

# Basic Linux Command Line Exercises Worksheet

## 1. Navigating the File System

**Objective:** Master navigating the Linux file system using essential commands.

**Commands to Learn:** - `pwd` – Print the working directory - `cd` – Change directory - `ls` – List directory contents - `cd ..` – Move to the parent directory - `cd -` – Switch to the previous directory - `ls -a` – List all files, including hidden files

**Exercises:** 1. Use the `pwd` command to display your current directory. What does the output tell you? 2. Use the `ls` command to list all files and directories in your current location. What type of files and directories do you see? 3. Change to the `/home` directory using `cd /home`. Use `pwd` to confirm you've moved. 4. Return to your previous directory using `cd -`. Use `pwd` again to confirm. 5. Navigate to the `/etc` directory. List the contents using `ls -a` to view all files, including hidden files. 6. Move up one level to the parent directory using `cd ..` and list the contents there.

**Challenge:** 1. Try combining commands. Use `ls -l /home` to list all files in `/home` with detailed information. 2. Use the `cd` command to navigate to three different directories and check your location each time with `pwd`.

---

## 2. Working with Files and Directories

**Objective:** Learn how to manage files and directories using the command line.

**Commands to Learn:** - `touch` – Create a new empty file - `mkdir` – Make a new directory - `cp` – Copy files and directories - `mv` – Move or rename files and directories - `rm` – Remove files - `rmdir` – Remove empty directories - `rm -r` – Remove directories and their contents

**Exercises:** 1. Create a new file named `testfile.txt` using the `touch` command. Confirm its creation with `ls`. 2. Create a new directory called `testdir` using `mkdir`. Confirm it exists with `ls`. 3. Move `testfile.txt` into `testdir` using the `mv` command. Use `ls testdir` to ensure it's inside the directory. 4. Copy `testfile.txt` back into the current directory using `cp testdir/testfile.txt ..`. Use `ls` to confirm the file is copied. 5. Rename `testfile.txt` to `newfile.txt` using `mv`. 6. Remove `newfile.txt` using `rm`. Verify it's deleted with `ls`. 7. Try to remove `testdir` using `rmdir`. What happens if the directory is not empty? 8. Use `rm -r` to delete `testdir` and its contents.

**Challenge:** 1. Create a directory structure like this: `project/ docs/ src/` Use `mkdir` to create the directories, and verify the structure with `tree` or `ls -R`. 2. Write a command to copy everything inside `project` to a new directory called `backup`.

---

## 3. Viewing and Editing Files

**Objective:** Master file viewing and basic text editing through the command line.

**Commands to Learn:** - `cat` – Display the entire content of a file - `more/less` – View file content page by page - `head` – View the first lines of a file - `tail` – View the last lines of a file - `nano` or `vi` – Edit files in the terminal

**Exercises:** 1. Use `cat /etc/hosts` to display the contents of the `hosts` file. What is the file used for? 2. Open a larger file (such as `/var/log/syslog`) using `less`. Use the arrow keys to scroll through the file. 3. View only the first 10 lines of `/var/log/syslog` using `head`. 4. Use `tail` to view the last 10 lines of `/var/log/syslog`. 5. Create a file called `notes.txt` using `nano` or `vi`. Write some text and save the file. Display the file contents using `cat`. 6. Open `notes.txt` again and append some new text. Save and close the file. Verify the changes with `cat`.

**Challenge:** 1. Use `tail -f /var/log/syslog` to monitor the system log in real-time. Open a new terminal and run a system command that generates log entries (e.g., `ping google.com`). Watch the log update live.

---

## 4. File Permissions

**Objective:** Understand how to view and modify file and directory permissions.

**Commands to Learn:** - `chmod` – Change file permissions - `chown` – Change file ownership - `ls -l` – View file permissions

**Exercises:** 1. Create a file called `securefile.txt` and view its permissions using `ls -l`. 2. Change the permissions of `securefile.txt` so that the owner has read, write, and execute permissions, and the group and others have no permissions. Use `chmod 700 securefile.txt`. 3. View the updated permissions using `ls -l`. 4. Create a new directory called `sharedir`

and change its permissions so that everyone can read, write, and execute files in it (`chmod 777 sharedir`). 5. Change the ownership of `securefile.txt` to another user (if applicable) using `chown username securefile.txt`.

**Challenge:** 1. Try to create a file inside `sharedir` as another user (if possible) and verify that the permissions allow the action. 2. Explore symbolic permissions (e.g., `chmod u+x` to give the owner execute permissions) and practice changing specific permissions using symbolic notation.

---

## 5. Process Management

**Objective:** Learn how to view and manage running processes in Linux.

**Commands to Learn:** - `ps` – List running processes - `top` – Monitor system processes in real-time - `kill` – Terminate a process - `kill -9` – Forcefully terminate a process - `bg/fg` – Manage background and foreground processes

**Exercises:** 1. Use `ps aux` to list all running processes. Identify the process ID (PID) of a process related to your user account. 2. Run `top` to monitor system processes. Identify which process is using the most CPU and memory. 3. Start a process in the background by running `ping google.com > pinglog.txt &`. Use `jobs` to list background jobs. 4. Bring the process to the foreground using `fg` and then stop it with `Ctrl+C`. 5. Start a long-running process in the background (e.g., `ping google.com`). Use `ps` to find its PID and then terminate it with `kill`. 6. Forcefully terminate the process if it doesn't stop using `kill -9`.

**Challenge:** 1. Start a background job, then use `bg` and `fg` to move it between the foreground and background. Practice stopping and restarting it. 2. Write a script that runs a process in the background and sends a notification when it finishes.

---

## 6. Basic Networking

**Objective:** Learn how to check and manage network settings using the command line.

**Commands to Learn:** - `ifconfig` or `ip addr` – View network interface configuration - `ping` – Test network connectivity - `netstat` – View network connections and routing tables

**Exercises:** 1. Use `ifconfig` (or `ip addr`) to view the current configuration of your network interfaces. What is your IP address? 2. Use `ping google.com` to test internet connectivity. How many packets were sent and received? 3. Use `netstat -tuln` to list all open ports and listening services on your machine. 4. Run `ping -c 5 google.com` to send exactly 5 pings. Analyze the output to understand round-trip time and packet loss. 5. Use `ifconfig` or `ip addr` to bring down (disable) a network interface and then bring it back up again.

**Challenge:** 1. Use `netstat` to identify any active network connections. Try to find connections to external servers. 2. Research how to use the `traceroute` command (if installed) to trace the path packets take to reach a website.

---

## 7. Finding Help

**Objective:** Learn how to use built-in help resources on Linux.

**Commands to Learn:** - `man` –

Display the manual for a command - `--help` – Display a quick help overview for a command

**Exercises:** 1. Use the `man` command to read the manual for the `ls` command. What are some useful options you can use with `ls`? 2. Try using `ls --help` to see a brief overview of its options. 3. Use `man` to find out what the `chmod` command does and how to use symbolic permissions. 4. Explore the `man` pages for other commands you've used today, like `ps`, `mv`, or `rm`.

**Challenge:** 1. Look up the manual entry for the `find` command, a powerful tool for searching files. Try using `find` to locate a file by name within your home directory.

---

## Bonus Challenge: Shell Script Writing

**Objective:** Write a basic shell script to automate common tasks.

**Exercise:** 1. Write a script that: - Creates a backup directory. - Copies a specific file into the backup directory. - Compresses the backup directory into a `.tar.gz` archive. 2. Run the script and verify that it works correctly by checking the files and archive.

---