**Day 1: Introduction to Linux & Shell Basics**

**Topics**

1. **Linux Filesystem**:
   * Linux follows a hierarchical directory structure. Key directories include /, /home, /etc, /var, /usr, etc.
   * Understanding the basics of the Linux filesystem is essential for navigating and managing files.
2. **Basic Linux Commands**:
   * ls, cd, pwd, mkdir, touch, cp, mv, rm
   * Learn how to list, create, move, copy, and delete files and directories.

**Practical:**

1. **Navigate the Filesystem**:
   * List files and directories:

bash

Copy code

ls -l # Long listing with details

ls -a # Show hidden files

1. **File Operations**:
   * Create directories and files:

bash

Copy code

mkdir my\_directory # Create a new directory

touch my\_file.txt # Create a new empty file

1. **Copy, Move, and Remove Files**:

bash

Copy code

cp my\_file.txt /path/to/destination # Copy file

mv my\_file.txt /path/to/destination # Move or rename file

rm my\_file.txt # Delete file

1. **Viewing File Content**:

bash

Copy code

cat file.txt # Display entire file

less file.txt # View file page by page

head file.txt # Display the first 10 lines of a file

tail -f file.txt # Continuously view the last lines of a file

**Day 2: User Management and Permissions**

**Topics**

1. **User and Group Management**:
   * Add, modify, and delete users and groups using useradd, groupadd, usermod, userdel, etc.
2. **File Permissions**:
   * Understanding file ownership (user, group, others).
   * Changing file permissions using chmod, chown, and chgrp.

**Practical:**

1. **Create and Manage Users and Groups**:
   * Create a new user:

bash

Copy code

sudo useradd myuser

sudo passwd myuser # Set password

* + Create a new group:

bash

Copy code

sudo groupadd mygroup

* + Add user to a group:

bash

Copy code

sudo usermod -aG mygroup myuser

1. **File Permissions**:
   * Check file permissions:

bash

Copy code

ls -l file.txt # View file permissions and ownership

* + Change file permissions:

bash

Copy code

sudo chmod 755 file.txt # Owner can read/write/execute, others can read/execute

sudo chmod u+x file.txt # Add execute permission to user

1. **Change File Ownership**:

bash

Copy code

sudo chown user:group file.txt # Change file owner and group

**Day 3: Process Management**

**Topics**

1. **Managing Processes**:
   * ps, top, htop, kill, pkill, nice, renice
   * Understanding process IDs (PIDs) and managing processes.
2. **System Monitoring**:
   * Check system resources (CPU, memory, disk) using tools like top, free, df, du, uptime.

**Practical:**

1. **View Running Processes**:
   * List processes:

bash

Copy code

ps aux # Show all running processes

* + Monitor system in real-time:

bash

Copy code

top # Real-time monitoring of processes

htop # Enhanced version of top (requires installation)

1. **Kill Processes**:
   * Kill a process by PID:

bash

Copy code

kill -9 <PID> # Forcefully terminate a process

* + Kill a process by name:

bash

Copy code

pkill process\_name

1. **Check System Resources**:
   * CPU usage:

bash

Copy code

top # In the summary area, CPU stats are displayed

* + Memory usage:

bash

Copy code

free -h # Display memory usage

* + Disk usage:

bash

Copy code

df -h # Display disk space usage

**Day 4: Package Management**

**Topics**

1. **Package Management (Debian-based)**:
   * apt-get, apt-cache, dpkg
   * Install, upgrade, and remove packages.
2. **Package Management (Red Hat-based)**:
   * yum, rpm

**Practical:**

1. **Install and Manage Packages (Debian-based)**:
   * Update package list:

bash

Copy code

sudo apt update

* + Install a package:

bash

Copy code

sudo apt install nginx # Install Nginx web server

* + Upgrade packages:

bash

Copy code

sudo apt upgrade # Upgrade all installed packages

1. **Install and Manage Packages (Red Hat-based)**:
   * Install a package:

bash

Copy code

sudo yum install httpd # Install Apache web server

* + Update all packages:

bash

Copy code

sudo yum update

**Day 5: Networking and Firewall Management**

**Topics**

1. **Basic Networking Commands**:
   * ifconfig, ip, ping, netstat, ss
2. **Firewall Management**:
   * ufw, iptables for managing firewall rules.

**Practical:**

1. **Check Network Configuration**:
   * View IP addresses:

bash

Copy code

ifconfig # Older tool

ip addr show # Modern tool for network config

1. **Check Network Connectivity**:
   * Test connection to a remote host:

bash

Copy code

ping google.com

1. **Check Open Ports**:
   * List listening ports:

bash

Copy code

netstat -tuln

1. **Manage Firewall (UFW)**:
   * Enable firewall:

bash

Copy code

sudo ufw enable

* + Allow a port (e.g., HTTP):

bash

Copy code

sudo ufw allow 80/tcp

* + Check firewall status:

bash

Copy code

sudo ufw status

**Day 6: Automating Tasks with Cron**

**Topics**

1. **Cron Jobs**:
   * Automate repetitive tasks using the cron scheduler (crontab).

**Practical:**

1. **Schedule a Cron Job**:
   * Edit the crontab file:

bash

Copy code

crontab -e # Edit cron jobs

* + Example of a cron job to run a script every day at midnight:

bash

Copy code

0 0 \* \* \* /path/to/script.sh

1. **List Cron Jobs**:
   * View current user's cron jobs:

bash

Copy code

crontab -l

**Day 7: Log Management and Monitoring**

**Topics**

1. **Log Files**:
   * Understanding system logs in /var/log/, e.g., /var/log/syslog, /var/log/auth.log.
2. **Log Rotation**:
   * Managing log file sizes with logrotate.

**Practical:**

1. **View System Logs**:
   * Check system logs:

bash

Copy code

tail -f /var/log/syslog # Follow system log

1. **Configure Log Rotation**:
   * Edit logrotate configuration:

bash

Copy code

sudo nano /etc/logrotate.d/nginx

**Day 8: Advanced Scripting and Automation**

**Topics**

1. **Bash Scripting**:
   * Writing bash scripts to automate tasks like backups, deployments, etc.
2. **Automating DevOps Tasks**:
   * Automate server provisioning, deployments, and monitoring tasks using bash.

**Practical:**

1. **Write a Simple Bash Script**:
   * Create a simple script that updates the system and restarts a service:

bash

Copy code

#!/bin/bash

sudo apt update

sudo apt upgrade -y

sudo systemctl restart nginx

echo "System updated and Nginx restarted"

1. **Make the Script Executable**:
   * Grant execute permission to the script:

bash

Copy code

chmod +x update\_system.sh

./update\_system.sh # Run the script

**Day 9: Docker and Containers in Linux**

**Topics**

1. **Docker Basics**:
   * Learn how to install Docker, build, run, and manage containers.

**Practical:**

1. **Install Docker**:

bash

Copy code

sudo apt install docker.io

1. **Run a Docker Container**:
   * Pull and run an official Docker image:

bash

Copy code

sudo docker run -d -p 80:80 nginx