pillar) can cause the entire system to fail.

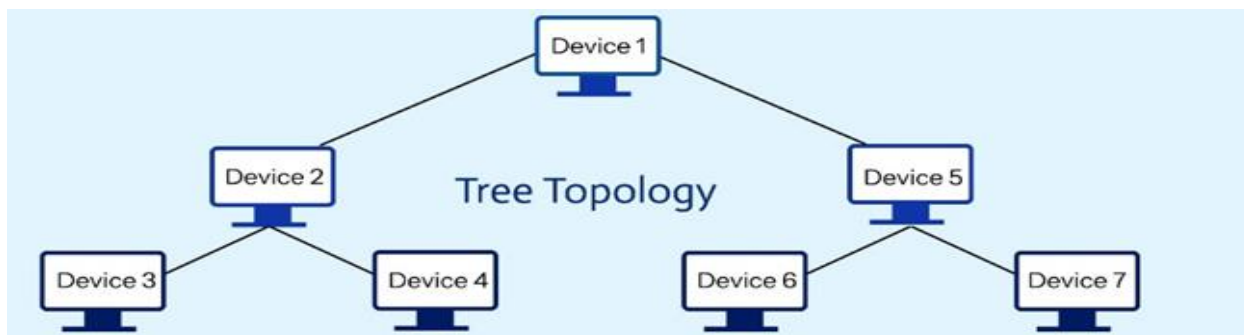


Fig 1.5. Tree topology

Pros of Tree Topology

- Allows other nodes to be attached to the main server, reducing the distance the signal must travel to connect with nodes directly.
- New devices can be added to the existing network.

Cons of Tree Topology

- If the central hub fails, the entire system will fail.
- High infrastructure cost is involved.

6. Hybrid Topology

This topology is a mixture of all the topologies discussed above. It is used when devices are arranged in various configurations. This means it can include elements of Star or Ring topology or a combination of different types of topologies mentioned earlier, as shown in Figure 1.6.

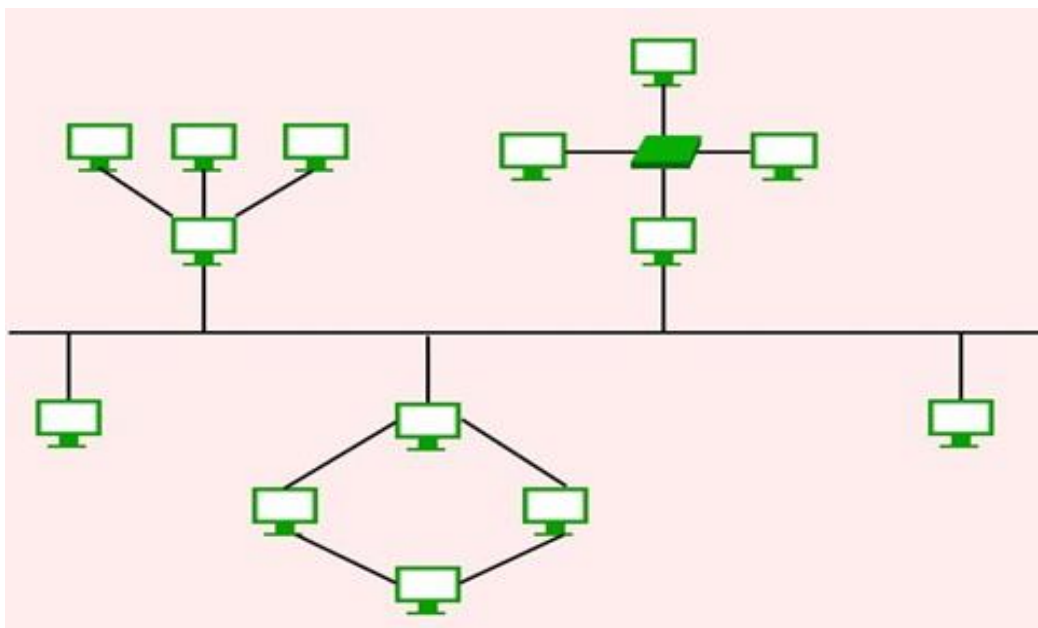


Fig 1.6. Hybrid topology

Pros of Hybrid Topology:

- A flexible type of topology.
- Expansion is possible by adding new devices to the network.

Cons of Hybrid Topology:

- Designing a hybrid network can be challenging.
- High organizational cost is involved, as a lot of cabling is required.

1.4. Networking Devices

To enable communication and data transmission within and between computer networks, various dedicated hardware devices are used. Each device plays a specific role in handling and directing network traffic. Let's explore some key networking devices:

1. Repeater:

Think of a repeater like a megaphone for a signal traveling via a long wire. As electrical signals travel over elongated distances in wired media, they tend to weaken or lose strength because of attenuation.

A repeater accepts a weak signal, strengthens it, and then retransmits the strengthened signal. This extends the distance a signal can travel without significant degradation.

2. Hub:

A hub is a central linking point for multiple devices in a system, typically used in a star topology. Broadcasting of packets takes place when a data packet is passed from one device to all other connected devices. If the packet is addressed to it, then the destination device processes it accordingly, as shown in Figure 1.7.



Fig 1.7. A hub

3. Switch:

A switch is a network device that connects multiple devices in a network and intelligently forwards data to the intended recipient. Unlike a hub, which sends data to all connected devices, a switch reads the MAC address of incoming data packets and forwards them only to the specific device (node) they are meant for. The pictorial representation of Switch is shown in Figure 1.8.

It acts like a smart post office, delivering messages directly to the right address. Switches are commonly used in star topologies and improve network efficiency by reducing unnecessary data traffic.



Fig 1.8: A switch

Question arises how does it differ from hub?

Individual port on a switch makes its own domain of collision, significantly reducing collisions and improving network efficiency. Also, every node connected to a switch typically gets bandwidth which is dedicated to node, leading to good performance, making it different from hub.

4. Bridge:

Imagine a bridge connecting two separate local roads (network segments), allowing traffic (data) to flow between them only when required, as shown in Figure 1.9.

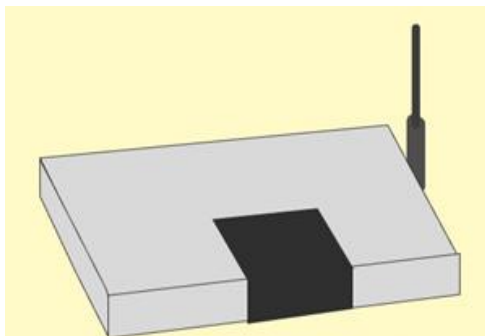


Fig 1.9: A bridge

A bridge connects one or more network sections that use the same network protocol (e.g., Ethernet). It examines the MAC addresses of incoming packets and forwards them only to the segment where the destination MAC address is situated. This helps to segment a network, reducing traffic within each segment and improving overall performance.

5. Router:

This is a crucial device that connects two or more different networks (which may use different protocols). The most common example is a home router linking your local network to the internet. Routers operate at the Network Layer and use IP addresses (logical addresses) to determine the best path for data packets to move from the source network to the target network. They manage the routing table, which stores information regarding network paths, as displayed in Figure 1.10.



Fig 1.10. A router

6. Modem (Modulator-Demodulator):

Think of a modem as a translator between the digital language of your computer and the analog signals that might travel over traditional telephone lines or cable TV lines, as shown in Figure 1.11.

A modem converts digital signals from a computer into analog signals to be transmitted over communication channels (modulation), and vice versa (demodulation).



Fig 1.11 A modem

1.4 Modes of Transmission

Transmit mode refers to the process of sending data between two devices. It is also known as the communication mode. The network is designed to allow communication between individually connected devices.

Types of transmission modes:

1. Simplex Mode

The communication is one-way, like a single-lane road. Only one of the two devices in the network can send, and the other can only receive. It allows data to be transmitted in one direction using the entire channel capacity.

For example, a keyboard provides input, while a monitor displays output.

Pros of Simplex Mode:

- Consistent and simplest mode of communication.
- Only requires one communication channel, making it cost-effective.
- No coordination needed between the devices involved.

Cons of Simplex Mode:

- One-direction communication only.
- No way to confirm whether the transmitted data has been correctly received at the destination.

2. Half-Duplex Mode

In this mode, each station can send and receive data, but not at the same time. If one device sends data, the other device can receive it, and vice versa. This is used when simultaneous two-way communication is not required.

For example, in a walkie-talkie, messages are sent one after another in either direction.

Pros of Half Duplex Mode:

- It allows bidirectional communication, where nodes can both send and receive data.
- More effective than simplex mode, as the channel can be used for both transmission and reception.
- Less expensive, as it requires only a single communication channel.

Cons of Half Duplex Mode:

- Less reliable.
- Requires coordination between communicating devices.

3. Full-Duplex Mode

Both stations can receive and transmit simultaneously. In full dual-mode, signals travel in one direction share the capacity of the link with signals travel in different directions. If communication is required in both directions. For example, a telephone network between two people via telephone line communication can speak at the same time.

Pros of Full-Duplex Mode:

- Most effective mode of communication, as both devices can operate simultaneously.
- Offers a high level of accuracy and reliability, as there is no need for error correction mechanisms.

Cons of Full-Duplex Mode:

- Most expensive mode, as it requires communication channels in both directions.
- More complex than the other two modes, as it needs two physically separate transmission paths or a division of the channel's capacity.

1.5 Transmission Media

Computer networks require media for data to travel between connected devices. Transmission media is the physical medium through which data is transferred from one device in a network to another. These media can be wired or wireless. The choice of media depends on factors such as distance, speed, and interference.

It is the physical path between the sender and the receiver — i.e., the route along which data is sent from one device to another. It is broadly categorized into different types:

Transmission Media types:**1. Guided Media**

This type of media is also called wired transmission media. Here, the signals being communicated are directed and confined within a narrow pathway using physical links. It offers very high speed and is secure in nature, making it suitable for long-distance communication.

Types of Guided Media:**i. Twisted Pair Cable:**

This consists of two discretely insulated conductor wires that are twisted together. Several pairs are assembled within a protective sheath. They are the most widely used transmission media. Twisted pair cable is classified into two categories:

a. Unshielded Twisted Pair (UTP): UTP consists of two insulated copper wires twisted around each other. This type of cable has the capability to block interference and does not rely on a physical shield for this purpose. It is used for telephone applications (see Figure 1.12).



Fig 1.12. Unshielded Twisted Pair (UTP)

Pros of Unshielded Twisted Pair:

- Inexpensive in nature.
- Smooth to connect.
- Capacity of high-speed communication.

Cons of Unshielded Twisted Pair:

- Lesser capacity and performance
- Small distance communication due to distortion.

b. Shielded Twisted Pair (STP): STP cable consists of distinct covering to block noise. Utilised in telephone lines. Shown in Figure 1.13.



Fig 1.13. Shielded Twisted Pair (STP)

Pros of Shielded Twisted Pair

- Optimised performance at an advanced data rate.
- Removes noise
- It is reasonably faster

Cons of Shielded Twisted Pair

- It is tough to manufacture and install.
- High cost involved.

ii. Coaxial Cable

It has an outer plastic covering that includes an insulation layer made of PVC and two parallel conductors, each enclosed in a separate insulated protective cover. It transmits information in two modes: Baseband mode (dedicated cable bandwidth) and Broadband mode (cable bandwidth is split into separate ranges). Coaxial cables are widely used for cable TV. Shown in Figure 1.14.



Fig 1.14. Coaxial cable

Pros of Coaxial Cable:

- Increased bandwidth.
- Easy installation.
- More tough and reliable.
- Not as much affected by noise or cross-talk or electromagnetic inference.

Cons of Coaxial Cable:

- Expensive in nature.
- To avoid any noise, it must be grounded.

iii. Optical Fiber Cable

Optical Fiber cable uses total internal reflection of light through a core made of glass. The core is enclosed by a less dense glass or plastic casing called the coating. It is used for transmitting large volumes of data. The cable can support communication in one direction or in both directions. Shown in Figure 1.15.



Fig 1.15. Optical fiber cable

Pros of Optical Fiber Cable:

- High Bandwidth and capacity.
- Lightweight in nature.
- Resistance in erosion for materials

Cons of Optical Fiber Cable:

- Tough in maintenance and installation.

2. Unguided Media

Referred to as wireless or unbounded transmission media. No physical medium is required for the transmission of signals, which are electromagnetic in nature. In this case, air serves as the medium for broadcasting signals. It is used for long-distance communication.

There are three categories of signals transmitted through unguided media:

i. Radio Waves

Radio waves are easy to produce and can penetrate through buildings. The two antennas do not need to be physically connected. Their frequency ranges from 3 kHz to 1 GHz. They are used by AM and FM radios for transmission. Shown in Figure 1.16.



Fig 1.16: Radio waves

ii. Microwaves

This is a straight-line transmission, i.e., the antennas on both sides—whether sending or receiving—need to be accurately aligned with each other. The distance covered by the signal is directly proportional to the height of the antenna. Its frequency ranges from 1 GHz to 300 GHz. These waves are used for communication by mobile phone networks. Shown in Figure 1.17.

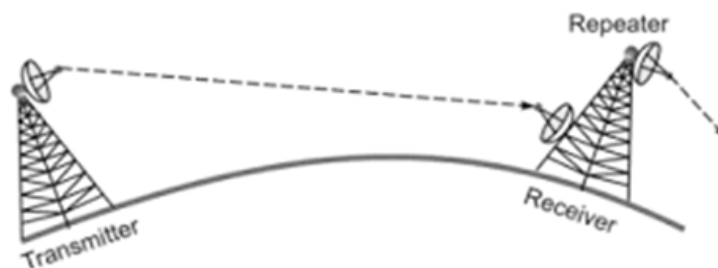


Fig 1.17. Microwaves

iii. Infrared

These are meant for short-distance communication, as shown in Figure 1.18. Obstacles cannot be penetrated by these waves. Their frequency ranges from 300 GHz to 400 THz. These waves are used in devices such as remote controls, mice, keyboards, printers, etc.

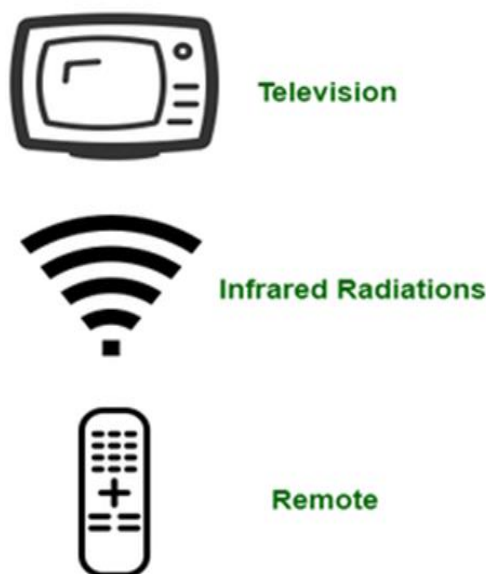


Fig 1.18. Applications of Infrared waves

1.6 Ethernet

Ethernet is like the typical set of roads and traffic rules within a town (your local area network), as shown in Figure 1.19. It is the most common way for computers and other devices to connect to each other within a confined space, such as your school's computer lab or your home network.



Fig 1.19. Ethernet

Imagine you have different houses in this town, and you want them to be able to send messages to each other. Ethernet provides the physical roads, which are usually special wires called twisted-pair cables that plug into your computer with a connector that looks a bit like a wider phone jack. These wires carry the electrical signals that represent your digital information.

But just having roads isn't enough, you need rules for how the traffic flows. Ethernet also provides these rules, called protocols. In the early days, everyone had to wait for a moment before sending a message. If two people tried to send a message at the same time, there would be a collision, and they would both have to try again later.

However, over time, Ethernet became much smarter, thanks to devices called switches. Think of a switch as a very efficient traffic controller. It learns the address (just like a house number) of each connected device and can create a direct, temporary link between the sender and the receiver—like setting up a private lane just for their message. This means multiple nodes can send and receive information at the same time without crashing into each other, making the network much faster and more efficient.

Over time, just like towns build bigger and faster roads, Ethernet has also gotten faster. We now have Fast Ethernet and Gigabit Ethernet, which are like superhighways for data, allowing us to share large files and stream videos smoothly within our local networks.

So, in simple terms, Ethernet is the most widely used and reliable technology that provides the physical connections and the basic rules for devices to communicate within a local area network. It's the fundamental way most of our computers and other gadgets connect in places like homes, schools, and offices.

Practical Activity 1.1: Identifying and Using Transmission Media and Network Tools

Materials Required:

Co-axial cable, UTP (Unshielded Twisted Pair) cable (Cat5e or Cat6), RJ45 connectors
Crimping tool, Wire stripper or cutter, LAN tester (optional)

Steps:

Step 1: Identify the Cables

- Observe and list the physical characteristics of co-axial cable and UTP cable.
- Note the inner conductor, shielding, twisting of wires, etc.

Step 2: Prepare the UTP Cable

- Use the wire stripper to carefully remove about 1 inch of the outer jacket.
- Untwist and straighten the 8 inner wires.
- Arrange the wires in the T568B color code order.

Step 3: Attach RJ45 Connector

- Insert the wires into an RJ45 connector, ensuring they reach the end.
- Insert the connector into the crimping tool and press firmly to crimp.

Step 4: Test the Cable (Optional)

- Use a LAN tester to check if the cable is working properly and wires are connected correctly.

Step 5: Repeat for Co-axial Cable (Optional Demo)

- Observe how the co-axial cable is prepared and connected using its specific connectors (like BNC).

Summary

- A computer network is a cluster of interrelated devices that share resources and data.
- Types of networks include PAN, LAN, MAN, and WAN, categorized by geographical range.
- Benefits of networking include resource sharing, communication, centralized data, increased productivity, and internet access.
- Network topologies define the structure: Bus, Star, Ring, Mesh, Tree, and Hybrid.
- Network components include nodes (devices), links (connections), NICs, and transmission media.

- Devices like routers, switches, hubs, and modems manage data flow.
- Different modes of transmission are: Simplex, Half-duplex, and Full-duplex.
- Media of transmission can be guided (cables) or unguided (wireless).
- Ethernet is the most common LAN technology, using protocols and cables for communication.

Check your progress**A. Multiple Choice Questions (MCQs)**

1. What is the full form of LAN?
 - a) Local Access Network
 - b) Large Area Network
 - c) Local Area Network
 - d) Line Access Network
2. Which topology uses a backbone wire?
 - a) Ring
 - b) Star
 - c) Mesh
 - d) Bus
3. A _____ connects multiple devices but sends data to all.
 - a) Switch
 - b) Repeater
 - c) Hub
 - d) Bridge
4. What is the function of a modem?
 - a) Converts analog to digital and vice versa
 - b) Stores data
 - c) Routes data
 - d) Filters data
5. Which device works at the Data Link Layer and performs MAC address learning?
 - a) Repeater
 - b) Hub
 - c) Switch
 - d) Modem
6. Which transmission medium has the highest bandwidth?
 - a) Twisted pair
 - b) Coaxial
 - c) Optical fibre
 - d) Radio waves
7. Which network has the smallest range?
 - a) MAN
 - b) WAN
 - c) PAN
 - d) LAN
8. In which topology does data pass in one direction in a loop?
 - a) Mesh
 - b) Ring
 - c) Star
 - d) Tree

9. Which layer of the OSI model is responsible for physical transmission?
 - a) Transport
 - b) Network
 - c) Data Link
 - d) Physical
10. Wi-Fi operates using which type of transmission media?
 - a) Coaxial
 - b) Fiber optic
 - c) Guided
 - d) Unguided

B. Fill in the Blanks

1. A _____ is a node that links two different networks.
2. A _____ transmits data only to the device it is meant for.
3. PAN stands for _____.
4. _____ topology joins all nodes to a central hub.
5. _____ cable uses total internal reflection to transmit data.
6. In _____ mode, transmission is only in one direction.
7. A _____ is a sole address given to a network device's hardware.
8. _____ media uses physical cables for data transmission.
9. The full form of NIC is _____.
10. The logical or physical structure of a network is called _____.

C. True or False

1. A switch sends data to all connected devices.
2. Bus topology is robust and fault-tolerant.
3. UTP cables are cheaper than STP cables.
4. Bluetooth is used for long-distance data transfer.
5. A bridge connects different network segments.
6. Star topology is the most fault-tolerant.
7. A repeater strengthens weak signals in a network.
8. Mesh topology has no dedicated links between nodes.
9. Radio waves can pass through walls.
10. Ethernet is a wireless network protocol.

D. Short Answer Questions

1. Define a computer network.
2. What is the main advantage of using a network?
3. Name two guided transmission media.
4. Explain the function of a switch.
5. Differentiate between LAN and WAN.
6. Define network topology.
7. What is the use of a repeater?
8. Differentiate between Simplex and Full-Duplex transmission.
9. What does MAC address stand for?
10. What is the purpose of a modem?

Answer Key**A. Multiple Choice Questions**

1. c), 2. d), 3. c), 4. a), 5. c), 6. c), 7. c), 8. b), 9. d), 10. d),

B. Fill in the blanks:

1. Router, 2. Switch, 3. Personal Area Network, 4. Star, 5. Fiber optic, 6. Simplex, 7. MAC address, 8. Wired, 9. Network Interface Card, 10. Topology

C. True or False

1. False, 2. False, 3. True, 4. False, 5. True, 6. True, 7. True, 8. False, 9. True, 10. False

Chapter-2**Protocol Models and Ip addressing**

One day, Ravi asked his sister Meera, “How does my message reach your phone so fast?” Meera smiled and said, “It’s like sending a letter! The IP address is like your home address—it tells the internet where to deliver the message. And the protocols are the rules the internet follows to send it safely, step by step.” She added, “Just like a post office follows a process to send letters, the TCP/IP model follows layers to deliver data correctly.” Ravi nodded, “So the internet speaks its own language, and addresses help it know where to go! The conversation between Ravi and Meera.



In this chapter, we will get to know about the OSI model, TCP/IP model, UDP and IP addressing. For example, Computers in a network talk using rules called protocols, like people use language. Each computer has an IP address, like a home address, so messages know where to go. Protocols like TCP/IP make sure data is sent properly, while UDP sends quick messages without waiting for a reply. Ports help different data—like websites or videos—reach the right place on your computer.

2.1 The OSI model

OSI stands for Open Systems Interconnection. It is a basic model that defines the process through which information from various software applications in a computer travels via a physical medium to the software application in another computer. Think of it as a blueprint that helps everyone involved in network communication, from hardware manufacturers to software developers, to understand the way data should be handled at each stage.

The OSI model comprises seven layers, and each layer performs a specific network function. The OSI model was established by the International Organization for Standardization (ISO) in 1984, and it is now considered an architectural model for inter-computer communications.

The OSI model splits the entire task into seven smaller, manageable tasks. Each layer performs a particular function and is self-contained, so the task allotted to each layer can be accomplished independently of the others. The OSI Model Features are shown in Figure 2.1.

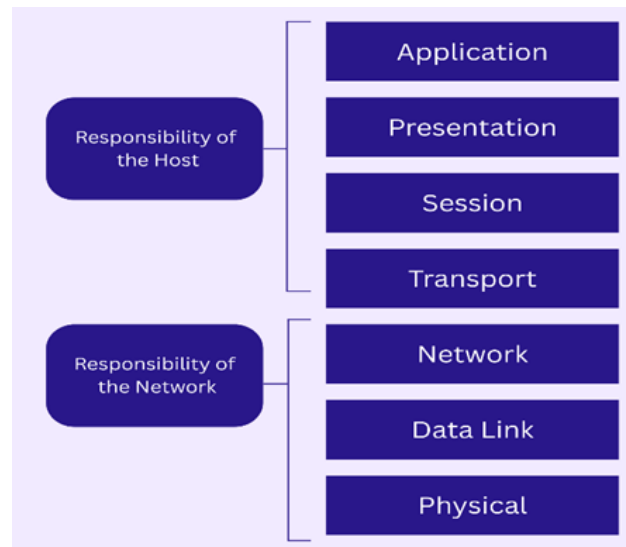


Fig 2.1. OSI model features

This model is categorized into two major layers: upper layers and lower layers.

1. The upper layer of this model primarily deals with issues associated with the application and is executed only in software. The application layer is adjacent to the end user. Both the end user and the application layer interact with the software applications. An upper layer refers to the layer just above another layer.
2. The lower layer of this model deals with data transport issues. The data link layer and the physical layer are implemented in both hardware and software. This layer is the lowest in the OSI model and is adjacent to the physical medium. The physical layer is mainly responsible for placing the information onto the physical medium.

2.1. Layers of OSI Model

The OSI model has seven layers. Individual layer performs various specific operations, which are mentioned below:

1. Physical Layer
2. Data-Link Layer
3. Network Layer
4. Transport Layer
5. Session Layer
6. Presentation Layer
7. Application Layer

2.2 TCP/IP model

Imagine the internet and many computer networks as a large postal service for digital information. This model acts like a set of steps and rules that everyone follows to send and receive these digital letters (data), as shown in Figure 2.2.

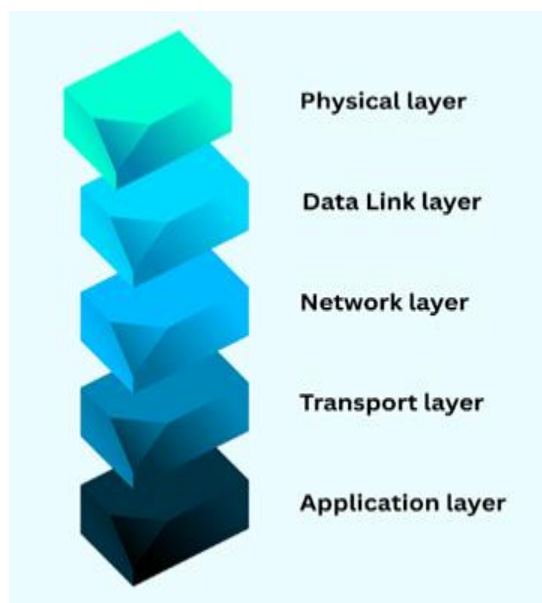


Fig 2.2. TCP/IP model

This model is a central framework for computer networking. Its full form is Transmission Control Protocol/Internet Protocol (TCP/IP), which comprises the fundamental protocols of the Internet. It defines how the network performs reliable operations between various devices and how data is transmitted to them. It consists of five layers:

1. Physical layer
2. Data Link Layer
3. Network Layer
4. Transport Layer
5. Application Layer

Each individual layer performs specific operations that help accomplish various critical functions required for communication between networks.

The Department of Defense (DoD) designed the TCP/IP model in the 1970s, and it closely resembles the OSI model.

What Does TCP/IP Do?

The primary task of this protocol is to transfer computer information from one device to another. Its main objective is to ensure the data is accurate so that the recipient receives exactly what the sender transmitted. To guarantee that each message reaches its final destination, the TCP/IP model divides the data into packets and reassembles them at the receiving end. This protocol helps maintain data accuracy during transmission from one end to the other. When network technologies and physical media are required, the TCP/IP model can be used.

Layers of TCP/IP Model

1. Physical Layer

This layer is the very basis of network communication, dealing with the raw transmission of data as electrical, light, or radio signals across a physical medium. Think of it as the actual wires (like Ethernet cables), optical fiber, or airwaves used for wireless communication.

The Physical Layer defines the physical characteristics of the network, such as voltage levels, timing of signals, and the use of physical connectors. It is concerned with how a single bit of data—a 0 or a 1—is represented and transmitted. The Physical Layer ensures that signals are correctly transmitted and received, setting the stage for the layers above to handle the organization and interpretation of this raw data. It doesn't understand the meaning of the bits; it simply moves them from one point to another.

2. Data Link Layer

This layer represents a collection of applications that require network communication. It is responsible for generating data and initiating connection requests. The network access layer is managed by the receiving process at the destination and begins at the sender's end, which handles the transmission of data while ensuring the recipient successfully receives it.

3. Network Layer

This layer corresponds to the operations of the Network Layer in the OSI model. It defines the protocols involved in the logical transmission of data across the entire network.

IP stands for Internet Protocol and is responsible for delivering packets from the source host to the destination host by using the IP address found in the packet header.

TCP, on the other hand, is a protocol that operates at the Transport Layer of the TCP/IP model. Its primary function is to ensure the reliable and orderly delivery of data between applications running on different computers.

4. Transport Layer

This layer's protocols exchange data, receive acknowledgments, and retransmit missing packets to ensure that packets arrive in order and without errors.

5. Application Layer

This layer in the TCP/IP model is responsible for end-to-end communication and the error-free transfer of data. It shields upper-layer applications from underlying complexities.

TCP/IP Protocols

- **IP:** This stands for Internet Protocol and is responsible for examining the IP address in the packet header and delivering the packet from the source host to the target host. There are two versions of IP: IPv4 and IPv6.
- **ICMP:** ICMP stands for Internet Control Message Protocol. It is encapsulated within an IP datagram and provides information about network issues.
- **ARP:** ARP stands for Address Resolution Protocol. Its task is to find the hardware address of a device from a known IP address. The various categories of ARP are: Reverse ARP, Proxy ARP, Gratuitous ARP, and Reverse ARP.
- **TCP:** Applications can interact as if they were physically connected in circuits using TCP. TCP sends data in a way that resembles sending characters in a continuous stream, rather than individual packets.
- **UDP:** Delivery of datagram services is provided by UDP, another transport layer protocol. The connection between the sending and receiving hosts is not verified by UDP. Applications that transmit only a small amount of data use UDP instead of TCP to avoid the overhead of establishing and verifying connections.
- **HTTP and HTTPS:** HTTP stands for Hypertext Transfer Protocol. It is used on the World Wide Web to manage communication between a web browser and a server. HTTPS stands for HTTP Secure, which is a combination of HTTP and SSL (Secure Sockets Layer). It is designed to be secure and efficient, especially when your browser is used to fill out forms, register, authenticate, or perform banking transactions.
- **SSH:** SSH stands for Secure Shell. It is a terminal emulation protocol similar to Telnet, but it is preferred due to its ability to maintain an encrypted and secure connection.
- **NTP:** NTP stands for Network Time Protocol. It is used to synchronize the system clock with standard time sources, and is especially useful in time-sensitive applications such as banking.

Assignment 2.1:

- List down all the Layers of TCP/IP Model.
- Describe OSI model features.

2.3 Port numbers

Imagine your computer as a bustling apartment building with many different applications (like your web browser or email app) living inside. Just like a postal address gets mail to the building, an IP address directs network data to your computer. But how does the data know which application it's intended for?

That's where port numbers come into play — they act like apartment numbers that direct the data to the correct application within your computer.

Once a data packet arrives at a specific device, the operating system needs to know which application should receive that data. Think of an IP address as the address of an apartment building, and the port number as the specific apartment number within it.

Port numbers are 16-bit integers, ranging from 0 to 65,535.

TCP vs UDP: Two Ways to Transport Data

As discussed earlier, TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are the two main protocols operating at the Transport Layer of the TCP/IP model, each offering a different approach to data delivery.

TCP is connection-oriented and provides reliable, ordered, and error-checked delivery. Before data transmission begins, a connection is established between the sender and the receiver.

UDP, on the other hand, is connectionless and offers a faster but less reliable method of data transfer. It does not establish a connection before transmitting data and does not guarantee error checking, delivery, or order. Examples include online gaming and video streaming.

Basis	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
Type of Service	It is protocol which is connection-based	This is a Datagram based.
Reliability	More reliable in nature	Less reliable in data delivery
Speed	Slower than UDP.	Faster, simpler, and more effective
Broadcasting	TCP doesn't provision Broadcasting.	UDP provisions Broadcasting.
Protocols	<u>HTTP</u> , <u>HTTPS</u> , <u>FTP</u> , <u>SMTP</u> and <u>Telnet</u> protocols use it.	<u>DNS</u> , <u>DHCP</u> , <u>TFTP</u> , <u>SNMP</u> , <u>RIP</u> , and <u>VoIP</u> use it.
Applications	Used in military operations, web browsing and emails	Used in all types of streaming.

2.4 IP Addressing

Just like all homes need clear street addresses, email delivery and all devices linked to a network in the Internet require a clear identifier called an IP address (Internet Protocol Address). This address enables devices to communicate with each other by providing location and identification capabilities within the network. IP address classes were determined based on the first few bits of the address. The main classes are A, B, C, D, and E.

These schemes are used to define, assign, and manage IP addresses on the network.

IP addressing are divided into two categories:

Static IP Addressing: In this scheme, IP addresses are manually configured by the network administrator for each device.

Dynamic IP Addressing: In this scheme, IP addresses are automatically assigned by the server using a protocol called DHCP (Dynamic Host Configuration Protocol).

An IP address (Internet Protocol Version 4 address) serves a similar purpose for Internet protocols, particularly for devices connected to networks that use the Internet. It is a logical, numerical label that can be assigned, identified, and located for all involved. (Shown in Figure 2.3.).

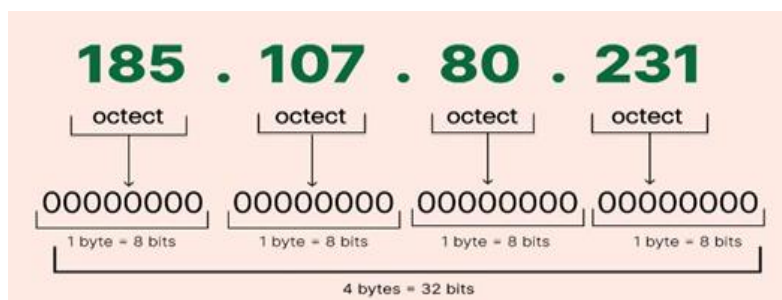


Fig 2.3. IPv4 Address

These IP addresses are important for routing data packets across your network. When you request a website, the computer's IP address acts as the sender's address, and the web server's IP address is the destination address. Routers along the way use these addresses to determine the best path for your data. Subnet masks work in a similar way on a network. This is a 32-bit number used in IP networks that are divided into small, manageable subnets.

Subnets help organize your network logically, improve security, and reduce network traffic by keeping local communications within the subnet. Network administrators can create separate broadcast domains, meaning that broadcast messages sent within a subnet are not automatically sent to other subnets.

Think of it as a clear serial number that is permanently etched on the factory network card. MAC addresses are primarily used for communication within a local network (LAN). If one device wants to transmit data to another node on the same LAN, it uses the MAC address of the destination device to ensure that the data frame is delivered to the correct physical hardware.

Practical Activity 2.1: Transmission of data using CLI tools:

Materials Required: Cisco Packet Tracer installed, 1 Router (e.g., 2911), 2 PCs (PC16 & PC17), 2 Copper Straight-Through Cables, Basic CLI knowledge (Router configuration)

Step 1: Open Packet Tracer.

Step 2: Add Devices:

- o from the device panel at the bottom left, select "End Devices." Drag and drop two PC devices onto the logical workspace. Name them PC16 and PC17 (you can rename the PCs to any name of your choice by simply double-clicking on their current names).
- o from the device panel, select "Network Devices" and then choose "Routers." Drag and drop a router (e.g., a 2911 Router) onto the workspace, placing it between the two PCs.

Step 3: Connect the Devices:

- o Select "Connections"
- o Choose "Copper Straight-Through" cable
- o Click on PC16, select its "FastEthernet0/0" interface, and then click on the router.

Step 4: Choose the "GigabitEthernet0/0" interface (or any other available Ethernet interface).

- o Connect PC17 to the router using another "Copper Straight-Through" cable, connecting its "FastEthernet0/0" interface to a different available Ethernet interface on the router (e.g., "GigabitEthernet0/1").

Practical Activity 2.2: Configure IP Addresses:

2.2.1. We need to allocate IP addresses to the PCs and the router interfaces so they can communicate.

- Configure PC16:
 - o Press PC16 to open its configuration window.
 - o Go to the "Desktop" tab.
 - o Click on "IP Configuration."

- o Enter the following details (you can adjust the IP range if you prefer):
 IP Address: 192.168.1.10
 Subnet Mask: 255.255.255.0 (This will auto-populate)
 Default Gateway: 192.168.1.1 (This will be the IP address of the router's interface)
- o Close the configuration window.
- Configure PC17:
 - o Click on PC17 to open its configuration window.
 - o Go to the "Desktop" tab.
 - o Click on "IP Configuration."
 - o Enter the following details:
 IP Address: 192.168.1.20
 Subnet Mask: 255.255.255.0
 Default Gateway: 192.168.1.1
 - o Close the configuration window.
- Configure the Router Interfaces:
 - o Click on the Router to open its configuration window.
 - o Go to the "CLI" tab (Command Line Interface).
 - o You'll likely be in the initial setup dialog. Type no and press Enter to arrive at the privileged EXEC mode.
 - o Enter the following commands, pressing Enter after each line:
 - o enable
 - o configure terminal
 - o interface GigabitEthernet0/0
 - o ip address 192.168.1.1 255.255.255.0
 - o no shutdown
 - o exit
 - o interface GigabitEthernet0/1
 - o ip address 192.168.1.2 255.255.255.0
 - o no shutdown
 - o exit
 - o endwrite memory

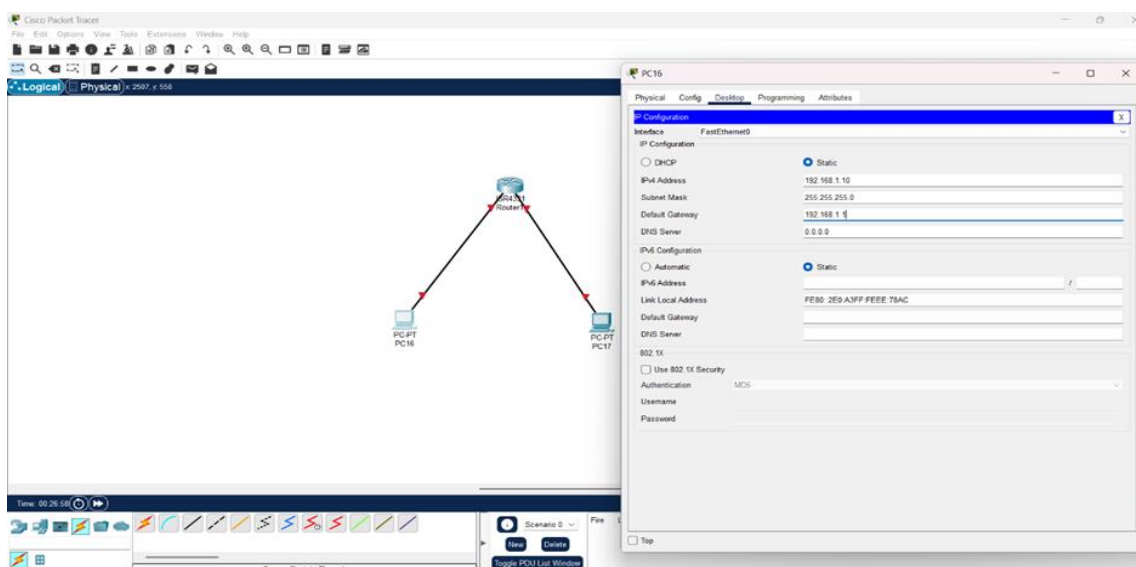


Fig 2.4. IP configuration of PC16

Below, Figure 2.5 shows the IP configuration of PC17.

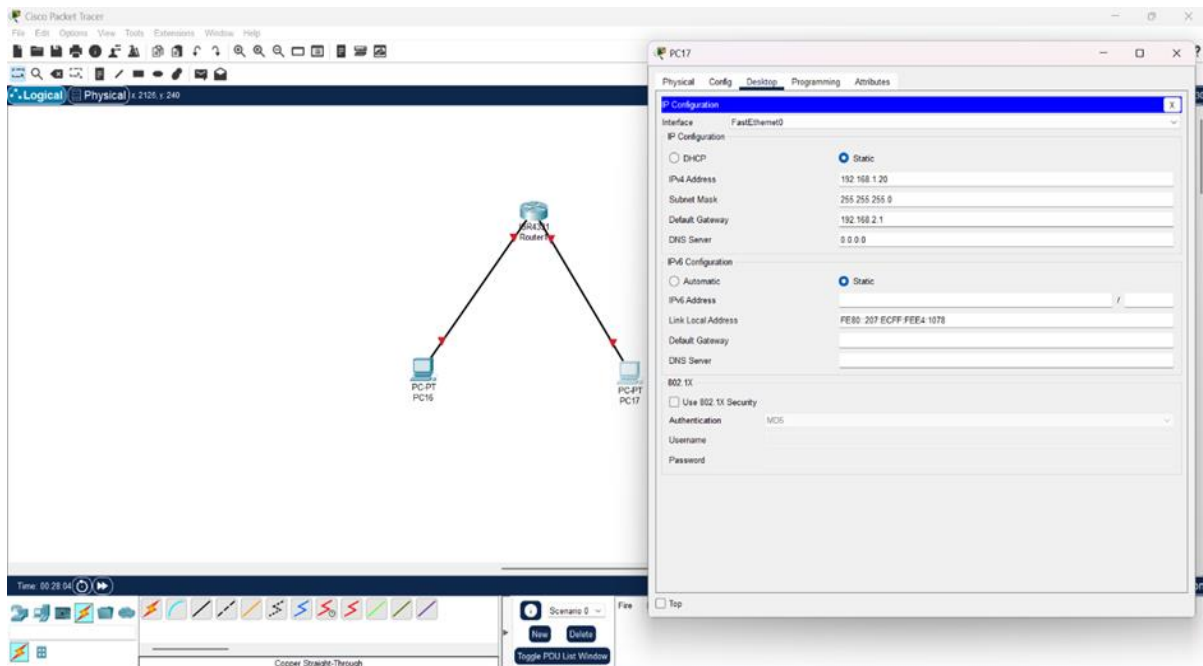


Fig 2.5. IP configuration of PC17

Below, Configuring of Router is shown in Figure 2.6.

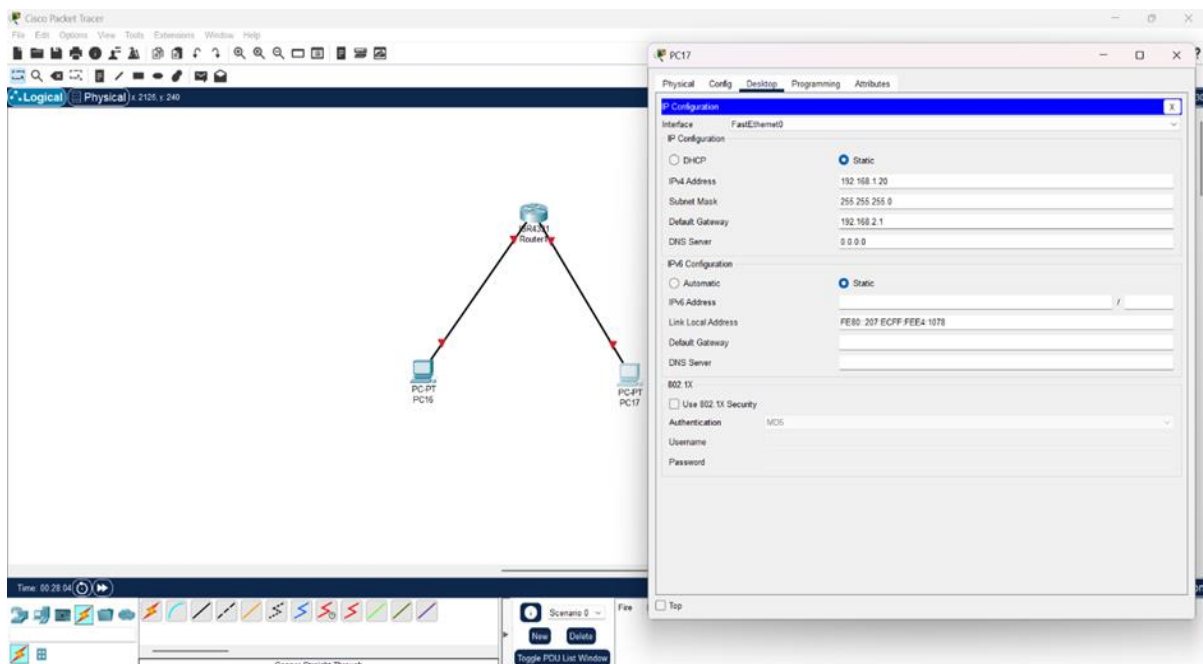
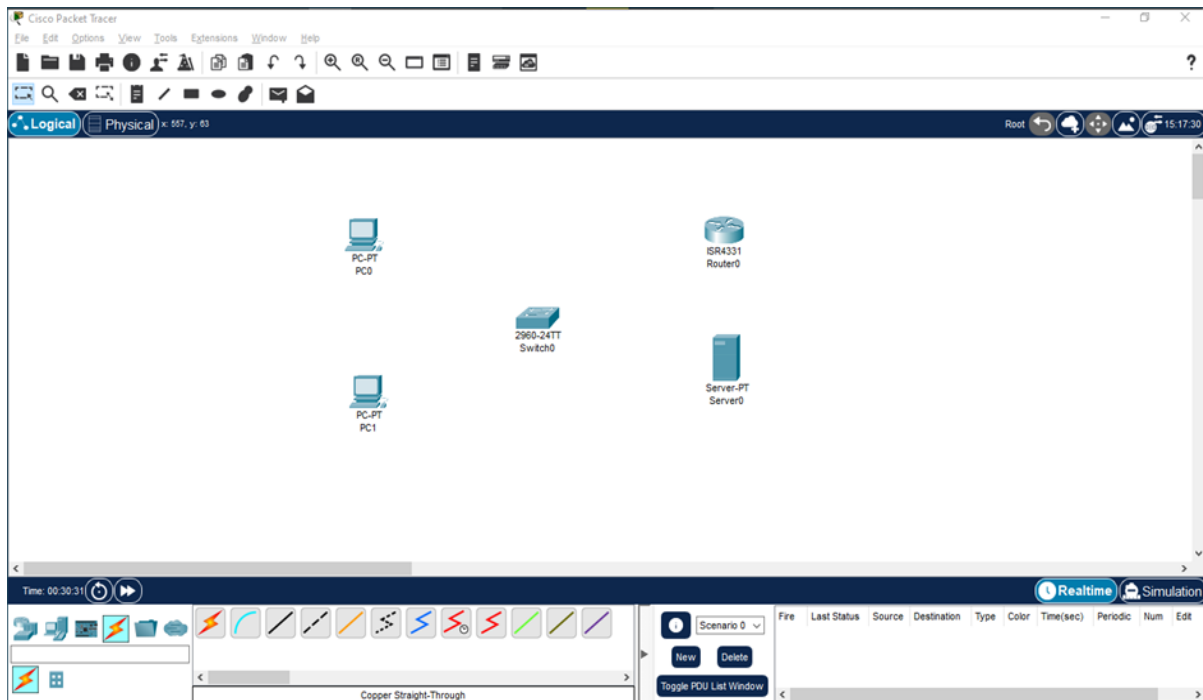


Fig 2.6 Configuring of Router

2.2.2. Configure the IP address manually in Windows and Linux for a computer connected to a LAN, and configure various network parameters in a web browser for the same computer:

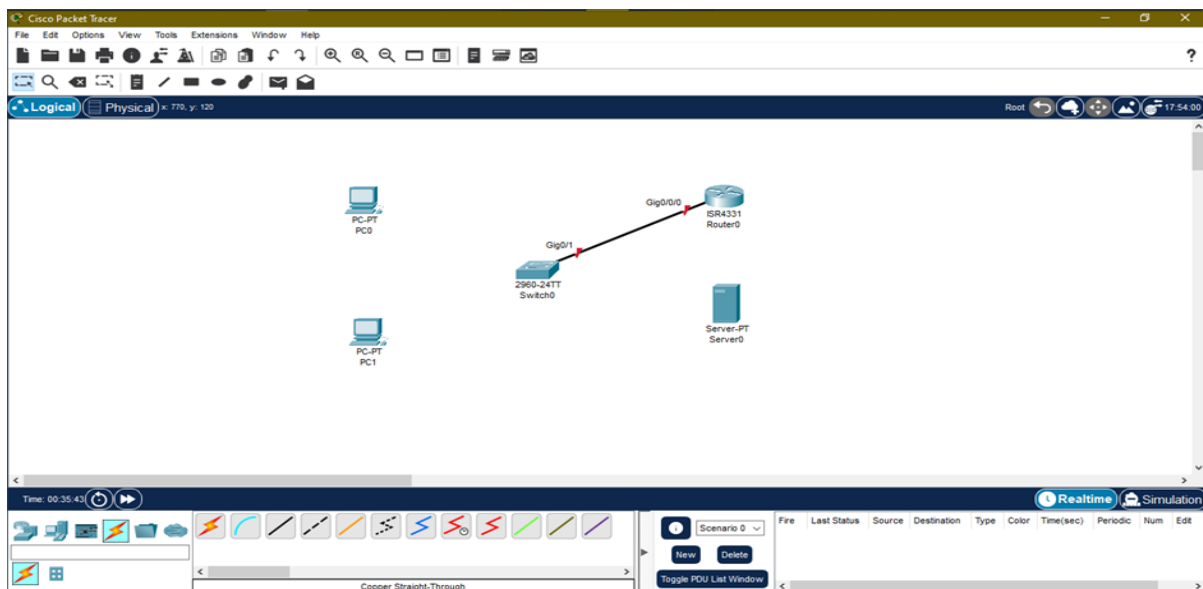
Step 1: Select the devices from the device panel. Here, we choose one PC for Windows, one PC for Linux, one server, one router, and one switch.

PC0 is designated as the Windows PC, while PC1 is designated as the Linux PC.

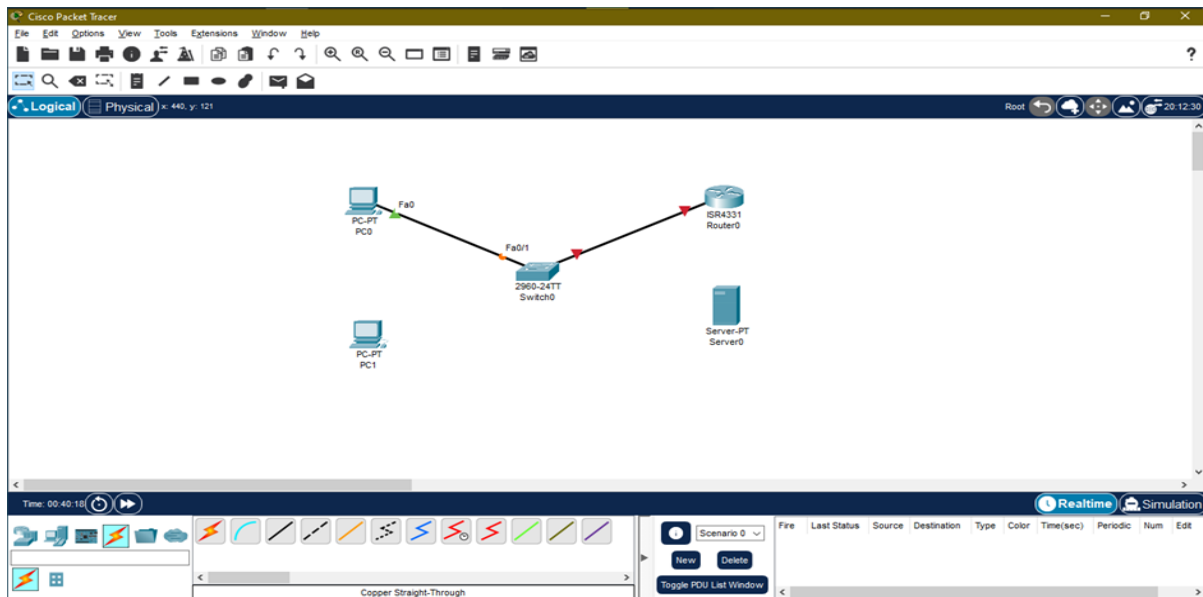


Step 2: Connect the router to the switch using a straight-through copper cable.

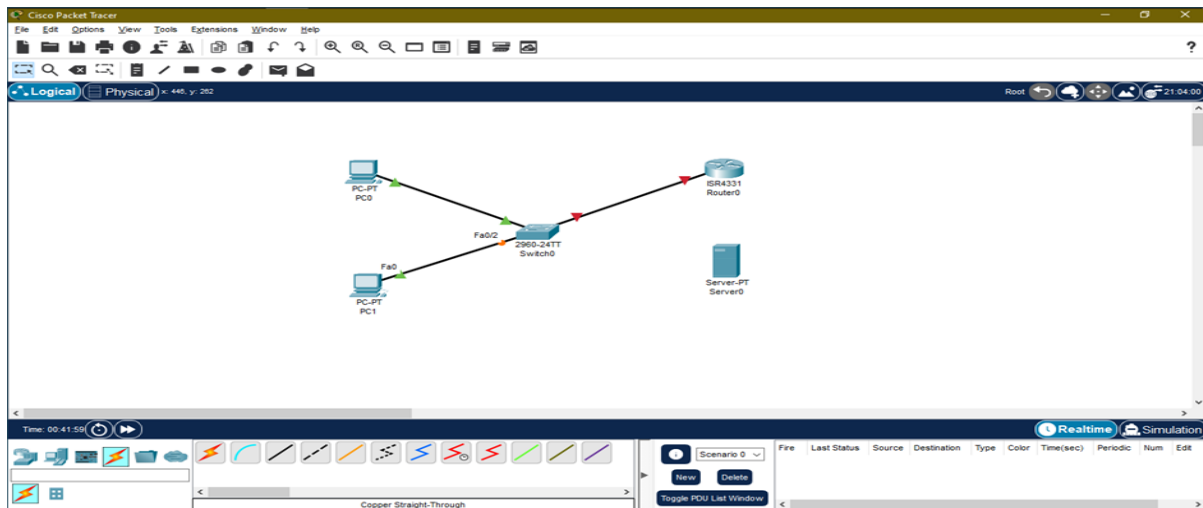
Connect the router's GigabitEthernet0/0 interface to the switch's GigabitEthernet0/1 interface.



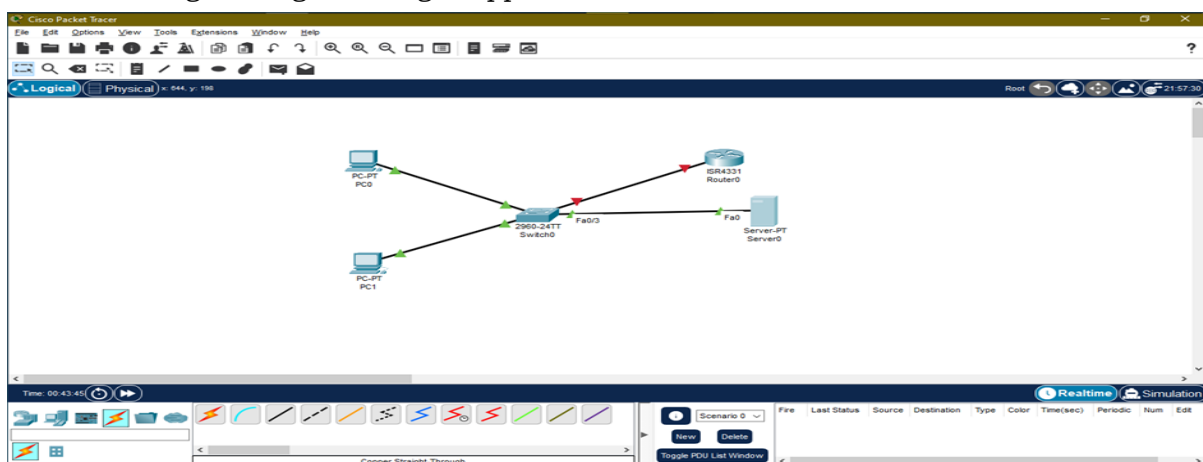
Step 3: Connect PC0's FastEthernet0 interface to the switch's FastEthernet0/1 interface using a straight-through copper cable.



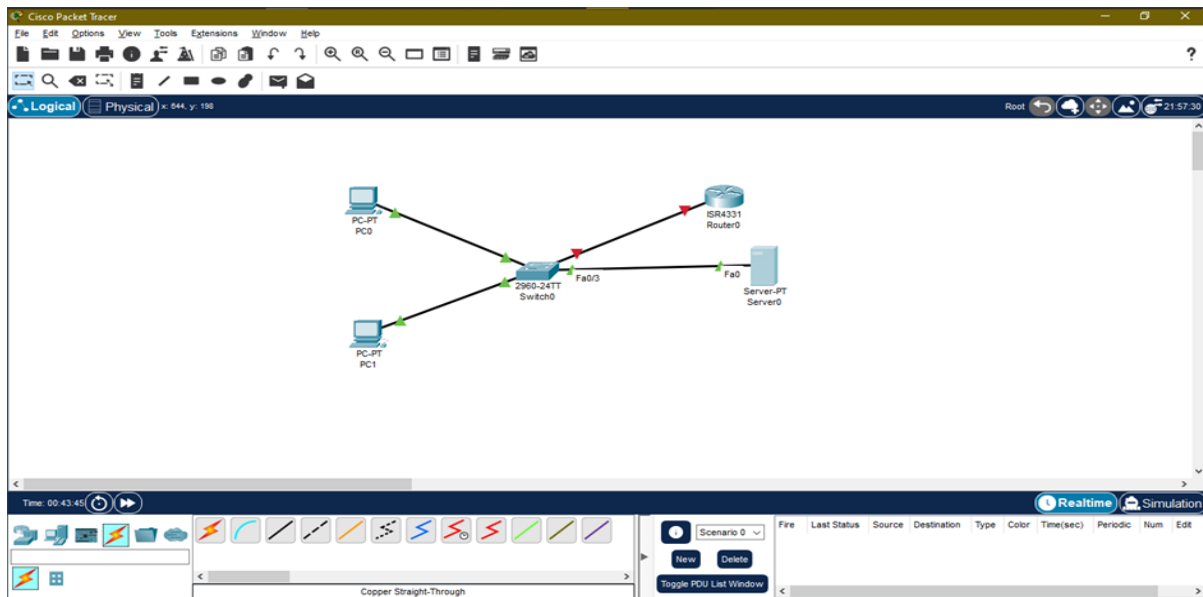
Step 4: Connect PC1's FastEthernet0 interface to the switch's FastEthernet0/2 interface using a straight-through copper cable.



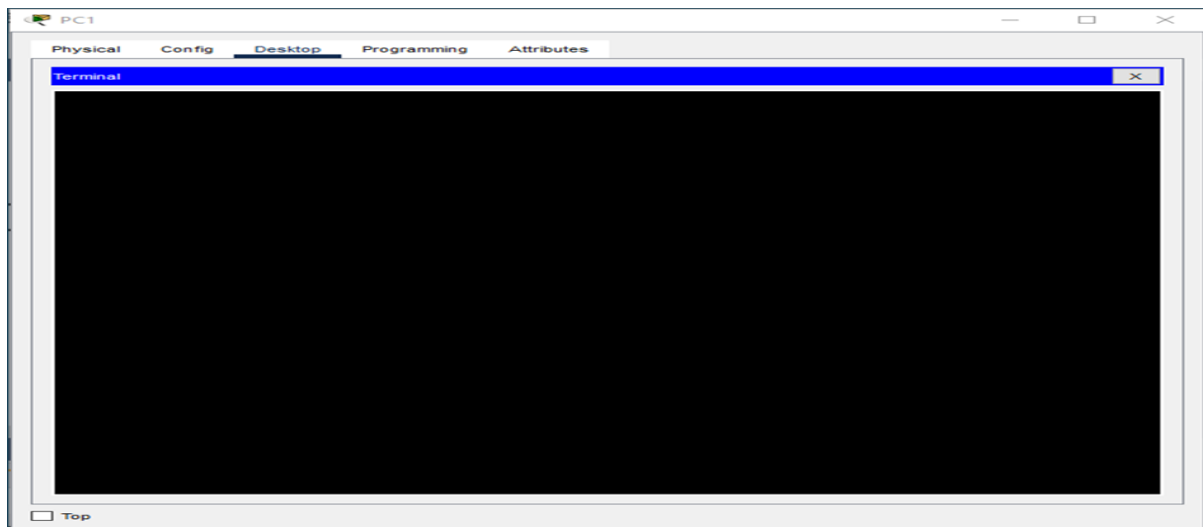
Step 5: Connect the server's FastEthernet0 interface to the switch's FastEthernet0/3 interface using a straight-through copper cable.



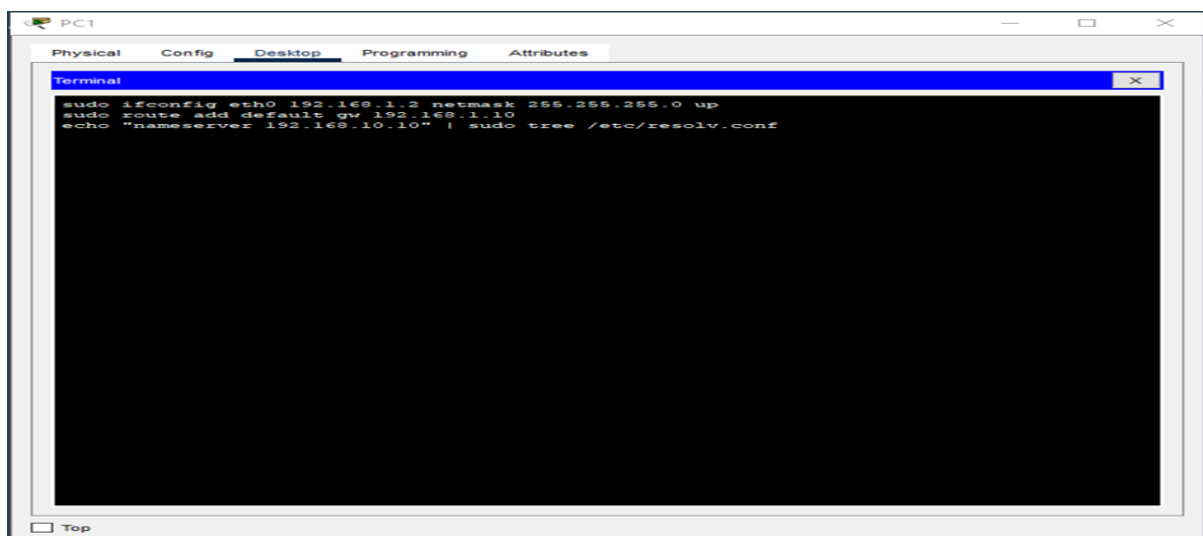
Step 6: Click on PC0 and navigate to the Desktop tab. Select IP Configuration to assign the IP address. Enter the IP address, subnet mask, default gateway, and DNS server.



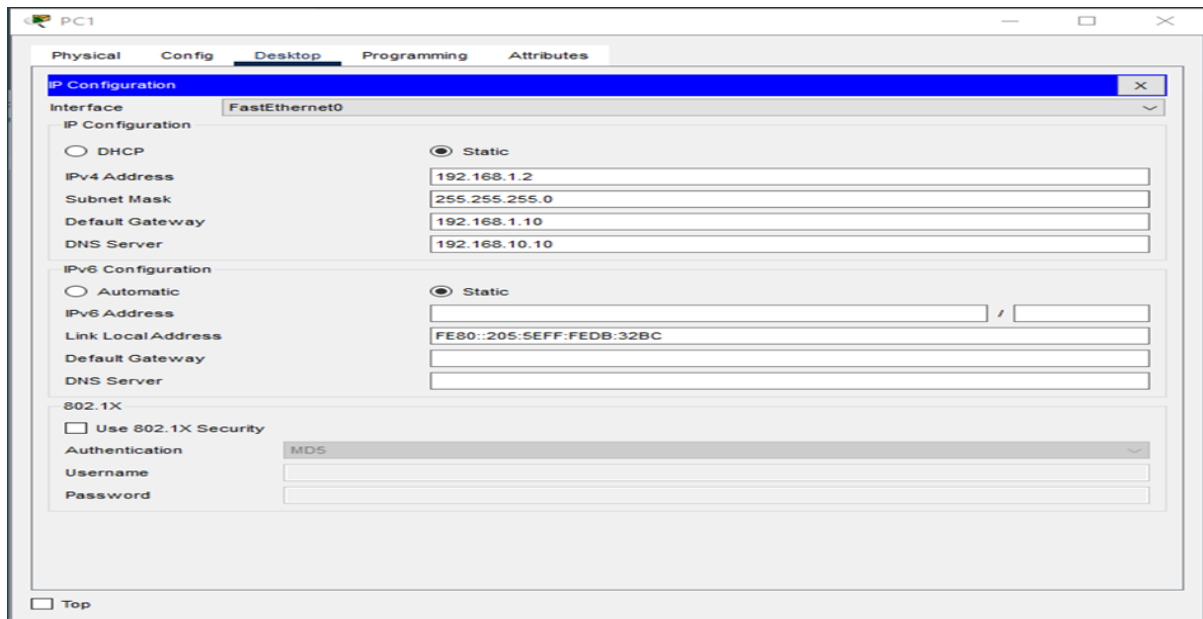
Step 7: Click on PC1 and navigate to the Desktop tab. Select the Terminal icon to open the command-line interface.



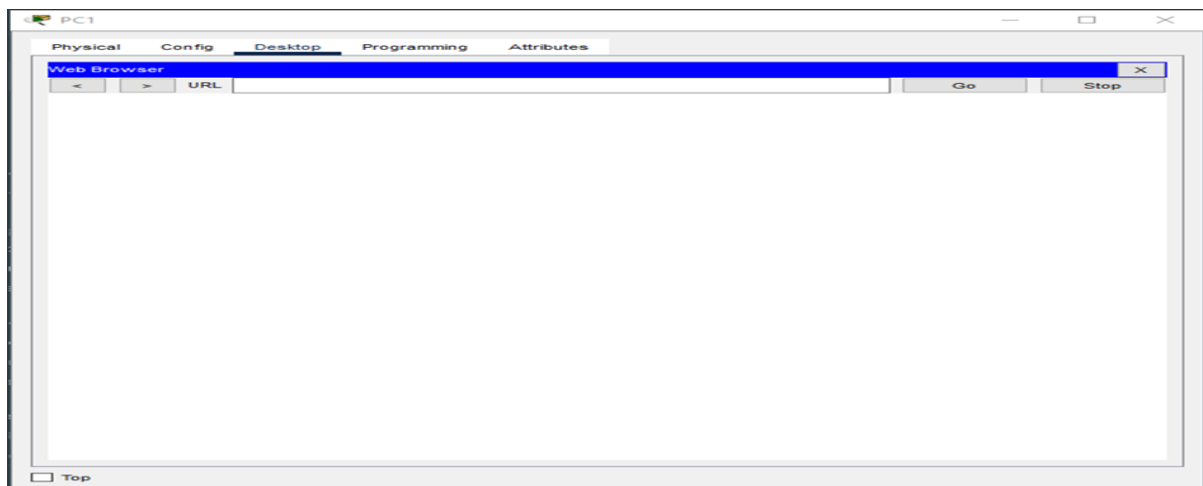
Here, we will enter commands similar to those used in Linux.



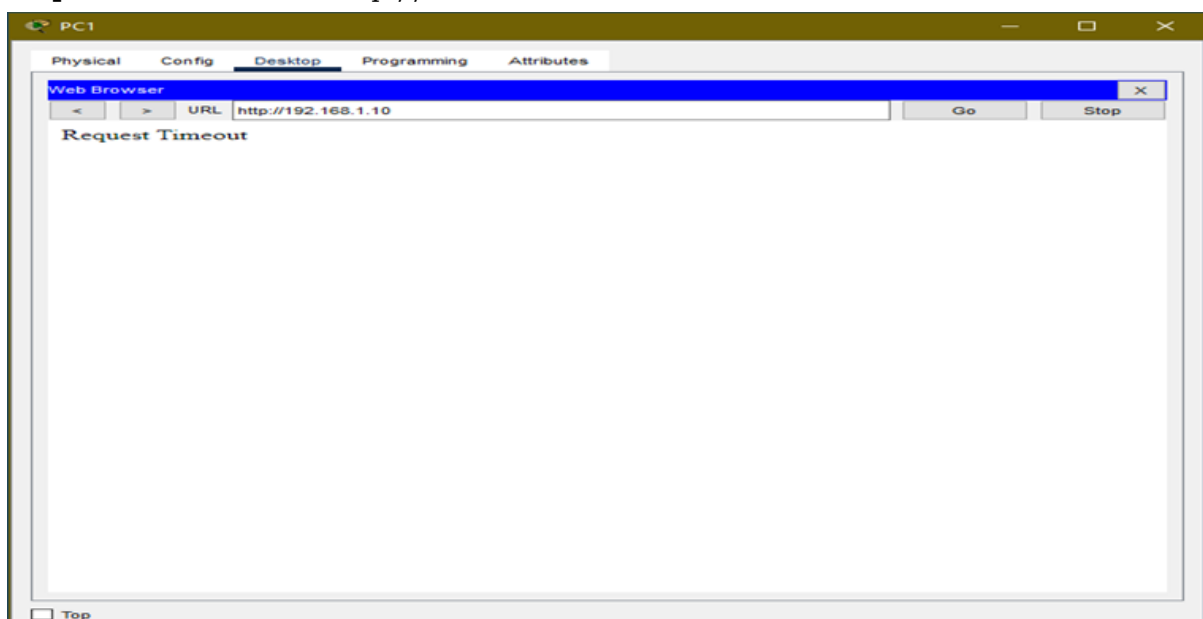
And we will verify this in the IP configuration.

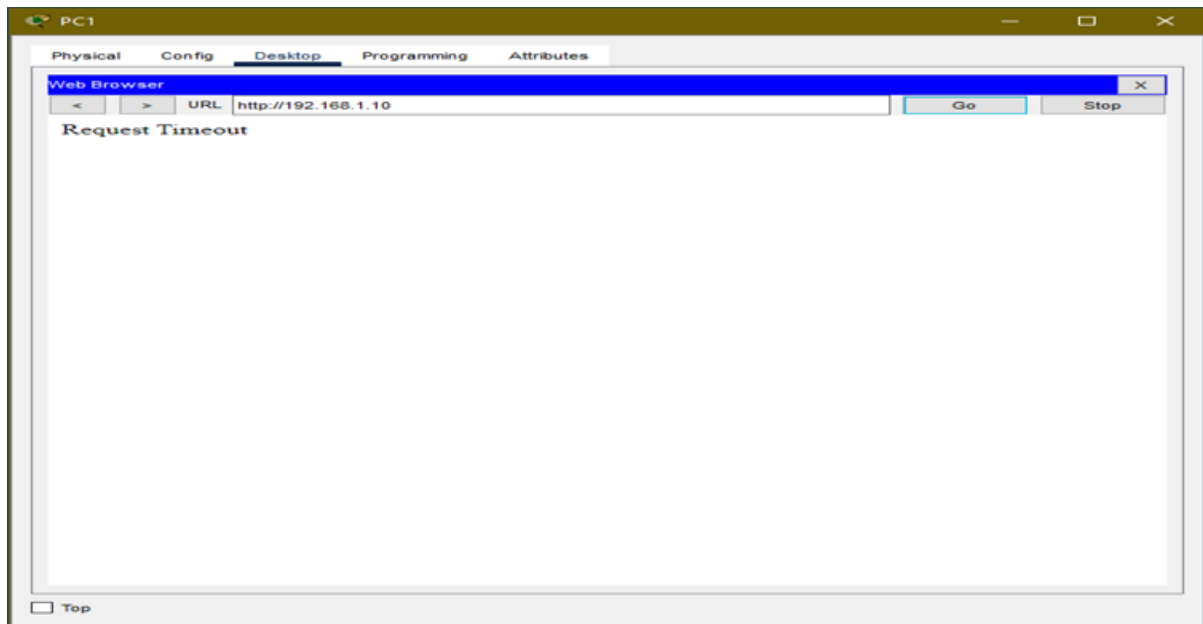


Step 8: Select any PC; we will use the Linux PC1. Navigate to the desktop tab and open the web browser.



Step 9: Enter URL here. <http://192.168.1.10> and click on the Go button.





Practical Activity 2.3: Demonstrations:

2.3.1. Ip address to class mapping:

1. Open Packet Tracer: Launch the Packet Tracer application.
2. Add PCs to the Workspace:

From the "End Devices" panel (bottom left), drag and drop three "PC" devices onto the logical workspace. Let us name them PC0, PC1, and PC2.

- Configure IP Addresses for Different Classes:
 - o PC0 (Class A):
 - Click on PC0 to open its configuration window.
 - Go to the Desktop tab.
 - Click on IP Configuration.
 - Select the Static radio button.
 - Enter the following:
 - IP Address: 10.1.1.10
 - Observe the Subnet Mask field. Packet Tracer will automatically fill in the default subnet mask for a Class A address, which is 255.0.0.0.
 - Close the "IP Configuration" and PC0's configuration windows.
 - o PC1 (Class B):
 - Click on PC1 to open its configuration window.
 - Go to the Desktop tab.
 - Click on IP Configuration.
 - Select the Static radio button.
 - Enter the following:
 - IP Address: 172.16.1.10
 - Observe the Subnet Mask field. Packet Tracer will automatically fill in the default subnet mask for a Class B address, which is 255.255.0.0.
 - Close the "IP Configuration" and PC1's configuration windows.
 - o PC2 (Class C):
 - Click on PC2 to open its configuration window.

Go to the Desktop tab.

Click on IP Configuration.

Select the Static radio button.

Enter the following:

- IP Address: 192.168.1.10
- Observe the Subnet Mask field. Packet Tracer will automatically fill in the default subnet mask for a Class C address, which is 255.255.255.0.
- Close the "IP Configuration" and PC2's configuration windows.

Explanation of IP Address to Class Mapping:

- Class A (1.0.0.0 - 126.255.255.255): The first octet ranges from 1 to 126. The default subnet mask is 255.0.0.0, meaning the first octet identifies the network, and the remaining three octets identify the host.
- Class B (128.0.0.0 - 191.255.255.255): The first octet ranges from 128 to 191. The default subnet mask is 255.255.0.0, meaning the first two octets identify the network, and the remaining two octets identify the host.
- Class C (192.0.0.0 - 223.255.255.255): The first octet ranges from 192 to 223. The default subnet mask is 255.255.255.0, meaning the first three octets identify the network, and the last octet identifies the host.

By observing the default subnet masks assigned by Packet Tracer based on the IP address you entered, we get the basic concept of IP address to class mapping. Figure 2.7 shows the IP configuration of PC0.

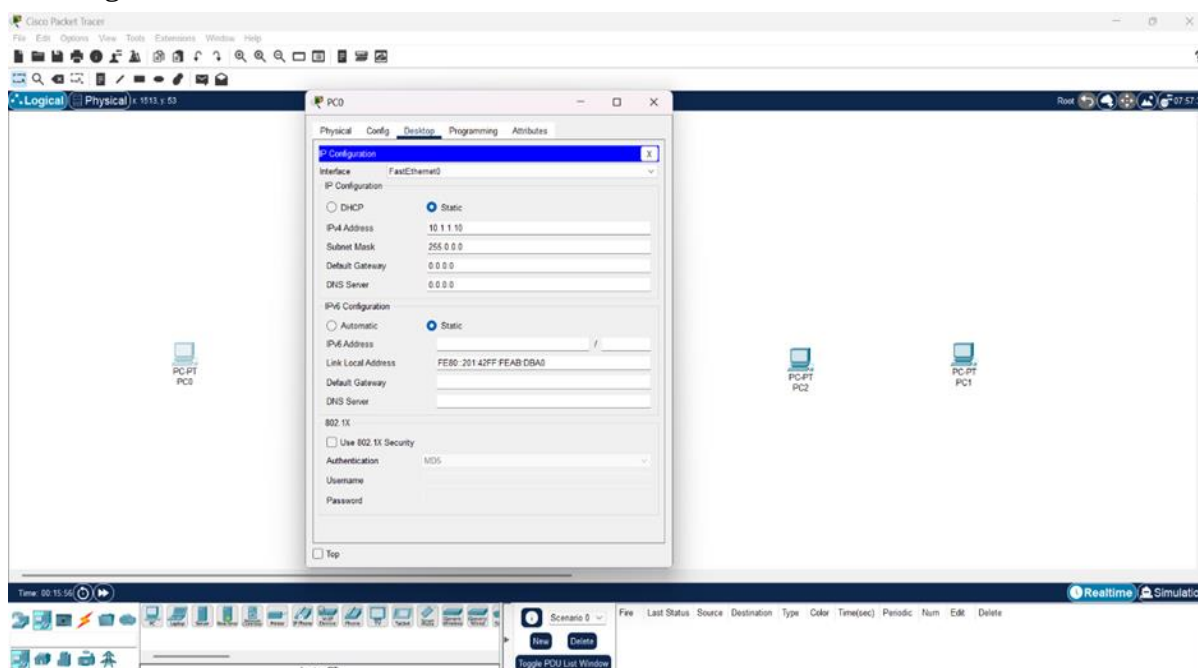


Fig 2.7. IP configuration of PC0

The IP configuration of PC2 as shown below in Figure 2.8.

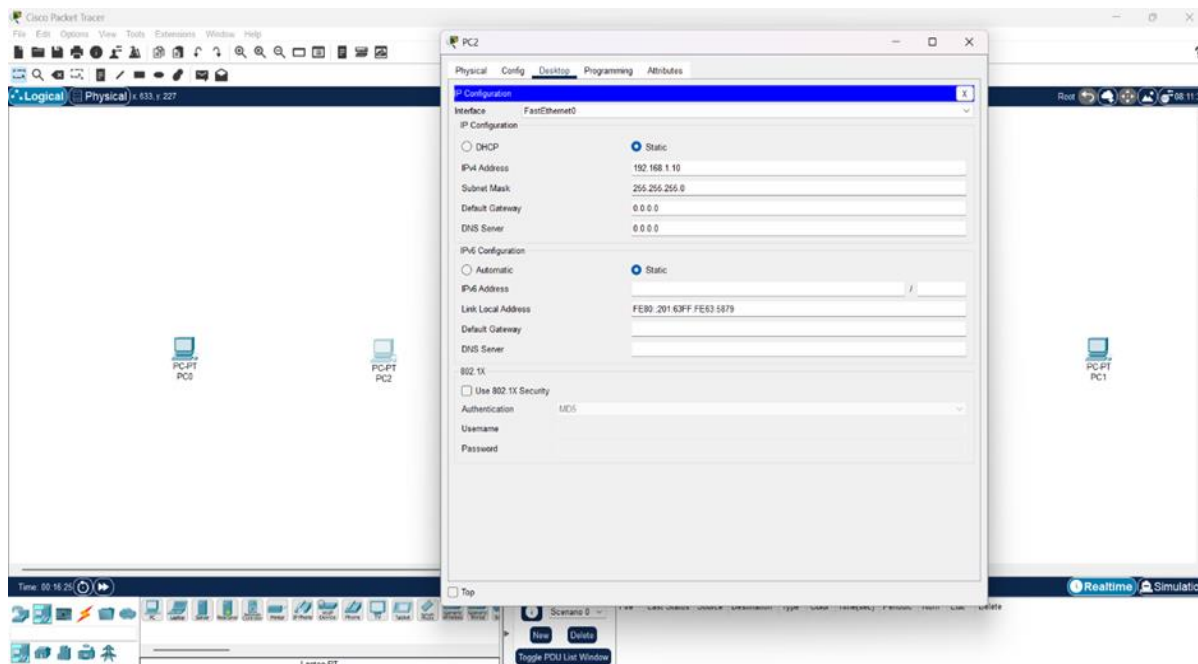


Fig 2.8. IP configuration of PC2

The IP configuration of PC1 as shown below in Figure 2.9.

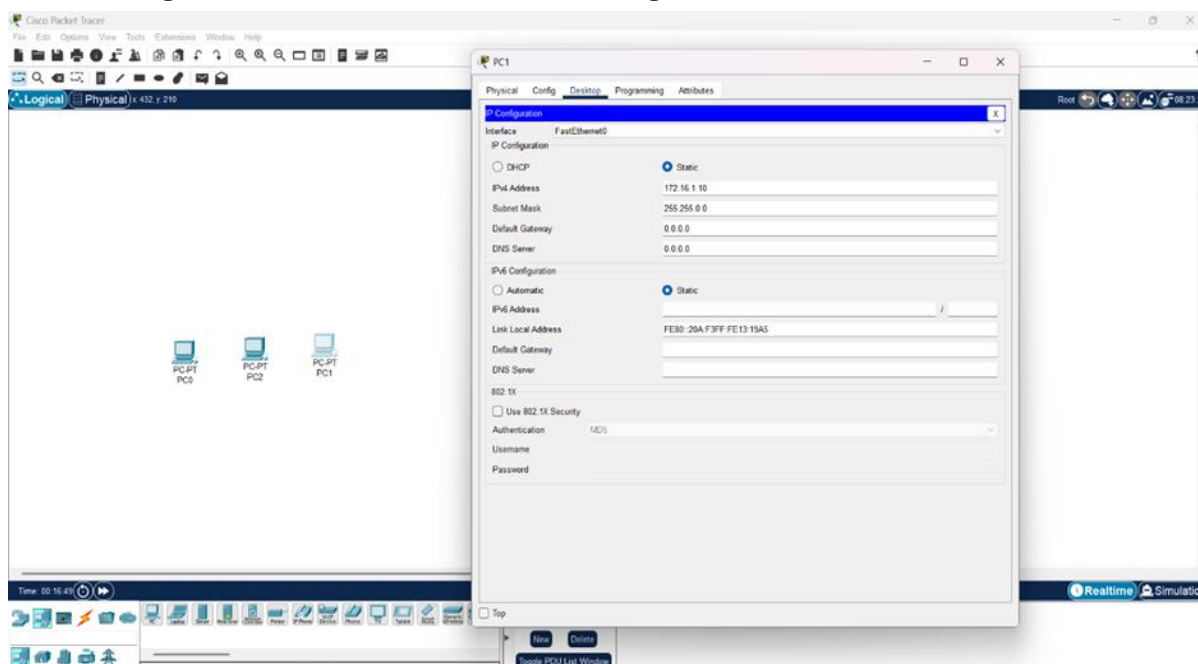


Fig 2.9. IP configuration of PC1

2.3.2. Ip address to binary conversion:

In this part, we'll manually convert an IP address to its binary representation and then verify it conceptually. Packet Tracer itself doesn't directly show the binary form in the IP configuration window, but we can understand the conversion process.

Step-by-Step Explanation (No Direct Packet Tracer Action):

Let's take the Class C IP address we used for PC2: 192.168.1.10. We'll convert each octet to its 8-bit binary equivalent.

1. Convert the First Octet (192):

- 128 (2^7) goes into 192 once ($192 - 128 = 64$) -> 1

- 64 (2^6) goes into 64 once ($64 - 64 = 0$) -> 1
 - 32 (2^5) goes into 0 zero times -> 0
 - 16 (2^4) goes into 0 zero times -> 0
 - 8 (2^3) goes into 0 zero times -> 0
 - 4 (2^2) goes into 0 zero times -> 0
 - (2^1) goes into 0 zero times -> 0
 - 1 (2^0) goes into 0 zero times -> 0
- o Binary: 11000000

2. Convert the Second Octet (168):

- 128 goes into 168 once ($168 - 128 = 40$) -> 1
 - 64 goes into 40 zero times -> 0
 - 32 goes into 40 once ($40 - 32 = 8$) -> 1
 - 16 goes into 8 zero times -> 0
 - 8 goes into 8 once ($8 - 8 = 0$) -> 1
 - 4 goes into 0 zero times -> 0
 - goes into 0 zero times -> 0
 - 1 goes into 0 zero times -> 0
- Binary: 10101000

3. Convert the Third Octet (1):

- Only the last bit (2^0 or 1) is set.
- Binary: 00000001

4. Convert the Fourth Octet (10):

- 8 goes into 10 once ($10 - 8 = 2$) -> 1
 - 4 goes into 2 zero times -> 0
 - goes into 2 once ($2 - 2 = 0$) -> 1
 - 1 goes into 0 zero times -> 0
- Binary: 00001010

The complete binary representation of the IP address 192.168.1.10 is:
11000000.10101000.00000001.00001010

2.3.3. Subnetting:

The following images are of Packet tracer for the subnetting via configuring the router as shown in Figure 2.10 (a), Figure 2.10 (b), Figure 2.10 (c), respectively.

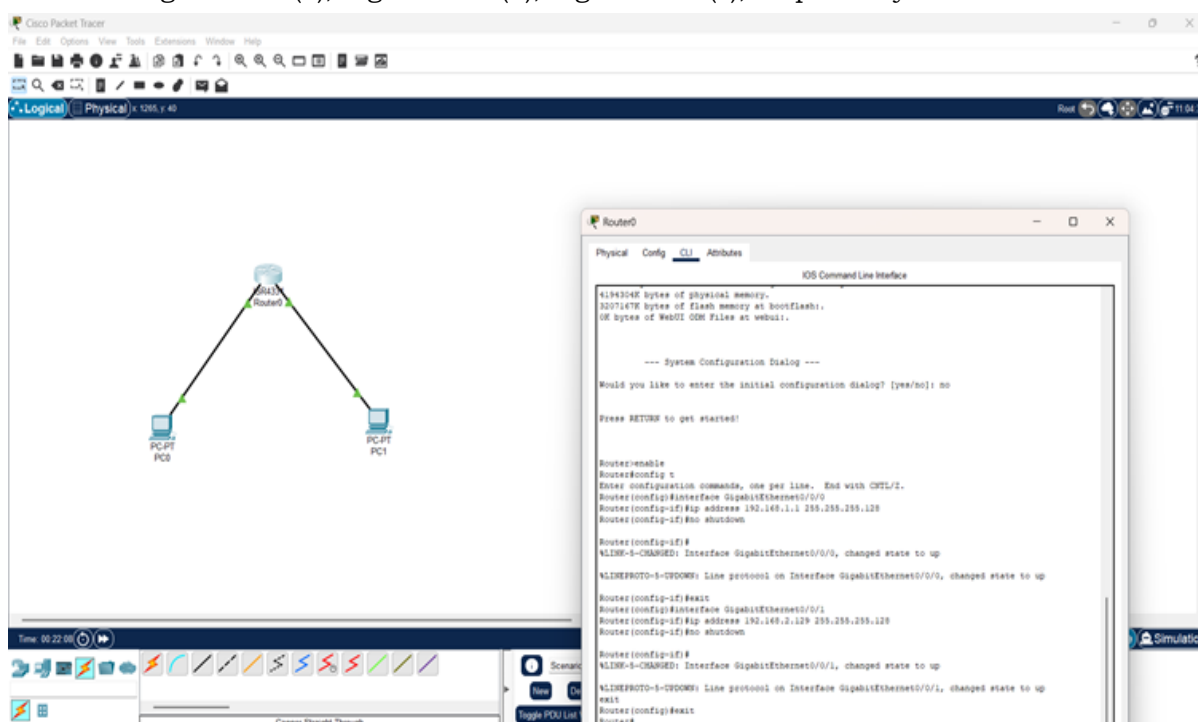


Fig 2.10 (a): Subnetting via configuring router

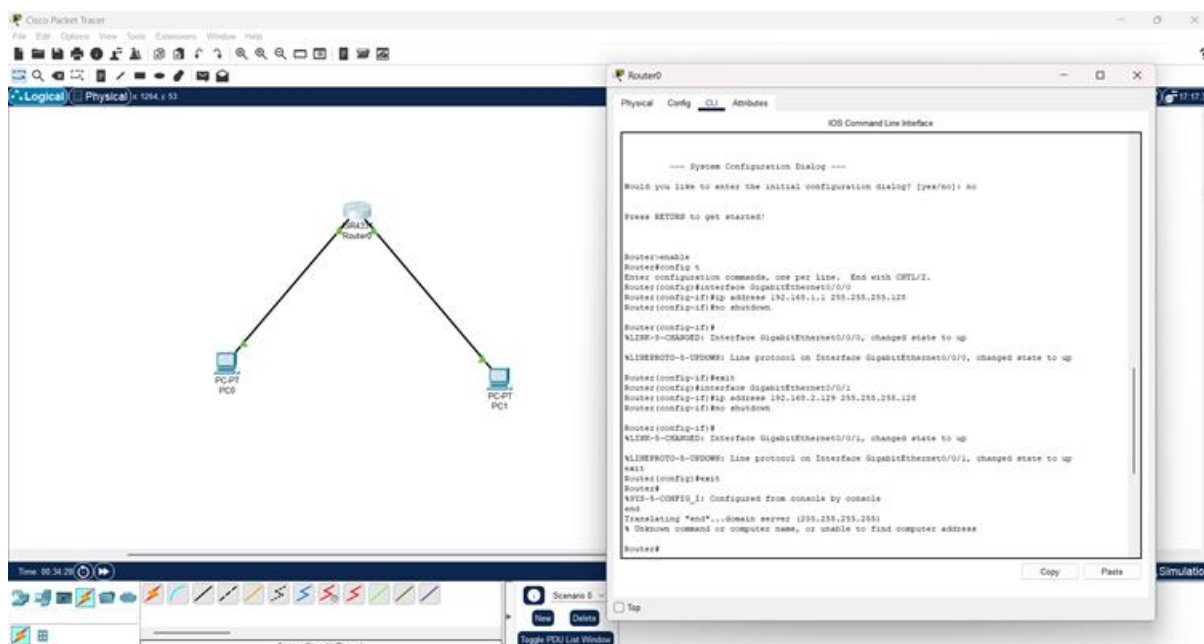


Fig 2.10 (b): Subnetting via configuring router

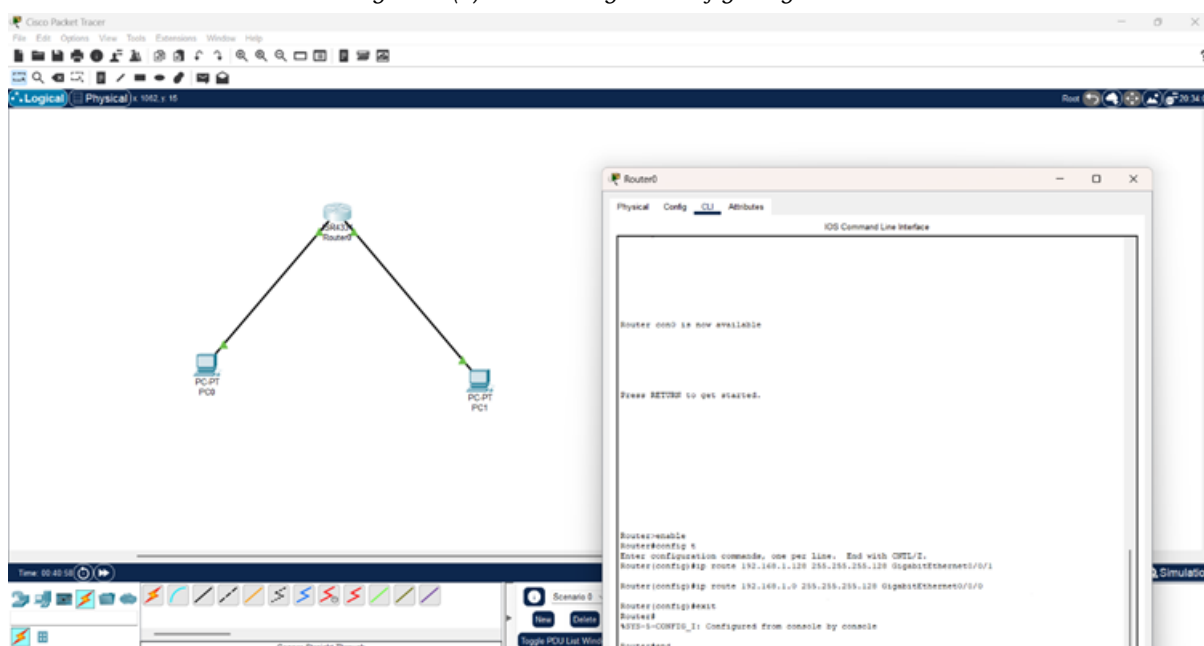


Fig 2.10 (c): Subnetting via configuring router

Summary

- OSI is a 7-layer model created by ISO for standardizing network connections.
- The 5-layer model of TCP/IP is used practically in the Internet and real-world networks.
- Physical, Data Link, Network, Transport, Session, Presentation, Application are OSI Model layers.
- Layers of TCP/IP: Physical, Data Link, Network, Transport, Application layer.
- Each layer in the models has a specific role, such as addressing, routing, or encryption.
- TCP ensures reliable, connection-based communication.
- UDP provides faster, connectionless communication.
- For addressing and routing of data packets, IP is responsible.
- At the Application layer, the protocols like HTTP, FTP, SSH, and DNS operate.
- Port numbers help direct data to the correct application on a device.

Check your progress**A. Multiple Choice Questions**

1. How many layers are there in the OSI model?
 - A. 5
 - B. 6
 - C. 7
 - D. 4
2. How many layers are there in the TCP/IP model?
 - A. 3
 - B. 5
 - C. 4
 - D. 7
3. Which layer of the OSI model ensures reliable communication?
 - A. Network
 - B. Transport
 - C. Session
 - D. Data Link
4. Which protocol operates at the transport layer?
 - A. IP
 - B. HTTP
 - C. TCP
 - D. DNS
5. What does IP stand for?
 - A. Internal Protocol
 - B. Internet Protocol
 - C. Integrated Packet
 - D. Interconnect Path
6. Which layer is responsible for data encryption in the OSI model?
 - A. Transport
 - B. Presentation
 - C. Session
 - D. Network
7. DNS works on which layer of the TCP/IP model?
 - A. Transport
 - B. Internet
 - C. Application
 - D. Network Access
8. Which protocol maps IP addresses to MAC addresses?
 - A. IP
 - B. ARP
 - C. RARP
 - D. DHCP
9. Logical addressing is done at which layer in OSI?
 - A. Data Link
 - B. Network
 - C. Transport
 - D. Session
10. Which OSI layer defines the electrical transmission of data?
 - A. Physical

- B. Data Link
- C. Network
- D. Transport

B. Fill in the Blanks

1. OSI stands for _____.
2. TCP/IP model was established by _____.
3. The _____ layer in OSI is closest to the user.
4. _____ layer handles reliable delivery of segments.
5. _____ protocol ensures data is sent error-free and in order.
6. The main function of IP is to provide _____.
7. Port numbers range from _____ to _____.
8. _____ protocol is faster but does not guarantee delivery.
9. FTP works on the _____ layer.
10. MAC addressing occurs at the _____ layer.

C. True or False

1. OSI model has 6 layers.
2. TCP is a connectionless protocol.
3. UDP is faster than TCP.
4. Application layer exists only in OSI.
5. TCP/IP is used in real-world networking.
6. Presentation layer compresses and encrypts data.
7. IP is responsible for reliable transmission.
8. TCP and UDP both operate at transport layer.
9. DNS is a transport layer protocol.
10. TCP/IP does not support session management.

D. Short Answer Questions

1. What is the OSI model and who developed it?
2. List all seven layers of OSI model.
3. What is the purpose of the transport layer?
4. Explain the difference between TCP and UDP.
5. What is the main function of IP?
6. Define a port number.
7. Discuss the role of the application layer in TCP/IP.
8. What is meant by encapsulation in network communication?
9. Which layers of OSI are implemented in software?
10. Explain the use of the presentation layer.

Answer Key**A. Multiple Choice Questions**

1. C., 2. C., 3. B., 4. C., 5. B., 6. B., 7. C., 8. B., 9. B., 10. A.

B. Fill in the blanks:

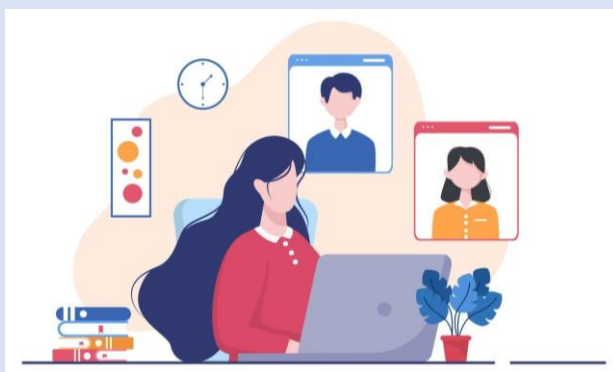
1. Open Systems Interconnection, 2. Department of Defense (DoD), 3. Application, 4. Transport, 5. TCP (Transmission Control Protocol), 6. Routing and addressing, 7. 0 to 65535, 8. UDP (User Datagram Protocol), 9. Application, 10. Data Link

C. True or False

1. False, 2. False, 3. True, 4. False, 5. True, 6. True, 7. False, 8. True, 9. False, 10. True

Chapter-3**Network Management and Cloud-based Networking**

Neha was the tech helper at her school. One day, the school got new computers and Wi-Fi routers. But soon, the internet was slow, and printers weren't connecting. Neha asked her teacher for help. The teacher explained, "This is where network management helps. It allows us to monitor, control, and fix problems in the school's network." He showed her a screen with all devices connected and how they could be managed from one place. Later, they used cloud-based networking to store class files and allow students to access them from home. Neha was amazed! Now, even if a computer stopped working, the data was safe online. Neha smiled, realizing how smart networks keep everything running smoothly, both in school and beyond.

**3.1 Network Management**

This is the procedure of administering, managing, and operating a data network using a network management system. The current system uses software and hardware to continuously collect and analyze data and push out configuration changes to improve performance, reliability, and security.

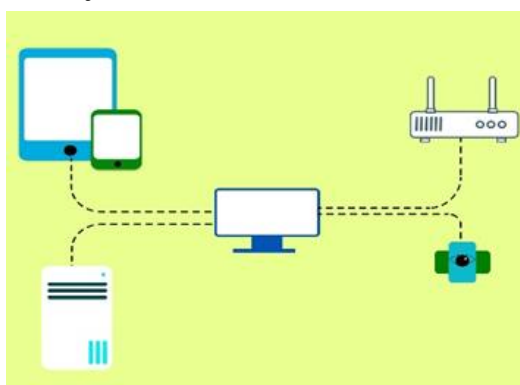


Fig 3.1. Network Management

Involves configuring, monitoring, and possibly reconfiguring components in a network with the goal of providing optimal performance, minimum downtime, proper security, accountability, and flexibility.

Importance of network management

Network management is extremely important because it ensures that computer networks operate smoothly and securely. Without it, networks can become slow, disorganized, or even experience complete failure. Here's why it's important:

- **Keeps Everything Running:**

Network management makes sure that all the computers, printers, and other devices are connected properly and can talk to each other without problems.

- **Fixes Problems Quickly:**

If something goes wrong — like slow internet, dropped connections, or broken devices — network management helps find the problem fast and fix it before it causes a big mess.

- **Protects Against Hackers:**

A good network management system also keeps an eye out for cyber threats. It helps protect personal information and stops hackers from getting into the system.

- **Saves Time and Money:**

By preventing big problems before they happen, network management saves a lot of time, effort, and money. Imagine if a whole school's internet went down during exams — fixing it would take forever without proper management!

- **Helps Networks Grow:**

As more computers, phones, and devices are added to a network, good management makes sure everything keeps working smoothly without slowing down.

Did you know how man came from manual to automated systems?

Well let's find out how it all became possible

Once upon a time, in a world not too different from ours, everything was done manually. People filled out forms by hand, counted large sums of money using paper and pen, and recorded important information in thick, dusty books. In offices, workers spent hours typing on old typewriters, sorting endless piles of paper, and running around delivering messages. In schools, teachers would call out every student's name one by one to take attendance, carefully marking each as present or absent in a large register. Everything depended on human effort, and while people tried their best, mistakes were common. A small error could cause a big problem, and fixing it often took even more time and effort.

As time passed, people started asking an important question: "What if there was a way to make our work easier and faster?" Slowly, technology began to grow. Simple machines like calculators were invented to help with basic math. Then came computers that were big, bulky, and slow at first, but still much faster than a person. These early inventions marked the beginning of automation. Tasks that once took hours could now be done in minutes. In businesses, computers started keeping track of money and inventory. In schools, attendance could be recorded using ID cards or biometric scanners. Hospitals, banks, factories — every place began to change.

Over time, automation became smarter and more powerful. Software was created to manage thousands of records without any human writing. Machines could now send emails instantly instead of delivering letters by hand. Even homes started using automation: lights could turn on by themselves, and robots could vacuum the floor! Automation didn't just make life faster; it also made it more accurate. With computers handling repetitive work, there were fewer mistakes, less stress, and more time for humans to focus on creative and important tasks.

This incredible journey, from the slow, tiring days of manual work to the fast, smart world of automated systems, changed the way people lived and worked forever. Today, automation is such a big part of our lives that it's hard to even imagine going back to the old days. It shows how human curiosity, creativity, and the need to make life better have shaped the world we live in now — and will continue to shape the future.

Functions Network management has different functions such as:

This is an automated procedure for configuring, handling, testing, provisioning, and operating different devices in your network. Availability increases significantly when daily functions and tasks are automatically controlled. It also includes power monitoring and software updates. These features enhance performance, customer experience, and network security. The assurance system supports not only AI and ML, but also network analysis, application analysis, and policy analysis to achieve complete network assurance. It consists

of proactive and reactive measures carried out in collaboration with network administrators, including the replacement of certain network equipment.

3.1.1 Key Network Management Functions:

Network management plays a major role in ensuring that computer networks—the systems that connect devices together—work smoothly, safely, and without interruptions. Without proper network management, networks can become slow, unsafe, or even stop working completely. There are five main functions of network management that help keep everything in order. Let's explore each one in detail:

Fault Management

Fault management is like the "emergency response team" of the network. Whenever something goes wrong—like when the internet at school suddenly stops working or a computer cannot connect to the printer—fault management steps in. Fault management systems often work automatically. They constantly monitor the network, looking for signs of trouble such as failed devices, broken cables, or overloaded servers. When a problem is detected, the system can alert the network administrator (the person in charge of the network) through alarms, emails, or notifications.

For example, if a router in the school's network fails, the fault management system might instantly send a message saying, "Router down in the library area," allowing technicians to fix it before students even notice a problem. This reduces downtime and keeps the network reliable.

Configuration Management

Configuration management is like keeping an organized "map" or "inventory list" of the network. It keeps track of every device (computers, printers, routers, switches, etc.) and their settings, such as IP addresses, passwords, and special configurations.

This helps in many ways:

- If a device suddenly stops working, the technician can quickly check what the original settings were and fix the issue faster.
- If the school wants to upgrade the network, knowing the existing setup saves time.

Without good configuration management, even small mistakes—like a wrong IP address—can cause major network issues, just like getting lost without a proper map!

Accounting Management

Accounting management is all about keeping track of how the network is being used.

Imagine if one student is downloading a massive file and using up all the school's internet speed — accounting management would help identify that student and show exactly how much bandwidth they are using.

Some important purposes of accounting management are:

- **Controlling Costs:** By finding out who is using a lot of resources, administrators can prevent unnecessary costs.
- **Fair Usage:** It ensures that everyone gets a fair share of network access.

Performance Management

Performance management checks how "healthy" the network is. It asks questions like:

- Is the network running fast enough?
- Are there any delays in sending or receiving data?
- Are any parts of the network overloaded with traffic?

Think of performance management as the "fitness coach" of the network — constantly measuring things like network speed, signal strength, number of active users, and load on servers.

If a problem is spotted, like a section of the network being slow because too many users are online at once, performance management suggests solutions, such as upgrading the connection speed or balancing the traffic.

This helps ensure that students, teachers, and administrators can enjoy fast internet browsing, smooth online classes, and quick access to files at any time without frustration.

Security Management

Security management acts like a bodyguard for the network. In today's world, cyber threats like hackers, viruses, malware, and ransomware are everywhere. Without strong security, a network can be easily attacked, leading to stolen data, damaged files, or even a complete shutdown.

Security management sets up protections such as:

- **Firewalls:** Barriers that block harmful traffic.
- **Antivirus Programs:** Tools that find and destroy viruses.
- **User Authentication:** Making sure that only authorized people can access the network using passwords, fingerprints, or special codes.
- **Data Encryption:** Scrambling data into secret codes so that even if hackers intercept it, they cannot understand it.

Security management also includes monitoring the network 24/7 for suspicious activities, like someone trying to log into a server from another country. If something dangerous is detected, the system can automatically block access and alert the security team.

Without security management, networks would be wide open to attacks, and important information like student records, exam papers, and personal details could be stolen.

3.2 Current Network Management Challenges

Even though technology has advanced a lot, managing a computer network is still not an easy job. In fact, as technology keeps growing, so do the challenges faced by network managers. Let's look at some of the biggest challenges they deal with every day:

Growing Networks

Today, more devices are being connected to networks than ever before. It's no longer just computers and phones—now we have smart TVs, smartwatches, tablets, gaming consoles, security cameras, and even refrigerators that connect to the internet!

Every time a new device is added, it needs to be properly set up and managed so that it doesn't cause problems. As the number of devices keeps growing, it becomes harder to keep track of everything.

Imagine trying to manage hundreds or thousands of students' laptops, phones, and tablets at a big school! Without careful planning and organization, things can easily become messy, slow, and confusing.

This explosion of connected devices is called the "Internet of Things" (IoT), and while it's exciting, it makes network management much more complex and riskier.

Cybersecurity Threats

Hackers and cybercriminals are always inventing new tricks to break into networks. Their goal might be to steal important information, cause damage, demand ransom money, or simply create chaos.

Viruses, malware, phishing attacks, ransomware — all these threats are growing every day. Even one small weakness in the network (like a weak password or an outdated system) can give hackers a chance to break in.

Network managers must constantly stay alert—updating security systems, patching vulnerabilities, setting up firewalls, and teaching users how to stay safe online.

Fighting cybersecurity threats is a never-ending battle because attackers are always getting smarter, and new types of attacks appear almost daily.

It's like a constant game of cat and mouse, where managers must always stay one step ahead.

Complex Technology

Networks today are no longer simple wires connecting computers. Modern networks are made up of complicated systems like:

Virtual Networks (where software creates networks inside computers)

Cloud Services (where data is stored on the internet instead of local computers)

Artificial Intelligence and Automation (where machines manage some tasks automatically)

Because of all these advancements, setting up and managing a network now requires highly trained experts.

If something goes wrong, it's not as easy as unplugging and plugging back a cable. Problems could involve complicated software errors, hardware failures, compatibility issues, or hidden bugs.

Network managers must constantly learn new skills and stay updated with the latest technologies—or risk falling behind.

Downtime

Downtime happens when the network stops working for a period of time. Even a few minutes of downtime can cause big problems:

- Schools might not be able to access online lessons.
- Businesses could lose money if customers can't buy products online.
- Hospitals could face life-threatening issues if systems go offline.

Network managers have the huge responsibility of making sure networks stay up and running 24/7.

They must set up backup systems, test networks regularly, fix small issues before they become big ones, and have emergency plans ready—just in case something goes wrong.

Downtime is not just a technical issue — it can damage reputations, trust, and profits.

Remote Work

After COVID-19, online learning and remote work became the “new normal” for many people. Even now, millions of students and workers operate from home instead of going to school or the office.

This puts extra pressure on networks, as they now have to connect people who are spread out across cities, countries, or even the world.

Remote work creates special challenges such as:

- Making sure connections are strong and fast no matter where a person is.
- Ensuring security, even though people are using different types of devices and home internet.
- Supporting video meetings, file sharing, and online collaboration tools without crashing the system.

Network managers must now build systems that are flexible and powerful enough to handle this new style of working — all while keeping everything safe and efficient.

3.3. Cloud-Based Network Management Models

In the early days of computers, managing a network meant being physically present at the location where all the servers, routers, and wires were installed — like inside a school's computer lab or a company's data center.

Technicians had to walk over to the machines, plug in cables, fix problems manually, and make sure everything was updated right there on the spot.

However, times have changed. With the rise of cloud technology, network management has become much smarter, faster, and more flexible.

Now, network administrators can manage, monitor, and even fix a network from anywhere in the world — just by logging into the internet!

This new method is called Cloud-Based Network Management.

In this type, the devices are connected to a cloud platform — a giant system of servers located somewhere else — that handles many important tasks like monitoring, updating, fixing, and protecting the network.

There are three main models of cloud-based network management, depending on how the network and the cloud are set up:

Public Cloud

In the Public Cloud model, a business, school, or organization rents cloud services from large companies. These companies have thousands of powerful servers spread across the world. They offer ready-to-use tools and services for anyone who wants to manage their network without having to build their own complicated systems.

Advantages of using the Public Cloud:

- **Cost-Effective:** It's much cheaper because you don't have to buy and maintain expensive servers.
- **Easily Scalable:** If the network grows, you can easily add more services by paying a little more.
- **Convenient:** You don't need a big IT team on-site. Everything can be controlled from a laptop — even from another city or country!

Disadvantages:

- **Less Private:** Since you're sharing resources with other customers on the same cloud, there is always a slight risk of losing some privacy or getting hacked if security isn't top-notch.

Example: Imagine a small online bookshop that doesn't have enough money to set up its own massive network system. Instead, it uses Amazon's public cloud to run its website and manage customer orders.

Private Cloud

In the Private Cloud model, a company or school builds and manages its own cloud. This means they set up their own servers, design their own cloud structure, and maintain full control over everything.

Advantages of Private Cloud:

- **High Security:** Since the cloud is completely private, only trusted people have access.
- **Full Control:** Organizations can customize everything to meet their specific needs — from storage to security to performance.
- **Better Privacy:** Sensitive data stays within the organization and doesn't mix with data from other users.

Disadvantages:

- **Expensive:** Building and maintaining a private cloud requires a lot of money, highly skilled workers, and ongoing maintenance.
- **Requires Expertise:** Running a private cloud is complex and needs a strong IT team.

Example: A major bank that deals with highly confidential customer information might choose to create a private cloud to ensure that no outside company can ever access their important data.

Hybrid Cloud

It is a combination of both private and public clouds.

Organizations use the public cloud for general tasks like hosting a public website or storing non-sensitive data, and the private cloud for more sensitive activities like managing financial records, personal data, or confidential projects.

Advantages of Hybrid Cloud:

- **Flexibility:** You can choose what to keep private and what to share publicly.
- **Cost Savings:** Expensive private cloud resources are used only where needed, while cheaper public cloud services handle the rest.
- **Business Continuity:** If one system fails, another can take over, reducing the risk of downtime.

Disadvantages:

- **Complex to Manage:** Managing two different types of clouds at the same time can get tricky and requires careful planning and integration.

Example: A university might use a public cloud to manage its website and share learning materials for students, while keeping exam results, personal student records, and financial information on a private cloud.

3.3.1 The Advantages of Cloud-Based Network Management

1. Remote Access and Flexibility: IT teams can connect to and monitor/troubleshoot networks from anywhere, which is helpful for remote workers and organizations with multiple locations.

2. Cost Effective & Scalable – Subscription-based services reduce capital expenditures on hardware, and networks can scale, or shrink, easily with minimal modifications.

3. Automatic Updates & Security – Cloud vendors have responsibility of managing patches and fixes, and built in protections like encryption and authentication.

4. Centralized Control – A single dashboard or monitoring software helps simplify monitoring, managing, and troubleshooting all network devices.

5. Improved Performance & Reliability – Proactive monitoring and automation improve RMA time and device efficiency.

3.4 Migration to cloud-based network management

Now imagine your school has a computer network. It's like the nervous system of the school, connecting all the computers, printers, and Wi-Fi so information can flow smoothly. Managing this network is like being the "traffic cop" for all the data – making sure everyone can connect, the internet is fast, and everything is secure.

Traditionally, managing this network meant having special physical boxes (servers and controllers) right there at the school. IT people had to set them up, maintain them, and fix them if they broke. This is called on-premises network management.

Now, let's talk about cloud-based network management. Instead of those physical boxes at the school, the management tools and software live on powerful computers somewhere else, linked through the internet – in the "cloud." You access and control everything through a website or an app.

Migration to cloud-based network management is the process of moving from managing the network the old, on-premises way to managing it using this new, cloud-based system. Think of it like moving your school's library catalog system from old physical index cards in cupboards to a digital database you can access from any computer. You're moving the

management process, not necessarily all the physical wires and devices themselves – though some devices might need updates or replacement to work with the cloud system.

So, why would a school or any organization go through this migration?

We touched on the benefits before, but let's put them in terms that might make more sense:

1. Managing from Anywhere (Remote Access):

Imagine your IT person is at home sick, but there's a Wi-Fi problem at school. With cloud management, they can often fix it from their house using their laptop. They don't need to drive to school just to flip a switch or check a setting. This is super helpful for supporting multiple school buildings or when staff are off-site.

2. Saving Money (Cost-Effective):

Those physical servers and controllers cost a lot to buy upfront. They also need electricity, cooling, and regular maintenance by expensive technicians. With cloud management, you usually pay a subscription fee, which is often less than the total cost of owning and maintaining all that hardware over time. The cloud provider handles the expensive equipment side of things. That saved money could go toward new computers or better sports equipment!

3. Growing (Scalability):

If your school adds a new building or gets hundreds of new tablets, the old physical system might struggle to keep up. You'd need to buy and install more hardware. With cloud management, you can usually just adjust your subscription or click a few buttons online to manage more devices. It's like instantly getting a bigger filing cabinet without having to go buy one and build it.

4. Automatic Fixes and Updates:

Just like your phone or computer gets software updates, network management software needs updates too — especially for security. With on-premises systems, the IT team has to manually install these updates, which takes time and can be tricky. In the cloud, the provider often handles these updates automatically, so your system is always current and protected against the latest online threats without the IT team having to do extra work.

5. Better Security (Often):

Good cloud companies specialize in security. They have huge teams and advanced systems dedicated to protecting their infrastructure and the data on it. While schools have security measures, they often don't have the resources of a major cloud provider. Moving management to the cloud can sometimes mean inheriting a higher level of protection against hackers and cyberattacks.

How Does the Migration Happen? (Simplified!)

It's not usually just flipping a switch — it's a process that takes careful planning:

1. Planning the Move:

The IT team first figures out exactly what network equipment the school has (routers, switches, Wi-Fi access points). They check if this equipment is compatible with cloud management systems. They also decide what they want from a new system — better monitoring? Easier guest Wi-Fi? Then they research and choose a cloud network management provider.

2. Setting Up the New "House":

The IT team sets up an account with the cloud provider. They establish the basic settings for the school's premises within the cloud platform — like naming different parts of the network or setting up basic security rules.

3. Trying It Out (Pilot):

Sometimes, they might try the cloud management on just a small part of the network first — maybe the computers in the library or the Wi-Fi in one section of a building. This is like a test run to make sure everything works as expected before committing fully.

4. Moving Things Over Gradually:

Instead of trying to switch everything at once (which could be chaotic — like moving all your furniture in one go), they usually migrate in stages. They might point one group of devices (like all the switches) to be managed by the cloud platform. Once that's working, they move on to the next group (like all the Wi-Fi access points). This minimizes disruption to students and staff.

5. Checking and Improving:

After devices are moved over, the IT team constantly monitors the cloud dashboard to make sure everything is running smoothly. They check if the internet speed is good, if devices are staying connected, and if the security is working. They might fine-tune settings based on what they see.

6. Saying Goodbye to the Old System:

Once the migration is complete and everyone is happy with the cloud system, the old on-premises management servers might be turned off or repurposed.

So, migrating to cloud-based network management is a strategic move to make networks easier to handle, more flexible, potentially cheaper in the long run, and often more secure — by using the power of the internet and specialized cloud services instead of relying solely on physical equipment located on-site. It's a big project, but it brings a lot of modern advantages!

Practical Activity 3.1.

Materials Required: Cisco Packet Tracer software (installed on the computer), Two End Devices (PC0 and PC1), Copper Straight-Through Cable (1 unit), Basic knowledge of IP configuration, Notebook or worksheet (for recording observations)

1. IP conflicts:

Step 1. Open Packet Tracer: Launch the application.

Step 2. Add End Devices: Drag and drop two PC devices (PC0 and PC1) onto the workspace.

Step 3. Connect the PCs: Use a “Copper Straight-Through” cable to connect PC0's FastEthernet0/0 interface to PC1's FastEthernet0/0 interface (this simulates a direct connection or being on the same switch, which we'll implicitly assume).

Step 4. Configure the First PC (PC0) with an IP Address:

- o Click on PC0.
- o Go to the Desktop tab.
- o Click on IP Configuration.
- o Select Static.
- o Enter the following:
IP Address: 192.168.1.10
Subnet Mask: 255.255.255.0
Leave Default Gateway and DNS Server blank.
- o Close the IP Configuration window.

Step 5. Configure the Second PC (PC1) with the SAME IP Address:

- o Click on PC1.
 - o Go to the Desktop tab.
 - o Click on IP Configuration.
 - o Select Static.
 - o Enter the EXACT SAME IP Address as PC0:
IP Address: 192.168.1.10
Subnet Mask: 255.255.255.0
Leave Default Gateway and DNS Server blank.
 - o Close the IP Configuration window.
6. Observe the Conflict (Simulation Mode is Best):

- o Switch to Simulation Mode (bottom right).
- o Click the "Add Simple PDU" icon (the envelope).
- o Click on PC0 (source) and then PC1 (destination).
- o Click the "Forward" button repeatedly. You might see ARP (Address Resolution Protocol) requests and replies. Observe if there are any error messages or unusual behaviour in the Event List.
- o Alternatively, go to Realtime Mode and try to ping PC1 from PC0's Command Prompt (ping 192.168.1.10). You might see inconsistent replies or error messages indicating a duplicate IP address on the network.

Identifying the Conflict: Configuring PC0 as shown in Figure 3.2, Configuring PC1 as shown in Figure 3.3, Pinging PC1 from PC0 as shown in Figure 3.4 and Simulating the connection as shown in Figure 3.5.

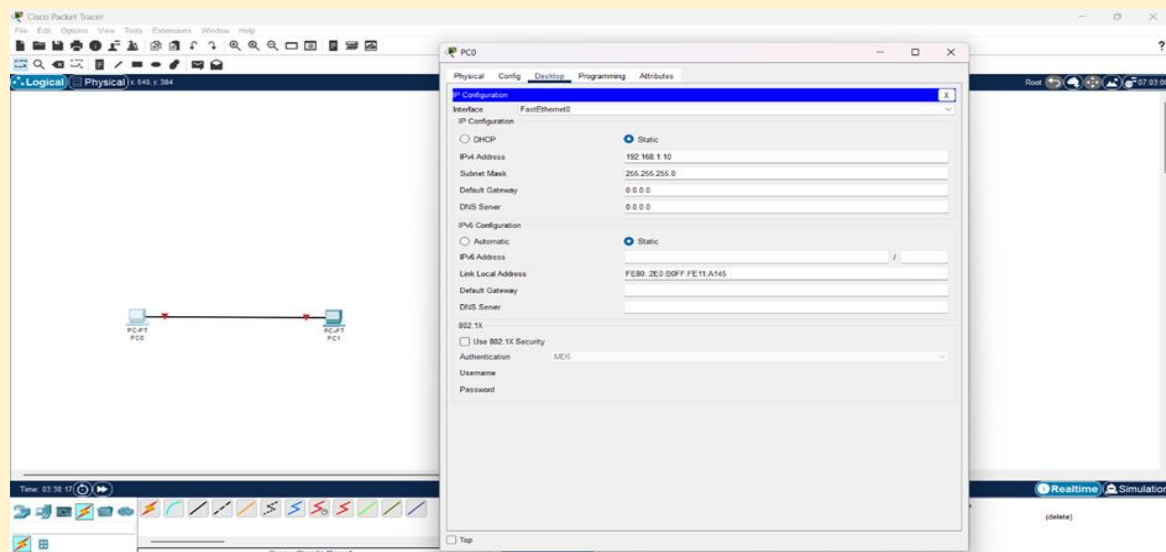


Fig 3.2. Configuring PC0

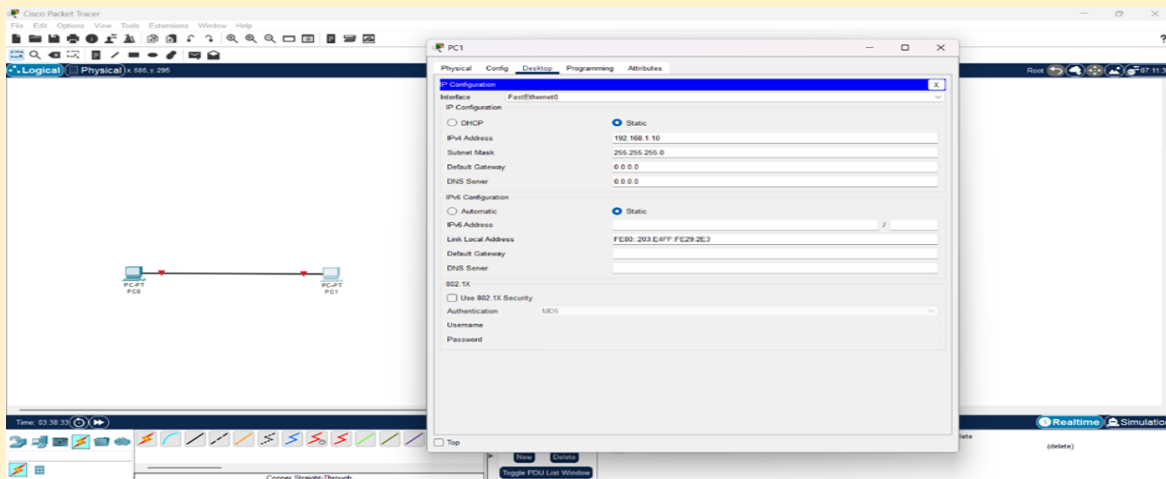


Fig 3.3. Configuring PC1

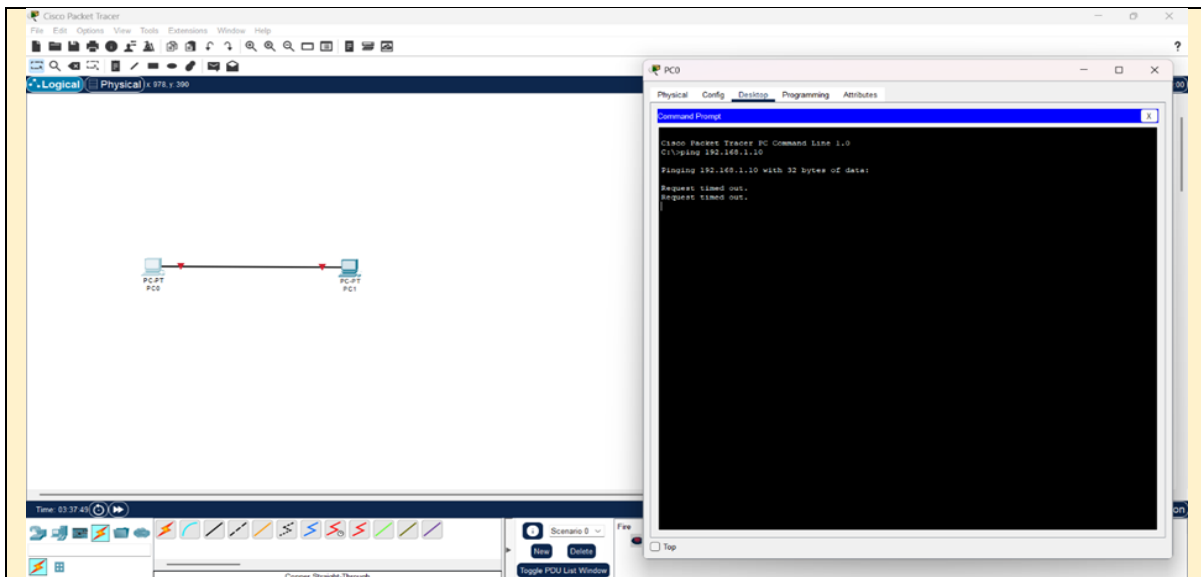


Fig 3.4. Pinging PC1 from PC0

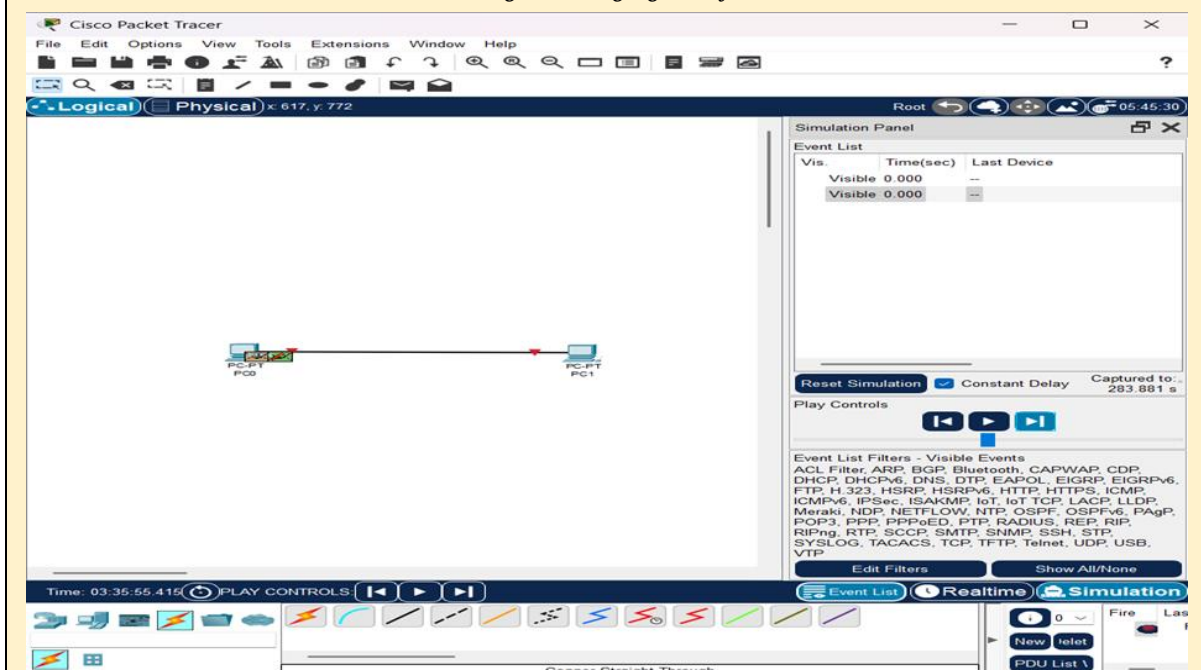


Fig 3.5. Simulating the connection

Practical Activity 3.2: Ip disconnections:

Step 1. Open Packet Tracer: Launch the application.

Step 2. Create a Simple Network: Add two PCs (PC2 and PC3) and a Switch (e.g., 2960).

Step 3. Connect the Devices:

- o Use "Copper Straight-Through" cables to connect PC2's FastEthernet0/0 to the Switch's FastEthernet0/1.
- o Connect PC3's FastEthernet0/0 to the Switch's FastEthernet0/2.

Step 4. Configure IP Addresses (on the same subnet):

- o PC2: IP Address 192.168.2.10, Subnet Mask 255.255.255.0
- o PC3: IP Address 192.168.2.11, Subnet Mask 255.255.255.0
- o Leave Default Gateway and DNS Server blank.

Step 5. Test Initial Connectivity (Ping):

- o Open PC2's Command Prompt (Desktop -> Command Prompt).
- o Ping PC3: ping 192.168.2.11. You should see successful replies.

Step 6. Simulate Disconnection:

- o Click on the "Select" tool (the arrow icon).
- o Click on the cable connecting PC3 to the Switch. You'll see the connection highlighted.
- o Press the Delete key on your keyboard (or right-click the cable and select "Delete"). This simulates a physical disconnection.

Step 7. Test Connectivity After Disconnection:

- o Go back to PC2's Command Prompt.
- o Ping PC3 again: ping 192.168.2.11. You should now see "Request timed out" messages, indicating a disconnection.

The inability to ping a previously reachable device clearly indicates a connectivity issue. This could be due to a physical disconnection, a faulty cable, or a problem with the network interface.

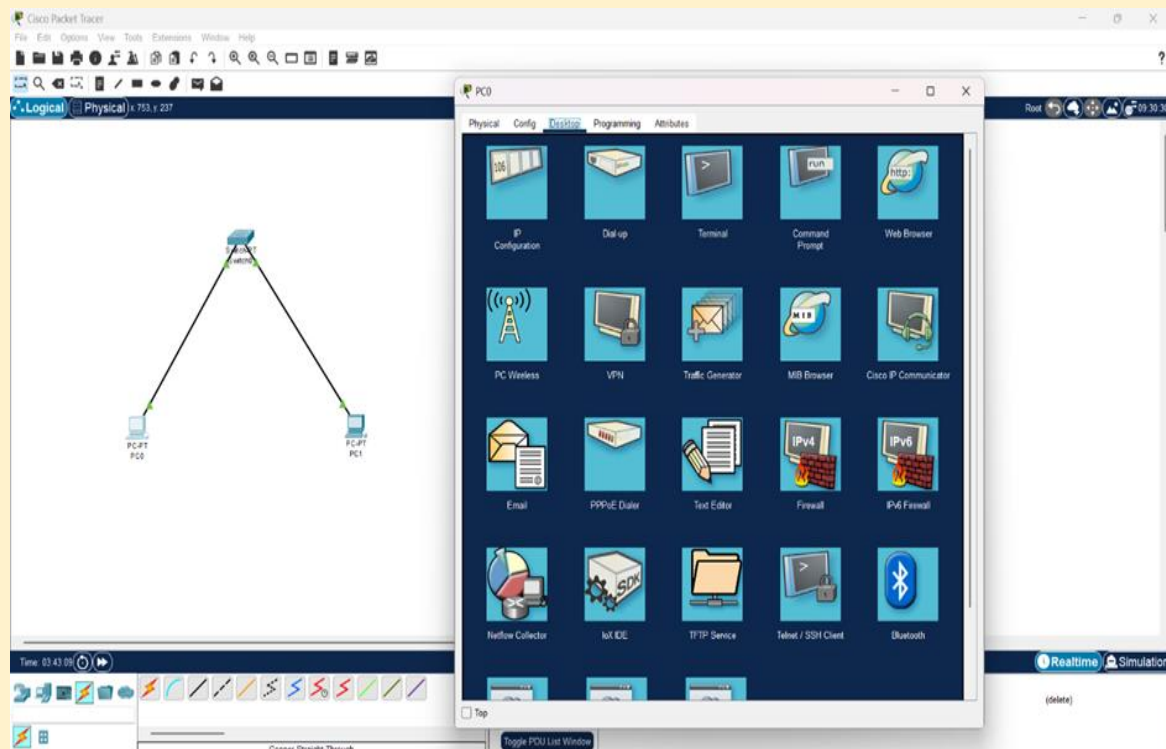


Fig 3.6. IP configuration of PC0

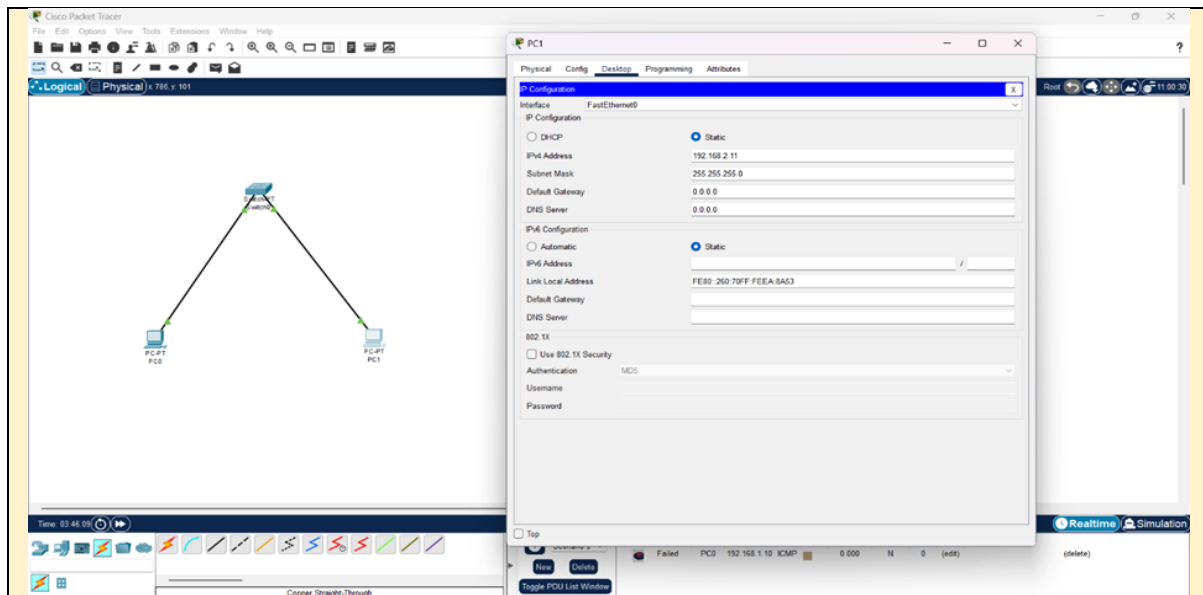


Fig 3.7. IP configuration of PC

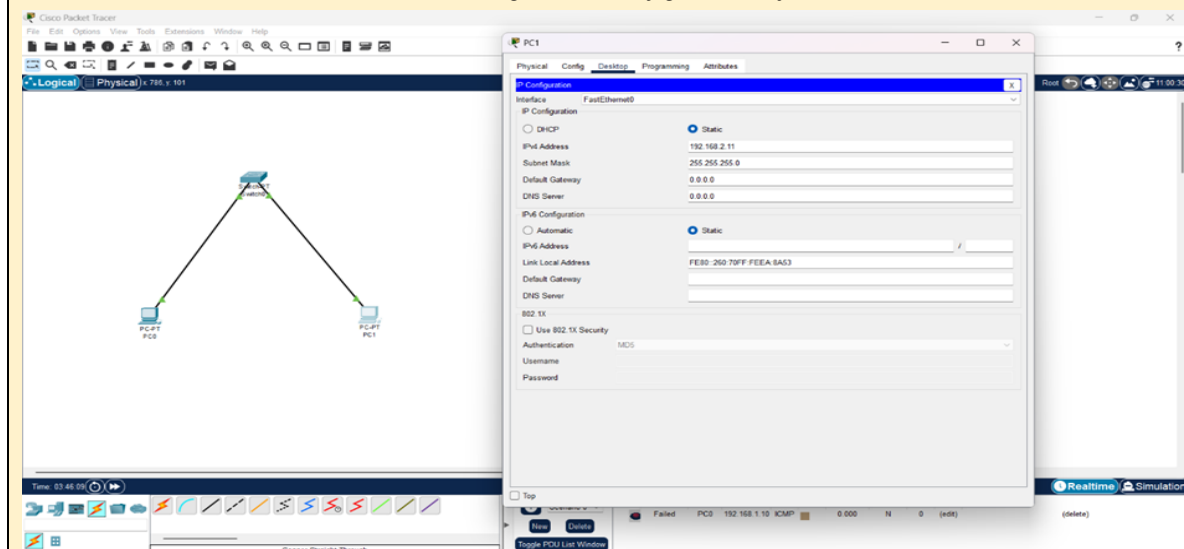


Fig 3.8. IP addresses for PC1

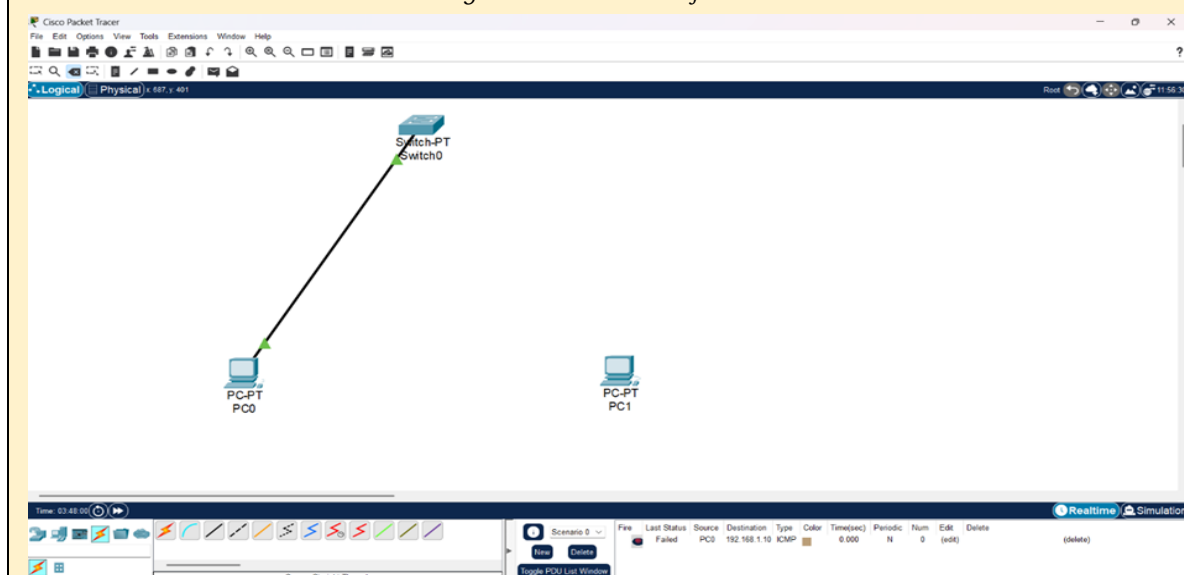


Fig 3.9. Removal of Switch to PC1 connection

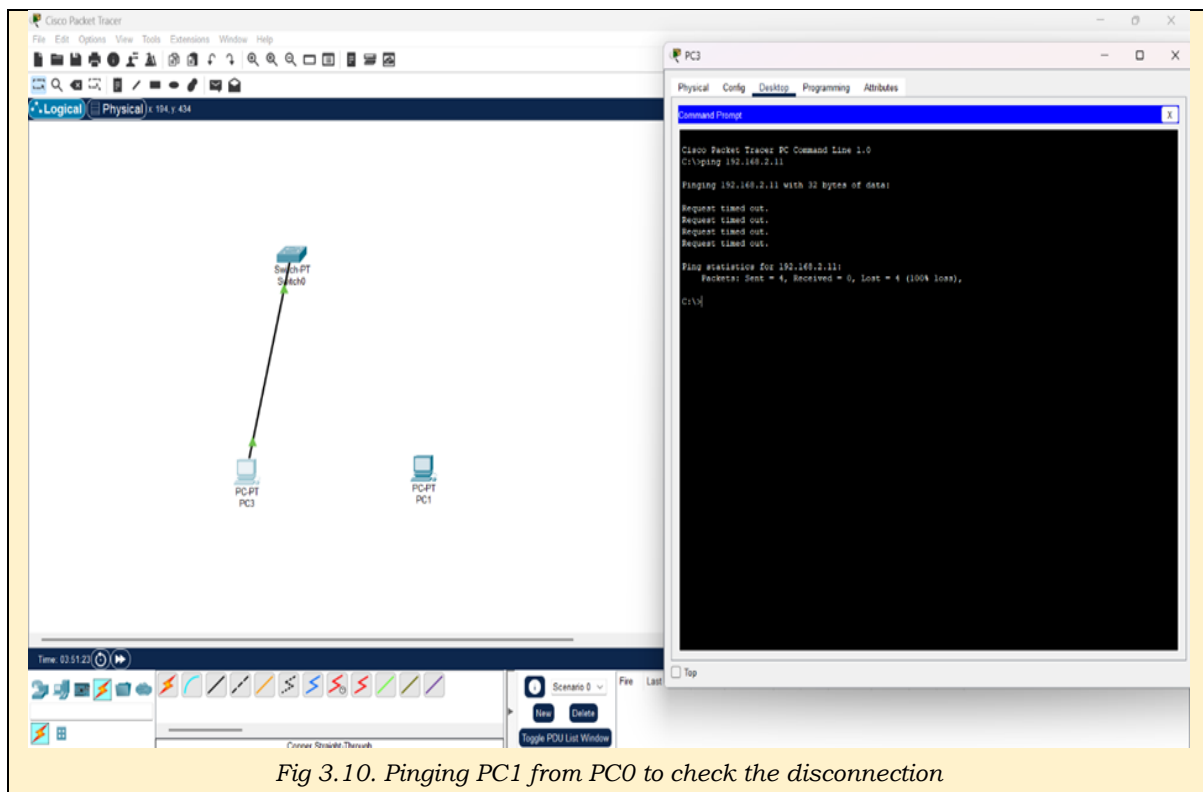
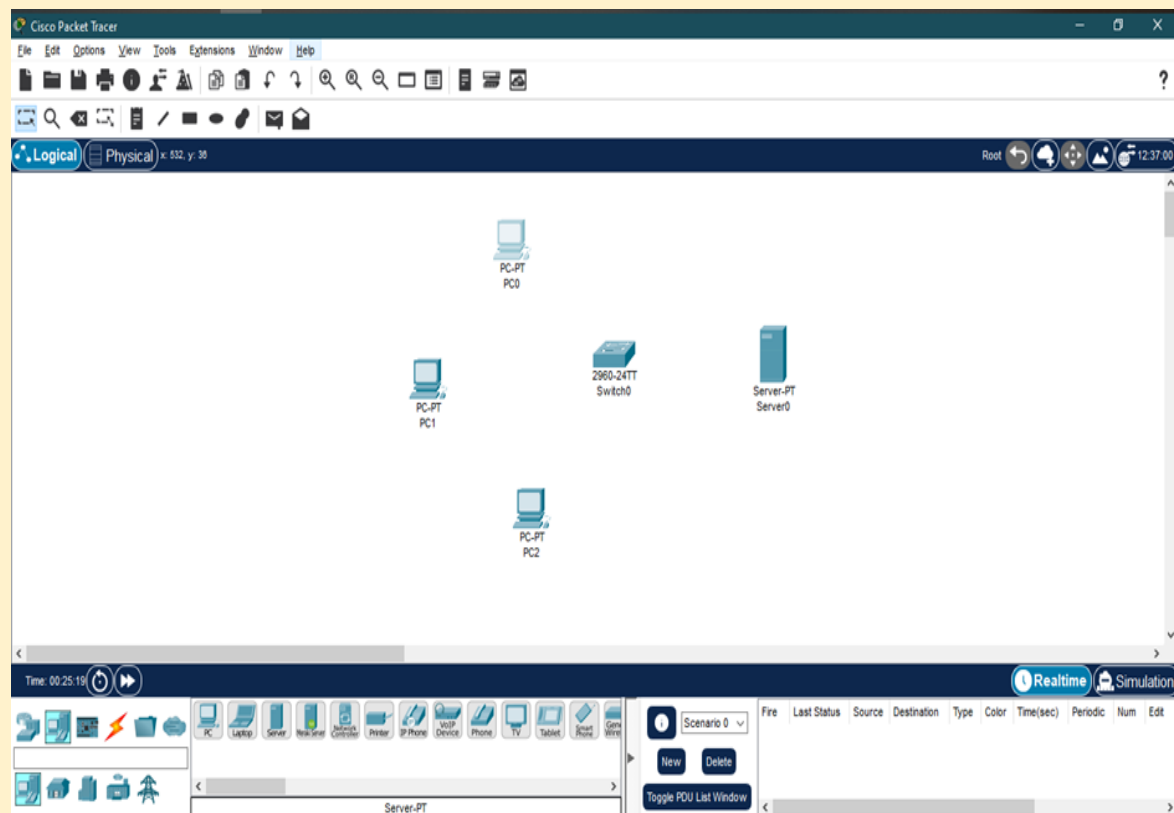


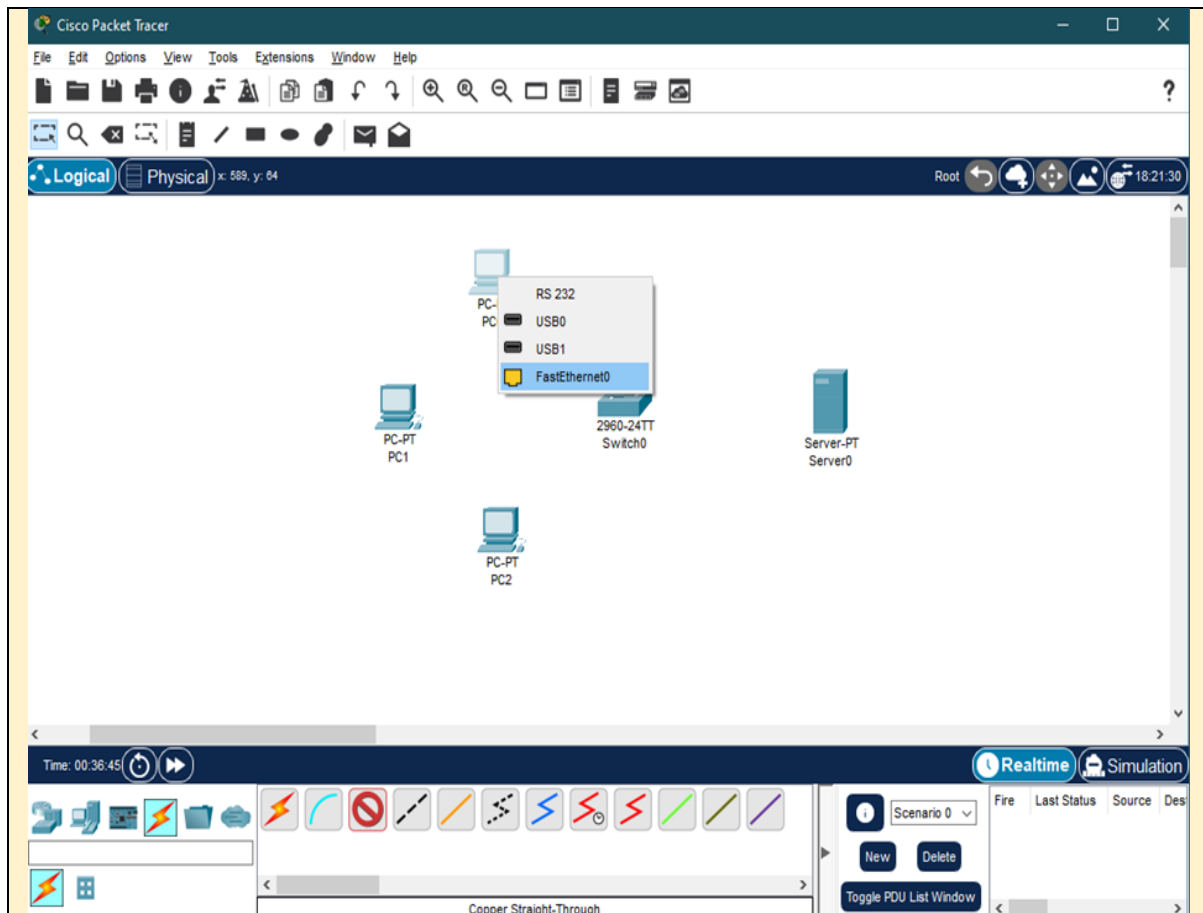
Fig 3.10. Pinging PC1 from PC0 to check the disconnection

Practical Activity 3.3: Cloud-based application of Network Management:

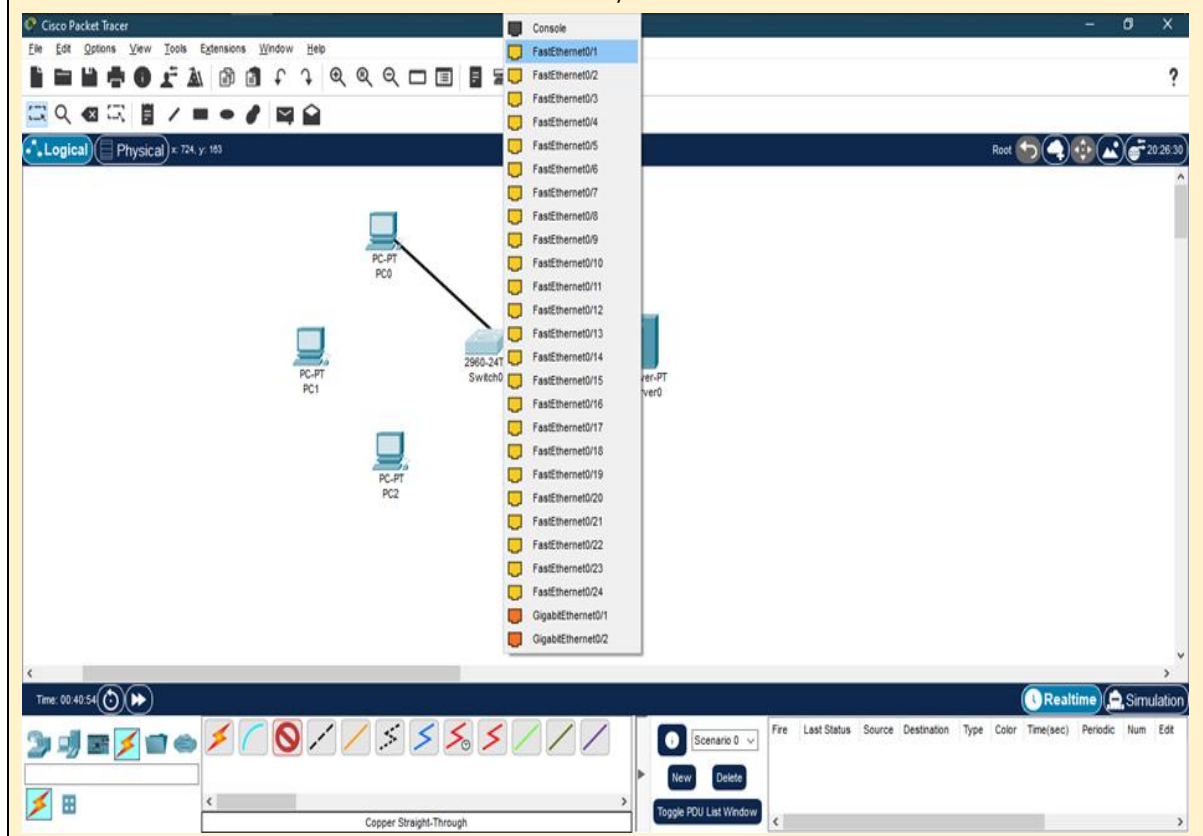
Step 1: Select the devices from the device panel. Here, we choose 3 PC, 1 server, and 1 switch.



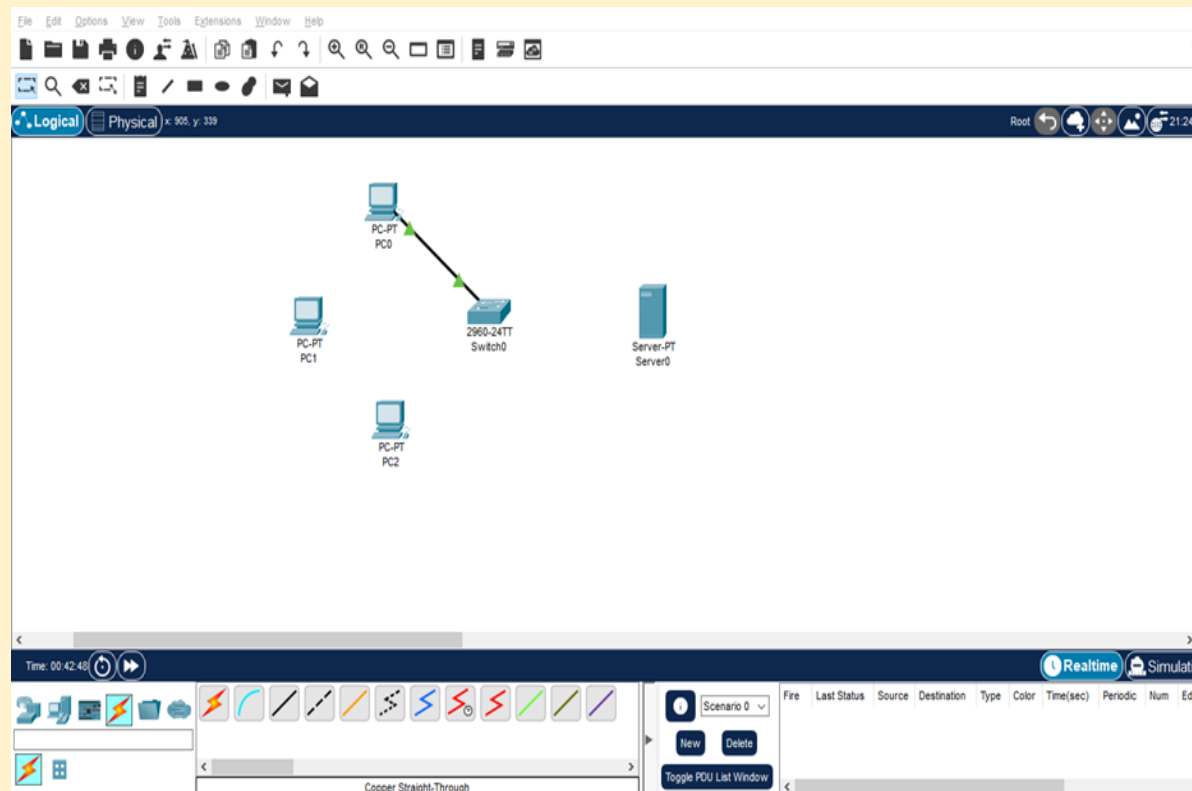
Step 2: Make a connection between the server, PCs and the switch using copper-straight-through wire. Take copper-straight-through wire from the connections panel. And click on PC0 and select FastEthernet0 network interface.



Then click on switch and select FastEthernet0/1 network interface to connect both devices.

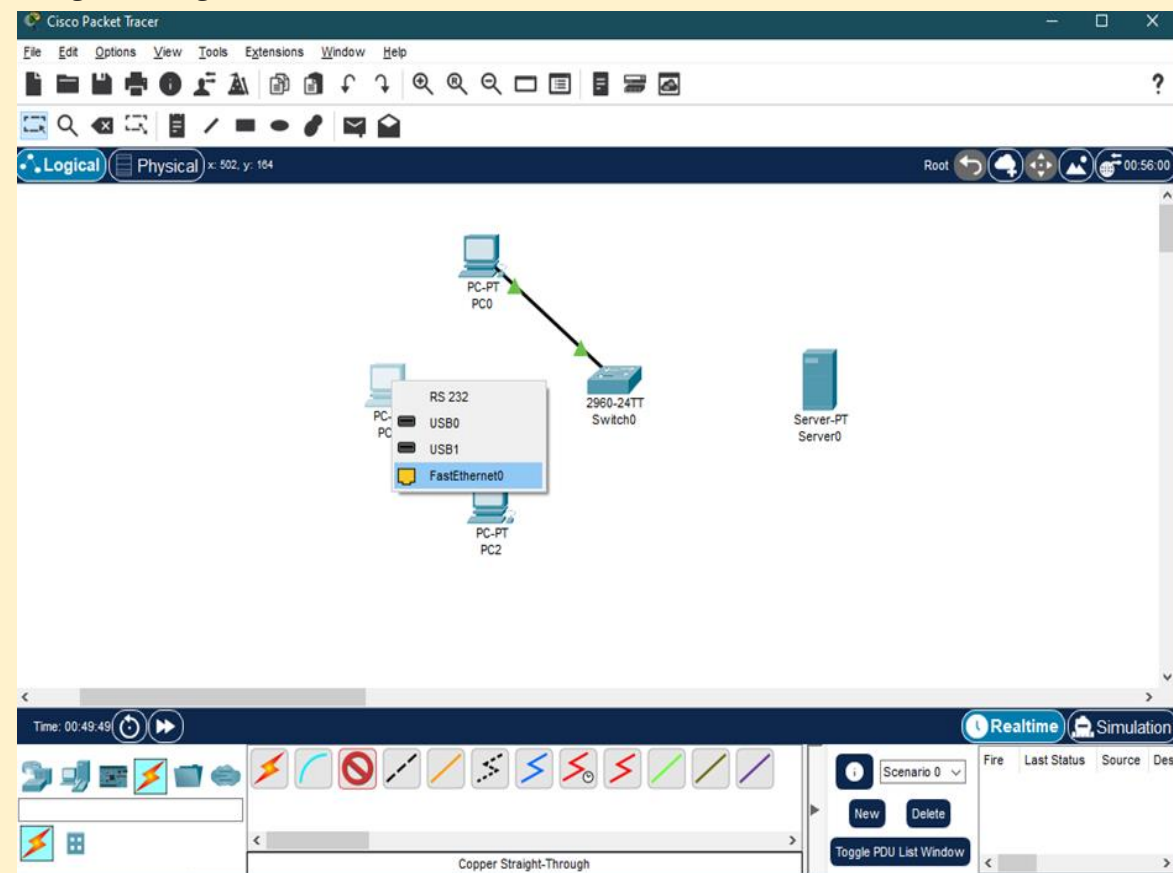


And then the connection looks like this.

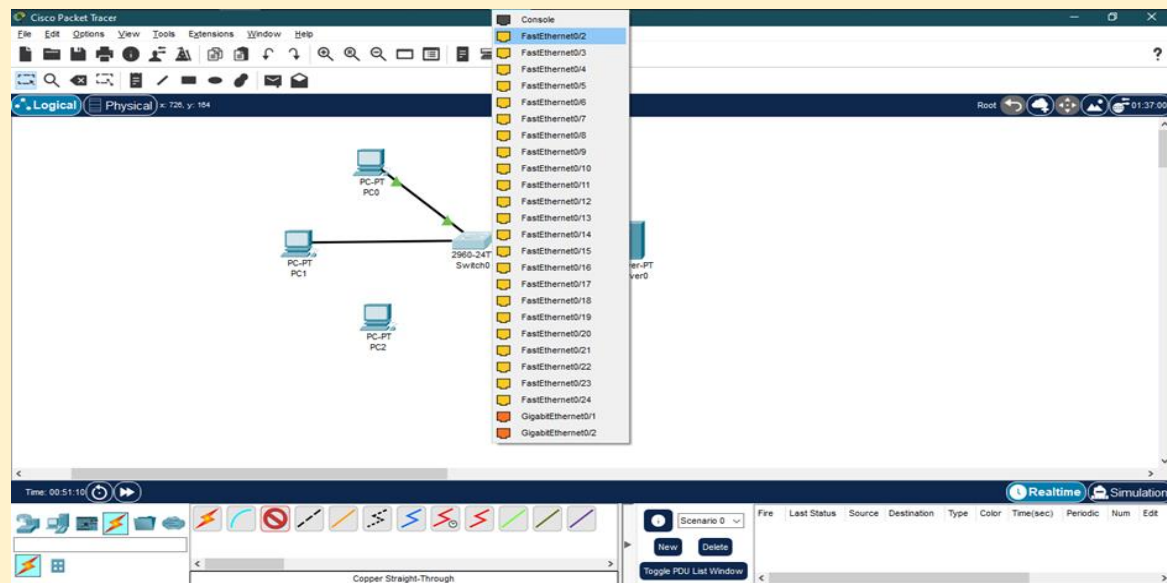


Like this, we connect the rest of the PCs.

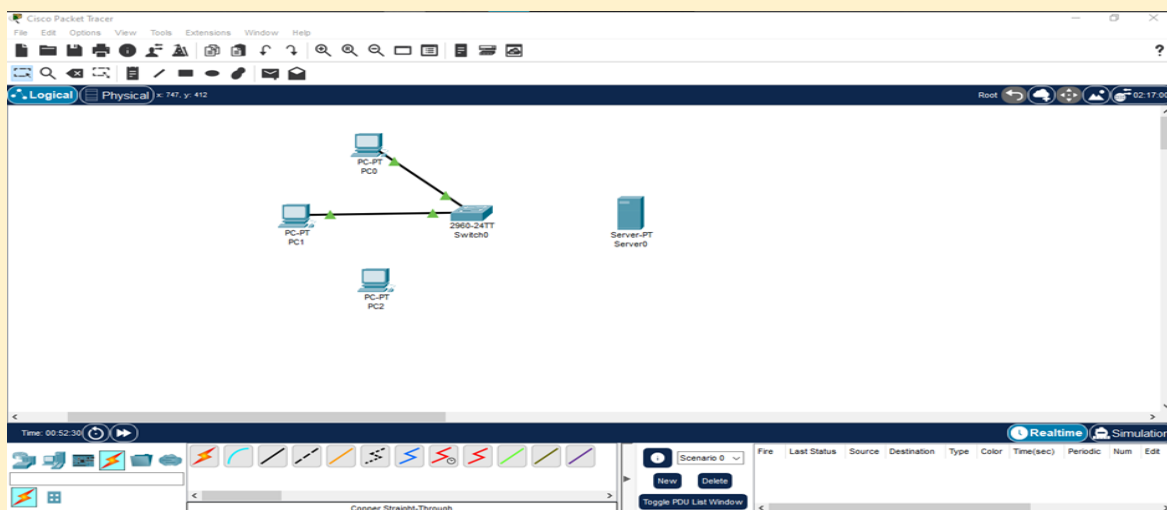
Step 3: Connect PC1 to the switch using copper-straight-through wire. We select copper-straight-through wire and click on PC1. Here we select FastEthernet0.



Then we click the switch and choose the FastEthernet0/2 interface for PC1.

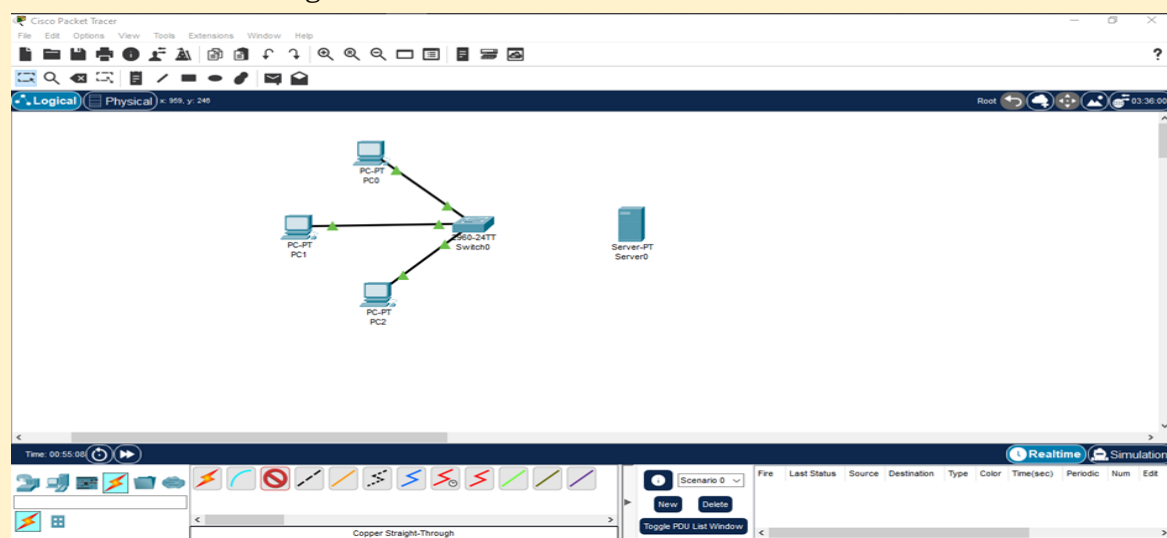


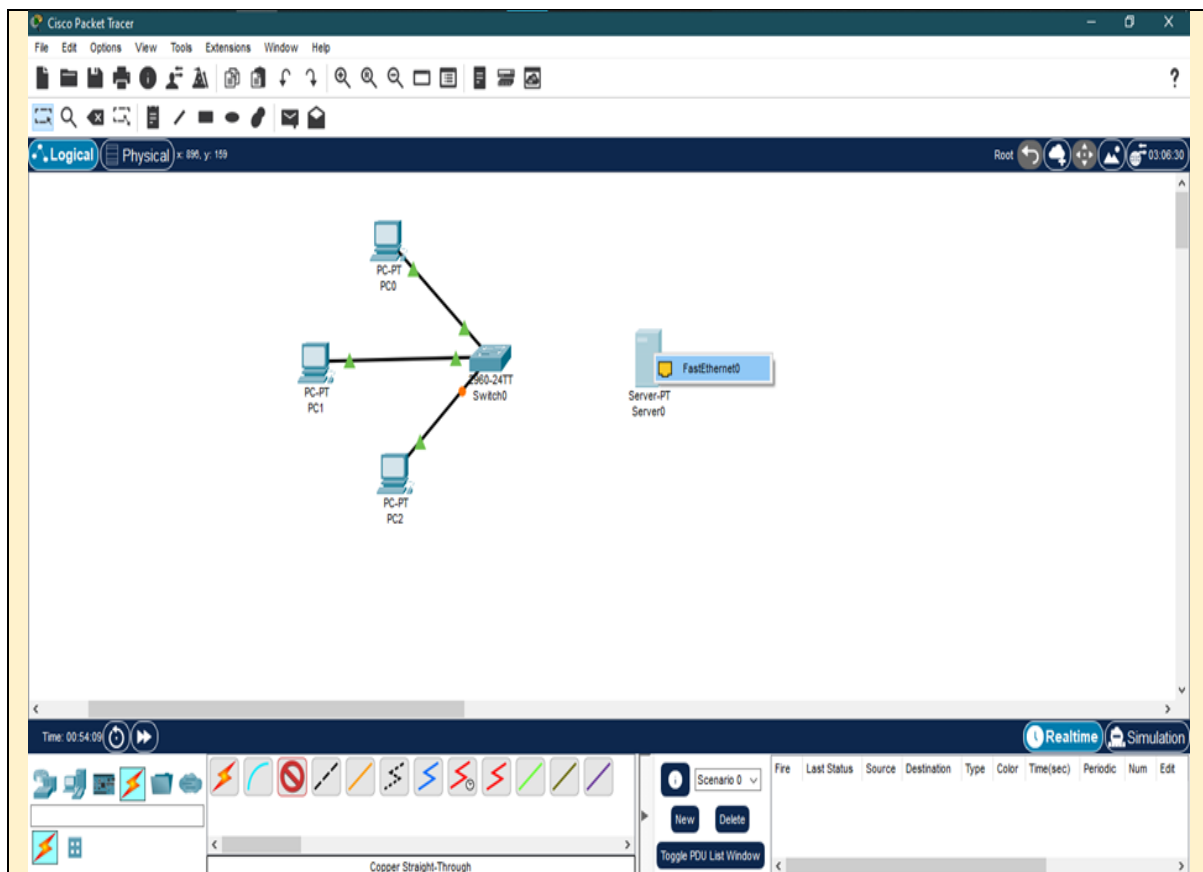
And the connection looks like this.



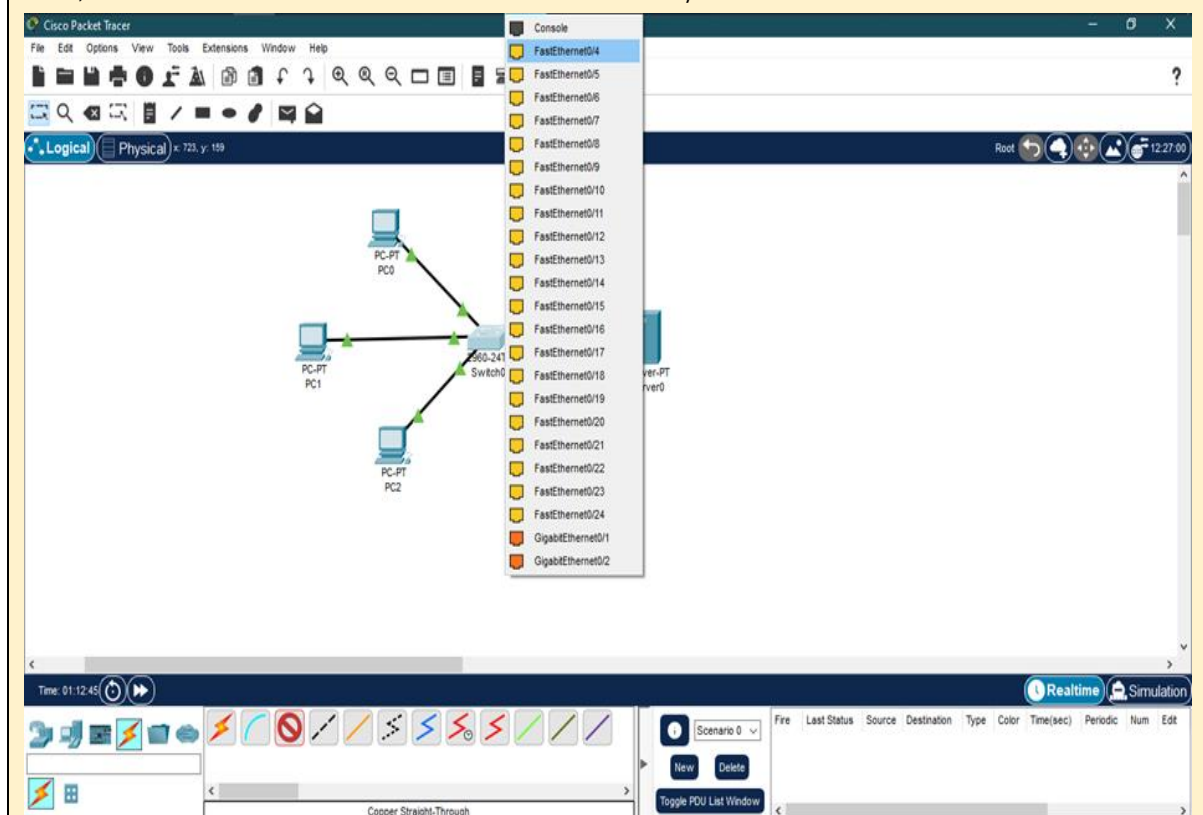
As with this, we also connect PC2 to the switch.

Step 4: Now, connect the server to the switch using a straight-through copper wire. Click on FastEthernet0 to begin.

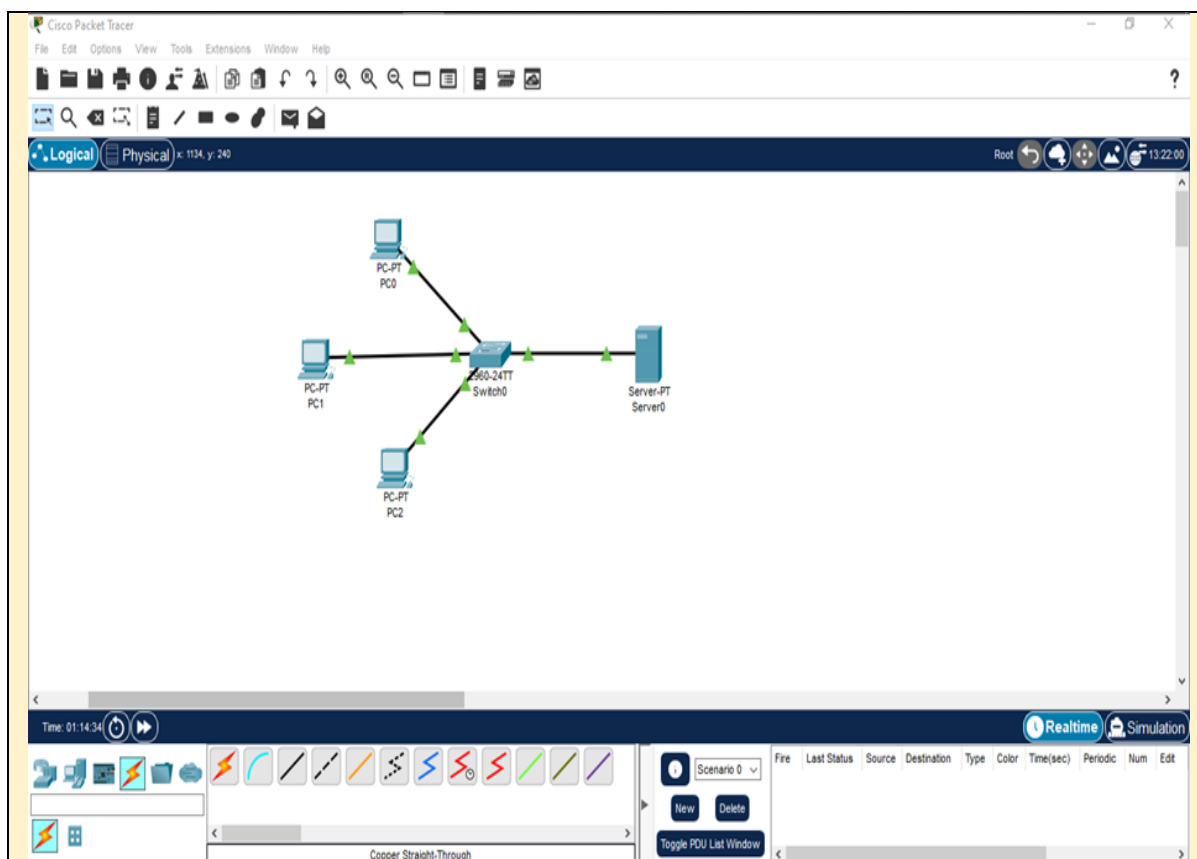




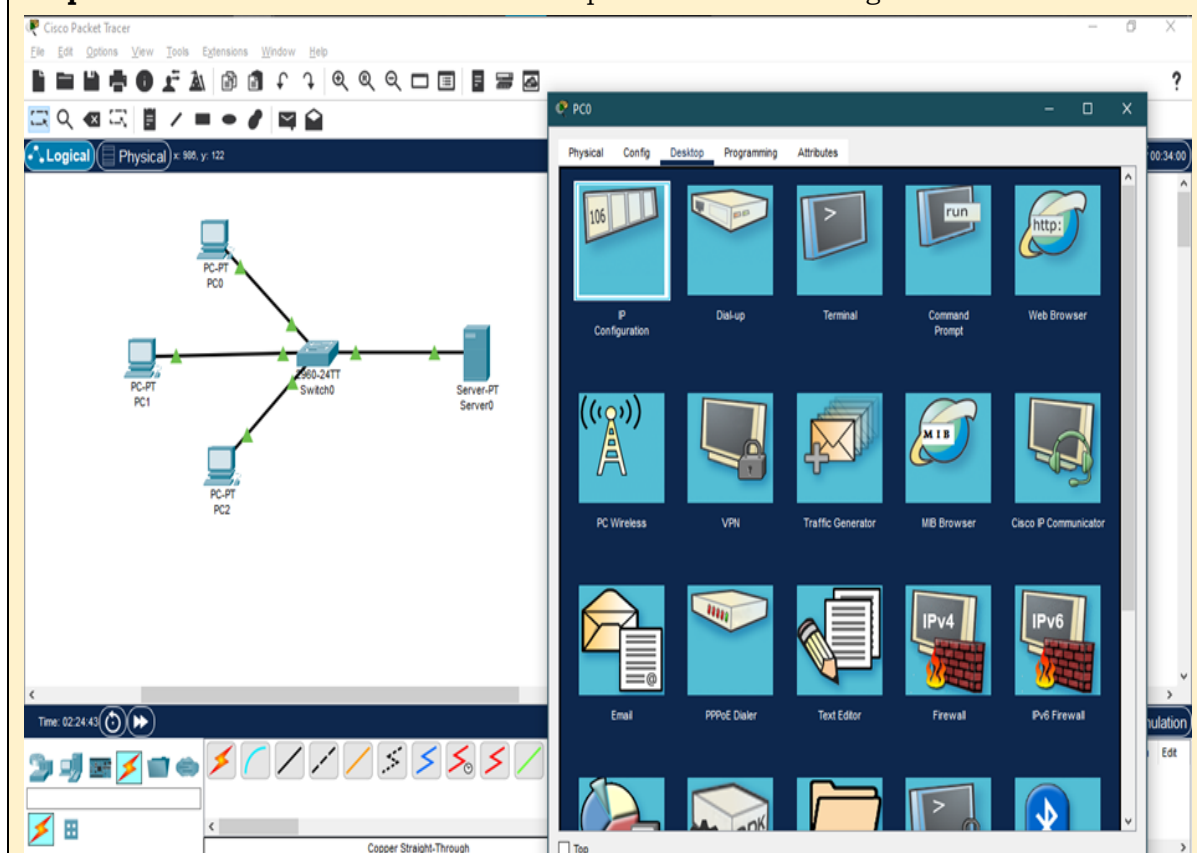
Now, click the switch and select the FastEthernet0/4 network interface.



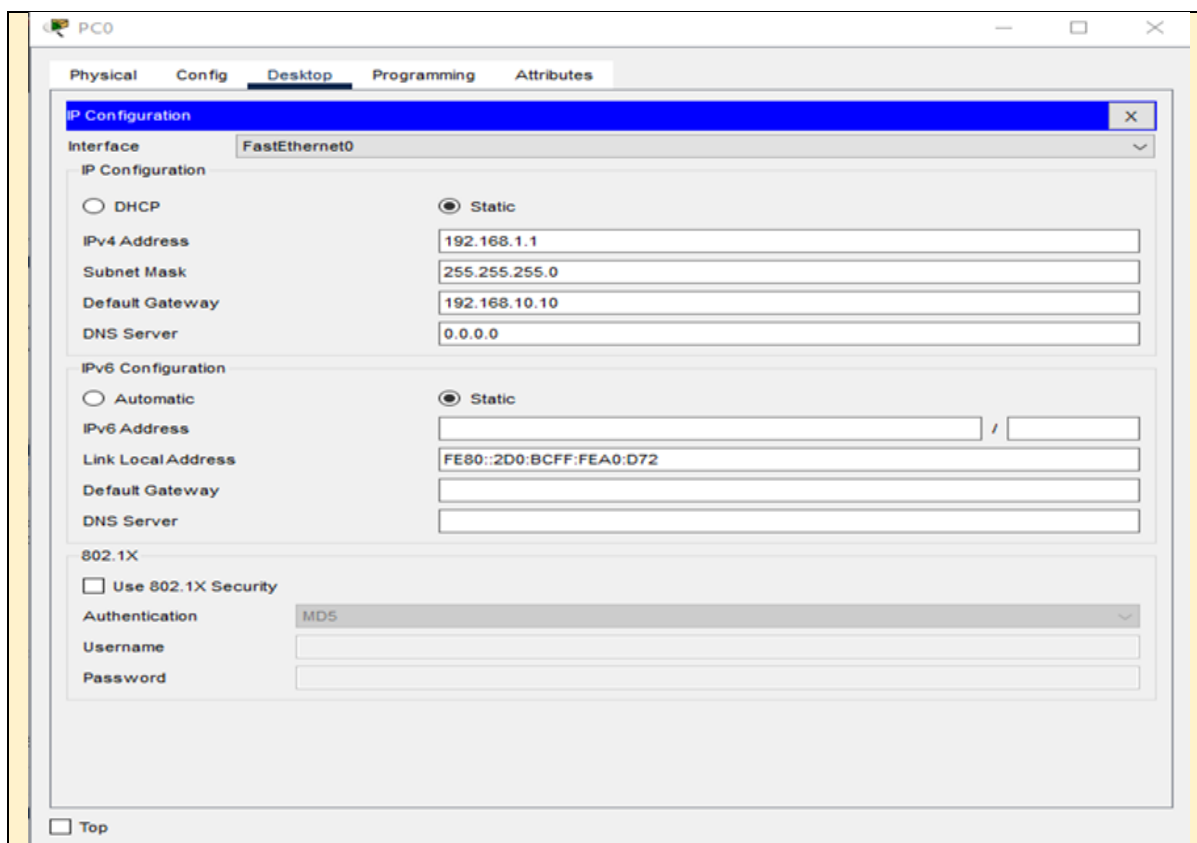
And now the network looks like this.



Step 5: Click on PC0 and select the Desktop icon. Select IP configuration.

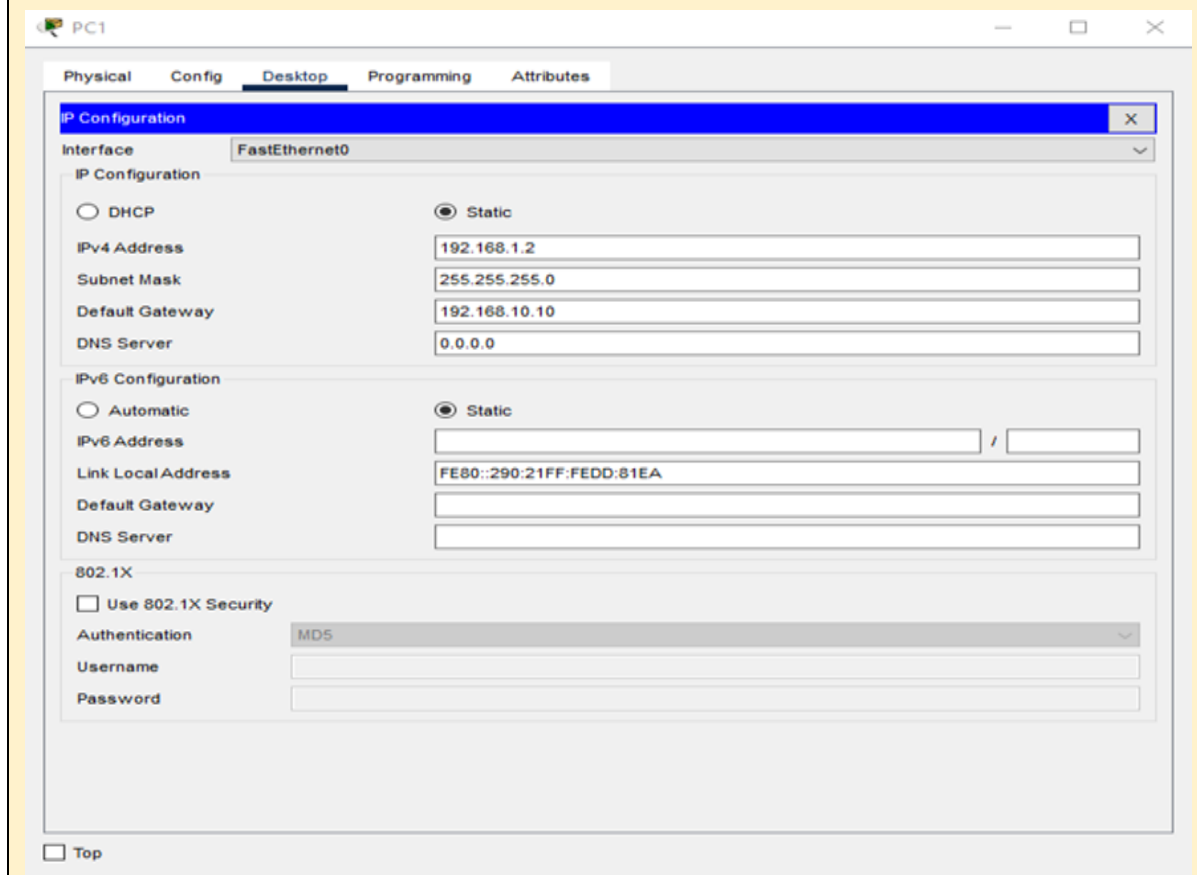


Assign an IPv4 address, subnet mask and default gateway. Here, we assign IPv4 address 192.168.1.1, subnet mask 255.255.255.0 and default address 192.168.10.10.

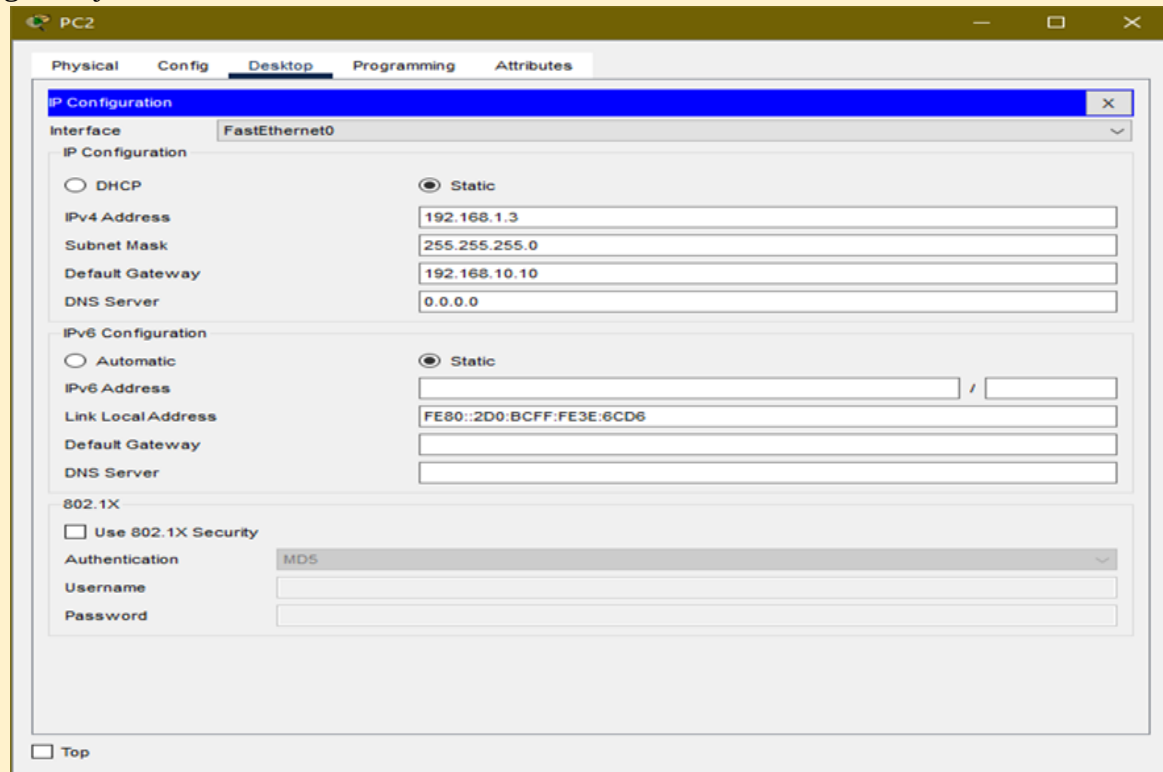


Step 6: Same as assigning PC1 an IP address, subnet mask and default address.

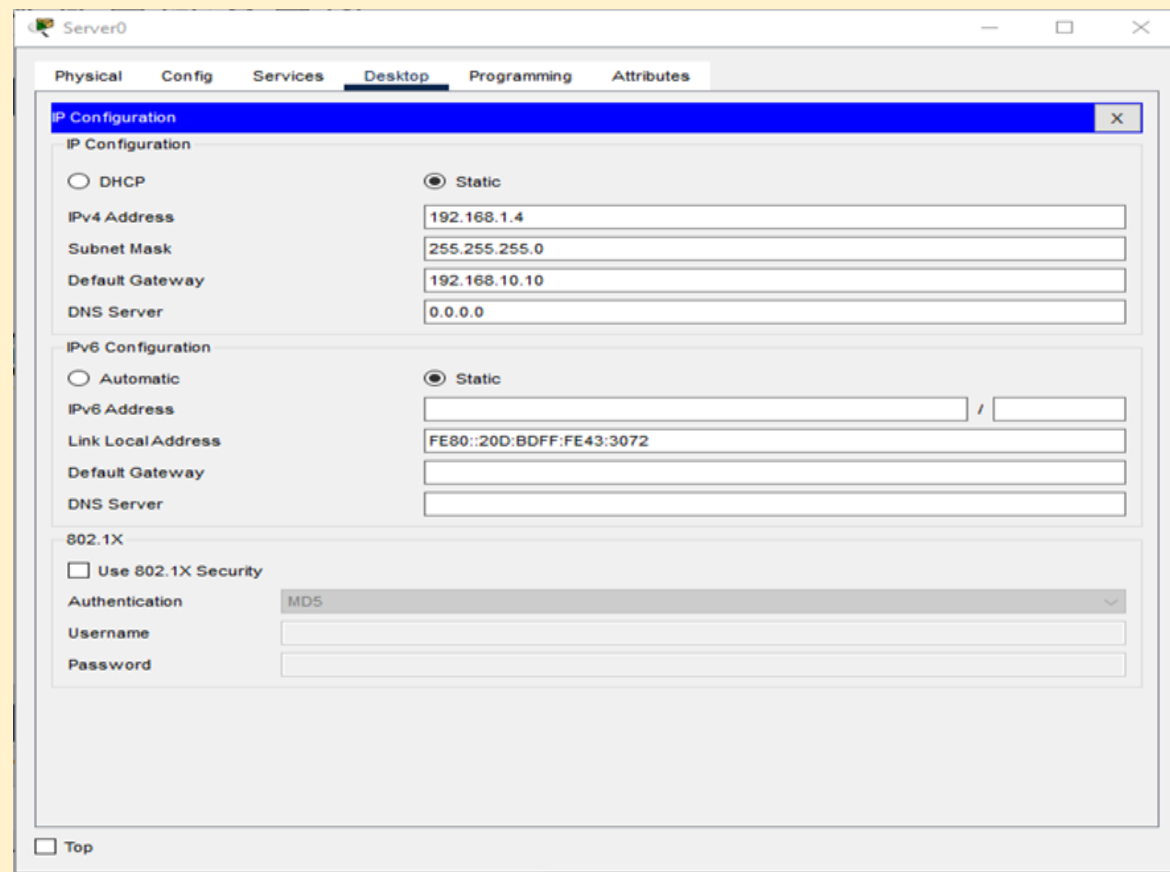
Here, we give IPv4 address 192.168.1.2, subnet mask 255.255.255.0 and default address 192.168.10.10.



Step 7: Assign an IPv4 address, subnet mask, and default gateway to PC2. For example, assign the IPv4 address 192.168.1.3, the subnet mask 255.255.255.0, and the default gateway 192.168.10.10.

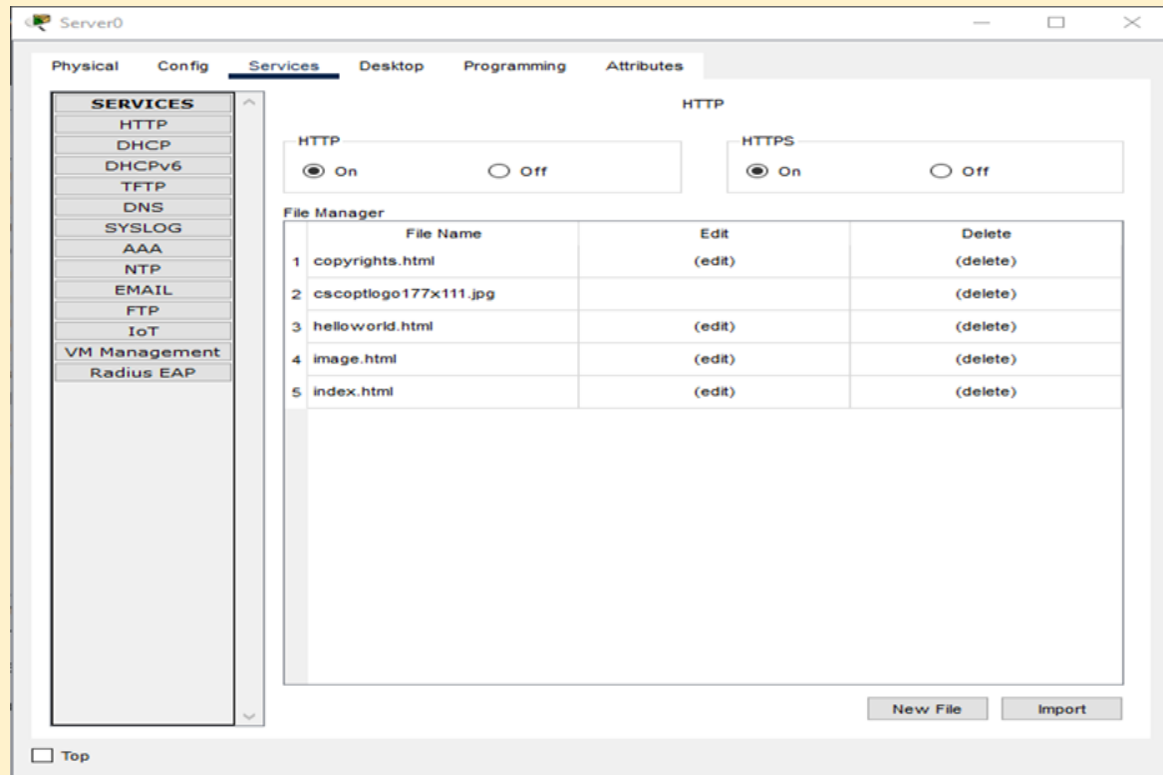


Step 8: Now, assign IPv4 address, subnet mask and default address of server.

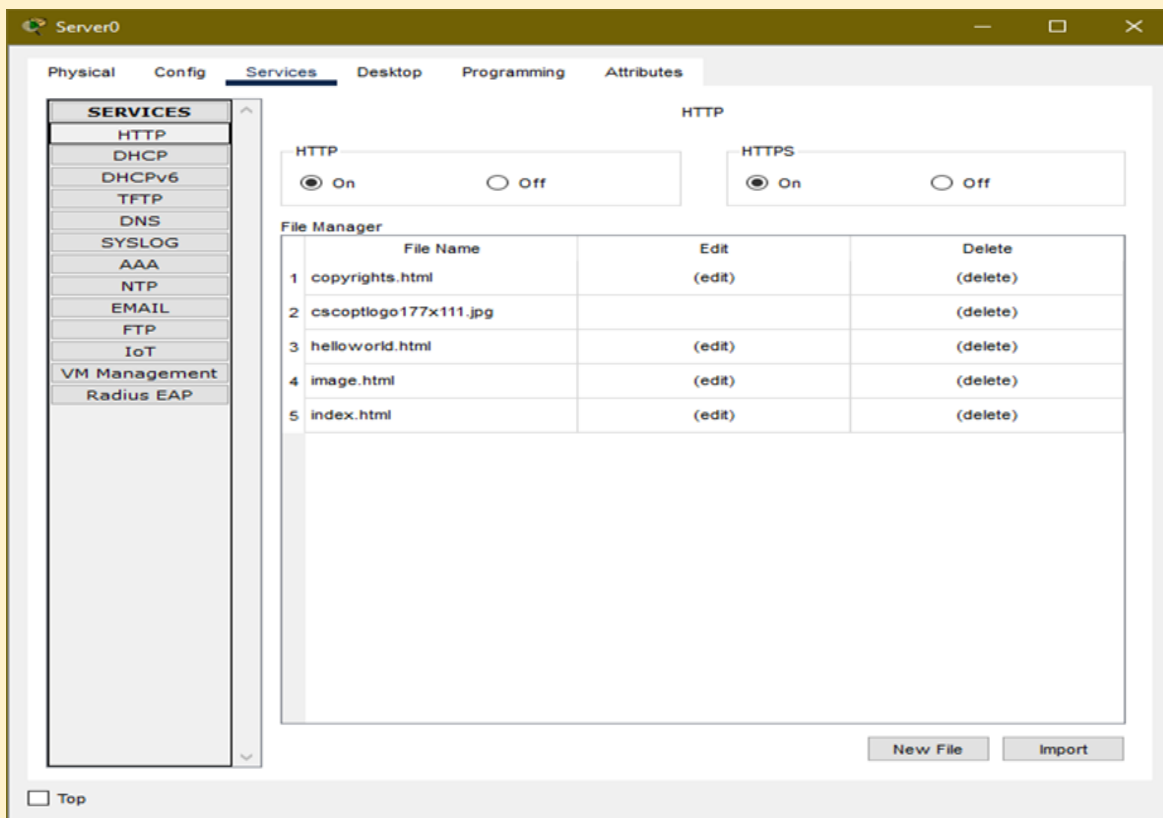


Here, we assign ipv4 address to all devices.

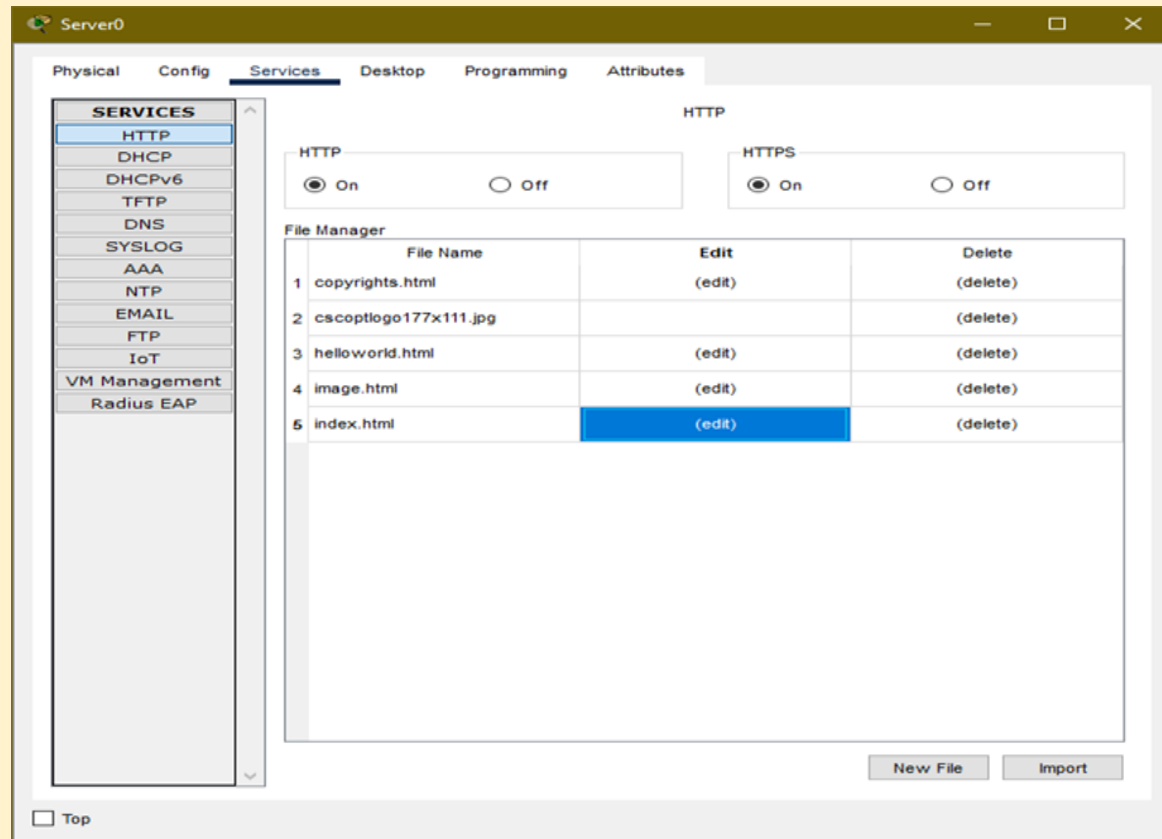
Step 9: Now, click on server and navigate to services tab.



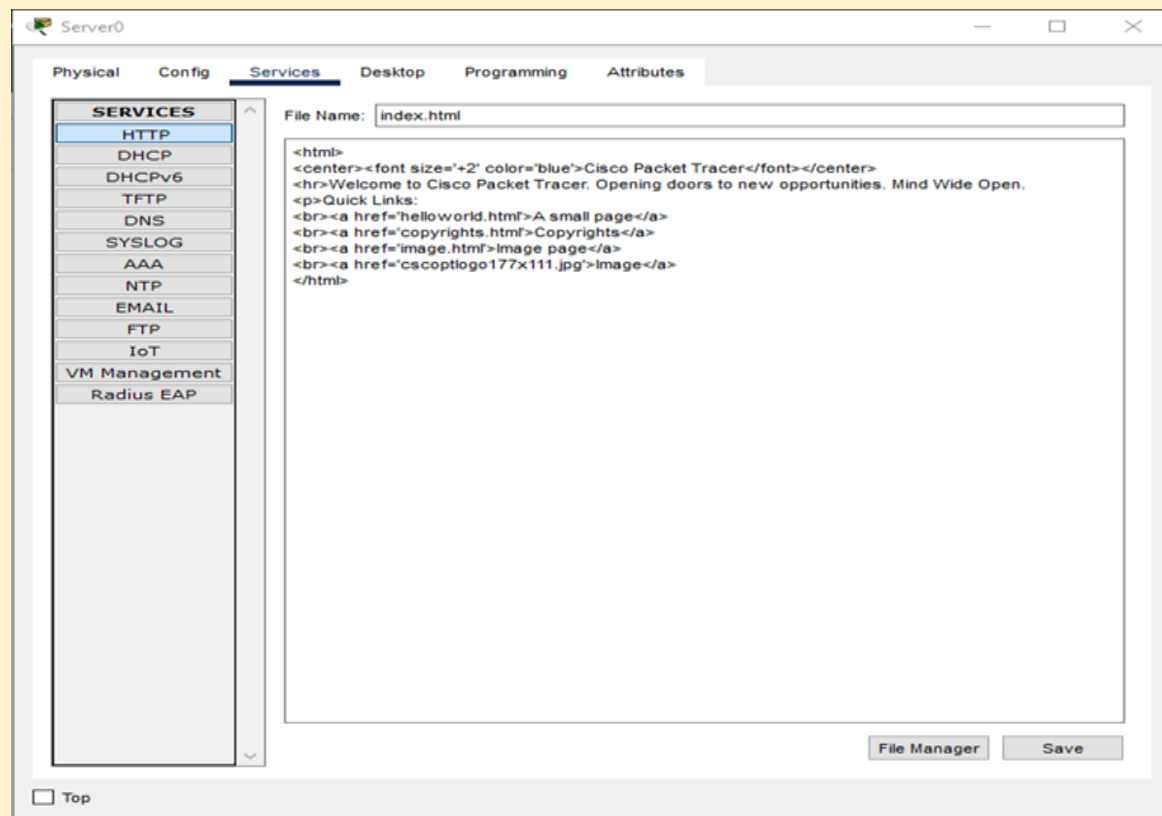
Step10: Now, click on HTTP icon which is on left side.



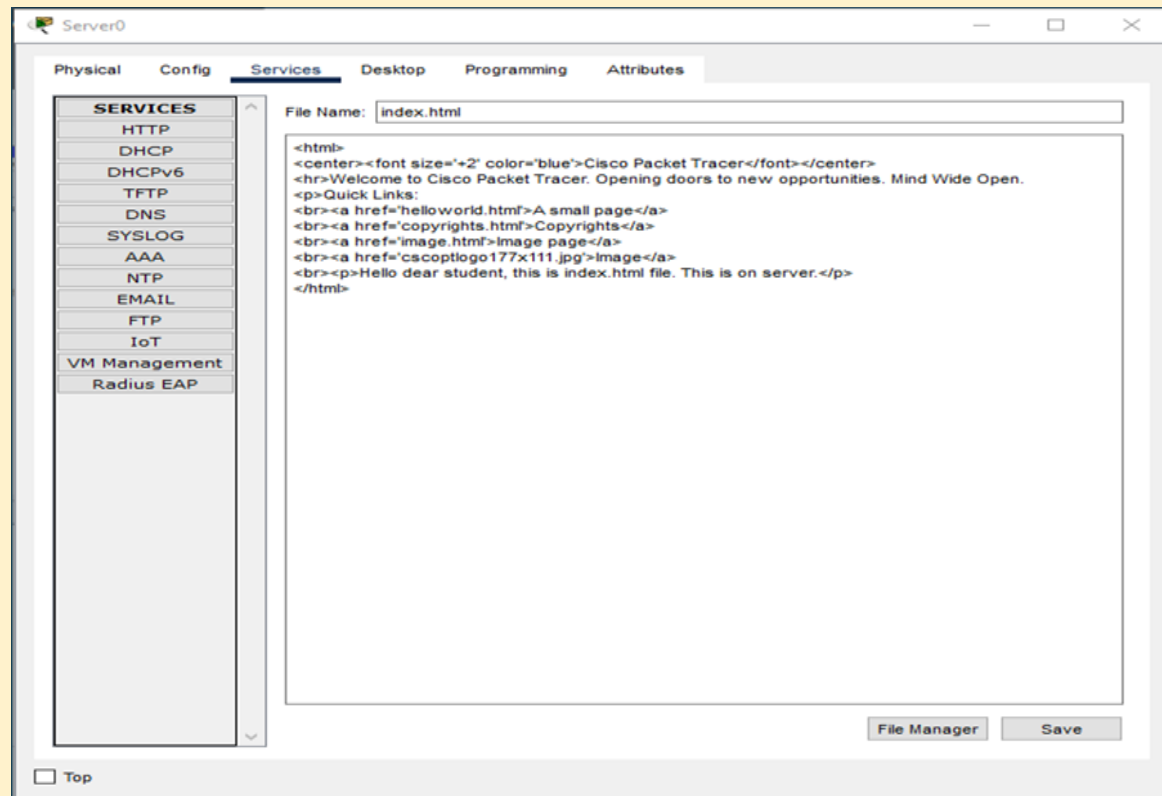
Step 11: Here, we can edit the index.file. You can add or modify any file as needed. Make sure your HTTP and HTTPS should be enabled.



This file will appear like this.

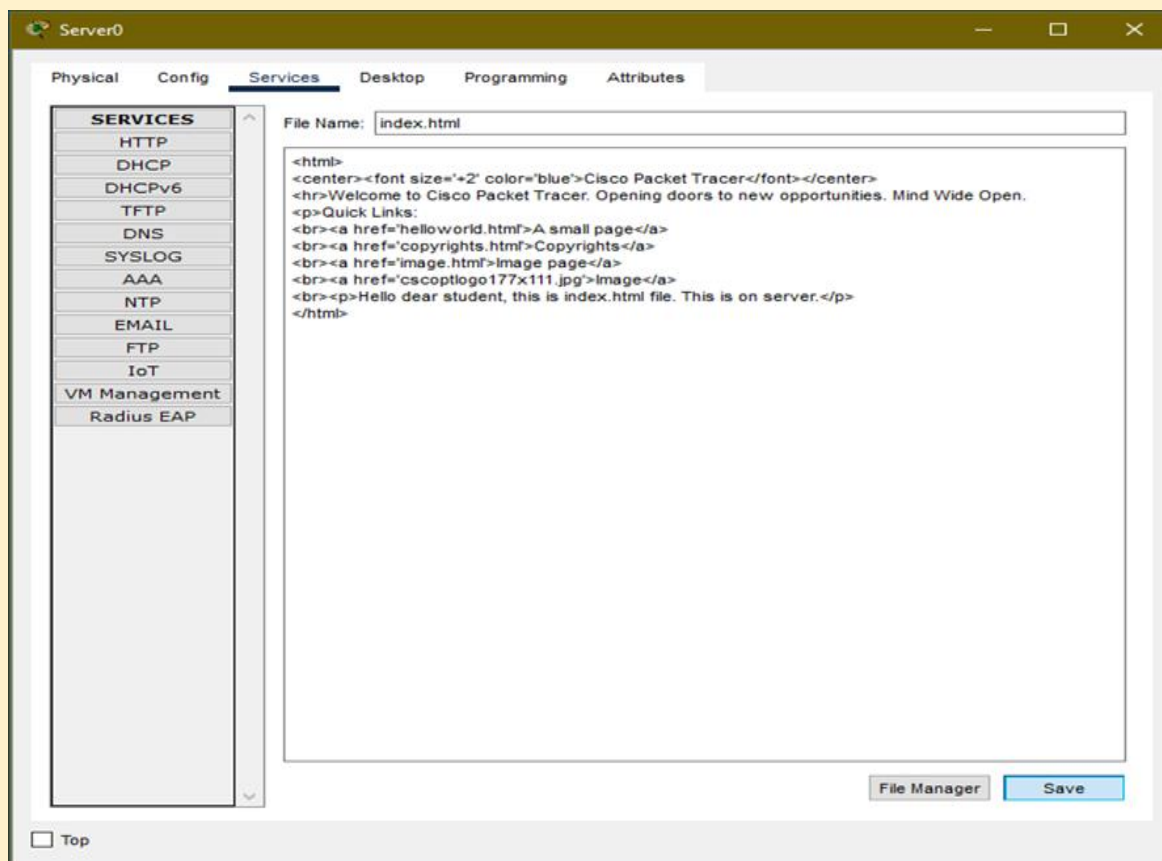


We can edit this code according to our preferences.

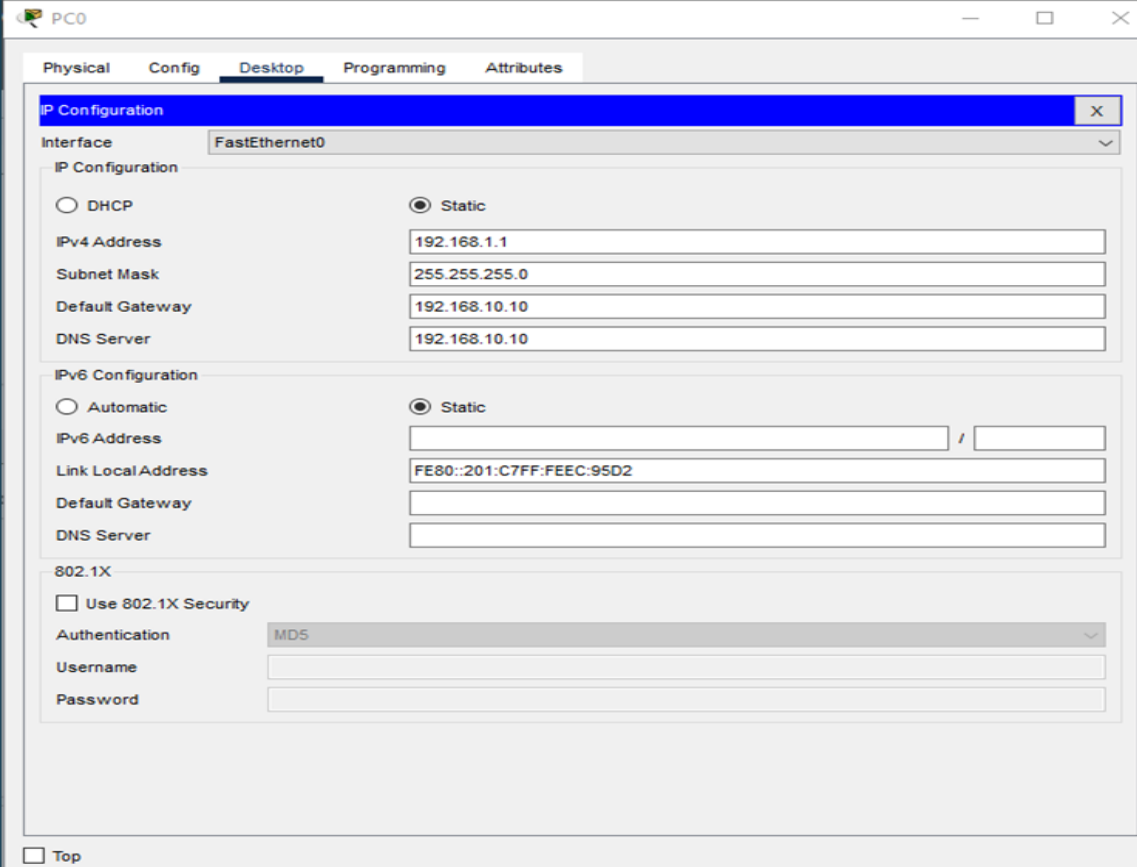


Here, we add a line 'Hello dear student, this is index.html file. This is on server. You can write anything here.

Now, save the file by clicking the Save button.

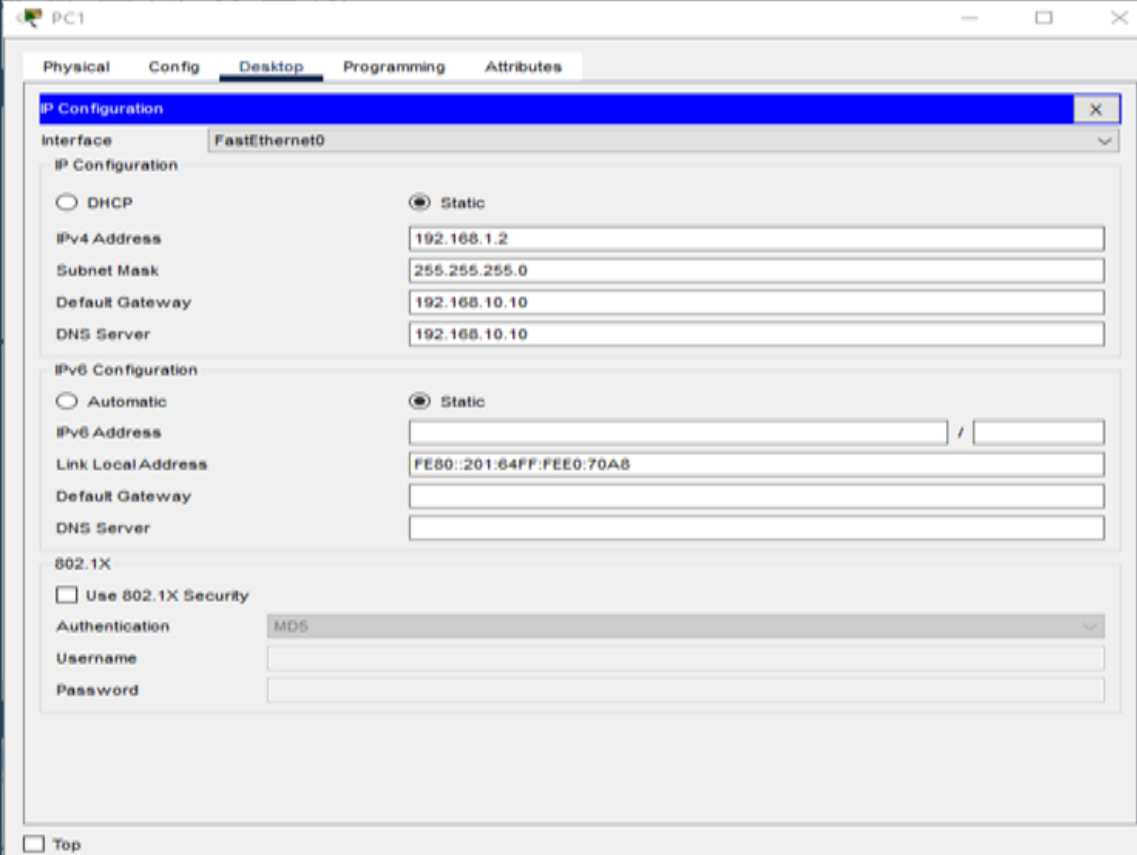


Step 12: Now assign DNS server to all devices. We assign here 192.168.1.10 as a DNS server in PC0, PC1, PC2 and server also.



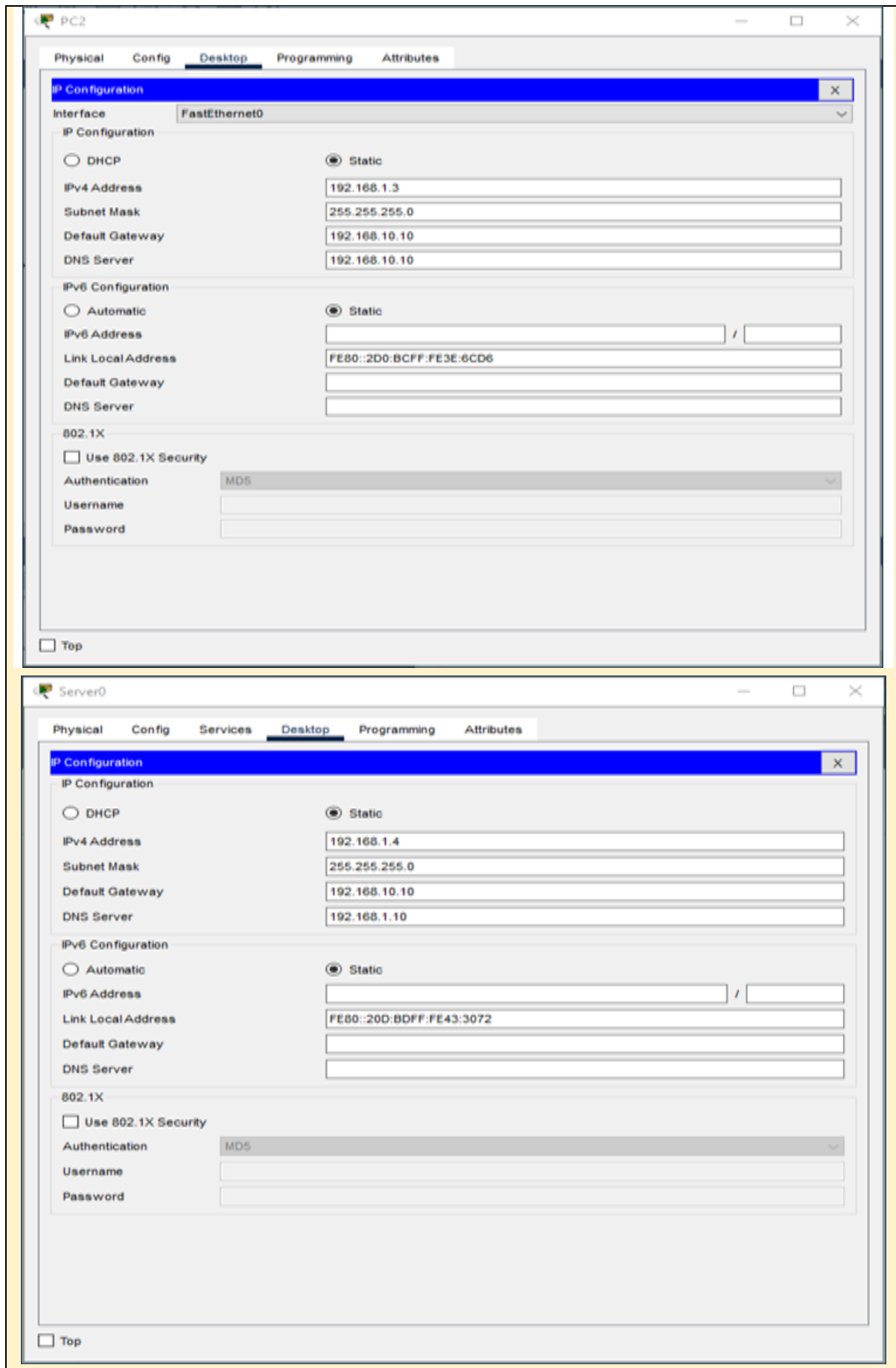
The screenshot shows the 'IP Configuration' window for PC0. The 'Interface' is 'FastEthernet0'. Under 'IP Configuration', 'Static' is selected. The fields are filled with: IPv4 Address: 192.168.1.1, Subnet Mask: 255.255.255.0, Default Gateway: 192.168.10.10, and DNS Server: 192.168.10.10. Under 'IPv6 Configuration', 'Static' is selected, with IPv6 Address and Link Local Address fields. The Link Local Address is set to FE80::201:C7FF:FEEC:95D2. Under '802.1X', 'Use 802.1X Security' is unchecked, and 'Authentication' is set to MDS.

IP Configuration	
Interface: FastEthernet0	
IP Configuration	
☐ DHCP	☒ Static
IPv4 Address	192.168.1.1
Subnet Mask	255.255.255.0
Default Gateway	192.168.10.10
DNS Server	192.168.10.10
IPv6 Configuration	
☐ Automatic	☒ Static
IPv6 Address	
Link Local Address	FE80::201:C7FF:FEEC:95D2
Default Gateway	
DNS Server	
802.1X	
☐ Use 802.1X Security	
Authentication	MDS
Username	
Password	

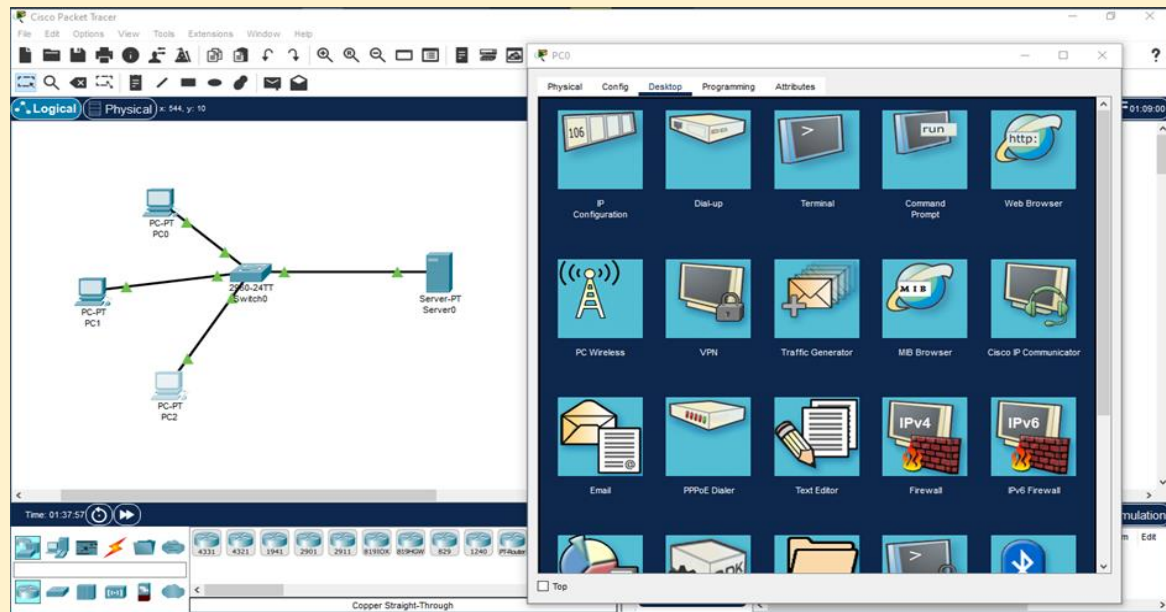


The screenshot shows the 'IP Configuration' window for PC1. The 'Interface' is 'FastEthernet0'. Under 'IP Configuration', 'Static' is selected. The fields are filled with: IPv4 Address: 192.168.1.2, Subnet Mask: 255.255.255.0, Default Gateway: 192.168.10.10, and DNS Server: 192.168.10.10. Under 'IPv6 Configuration', 'Static' is selected, with IPv6 Address and Link Local Address fields. The Link Local Address is set to FE80::201:64FF:FEE0:70A8. Under '802.1X', 'Use 802.1X Security' is unchecked, and 'Authentication' is set to MDS.

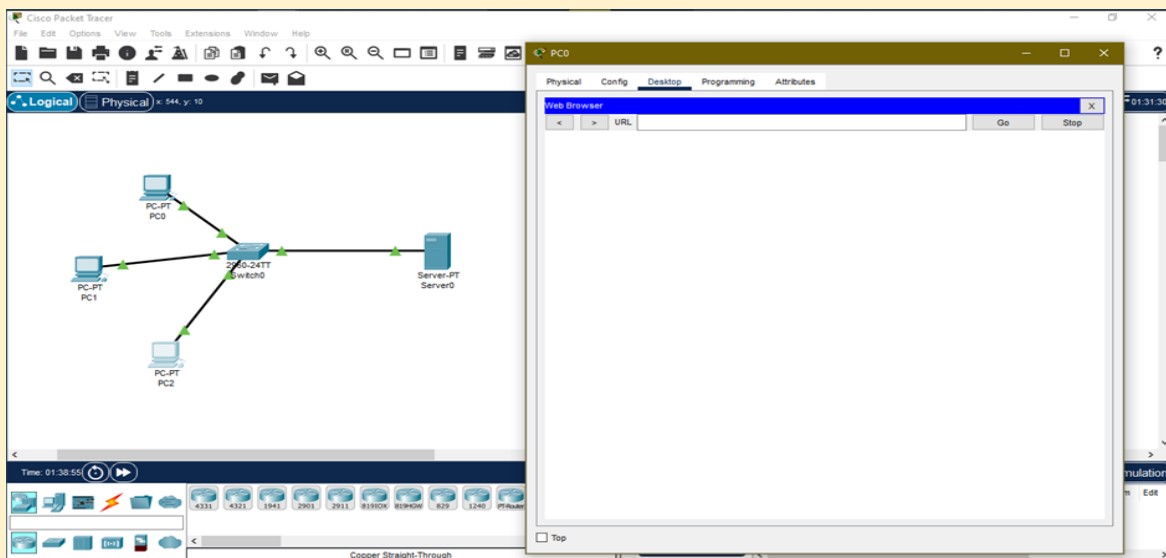
IP Configuration	
Interface: FastEthernet0	
IP Configuration	
☐ DHCP	☒ Static
IPv4 Address	192.168.1.2
Subnet Mask	255.255.255.0
Default Gateway	192.168.10.10
DNS Server	192.168.10.10
IPv6 Configuration	
☐ Automatic	☒ Static
IPv6 Address	
Link Local Address	FE80::201:64FF:FEE0:70A8
Default Gateway	
DNS Server	
802.1X	
☐ Use 802.1X Security	
Authentication	MDS
Username	
Password	



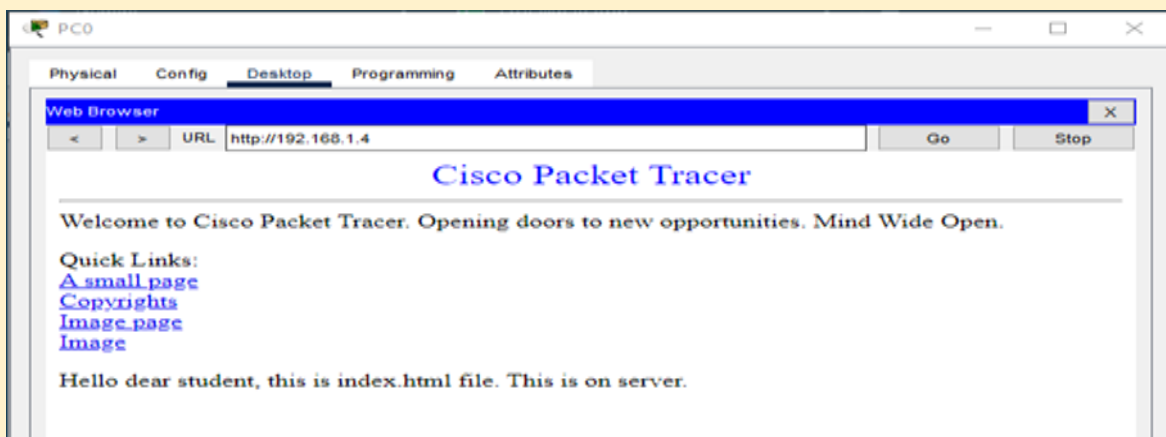
Step 13: Now open any PC. Here, we will open PC0 and navigate to the desktop tab.



Open Web Browser.



Here, we write server IP address as URL. and click on Go.



As you can see, this is the index.html file and we edited it by adding a line. It is shown here.

Assignment 3.1.

- (a) List all Cloud-Based Network Management Models
- (b) List all Key Network Management Functions

Summary

- Networking devices help manage and direct data within a network.
- Weak signals are amplified using repeaters.
- Data is broadcast to all connected devices through hubs.
- Data is forwarded only to the destination using switches.
- Bridges connect LAN segments and filter traffic.
- Routers connect different networks and route packets.
- Modems convert digital signals to analog and vice versa.
- Transmission media can be either guided (wired) or unguided (wireless).
- Guided media includes UTP, STP, fiber optic cables, and coaxial cables.
- Radio waves, microwaves, and infrared are types of unguided media.

Check your progress**A. Multiple Choice Questions**

1. What is the primary goal of Network Management?
 - a) To only fix problems quickly when they arise.
 - b) To continuously collect and analyze data to improve performance, reliability, and security.
 - c) To manually configure every device in the network.
 - d) To replace all hardware components with software.
2. Which key network management function is responsible for detecting and alerting network administrators about issues like failed devices or broken cables?
 - a) Configuration Management
 - b) Accounting Management
 - c) Fault Management
 - d) Performance Management
3. Keeping track of every device and its settings like IP addresses and passwords falls under which network management function?
 - a) Security Management
 - b) Accounting Management
 - c) Configuration Management
 - d) Performance Management
4. Which network management function aims to identify how much bandwidth individual users or applications are consuming?
 - a) Fault Management
 - b) Accounting Management
 - c) Performance Management
 - d) Security Management
5. What is a significant challenge faced by network managers due to the increasing number of smart devices like smart TVs and smartwatches connecting to the internet?
 - a) Reduced cybersecurity threats
 - b) Simpler network technology
 - c) Growing networks (Internet of Things)

- d) Decreased need for remote work solutions
6. Which network management model involves an organization building and managing its own cloud infrastructure, offering high security and full control?
- Public Cloud
 - Hybrid Cloud
 - Private Cloud
 - Distributed Cloud
7. A small online bookshop that rents cloud services from a large company like Amazon to run its website and manage customer orders is an example of which cloud-based network management model?
- Private Cloud
 - Hybrid Cloud
 - On-premises
 - Public Cloud
8. What is a key advantage of cloud-based network management, especially beneficial for supporting remote workforces?
- Requires significant upfront capital expenditure on hardware.
 - Decreased reliance on internet connectivity.
 - Remote access and monitoring capabilities.
 - Manual software updates and security fixes.
9. Which of the following is NOT typically a benefit of migrating to cloud-based network management?
- Cost-effectiveness due to reduced upfront hardware costs.
 - Automatic software and security updates handled by the provider.
 - Increased complexity in managing multiple different cloud types.
 - Enhanced scalability to adapt to changing business needs.
10. In the context of cloud-based network management migration, what does "Saying Goodbye to the Old System" usually involve?
- Physically destroying all old network equipment.
 - Turning off or repurposing old on-premises management servers.
 - Immediately transferring all data to the cloud in one go.
 - Disconnecting all existing network cables.

B. Fill in the Blanks

- A _____ boosts a weak network signal.
- A _____ connects different types of networks.
- _____ media uses electromagnetic waves to transmit data.
- A _____ is a central device in a star topology.
- _____ cable uses light to transmit data.
- Bluetooth and Wi-Fi are examples of _____ transmission.
- The full form of NIC is _____.
- _____ is used to filter data using MAC addresses.
- _____ cables are most vulnerable to EMI.
- Coaxial cable is commonly used in _____ systems.

C. True or False

1. A hub sends data to only the intended device.
2. Switches are smarter than hubs.
3. Modems operate at the application layer.
4. UTP is cheaper than STP.
5. Infrared signals can pass through walls.
6. Routers operate at the Network layer.
7. STP has shielding to reduce interference.
8. Optical fibre is immune to electromagnetic interference.
9. Bluetooth operates in long-range communication.
10. A bridge connects devices using IP addresses.

D. Short Answer Questions

1. Define a switch and how it functions in a network.
2. Discuss the primary role of a router.
3. Differentiate between hub and switch.
4. Explain what a bridge does in a network.
5. Describe the function of a repeater.
6. What is the difference between UTP and STP cables?
7. List two advantages of using optical fiber.
8. What is a modem and why is it needed?
9. Name two examples of unguided transmission media.
10. What is the role of a NIC?

Answer Key**A. Multiple Choice Questions**

1. b, 2. c, 3. c, 4. b, 5. c, 6. c, 7. d, 8. c, 9. c, 10. b

B. Fill in the blanks:

1. Repeater, 2. Router, 3. Wireless, 4. Hub (or Switch), 5. Fiber optic, 6. Wireless, 7. Network Interface Card, 8. Bridge, 9. Unshielded Twisted Pair (UTP), 10. Cable television (CATV)

C. True or False

1. False, 2. True, 3. False, 4. True, 5. False, 6. True, 7. True, 8. True, 9. False, 10. False

Chapter-4**Switching, Routing, and Network Components**

Riya was curious about how the internet works. One day, her cousin showed her a small device with blinking lights and said, “This is a switch. It connects all the computers in our home and sends data to the right one—like passing notes to the correct friend in class.” Then he pointed to another device. “And this is a router. It connects us to the internet and decides the best path for the data to travel—like choosing the fastest road to reach your school.” Riya smiled, “So switches and routers are like traffic managers for data?” “Exactly,” her cousin laughed. “And together with cables, servers, and other network components, they help all our devices talk to each other.”



In this chapter, we will learn the detailed concept about switching, routing and various network components.

4.1 Switching

Switching allows the transmission of data packets between devices in a network using a specific device called a switch.

For example, a computer user accesses data from another computer on the Internet whenever they request to open a website. These requests are handled through the switching of data packets. This means that after a data packet is generated at the physical layer, switching is the next crucial step in data communication.

The switch primarily processes incoming data packets from the source computer or network and determines the correct port to forward the packet to the target computer or network. It does this efficiently by maintaining a switch table. Network switches are more efficient than hubs and repeaters. They simplify tasks, manage switch tables to reduce network overload, and significantly improve overall network performance.

The three major types of switching are listed below and illustrated in Figure 4.1.

- a. Message Switching
- b. Packet Switching
- c. Circuit Switching

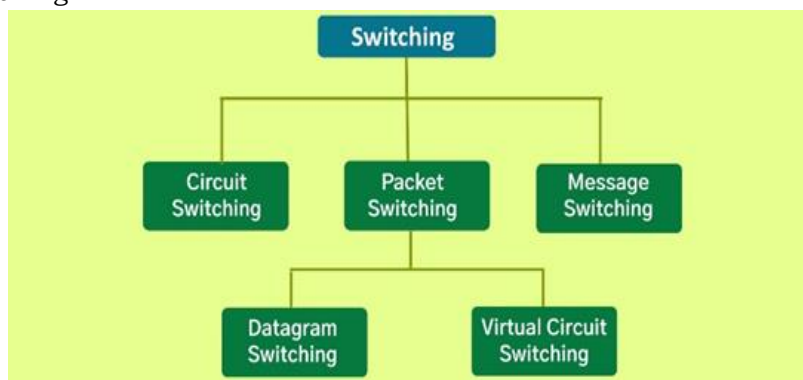


Fig 4.1. Types of switching

1. Message Switching:

This is an old method of switching that is rarely used today. In message switching, the entire message or data block is sent through the network at once. This method was slow and inefficient because each message had to be stored and forwarded at every point along the path.

2. Circuit Switching:

In this method, a path is established between the sender and receiver before any data is transmitted. The path remains open for the entire duration of the communication. It uses the full network bandwidth for the connection, similar to how traditional phone calls work.

3. Packet Switching:

In this method, data is divided into smaller units called packets. Each packet travels through the network independently, using any available path. This method is faster and more efficient, and it is how the Internet and most modern networks operate today. Every packet includes additional information such as the destination address and routing instructions.

3a. Datagram Packet Switching:

In this type of packet switching, each packet is handled independently and can take different paths to reach the destination. No dedicated connection is established before transmission. While this makes the system flexible and fast, it can also result in packets arriving out of order or even being lost.

3b. Virtual-Circuit Packet Switching:

In this method, a "virtual" path is established between the sender and receiver before any data is transmitted. All packets then follow this predefined path, making the communication more reliable and reducing the chance of packet loss.

4.2 Method of Routing

Network routing is the process of selecting a path between two or more networks. In today's world, where everyone heavily relies on the Internet, efficient routing is crucial for smooth communication. As shown in Figure 4.2.

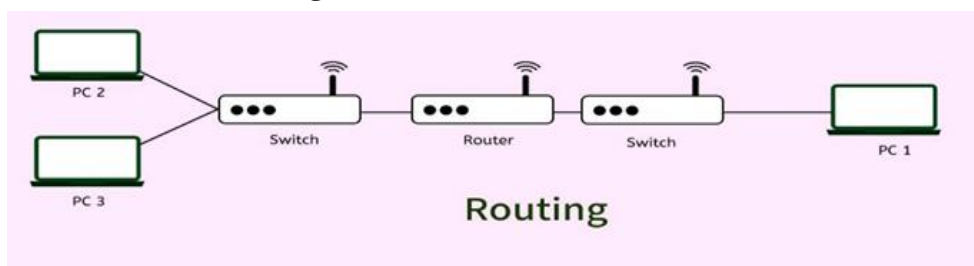


Fig 4.2. Routing

This process adopts routes along which packets are transferred from the host to the target destination with the help of the Internet Protocol (IP) in packet-switching networks.

Sending or directing data packets from one node to another is known as routing. Note that the term node here refers to a network device called a router.

Routing is a crucial mechanism that enables the transfer of information from one location to another across a network. The network type can be LAN, WAN, or MAN, depending on the scale and structure.

4.3 LAN and WAN Technologies

1. Ethernet (Wired LAN)

This is the most prevalent and reliable technology used in these networks. It uses special cables, often called Ethernet cables (like Cat5e or Cat6), to physically connect printers, computers, and other nodes to a network. These cables plug into devices through Ethernet ports, typically found on the back of desktop computers or in network switches. Ethernet

networks are known for being fast and stable because the data travels through wires instead of the air. This reduces signal interference and increases speed and reliability. The speed range of Ethernet can be from 100 megabits per second (Mbps) to 10 gigabits per second (Gbps), making it ideal for transferring large files or streaming high-quality videos without lag. Most schools and offices use Ethernet for critical systems like teacher computers or file servers, where speed really matters.

2. Wi-Fi (Wireless LAN)

Wi-Fi is a wireless technology that allows various nodes like laptops, smartphones, and tablets to connect to a LAN without using any wires. A router is connected to the internet and acts as a central point where all the wireless devices communicate. Wi-Fi is highly convenient because it allows people to move around freely with their devices while staying connected. However, the strength and speed of a Wi-Fi connection can vary depending on how far you are from the router, how many walls are in the way, and how many people are using it at the same time. Even though it is slightly slower and less secure than wired Ethernet, Wi-Fi remains extremely popular in homes, schools, and public places like coffee shops and airports.

3. Switches

A switch is a device used in wired LANs to connect multiple devices and computers together. You can think of a switch as a smart traffic controller that knows exactly where each piece of data needs to go. When a device on the network sends information, the switch reads the destination and sends it only to the device it is intended for, rather than broadcasting it to every device on the network. This makes communication faster and more efficient. Switches are usually placed in wiring closets or network rooms and have multiple ports where Ethernet cables are plugged in. In a classroom or office, all the computers might be connected to a single switch, which then connects to the internet through a router.

4. Leased Lines

This contributes to a dedicated, private connection between two or more locations, often rented from a telecom company. Unlike a regular internet connection that is shared with many users, a leased line is used only by the business or organization that pays for it. This means it is much more reliable, secure, and consistent in speed, which is why big companies or banks often use it to connect their branch offices or data centres. Leased lines offer high speeds and guaranteed uptime, making them ideal for handling sensitive or large-scale data transfers. However, they are also very expensive and usually only used by organizations that absolutely need constant, high-performance connections over long distances.

5. Fiber Optic

Fiber optic technology is one of the fastest and most advanced types of WAN connections available today. Instead of using electrical signals, fiber uses signals of light to send data through plastic strands or thin glass. Because light can travel incredibly fast and doesn't weaken over long distances like electricity does, fiber optic connections can reach speeds of hundreds of Mbps or even Gbps (gigabits per second). Fiber is ideal for activities like streaming, gaming, video conferencing, and business services that require high-speed data at all times. The downside is that laying fiber optic cables is expensive and time-consuming, so it's mostly available in cities or well-developed areas.

6. Satellite Internet

Satellite internet is a type of WAN technology used in places where there are no phone lines, cables, or fiber optics available—such as remote mountains, deserts, or rural villages. It works by sending signals to a satellite in space, which then bounces them back to a dish at your location. It's remarkable because it brings internet access to places where no other option exists, but it's also slower and more prone to delays due to the vast distance the signals

must travel. Weather conditions like heavy rain or snow can also interfere with the connection. Still, satellite internet is incredibly valuable for emergency services, scientific research in remote locations, or simply helping people get online in areas with no other access.

4.4 Wireless Networking

Networks that are made up of electromagnetic waves or infrared waves are known as Wireless networks.

Types of Networks

(a) Wi-Fi

Wi-Fi is the wireless variant of a wired Ethernet network and is often used in combination with it (see Ethernet). Wireless networks use radio frequencies at both ends to transmit data. Wi-Fi enables high-speed internet access without the need for cables. The term Wi-Fi stands for “Wireless Fidelity” and is commonly used to refer to wireless networking technologies. A wireless router is used to connect to the internet. When you connect to Wi-Fi, you are actually connecting to a wireless router that links your Wi-Fi-enabled devices to the internet.

(b) Bluetooth

This is a technique that allows mobile devices, PCs, and other equipment to communicate over short distances. It enables nodes to communicate with one another. Infrared offers similar capabilities as Bluetooth but comes with some disadvantages—its transmission is very slow, and devices are often not compatible with one another.

4.5 NETWORK SECURITY

What is Network Security?

Network security is the practice of protecting various networks and the information they carry from unauthorized access, misuse, and attacks. In simple terms, it's like building digital walls, gates, and alarm systems around the data and systems that computers use to communicate. Whether it's personal data, school records, business documents, or online messages, network security ensures that information remains safe from those who shouldn't access or alter it. As more devices become connected through the internet, having strong network security is more important than ever to protect privacy and prevent cybercrime.

Authentication

Authentication is the process that confirms someone's identity before they are allowed to access a network or system. It's like the digital version of showing your ID card before entering a secure area. The most common form of authentication is a username and password, but more secure systems use multi-factor authentication. Without authentication, anyone could access sensitive data, making it one of the first lines of defense in any secure connection, as shown in Figure 4.3.

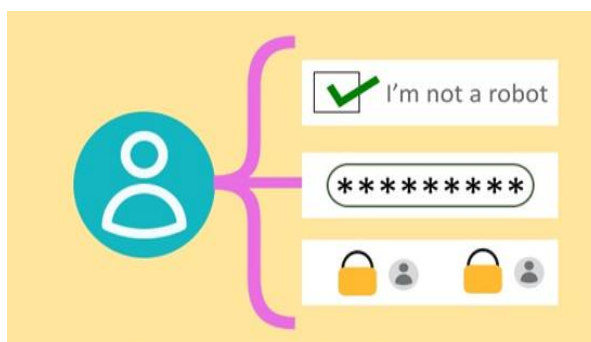


Fig 4.3. Network Security: Authentication

Firewall

The firewall acts like a virtual obstacle between a local network or your device and the internet. Just like a physical wall keeps out unwanted visitors, a firewall watches all the data entering and leaving a network and decides what should be allowed through and what should be blocked. Firewalls can be either hardware (physical devices often used in businesses) or software (programs installed on your computer or router). They use a set of rules to block suspicious or harmful traffic, such as data from untrusted sources, preventing many types of cyberattacks from reaching your system.

Antivirus and Anti-malware

These software programs are designed with the intent to detect, block, and remove software that appears malicious and can harm your computer or steal your information. Malware is a broad term that includes viruses, worms, ransomware, spyware, and other programs meant to damage or control a system. Antivirus software regularly scans your device for known threats and also watches for suspicious behavior that might indicate a new or unknown threat. Keeping this software updated is crucial, as new types of malware are created all the time, and updates include new definitions to keep your protection current.

Encryption

Encryption is a technique used to save data by transforming it into a coded format that can only be read by someone who has the correct decryption key. It is like sending a message in a language that is not understood by everyone—only your friend understands it. Even if a hacker intercepts the message, they won't be able to understand it without the key. Encryption is used in many everyday situations—when you shop online, your payment information is encrypted; when you use messaging apps, your conversations are encrypted to ensure privacy. It's one of the strongest tools for protecting data as it moves through the internet, as shown in Figure 4.4.

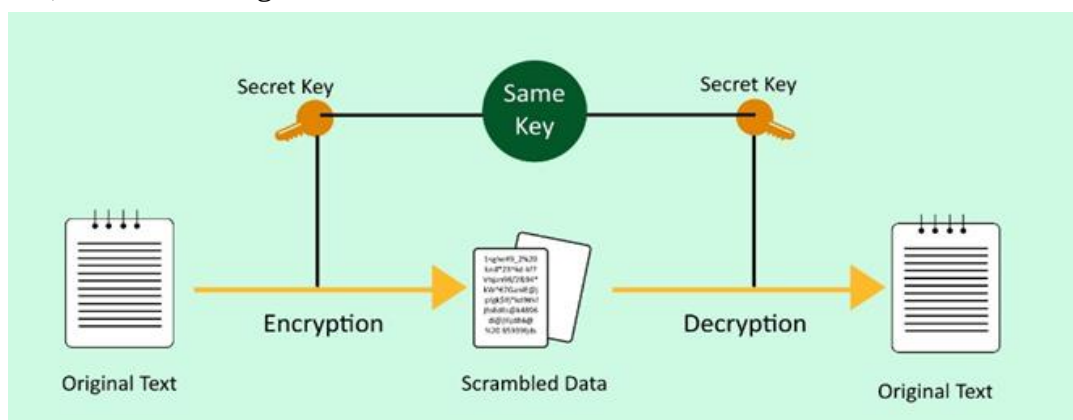


Fig 4.4. Process of Encryption and Decryption

Access Control

Access control is the practice of limiting who can see or use different parts of a network. In a school, for example, students might only have access to their homework and assignments, while teachers can access grading systems and staff communication tools. This ensures that people only see the data or tools they need, minimizing the risk of sensitive information being used in the wrong manner. Access control can be applied through user roles, passwords, biometric scans, or physical controls like key cards.

Virtual Private Network (VPN)

This type of connection, known as a Virtual Private Network (VPN), allows users to use the internet safely, even on public networks like those in airports, offices, or coffee shops. It works by forming a private “tunnel” between a remote server and your device, hiding all the

information that passes through it. This makes it much harder for internet service providers, various types of hackers, or even government agencies to see what you're doing online or steal your data. VPNs are used by individuals who want privacy, and also by companies to let employees safely access their work networks from anywhere, as shown in Figure 4.5.



Fig 4.5. Virtual Private Network

Network Cables

1. Unshielded Twisted Pair (UTP) Cable

UTP is a type of twisted pair cable. It stands for Unshielded Twisted Pair. Data and voice are sent through UTP because of its suitable frequency range.

Features:

- Inexpensive: These cables are relatively inexpensive.
- Easy Installation: This is a more prevalent choice as they are easy to install and terminate.
- Vulnerable to Interference: They are susceptible to interference from nearby sources, which causes signal attenuation.
- Limited Distance: They can transmit data reliably over short distances of up to 100 meters.

2. Shielded Twisted Pair (STP) Cable

STP cable is similar to UTP but includes a layer of shielding (metal foil or braided mesh) around the twisted wire pairs. This shielding helps protect the signals from electromagnetic interference (EMI) and crosstalk, making it more reliable in areas with lots of electrical noise. STP is used in industrial environments or places where interference is a concern.

Features:

- Shielded with metal foil or braided mesh
- Reduces electromagnetic interference (EMI)
- Less crosstalk between wire pairs
- Better signal quality over longer distances than UTP
- More expensive and less flexible than UTP
- Used in industrial or high-interference environments

Assignment 4.1.

- Describe Network Security
- Describe Virtual Private Network

4.6 PATCH PANEL

A patch panel in a Local Area Network (LAN) is a mounted hardware assembly that contains ports used to connect and manage incoming and outgoing LAN cables. A patch panel provides a way to keep a large number of cables organized and enables flexible connectivity within the network wiring closet, as shown in Figure 4.6.



Fig 4.6. Patch panel

The panels can be mounted in either a 19-inch or 23-inch rack. The hardware assembly of a patch panel has open ports on one side and termination points on the other. Cables within buildings or premises can be terminated at the patch panel, labeled, and then patched into network or audiovisual (AV) hardware. These are also referred to as patch bays, patch fields, or jackfields.

Working of Panel Work

In a network, it acts as a type of static switching board. It allows cables to be interconnected with external lines, including networked computers within the LAN, the Internet, and other wide-area networks. The patch panel uses RJ-45 connectors for twisted-pair Ethernet connections.

In facilities requiring satellite TV or cable with centralized controls, coaxial patch panels are used to distribute television signals across the premises. Patch panels serve as connection points using patch cords or jumper cables, creating links between the patch panel and network devices such as switches, routers, firewalls, and more. They allow easy reorganization of circuits and devices by simply moving patch cables between different ports. Companies and other organizations typically install patch panels in cable cabinet spaces. A cable cabinet space is a small room housing network and other electrical connections.

Patch panels are designed to support specific copper cable standards, such as CAT5e, CAT6, CAT6A, and CAT7. These standards use different measurement parameters for copper cables, so it's important to identify the types of cables running through a cable cabinet or data center and match them to the correct patch panel specifications. UTP cables are commonly used in typical office environments.

4.7 R J 45

RJ45 stands for Registered Jack 45 and is the most frequently used connector in wired networks. These sockets are primarily used to connect devices to Local Area Networks (LANs). RJ45 was originally developed for telephone systems but is now widely used in Ethernet networking. The “45” in RJ45 refers to a standard specification number.

The width of an RJ45 connector is typically greater than that of a telephone cable or other registered jacks. Compared to other connectors, RJ45 offers higher bandwidth—typically supporting speeds up to 10 Gbit/s. It is commonly used to connect personal computers with servers, routers, and other networking devices due to its speed and enhanced security. These connectors are primarily used in star topologies.

RJ45 uses an 8P8C configuration, meaning it has eight positions and eight conductors. The sockets are primarily used with Shielded Twisted Pair (STP) or Unshielded Twisted Pair (UTP) cables. If you look closely at the end of an Ethernet cable connected to an RJ45 connector, you will see eight wires: four solid-colored and four striped.

RJ45 cable color codes include:

White/Green, Green, White/Orange, Blue, White/Blue, Orange, White/Brown, and Brown.

There are two main wiring standards for RJ45: T568A and T568B. They differ in the arrangement of the green and orange wire pairs. In T568A, the green pair precedes the orange pair; in T568B, the orange pair comes before the green. Another difference is that T568A is more compatible with older cabling systems, while T568B offers slightly better resistance to signal noise. As shown in Figure 4.7.

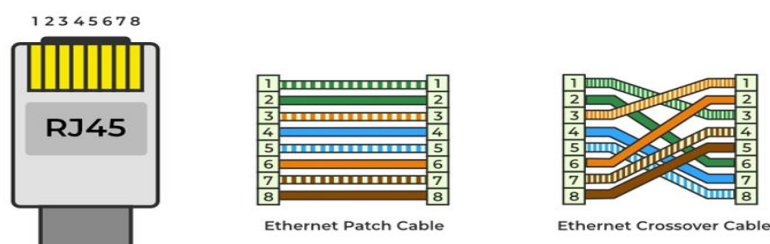


Fig 4.7. RJ 45

Practical Knowledge**1. Networking Components**

1a. LAN Adapters: When a device (such as a computer, laptop, or server) wants to connect to a Local Area Network (LAN), a LAN adapter is used, which is a hardware component, as illustrated in Figure 4.8.



Fig 4.8. LAN Adapters

1b. Hubs: This is a basic device that connects multiple Ethernet devices together in a LAN. It operates at Layer 1, i.e., the Physical Layer of the OSI model, as illustrated in Figure 4.9.



Fig 4.9. Hubs

1c. Switches: This is a more advanced networking device that also connects multiple Ethernet devices together in a LAN. It operates at Layer 2, i.e., the Data Link Layer of the OSI model, as illustrated in Figure 4.10.



Fig 4.10. Switches

1d. Routers: It is a networking device used to connect two or more different networks together. These networks can be Local Area Networks (LANs), Wide Area Networks (WANs) such as the Internet, or different subnets within a larger network. It operates at Layer 3, i.e., the Network Layer of the OSI model. Router illustrated in Figure 4.11.

*Fig 4.11. Router*

1e. NIC: A NIC serves the same purpose as a LAN adapter—it's the physical and logical interface that allows a device to connect to a network. It handles the physical signaling and data link layer protocols necessary for network communication. NIC is illustrated in Figure 4.12.

*Fig 4.12. NIC***Practical Activity 4.1: File sharing over LAN:**

Material Required: A computer system with Cisco Packet Tracer software installed, Two PCs, one Server, and one Switch (from Packet Tracer device list), Copper Straight-Through network cables for connections, Built-in configuration tools in Packet Tracer (IP Configuration, Command Prompt, Services/FTP tab)

Step-1: Build the LAN:

- Add two "PC" devices (PC0 and PC1) and one "Server" device onto the workspace.
- Add a "Switch".
- Connect PC0 and PC1 to the Switch using "Copper Straight-Through" cables (e.g., PC0 FastEthernet0/0 to Switch FastEthernet0/1, PC1 FastEthernet0/0 to Switch FastEthernet0/2).
- Connect the Server to the Switch using a "Copper Straight-Through" cable (e.g., Server FastEthernet0 to Switch FastEthernet0/3).

Step-2: Configure IP Addresses:

- Server:
 - Click on the Server.
 - Go to Desktop -> IP Configuration.
 - Select Static.
 - IP Address: 192.168.1.5
 - Subnet Mask: 255.255.255.0
 - Close the window.

PC4: (you can name the PC's any name according to your preference just by double clicking on the current name of PC)

- Click on PC4.
- Go to Desktop -> IP Configuration.
- Select Static.
- IP Address: 192.168.1.10
- Subnet Mask: 255.255.255.0
- Close the window.

PC5: (you can name the PC's any name according to your preference just by double clicking on the current name of PC)

- Click on PC5.
- Go to Desktop -> IP Configuration.
- Select Static.
- IP Address: 192.168.1.11
- Subnet Mask: 255.255.255.0
- Close the window.

Step-5: Enable File Sharing on the Server:

- Click on the Server.
- Go to the Services tab.
- On the left menu, click on FTP.
- Ensure the FTP service is "On". You can also create user accounts with specific permissions here if you want more controlled access. For simplicity, the default "cisco" user with password "cisco" can be used.

Step-6: Access Shared Files from a PC (PC4):

- Click on PC4.
- Go to Desktop -> Command Prompt.
- Use the ftp command followed by the server's IP address:
- ftp 192.168.1.5
- Press Enter. You will be prompted for a username. Enter cisco (or another username you configured).
- Press Enter. You will be prompted for a password. Enter cisco (or the password for the user).
- If successful, you will see an ftp> prompt. You can use basic FTP commands:
 - ls or dir: List files on the server.
 - get <filename>: Download a file from the server to the PC.
 - put <filename>: Upload a file from the PC to the server (if permissions allow).
 - bye: Exit the FTP session.

Step-7: Access Shared Files from another PC (PC1):

- Repeat step 4 on PC5 to access the files on the server. Below Figure 4.13 to Figure 4.17 shows the several stages of IP configuration.

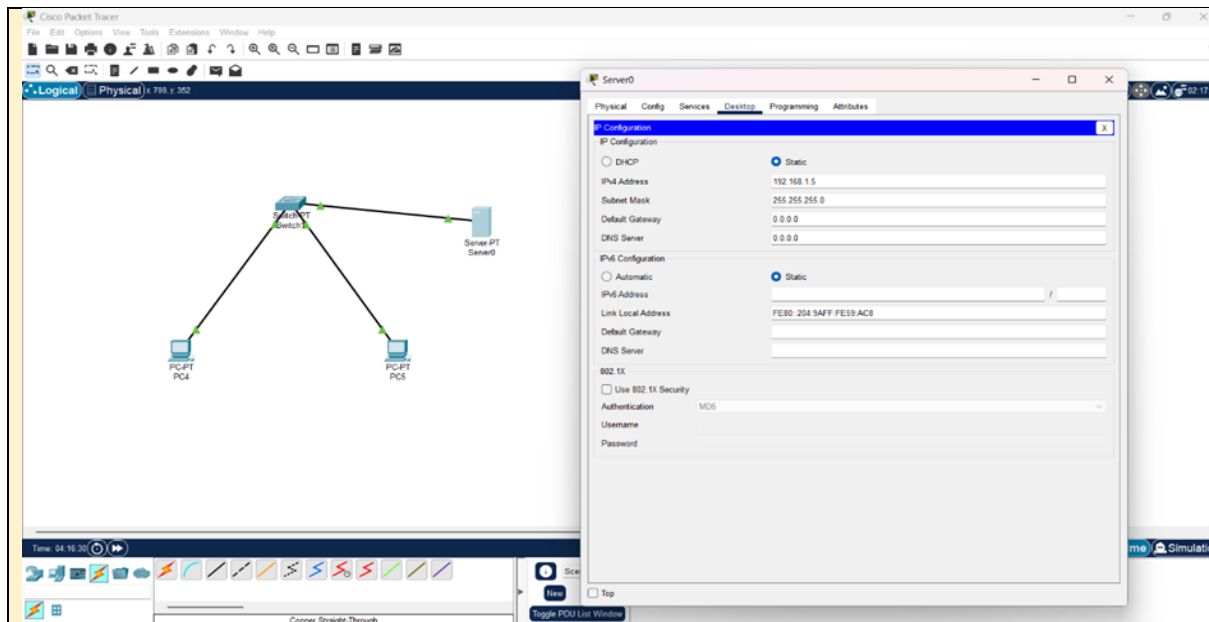


Fig 4.13. IP configuration of Server0

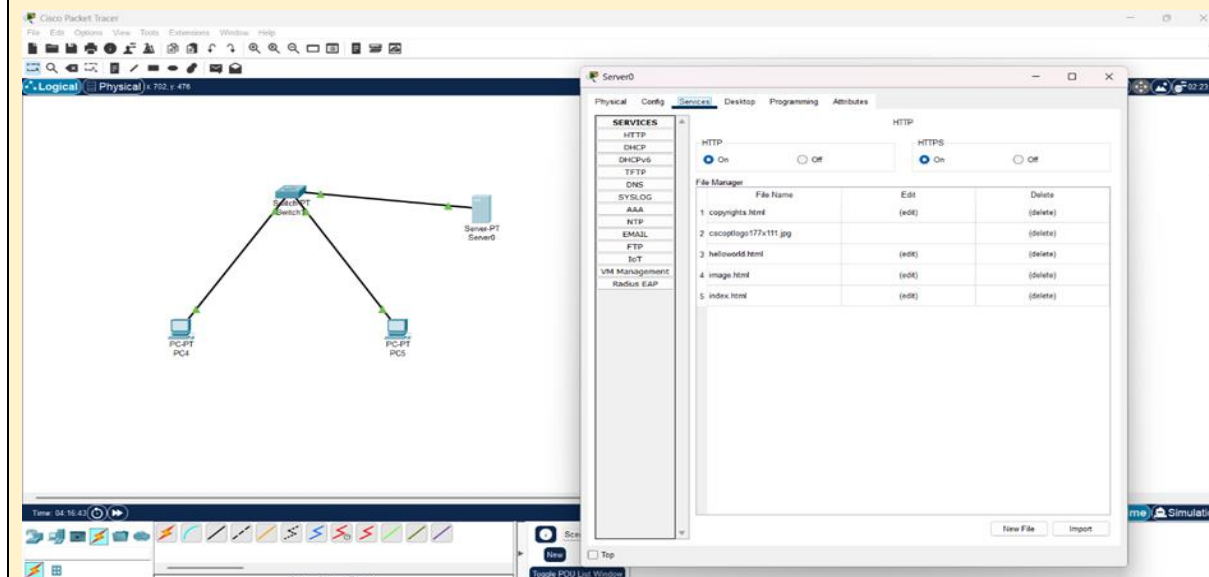


Fig 4.14. Using service tab to enable file sharing in Server0

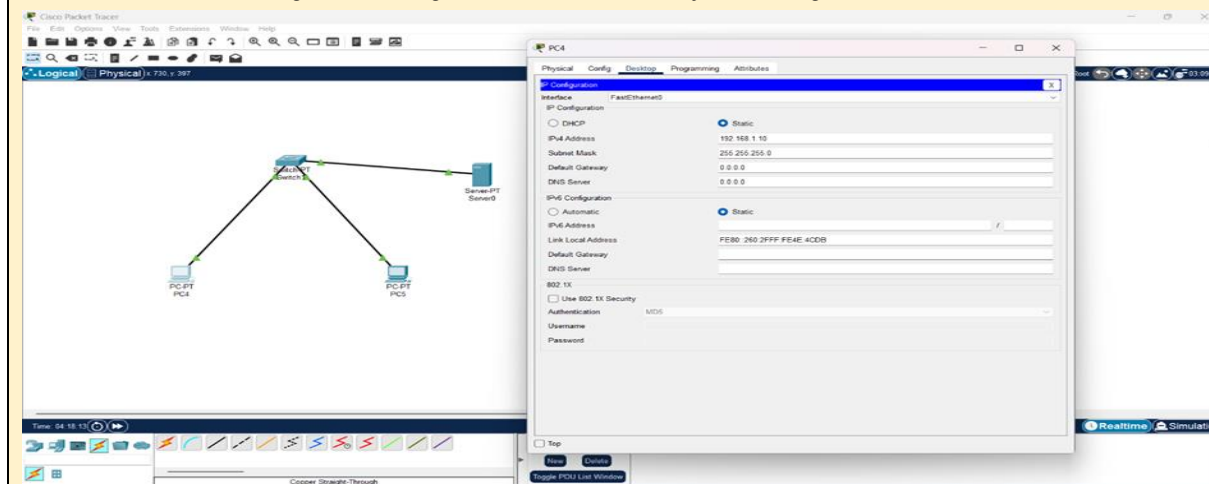


Fig 4.15. IP configuration of PC4

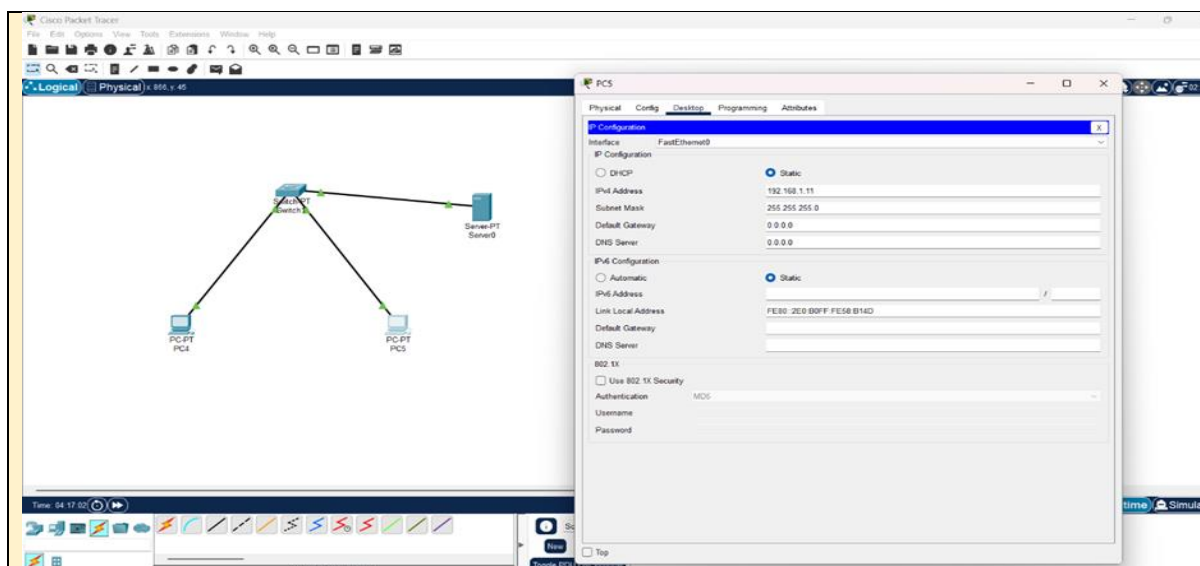


Fig 4.16. IP configuration of PC5

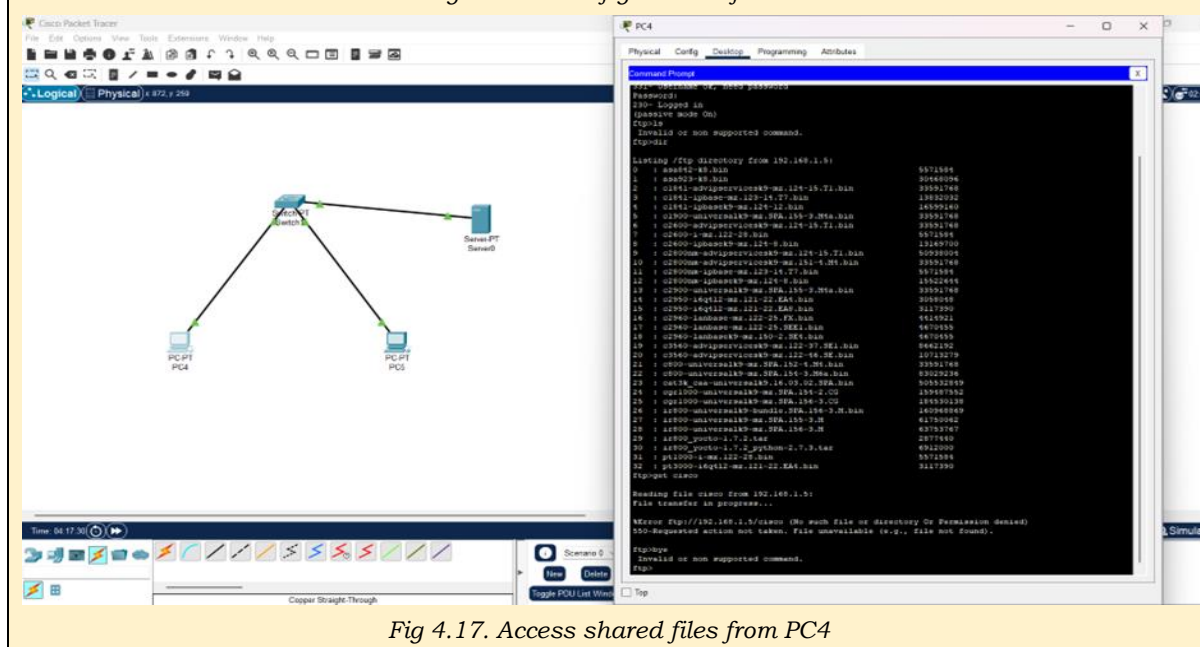


Fig 4.17. Access shared files from PC4

Summary

- Transmission modes define how data travels: simplex, half-duplex, or full-duplex.
- Simplex allows one-way communication.
- Half-duplex allows two-way communication, but only one direction at a time.
- Full-duplex allows simultaneous communication in both directions.
- Topology refers to the layout of network devices.
- Bus topology uses a single backbone.
- Star topology connects devices through a central hub or switch.
- Ring topology forms a closed loop.
- For device-to-device connections, mesh topology is used.
- Tree and hybrid topologies combine elements of other topologies.

Check your progress**A. Multiple Choice Questions (MCQs)**

1. What is the primary function of a "switch" in a network?
 - a) To broadcast data to all devices on the network.
 - b) To maintain a switch table to efficiently direct data packets to their specific destination.
 - c) To connect to the internet and act as a central point for wireless devices.
 - d) To divide data into smaller parts called packets.
2. Which type of switching establishes a dedicated path between the sender and receiver before any data is sent, similar to traditional phone calls?
 - a) Message Switching
 - b) Packet Switching
 - c) Circuit Switching
 - d) Datagram Packet Switching
3. Which type of switching is used by the Internet and most modern networks, where data is divided into smaller parts that travel independently?
 - a) Message Switching
 - b) Circuit Switching
 - c) Virtual-Circuit Packet Switching
 - d) Packet Switching
4. What is the most prevalent and reliable wired LAN technology that uses special cables to connect devices physically?
 - a) Wi-Fi
 - b) Bluetooth
 - c) Ethernet
 - d) Satellite Internet
5. Which of the following WAN technologies offers a dedicated, private connection between two or more locations, known for high reliability and consistent speed, often used by large businesses?
 - a) Wi-Fi
 - b) Ethernet
 - c) Leased Lines
 - d) Satellite Internet
6. Which network security measure acts like a virtual obstacle, watching all data entering and leaving a network and blocking suspicious traffic?
 - a) Authentication
 - b) Encryption
 - c) Firewall
 - d) Access Control
7. What is the purpose of encryption in network security?
 - a) To confirm someone's identity before allowing access.
 - b) To limit who can see or use different parts of a network.
 - c) To transform data into a coded format that can only be read with a decryption key.
 - d) To detect and remove malicious software.
8. Which type of network cable is inexpensive, easy to install, but vulnerable to interference and has a limited reliable distance of up to 100 meters?
 - a) Fiber Optic cable
 - b) Coaxial cable

- c) Unshielded Twisted Pair (UTP) cable
 - d) Shielded Twisted Pair (STP) cable
9. What is the primary function of a patch panel in a LAN?
- a) To provide wireless connectivity to devices.
 - b) To act as a central processing unit for network data.
 - c) To organize and manage incoming and outgoing LAN cables, enabling flexible connectivity.
 - d) To secure the network from unauthorized access.

What is the most frequently used connector in wired Ethernet networks, commonly used to connect personal computers with servers and routers?

- a) USB
- b) HDMI
- c) RJ11
- d) RJ45

B. Fill in the Blanks

1. In _____ mode, data flows in both directions but not simultaneously.
2. _____ mode allows simultaneous data transfer in both directions.
3. _____ topology has a central hub to which all devices connect.
4. _____ topology uses a single communication line.
5. Mesh topology provides _____ links between devices.
6. _____ topology forms a hierarchy.
7. Simplex mode allows _____-way communication only.
8. Hybrid topology allows _____ configurations.
9. In a _____ transmission mode, data can travel in both directions, but only one direction at a time.
10. A _____ topology connects all devices through a central hub or switch.

C. True or False

1. In full-duplex mode, data can be transmitted in both the directions.
2. Half-duplex is more efficient than full-duplex.
3. Bus topology and Star topology forms the Tree topology.
4. In mesh topology, every node is linked to every other node.
5. Star topology depends on a central hub or switch.
6. Bus topology is very robust.
7. Ring topology allows bidirectional communication.
8. Simplex communication is two-way.
9. Hybrid topology is flexible and scalable.
10. In a ring topology, data moves in a circular fashion.

D. Short Answer Questions

1. What is simplex transmission mode?
2. Define half-duplex with an example.
3. What are the advantages of full-duplex mode?
4. Explain mesh topology.
5. What happens when the central hub in a star topology fails?

6. What is a hybrid topology?
7. List one advantage and one disadvantage of the bus topology.
8. Why is ring topology considered less secure?
9. Differentiate between logical and physical topologies.
10. Define tree topology with an example.

Answer Key**A. Multiple Choice Questions**

1. b), 2. c), 3. d), 4. c), 5. c), 6. c), 7. c), 8. c), 9. c), 10. d)

B. Fill in the blanks:

1. Half-duplex, 2. Full-duplex, 3. Star, 4. Bus, 5. Dedicated, 6. Tree, 7. One, 8. Multiple, 9. half-duplex, 10. star

C. True or False

1. True, 2. False, 3. True, 4. True, 5. True, 6. False, 7. True, 8. False, 9. True, 10. True

Unit-3.

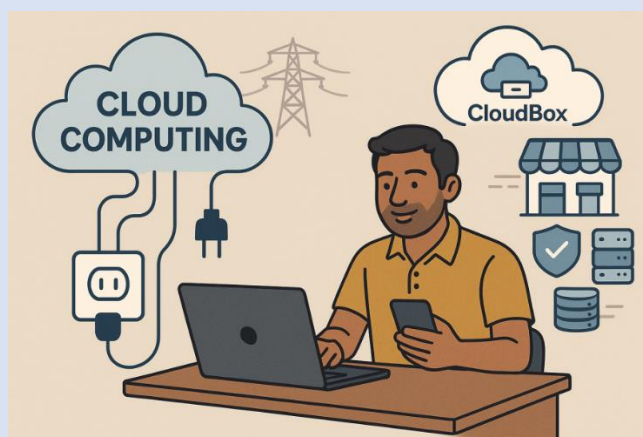
Implementing Virtualization and Basics of Cloud Computing

Chapter-1**Understanding the Cloud Computing Fundamentals**

Cloud computing is like using electricity—you don't need to own a power plant to use it. Instead of buying big computers or servers, people and businesses can use computing power, storage, and software over the internet, just like they use electricity from a plug.

For example, imagine Ravi, who runs a small online store. Instead of buying expensive servers to host his website and store data, Ravi uses a cloud service like "CloudBox." CloudBox provides him with storage space, runs his website, and keeps everything backed up and safe. Ravi, as shown in Figure, only pays for what he uses, and he can access his data anytime, anywhere—from his laptop or even his mobile phone.

That is the beauty of cloud computing—it's flexible, cost-effective, and very easy to scale up or down as needed.



Cloud computing marks a significant evolution in technology, allowing users to access computing assets such as servers, storage, databases, and software applications via the internet, rather than depending exclusively on on-premises hardware. This model provides organizations with enhanced flexibility, scalability, and cost-effectiveness, enabling rapid adjustment to evolving needs.

Fundamentally, cloud computing relies on centralized data centers managed by third-party providers, delivering resources as needed and typically charging based on consumption. To fully understand the benefits of cloud adoption, it is important to recognize the three primary service models:

Infrastructure as a Service (IaaS): Offers virtualized infrastructure.

Platform as a Service (PaaS): Supplies development tools and environments.

Software as a Service (SaaS): Makes applications accessible over the web.

As its usage continues to expand across sectors, cloud computing has become a cornerstone of modern IT strategies, driving innovation, operational agility, and digital advancement.

In this session, you will learn about cloud computing fundamentals, its types, and security-related techniques.

1.1 Introduction to Cloud Computing

In the rapidly advancing world of information technology, cloud computing stands out as one of the most transformative innovations of the 21st century. It has fundamentally altered how data is accessed, stored, and managed, replacing traditional hardware-centric infrastructures with flexible, scalable, and on-demand services delivered over the internet. This shift eliminates the need for heavy upfront investments in physical equipment and introduces a

utility-based consumption model—similar to electricity or water—where users pay only for the resources they utilize.

At its essence, cloud computing refers to the provision of computing services, including servers, storage, databases, networking, software, analytics, and artificial intelligence, over the internet—commonly known as "the cloud." These services are made available by major cloud service providers (CSPs) like Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), among others. These providers maintain extensive global data center networks to ensure that services are consistently available, high-performing, and reliable. Cloud computing enables businesses to deploy applications across the globe in minutes, maintain high availability, and scale resources on demand according to workload fluctuations.

Cloud services are typically delivered through three main models:

- **Infrastructure as a Service (IaaS):** Supplies virtualized infrastructure components such as virtual machines, storage, and networking, allowing users to avoid the complexities of managing physical hardware.
- **Platform as a Service (PaaS):** Provides a comprehensive development and deployment environment with integrated tools and frameworks, enabling developers to build, test, and launch applications without managing underlying infrastructure.
- **Software as a Service (SaaS):** Delivers software solutions over the internet via subscriptions, accessible through web browsers or APIs, relieving users from the burden of installation and ongoing maintenance.
- Beyond these service models, cloud computing is implemented using various deployment approaches:
- **Public Cloud:** Offers resources over the internet that are shared among multiple users.
- **Private Cloud:** Delivers cloud services exclusively for a single organization, offering enhanced control and security.
- **Hybrid Cloud:** Combines public and private cloud environments, allowing seamless data and application movement between them.
- **Multi-Cloud:** Involves the use of cloud services from multiple providers, helping organizations avoid vendor lock-in and optimize performance.

The advantages of cloud computing are far-reaching. It reduces capital expenses, accelerates time to market through rapid provisioning, supports business continuity with automated backups and disaster recovery, and enhances productivity by allowing remote access from anywhere with an internet connection. Additionally, cloud infrastructure is critical in powering emerging technologies such as artificial intelligence (AI), machine learning (ML), the Internet of Things (IoT), and big data analytics—offering the scalability and computing power essential for their deployment and operation.

As digital transformation accelerates across all sectors, understanding cloud computing is not just an advantage but a necessity for IT professionals, students, and decision-makers. This chapter aims to provide a solid foundation in cloud computing concepts, explore its architectural components, examine real-world applications, and discuss the challenges and future directions of cloud-based systems. Whether you're a novice seeking to understand the basics or a practitioner looking to deepen your knowledge, this chapter sets the stage for exploring the vast and evolving domain of cloud computing.

Assignment.1.1:

1. What is the primary role of virtualization in modern IT infrastructure?
2. Name the three main cloud service models and briefly describe each.
3. Differentiate between Type 1 and Type 2 hypervisors with an example use case.

1.2 Cloud Service Models (IaaS, PaaS, SaaS)

Cloud computing has revolutionized the way organizations consume and deliver IT services. Instead of maintaining expensive infrastructure, businesses can now leverage scalable, on-

demand computing resources provided by cloud service providers. These services are typically delivered through three fundamental models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

1. Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) provides virtualized computing resources over the internet. This includes servers, storage, networking, and virtualization. It is the most basic level of cloud service, giving users high-level control over the infrastructure without the need to physically manage it.

- Virtual machines with customizable configurations
- Scalable storage solutions (block, file, object storage)
- Networking resources (IP addresses, VPNs, load balancers)
- Users manage operating systems and applications
- Pay-as-you-go pricing model

Advantages:

- Cost-effective (no capital investment in hardware)
- High scalability and flexibility
- Full control over computing infrastructure
- Quick provisioning of resources

Use Cases:

- Hosting websites and applications
- Storage and backup
- High-performance computing
- Disaster recovery solutions

Examples of IaaS:

- Amazon Web Services (AWS) EC2
- Microsoft Azure Virtual Machines
- Google Compute Engine (GCE)
- IBM Cloud Infrastructure

2. Platform as a Service (PaaS)

Platform as a Service (PaaS) provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the underlying infrastructure. It offers a development environment with tools, libraries, and frameworks.

- Application hosting and development framework
- Integrated development tools and services
- Support for multiple programming languages
- Built-in scalability and high availability
- Automatic updates and security patches

Advantages:

- Accelerated application development
- Simplified deployment and management
- Reduces time to market
- Allows developers to focus on coding rather than infrastructure

Use Cases:

- Developing mobile or web applications
- API development and management

- Microservices architecture deployment
- Continuous integration and continuous deployment (CI/CD)

Examples of PaaS Providers:

- Google App Engine
- Microsoft Azure App Service
- AWS Elastic Beanstalk
- Heroku
- Red Hat OpenShift

3. Software as a Service (SaaS)

Software as a Service (SaaS) refers to the delivery of software applications over the internet, typically through a subscription model. The service provider is responsible for hosting, managing, and maintaining the applications, which users can access directly via web browsers eliminating the need for local installation or complex configuration.

- Instant access to applications as needed
- Fully hosted and maintained by the provider
- Usable from any location with internet access
- Built on a multi-tenant model, where a single application instance serves multiple users

Advantages:

- No installation or maintenance required
- Automatic updates and patches
- Scalable with user demand
- Accessible across devices and platforms

Use Cases:

- Email and collaboration tools
- Customer Relationship Management (CRM)
- Enterprise Resource Planning (ERP)
- Accounting and HR software
- Office productivity tools

Examples of SaaS Providers:

- Google Workspace (Docs, Gmail, Sheets)
- Microsoft 365 (Word, Excel, Outlook)
- Salesforce
- Dropbox
- Zoom

Table 1.1. shows the comparison between IaaS vs PaaS vs SaaS

Table 1.1: Comparison between IaaS vs PaaS vs SaaS

Feature / Model	IaaS	PaaS	SaaS
User Control	Infrastructure & OS	Applications & Data	Application Use Only
Management	User manages apps, OS	User manages apps only	Provider manages everything
Target Audience	System administrators	Developers	End-users
Customization	High	Moderate	Low
Cost Model	Pay-as-you-go	Subscription or pay-as-you-go	Subscription
Example Providers	AWS EC2, Azure VMs	Google App Engine, Heroku	Salesforce, Office 365

Each model offers unique benefits and trade-offs depending on the specific needs of an organization. A strong grasp of these service models also helps in addressing security, compliance, and operational strategies in cloud environments.

Practical Activity 1.1. Understanding the fundamentals of virtualization and its advantages in cloud computing through hands-on implementation using an open-source operating system.

Materials Required:

1. A physical machine with at least:
 - 8 GB RAM
 - 50+ GB disk space
 - Intel VT-x / AMD-V enabled CPU
2. Ubuntu Linux OS (Desktop or Server Edition – latest LTS preferred)
3. Virtualization Software:
 - VirtualBox (open-source, easy for beginners)
 - OR KVM (Kernel-based Virtual Machine – native to Linux)
4. ISO images of guest operating systems (Ubuntu, CentOS, etc.)
5. Internet connection (for downloading packages/updates)
6. Terminal/CLI knowledge (basic Linux commands)

Practical Steps:

Step 1: Install Ubuntu Linux OS

- Download and install the latest Ubuntu (LTS) version.
- Update the system using:
 - ‘sudo apt update && sudo apt upgrade’

Step 2: Install Virtualization Software

Option A: Using VirtualBox

- ‘sudo apt install virtualbox’

Option B: Using KVM and Tools

- ‘sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils virt-manager’

Step 3: Verify Virtualization Support

- ‘egrep -c '(vmx|svm)' /proc/cpuinfo’

Output ≥1 means virtualization is supported and enabled in BIOS.

Step 4: Create a Virtual Machine (VM)

Launch VirtualBox or Virtual Machine Manager

Create a new VM:

Assign name, memory (2+ GB), disk (20+ GB), and ISO image for guest OS.

Step 5: Install Guest OS

Boot the VM using ISO and install the OS as you would on a physical machine.

Set hostname, user credentials, and update the guest OS post-installation.

Step 6: Explore and Demonstrate Benefits

Snapshot Management: Take VM snapshots for quick rollback.

Resource Efficiency: Monitor system usage using `top` or `htop`.

Isolation: Run separate apps on different VMs without interference.

Scalability: Clone VM for easy scaling.

Multi-tenancy Simulation: Create multiple VMs and simulate different user roles.

Step 7: (Optional) Network and Storage Experiments

Configure shared folders or networks between host and VMs.

Test data transfer, latency, and isolation.

1.3 Cloud Deployment Models (Public, Private, Hybrid)

Cloud computing deployment strategies vary based on an organization's specific requirements, including security policies, regulatory obligations, and financial considerations. Deployment models define how the cloud environment is structured, who owns it, and how it is accessed. The three main models in this category are the Public Cloud, Private Cloud, and Hybrid Cloud.

1. Public Cloud

The public cloud model involves infrastructure that is owned and maintained by third-party cloud service providers and is made available to users over the internet.

Key Characteristics:

- Operated by external vendors
- Infrastructure is shared among multiple tenants
- Accessible via the public internet
- Emphasizes scalability and cost-effectiveness

Benefits:

Public cloud services are known for being budget-friendly since they eliminate the need for capital investment in physical infrastructure and ongoing maintenance. They allow for elastic scalability, enabling organizations to scale resources up or down depending on demand. High availability is built-in, supported by redundant systems and disaster recovery protocols. Additionally, users are relieved from hardware upkeep and software updates, as these responsibilities lie with the provider.

Limitations:

Users have limited control over the infrastructure, which may pose challenges for organizations with specific customization or regulatory needs. Since the environment is shared, there may be concerns around data privacy and compliance, especially for sectors with stringent legal requirements.

Ideal Use Cases:

- Hosting websites or customer-facing applications
- Cloud-based backup and storage
- Application development and testing environments
- Short-duration or dynamic workload projects

Leading Providers:

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform (GCP)
- IBM Cloud

2. Private Cloud

Private cloud refers to a computing environment that is used exclusively by one organization. It can either be located on-premises or hosted by a trusted third party.

Key Characteristics:

- Dedicated environment for a single entity
- Can be deployed internally or externally
- Enhanced security and regulatory compliance
- Greater flexibility for customization

Benefits:

Private clouds offer strong data security due to isolated infrastructure and granular access controls. They are ideal for organizations in regulated industries like healthcare or finance due to their ability to meet strict compliance requirements. With full control over the environment, organizations can fine-tune configurations to match specific needs, and dedicated resources ensure reliable performance.

Challenges:

Deploying a private cloud requires a higher financial investment for setup and upkeep. Additionally, managing such environments, particularly in-house, demands specialized IT expertise. Private clouds may also lack the scalability offered by public cloud solutions.

Ideal Use Cases:

- Government and military applications
- Financial and healthcare institutions
- Enterprises managing confidential or sensitive data
- Organizations with regulatory obligations

Notable Platforms:

- VMware vSphere
- OpenStack
- Microsoft Azure Stack
- IBM Cloud Private

3. Hybrid Cloud

- Hybrid cloud is a strategic blend of public and private cloud environments that enables data and applications to be moved seamlessly between them.

Key Characteristics:

- Combines private and public cloud infrastructure
- Supports interconnectivity and orchestration
- Promotes workload mobility across environments
- Allows strategic segregation of workloads

Benefits:

The hybrid approach delivers flexibility, allowing businesses to run less sensitive operations on the public cloud while reserving the private cloud for mission-critical workloads. It strikes a balance between performance, control, and cost, enabling organizations to scale when needed without losing governance over sensitive data. Hybrid models also support strong business continuity strategies with robust backup and failover mechanisms across environments.

Drawbacks:

Managing hybrid cloud systems can be complex due to the need for seamless integration and consistent security across platforms. Organizations must ensure that tools, APIs, and security policies are uniformly implemented to avoid gaps in protection or functionality.

Ideal Use Cases:

- Businesses with fluctuating or seasonal demand
- Disaster recovery and high availability planning
- Workloads requiring data sovereignty
- Organizations undergoing gradual cloud migration

Prominent Solutions:

- Microsoft Azure Arc
- AWS Outposts
- Google Anthos
- IBM Hybrid Cloud

Comparison Table: Public vs. Private vs. Hybrid Cloud

Feature/Model	Public Cloud	Private Cloud	Hybrid Cloud
Ownership	Third-party provider	Single organization	Combination of both
Accessibility	Internet-based	Internal or dedicated network	Both internet and internal access
Cost	Pay-per-use, low upfront	High upfront and maintenance	Balanced cost depending on usage
Security	Standard security	High, customizable	High (requires proper integration)
Scalability	Very high	Limited to internal resources	High (with controlled extension)
Management	Managed by provider	Managed by the organization	Shared management

Security Considerations for Deployment Models

- **Public Cloud:** Strong identity and access management (IAM), encryption, and third-party audits are essential.
- **Private Cloud:** Emphasis on internal access control, physical security, and endpoint protection.
- **Hybrid Cloud:** Ensuring secure data transfer, unified security policies, and endpoint integration is key.

Assignment.1.2:

1. What are the three fundamental service models of cloud computing?
2. Name any two examples of Infrastructure as a Service (IaaS) providers.
3. What is the main difference between a Public Cloud and a Private Cloud in terms of ownership?

1.4 Cloud Security and Compliance

Ensuring cloud security and regulatory compliance is crucial for the successful adoption and operation of cloud computing solutions. As businesses migrate their data, applications, and workloads to cloud platforms, safeguarding the confidentiality, integrity, and availability of information becomes a top priority. Cloud security refers to the integrated set of policies, technologies, controls, and processes designed to protect cloud-based environments. It spans multiple layers, including network protection, data encryption, identity and access management (IAM), threat detection systems, and incident response protocols. These safeguards must address both the physical infrastructure controlled by the cloud provider and the virtual components managed by the customer.

A central principle in cloud security is the shared responsibility model, where security duties are divided between the cloud provider and the customer. Under this model, the provider ensures the safety of the core infrastructure such as servers, storage, and networking while the customer is responsible for securing their data, applications, and user access controls. For instance, in an Infrastructure as a Service (IaaS) model, the cloud vendor manages physical security, but the user must secure virtual machines and data. In contrast, with Software as a Service (SaaS), more of the security burden shifts to the provider, including aspects like application-layer protection.

Compliance, in contrast, deals with the observance of legal, regulatory, and organizational standards concerning data privacy, security, and usage. Organizations operating in different sectors or jurisdictions must adhere to various frameworks, such as the General Data Protection Regulation (GDPR) in the EU, the Health Insurance Portability and Accountability

Act (HIPAA) for the healthcare sector, the Payment Card Industry Data Security Standard (PCI-DSS) for financial transactions, and ISO/IEC 27001 for managing information security. Although many cloud providers offer services aligned with these compliance standards and support compliance assessments, the ultimate responsibility for meeting regulatory requirements rests with the client.

Major security concerns in cloud environments include data leaks, compromised credentials, exposed APIs, poor identity governance, and misconfigured cloud storage. To counter these threats, organizations deploy advanced security measures like encryption for data at rest and in transit, multi-factor authentication (MFA), strict IAM policies, secure development methodologies (such as DevSecOps), and ongoing monitoring with detailed logging.

Additionally, compliance is closely linked to data residency and sovereignty issues, particularly for global enterprises. Understanding where data is physically stored and the legal implications of data transfers across borders is essential. A comprehensive approach to cloud security and compliance should integrate technical controls, governance structures, risk management practices, and continuous audit mechanisms to ensure a strong security posture and compliance with applicable laws and standards. Table 1.2 shows the aspect and details of cloud security.

Table 1.2: Cloud security aspect and details

Aspect	Details
Security Domains	Network security, data protection, IAM, application security, threat detection, physical security
Shared Responsibility	Varies by model (IaaS: user manages OS/data; SaaS: provider manages more)
Key Security Measures	Encryption (at rest/in transit), MFA, role-based access control (RBAC), logging, security automation, regular audits
Compliance Frameworks	GDPR, HIPAA, PCI-DSS, ISO 27001, SOC 2, FISMA, NIST
Security Challenges	Data breaches, misconfigurations, weak IAM, insecure APIs, insider threats
Compliance Challenges	Data sovereignty, multi-jurisdictional compliance, evolving regulatory standards, audit readiness
Risk Mitigation Tools	Cloud Security Posture Management (CSPM), Security Information and Event Management (SIEM), Cloud Access Security Brokers (CASB), DLP solutions
Best Practices	Zero trust architecture, least privilege principle, automated compliance checks, cloud-native security tools

Practical Activity 1.2 Install Virtualbox with different flavors of OS

Materials Needed:

1. A computer with internet access (Windows/Linux/macOS)
2. Oracle VM VirtualBox installer
Download from "<https://www.virtualbox.org/>"
3. ISO files of different operating systems:
 - Ubuntu (Linux)
 - Windows 10 or 11
 - Fedora or CentOS
 - Kali Linux (Optional for cybersecurity)
4. Minimum 8 GB RAM (Recommended: 16 GB)
5. At least 100 GB free disk space

Practical Steps:**Step 1:** Download and Install VirtualBox

- Go to “<https://www.virtualbox.org/>”
- Click on "Downloads"
- Choose the installer for your host OS (Windows/Mac/Linux)
- Run the installer and follow the on-screen setup instructions
- Launch VirtualBox after installation

Step 2: Download ISO Images of OS Flavors

- Ubuntu: “<https://ubuntu.com/download/desktop>”
- Windows: “<https://www.microsoft.com/software-download/>”
- Fedora/CentOS: “<https://getfedora.org/>” or “<https://www.centos.org/download/>”
- Kali Linux (Optional) : “<https://www.kali.org/get-kali/>”

Step 3: Create a New Virtual Machine

- Open VirtualBox
- Click "New"
- Enter VM name (e.g., Ubuntu_VM), select Type (Linux), and Version (Ubuntu 64-bit)
- Allocate RAM (2 GB for Linux, 4 GB+ for Windows)
- Create a virtual hard disk (e.g., 20 GB or more)
- Finish the VM creation

Step 4: Mount the ISO and Start Installation

- Select your VM → Click "Settings"
- Go to "Storage" → Click Empty under Controller: IDE
- Click the CD icon → Choose “Choose a disk file”
- Select the downloaded ISO file
- Click "Start" to boot the VM and follow the OS installation steps

Step 5: Repeat for Other OS Flavors

- Follow the same procedure to create and install VMs for other operating systems
- You can also run multiple VMs (if hardware permits)

1.5 Cloud Service Providers Overview

The cloud computing landscape is characterized by a few dominant entities known as Cloud Service Providers (CSPs). These vendors offer a broad spectrum of cloud computing services including compute, storage, networking, databases, analytics, machine learning, and security tools. The three most prominent CSPs are Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP), often referred to as the "Big Three." In addition, other notable providers such as IBM Cloud, Oracle Cloud Infrastructure (OCI), and Alibaba Cloud serve specialized markets and global regions.

Amazon Web Services (AWS) is the market leader, known for its vast portfolio of services and global availability zones. It offers extensive infrastructure solutions under IaaS, PaaS, and SaaS categories. AWS also excels in machine learning, IoT, and DevOps integrations, making it suitable for a wide variety of business needs. Its services such as EC2, S3, Lambda, and RDS are industry standards.

Microsoft Azure offers strong hybrid cloud capabilities, seamlessly integrating with on-premises Microsoft environments through tools like Azure Arc and Azure Stack. Azure is highly favored by enterprises already using Microsoft tools such as Office 365 and Active Directory. It provides robust security, governance, and compliance options and is a leader in enterprise-grade cloud solutions.

Google Cloud Platform (GCP) is recognized for its data analytics, artificial intelligence, and open-source innovation. GCP's BigQuery, Kubernetes Engine (GKE), and Vertex AI are popular among data-centric organizations and startups looking for scalable machine learning platforms.

IBM Cloud is known for its enterprise focus, particularly in regulated industries such as finance and healthcare. It integrates well with legacy systems and emphasizes AI through Watson and hybrid cloud via IBM Cloud Satellite.

Oracle Cloud Infrastructure (OCI) caters to customers needing high-performance computing and optimized database environments, especially for Oracle workloads. OCI offers competitive pricing and strong security measures.

Alibaba Cloud, the largest cloud provider in China and Asia-Pacific, delivers a comprehensive suite of services and is increasingly expanding its global footprint. It is particularly suited for businesses targeting the Asian market.

Provider	Key Offerings	Strengths	Ideal Use Cases
Amazon AWS	EC2, S3, RDS, Lambda, SageMaker, VPC, CloudTrail	Broadest service portfolio, global infrastructure, maturity	Enterprises, startups, AI/ML, DevOps, e-commerce
Microsoft Azure	Virtual Machines, Azure Functions, Cosmos DB, Azure AD, Azure DevOps	Strong hybrid cloud, enterprise integration, compliance-ready	Enterprises using Microsoft stack, hybrid cloud, government
Google Cloud	BigQuery, GKE, Vertex AI, Firebase, Cloud Run	Advanced analytics, ML, open-source integration	Data-centric businesses, analytics, AI/ML, startups
IBM Cloud	Watson AI, IBM Cloud Satellite, Kubernetes, Cloud Foundry	Enterprise focus, hybrid cloud, industry-specific compliance	Finance, healthcare, hybrid cloud, AI-driven workloads
Oracle OCI	Autonomous DB, Exadata Cloud, Oracle Apps, High-Performance Computing	Optimized for Oracle apps, security, cost-effectiveness	Enterprises using Oracle databases, finance, high-performance apps
Alibaba Cloud	ECS, OSS, ApsaraDB, Cloud Firewall, Machine Learning Platform for AI	Market leader in Asia, strong presence in e-commerce and AI	Asian markets, retail, global expansion into China

Assignment.1.3:

1. What is the shared responsibility model in cloud security, and how does it vary across different cloud service models?
2. Name any two compliance frameworks that organizations must consider when operating in cloud environments.
3. Mention two key security challenges in the cloud and briefly explain how organizations can mitigate them.

SUMMARY

- Cloud computing provides computing services like storage and servers over the internet on a pay-per-use basis. It eliminates the need for owning expensive infrastructure by offering flexible and scalable virtual resources.

- Like electricity, users can access cloud services anytime, anywhere without managing the backend infrastructure.
- There are three key service models: IaaS (infrastructure), PaaS (platform), and SaaS (software).
- IaaS provides virtual machines and storage, giving users control over OS and apps without physical hardware.
- PaaS offers a development environment with tools to build and deploy applications without managing servers.
- SaaS delivers fully managed applications over the internet, accessible via browsers with no installation needed.
- Cloud deployment models include Public (shared), Private (exclusive), and Hybrid (combined) clouds.
- Benefits of cloud computing include cost savings, high availability, scalability, collaboration, and innovation support.
- Understanding cloud fundamentals, service/deployment models, and security is essential for digital transformation.
- Cloud security ensures the confidentiality, integrity, and availability of data in cloud environments.
- It involves multiple layers like network security, encryption, IAM, threat detection, and incident response.
- The shared responsibility model divides security roles between cloud providers and customers.
- In IaaS, users secure VMs and data; in SaaS, providers handle more of the security stack.
- Compliance involves meeting legal, regulatory, and industry-specific standards like GDPR, HIPAA, and ISO 27001.
- Cloud providers offer compliance-ready infrastructure but customers are accountable for their data usage.
- Key threats include data breaches, insecure APIs, account hijacking, and misconfigurations.
- Mitigation tools include MFA, RBAC, DevSecOps, encryption, and continuous monitoring.
- Compliance also addresses data sovereignty, requiring awareness of data location and jurisdictional laws.
- A strong cloud security and compliance strategy blends technical safeguards with policy enforcement and regular audits.

Check Your Progress

A. Multiple choice Questions

1. What is the main benefit of cloud computing?
 - a) Increased capital investment
 - b) Long deployment times
 - c) Pay-as-you-go resource usage
 - d) Manual hardware configuration
2. Which of the following is NOT a cloud deployment model?
 - a) Private Cloud
 - b) Public Cloud
 - c) Hybrid Cloud

- d) Local Cloud
3. Ravi, who runs an online store, benefits from cloud computing primarily due to:
 - a) Hardware installation
 - b) Physical storage setup
 - c) Flexibility and scalability
 - d) Electricity consumption
 4. In which cloud model do users manage the operating system?
 - a) SaaS
 - b) PaaS
 - c) IaaS
 - d) DaaS
 5. Which service model provides a development platform with tools?
 - a) SaaS
 - b) IaaS
 - c) PaaS
 - d) NaaS
 6. Software like Google Workspace is an example of:
 - a) IaaS
 - b) SaaS
 - c) PaaS
 - d) DaaS
 7. What is a major advantage of the Public Cloud?
 - a) Full control over security
 - b) Limited scalability
 - c) High capital expense
 - d) Cost efficiency
 8. Which deployment model combines the benefits of both public and private clouds?
 - a) Community Cloud
 - b) Virtual Cloud
 - c) Hybrid Cloud
 - d) Shared Cloud
 9. What tool is used for virtualization in open-source systems?
 - a) Microsoft Office
 - b) VMware
 - c) VirtualBox
 - d) Chrome
 10. In SaaS, who is responsible for managing the application infrastructure?
 - a) End-user
 - b) Cloud provider
 - c) Developer
 - d) System administrator

B. Fill in the Blanks:

1. Cloud computing allows users to access computing resources over the _____.
2. _____ is a model where software is delivered over the internet as a service.

3. Virtual machines in IaaS are hosted on _____ servers.
4. _____ Cloud offers both security and scalability by combining two models.
5. In PaaS, the cloud provider manages the infrastructure and _____.
6. Public cloud is ideal for organizations looking for _____ and cost-effectiveness.
7. Cloud computing follows a _____ model, similar to utilities like electricity.
8. Google App Engine is an example of a _____ cloud service model.
9. AWS EC2 is a well-known example of a _____ service.
10. The physical machine used for virtualization must support _____ technology.

C. True or False

1. Cloud computing eliminates the need to invest in physical servers
2. In a Public Cloud, resources are used exclusively by a single organization.
3. SaaS applications are installed locally by the user.
4. PaaS allows developers to focus on building applications without managing infrastructure.
5. Ravi paid upfront for his CloudBox server and hardware.
6. VirtualBox is an open-source virtualization software.
7. Private clouds are more secure but less scalable than public clouds.
8. Hybrid clouds can increase complexity in management and security.
9. A company must own data centers to use cloud services.
10. IaaS gives users full control over applications and the operating system.

D. Short Answer Questions

1. What is cloud computing and how is it similar to utility services like electricity or water?
2. Name the three main cloud service models.
3. Which cloud model allows users to develop and run applications without managing the infrastructure?
4. What is the key difference between a public cloud and a private cloud?
5. Mention any two advantages of using cloud computing.
6. What is the shared responsibility model in cloud security?
7. Name any two common compliance standards related to cloud computing.
8. List any two major cloud service providers mentioned in the chapter.
9. Which cloud provider is known for its strong integration with Microsoft tools like Office 365?
10. Why is encryption important in cloud security?

Answer Key**A. Multiple Choice Questions (MCQ)**

1. (c) 2. (d) 3. (c) 4. (c) 5. (c) 6. (b) 7. (d) 8. (c) 9. (c) 10. (b)

B. Fill in the blanks

1. Internet 2. SaaS 3. Virtualized 4. Hybrid 5. Runtime Environment 6. Scalability 7. Pay-as-you-go 8. PaaS 9. IaaS 10. VT-x/AMD-V

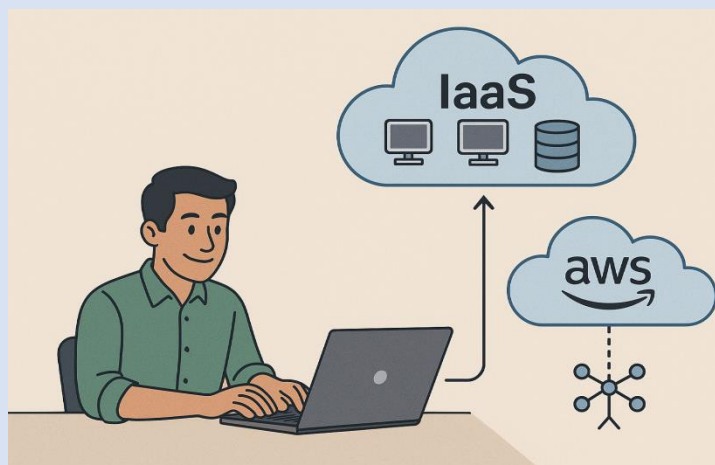
C. True or False

1. (T) 2. (F) 3. (F) 4. (T) 5. (F) 6. (T) 7. (T) 8. (T) 9. (F) 10. (T)

Chapter-2**Cloud Infrastructure as a Service (IaaS)**

Ravi, a technology entrepreneur, aimed to launch his new e-commerce platform swiftly. To avoid the expense of purchasing physical servers, he opted for Infrastructure as a Service (IaaS) for its adaptability, as illustrated. He effortlessly deployed virtual machines, integrated storage solutions, and set up network configurations through an online interface. During festive seasons, the platform scaled automatically, accommodating thousands of users with ease.

While Ravi maintained full control over his data and applications, the cloud provider managed the underlying hardware infrastructure. By leveraging IaaS, he bypassed the need for large capital investments in IT infrastructure and concentrated on business expansion. He utilized services from both AWS and Azure to ensure deployment across multiple geographic regions. Concerns about downtime were alleviated, thanks to automatic load balancing across servers. His team also performed data analytics and managed backups smoothly using integrated IaaS tools.



Infrastructure as a Service (IaaS) is a core model of cloud computing that delivers virtualized computing resources over the internet. It supplies foundational elements such as virtual machines, storage, networking, and load balancing, allowing organizations to operate applications and workloads without maintaining physical infrastructure. IaaS facilitates the quick deployment and scaling of resources as needed, making it well-suited for businesses seeking agility, cost savings, and rapid implementation of IT solutions.

In the IaaS model, users retain control over their operating systems, applications, and data, while the cloud service provider oversees the underlying physical components like servers, storage systems, and network hardware. Leading IaaS providers—including Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) offer a variety of services designed to meet different organizational requirements. This cloud model accommodates a broad spectrum of use cases, ranging from web hosting and enterprise software deployment to disaster recovery and high-performance computing scenarios.

2.1 Virtual Machines in the Cloud

Virtual Machines (VMs) serve as the core components of the Infrastructure as a Service (IaaS) model in cloud computing. A VM is a software-based simulation of a physical computer, capable of running an operating system and applications similarly to a physical server. Several VMs can operate on a single physical machine, isolated by a hypervisor, allowing efficient utilization of resources.

In cloud platforms, VMs can be created and deployed rapidly, enabling businesses to scale resources dynamically as demands change. Cloud vendors provide a variety of VM configurations general-purpose, compute-optimized, memory-optimized, and storage-optimized so that users can select instances based on processing power, RAM, disk space, and budget.

VMs support multiple operating systems and software stacks, allowing users to install custom applications, use pre-configured images, or create reusable custom images. Cloud providers such as AWS, Azure, and Google Cloud enable lifecycle management of VMs through management consoles, APIs, and Infrastructure as Code (IaC) tools like Terraform and CloudFormation.

Security and isolation are ensured through sandboxing, disk encryption, network segmentation, secure boot, and integration with IAM. VMs can also be deployed across multiple availability zones for high availability, fault tolerance, and disaster recovery. Additional features like snapshots, backups, and live migration provide data protection and seamless upgrades.

Performance monitoring tools further enhance VM management by collecting metrics such as CPU utilization and network throughput, enabling administrators to optimize resources, troubleshoot issues, and manage costs effectively.

2.2 Benefits of Using Virtual Machines in the Cloud

- **Cost Efficiency:** Cloud VMs save money because you pay only for what you use instead of buying costly hardware.
- **Scalability & Elasticity:** Cloud VMs can automatically increase or decrease in number when the demand for resources goes up or down.
- **Flexibility:** Users can select the type of VM they need and install any operating system or software they want.
- **Automation & Agility:** Cloud tools help create and manage VMs quickly and in the same way every time.
- **Security & Isolation:** Each VM is protected separately with features like encryption and secure login, so data stays safe.
- **High Availability & Disaster Recovery:** VMs can run in different data centers and recover quickly if one system fails.
- **Performance Optimization:** Cloud providers give monitoring tools to check performance, fix problems, and control costs.
- **Versatility:** VMs can be used for many purposes such as hosting websites, running company software, or testing applications.

VM Type	Description	Typical Use Cases
General-purpose	Balanced CPU and memory	Web servers, development environments
Compute-optimized	High-performance CPUs	Batch processing, high-performance computing
Memory-optimized	Large memory allocation	In-memory databases, real-time analytics
Storage-optimized	High IOPS and throughput storage	Big data workloads, data warehousing
GPU-accelerated	Graphics and compute acceleration via GPUs	Machine learning, AI workloads, 3D rendering

Practical Activity 2.1 Install a virtual machine**Materials Required:**

- Computer system (Windows/Linux/macOS)
- Internet connection
- Virtualization software (e.g., Oracle VirtualBox or VMware Workstation Player)
- ISO file of an operating system (e.g., Ubuntu Linux)

Practical Steps:**Step 1: Download and Install VirtualBox**

1. Open your web browser and go to: <https://www.virtualbox.org>
2. Click on "**Download VirtualBox**".
3. Choose your operating system (Windows, macOS, Linux).
4. Download and **run the installer** file.
5. Follow the on-screen instructions and click "**Next**" until installation is complete.
6. Click **Finish** to launch VirtualBox.

Step 2: Download Ubuntu ISO File (The Guest OS)

1. Visit: <https://ubuntu.com/download/desktop>
2. Click on "**Download Ubuntu Desktop**".
3. Save the ISO file (It's around 4–5 GB).

Note: An ISO file is like a virtual DVD of the operating system.

Step 3: Create a New Virtual Machine

1. Open **VirtualBox**.
2. Click "**New**".
3. Give your VM a name like "**UbuntuVM**".
4. Set **Type** as: Linux
5. Set **Version** as: Ubuntu (64-bit)
6. Click **Next**.

Step 4: Allocate RAM to the VM

1. Choose how much memory (RAM) to give the VM.
2. For Ubuntu, **2048 MB (2 GB)** is minimum.
3. Click **Next**.

Step 5: Create a Virtual Hard Disk

1. Select "**Create a virtual hard disk now**".
2. Click **Create**.
3. Choose **VDI (VirtualBox Disk Image)** → Click **Next**.
4. Choose **Dynamically allocated** → Click **Next**.
5. Set disk size to **20 GB or more** → Click **Create**.

Step 6: Load the Ubuntu ISO File

1. Select your VM from the left panel.
2. Click "**Settings**" → Go to the "**Storage**" section.
3. Under **Controller: IDE**, click on **Empty**.
4. On the right side, click the CD icon → **Choose a disk file**.
5. Select the **Ubuntu ISO file** you downloaded.
6. Click **OK**.

Step 7: Start the Virtual Machine

1. Click **Start** (green arrow button).
2. Ubuntu will start from the ISO file (like a live CD).

3. Select **“Install Ubuntu”** when prompted.

Step 8: Install Ubuntu Inside the Virtual Machine

1. Choose **language**, then click **Continue**.
2. Choose **"Normal installation"**.
3. Select **Erase disk and install Ubuntu** (*Don't worry, it's inside the VM only*).
4. Click **Install Now** → Continue.
5. Select your **location**, and set your **name**, **username**, and **password**.
6. Ubuntu will now install. This may take 10–15 minutes.
7. When done, click **Restart Now**.

2.3 Storage Services Types: Block, Object, File; Data Persistence; Access Control

Cloud storage plays a critical role in cloud computing by enabling the online storage, retrieval, management, and protection of data. As organizations increasingly migrate their IT infrastructure to the cloud, understanding the distinct types of cloud storage becomes essential. Cloud service providers typically offer three primary storage types: block storage, object storage, and file storage. Each type is designed for specific use cases, access patterns, performance requirements, and scalability demands.

Block storage segments data into fixed-sized units called blocks, each identified uniquely. Applications can access these blocks directly, without needing to retrieve the entire file. Functioning similarly to a locally attached hard drive, block storage is favored in use cases that require high performance, such as hosting databases or running virtual machines. Its architecture supports low-latency, high-throughput data operations.

In cloud platforms, block storage is usually delivered in the form of volumes that can be attached to virtual machines. Services such as Amazon Elastic Block Store (EBS), Azure Managed Disks, and Google Cloud Persistent Disks offer these capabilities. These volumes are detachable and can be reattached to other VMs, offering flexibility in managing workloads and resources.

This storage model is particularly well-suited for applications that demand high Input/Output Operations Per Second (IOPS), including transactional databases like MySQL and Oracle. Block storage supports standard file systems such as NTFS, ext4, and XFS, allowing users to format and use the volumes as though they were local drives. Common features include support for snapshots, encryption, and integration with backup tools.

Object storage is built to manage massive volumes of unstructured data. Data is stored as objects, each containing the data itself, relevant metadata, and a globally unique identifier. With its strong scalability and resilience, object storage is ideal for storing media files, system logs, backups, and outputs from big data processes. It operates without a traditional directory or folder structure.

Leading object storage offerings include Amazon S3, Azure Blob Storage, and Google Cloud Storage. These services are engineered to provide extreme durability—often up to 99.999999999% (11 nines)—through automatic replication across geographically distributed data centers. Access to data is typically available via RESTful APIs, enabling seamless integration with various applications and facilitating global accessibility.

The advantage of object storage lies in its metadata management and scalability. Users can attach custom metadata to objects, enabling enhanced search and data categorization. Moreover, object storage systems are virtually unlimited in capacity, allowing organizations to grow their data footprint without worrying about infrastructure constraints. Cost-efficiency is another benefit, especially for archival and infrequently accessed data, supported by tiered storage options like S3 Glacier.

File storage is a third model that closely mimics traditional shared file systems used in on-premises environments. Data is stored in a hierarchical directory structure using familiar file and folder conventions. File storage supports protocols such as NFS (Network File System) and SMB (Server Message Block), making it ideal for applications that require simultaneous access by multiple clients, like content management systems, home directories, and shared workspaces.

Major cloud providers offer file storage solutions including Amazon Elastic File System (EFS), Azure Files, and Google Filestore. These services provide managed file shares that can be mounted by cloud VMs or on-premises machines. One of the key advantages of file storage is its ease of use, especially for teams transitioning from legacy systems that rely on shared file access.

While file storage provides moderate performance compared to block storage, it supports concurrent access and is well-suited for distributed applications. With built-in redundancy, data is replicated across multiple zones or disks to ensure availability. Cloud file storage solutions also offer features such as automatic scaling, data encryption, access logs, and integration with identity services for secure authentication.

Beyond storage types, **Data persistence** is a critical concept in cloud storage. Data persistence refers to the ability of storage to retain data even after the system that created or used the data has been shut down. In ephemeral environments, like temporary virtual machines or containers, persistent storage must be provisioned externally using block, file, or object storage to ensure data longevity beyond instance lifespans.

Persistent storage is particularly crucial for stateful applications, databases, and systems that manage critical user or transaction data. Cloud providers often offer persistent volumes that are detached from compute instances, allowing them to be reassigned or reused as needed. This architecture ensures that data remains intact despite service restarts, failures, or scaling events.

Cloud platforms support replication and backup mechanisms as part of their persistent storage offerings. Replication involves maintaining copies of data across different physical locations to mitigate the risk of data loss due to hardware failure or disaster. Automated backup services allow users to schedule regular backups, maintain version histories, and restore data to specific points in time.

Access control in cloud storage is another cornerstone of secure and compliant data management. Access to data must be tightly controlled to prevent unauthorized exposure or manipulation. Cloud providers implement access control mechanisms at multiple levels, including identity-based, role-based, and policy-based access controls. These systems define who can access what resources and under what conditions.

Identity and Access Management (IAM) is a foundational component in cloud environments, used to authenticate users and assign specific roles. Each role is associated with defined permissions, enabling detailed and precise access control. For instance, a developer might be granted both read and write access to a storage bucket, whereas a business analyst could be limited to read-only access. In object storage systems, policies can be enforced at both the bucket and object level, incorporating conditions such as IP restrictions, time-based access, or encryption requirements.

Encryption plays a crucial role in safeguarding cloud-stored data and controlling access. Most cloud providers implement server-side encryption by default, automatically encrypting data during storage and decrypting it upon retrieval. Additionally, client-side encryption allows users to secure data before it enters the cloud, ensuring they retain control of the encryption keys and enhancing data privacy.

Audit logging is another essential security feature, providing insights into how and when data is accessed. These logs typically capture user identities, timestamps, actions taken, and source IP addresses. Such information supports anomaly detection, incident response, and regulatory compliance with standards like GDPR, HIPAA, and PCI-DSS.

To bolster security further, many cloud storage platforms support integration with external identity providers using protocols such as SAML or OAuth. These integrations enable features like single sign-on (SSO) and multi-factor authentication (MFA), reducing the likelihood of credential compromise and enhancing access governance.

In large scale enterprise settings, access policies must also consider factors like data sensitivity, geographic access constraints, and organizational workflows. For example, confidential financial records may only be accessible via the company's VPN, while publicly available resources like marketing assets can be made accessible over the internet. Through policy-driven management, organizations can automate such rules, effectively applying them across vast datasets and user groups.

Overall, the selection of storage types and associated security practices must align with the application's performance, compliance, and cost requirements. By choosing the right storage service block, object, or file combined with robust access control and persistent data strategies, organizations can achieve both operational efficiency and data integrity in the cloud.

Cloud storage is continuously evolving, with innovations like intelligent tiering, lifecycle policies, and serverless file storage shaping the next generation of data services. These advancements promise greater automation, cost savings, and scalability, enabling enterprises to manage exponential data growth effectively.

Understanding the differences among block, object, and file storage, along with implementing persistent data management and rigorous access controls, is fundamental to building secure and scalable cloud-native applications. Each storage type has a place in modern IT architecture, and when used appropriately, contributes to achieving resilience, performance, and regulatory compliance in the cloud ecosystem. Table 2.2. shows the storage types, their characteristics and uses.

Table 2.2: Storage types, characteristics and uses

Storage Type	Characteristics	Uses
Block Storage	High performance, low latency, attached to VMs	Databases, VM disk volumes, I/O intensive apps
Object Storage	Scalable, metadata-rich, accessed via APIs	Backup, media files, big data, web-scale apps
File Storage	Hierarchical structure, shared access, file protocols	File servers, collaborative apps, shared drives

Assignment. 2.1:

1. Explain how virtualization through a hypervisor enables multiple virtual machines to run on a single physical server in a cloud environment.
2. Describe two benefits of using cloud-based virtual machines for managing fluctuating application workloads.
3. Compare block storage and object storage in terms of structure, use cases, and performance characteristics.
4. Why is persistent storage essential in cloud environments, and how do cloud providers ensure data durability and availability?

2.4 Networking in the Cloud

Cloud networking is about connecting cloud resources, company systems, and users. It is the base of cloud computing and makes sure data flows safely and quickly. It helps in communication inside the cloud and also manages important services like IP addresses, DNS, and traffic distribution.

In cloud networking, normal networking devices like switches, routers, and firewalls are created and managed using software instead of only hardware. This makes the network more flexible, scalable, and easy to automate.

A key part of cloud networking is the **Virtual Private Cloud (VPC)**, which gives companies a private section in the public cloud. Using VPC, they can choose IP ranges, create subnets, set routing rules, and manage gateways securely.

Popular platforms like **AWS, Microsoft Azure, and Google Cloud** offer strong VPC services. For example, AWS VPC allows users to launch servers in a secure, private network with full control over security and routing.

Security in cloud networking is very important. Tools like **firewalls, access control lists (ACLs), and security groups** help control incoming and outgoing traffic to keep data safe.

To connect cloud with company data centers, businesses use **VPNs (Virtual Private Networks)** or direct lines like AWS Direct Connect. VPNs create safe tunnels over the internet, while direct lines give faster and more stable connections.

2.5 Cloud Networking and Benefits

Modern cloud networking uses **Software-Defined Networking (SDN)** and **Network Functions Virtualization (NFV)** to make networks more flexible and reduce the need for physical hardware. Services like load balancers, intrusion detection, and firewalls can run virtually in the cloud.

Other important services include:

- **DNS (Domain Name System):** Converts website names into IP addresses. Cloud providers like AWS Route 53 or Azure DNS make this faster and more reliable.
- **Load Balancing:** Spreads traffic across many servers so applications don't crash and stay available.
- **Content Delivery Networks (CDNs):** Deliver content (videos, images, websites) from servers close to users to reduce delay.
- **VPC Peering and Transit Gateways:** Allow safe communication between different cloud networks.

Cloud networking also supports **IPv6**, which provides many more IP addresses needed for IoT and mobile devices.

For security and performance, providers use:

- **QoS (Quality of Service):** Gives priority to important traffic.
- **Traffic Monitoring:** Tools like AWS Flow Logs track network usage and detect problems.
- **Private Links:** Connect services securely without using the public internet.

Table 2.3. Cloud Networking Components

Component	Description	Function
VPC	User-defined virtual network	Provides isolated networking environment
Subnet	IP address range within VPC	Organizes resources, segregates traffic
Load Balancer	Distributes network traffic	Improves availability and distributes workloads
VPN Gateway	Connects on-premises with cloud via secure tunnel	Enables hybrid cloud setups
NAT Gateway	Provides internet access for private resources	Maintains security while enabling outbound traffic
Route Table	Controls routing of traffic within VPC	Directs packets to correct destinations

Assignment.2.2:

1. What is a Virtual Private Cloud (VPC), and how does it contribute to secure and customizable cloud networking?
2. Explain how Elastic Load Balancing (ELB) and Content Delivery Networks (CDNs) enhance the performance and availability of cloud-based applications.

Practical Activity 2.2 Attach virtual block to the virtual machine and check whether it holds the data even after the release of the virtual machine

Materials Required:

- Cloud service account (e.g., AWS, Azure, GCP, or any virtualization platform like VirtualBox with simulated block storage)
- One virtual machine instance (e.g., AWS EC2, Azure VM)
- One virtual block storage volume (e.g., AWS EBS, Azure Managed Disk)
- SSH client or terminal access
- Internet connection

Practical Steps:**Step 1: Install Oracle VirtualBox**

1. Visit <https://www.virtualbox.org>
2. Download and install VirtualBox for your operating system (Windows, macOS, Linux).
3. Open VirtualBox after installation.

Step 2: Create a Virtual Machine

1. Click on **"New"**.
2. Give your VM a name (e.g., "TestVM").
3. Select the type as **Linux** (or **Windows** if you have the ISO).
4. Allocate memory (e.g., 1024 MB).
5. Choose **"Do not add a virtual hard disk"** for now.
6. Click **"Create"**.

Step 3: Create a Virtual Hard Disk (Virtual Block)

1. Go to **File** → **Virtual Media Manager**.
2. Click on **"Create"**.
3. Select **VDI (VirtualBox Disk Image)** → Click **Next**.
4. Choose **Dynamically allocated** → Click **Next**.
5. Give it a name (e.g., "MyBlockStorage") and size (e.g., 2 GB).
6. Click **Create**.

Step 4: Attach the Virtual Disk to the VM

1. Select the VM (e.g., TestVM) → Click **Settings**.
2. Go to **Storage**.
3. Click on the **empty controller icon (Add Hard Disk)** under SATA or IDE controller.
4. Choose **"Choose Existing Disk"** and select the **"MyBlockStorage.vdi"** file.
5. Click **OK**.

Step 5: Start the Virtual Machine and Add Data

1. Start the VM (you'll need an OS ISO like Ubuntu or Windows).
2. Inside the OS, open the **File Explorer** or **Disk Manager**.
3. Format and mount the new disk if prompted.

4. Create a folder or save a file (e.g., a text file named "test.txt").
5. Shutdown the VM after saving the file.

Step 6: Delete the Virtual Machine but Keep the Disk

1. In VirtualBox, right-click the VM → **Remove**.
2. Click "**Remove only**" (don't delete all files).
3. Now the VM is gone, but the disk **MyBlockStorage.vdi** is still saved.

Step 7: Attach the Disk to a New Virtual Machine

1. Create a new VM (name it "TestVM2").
2. Go to **Settings** → **Storage**.
3. Add the **existing disk (MyBlockStorage.vdi)** like before.
4. Boot the new VM.

Step 8: Check if the Data is Still There

1. Inside the new VM OS, open File Explorer or Terminal.
2. Browse the disk you should see the "**test.txt**" file.
3. This shows that the **data remains intact** even after the original VM was deleted.

2.6 Disaster Recovery in the Cloud

Disaster Recovery (DR) in the cloud means bringing back important data and systems after problems like cyberattacks, hardware failures, mistakes, or natural disasters. In the past, companies needed expensive backup data centers and complex setups for recovery. But now, cloud computing makes disaster recovery cheaper, faster, and easier.

With cloud DR, companies don't need to keep extra physical hardware. Cloud providers offer recovery services that can be quickly used during an emergency. Data and applications can be copied to different regions using services like **AWS, Microsoft Azure, and Google Cloud**. Automation tools such as **Terraform, AWS CloudFormation, and Azure Resource Manager** help create recovery environments faster and without errors.

Different types of cloud DR exist—from simple backups to advanced real-time replication. Two key terms are:

- **RTO (Recovery Time Objective):** How quickly systems should be restored.
- **RPO (Recovery Point Objective):** How much data loss is acceptable.

Cloud storage (like Amazon S3, Azure Blob Storage, or Google Cloud Storage) is also important, as it keeps backups safe and secure.

2.7 Features and Benefits of Cloud Disaster Recovery

Cloud DR uses not just storage but also computing services like **VMs, containers, and serverless functions** for fast recovery. Many providers also offer ready-to-use DR services such as **AWS Elastic Disaster Recovery** and **Azure Site Recovery**. Regular testing of DR plans is possible in the cloud without disturbing live systems, making organizations better prepared.

Some key features and benefits are:

- **High Availability:** Data is copied across multiple regions for safety.
- **Security:** Encryption and IAM (Identity and Access Management) protect data and restrict access.
- **Database Recovery:** Cloud databases like **Amazon RDS and Azure SQL** allow quick restoration using backups.
- **Multi-Cloud DR:** Data is stored across multiple cloud providers to avoid dependence on a single vendor.

- **Hybrid DR:** Combines on-premises systems with cloud recovery for extra safety.
- **Cost Saving:** Pay-as-you-go pricing means businesses pay only for what they use.
- **AI & ML in DR:** Artificial intelligence helps predict failures and automate recovery steps.
- **Business Continuity:** Ensures critical applications and communication remains active during disasters.
- **Edge DR:** Even edge devices (like IoT systems) can sync data with the cloud for recovery.

Cloud disaster recovery is modern, efficient, and cost-friendly. It gives faster recovery, stronger security, and better business continuity, making it essential for today's digital world.

Assignment.2.3:

1. How do cloud-based disaster recovery solutions reduce the need for traditional secondary data centres, and what are the cost benefits?
2. Differentiate between Recovery Time Objective (RTO) and Recovery Point Objective (RPO) in the context of cloud disaster recovery.
3. What role does automation and Infrastructure as Code (IaC) play in enhancing disaster recovery processes in the cloud?

SUMMARY

- VMs are software-based emulations of physical computers that form the core of IaaS, enabling scalable and isolated environments on shared hardware.
- Cloud VMs offer rapid provisioning, allowing users to scale resources dynamically through automation and APIs based on workload demands.
- The pay-as-you-go model for VMs reduces upfront costs, shifting CapEx to manageable OpEx and minimizing infrastructure maintenance.
- Elasticity in cloud VMs enables auto-scaling based on traffic, ensuring cost-efficiency and performance during fluctuating demand.
- Cloud VMs support diverse OSs, custom configurations, and robust security features, making them ideal for a wide range of computing needs.
- Cloud networking enables secure connectivity between cloud resources, on-premises environments, and end users, ensuring seamless data movement across systems.
- It leverages software-defined components like virtualized switches and routers to enhance flexibility, scalability, and automation in cloud environments.
- Key services like Virtual Private Clouds (VPCs), DNS management, and load balancing ensure secure, reliable, and optimized cloud connectivity.
- Emerging technologies like Service Mesh, Zero Trust Architecture, and 5G integration are shaping the future of cloud networking, enhancing security and performance.
- Cloud-based disaster recovery (DR) offers scalable, flexible, and cost-effective solutions, eliminating the need for dedicated physical infrastructure.
- Automation tools like Infrastructure as Code (IaC) enable rapid recovery by rebuilding environments in alternate regions with minimal downtime.
- Cloud DR supports various resilience tiers, from basic backup to real-time replication, meeting diverse Recovery Time Objectives (RTO) and Recovery Point Objectives (RPO).
- Cloud providers offer integrated DR solutions, secure data transfer, and geographic redundancy, ensuring minimal disruption during failures.
- Emerging technologies like AI and machine learning optimize disaster recovery by predicting failures, automating processes, and enhancing business continuity planning.

Check Your Progress**A. Multiple choice Questions**

1. What is the primary function of a Virtual Private Cloud (VPC) in cloud networking?
 - a) To manage DNS services
 - b) To enable secure, isolated network environments within a public cloud
 - c) To provide internet access to cloud resources
 - d) To optimize server performance in the cloud
2. Which of the following technologies is used to reduce dependency on hardware and enhance cloud networking agility?
 - a) Virtual Machine (VM)
 - b) Software-Defined Networking (SDN)
 - c) Domain Name System (DNS)
 - d) Content Delivery Networks (CDNs)
3. What service allows organizations to securely extend their cloud networks to on-premises environments?
 - a) Virtual Private Network (VPN)
 - b) Elastic Load Balancing (ELB)
 - c) Content Delivery Network (CDN)
 - d) Domain Name System (DNS)
4. Which cloud networking service helps to distribute incoming traffic across multiple servers or virtual machine instances to improve fault tolerance?
 - a) Elastic Load Balancing (ELB)
 - b) Virtual Private Cloud (VPC)
 - c) Transit Gateway
 - d) Software-Defined Networking (SDN)
5. Which of the following is a key component of Zero Trust Network Architecture (ZTNA)?
 - a) Role-Based Access Control (RBAC)
 - b) Network Address Translation (NAT)
 - c) Cloud Firewalls
 - d) Virtual Network Peering
6. What is one of the primary advantages of cloud-based disaster recovery (DR) over traditional DR methods?
 - a) Need for dedicated physical infrastructure
 - b) Higher capital expenditures
 - c) Elimination of idle servers and on-demand provisioning
 - d) Complex hardware replication
7. Which of the following cloud platforms provide built-in tools for disaster recovery?
 - a) AWS, Microsoft Azure, Google Cloud Platform
 - b) AWS, Oracle Cloud, IBM Cloud

- c) AWS, SAP Cloud, VMware Cloud
- d) AWS, Google Cloud, Oracle Cloud

8. What does the term 'Recovery Time Objective' (RTO) refer to in disaster recovery planning?

- a) The maximum tolerable data loss in time
- b) The maximum acceptable downtime after a disaster
- c) The time it takes to restore a database
- d) The maximum amount of data to be replicated

9. Which of the following cloud storage services is commonly used in disaster recovery strategies to store backups and historical versions of data?

- a) Amazon S3, Azure Blob Storage, Google Cloud Storage
- b) Google Drive, Dropbox, iCloud
- c) Amazon Glacier, Azure Files, Google Cloud Firestore
- d) AWS EBS, Google Cloud SQL, Azure Blob Storage

10. What is a key challenge of implementing a multi-cloud disaster recovery strategy?

- a) Higher security costs
- b) Compatibility and orchestration across platforms
- c) Lack of geographical redundancy
- d) Limited cloud service provider options

B. Fill in the Blanks:

1. A virtual machine runs on a software layer called the _____, which enables multiple VMs to share the same physical hardware.
2. The cloud's _____ model allows users to pay only for the virtual machine resources they consume, reducing upfront capital costs.
3. Cloud VMs support scalability through _____ features that automatically adjust the number of VM instances based on demand.
4. Users can deploy VMs using automated tools and scripts provided by _____ technologies like Terraform or AWS CloudFormation.
5. To ensure high availability, cloud providers distribute VMs across multiple _____ to minimize the impact of failures.
6. Cloud networking replicates traditional network functions in a virtual environment using _____ components.
7. A _____ is a logically isolated section of a public cloud where users can define IP ranges, subnets, and route tables.
8. _____ establishes secure encrypted tunnels over the internet for hybrid cloud connectivity.
9. Cloud-native DNS services like Amazon Route 53 and Azure DNS provide fast, reliable, and globally distributed _____ resolution.
10. _____ Load Balancing distributes incoming traffic across multiple servers to improve availability and fault tolerance.

C. True or False

1. Virtual Machines in the cloud cannot be automatically provisioned or scaled based on application demand.

2. Cloud-based virtual machines support multiple operating systems and allow users to install their own software stack.
3. Organizations using cloud VMs must still invest in and manage their own physical servers.
4. Block storage is best suited for storing large, unstructured data like multimedia files and logs.
5. Cloud file storage uses familiar file and folder structures and supports protocols like NFS and SMB.
6. Data stored in ephemeral virtual machines is automatically persistent across shutdowns and restarts.
7. Identity and Access Management (IAM) in cloud storage helps define who can access specific resources and under what conditions.
8. Cloud-based disaster recovery eliminates the need to maintain dedicated physical infrastructure for backups.
9. Recovery Time Objective (RTO) refers to the amount of data loss an organization can tolerate in a disaster.
10. Testing disaster recovery plans in the cloud is difficult and often disrupts production systems.

D. Short Answer Questions

1. What is the role of a hypervisor in managing virtual machines in the cloud?
2. List the three main types of cloud storage and mention one use case for each.
3. What is a Virtual Private Cloud (VPC) and why is it important in cloud networking?
4. How does auto-scaling work with cloud-based virtual machines?
5. What is the difference between block storage and object storage in terms of structure and usage?
6. What is the main advantage of using cloud-based disaster recovery compared to traditional methods?
7. Define RTO and RPO in the context of disaster recovery.
8. Name any two cloud providers that offer built-in disaster recovery services.
9. How does automation help in cloud-based disaster recovery?
10. What role does encryption play in securing cloud disaster recovery systems?

Answer Key

A. Multiple Choice Questions (MCQ)

1. (b) 2. (b) 3. (a) 4. (a) 5. (a) 6. (c) 7. (a) 8. (b) 9. (a) 10. (b)

B. Fill in the blanks

1. hypervisor, 2. pay-as-you-go, 3. auto-scaling, 4. Infrastructure as Code (IaC), 5. availability zones, 6. Software, 7. Virtual Private Cloud, 8. VPN (Virtual Private Network), 9. DNS, 10. Elastic

C. True or False

1. False 2. True 3. False 4. False 5. True 6. False 7. True 8. True 9. False 10. False

Chapter-3

Platform as a Service (PaaS)

Arya was a bright student who wanted to build a weather app. But she didn't have a big computer or a server to run it. One day, her teacher told her about PaaS – Platform as a Service. It's like a ready-made kitchen for developers — you bring the recipe (your app idea), and everything else like the stove, utensils, and ingredients (tools, coding environment, and storage) is already there. Arya used a PaaS platform, wrote her code online, tested it, and shared her app with friends – all without setting up anything on her own computer. She was amazed at how easy it was to create and run her app! Arya illustrated in Figure.



In this chapter, we will be able to learn about Platform as a Service (PaaS). PaaS provides developers with a fully managed cloud environment designed to support the entire application development lifecycle from creation and testing to deployment and maintenance.

3.1 Platform as a Service (PaaS) Overview and Use Cases

Platform as a Service (PaaS) is a cloud computing model designed to offer developers a complete environment for building, deploying, and managing applications without the need to handle the underlying infrastructure. With PaaS, essential components like servers, storage, networking, development tools, database systems, and analytics capabilities are provided through a centralized platform, enabling streamlined application development and deployment.

PaaS environments come pre-configured with essential services and tools, significantly reducing the complexity of setup and infrastructure management. Developers can focus solely on writing code and refining application logic rather than dealing with system updates, security configurations, or manual scalability. Typical PaaS offerings include built-in IDEs, application hosting capabilities, version control, and support for a variety of programming languages. Table 3.1 shows the PaaS features and their description.

Table 3.1. PaaS features and their description

PaaS Feature	Description
Infrastructure Abstraction	Eliminates need for hardware and OS management
Development Tools	Pre-configured IDEs, debuggers, testing tools
Scalability	Auto-scaling based on demand
CI/CD Integration	Support for agile workflows and automated pipelines
Multi-language Support	Works with Java, Python, Node.js, etc.
Use Cases	Web apps, APIs, mobile apps, analytics, IoT
Providers	AWS Elastic Beanstalk, Azure App Service, Google App Engine, Heroku

Use Cases of PaaS:

- **Web and App Development:** Developers can quickly build and deploy applications without managing servers or networking.
- **Mobile Application Development:** PaaS supports mobile app frameworks and tools, making it easy to create and test mobile apps.
- **API Development and Management:** PaaS helps developers build and host APIs for other apps or services.
- **Testing and Staging Environments:** Applications can be tested in a safe environment before deployment.
- **Business Analytics and Intelligence:** Some PaaS solutions include tools for analyzing data and generating reports.
- **IoT Applications:** Developers can use PaaS to build, deploy, and manage applications that control or process data from IoT devices.

Advantages of PaaS:

1. **Faster Development:** Pre-configured tools and templates speed up the app development process.
2. **Cost Savings:** No need to invest in hardware or infrastructure. Pay only for what you use.
3. **Scalability:** Easily scale apps up or down based on demand.
4. **Simplified Maintenance:** Providers handle security, updates, and system management.
5. **Supports Team Collaboration:** Developers can work from different locations on the same project.
6. **Focus on Coding:** Developers can focus on building features instead of managing infrastructure.

Assignment.3.1:

1. Explain how PaaS simplifies the application development process for developers?
2. Discuss the various use cases of PaaS with examples.
3. How does PaaS contribute to agile and DevOps practices in software development?

Practical Activity 3.1 Show the virtual machine migration based on the certain condition from one node to the other.

Materials Required:

- Two physical host machines with virtualization software (e.g., Proxmox VE, KVM, or VMware ESXi).
- A virtualization management tool (e.g., Proxmox GUI, oVirt, or vCenter).
- A shared storage setup (e.g., NFS or iSCSI) accessible by both hosts.
- One Virtual Machine created and running.
- Monitoring tool or command-line access to monitor CPU usage (e.g., htop, top, or virtualization dashboard).
- Preconfigured condition rule for migration (e.g., >80% CPU for 5 mins).

Practical Steps:

Step 1: Setup Environment: Ensure both physical hosts are configured in the same cluster and can access the shared storage.

- Deploy and power on a VM on Host A.

Step 2: Enable Live Migration:

- Ensure live migration is enabled on both hosts and that the shared storage is properly mounted.

Step 3: Configure Monitoring Rule:

- Set a threshold-based rule in the management interface (e.g., Proxmox HA Group or VMware DRS).
- Example: Migrate VM if CPU usage > 80% for more than 5 minutes.

Step 4: Generate CPU Load:

- SSH into the VM and run a CPU stress test using a tool like stress or sysbench.
- Example command: `stress --cpu 4 --timeout 600`.

Step 5: Trigger Migration:

- Wait for the management tool to detect high CPU usage and automatically initiate VM migration from Host A to Host B.

Step 6: Verify Migration:

- Check the management dashboard to verify the VM has migrated to Host B.
- Confirm that there was minimal or no downtime during the migration process.

Step 7: Cleanup:

- Stop the stress test.
- Optionally, revert any automated migration rules set up for the activity.

Practical Activity 3.2 Visual walkthrough of how multiple containers are managed**Materials Needed**

- A computer with:
 - **Docker Desktop** installed (Windows/Mac) OR Docker Engine (Linux)
 - **Docker Compose** installed
 - Code/text editor (e.g., VS Code)
 - Internet connection (to pull images)
- Terminal/Command Prompt

Practical Steps:**Step 1: Create a Project Directory**

- Open terminal
- Create a directory:
- `mkdir multi-container-demo && cd multi-container-demo`

Step 2: Create Docker Compose File

- Open your code editor
 - Create a file named `docker-compose.yml`
 - Add the following sample content:
- ```
version: '3'
services:
 web:
 image: nginx
 ports:
 - "8080:80"
 app:
 image: node
 command: node -e "require('http').createServer((_,res)=>res.end('Hello from Node')).listen(3000)"
 ports:
 - "3000:3000"
```

**Step 3: Start the Containers**

- In terminal, run:
- `docker-compose up -d`

**Step 4: List Running Containers**

- Run the command to see all containers:
- `docker ps`

**Step 5: Access the Services**

- Open a browser and visit:
  - `http://localhost:8080` → NGINX default page

- `http://localhost:3000` → Node.js message

#### Step 6: View Logs of Each Container

- For the web container:
- `docker-compose logs web`
- For the app container:
- `docker-compose logs app`

#### Step 7: Inspect Network Communication

- Check Docker network:
- `docker network ls`
- Inspect the default network created:
- `docker network inspect multi-container-demo_default`

#### Step 8: Stop and Remove Containers

- To bring down everything:
- `docker-compose down`

### Practical Activity 3.3 Demonstration of PaaS

#### Material Required:

1. Chart Paper (1-2 large sheets)
2. Markers (variety of colors)
3. Sticky notes or small paper cutouts
4. Printouts of cloud service diagrams or components (optional)
5. Glue or tape (for attaching printed materials, if needed)
6. Pen or pencil
7. Laptop/Smartphone (optional, for additional reference)

#### Instructions:

##### Step 1: Prepare the Chart Paper:

- Use a large piece of chart paper to outline the cloud architecture, which will represent the PaaS environment. Divide the paper into three main sections: **User Layer**, **Platform Layer**, and **Infrastructure Layer**.

##### Step 2: Draw the Architecture:

- **User Layer:** Draw a section at the top of the chart representing the user interface. This will include users who interact with the PaaS through applications. Label this section with components like "Developers", "End Users", and "Applications".
- **Platform Layer:** The middle section should illustrate the platform itself. This will include components such as "Operating System", "Database Management System", "Development Frameworks", and "Middleware". Draw boxes or circles for each of these components and connect them to show the relationship between them.
- **Infrastructure Layer:** At the bottom, draw the physical infrastructure, including servers, networking, and storage. This is the underlying layer that supports PaaS.

##### Step 3: Add Key Components:

- Label each part of the chart with descriptions. For example, under the Platform Layer, write about how developers can use programming languages, databases, and middleware to build applications without worrying about the underlying hardware.
- Use sticky notes or small cutouts to list and explain services included in PaaS, such as web hosting, database management, and development tools.

##### Step 4: Show Use Cases:

- On the side of the chart, include a small section listing real-world use cases of PaaS. Examples could include:
  - "Web Application Development"
  - "Mobile App Hosting"
  - "Data Analytics Platforms"
  - "IoT Application Support"

**Step 5: Add Interaction:**

- If possible, demonstrate how a developer would interact with a PaaS offering (like AWS Elastic Beanstalk, Google App Engine, or Microsoft Azure App Services) using simple diagrams or references. You can sketch this interaction in a step-by-step process.

**Step 6: Discuss the Benefits:**

Near the bottom of the chart, list the benefits of PaaS such as:

- "Scalability"
- "Cost Efficiency"
- "Simplified Development"
- "No Infrastructure Management"

### 3.2 Containerization (Docker)

Docker is a popular platform that helps developers build, run, and manage applications using **containers**.

A **container** is like a small box that carries an application along with everything it needs to run (libraries, files, settings).

Unlike **Virtual Machines (VMs)**, which need a full operating system for every app, containers share the same OS but keep the applications separate. This makes them:

- **Lightweight** (use fewer resources),
- **Faster** (start in seconds),
- **Portable** (can run anywhere – laptop, server, or cloud).

Docker has three main parts:

- **Docker Engine** – runs the containers,
- **Docker Image** – blueprint of what goes inside a container,
- **Docker Container** – the running version of an image.

Developers use **Dockerfiles** (a list of instructions) to build images. This ensures that the application runs the same way on every computer (no “it works on my system” problem).

Containers can communicate with each other through networks and can also save data permanently using **volumes**. Docker Hub (an online storehouse) allows sharing and downloading of pre-built images easily.

### 3.3 Features, Benefits, and Uses of Docker

Docker offers many advanced features that make it powerful:

- **Fast & Scalable** – Containers start in milliseconds and can be increased or reduced quickly.
- **Portable** – “Build once, run anywhere” (local, cloud, or hybrid systems).
- **Efficient Storage** – Images are stored in layers, so shared parts are saved only once.
- **Multi-Container Apps** – With **Docker Compose**, many containers can be managed together.

- **Security** – Containers are isolated, and extra tools (like SELinux, AppArmor) keep them safe.
- **Networking** – Different network types (bridge, host and overlay) allow containers to connect easily.
- **Performance Optimization** – By using small base images and fewer layers, apps run faster.
- **DevOps & CI/CD** – Docker fits perfectly in automated software pipelines, testing, and deployments.
- **Orchestration** – Tools like **Docker Swarm** or **Kubernetes** manage large numbers of containers.
- **Cloud Integration** – Works with AWS, Azure, and Google Cloud directly.

Students and developers can quickly set up labs, practice coding, and learn modern app development. Docker has changed how software is developed and delivered. Its speed, efficiency, and portability make it very important in today's cloud and DevOps world.

Table 3.2. Docker Concepts and Their Descriptions

| Concept            | Description                                                               |
|--------------------|---------------------------------------------------------------------------|
| Docker Engine      | Core component that runs and manages containers on a host machine         |
| Docker Image       | Blueprint for a container, including OS, application, and dependencies    |
| Docker Container   | Running instance of a Docker image with isolated execution environment    |
| Dockerfile         | Script defining steps to create a Docker image                            |
| Docker Hub         | Public registry to store and share Docker images                          |
| Docker Volume      | Persistent storage mechanism for containers                               |
| Docker Compose     | Tool to define and manage multi-container applications using YAML files   |
| Networking         | Built-in support for container communication via bridge, host, or overlay |
| Container Registry | Repository for storing and managing container images (e.g., Docker Hub)   |
| Docker Security    | Isolation, namespaces, and best practices for securing containers         |

### 3.4 Kubernetes Orchestration

Kubernetes (often called **K8s**) is a free and open-source platform used to **manage containers**. Containers are like small boxes that carry applications with everything they need to run.

Kubernetes was first created by **Google** and is now managed by the **Cloud Native Computing Foundation (CNCF)**. It helps in handling big and complex applications by **automating tasks** such as:

- Deploying applications,
- Scaling (increasing or decreasing resources),
- Managing workloads,
- Recovering from failures.

The **Kubernetes system** has two main parts:

1. **Control Plane** – It controls the whole cluster, decides where apps should run, and checks the system's health. Its important components are:
  - API Server
  - Scheduler
  - Controller Manager
  - etcd (stores cluster data)

2. **Worker Nodes** – These are the machines where applications actually run. Each node has:
- **Container Runtime** (like Docker)
  - **Kubelet** (connects node to control plane)
  - **Kube-proxy** (manages networking).

In Kubernetes, the smallest unit is called a **Pod**. A Pod may have one or more containers that work closely together and share the same network and storage.

### 3.5 Features and Advantages of Kubernetes

Kubernetes is very powerful because of its wide features:

- **Declarative Deployment** – Users simply write what they want in a file (YAML/JSON), and Kubernetes makes sure the system matches it.
- **Services** – A way to connect Pods with each other or make them available to users outside. (Types: ClusterIP, NodePort, LoadBalancer).
- **Namespaces** – Divide a cluster into groups for different teams or projects.
- **Self-Healing** – If a Pod or Node fails, Kubernetes restarts or moves it automatically.
- **Scaling** – It can add or remove Pods automatically based on CPU or memory use (Horizontal Pod Autoscaler).
- **Stateful Applications** – Using **StatefulSets** and **Persistent Volumes**, apps like databases can save data even if containers restart.
- **Configuration & Secrets** – Store app settings and sensitive data separately for better security.
- **Security** – Role-Based Access Control (RBAC), Network Policies, and Pod Security help protect apps.
- **Rolling Updates & Rollbacks** – Update apps without downtime, and easily go back if something breaks.
- **Monitoring & Logging** – Works with tools like Prometheus, Grafana, and ELK stack to check system health.
- **Helm** – A package manager that makes installing apps on Kubernetes easier.
- **Multi-Cloud & Hybrid** – Runs on any cloud (AWS, Azure, Google) or even on local servers.
- **DevOps Integration** – Works with CI/CD tools (like Jenkins, ArgoCD) for continuous delivery.

Kubernetes is like the **brain of containerized applications**. It makes apps **scalable, reliable, and portable**, which is why it has become the backbone of **modern cloud computing**.

#### Assignment.3.2:

1. Explain how containerization differs from traditional virtualization and describe the benefits it offers.
2. Describe the core components of Docker and how they work together to build and run containerized applications.
3. What are some key features and benefits of Kubernetes in managing containerized applications?
4. How do Docker and Kubernetes support DevOps practices?

### 3.6 Building and Deploying Applications in the Cloud

Cloud application development is about building software that runs on cloud platforms, using their benefits like **scalability, reliability, and managed services**. Developers can focus more on writing code and adding features instead of worrying about hardware or infrastructure.

The development process starts with **planning and design**. Many cloud apps use **microservices** or **serverless architectures**, which break applications into smaller, independent parts. This makes it easier to scale, update, and manage different components. Developers use cloud-ready tools such as online IDEs (like AWS Cloud9), version control (GitHub), and containers (Docker) to set up their work environment. These tools help teams collaborate smoothly, even if they are working from different locations.

Another key practice is **Continuous Integration (CI)**, where developers frequently add their code to a shared repository. Automated tools like Jenkins, GitLab CI/CD, or GitHub Actions test the code to ensure quality. After CI, **Continuous Delivery (CD)** pushes the code into staging or production environments automatically, making updates faster and more reliable. Cloud providers also offer tools like AWS CodePipeline and Azure DevOps for CI/CD.

### 3.7 Core Practices and Operations in Cloud Development

Cloud development heavily relies on **containers** to package applications so they run the same everywhere. Platforms like Kubernetes or ECS manage these containers. Infrastructure is set up and controlled using **Infrastructure as Code (IaC)** tools like Terraform or AWS CloudFormation, reducing manual errors.

To deploy updates safely, methods like **blue-green** and **canary releases** are used, which gradually shift users to the new version and roll back if needed. **Monitoring tools** such as Prometheus, Datadog, and CloudWatch track performance, logs, and errors to keep apps healthy.

**Security** is another major focus, with practices like encrypting data, strong IAM policies, and API gateways. Cloud providers also supply auditing tools to help enforce compliance. Developers often use **managed services** like Amazon RDS (for databases) or caching services to reduce operational work.

Most cloud apps use **auto scaling** and **load balancing** to handle changes in user traffic automatically. Updates are released through **rolling deployments** or **feature flags**, so new features can be tested gradually.

Other important aspects include **backup and disaster recovery**, **network design** (VPCs, firewalls, DNS), storing **artifacts** in registries like Docker Hub or ECR, and managing settings with **environment variables or secrets**. Cloud apps also depend on **APIs** for communication, often managed through API Gateways.

Testing is built into the process with unit, integration, performance, and security tests, often automated within the CI/CD pipeline. Finally, feedback from monitoring and user analytics helps teams **continuously improve** their applications, making cloud development fast, flexible, and future-ready.

#### SUMMARY

- PaaS provides a cloud-based environment for developing, testing, deploying, and managing applications.
- PaaS abstracts infrastructure management, allowing developers to focus solely on coding and innovation.
- Integrated tools like IDEs, middleware, and databases support rapid and efficient application development.
- PaaS platforms offer built-in scalability, enabling automatic adjustment to workload demands.
- Security, compliance, and maintenance are handled by the cloud provider, enhancing reliability.
- Supports agile practices with CI/CD pipelines for faster and safer software releases.
- Common use cases include web and mobile app development, API management, and data analytics.
- PaaS helps modernize legacy systems by supporting microservices and container-based architectures.

- Widely used in education, gaming, IoT, and AI/ML development for its ease of deployment and scaling.
- Popular PaaS providers include Google App Engine, AWS Elastic Beanstalk, Microsoft Azure App Services, and Heroku.

### Check Your Progress

#### A. Multiple choice Questions

1. What is the primary focus for developers when using a PaaS solution?
  - a) Managing infrastructure and networks
  - b) Coding and application logic
  - c) Configuring hardware manually
  - d) Installing operating systems
2. Which of the following is NOT typically included in a PaaS offering?
  - a) Virtual Machines
  - b) Integrated development tools
  - c) Middleware
  - d) Application hosting
3. What is one of the core benefits of PaaS in software development?
  - a) Manual scaling of resources
  - b) Custom server hardware configuration
  - c) Rapid prototyping and deployment
  - d) Elimination of the need for developers
4. Which of the following is a use case for PaaS?
  - a) Hosting physical data centers
  - b) Legacy database migration to on-premises servers
  - c) Real-time multiplayer game development
  - d) Manufacturing automation
5. Which of the following is a common PaaS provider?
  - a) Amazon EC2
  - b) Microsoft Azure App Services
  - c) Oracle VirtualBox
  - d) VMware vSphere
6. How does PaaS help support DevOps practices?
  - a) By limiting deployment automation
  - b) By enforcing manual testing only
  - c) Through CI/CD integration and automated monitoring
  - d) By avoiding cloud-based toolchains
7. What type of scalability does PaaS typically offer?
  - a) Vertical scaling only
  - b) No scalability
  - c) Manual scaling through admin consoles
  - d) Automatic scaling based on demand
8. Which component is NOT commonly part of a PaaS environment?
  - a) Version control systems
  - b) IDEs
  - c) Load balancers
  - d) Physical server rack configuration tools
9. How does PaaS enhance collaborative software development?
  - a) It restricts access to a single developer at a time

- b) It removes version control to simplify work
- c) It provides shared environments and version control tools
- d) It avoids debugging to improve speed

10. In the context of cloud deployment, which of the following is supported by PaaS platforms?

- a) Local-only deployment models
- b) Single-machine software testing
- c) Continuous integration and delivery (CI/CD)
- d) On-premises data backup

**B. Fill in the Blanks:**

1. Platform as a Service (PaaS) abstracts the underlying \_\_\_\_\_, allowing developers to focus on application development rather than hardware or system management.
2. PaaS provides integrated development tools, middleware, and \_\_\_\_\_ to support the entire application lifecycle.
3. Kubernetes, also known as \_\_\_\_\_, is an open-source container orchestration system for automating application deployment, scaling, and management.
4. Unlike traditional virtual machines, containers share the host \_\_\_\_\_ while still isolating application processes.
5. A/An \_\_\_\_\_ is a script used to define the instructions for building a Docker Image, including base images and commands.
6. PaaS platforms enable developers to focus on \_\_\_\_\_ by abstracting infrastructure management.
7. Built-in \_\_\_\_\_ allows PaaS platforms to automatically adjust resources based on workload demands.
8. PaaS supports agile methodologies through integrated \_\_\_\_\_ pipelines for faster and safer software releases.
9. With PaaS, \_\_\_\_\_, compliance, and system maintenance are managed by the cloud provider.
10. Popular PaaS providers include Heroku, AWS Elastic Beanstalk, Google App Engine, and \_\_\_\_\_.

**C. True or False**

1. Containers require a separate operating system to be bundled with each application, similar to virtual machines.
2. Dockerfiles are used to define the steps for building Docker Images and help ensure consistency across environments.
3. Docker containers cannot communicate with each other under any circumstances due to strict isolation.
4. Docker Compose uses a YAML file to define and manage multi-container applications, making local simulation of complex environments easier.
5. Kubernetes only supports horizontal scaling of applications and does not support vertical scaling.
6. Kubernetes enables high availability by automatically restarting failed containers and rescheduling pods if a node goes down.
7. PaaS allows developers to manage and configure hardware and network infrastructure manually.
8. Security and compliance responsibilities in PaaS are fully handled by the developers using the platform.
9. PaaS is suitable for use cases like mobile app development, API management, and data analytics applications.

10. PaaS environments typically lack tools for version control, debugging, and team collaboration.

**D. Short Answer Questions**

1. What is Platform as a Service (PaaS) and how does it benefit developers?
2. How do containers differ from traditional virtual machines?
3. What is the role of Dockerfiles in building Docker images?
4. What is Kubernetes used for in containerized environments?
5. Name two advantages of using Docker containers for application deployment.
6. What is the main purpose of using Platform as a Service (PaaS) in cloud application development?
7. What is the role of Continuous Integration (CI) and Continuous Delivery (CD) in cloud development?
8. Name any two tools used for Infrastructure as Code (IaC).
9. What is the function of container orchestration platforms like Kubernetes?
10. How do blue-green and canary deployments help in reducing risks during application updates?

**Answer Key****A. Multiple Choice Questions (MCQ)**

1. (b) 2. (a) 3. (c) 4. (c) 5. (b) 6. (c) 7. (d) 8. (d) 9. (c) 10. (c)

**B. Fill in the blanks**

1. Infrastructure 2. database 3. K8s 4. OS Kernel 5. Dockerfile 6. coding, 7. scalability, 8. CI/CD, 9. security, 10. Microsoft Azure App Services

**C. True or False**

1. False 2. True 3. False 4. True 5. False 6. True 7. False 8. False 9. True 10. False

**Chapter-4****Software as a Service (SaaS) and Cloud Applications**

Ananya was a high school student preparing for her final exams. She didn't have Microsoft Word or Excel installed on her laptop. Her friend Ravi suggested, "Why don't you use Google Docs and Sheets? They work online, and you don't have to install anything!" Ananya gave it a try. All she had to do was open her browser, sign in to her Google account, and start writing her assignments. Everything was saved automatically. She could even share her work with Ravi to get feedback instantly. One day, her laptop crashed. She panicked—what about all her notes? But then she smiled. "Everything is in the cloud," she said. "Thanks to SaaS!" She logged in on a friend's computer, and all her work was there—safe and ready. Happy Ananya is illustrated in Figure below.



In this chapter, we will learn detailed concept about the Software as a Service (SaaS). Software as a Service (SaaS) is a cloud delivery model where software applications are hosted by a provider and accessed by users over the internet. Rather than purchasing and installing software on individual devices, users typically interact with SaaS applications via web browsers. This approach offers advantages like improved ease of use, automatic updates, and reduced upfront expenses.

**4.1 Software as a Service (SaaS): Characteristics and Examples**

Software as a Service (SaaS) is a cloud delivery model where software is hosted by providers and accessed online, usually through a web browser and subscription model. It eliminates the need for installation, maintenance, and infrastructure management, allowing users to focus purely on using the application. SaaS is highly accessible, supporting remote work and multi-device usage. It operates on a multi-tenant architecture, ensuring secure data isolation while optimizing resources and simplifying updates for all users. Providers manage automatic updates and patches, keeping software secure and up to date without user intervention. SaaS is also scalable, letting organizations adjust usage easily based on demand, supported by flexible pricing models. Additional strengths include predictable costs through subscriptions, seamless integration with enterprise systems via APIs, and a shared security responsibility model, where providers ensure platform security while customers manage access and governance.

**Software as a Service (SaaS): Examples**

There are numerous SaaS applications widely adopted across industries. **Salesforce** revolutionized CRM by providing web-based customer relationship management. **Google Workspace** and **Microsoft 365** deliver productivity tools like email, documents, spreadsheets, and video conferencing for collaboration. Cloud storage platforms such as **Dropbox** and **Box** enable secure file sharing and version control. Customer support tools like **Zendesk** and **Freshdesk**, project management tools such as **Trello**, **Asana**, and **Monday.com**, and communication platforms like **Slack** and **Zoom** have become essential for modern teams. Finance is supported by **QuickBooks Online** and **Xero**, education by **Google Classroom** and **Canvas**, and e-commerce by **Shopify** and **BigCommerce**. Other examples

include **HubSpot** and **Mailchimp** for marketing automation, **NetSuite** for ERP, **Okta** for identity management, **Adobe Creative Cloud** for design, and **Workday** for HR. These examples show SaaS's versatility, powering everything from business collaboration to specialized industry needs. Characteristics are shown in Figure 4.1

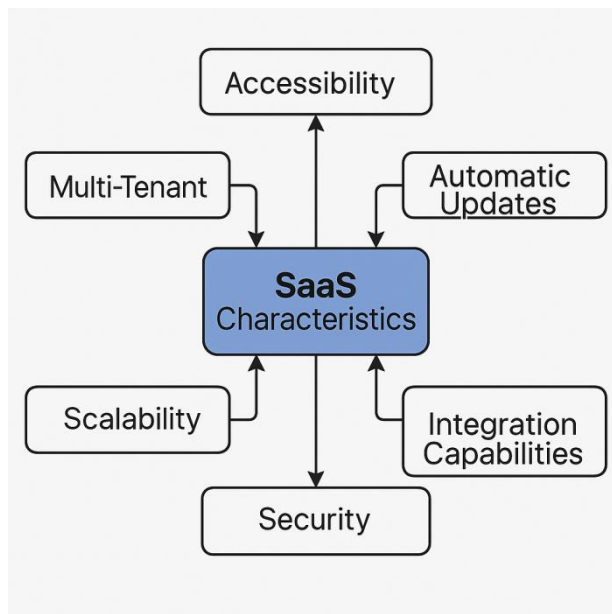


Fig. 4.1. SaaS Characteristics

Compliance and regulatory support are often built into SaaS solutions, especially those handling sensitive data. Many providers undergo certifications such as ISO 27001 (International Organization for Standardization), SOC 2 (System and Organization Controls 2), HIPAA (Health Insurance Portability and Accountability Act), and GDPR (General Data Protection Regulation) to assure clients of their data handling standards.

SaaS supports globalization and multi-tenancy across geographies. Companies can deploy applications to distributed teams and customers with multi-language, multi-currency, and region-specific compliance features.

The rapid growth of SaaS continues to transform how software is delivered and consumed. With innovations like AI integration, data analytics, and automation becoming standard, SaaS is not just a delivery model but a foundation for digital transformation.

#### Assignment.4.1:

1. What is Software as a Service (SaaS), and how does it benefit users compared to traditional software installation?
2. List and briefly describe any three examples of popular SaaS applications.
3. How does SaaS support collaboration and remote work?

**Practical Activity 4.1** Demonstration of procedure to transfer the files from one virtual machine to another virtual machine

#### Materials Required:

- Two Virtual Machines (VM1 and VM2), both running a Linux-based operating system (Ubuntu recommended)
- Network connectivity between the VMs (same subnet or proper network configuration)
- SSH access to both VMs (ensure SSH is installed and running on both machines)
- SCP utility (installed by default in most Linux distributions)

#### Practical Steps:

**Step 1:** Prepare VM1 and VM2

##### 1. Log in to VM1:

- Use the terminal to access VM1.

- Ensure SSH is enabled on both VMs. On Ubuntu, you can check the SSH service status by running:
    - `sudo systemctl status ssh`
  - 2. **Check the IP Address of VM2:**
    - Log into VM2 or get its IP address from the network configuration.
    - On VM2, run the following command to check the IP:
      - `ip a`
- Step 2:** Enable SSH Access on Both VMs
1. **Install SSH server (if not already installed):** On both VMs, run:
  2. `sudo apt update`
  3. `sudo apt install openssh-server`
  4. **Verify SSH is running:** On both VMs, check that the SSH service is running:
  5. `sudo systemctl status ssh`
- Step 3:** Transfer a File from VM1 to VM2
1. **Create a test file on VM1:** On VM1, create a sample file to transfer:
  2. `echo "This is a test file for transfer" > /home/user/testfile.txt`
  3. **Transfer the file using SCP:** Use the scp command to copy the file from VM1 to VM2. From VM1, run the following command:
  4. `scp /home/user/testfile.txt username@VM2_IP:/home/user/`  
Replace username with the actual username on VM2, and VM2\_IP with the IP address of VM2.
    - You'll be prompted to enter the password for the user on VM2.
  5. **Verify the Transfer on VM2:**
    - Log in to VM2 and check if the file has been successfully transferred:
  6. `ls /home/user/testfile.txt`
- Step 4:** Troubleshooting
- If the SCP command fails, ensure that:
    - Both VMs are on the same network or can reach each other over SSH.
    - SSH is correctly configured on both VMs.
    - The firewall is not blocking port 22 (SSH).

## 4.2 Cloud-based Productivity Applications

Cloud-based productivity applications are software tools that allow users to create, collaborate, manage, and share work-related content over the internet. These tools are typically offered through a SaaS model and are designed to boost efficiency, streamline workflows, and support remote and hybrid work environments.

The foundation of cloud productivity lies in its **real-time collaboration** capabilities. Multiple users can work on a single document, spreadsheet, or presentation simultaneously, seeing each other's changes in real time. This reduces version conflicts and improves team synchronization.

One of the most prominent suites is **Google Workspace**, which includes Gmail, Google Docs, Sheets, Slides, Calendar, and Meet. These tools are browser-based and tightly integrated, allowing users to manage communication, documentation, scheduling, and meetings from a unified interface.

Similarly, **Microsoft 365** offers cloud versions of familiar desktop applications such as Word, Excel, and PowerPoint, alongside OneDrive, Outlook, and Microsoft Teams. These applications support real-time co-authoring and cloud storage to ensure seamless access across devices.

**Document management** is a critical aspect of productivity. Cloud-based platforms like Dropbox, OneDrive, and Box provide centralized storage with version control, access

permissions, and sharing capabilities. They ensure that documents are always up-to-date and recoverable in case of data loss.

**Task management and planning** tools like Trello, Asana, and Todoist allow users to organize work into boards, lists, and calendars. These tools are essential for managing personal to-do lists or complex project timelines with team involvement.

**Communication platforms** are the lifeline of cloud productivity. Slack and Microsoft Teams combine instant messaging, video calls, file sharing, and integrations with third-party applications. These tools promote quick decision-making and reduce reliance on email.

Cloud-based **video conferencing** solutions like Zoom, Google Meet, and Webex are now integral for virtual meetings, especially in remote-first organizations. They support high-definition video, screen sharing, breakout rooms, and real-time captioning.

**Calendar and scheduling tools** ensure that teams stay organized. Google Calendar and Outlook Calendar offer shared calendars, event invitations, and reminders. Integration with conferencing and project management tools adds further efficiency.

**Note-taking and idea capture** have gone digital with tools like Evernote, Notion, and Microsoft OneNote. These applications allow users to write, organize, and tag notes with media attachments, links, and collaborative editing capabilities.

Productivity apps often integrate with **cloud storage** for automatic saving and retrieval. This ensures that work is never lost due to local device failure and supports mobile access, versioning, and collaboration on the go.

Cloud-based productivity apps enhance **remote workforce enablement**. Employees can work from any location without needing VPN access to internal resources. This democratizes access and supports business continuity during disruptions.

These tools are **platform-agnostic**, working across Windows, macOS, Android, and iOS environments. Whether on a desktop, tablet, or smartphone, users enjoy a consistent experience and full functionality. Figure 4.2 shows the Cloud based applications.

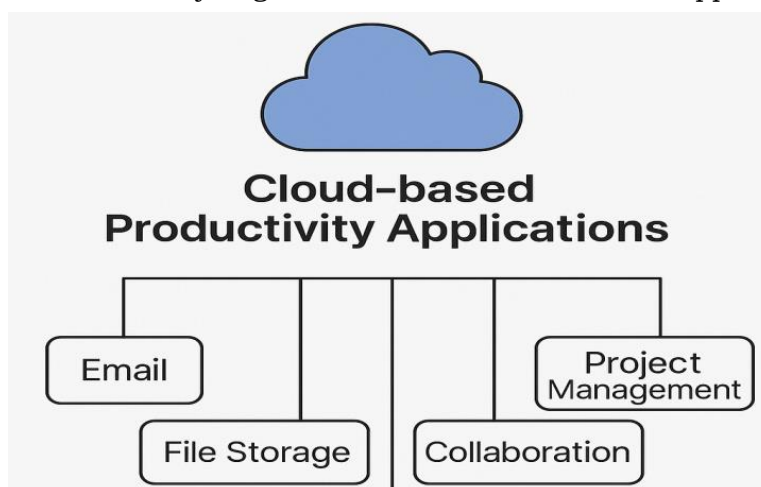


Fig 4.2 Cloud based applications

**Security features** such as two-factor authentication, encryption, and centralized access control are built into most productivity apps. Admins can enforce policies, manage user roles, and monitor activity to maintain compliance and protect data.

**Automation and integration** with third-party tools are becoming a norm. Platforms like Zapier or native integrations allow users to connect productivity apps with CRMs, HR systems, and marketing platforms, enhancing workflows and reducing manual tasks.

Cloud-based applications are ideal for **scaling with organizational growth**. Startups, SMEs, and enterprises can use the same tools, adjusting subscriptions as their workforce expands or contracts without infrastructure changes.

**Data analytics** and usage insights are also integrated into these platforms. Administrators can track application usage, collaboration trends, and engagement levels, enabling data-driven decisions on workflow improvements.

Cloud-based applications ensure **continuous delivery of features and improvements**. Users automatically receive new functionalities, bug fixes, and performance enhancements, ensuring a modern and efficient work environment.

**Templates and AI enhancements** speed up content creation. Smart compose in Google Docs, design suggestions in PowerPoint, or Excel's predictive formulas help users work faster and smarter with less effort.

For teams working across **different time zones**, cloud productivity tools provide asynchronous collaboration options. Users can leave comments, assign tasks, or tag colleagues in shared documents, enabling communication beyond real-time constraints.

These platforms support **cross-functional collaboration**. Marketing, sales, development, and operations teams can co-create documents, track milestones, and manage campaigns from a centralized cloud environment.

**Content publishing and sharing** are streamlined. From creating client-ready reports to internal newsletters, cloud tools support direct publishing to websites, intranets, or external sharing with controlled access.

Many tools provide **mobile-first experiences**, allowing users to respond to messages, edit documents, or join meetings on the go. Push notifications and offline modes further enhance mobile productivity.

Cloud-based productivity platforms reduce **IT maintenance overhead**. Organizations no longer need to install, update, or troubleshoot applications on every device. Centralized management through admin consoles streamlines user provisioning and support.

In the education sector, cloud tools like Google Classroom, Microsoft Teams for Education, and Canva for Education empower instructors and students to collaborate, distribute materials, and manage assignments from a shared virtual environment.

The future of productivity applications is being shaped by **generative AI and machine learning**, with capabilities like summarizing documents, suggesting action items, or automating routine responses based on historical data patterns.

Cloud-based productivity applications have revolutionized how individuals and organizations operate. They provide the infrastructure for collaboration, creativity, and communication in the digital age, making work more agile, connected, and intelligent.

### 4.3 Collaboration Tools in the Cloud

Cloud collaboration tools allow people from different locations to work together easily. They support file sharing, real-time communication, task management, video conferencing, and more. These tools are flexible, scalable, and easy to use. A major advantage is **accessibility**, as team members can access files and projects from any internet-connected device. Real-time data synchronization ensures that updates appear immediately for all collaborators, reducing confusion and version conflicts. **Security features** like permissions, identity management, and encryption protect data, while **cloud storage** services such as Google Drive, OneDrive, and Dropbox make it easy to store and share files. Many platforms also include **task and project management, video conferencing, and instant messaging**, allowing teams to organize, communicate, and collaborate efficiently. Cross-platform support, third-party integration, AI-driven features, and collaboration analytics further enhance productivity. Features like document co-authoring, cloud whiteboards, offline access, backup and recovery, scalability, and cost-efficient subscription models make these tools reliable for both small teams and large organizations.

### 4.4 Examples and Applications of Cloud Collaboration Tools

Popular cloud collaboration platforms include **Google Workspace** and **Microsoft 365**, which provide shared documents, email, and virtual classrooms. **Trello, Asana, and Monday.com** help manage tasks and projects, while **Slack, Microsoft Teams, and Discord** support instant messaging and team chat. **Zoom, Google Meet, and Microsoft Teams** offer video

conferencing capabilities. In software development, **GitHub** and **GitLab** allow developers to collaborate on code, conduct peer reviews, and deploy updates. Healthcare professionals use **HIPAA-compliant tools** to share patient information securely. Creative collaboration is supported by platforms like **Miro** and **Microsoft Whiteboard** for brainstorming and visual planning. Enterprises benefit from multi-tenant support, compliance with **GDPR, ISO 27001, and SOC 2**, and unified collaboration hubs that combine chat, meetings, file sharing, and app integration. These tools help organizations innovate faster, maintain team productivity, and adapt to hybrid or remote work environments.

#### **Assignment.4.2:**

1. What are cloud-based productivity applications?
2. List any four popular cloud productivity tools and describe their primary use.
3. How do cloud-based productivity tools support real-time collaboration?
4. What are some security features commonly found in cloud-based collaboration platforms?
5. Discuss two benefits and two challenges of using cloud-based productivity tools in organizations.

#### **Practical Activity 4.2** Demonstration of SaaS

##### **Materials Needed**

- Chart Paper (1-2 large sheets)
- Markers (variety of colors)
- Sticky notes or small paper cutouts
- Glue or tape (for attaching printed materials, if needed)
- Pen or pencil

##### **Practical Steps:**

1. **Title & Layout:**
  - Write the title "**Software as a Service (SaaS)**" at the top of the chart.
  - Divide the chart into 3 parts:
    - **SaaS Cloud Provider**
    - **Internet/Cloud**
    - **End Users/Clients**
2. **Cloud Provider Section:**
  - Paste logos of SaaS providers like **Google Workspace, Microsoft Office 365, Dropbox, Salesforce.**
  - Write short descriptions under each logo describing what service they provide.
3. **Cloud (Middle) Section:**
  - Draw a big cloud in the center, label it as "Internet / Cloud Platform".
  - Use arrows to show the flow of service from the provider to users.
4. **User Side Section:**
  - Create figures of laptops, tablets, and smartphones.
  - Paste icons representing different users: student, business user, developer, etc.
  - Use speech bubbles/sticky notes to show what users say:
    - "Accessing Gmail from browser"
    - "Editing file on Google Docs"
    - "Using Salesforce CRM online"
5. **Connectivity:**
  - Use string/thread or drawn arrows to show connectivity from cloud provider → internet → user.
6. **Features Section (optional):**
  - Add a side area with points like:
    - No installation needed
    - Access from anywhere
    - Subscription-based
    - Automatic updates

### 4.5 Integration with Cloud APIs

Cloud APIs (Application Programming Interfaces) serve as essential connectors in cloud computing, enabling applications to communicate programmatically with cloud services. These APIs facilitate integration, support automation, and empower developers to create cloud-native applications that utilize resources like storage, computing power, AI, and security features.

Fundamentally, cloud APIs abstract the complexity of the underlying infrastructure. Rather than manually setting up servers or configuring services through graphical interfaces, developers can write code to interact directly with these APIs, automating such tasks. This approach accelerates development, ensures consistency, and enhances scalability.

APIs can be categorized by their purpose: Infrastructure APIs, Platform APIs, Service APIs, and Application APIs. Infrastructure APIs manage basic resources such as virtual machines and networking, while Service APIs handle higher-level offerings like databases, messaging, and analytics.

Leading cloud providers including AWS, Microsoft Azure, and Google Cloud Platform provide extensive APIs covering nearly all their services. These APIs come in various styles REST, GraphQL, and gRPC allowing developers to choose based on their specific needs and preferences.

REST APIs are the most prevalent and widely adopted. They operate over standard HTTP methods like GET, POST, PUT, and DELETE, and typically exchange data in JSON or XML formats. Their stateless design makes them highly suitable for scalable interactions in cloud environments.

Cloud APIs play a pivotal role in DevOps workflows. Continuous Integration and Continuous Deployment (CI/CD) systems rely heavily on APIs to automate infrastructure provisioning, application deployment, and system monitoring, reducing the need for manual tasks.

Security is a critical aspect of cloud API usage, as illustrated in Figure 4.3. Providers enforce strict authentication and authorization through mechanisms such as OAuth, API keys, and Identity and Access Management (IAM) roles. These measures ensure that only authorized entities can access APIs, safeguarding data privacy and maintaining system integrity.

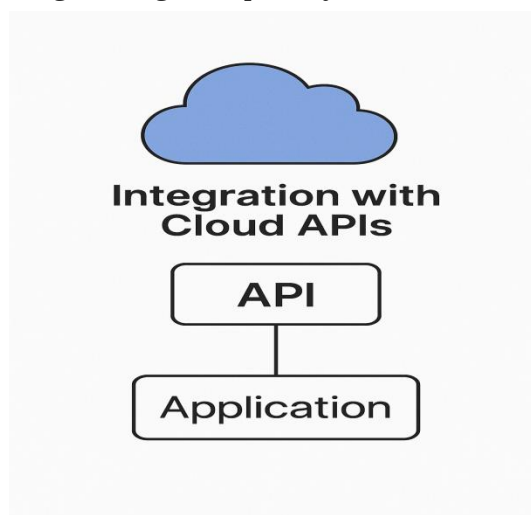


Fig. 4.3. Cloud APIs

### 4.6 Cloud API Connectivity

Integration with cloud APIs allows developers to connect services across different cloud platforms, making multi-cloud and hybrid cloud setups possible while keeping workflows consistent. APIs are crucial for serverless and event-driven applications, where actions like uploading a file to a storage bucket can automatically trigger functions such as AWS Lambda. They also help with monitoring and logging by providing programmatic access to metrics, logs, and events, which is important for managing large-scale systems. Many apps, including mobile and web applications, rely on cloud APIs to fetch data and provide interactive user

experiences, such as a weather app using Google Cloud APIs to display real-time weather updates. Security APIs help enforce encryption, identity checks, firewalls, and threat detection, while analytics and AI APIs, like Amazon Recognition or Google Vision, add intelligence like image recognition, natural language processing, and machine learning to applications.

APIs also manage costs by tracking cloud expenses and budgets, maintain compliance and auditing by logging user activity, and enable scalable storage and database interactions without manual effort. API gateways improve reliability by managing traffic, enforcing authorization, caching, and rate limiting, and SDKs in languages like Python, Java, Node.js, and Go simplify integration for developers. Workflow automation tools like Zapier and Microsoft Power Automate use APIs to link services and automate repetitive tasks, while testing tools like Postman and Swagger ensure integrations work correctly. API versioning and lifecycle management allow updates without breaking existing apps, and security practices like input validation, encryption, and token use prevent vulnerabilities. Cloud APIs improve business agility by enabling component reuse, third-party integration, and faster service launches, and they are essential in edge computing and IoT, where devices communicate with cloud services in real time.

APIs form the backbone of cloud-native architecture, allowing microservices to communicate efficiently, making systems modular, scalable, and easier to maintain, and mastering API integration is key for modern cloud application development.

#### 4.7 Customizing SaaS Applications

Customizing Software as a Service (SaaS) helps businesses adapt general software to meet their specific needs. This can include changing the look of the application, such as themes, logos, layouts, and user interface, to match a company's branding. Workflows can also be tailored, like adjusting sales processes in a CRM. Businesses can add or change data fields and forms to collect the right information. Role-based access control allows different users to see only what they need. Conditional rules, formulas, and scripting help automate tasks and follow business logic. Many SaaS platforms let users use plugins, modules, or low-code/no-code tools to add new features without touching the core system. Multi-tenant systems let each organization customize independently, and white-labeling lets companies rebrand the software entirely. Localization allows support for different languages, currencies, and regulations. Sandbox environments are provided to test changes safely before applying them to the main system.

#### 4.8 Extending SaaS Applications

Extending **Software as a Service (SaaS)** means connecting it with other tools and adding extra functionality. **Application Programming Interfaces (APIs)** let the software communicate with other platforms, like **Enterprise Resource Planning (ERP)** systems, payment gateways, or collaboration tools. **Webhooks** can trigger real-time actions between systems. Custom scripts allow advanced automation, while app marketplaces provide pre-built add-ons like analytics, chatbots, or document management. Monitoring and logging help track performance and fix errors. Security and compliance must be maintained, following rules like **General Data Protection Regulation (GDPR)**, **Health Insurance Portability and Accountability Act (HIPAA)**, or **System and Organization Controls 2 (SOC 2)**. Updates from the SaaS provider must be considered, ensuring customizations continue to work after upgrades. Careful planning prevents vendor lock-in and ensures custom features do not interfere with system performance. Customization and extension together give businesses a SaaS application that fits their workflow, branding, and goals.

##### **Assignment.4.3:**

1. What is the primary role of cloud APIs in cloud computing environments?
2. Differentiate between Infrastructure APIs and Service APIs with one example of each.
3. How do cloud APIs support DevOps practices ?

4. Why is authentication and authorization critical in cloud API integration, and what mechanisms are commonly used?

**SUMMARY**

- SaaS delivers software over the internet, eliminating local installations and simplifying maintenance through subscription-based access.
- It enables remote and cross-device access, fostering real-time collaboration, mobility, and productivity.
- SaaS platforms are scalable, cost-efficient, and automatically updated, making them ideal for businesses of all sizes.
- Integration with third-party tools and robust security features empowers users to build customized, secure workflows.
- Cloud-based collaboration tools like Google Workspace, Microsoft 365, and Slack streamline teamwork through shared documents, messaging, and video conferencing.
- Cloud APIs enable seamless, programmatic interaction with cloud services, driving automation and cloud-native development.
- Cloud APIs abstract infrastructure complexity, allowing developers to manage cloud resources through code instead of manual configuration.
- Cloud providers offer diverse APIs REST, GraphQL, gRPC for services like compute, storage, AI, security, and analytics.
- APIs support DevOps, serverless architectures, hybrid/multi-cloud deployments, and secure access via IAM, OAuth, and API keys.
- API gateways, SDKs, monitoring tools, and security practices enhance API reliability, scalability, and integration flexibility.

**Check Your Progress****A. Multiple choice Questions**

1. Which of the following best describes the SaaS model of cloud computing?
  - a. Software is installed and maintained on users' personal devices.
  - b. Applications are delivered over the internet and managed by a service provider.
  - c. Users manage the servers and infrastructure of the software.
  - d. Software is distributed via physical media like CDs or USBs.
2. What is a key benefit of multi-tenancy in SaaS applications?
  - a. Users can customize the server's operating system.
  - b. Each user gets a dedicated instance of the application.
  - c. Multiple customers share the same software instance with data isolation.
  - d. It prevents resource sharing among users.
3. Which of the following is NOT typically a characteristic of SaaS platforms?
  - a. Automatic updates
  - b. Browser-based access
  - c. Local hardware requirements for each user
  - d. Subscription-based pricing
4. How do SaaS applications support remote and hybrid work environments?
  - a. By requiring VPN access to local servers
  - b. Through real-time collaboration and device-agnostic access
  - c. By installing software on all user devices
  - d. Through local database synchronization
5. Which example below is a SaaS platform designed specifically for customer relationship management (CRM)?
  - a. Dropbox
  - b. Trello

- c. Salesforce
  - d. Canva
6. What is the primary benefit of using Cloud APIs in application development?
- a. They increase the cost of development
  - b. They allow manual configuration of infrastructure
  - c. They automate interactions with cloud services
  - d. They require developers to access physical hardware
7. Which type of Cloud API is commonly used for creating, deleting, and moving files across storage systems?
- a. Monitoring API
  - b. Storage API
  - c. Database API
  - d. Security API
8. Which authentication method is commonly used to secure Cloud APIs?
- a. FTP
  - b. OAuth
  - c. Telnet
  - d. POP3
9. What feature allows SaaS platforms to provide real-time integration with external systems when an event occurs?
- a. Database triggers
  - b. Webhooks
  - c. Role-based access control
  - d. Multi-tenancy
10. What is a key risk associated with over-customizing a SaaS application using proprietary APIs or scripting?
- a. Reduced branding flexibility
  - b. Vendor lock-in
  - c. Higher system performance
  - d. Improved compliance

**B. Fill in the Blanks:**

1. In the **Software as a Service (SaaS)** model, applications are accessed via a \_\_\_\_\_ instead of being installed on individual devices.
2. SaaS platforms are typically \_\_\_\_\_, meaning a single instance of the software serves multiple customers while keeping their data isolated.
3. One of the major advantages of SaaS is that updates and patches are handled \_\_\_\_\_ by the service provider.
4. SaaS applications support scalability, allowing businesses to scale up or down without needing to purchase additional \_\_\_\_\_ or software licenses.
5. Cloud APIs act as communication bridges that allow applications to interact with \_\_\_\_\_ services programmatically.
6. REST APIs typically use standard HTTP methods such as GET, POST, PUT, and \_\_\_\_\_.
7. SaaS customization often involves modifying the application's look and feel, such as changing themes, layouts, and \_\_\_\_\_.
8. Role-based access control allows organizations to assign fine-grained \_\_\_\_\_ for different users.
9. Low-code and no-code platforms use drag-and-drop tools to allow customization without deep \_\_\_\_\_ skills.

10. SaaS platforms often support localization by allowing configuration of multiple languages, currencies, and \_\_\_\_\_ formats.

**C. True or False**

1. Cloud APIs allow developers to interact with cloud services only through graphical user interfaces.
2. Security is not a concern while integrating with cloud APIs, as most services are already protected by the provider.
3. API Gateways help in managing, securing, and monitoring API traffic in cloud environments.
4. Webhooks are used in SaaS platforms to trigger real-time actions based on specific events.
5. SDKs are not necessary for API integration because APIs are self-sufficient.
6. Over-customizing a SaaS application using proprietary APIs may lead to vendor lock-in.
7. SaaS applications require users to install software locally on their devices to function properly.
8. One of the benefits of SaaS is that the service provider manages software updates and maintenance.
9. In a SaaS environment, each customer has a completely separate software instance to ensure data isolation.
10. Organizations using SaaS must invest heavily in IT infrastructure to support scalability.

**D. Short Answer Questions**

1. What is Software as a Service (SaaS) and how is it accessed by users?
2. What does a multi-tenant architecture in SaaS mean?
3. Mention any two key advantages of using SaaS.
4. Name any three popular SaaS-based applications used for productivity or communication.
5. In SaaS, who is responsible for managing platform-level security—provider or user?
6. How do cloud collaboration tools enhance teamwork across different geographical locations?
7. What is the benefit of real-time data synchronization in cloud-based document editing?
8. Name two popular cloud storage services used for file sharing and collaboration.
9. What role do tools like Trello and Asana play in cloud collaboration platforms?
10. How do cloud collaboration tools ensure security and control over shared data?

**Answer Key****A. Multiple Choice Questions (MCQ)**

1. (b) 2. (c) 3. (c) 4. (b) 5. (c) 6. (c) 7. (b) 8. (b) 9. (a) 10. (b)

**B. Fill in the blanks**

1. web browser 2. multi-tenant 3. automatically 4. hardware 5. cloud 6. DELETE 7. logos  
8. permissions 9. coding 10. date

**C. True or False**

1. False 2. False 3. True 4. True 5. False 6. True 7. False 8. True 9. False 10. False

**Chapter-5****Cloud Security and Governance**

Riya loved taking photos and saving school projects on her cloud drive. One day, she heard that someone's account got hacked, and all their files were stolen. Worried, Riya asked her teacher, "Is storing things on the cloud safe?" Her teacher smiled and replied,

"Yes, Riya. Cloud providers use strong security tools like encryption and two-step verification to keep your data safe. It's like having a strong lock and secret code on your locker." Riya also learned that cloud companies follow governance rules, which means they manage and protect user data properly, like using clear rules for who can access what and when. From that day on, Riya felt confident using the cloud, knowing her digital world was guarded by smart locks and trusted rules.



In this chapter, we will learn in detailed about Cloud security and governance. Cloud security and governance are critical components of any cloud computing strategy, ensuring that data, applications, and infrastructure are protected against threats while complying with organizational policies and regulatory standards. Cloud security focuses on safeguarding cloud-based systems through measures like encryption, access controls, threat detection, and secure configurations, while governance establishes the framework for managing cloud resources responsibly and transparently.

**5.1 Security Challenges in the Cloud**

Cloud security is a crucial concern for organizations adopting cloud services. As businesses increasingly migrate to the cloud, they must address various challenges that arise in securing data, applications, and infrastructure.

One of the primary concerns is **data confidentiality**. Cloud environments often involve shared resources and multi-tenant architectures, making it difficult to ensure that sensitive data is only accessible to authorized users. Without strong encryption and robust access control mechanisms, businesses face the risk of unauthorized access to critical information.

Another challenge is **data integrity**. With data being stored and processed across multiple locations and managed by third-party providers, it can be difficult to ensure that data has not been tampered with. Ensuring that data remains consistent and accurate is particularly important for industries such as healthcare, finance, and government, where regulatory compliance demands the integrity of data for audit trails and reporting.

**Identity and access management (IAM)** is a major hurdle in cloud security. Managing user identities across different cloud platforms requires effective IAM policies. This includes controlling who has access to which resources and ensuring that access is only granted on a need-to-know basis. The lack of strong IAM can lead to the misuse of privileges or unauthorized access to cloud resources, particularly in scenarios where employees or third-party vendors may have excessive access rights.

**Shared responsibility models** in the cloud introduce another challenge. Cloud providers are responsible for securing the infrastructure, while customers are responsible for securing their applications and data. This division of responsibility can lead to misunderstandings about who is accountable for certain security aspects. Inadequate clarity on the delineation of responsibilities often leaves gaps in security, exposing customers to risks.

The **lack of visibility** in cloud environments is another significant issue. Traditional on-premise security models allow businesses to monitor and control their infrastructure directly. In the cloud, however, visibility into security operations is limited. Organizations may not have full insight into the underlying infrastructure or the ability to monitor all traffic within the environment. This lack of visibility complicates threat detection, making it harder to identify and respond to security incidents in a timely manner.

The **dynamic and flexible nature of the cloud** also presents challenges. Cloud resources can be quickly spun up and scaled down, which means that the attack surface is constantly changing. Attackers can exploit misconfigurations or weaknesses in rapidly deployed resources before organizations have the chance to apply security controls. This elasticity of the cloud requires constant vigilance and automated security measures to keep up with the evolving environment.

Another challenge is **compliance** with regulatory frameworks and industry standards. Different countries and industries have varying data protection regulations, such as GDPR in Europe, HIPAA in the United States, and PCI-DSS for payment card information. Organizations must ensure that their cloud service providers comply with these regulations, but the responsibility for ensuring compliance often falls on the customer. This can be complex and requires careful consideration when selecting cloud providers and configuring cloud environments to meet regulatory requirements.

**Data residency and sovereignty** are additional concerns in the context of cloud security. Data stored in the cloud may be processed in data centers located in different jurisdictions, potentially subjecting it to different laws and regulations. For businesses that deal with highly sensitive data, ensuring that data is stored and processed in compliance with local laws becomes critical. This often requires clear agreements with cloud providers to define where data can reside and how it is protected across borders.

Another challenge comes from **cloud provider outages and disruptions**. While cloud providers strive for high availability, they are not immune to technical failures, natural disasters, or cyberattacks that could lead to service downtime or data loss. Organizations must have contingency plans, including disaster recovery strategies, to ensure business continuity in case of such incidents. This includes leveraging backup solutions and designing fault-tolerant architectures to mitigate the risk of outages.

**Multi-cloud and hybrid cloud** environments introduces additional complexity. Many organizations choose to use multiple cloud providers or combine on-premise and cloud infrastructure to avoid vendor lock-in or to meet specific needs. However, managing security across multiple platforms can be difficult. It requires consistent policies and practices across different environments, as well as tools that provide unified visibility and control over security across all cloud platforms.

The **threat landscape in the cloud** is constantly evolving. Cybercriminals are increasingly targeting cloud environments with sophisticated attacks such as Distributed Denial of Service (DDoS) attacks, phishing, and ransomware. Cloud environments are often seen as lucrative targets because they house a wealth of valuable data. Protecting against these evolving threats requires cloud security strategies to be adaptive and proactive.

Cloud security also requires **strong network security practices**. Cloud networks are often more complex than traditional on-premise networks, with traffic flowing between various cloud regions, applications, and services. Securing these networks involves implementing firewalls, intrusion detection systems, and VPNs, among other measures. Without proper network security, organizations may be vulnerable to attacks that exploit weak points in the network infrastructure.

Another challenge is the **security of APIs** in cloud environments. APIs are essential for enabling communication and integration between various cloud services and applications. However, poorly secured APIs can be an entry point for attackers. Securing APIs requires

adopting best practices such as authentication, encryption, and rate limiting to prevent misuse.

**DevSecOps**, which integrates security into the development and operations process, is becoming increasingly important in cloud environments. Security must be baked into every phase of application development, from design to deployment, to minimize vulnerabilities. Implementing DevSecOps requires collaboration between developers, security teams, and operations staff to ensure that security controls are applied consistently throughout the software development lifecycle.

Cloud service providers offer various **security features**, but customers must configure and implement these features correctly. Misconfigurations are one of the leading causes of security incidents in the cloud. Ensuring that cloud configurations adhere to best practices and are regularly reviewed can mitigate the risk of exposure. Automated configuration management tools can assist in maintaining consistent and secure configurations across the cloud environment.

**Cloud security training and awareness** are essential for mitigating human error, which remains one of the most significant security risks. Employees and contractors must be educated about security best practices, phishing attacks, and the importance of using strong passwords. Regular security awareness training can help ensure that all individuals interacting with the cloud environment understand the risks and how to mitigate them.

Cloud vendors are continuously enhancing security features, but the **pace of innovation** in cloud technology also introduces new challenges. As cloud providers release new services and functionalities, organizations must stay informed about potential security implications. Regularly updating security policies and adapting to new technologies are crucial for keeping up with the rapidly changing cloud landscape.

Cloud security challenges are **multifaceted** and require a comprehensive approach that includes strong encryption, robust access controls, visibility, compliance, and proactive threat detection. Organizations must understand the shared responsibility model and ensure that both their cloud provider and internal security practices align to safeguard their data and applications. By staying vigilant and adopting best practices, businesses can mitigate security risks and confidently embrace the advantages of the cloud.

#### Assignment.5.1:

1. Why is identity and access management (IAM) critical in cloud security, and what are the risks of poor IAM practices?
2. Explain the shared responsibility model in cloud security. How can misunderstanding it lead to security gaps?
3. What are some of the main challenges organizations face in securing data integrity in cloud environments?

#### Practical Activity 5.1 Discuss the case studies related to Cloud attack

Materials Required:

- **Internet-enabled Computer System** (1 per participant or group)
- **Case Study Printouts or PDFs** (provided by the instructor)
- **Note-taking Materials or Lab Notebook**
- Access to the following websites
- Lab worksheet template (provided below)

#### Case Study 1: Leaky S3 Bucket (Amazon Cloud)

##### Scenario

A company stored confidential documents in an Amazon S3 bucket but didn't set the right access permissions. The bucket was **public**, allowing anyone with the link to view the files. What Went Wrong?

- Misconfigured access controls
- No authentication needed to view sensitive files

##### Solution

1. Set the bucket to private

2. Use IAM roles and policies for access control
3. Enable logging and monitoring

**Practical Classroom Activity**

**Task:** Use a simulation tool or AWS Educate sandbox (if available).

- Create an S3 bucket
- Upload a sample file (e.g., notes.txt)
- Set public access and then fix it
- Observe access before and after change

**Case Study 2: Cloud Ransomware via Email****Scenario**

An employee opened a fake email with a malicious link. It accessed the employee's cloud storage (e.g., Google Drive) and encrypted the files.

What Went Wrong?

- Phishing attack successful
- No antivirus or security controls
- Files not backed up

**Solution**

1. Educate employees about phishing
2. Use 2FA (Two-Factor Authentication)
3. Backup cloud data regularly

**Practical Classroom Activity**

**Task:**

- Show a fake phishing email example (demo only)
- Ask students to identify the suspicious elements
- Discuss how 2FA and backups could prevent data loss

**Case Study 3: Unsecured API Access****Scenario**

A developer exposed a cloud API (for a weather app) but didn't secure it. Hackers used it to flood the server (DDoS attack), causing high bills and service outage.

What Went Wrong?

- No authentication or rate-limiting
- API was publicly accessible

**Solution**

1. Use **API keys and authentication**
2. Apply **rate limits**
3. Monitor API usage

**Practical Classroom Activity**

**Task:**

- Use Postman or a mock API
- Show how requests are made with and without API key
- Discuss the need for rate limits and security tokens

**Bonus Activity: Group Discussion / Role Play**

- **Form teams:** Attackers, Defenders, Cloud Admins
- **Scenario:** One team creates a simulated attack, others defend
- **Goal:** Spot security issues and suggest fixes

**5.2 Threats in Cloud**

Cloud security and governance are critical components of any cloud computing strategy, ensuring that data, applications, and infrastructure are protected against threats while complying with organizational policies and regulatory standards. Cloud security focuses on safeguarding cloud-based systems through measures like encryption, access controls, threat detection, and secure configurations, while governance establishes the framework for managing cloud resources responsibly and transparently.

Effective cloud governance provides a structured approach to decision-making, risk management, and compliance by defining roles, responsibilities, and processes. It ensures that cloud usage aligns with business goals, optimizes costs, and adheres to legal and regulatory requirements. Together, cloud security and governance create a resilient cloud environment that supports innovation without compromising control or safety.

The cloud has transformed the way businesses store and manage their data, but it has also introduced new security threats that organizations must address. One of the most significant threats is data breaches, where sensitive information is accessed or disclosed by unauthorized individuals. Given the shared nature of cloud environments, where multiple clients share the same infrastructure, an attacker can exploit vulnerabilities in one tenant's system to gain access to another's data. This makes it imperative for organizations to ensure that strong encryption and access control mechanisms are in place to protect their information.

Denial of Service (DoS) attacks are another prevalent threat to cloud security. In a DoS attack, an attacker overwhelms a cloud service with massive amounts of traffic, rendering the service unavailable to legitimate users. In a cloud environment, DoS attacks can target the entire infrastructure of a cloud service provider or a specific application, causing significant downtime and business disruption. Organizations must implement DDoS protection strategies to mitigate this risk, which often involves leveraging the cloud provider's infrastructure to scale resources dynamically and maintain service availability.

Data loss is another considerable threat that organizations face in the cloud. Cloud services may experience issues such as system failures, data corruption, or inadvertent deletion, which can result in the permanent loss of data. Although most cloud providers implement redundancy and backup measures to protect against data loss, organizations must also create their own backup and disaster recovery plans. Data loss can have severe consequences, especially for industries with strict compliance regulations, such as healthcare and finance, where data integrity is essential for legal and operational reasons.

Insider threats remain a major concern in the cloud. Employees, contractors, or other individuals with authorized access to cloud systems can intentionally or unintentionally misuse their access to compromise data or systems. Insider threats are often harder to detect because the perpetrators have legitimate access. Therefore, organizations need to adopt strict Identity and Access Management (IAM) policies, continuously monitor user activity, and enforce the principle of least privilege to reduce the potential impact of insider threats.

Inadequate API security is another critical threat in cloud environments. APIs are fundamental for cloud services, allowing different systems and applications to communicate with each other. However, weak or poorly configured APIs can expose cloud systems to vulnerabilities, making them susceptible to attacks such as man-in-the-middle (MITM) or SQL injection. Secure coding practices, API rate limiting, and proper authentication mechanisms are essential to mitigate API vulnerabilities and protect sensitive data.

Malware and ransomware attacks are increasingly targeting cloud environments. Malware can infect cloud infrastructure, spreading across systems and compromising sensitive data. Ransomware—a type of malware that encrypts data and demands payment for its release—can cripple an organization's operations if critical data or systems are locked. Since cloud environments often host mission-critical applications, ensuring robust endpoint protection and having a solid backup strategy are vital for mitigating these risks.

Misconfiguration of cloud resources is one of the leading causes of security incidents. Cloud environments provide users with significant control over their resources, but this flexibility can be a double-edged sword. A simple misconfiguration, such as leaving a storage bucket open to the public, can expose sensitive data to unauthorized access. Organizations must implement strict configuration management practices, continuously monitor cloud environments for misconfigurations, and use automated tools to detect and correct security weaknesses.

Another threat arises from inadequate monitoring and logging of cloud activities. Without comprehensive monitoring, organizations may fail to detect security breaches or unusual behavior in a timely manner. Cloud providers offer tools to track user activity and system performance, but organizations need to ensure that they are actively logging and analyzing

this data for signs of malicious activity. A lack of proper logging can hinder incident response and forensic investigations following a security breach.

Third-party risks are also a growing concern in the cloud. Many organizations rely on third-party vendors for various cloud services, including Software as a Service (SaaS) and Platform as a Service (PaaS). While outsourcing these services can provide operational efficiencies, it also introduces additional security risks. The security posture of a third-party vendor may not align with an organization's own security practices, potentially exposing sensitive data or systems to compromise. It is essential for businesses to conduct thorough due diligence and establish clear security requirements with their cloud providers and third-party vendors.

Account hijacking is another threat that affects cloud services. In an account hijacking attack, cybercriminals steal login credentials to gain unauthorized access to cloud systems. Once attackers have control of an account, they can cause significant damage, such as stealing or deleting data, changing security settings, or performing other malicious activities. Protecting against account hijacking involves implementing strong multi-factor authentication (MFA), regularly updating passwords, and educating users about the risks of phishing and other credential-stealing tactics.

Cross-site scripting (XSS) and other injection attacks are common threats in cloud-hosted web applications. XSS vulnerabilities allow attackers to inject malicious scripts into web pages, which are then executed by users' browsers. Similarly, SQL injection attacks can allow attackers to manipulate databases through vulnerable input fields. Cloud applications must be developed with security in mind, including input validation and output encoding to prevent these types of attacks.

Privilege escalation is another significant threat in cloud environments. In privilege escalation attacks, an attacker gains elevated privileges within a cloud environment, allowing them to perform unauthorized actions or access sensitive data. These attacks can occur when an attacker exploits vulnerability in a cloud application or service to gain higher-level access. Organizations should implement least-privilege access models, conduct regular access reviews, and enforce strict access controls to reduce the risk of privilege escalation.

The multi-cloud and hybrid cloud models, while offering flexibility and redundancy, can also introduce additional risks. Managing security across multiple cloud platforms can be complex, and inconsistency in security practices can lead to vulnerabilities. Organizations need to ensure that their security policies are aligned across all cloud environments and that tools for monitoring and managing security are integrated. This reduces the risk of gaps in security and makes it easier to respond to threats across different platforms.

Supply chain attacks are another emerging threat in cloud environments. Attackers may target the software development lifecycle or supply chain to compromise cloud-based systems. By exploiting vulnerabilities in third-party libraries, open-source software, or development tools, attackers can introduce malicious code into cloud applications. Organizations must vet their suppliers and software providers carefully, monitor for vulnerabilities, and implement security measures such as code reviews and software signing to mitigate the risk of supply chain attacks.

Cloud services that rely on shared resources are especially vulnerable to side-channel attacks. In a side-channel attack, an attacker exploits weakness in the physical infrastructure to gather information—such as power consumption or electromagnetic emissions—to extract sensitive data. Since cloud providers often share physical hardware among multiple tenants, attackers could potentially access data from other customers using these methods. While this type of attack is more difficult to execute, it remains a potential risk in certain cloud environments.

As more organizations adopt cloud services, cloud-based social engineering attacks are becoming more common. These attacks exploit human behavior to gain access to cloud systems. Common tactics include phishing emails that trick users into revealing login credentials or fake technical support calls designed to gather personal information. It's critical for organizations to invest in security awareness training and phishing prevention tools to protect employees from falling victim to social engineering tactics.

Cloud providers' security features are often robust, but they are not foolproof. Cloud services typically offer a wide array of tools for securing applications, databases, and networks.

However, the configuration of these tools is ultimately the responsibility of the customer. Misconfiguring cloud security settings, even with the best tools in place, can leave organizations exposed to threats. To mitigate this risk, companies must regularly audit their cloud environments and ensure they are utilizing the full range of security features provided by their vendors.

Cloud providers are also targeted by advanced persistent threats (APTs), which are stealthy, prolonged attacks aimed at stealing valuable data or infiltrating systems over an extended period. These attacks are often highly sophisticated and difficult to detect, with attackers using various methods to stay hidden in cloud networks. Organizations must work with their cloud providers to ensure that they have strong defenses against APTs, including continuous monitoring, advanced threat detection systems, and incident response capabilities.

The threats faced in cloud environments are diverse and constantly evolving. Data breaches, DDoS attacks, insider threats, and misconfigurations are just a few of the risks organizations must address to protect their cloud infrastructure. To mitigate these threats, businesses must adopt a comprehensive approach to cloud security involving proper risk management, continuous monitoring, and strong collaboration with their cloud service providers. Only through proactive security practices can organizations fully benefit from the scalability and flexibility the cloud offers while minimizing the associated risks.

### Practical Activity 5.2 Introduction to Security tools

#### Material Required:

| Item                      | Details                                          |
|---------------------------|--------------------------------------------------|
| Computer System           | Windows/Linux-based system with internet access  |
| Virtual Machine Software  | VirtualBox or VMware Workstation                 |
| Operating System (VM)     | Kali Linux or Parrot Security OS                 |
| Nmap                      | Network scanning tool (pre-installed in Kali)    |
| Wireshark                 | Network protocol analyzer (can be installed)     |
| Nikto                     | Web server vulnerability scanner (pre-installed) |
| Sample vulnerable web app | DVWA or OWASP Juice Shop (for safe scanning)     |
| Admin privileges          | Required for installing and running tools        |

#### Part 1: Network Scanning with Nmap

1. Open the terminal in Kali Linux.
2. Run a basic scan on your network:
3. `nmap -sP 192.168.1.0/24`
4. Perform a more detailed scan on a specific IP:
5. `nmap -sV 192.168.1.10`
6. Record open ports and services.

#### Part 2: Traffic Analysis with Wireshark

1. Open Wireshark and select the active network interface.
2. Start capturing packets.
3. Visit a website (e.g., <http://example.com>).
4. Stop the capture after 1 minute.
5. Use filters like `http`, `tcp.port == 80`, etc., to analyze traffic.

#### Part 3: Vulnerability Scanning with Nikto

1. Launch terminal in Kali Linux.
2. Run Nikto against the sample vulnerable web app:
3. `nikto -h http://192.168.1.20`
4. Observe the output for vulnerabilities like XSS, outdated software, etc.

**Observations Table:**

| Tool      | Target       | Command Used                 | Findings/Output                         |
|-----------|--------------|------------------------------|-----------------------------------------|
| Nmap      | 192.168.1.10 | nmap -sV 192.168.1.10        | Open ports: 22, 80; Services: SSH, HTTP |
| Wireshark | example.com  | Filter: http                 | GET requests, HTTP headers, packet size |
| Nikto     | 192.168.1.20 | nikto -h http://192.168.1.20 | XSS found, outdated Apache version      |

### 5.3 Identity and Access Management

Identity and Access Management (IAM) is a critical component of any organization's security framework. IAM involves defining and managing the roles and access privileges of users within a network, ensuring that only authorized individuals have access to specific resources. The core function of IAM is to guarantee that individuals are who they say they are and that their access to systems, applications, and data is appropriate based on their role and responsibilities.

At the heart of IAM is identity verification. This process ensures that users are who they claim to be, typically through mechanisms such as passwords, biometrics, security tokens, or multi-factor authentication (MFA). MFA adds an additional layer of security by requiring users to provide more than one form of verification, making it more difficult for malicious actors to gain unauthorized access.

Authentication is only one part of the IAM puzzle. Once a user's identity is verified, access control mechanisms come into play. Access control defines what resources a user is allowed to interact with and what actions they can perform. This is typically implemented through roles and permissions, which are assigned based on the user's position within an organization and their specific needs. The principle of least privilege is a key concept in access control, ensuring that users only have access to the resources necessary for their work.

Role-based access control (RBAC) is one common approach to managing access. RBAC allows administrators to create roles with specific privileges and then assign users to these roles. By grouping users into roles based on their job functions, organizations can efficiently manage permissions and reduce the risk of unauthorized access. This method is especially useful in large organizations where users may have similar access needs based on their role in the company.

In addition to RBAC, other models such as attribute-based access control (ABAC) and discretionary access control (DAC) are used in IAM systems. ABAC assigns access based on attributes such as the user's department, location, or security clearance, while DAC gives the owner of the resource control over who can access it. Each of these models offers advantages depending on the organization's size, structure, and security requirements.

Managing access across multiple systems and applications can be a significant challenge, especially as organizations increasingly adopt cloud-based platforms. Single sign-on (SSO) is a solution to this challenge, allowing users to authenticate once and gain access to multiple applications without having to re-enter their credentials. SSO improves user experience by reducing the number of login prompts and simplifying access management for administrators.

In a modern IAM environment, federated identity management (FIM) is another important concept. FIM enables users to access applications across different domains or organizations without needing separate credentials for each one. By establishing trust relationships between identity providers, federated identity systems enable seamless access across different platforms, facilitating collaboration while maintaining security.

IAM systems must be able to scale with an organization's growth. As companies add new employees, systems, and applications, the IAM system must be flexible enough to accommodate changing needs. This scalability is essential for ensuring that access

management remains efficient and effective as the organization expands—especially in dynamic environments like cloud computing, where resources can be provisioned on demand. Monitoring and auditing are essential components of a robust IAM strategy. Continuous monitoring of user activity helps identify suspicious behavior and potential security threats. IAM systems should log user actions and generate reports that allow administrators to review access patterns and investigate any anomalies. Auditing ensures compliance with internal security policies and external regulatory requirements such as GDPR, HIPAA, or PCI-DSS.

A crucial aspect of IAM is managing the lifecycle of user identities. From onboarding to offboarding, the process of creating, updating, and deleting user accounts must be tightly controlled. When employees leave an organization or change roles, their access rights must be promptly adjusted or revoked to prevent unauthorized access. Automating these processes reduces the risk of human error and ensures that access rights are consistently enforced.

Integration with other security tools is also essential for effective IAM. For instance, IAM systems can be integrated with security information and event management (SIEM) systems to provide a comprehensive view of security events. This integration allows for quicker identification of potential threats, as IAM data such as login attempts and access patterns can be correlated with other security information to detect malicious activities.

The adoption of cloud technologies presents unique challenges for IAM. Traditional IAM systems were designed for on-premises infrastructure, but as businesses move to the cloud, new strategies are required. Cloud IAM solutions enable organizations to manage access to cloud services and resources while providing the same level of control and security as on-premises systems. These solutions often support features such as SSO, MFA, and automated provisioning to help organizations maintain security across both on-premises and cloud environments.

Managing privileged access is another critical component of IAM. Privileged accounts, such as those of system administrators, have elevated access rights and can pose significant security risks if compromised. It is essential to implement strict controls over privileged access, such as monitoring the use of these accounts and limiting their number. Privileged access management (PAM) tools can help organizations secure and manage these accounts by enforcing policies such as least privilege and session recording.

With the increasing use of mobile devices, IAM must also extend to mobile platforms. Mobile device management (MDM) solutions allow organizations to enforce security policies on employee-owned or corporate-issued mobile devices. Integrating IAM with MDM ensures that users can securely access corporate resources from their mobile devices while also ensuring that devices are compliant with security standards.

IAM is not just a technical issue but also a strategic one. As organizations become more reliant on digital transformation, IAM becomes a key enabler of business agility and security. By ensuring that the right people have access to the right resources at the right time, IAM helps organizations protect sensitive data, comply with regulations, and minimize the risks associated with unauthorized access.

Security awareness training is an essential part of any IAM strategy. Employees must understand the importance of strong passwords, multi-factor authentication, and other security measures. Ongoing training can help prevent social engineering attacks, such as phishing, which often target users to gain access to systems and data. Employees should also be educated about the consequences of poor access management, such as data breaches or identity theft.

In the age of advanced persistent threats (APTs) and insider attacks, IAM plays a crucial role in detecting and preventing breaches. By employing behavioral analytics and anomaly detection, IAM systems can identify abnormal patterns of access, such as a user attempting to access resources outside their normal scope of work. These insights allow security teams to respond quickly to potential threats and mitigate damage before it escalates.

As organizations adopt more complex IAM strategies, the role of artificial intelligence (AI) and machine learning (ML) is becoming increasingly important. AI and ML can analyze large volumes of access data to detect patterns, predict potential risks, and automate routine tasks. These technologies can enhance IAM systems by identifying potential vulnerabilities and suggesting ways to strengthen access controls.

Identity and Access Management is a fundamental aspect of modern cybersecurity. It provides the foundation for secure access to systems, applications, and data by verifying identities and managing permissions. As organizations continue to embrace digital transformation, effective IAM practices will be essential for protecting sensitive information, ensuring compliance, and mitigating risks. By implementing robust IAM systems, businesses can achieve a secure and efficient framework for managing access across their entire IT environment.

**Assignment.5.2:**

1. Why are data breaches considered one of the most significant threats in cloud computing, and how can organizations mitigate this risk?
2. What are Denial of Service (DoS) attacks, and what strategies can organizations use to protect against them in cloud environments?
3. How can insider threats compromise cloud security, and what measures can organizations implement to reduce these risks?
4. What role does API security play in cloud environments, and how can weak APIs expose systems to threats?

**5.4 Incident Response in the Cloud**

Incident response in the cloud is a critical aspect of maintaining the security and integrity of cloud-based systems and data. When an incident occurs—such as a data breach, service disruption, or cyberattack—it is essential to have a well-defined and effective response plan in place. In cloud environments, the complexities of shared responsibility models, dynamic resource allocation, and distributed architectures require tailored incident response strategies. Unlike traditional on-premise systems, the cloud offers flexibility and scalability but also presents unique challenges in terms of detection, containment, and recovery.

The first step in cloud incident response is preparation. This involves understanding the cloud service model—whether it is Infrastructure as a Service (IaaS), Platform as a Service (PaaS), or Software as a Service (SaaS)—and the specific security responsibilities of both the cloud provider and the customer. Customers need to work closely with their cloud service provider to ensure that their incident response plans align with the shared responsibility model. This includes defining which security controls are in place, who handles specific types of incidents, and how response actions are coordinated.

One of the main challenges in cloud-based incident response is the lack of visibility. Since cloud environments are hosted and managed by third-party providers, organizations may not have direct access to certain data or logs. To address this, cloud customers must ensure that the necessary monitoring and logging capabilities are configured from the outset. Tools like AWS CloudTrail or Azure Security Center provide visibility into the activities within the cloud environment, enabling organizations to detect suspicious activity in real-time and respond accordingly.

When an incident is detected, the next step is to contain the issue. In cloud environments, this can be particularly challenging due to the fluid nature of cloud resources. Instances can be spun up and shut down at any time, making it difficult to isolate compromised systems. However, with the right cloud security posture—such as the use of virtual private clouds (VPCs), security groups, and network segmentation—it is possible to isolate affected systems and limit the spread of the incident. Automation also plays a critical role here, with predefined security rules and automated workflows helping to contain and mitigate threats quickly.

Communication during an incident is vital to ensure that all stakeholders are aware of the issue and are involved in the resolution process. In a cloud environment, this communication must include both internal teams and the cloud provider's support teams. Establishing clear communication channels and escalation procedures is essential for timely decision-making. It is also important to have a plan in place for informing customers, regulatory bodies, and other external parties as necessary, especially if the incident involves a breach of sensitive data.

After containment, the next step is eradication. This involves eliminating the root cause of the incident, which may require identifying vulnerabilities or misconfigurations that allowed

the attack or breach to occur. In cloud environments, this can involve working with the cloud service provider to address any underlying infrastructure vulnerabilities. For example, patching a vulnerable system image, closing unused ports, or tightening access controls may be necessary steps in eradicating the threat.

Recovery is the next phase, where services and systems are restored to normal operations. This process involves restoring data from backups, rebuilding systems, and ensuring that the environment is free from any residual threats. In the cloud, the recovery process can be streamlined due to the availability of tools such as automated backups, Disaster Recovery as a Service (DRaaS), and high-availability configurations. These services can reduce downtime and expedite the restoration of critical services.

Once the system is restored, a post-incident review is essential. This review aims to assess the effectiveness of the response and identify areas for improvement. The review should involve a thorough analysis of the incident, including how it was detected, contained, and resolved. In cloud environments, it is essential to evaluate whether the existing cloud security policies and controls were sufficient and to identify any gaps in the response process. This step also includes revisiting lessons learned and updating incident response plans to improve future response capabilities.

In addition to these steps, incident response in the cloud requires continuous monitoring and improvement. Cloud environments are constantly evolving, with new services, configurations, and integrations being added regularly. This dynamic nature means that incident response plans and strategies must be adaptable to new threats and challenges. Regular updates to monitoring tools, response workflows, and security policies are necessary to stay ahead of emerging risks.

For effective incident response in the cloud, organizations must integrate security into their development lifecycle. DevSecOps is an approach that involves incorporating security practices into the DevOps workflow, ensuring that security is prioritized from the start of software development to deployment. By adopting a DevSecOps mindset, organizations can ensure that security vulnerabilities are detected and addressed early, reducing the likelihood of incidents in the cloud.

One key aspect of incident response in the cloud is threat intelligence. Cloud providers typically offer threat intelligence feeds and integrate security services that can identify known vulnerabilities or attacks in real-time. Customers can use these feeds to stay informed about emerging threats and vulnerabilities in the cloud environment. By leveraging threat intelligence, organizations can enhance their ability to proactively defend against attacks before they occur.

Security automation plays a significant role in cloud incident response. Cloud security automation tools can detect threats, apply remediation actions, and trigger workflows without manual intervention. This reduces response times and minimizes human error. Automation can include actions such as shutting down an instance, applying patches, or blocking IP addresses associated with malicious activities. For example, AWS Lambda can be used to trigger automated responses based on security events, such as a DDoS attack or unauthorized access attempt.

Another important consideration is compliance and legal requirements. In the event of a security incident, organizations must ensure compliance with data protection regulations such as the General Data Protection Regulation (GDPR) or the Health Insurance Portability and Accountability Act (HIPAA). In cloud environments, compliance can be particularly challenging, as data may reside in multiple jurisdictions. Incident response plans should include procedures for reporting incidents in a manner that meets regulatory requirements, including timelines for notification and the type of information that must be provided.

Collaboration between the cloud service provider and the customer is crucial during an incident. Many cloud providers offer dedicated support teams and incident response resources to help customers manage security incidents. It is important to establish a relationship with the provider's support team in advance and ensure that communication protocols are in place. Cloud providers may also offer additional security services or expertise to help mitigate and resolve incidents more effectively.

Cloud-native tools and services can significantly aid in incident response. For instance, Amazon GuardDuty, Google Cloud Security Command Center, and Microsoft Azure Sentinel provide advanced threat detection capabilities. These services can automatically analyze logs, network traffic, and other data sources to identify potential security incidents. By utilizing these native tools, organizations can enhance their detection and response capabilities and reduce the time required to respond to threats.

In the cloud, business continuity planning is deeply intertwined with incident response. A key component of this planning is ensuring that backup and disaster recovery solutions are in place to restore critical data and systems. Cloud environments offer a variety of backup solutions—such as snapshot-based backups and geographically redundant data storage—that can help organizations recover from incidents quickly. Ensuring that backup systems are secure and tested regularly is essential to prevent data loss during incidents.

Ongoing training and awareness are vital for maintaining an effective cloud incident response strategy. Staff at all levels should be trained to recognize security threats and understand their roles in responding to incidents. This includes regular drills and simulations to practice responses to cloud-specific incidents, such as data breaches or service outages. Ensuring that employees are aware of best practices for securing cloud environments can help reduce the likelihood of incidents occurring in the first place.

In summary, incident response in the cloud requires careful planning, collaboration, and continuous improvement. Given the dynamic nature of cloud environments, organizations must be proactive in developing response plans that are flexible, scalable, and integrated with cloud-native tools. By focusing on preparation, detection, containment, eradication, and recovery, businesses can respond effectively to incidents in the cloud and minimize the impact on their operations and customers.

## 5.5 Security Best Practices

Security best practices are essential for ensuring the integrity, confidentiality, and availability of information in an increasingly complex and interconnected digital landscape. As cyber threats become more sophisticated, organizations must adopt comprehensive security measures to protect their assets, applications, and data. One of the most fundamental best practices is implementing a robust access control framework. By ensuring that only authorized users can access specific systems and data, businesses can prevent unauthorized access and minimize the potential for data breaches. Access controls should be based on the principle of least privilege, ensuring that users only have the permissions necessary for their job roles.

Encryption is another crucial security best practice that organizations must implement. Encrypting sensitive data both in transit and at rest ensures that even if unauthorized parties gain access to the data, it remains unreadable. For instance, using protocols such as TLS for encrypting data in transit and AES for encrypting data at rest can safeguard valuable information. Additionally, organizations should ensure that encryption keys are managed securely to prevent unauthorized decryption of the data.

Multi-factor authentication (MFA) should be used to add an extra layer of security to authentication processes. MFA requires users to provide multiple forms of identification before gaining access to sensitive systems. This can include something the user knows (like a password), something the user has (like a phone or hardware token), or something the user is (such as biometric data). MFA reduces the likelihood of unauthorized access, even if a password is compromised.

Regular patching and software updates are essential for maintaining the security of systems. Many cyberattacks exploit known vulnerabilities in software, and timely patching can help mitigate these risks. Organizations should implement a patch management process that ensures all systems, including operating systems, applications, and third-party software, are kept up to date with the latest security fixes. Automating the patching process can reduce human error and ensure that critical vulnerabilities are addressed in a timely manner.

Network security is another critical area where best practices must be followed. Firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS) can be used to monitor and protect network traffic. By segmenting networks into different zones and

applying appropriate security controls to each, organizations can limit the impact of an attack. Network security also involves using virtual private networks (VPNs) to secure remote connections and prevent unauthorized access to the corporate network.

Regular data backups are an essential practice for safeguarding information against data loss caused by cyberattacks, hardware failures, or natural disasters. Backups should be performed regularly and stored securely, either in an offsite location or in the cloud. It's important to test backup processes periodically to ensure that data can be restored quickly and accurately in the event of an incident. Encrypting backup data is also important to prevent exposure in case of a breach.

One of the most overlooked aspects of security is employee training and awareness. Employees are often the first line of defense against cyberattacks, so educating them on the latest security threats and best practices is crucial. Training should cover topics such as phishing, password management, social engineering, and how to report suspicious activities. Regularly conducting security awareness programs and phishing simulations can help employees recognize and respond to threats before they cause harm.

Data classification and labeling are also important best practices. By categorizing data based on its sensitivity, organizations can apply appropriate security controls to protect it. For example, highly sensitive data, such as financial or personal health information, may require stricter access controls and encryption, while less sensitive data may be subject to lighter security measures. Classifying and labeling data also aids in compliance with data protection regulations, such as GDPR or HIPAA, which mandate specific security controls for different types of data.

To protect against insider threats, organizations should implement activity monitoring and auditing practices. By regularly reviewing logs and activity reports, security teams can identify suspicious behavior and intervene before significant damage occurs. Implementing a centralized logging system that aggregates logs from various systems and services can help create a comprehensive view of security events and make it easier to detect anomalies.

Securing endpoints is another essential aspect of security. Endpoints, such as laptops, mobile devices, and workstations, are often targeted by attackers as entry points into a network. Organizations should ensure that endpoints are protected with antivirus software, firewalls, and endpoint detection and response (EDR) solutions. Mobile device management (MDM) can also be used to secure mobile devices, ensuring that they comply with organizational security policies and that sensitive data is protected even if the device is lost or stolen.

Identity and access management (IAM) systems are crucial for enforcing security best practices related to user authentication and authorization. IAM solutions help organizations manage user identities, roles, and permissions across various systems and applications. By using IAM tools, organizations can enforce policies such as password complexity requirements, time-based access controls, and account lockout mechanisms to prevent unauthorized access and reduce the likelihood of credential theft.

Cloud security is a growing area of concern for organizations, as more businesses adopt cloud services for their operations. Following security best practices in the cloud involves understanding the shared responsibility model, where the cloud provider is responsible for securing the infrastructure, while the customer is responsible for securing their data and applications. Organizations should use encryption, access controls, and network security measures to protect their cloud environments. Regular audits of cloud configurations can also help identify vulnerabilities or misconfigurations that could expose the organization to risk.

Incident response planning is an essential part of any security strategy. Having a clear and well-documented incident response plan in place allows organizations to respond quickly and effectively to security incidents, minimizing potential damage. The plan should include procedures for detecting, containing, eradicating, and recovering from incidents. Organizations should also conduct regular tabletop exercises and simulations to test their response capabilities and ensure that their teams are prepared to handle various types of security incidents.

For organizations to comply with regulatory frameworks, they must implement security controls that meet the specific requirements set forth by industry regulations. Regulatory requirements such as the General Data Protection Regulation (GDPR), the Health Insurance Portability and Accountability Act (HIPAA), and the Payment Card Industry Data Security Standard (PCI DSS) mandate specific security practices to protect sensitive data. Compliance with these regulations not only helps organizations avoid legal and financial penalties but also demonstrates a commitment to data protection.

Securing supply chains is an increasingly important best practice. As organizations rely on third-party vendors for various services and products, it's essential to ensure that these vendors follow appropriate security practices. Organizations should assess the security posture of their vendors before entering into contracts and require that they meet specific security standards. Conducting regular security assessments and audits of third-party vendors can help identify potential risks in the supply chain.

Security testing and vulnerability assessments are essential for identifying weaknesses in systems and applications. Regularly conducting penetration testing, vulnerability scanning, and risk assessments can help uncover vulnerabilities before they are exploited by attackers. Organizations should also participate in bug bounty programs, where external security researchers are incentivized to identify and report vulnerabilities in the system.

Organizations must adopt a zero-trust security model, where trust is never assumed, and every request for access is verified, regardless of where it originates. In a zero-trust environment, security is enforced at every layer of the network, and access to resources is granted based on strict authentication and authorization policies. This model helps mitigate the risks associated with insider threats and external attacks, ensuring that systems and data are always protected.

Security best practices as shown in Figure. 5.1 are a vital component of an organization's cybersecurity strategy. By implementing access controls, encryption, MFA, patch management, and other essential practices, businesses can reduce their exposure to cyber threats and protect their data and systems. A proactive approach to security, combined with ongoing training, monitoring, and incident response planning, ensures that organizations can respond effectively to evolving threats and maintain a secure digital environment.

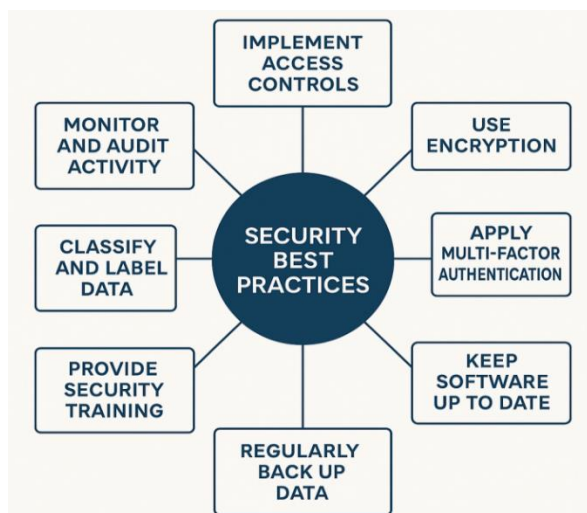


Fig 5.1. Security Best Practices

### Assignment.5.3:

1. What is the significance of the shared responsibility model in preparing for incident response in the cloud?
2. Why is visibility a challenge in cloud-based incident response, and how can organizations address it?
3. Explain how automation contributes to incident containment in cloud environments.

4. Why is communication important during a cloud security incident, and who should be involved?
5. How does the use of cloud-native tools enhance incident detection and response?

**SUMMARY**

- Data protection is critical in the cloud, with challenges in maintaining confidentiality, integrity, and compliance across shared and distributed environments.
- Effective identity and access management (IAM) is essential to prevent unauthorized access and enforce least privilege principles in multi-cloud setups.
- Organizations must adapt to evolving threats and misconfigurations by implementing continuous monitoring, DevSecOps, and automated security controls.
- Understanding the shared responsibility model and ensuring clear accountability between cloud providers and customers is vital for maintaining a secure cloud ecosystem.
- Data breaches remain the most critical cloud threat, often caused by poor encryption or access control.
- Denial of Service (DoS) and Distributed DoS attacks can cripple cloud services and must be countered with scalable protection.
- Data loss due to failures or deletions necessitates both provider and organizational backup strategies.
- Insider threats exploit legitimate access, making IAM policies and user monitoring essential.
- Unsecured APIs expose cloud environments to attacks like SQL injection and require rigorous coding and authentication practices.
- Malware and ransomware targeting cloud systems can be mitigated with endpoint protection and regular backups.
- Cloud misconfigurations, such as public data exposure, are preventable through automated configuration management tools.
- Lack of monitoring and logging weakens breach detection and response, requiring continuous activity tracking.
- IAM ensures secure user access by verifying identities and enforcing role-based or attribute-based access control.
- Modern IAM systems, integrated with AI, MFA, and SIEM, form the backbone of cloud security and compliance efforts.
- Preparation is crucial, requiring alignment with the shared responsibility model and understanding the chosen cloud service model (IaaS, PaaS, SaaS).
- Detection and visibility challenges demand robust monitoring tools and logging configurations for real-time threat identification.
- Containment and eradication involve isolating compromised resources using cloud-native tools and automating mitigation steps.
- Recovery and post-incident review focus on restoring operations, analyzing the root cause, and improving response plans.
- Continuous improvement through DevSecOps, automation, and compliance ensures proactive defence and effective incident handling in dynamic cloud environments.

**Check Your Progress****A. Multiple choice Questions**

1. What is a major concern regarding data confidentiality in cloud environments?
  - a) Limited data storage capacity
  - b) Shared resources and multi-tenant architectures
  - c) Expensive service subscriptions
  - d) Limited software compatibility
2. Why is data integrity especially critical in industries like healthcare and finance?
  - a) They use large-scale databases
  - b) They require high-speed processing

- c) They are subject to strict regulatory compliance
  - d) They deal with international client
3. What is the purpose of Identity and Access Management (IAM) in cloud security?
    - a) To monitor physical access to data centers
    - b) To backup cloud resources
    - c) To control who has access to which resources
    - d) To manage DNS configurations
  4. What does the shared responsibility model in cloud security imply?
    - a) Both provider and customer are jointly liable for service downtime
    - b) Cloud providers manage all security responsibilities
    - c) Customers handle infrastructure security alone
    - d) Providers and customers share distinct security responsibilities
  5. Which challenge is caused by limited visibility into the cloud infrastructure?
    - a) Reduced scalability
    - b) Inability to automate billing
    - c) Difficulty in threat detection
    - d) Slow data transmission
  6. What is a common security risk associated with APIs in cloud environments?
    - a) Slow performance
    - b) Poor integration with firewalls
    - c) Unauthorized access due to poor security
    - d) High bandwidth usage
  7. Why are misconfigurations a major cause of cloud security incidents?
    - a) They reduce storage space
    - b) They slow down service delivery
    - c) They expose resources to unauthorized access
    - d) They improve network visibility
  8. Which of the following attacks involves flooding a cloud service with traffic to render it unavailable?
    - a) Man-in-the-Middle (MITM)
    - b) Denial of Service (DoS)
    - c) Privilege Escalation
    - d) Cross-Site Scripting (XSS)
  9. What is a common challenge faced during incident response in cloud environments?
    - a) Lack of on-premises servers
    - b) Overprovisioning of physical hardware
    - c) Lack of visibility into logs and data
    - d) Inflexibility in scaling resources
  10. Which cloud-native tool is used in AWS for detecting suspicious activity and aiding in incident response?
    - a) Azure Security Center
    - b) Google Cloud Security Command Center
    - c) AWS CloudTrail
    - d) Oracle Cloud Guard

**B. Fill in the Blanks:**

1. Cloud environments often involve shared resources and \_\_\_\_\_ architectures, increasing the complexity of maintaining data confidentiality.
2. Ensuring \_\_\_\_\_ of data is critical for industries like healthcare, finance, and government, where compliance and accuracy are essential.

3. IAM stands for \_\_\_\_\_ and is essential for controlling user access in cloud environments.
4. To ensure business continuity during cloud outages, organizations must implement effective \_\_\_\_\_ recovery strategies.
5. The use of \_\_\_\_\_ such as VPCs and security groups helps isolate compromised systems during a cloud incident.
6. The \_\_\_\_\_ phase of incident response involves eliminating the root cause of the incident, such as vulnerabilities or misconfigurations.
7. DevSecOps integrates security into the \_\_\_\_\_ lifecycle to detect vulnerabilities early.
8. Multi-factor authentication (MFA) enhances security by requiring users to provide \_\_\_\_\_ forms of identification.
9. The principle of \_\_\_\_\_ ensures that users only have the access necessary for their job roles.
10. A \_\_\_\_\_ security model assumes no implicit trust and verifies every access request.

### C. True or False

1. Data breaches in the cloud are primarily caused by hardware failures and not by unauthorized access.
2. Denial of Service (DoS) attacks can impact both individual cloud applications and the entire infrastructure of a cloud service provider.
3. Cloud providers alone are responsible for preventing data loss in the cloud.
4. Secure APIs are not essential for cloud security as long as encryption is used.
5. Cloud-based social engineering attacks often involve manipulating users into giving up credentials.
6. Security awareness training is unnecessary for an effective IAM strategy, as technical controls alone are sufficient.
7. Cloud incident response strategies must be tailored to the shared responsibility model between the cloud provider and the customer.
8. The lack of visibility in cloud environments does not impact incident response efforts.
9. Automated security tools in the cloud can help mitigate threats and contain incidents quickly.
10. The zero-trust security model ensures that no access is granted without verification, regardless of the request origin.

### D. Short Answer Questions

1. What is the shared responsibility model in cloud security, and why can it lead to security gaps?
2. How do misconfigurations pose a risk to cloud environments?
3. Why is identity and access management (IAM) crucial in securing cloud systems?
4. Name two common threats that can disrupt cloud services.
5. How do insider threats differ from external attacks in the context of cloud security?
6. What is the purpose of using data encryption in cloud computing?
7. Why is the Secure Software Development Lifecycle (SSDLC) important?
8. What is the principle of least privilege in Identity and Access Management (IAM)?
9. How does multi-factor authentication enhance cloud security?
10. What is the role of regular security audits in a cloud environment?

**Answer Key****A. Multiple Choice Questions (MCQ)**

1. (b) 2. (c) 3. (c) 4. (d) 5. (c) 6. (c) 7. (c) 8. (b) 9. (c) 10. (c)

**B. Fill in the blanks**

1. multi-tenant 2. integrity 3. Identity and Access Management 4. disaster 5. network segmentation 6. eradication 7. development 8. multiple 9. least privilege 10. zero-trust

**C. True or False**

1. False 2. True 3. False 4. False 5. True 6. False 7. True 8. False 9. True 10. True

विद्यया ऽ मृतमश्नुते

एन सी ई आर टी  
NCERT**PSS CENTRAL INSTITUTE OF VOCATIONAL EDUCATION**

(a constituent unit of NCERT, under Ministry of Education, Government of India)  
Shyamla Hills, Bhopal- 462 002, M.P., INDIA

[www.psscive.ac.in](http://www.psscive.ac.in)