,

BAYELSA STATE POLYTECHNIC

SCHOOL OF APPLIED SCIENCES

Agent
Agent
Agent
Agent
Agent
Agent
Server Software
Video Data store
Administrator panel

Network Operating Systems

DEPARTMENET OF COMPUTER SCIENCE
(NETWORKING AND CLOUD COMPUTING)
PRACTICAL MANUAL / WORKBOOK

ON

OPERATING SYSTEM FOR NETWORKING ENVIRONMENT
COURSE CODE: NCC 313

Name:	
Matric No.	
Department:	
Course:	
Group/Session:	

Introduction Guide for Computer Practical Exercises

Welcome to your computer-based practical sessions! These exercises are designed to help you apply theoretical knowledge and develop hands-on skills that are crucial for your academic and professional success. To ensure that you get the most out of these sessions, it’s important to approach each task with focus, preparation, and attention to detail. This guide will walk you through essential steps to help you handle the exercises in your practical manuals effectively.

Why is this important

Computer practical exercises often involve complex software tools, programming environments. The objective is not just to complete the task but to learn and understand the process. By following the right approach, you can maximize your learning, minimize mistakes, and ensure that you can solve problems efficiently and independently.

Steps to Handle Your Computer Practical Exercises Effectively

1. Preparation is Key

Before your practical session, make sure your computer is set up and ready to go. Ensure that all necessary software is installed and updated, and all peripherals are functioning properly. Taking the time to set up your workstation will save your time during the session and allow you to focus on the tasks.

2. Familiarize Yourself with the Manual

Read through the practical manual and the instructions for the exercise thoroughly before you begin. Make sure you understand the objectives, the steps involved, and the tools or software you’ll need. If any part of the manual is unclear, take note of your questions and ask your instructor for clarification before starting.

3. Follow the Instructions Step-by-Step

Each practical exercise is structured to guide you through a sequence of steps. Pay close attention to these instructions and follow them in order. Skipping steps can result in errors or incomplete work, so it’s crucial to proceed in the correct sequence to achieve the desired outcome.

4. Ask for Help When Needed

Don’t hesitate to ask your instructor or classmates for assistance if you encounter difficulties. It’s okay not to know everything, practical exercises are learning opportunities. Whether it’s a technical issue or confusion about the task, getting help when needed will keep you on track.

5. Review Your Work

Once you’ve completed the practical exercise, take time to review your work. If applicable, test the program or functionality you’ve worked on. Debug any errors, recheck your results, and ensure everything meets the given specifications.

6. Ensure Timely Submission

Follow any instructions for submitting your work carefully. Save your files in the correct format and back them up if needed. Timely submission of your completed work ensures you meet the requirements and deadlines set by your instructor.

Final Thoughts

Your practical sessions are an opportunity to sharpen your skills, build confidence, and gain a deeper understanding of computer applications and tools. By preparing ahead, staying organized, and following the instructions carefully, you will ensure that you handle each exercise effectively and make the most out of your practical experience.

Good luck, and remember that consistent practice and attention to detail will lead to success!

Introduction

Operating systems in networking environments serve as the critical foundation that enables devices to communicate, share resources, and work together seamlessly. From basic file sharing in home networks to complex distributed systems spanning the globe, the networking capabilities of operating systems make modern computing possible.

As technology continues to evolve, operating systems will adapt to new networking paradigms, security challenges, and user requirements. Understanding the fundamental concepts of how operating systems function in networked environments provides essential knowledge for anyone working with computers in today's interconnected world.

An operating system (OS) forms the critical foundation between computer hardware and the applications that users interact with daily. When we place computers in a networking environment, the role of the operating system expands dramatically - it must not only manage local resources but also facilitate communication and resource sharing across multiple devices.

In today's interconnected world, understanding how operating systems function in networking environments is essential knowledge for IT professionals and even everyday computer users.

What is an Operating System

An operating system is software that manages computer hardware and provides services for computer programs. It acts as an intermediary between users and the computer hardware, making the hardware usable and providing a platform for applications to run.

An Operating System (OS) is system software that manages hardware and software resources, providing a user-friendly interface for interaction with a computer. In a networking environment, the role of the OS expands to include managing network communication, resource sharing, security, and distributed computing.

This lecture explores the concept of an OS in a networking environment, discussing its functions, types, and key features that enable seamless network operations.

The core functions of any operating system include:

1. Process management: Creating, scheduling, and terminating processes
2. Memory management: Allocating and deallocating memory to programs
3. File system management: Organizing and controlling access to files
4. Device management: Controlling input/output operations to hardware
5. User interface: Providing ways for users to interact with the computer

Types of Operating Systems

Several types of operating systems exist today:

1. Single-user systems (Windows, macOS): Designed for personal computers
2. Multi-user systems (Linux, UNIX): Allow multiple users to work simultaneously
3. Network operating systems: Specifically designed to support networked environments
4. Distributed operating systems: Manage resources across multiple physical computers
5. Mobile operating systems (Android, iOS): Optimized for mobile devices
6. Real-time operating systems: Guarantee task completion within strict time constraints

What is a Computer Network?

A computer network is a collection of interconnected computing devices that can exchange data and share resources. Networks range from small home setups to massive global infrastructures like the internet.

Networks provide several key benefits:

- Resource sharing (files, printers, internet connections)
- Communication and collaboration
- Centralized data storage and backup
- Software and hardware cost reduction through sharing

Network Architecture Models

The two primary network architecture models are:

- Client-Server Model: Dedicated servers provide resources to client computers
- Peer-to-Peer Model: Each computer acts as both client and server, sharing resources directly

These models influence how operating systems are designed to interact with the network.

Operating Systems in Networking Environments

A Network Operating System is specialized software that allows a computer to communicate with other computers and devices on a network. A network operating system (NOS) is an OS designed to support workstations, personal computers, and servers connected over a network. It facilitates:

Examples include:

- Windows Server
- Linux-based servers (Ubuntu Server, Red Hat Enterprise Linux)
- Novell NetWare (legacy)
- macOS Server
- Cisco IOS (for network devices)

Characteristics of a Network Operating System include:

- Multiple user support: Concurrent users accessing resources
- Security features: Authentication, authorization, and encryption
- Server management tools: Remote administration capabilities
- Directory services: Centralized resource management (like Active Directory)
- File and print services: Sharing storage and output devices
- Network protocols support: TCP/IP, HTTP, FTP, etc.

Networking Capabilities in Modern Operating Systems

Even standard desktop operating systems now include robust networking capabilities:

- Network protocol implementation: Built-in support for TCP/IP and other protocols
- Network configuration tools: Tools to set up and troubleshoot network connections
- File sharing services: Ability to share local resources with other network users
- Remote access features: SSH, remote desktop, VPN support
- Firewall protection: Built-in security to control network traffic

Core Networking Components in Operating Systems

Network Protocol Stack

The network protocol stack is a set of software layers that work together to enable network communication. The most common implementation is the TCP/IP protocol suite, which follows these layers:

1. Application Layer: User programs (web browsers, email clients)
2. Transport Layer: End-to-end communication (TCP, UDP)
3. Internet Layer: Addressing and routing (IP)
4. Network Interface Layer: Physical connection to the network

Operating systems implement this stack to handle all network communications.

1. Network Drivers

Network interface cards (NICs) require special software called drivers to function. The operating system:

- Loads appropriate drivers for hardware
- Manages data flow between applications and network hardware
- Handles interrupt requests from network cards
- Buffers incoming and outgoing data

2. Network Services

Modern operating systems provide numerous network services, including:

- DHCP client: Obtaining network configuration automatically
- DNS resolver: Converting domain names to IP addresses
- Network file systems: Accessing remote files as if they were local
- Network printing: Sending print jobs to remote printers
- Service discovery: Finding available services on the network

OS Functions in Networked Environments

1 Resource Sharing

One of the primary purposes of networking is resource sharing, which the OS facilitates through:

- File sharing: Making local files available to remote users
- Printer sharing: Allowing multiple users to use a single printer
- Application sharing: Providing access to centrally installed software
- Internet connection sharing: Enabling multiple devices to use one connection

2 Communication Management

Operating systems manage communication between networked devices by:

- Establishing and maintaining connections
- Packaging data for transmission
- Routing data to correct applications
- Handling transmission errors
- Managing connection quality

3 Security Implementation

Security is critical in networked environments. Operating systems provide:

- User authentication: Verifying user identities

- Access control: Determining who can access which resources
- Encryption: Protecting data during transmission
- Firewall functionality: Filtering network traffic
- Malware protection: Scanning for threats from the network

4 Network Administration

Operating systems offer tools for network administration, including:

- Command-line utilities for network troubleshooting
- GUI-based network management tools
- Remote administration capabilities
- Performance monitoring for network connections
- Logging of network activities

Practical 1

Practical 1: Follow the step below to Create a new user account named **"NetStudent"** with the password Hint@123, and ensure the user is set as a **Standard User**.

Steps:

1. Open Control Panel > User Accounts > Manage another account.
2. Click Add a new user in PC settings.
3. Select Add someone else to this PC.
4. Click I don't have this person's sign-in information, then select Add a user without a Microsoft account.
5. Enter:
   - Username: NetStudent
   - Password: Hint@123
6. Click Next.
7. Under Family & other users, click on the user account and select **Change account type.**
8. Ensure it is set to Standard User.

Question 2: Follow the same steps and explain how to change the account type of **"NetStudent"** to **Administrator**, and then delete the account while choosing to **retain the user's files**.

…………………………………………………………………………………………
…………………………………………………………………………………………
…………………………………………………………………………………………
…………………………………………………………………………………………
……………………………………………………………

Practical 2

Create, Delete, Copy, and Move Files/Directories Using Windows OS

Question 1: Create a folder called "IT_Project" on the Desktop. Inside it, create two files named tasklist.txt and projectplan.docx. Then, copy the entire folder to the Documents directory.

Follow this Steps:

1. Right-click on the Desktop > New > Folder > Name it IT_Project.
2. Open IT_Project and create two files:
   - Right-click > New > Text Document > Name it tasklist.txt.
   - Open Microsoft Word > Save a blank document as projectplan.docx inside the folder.
3. Right-click on IT_Project > **Copy**.
4. Navigate to Documents > **Paste**.

Question 2:

Explain the step to Move all contents of the Downloads folder into a new folder named Backup2022 under Documents, then delete the original files from Downloads.

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

…………………………………………………………………………………………....

Practical 3

Use Task Manager / Activity Monitor to Monitor System Resources

Question 1:
Press Ctrl + Shift + Esc to open Task Manager and open Task Manager and identify which application is using the most CPU resources. End the process and observe how system performance changes.

………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………

Question 2:

Use the Performance tab in Task Manager to monitor Memory and Disk usage. Take a screenshot before and after opening three programs: Chrome, Word, and Excel.

………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………..

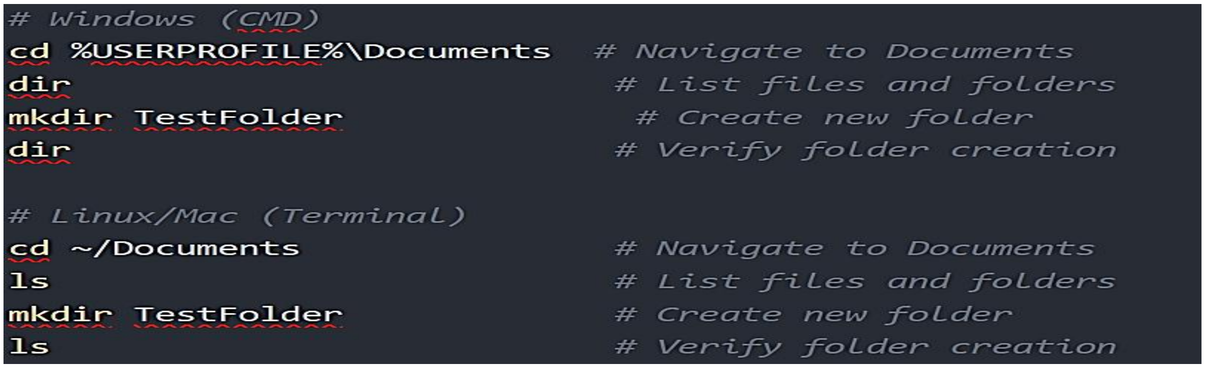
Practical 4

Question 1: Navigating Directories & Listing Files

Task:

1. Open the **Command Prompt (Windows)** or **Terminal (Linux/Mac)**.
2. Navigate to the Documents directory.
3. List all files and subdirectories inside it.
4. Create a new directory called **"TestFolder"** inside **Documents**.
5. List the contents again to confirm the new folder exists.

Follow this:



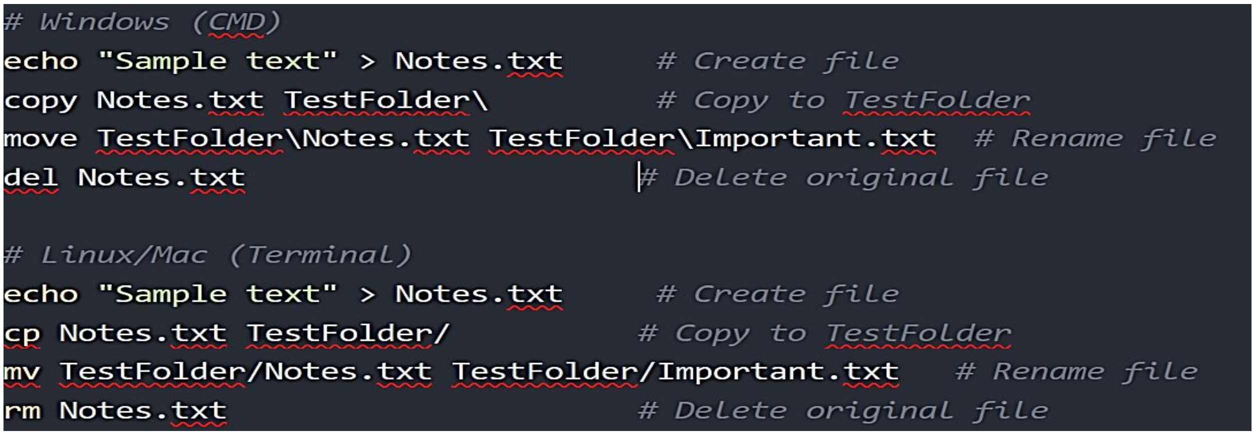
```
# Windows (CMD)
cd %USERPROFILE%\Documents  # Navigate to Documents
dir                          # List files and folders
mkdir TestFolder              # Create new folder
dir                          # Verify folder creation

# Linux/Mac (Terminal)
cd ~/Documents               # Navigate to Documents
ls                           # List files and folders
mkdir TestFolder             # Create new folder
ls                           # Verify folder creation
```

Question 2: Copying, Moving, and Deleting Files

Follow this Task:

1. Create a text file named **"Notes.txt"** in the **Documents** directory.
2. Copy this file to the **"TestFolder"** directory.
3. Rename (move) the copied file to **"Important.txt"** inside **TestFolder**.
4. Delete the original **"Notes.txt"** from **Documents**.

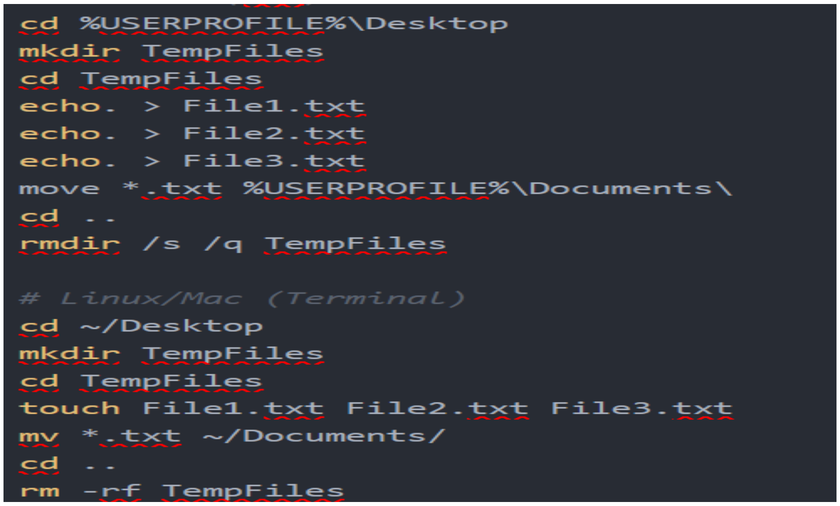


```
# Windows (CMD)
echo "Sample text" > Notes.txt     # Create file
copy Notes.txt TestFolder\         # Copy to TestFolder
move TestFolder\Notes.txt TestFolder\Important.txt # Rename file
del Notes.txt                     # Delete original file

# Linux/Mac (Terminal)
echo "Sample text" > Notes.txt     # Create file
cp Notes.txt TestFolder/          # Copy to TestFolder
mv TestFolder/Notes.txt TestFolder/Important.txt  # Rename file
rm Notes.txt                      # Delete original file
```

Task:

1. Navigate to the **Desktop** directory.
2. Create a folder called **"TempFiles"**.
3. Inside it, create **3 empty text files**: **File1.txt, File2.txt, File3.txt**.
4. Move all files to the **Documents** directory.
5. Delete the **"TempFiles"** folder.



```
cd %USERPROFILE%\Desktop
mkdir TempFiles
cd TempFiles
echo. > File1.txt
echo. > File2.txt
echo. > File3.txt
move *.txt %USERPROFILE%\Documents\
cd ..
rmdir /s /q TempFiles

# Linux/Mac (Terminal)
cd ~/Desktop
mkdir TempFiles
cd TempFiles
touch File1.txt File2.txt File3.txt
mv *.txt ~/Documents/
cd ..
rm -rf TempFiles
```

Practical 5

Package Management in Linux

Question 1: How would you install a package and ensure it's updated to the latest version in Linux?

Follow This task:

1. First update the package repositories by entering this in the terminal:
   ```
   sudo apt update
   ```
2. Then install the package, enter (example: nginx)
   ```
   sudo apt install nginx
   ```
3. To ensure the package is the latest version, enter
   ```
   sudo apt upgrade nginx
   ```

Networking Setup

Question 1: How would you configure a static IP address using the ip command?

Follow This steps:

1. Temporarily assign a static IP
   ```
   sudo ip addr add 192.168.1.100/24 dev eth0
   ```
2. Add default gateway
   ```
   sudo ip route add default via 192.168.1.1
   ```
3. For persistence, edit configuration file:
   ```
   # /etc/netplan/01-netcfg.yaml
   ```

```
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      addresses: [192.168.1.100/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

Then apply: sudo netplan apply

Question 2: How to test connectivity and diagnose network issues?

Command	Purpose
ip a	🖥️ View network interfaces
ping 192.168.1.1	🏠 Test local network
ping 8.8.8.8	🌍 Test internet connectivity
ping google.com	🔤 Test DNS resolution
traceroute google.com	📨 Trace network path
ss -tuln	👂 Check listening services
ip route	🗺️ View routing table
cat /etc/resolv.conf	📝 Check DNS configuration

\

Question 3: Three ways to find help on the find command?

Answer:

Method	Command	What It Provides
📚 Man pages	man find	Comprehensive documentation with all options and examples
📖 Info pages	info find	Detailed, structured, hyperlinked documentation
⚡ Built-in help	find --help	Concise summary of common options

Practical 6

Linux Administration:
User Authentication and Permissions

Question 1: How to create a new user with sudo privileges and set password expiration policies?

Follow this step:

1. Create new user by entering this in the terminal
   ```
   sudo useradd -m -s /bin/bash jsmith
   ```
2. Set initial password
   ```
   sudo passwd jsmith
   ```
3. Add user to sudo group
   ```
   sudo usermod -aG sudo jsmith
   ```
4. Set password to expire in 90 days
   ```
   sudo chage -M 90 jsmith
   ```
5. Verify password expiration settings
   ```
   sudo chage -l jsmith
   ```

The -m flag creates a home directory, -s sets the shell, and -aG adds the user to a supplementary group. The change command manages password expiration policies.

Question 2: How to modify permissions on a directory to allow a group of users to read and write files, but prevent other users from accessing the content?

Follow this step:

1. Create a group for users who need access
   ```
   sudo groupadd project_team
   ```
2. Add users to the group
   ```
   sudo usermod -aG project_team user1
   sudo usermod -aG project_team user2
   ```
3. Create project directory
   ```
   sudo mkdir /projects/alpha
   ```
4. Set ownership to appropriate user and group
   ```
   sudo chown manager:project_team /projects/alpha
   ```

5. Set permissions (owner: rwx, group: rwx, others: none)
   ```
   sudo chmod 770 /projects/alpha
   ```
6. Set group ID bit so new files inherit the group
   ```
   sudo chmod g+s /projects/alpha
   ```

The chmod 770 command sets read, write, and execute permissions for the owner and group but no permissions for others. The g+s (SGID bit) ensures new files created in the directory inherit the project team group.

RAID Configuration

Question 1: How to create a RAID 1 (mirroring) array using two disks and format it with an ext4 filesystem?

Follow this step:

1. Install mdadm if not already installed
   ```
   sudo apt update
   sudo apt install mdadm
   ```
2. Create RAID 1 array using /dev/sdb and /dev/sdc
   ```
   sudo mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdb /dev/sdc
   ```
3. Format with ext4 filesystem
   ```
   sudo mkfs.ext4 /dev/md0
   ```
4. Create mount point
   ```
   sudo mkdir /mnt/raid1
   ```
5. Mount the RAID array
   ```
   sudo mount /dev/md0 /mnt/raid1
   ```
6. Add to fstab for persistent mounting
   ```
   echo '/dev/md0 /mnt/raid1 ext4 defaults 0 2' | sudo tee -a /etc/fstab
   ```
7. Save RAID configuration
   ```
   sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/mdadm.conf
   sudo update-initramfs -u
   ```

This creates a RAID 1 array that mirrors data across two disks for redundancy.

Question 3: How to check the status of an existing RAID array and replace a failed disk?
Follow this step:

1. Check status of all RAID arrays
   ```
   cat /proc/mdstat
   ```
2. Get detailed information about a specific array
   ```
   sudo mdadm --detail /dev/md0
   ```
3. Mark a failed disk as faulty and remove it
   ```
   sudo mdadm /dev/md0 --fail /dev/sdb
   sudo mdadm /dev/md0 --remove /dev/sdb
   ```
4. Add a new disk to the array
   ```
   sudo mdadm /dev/md0 --add /dev/sdd
   ```
5. Monitor the rebuild process
   ```
   watch cat /proc/mdstat
   ```

After replacing the physical disk, the RAID array will automatically rebuild using the redundant data from the remaining good disk(s).

Practical 7
Networked Storage Management

Question 1: How to mount an NFS share at boot time with specific options?

Follow this step:

1. Install NFS client
   ```
   sudo apt install nfs-common
   ```
2. Create mount point
   ```
   sudo mkdir -p /mnt/shared_data
   ```
3. Test mount the NFS share
   ```
   sudo mount -t nfs fileserver.example.com:/shared_data /mnt/shared_data
   ```
4. Add to fstab for persistent mounting
   ```
   echo'fileserver.example.com:/shared_data/mnt/shared_datanfs
   rw,soft,intr,rsize=8192,wsize=8192 0 0' | sudo tee -a /etc/fstab
   ```
5. Mount all entries in fstab
   ```
   sudo mount -a
   ```

The options set are:

- *rw: Read-write access*
- *soft: Return errors if the server doesn't respond*
- *intr: Allow NFS requests to be interrupted*
- *rsize=8192,wsize=8192: Read and write buffer sizes for better performance*

Question 2: How to connect to and configure an iSCSI target as persistent storage?

Follow this step:

1. Install iSCSI initiator packages
   ```
   sudo apt install open-iscsi
   ```
2. Discover iSCSI targets on a specific portal
   ```
   sudo iscsiadm -m discovery -t sendtargets -p 192.168.1.100
   ```
3. Login to the iSCSI target
   ```
   sudo iscsiadm -m node --targetname "iqn.2023-01.com.example: storage" --portal
   "192.168.1.100:3260" --login
   ```
4. Find the newly connected SCSI disk
   ```
   lsblk
   ```

5. Format the disk (assuming it appears as /dev/sdb)

   ```
   sudo mkfs.ext4 /dev/sdb
   ```

6. Create mount point

   ```
   sudo mkdir /mnt/iscsi_storage
   ```

7. Add to fstab (using disk UUID for persistence)

   ```
   UUID=$(sudo blkid -s UUID -o value /dev/sdb)
   echo "UUID=$UUID /mnt/iscsi_storage ext4 _netdev 0 0" | sudo tee -a /etc/fstab
   ```

8. Mount the filesystem

   ```
   sudo mount /mnt/iscsi_storage
   ```

The _netdev option ensures the filesystem is mounted after the network is available.

Practical 8

File and Directory Management

Question 1: How would you efficiently find and copy all PDF files from multiple subdirectories to a single location while preserving directory structure?

Follow this step:

1. Create destination directory

   ```
   mkdir -p /backup/documents
   ```

2. Find all PDFs and copy with directory structure

   ```
   find /home/user/documents -type f -name "*.pdf" -exec cp --parents {} /backup/documents \;
   ```

3. Check the results

   ```
   ls -la /backup/documents/home/user/documents
   ```

4. Count the number of PDFs copied

   ```
   Find /backup/documents -type f -name "*.pdf" | wc -l
   ```

The --parents option with cp preserves the source directory structure under the target directory.

Question 2: How would you compare two configuration files and merge specific changes using a text editor?

Answer:

```
# First, make a copy of the original for safety
cp server.conf server.conf.original

# View differences between files
diff -u server.conf.original server.conf.new

# Use vimdiff to interactively merge changes
vimdiff server.conf.original server.conf.new

# In vimdiff, use these commands:
# - ]c - jump to next difference
# - [c - jump to previous difference
# - dp - put changes from current buffer to other
# - do - get changes from other buffer to current
```

```
# - :wqa - save and exit both files
```

For a visual diff and merge tool:

```
# Install and use meld
sudo apt install meld
meld server.conf.original server.conf.new
```

Practical 9

File Sharing Setup

Question 1: How to set up an NFS server to share directories with specific client systems?

Follow this steps

# Install NFS server

```
sudo apt install nfs-kernel-server
```

# Create a directory to share

```
sudo mkdir -p /exports/data
```

# Set permissions

```
sudo chown nobody:nogroup /exports/data
sudo chmod 755 /exports/data
```

# Edit exports file

```
sudo vim /etc/exports
```

# Add this line to the file (restricted to specific subnet):

```
/exports/data 192.168.1.0/24(rw,sync,no_subtree_check)
```

# Apply the configuration

```
sudo exportfs -a
```

# Restart NFS server

```
sudo systemctl restart nfs-kernel-server
```

# Check what's being shared

```
showmount -e localhost
```

# On the client machine:

```
sudo apt install nfs-common
sudo mkdir -p /mnt/nfs/data
sudo mount server_ip:/exports/data /mnt/nfs/data
```

The options used in the exports file:

- rw: Allow clients read and write access
- sync: Write changes to disk before replying
- no_subtree_check: Improves reliability when sharing a subdirectory

Practical 8

Monitor and Control Processes using Linux and Windows Tools

Question 1 (Windows)

Task: Use Task Manager to identify and end a high-resource process.

Steps:

1. Press Ctrl + Shift + Esc to open Task Manager.
2. Click on the **Processes** tab.
3. Identify the application using the highest **CPU or Memory**.
4. Right-click the process and select **End Task**.

Expected Outcome:

……………………………………………………………………………………………
……………………………………………………………………………….

Question 2 Task: Use the terminal to monitor and stop a process.

Steps:

1. Open the Terminal in Linux environment.
2. Type top or htop (if installed) to view live processes.
3. Note the **PID** (Process ID) of a high-CPU process.
4. Exit the view using q, then type: kill -9 [PID]
   (replace [PID] with the actual number).

Expected Outcome:

……………………………………………………………………………………………
……………………………………………………………………………….

The process is forcefully terminated using its PID.

Practical 9

Demonstrate How OS Schedules and Switches Between Threads in a Multithreaded Application

Question 1 (Conceptual Simulation in Windows)

Task: Use Process Explorer to observe thread behavior in a multithreaded program.

Steps:

1. Download and run **Process Explorer** from Microsoft Sysinternals.
2. Open a multithreaded application (e.g., a browser).
3. Locate the application in Process Explorer.
4. Right-click it → **Properties** → **Threads** tab.

Expected Outcome:

……………………………………………………………………………………………
…………………………………………………………………………………….

Question 2 (Linux Command Line)

Task: View thread scheduling using top.

Steps:

1. Open Terminal.
2. Run:
   top -H -p [PID]
3. (replace [PID] with the PID of a multithreaded app, such as Firefox or Chromium).
4. Observe the threads and their scheduling behavior.

Expected Outcome:

……………………………………………………………………………………………
…………………………………………………………………………………….

Question 1

Task: Create a background process and terminate it.

Steps:

1. Open Notepad.
2. Minimize it to simulate a background process.
3. Open Task Manager (Ctrl + Shift + Esc) → Find Notepad → **End Task**.

Expected Outcome:

…………………………………………………………………………………………
……………………………………………………………………………….

Question 2 (Linux)

Task: Create a background process and monitor it.

Steps:

1. Open Terminal.
2. Run:
   gedit & *(or any other app like leafpad &)*
3. Type: ps aux | grep gedit to verify it's running.
4. Type: killall gedit to terminate all instances.

Expected Outcome:

…………………………………………………………………………………………
……………………………………………………………………………….

The process is launched in the background, then monitored and terminated via commands.

Practical 9
Configure Memory Settings and Memory Management Strategies in Linux and Windows

Question 1 (Linux): How would you check the current memory usage on a Linux system, and what command can you use to monitor memory consumption in real-time.

Answer:

…………………………………………………………………………………………
……………………………………………………………………………….
…………………………………………………………………………………………
……………………………………………………………………………….

Memory Management and Virtualization

1. Understanding Memory Management Fundamentals

Memory management is a critical function of operating systems, responsible for:

- Allocating memory to processes
- Tracking memory usage
- Optimizing memory utilization
- Protecting memory spaces
- Facilitating efficient data access

Memory Hierarchy:

1. **Registers**: Fastest, smallest storage within CPU
2. **Cache Memory**:
   - L1 Cache: Smallest, fastest (32-64 KB)
   - L2 Cache: Larger, slightly slower (256 KB - 8 MB)
   - L3 Cache: Largest on-chip cache (4-32 MB)
3. **RAM (Random Access Memory)**:
   - Primary working memory
   - Volatile storage
   - Typically, 4-64 GB in modern systems
4. **Virtual Memory**:
   - Uses hard disk as an extension of physical RAM
   - Enables running larger applications than physical memory

Memory Management Strategies

A. Contiguous Memory Allocation

- Allocates consecutive memory blocks to processes
- Simple but leads to fragmentation
- Types:
  1. Single Partition Allocation
  2. Multiple Partition Allocation

B. Paging

- Divides memory into fixed-size blocks
- Eliminates external fragmentation
- Allows non-contiguous memory allocation

Components:

- Page: Fixed-size memory block in process
- Frame: Fixed-size memory block in physical memory
- Page Table: Maps logical to physical addresses

C. Segmentation

- Divides memory into variable-sized segments
- Each segment represents a logical unit (code, data, stack)
- Provides more flexible memory management

How to Capture and Analyze Network Traffic

A. Using tcpdump (Basic Packet Capture)

bash

```
# Capture packets on specific interface
sudo tcpdump -i eth0

# Capture packets and save to file
sudo tcpdump -i eth0 -w capture.pcap

# Filter by host (e.g., only traffic to/from 8.8.8.8)
sudo tcpdump -i eth0 host 8.8.8.8

# Filter by port (e.g., HTTP/HTTPS traffic)
```

```
sudo tcpdump -i eth0 port 80 or port 443

# Read captured file
sudo tcpdump -r capture.pcap
```

B. Using Wireshark (GUI for Deep Analysis)

1. Install Wireshark:

bash

```
sudo apt install wireshark  # Debian/Ubuntu
sudo dnf install wireshark  # Fedora/RHEL
```

Capture live traffic:

bash

```
sudo wireshark
```

- ✓ Select the interface (e.g., eth0, wlan0).
- ✓ Apply filters (e.g., http, tcp.port == 80).
- ✓ Save to .pcap for later analysis.

C. Using tshark (CLI Alternative to Wireshark)

```
# Capture HTTP traffic
sudo tshark -i eth0 -Y "http"

# Capture and save to file
sudo tshark -i eth0 -w traffic.pcap
```

2. Configure a Linux Computer to Obtain IP Details

A. Using dhclient (Dynamic IP Assignment)

```
# Release current lease (if any)
sudo dhclient -r eth0

# Request new IP via DHCP
sudo dhclient eth0
```

B. Using nmcli (NetworkManager)

```
# Check available connections
nmcli con show

# Set interface to DHCP
sudo nmcli con mod "Wired Connection" ipv4.method auto
```

```
# Restart connection
    sudo nmcli con down "Wired Connection" && sudo nmcli con up "Wired Connection"
```

C. Using netplan (Ubuntu 18.04+)

Edit /etc/netplan/01-netcfg.yaml:

```
yaml
network:
  version: 2
  ethernets:
    eth0:
      dhcp4: true
```

Apply changes:

```
bash
sudo netplan apply
```

D. Verify IP Assignment

```
ip addr show eth0
ifconfig eth0  # (if net-tools installed)
nmcli dev show eth0
```

3. Configure IP Addresses, Subnetting, and Routing

A. Assign Static IP Address

Using ip command (Temporary)

```
sudo ip addr add 192.168.1.100/24 dev eth0
sudo ip link set eth0 up
sudo ip route add default via 192.168.1.1
```

Using nmcli (Persistent)

```
sudo nmcli con mod "Wired Connection" ipv4.addresses "192.168.1.100/24"
sudo nmcli con mod "Wired Connection" ipv4.gateway "192.168.1.1"
sudo nmcli con mod "Wired Connection" ipv4.dns "8.8.8.8,8.8.4.4"
sudo nmcli con mod "Wired Connection" ipv4.method manual
sudo nmcli con up "Wired Connection"
```

Using /etc/network/interfaces (Debian/Ubuntu)

Edit /etc/network/interfaces:

```
auto eth0
iface eth0 inet static
   address 192.168.1.100
   netmask 255.255.255.0
   gateway 192.168.1.1
   dns-nameservers 8.8.8.8 8.8.4.4
```

Restart networking:

bash

```
sudo systemctl restart networking
```

B. Subnetting Basics

- **Example:** 192.168.1.0/24 means:
  - ✓ **Network IP:** 192.168.1.0
  - ✓ **Usable Hosts:** 192.168.1.1 to 192.168.1.254
  - ✓ **Broadcast:** 192.168.1.255
  - ✓ **Netmask:** 255.255.255.0

C. Configure Routing

View Routes

bash

```
ip route show
route -n
```

Add Temporary Route

bash

```
sudo ip route add 10.0.0.0/8 via 192.168.1.1
```

Add Persistent Route (Debian/Ubuntu)

Edit /etc/network/interfaces:

```
up ip route add 10.0.0.0/8 via 192.168.1.1
```

4. Network Utilities for Diagnosis & Troubleshooting

A. Basic Connectivity Checks

Command	Purpose
ping 8.8.8.8	Test internet connectivity
traceroute google.com	Check network path
mtr google.com	Continuous traceroute

B. DNS Troubleshooting

Command	Purpose
nslookup google.com	Check DNS resolution
dig google.com	Detailed DNS query
cat /etc/resolv.conf	Check DNS servers

C. Network Interface & Port Checks

Command	Purpose
ip link show	Check interface status
ethtool eth0	Check NIC details
ss -tulnp	List open ports

D. Bandwidth Monitoring

Command	Purpose
nload eth0	Real-time bandwidth
iftop -i eth0	Live connections
vnstat -l	Traffic statistics

E. Firewall & Security Checks

bash

```
sudo iptables -L -n -v # Check firewall rules
sudo ufw status        # Check UFW firewall
```

Summary

Task	**Command**
Capture Traffic	sudo tcpdump -i eth0 -w capture.pcap
Set DHCP	sudo nmcli con mod "Wired" ipv4.method auto
Set Static IP	sudo nmcli con mod "Wired" ipv4.addresses "192.168.1.100/24"
Check Routes	ip route show
Test DNS	dig google.com
Check Open Ports	ss -tulnp
Monitor Traffic	nload eth0