Instructions for Journal Writeup

1. **Write the following content for each practical in your manual assigned for the OST subject.**
2. **You need not print any page for the lab manual.**
3. **Make sure you start new practicals on a new page.**
4. **Use INDEX relevant to your Division only (Div-A / Div-B)**
5. **Mention your OWN PAGE NUMBERS for lab manual and mention it correctly under page column of your INDEX**
6. **Deadline to get your completed lab manual for OST subject checked by your faculty is 17/04/2025**

Walchand Institute of Technology, Solapur

Department of Electronics and Telecommunication Engineering

T.Y. B. Tech. (Electronics & Telecommunication Engineering), Semester-VI
22ETU6CC5P – Open-Source Technologies

Index

Walchand Institute of Technology, Solapur

Department of Electronics and Telecommunication Engineering

T.Y. B. Tech. (Electronics & Telecommunication Engineering), Semester-VI
22ETU6CC5P – Open-Source Technologies

Index

Sr. No.	Title of Practical	Batchwise Dates
1	Linux Installation and Basic Setup	B1: 09/01/2025
		B2: 07/01/2025
		B3: 06/01/2025
2	File and Directory Management in Linux	B1: 16/01/2025
		B2: 14/01/2025
		B3: 13/01/2025
3	Understanding File Permissions and Ownership in Linux	B1: 06/02/2025
		B2: 04/02/2025
		B3: 03/02/2025
4	User Account Management in Linux	B1: 13/02/2025
		B2: 11/02/2025
		B3: 17/02/2025
5	Monitoring and Managing Services in Linux	B1: 27/02/2025
		B2: 25/02/2025
		B3: 24/02/2025
6	Shell Programming Using Bash	B1: 06/03/2025
		B2: 04/03/2025
		B3: 03/03/2025
7	Installation and Configuration of Git	B1: 20/03/2025
		B2: 18/03/2025
		B3: 17/03/2025
8	File Handling and Repository Management with GitHub	B1: 03/04/2025
		B2: 01/04/2025
		B3: 07/04/2025

1. Linux Installation and Basic Setup

Theory: Linux is an open-source operating system widely used in servers, desktops, and embedded systems. It is based on the Unix architecture and provides a powerful platform for development, system administration, and security management. Common Linux distributions include Ubuntu, Fedora, Debian, and CentOS.

Installation Process:

1. Download ISO image from the official website of the desired Linux distribution.
2. Create a bootable USB using tools like Rufus or BalenaEtcher.
3. Boot the system from the USB drive.
4. Choose installation options such as disk partitioning, timezone, and user setup.
5. Complete installation and reboot the system.

Essential Post-Installation Steps:

- Update system: `sudo apt update && sudo apt upgrade`
- Install necessary software: text editors, compilers, etc.
- Configure network settings if needed.

Viva Tips: Know the difference between various Linux distros, understand what a kernel is, and explain the boot process.

Conclusion: Installing Linux is a fundamental skill for system administrators and developers. Understanding the installation process helps in setting up a stable and secure working environment tailored to specific needs.

---

2. File and Directory Management in Linux

Theory: In Linux, everything is treated as a file, including hardware and processes. The file system is hierarchical, starting from the root directory `/`.

Common Commands:

- `ls` - list files
- `cd` - change directory
- `mkdir` - create directory
- `rm` - remove file or directory
- `cp` - copy files
- `mv` - move or rename files
- `touch` - create an empty file

Paths:

- Absolute path: starts from root, e.g., `/home/user/docs`
- Relative path: relative to the current directory, e.g., `../folder`

Viva Tips: Be able to navigate the file system and use basic file handling commands confidently.

Conclusion: Efficient file and directory handling is crucial for managing a Linux system. Mastering these commands allows users to perform tasks quickly and accurately in any Linux environment.

---

3. Understanding File Permissions and Ownership in Linux

Theory: Linux uses a permission system to protect files and directories. Each file has an owner and a group. Permissions are defined for the user (owner), group, and others.

Permission Types:

- `r` - read
- `w` - write
- `x` - execute

Command Examples:

- `ls -l` - shows permission details
- `chmod` - changes permissions (symbolic or numeric)
- `chown` - changes ownership

Permission Representation: Example: `-rwxr-xr--`

- User: rwx (read, write, execute)
- Group: r-x (read, execute)
- Others: r-- (read only)

Viva Tips: Understand symbolic vs numeric permission notation, and how to use `chmod`, `chown`.

Conclusion: Proper file permissions and ownership settings ensure data security and prevent unauthorized access. Understanding these concepts is vital for maintaining a secure Linux system.

---

4. User Account Management in Linux

Theory: Linux supports multiple users. Each user has a unique UID, home directory, and shell.

User Management Commands:

- `adduser` or `useradd` - add a user
- `passwd` - set/change password
- `deluser` or `userdel` - delete a user
- `usermod` - modify user accounts

Group Management:

- `groupadd`, `groupdel`, `usermod -aG` for adding users to groups

Files to Know:

- `/etc/passwd` - user account info
- `/etc/shadow` - encrypted password
- `/etc/group` - group info

Viva Tips: Know the purpose of each system file and differences between `adduser` and `useradd`.

Conclusion: User and group management is essential for a multi-user operating system like Linux. These tools ensure that user access is appropriately controlled and system integrity is maintained.

---

5. Monitoring and Managing Services in Linux

Theory: Linux uses services (also called daemons) to manage background tasks.

Systemd Commands:

- `systemctl start servicename`
- `systemctl stop servicename`
- `systemctl enable servicename`
- `systemctl status servicename`

Service Example:

- `ssh`, `apache2`, `cron`, etc.

Legacy: Older systems used `service` command and init.d scripts.

Viva Tips: Understand the role of `systemctl`, and how to check logs using `journalctl`.

Conclusion: Service management is critical for ensuring the availability and functionality of server processes. Knowing how to control services helps in system maintenance and troubleshooting.

---

6. Shell Programming Using Bash

Theory: Bash (Bourne Again Shell) is a Unix shell and command language. Shell scripts automate repetitive tasks.

Basic Script Structure:

#!/bin/bash

This is a comment

echo "Hello, World!"

Variables, Loops, and Conditionals:

- Variables: `name=John`
- If statement: `if [ $a -eq $b ]; then ... fi`
- Loop: `for`, `while`

Execution:

- `chmod +x script.sh`
- `./script.sh`

Viva Tips: Be ready to explain variables, if conditions, and loops with syntax.

Conclusion: Bash scripting enables automation of tasks and customization of the Linux environment. It's a powerful tool for administrators and developers alike.

---

7. Installation and Configuration of Git

Theory: Git is a distributed version control system that tracks changes in source code.

Installation:

- Ubuntu: `sudo apt install git`

Basic Configuration:

git config --global user.name "Your Name"

git config --global user.email "you@example.com"

Common Commands:

- `git init`
- `git status`
- `git add .`
- `git commit -m "message"`

Viva Tips: Understand what a commit is, and how Git tracks changes.

Conclusion: Git is a critical tool for modern software development. Proper installation and configuration lay the groundwork for efficient version control and collaboration.

---

8. File Handling and Repository Management with GitHub

Theory: GitHub is a web-based Git repository hosting service. It allows developers to share code and collaborate.

Steps to Use GitHub:

1. Create a repository on GitHub.
2. Link local repo: `git remote add origin <URL>`
3. Push changes: `git push -u origin master`

Cloning and Pulling:

- `git clone <URL>`
- `git pull`

Viva Tips: Be able to explain `clone`, `push`, `pull`, and resolve merge conflicts.

Conclusion: GitHub streamlines collaborative development and code management. Mastery of repository handling enables teams to work efficiently on shared projects.

Additional Resources

Commands to configure GIT local and remote repository

```
mkdir manost
cd manost/
git init
git status
sudo git config --global user.name "your_GIT_username"
sudo git config --global user.email "your_GIT_emailID"
git remote add origin your_GIT_repo_URL
git remote set-url origin
https://your_GIT_username:your_GIT_TOKEN@github.com/your_GIT_username/your_GIT_Repo_Name
sudo nano mycode.py
cat mycode.py
git add .
git status
git commit -m "my first code"
git push origin master
```