

Linux Commands Course – Final Project

Hands-on Terminal Challenge (Standardized Edition)

Welcome to the Final Project

You've completed the entire Linux journey – from shell fundamentals to scripting and automation.

Now it's time to **prove your mastery**.

You will complete this project entirely from the terminal.

Each stage must be recorded separately using **asciinema** (one recording per stage).

Preparation

1. Work in your **home directory**.
 2. Create a folder named **final_mission** and move into it.
 3. Inside it, store all files, scripts, and outputs.
 4. Maintain a clean folder structure and logical file names.
-

Stage 1 – Orientation

Show understanding of basic shell usage.

Tasks:

1. Identify your current shell and display your username and home directory.
2. Demonstrate at least two ways to find what the `echo` command does.
3. Print the following text:

```
My name is Elnur Student
```

4. Create a text file `shortcuts.txt` with these exact contents:

```
Ctrl+C - Stop running command  
Ctrl+D - Exit shell  
Ctrl+L - Clear screen
```

Stage 2 – Files & Navigation

Work with files and directory structures.

Tasks:

1. Create folders: `reports`, `logs`, and `scripts`.
2. Inside `reports/`, create `mission_text.txt` with the text below, and btw write definitions of those:

CIA -
AAA -
Zero Trust -
Linux/GNU -

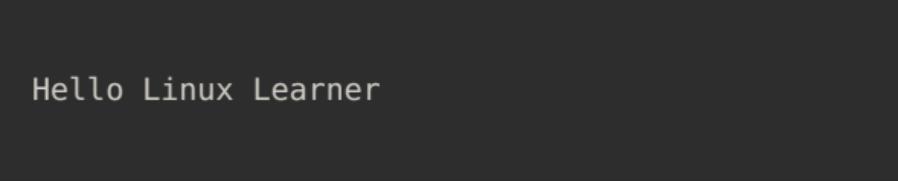
3. Copy it to `reports/backup_text.txt`.
4. Create a symbolic link `latest_text` pointing to the backup file.
5. Inspect and confirm links and metadata; store verification in `reports/structure_check.txt`.

Stage 3 – Permissions & Ownership

Show control over access rights and file security.

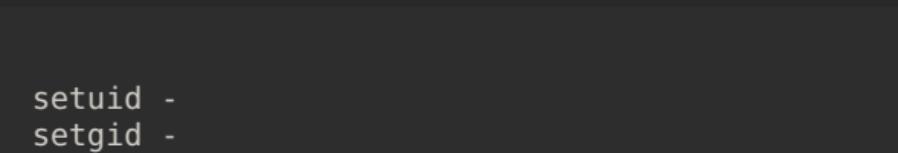
Tasks:

1. Create `hello.sh` that prints:



```
Hello Linux Learner
```

2. Make it executable only by you.
3. Create `/tmp/final_shared` (with `sudo`) and apply group write + setgid bit.
4. Create `/tmp/final_public` with sticky bit enabled.
5. Save permission verification into `logs/perm_check.log`.
6. Create `logs/special_bits.txt` containing its definitions:



```
setuid -  
setgid -
```

Stage 4 – Searching & Filtering

Apply text searching and filtering logic.

Tasks:

1. Ensure there are at least three .txt files in your project. Create `notes.txt` with:

```
The Linux project teaches mastery of the shell.  
Project stages are designed for practice.  
Every project must be completed fully.
```

2. Search for lines containing `linux` (case-insensitive).
3. Exclude results from `backup_brief.txt`.
4. Save all matching lines into `found_lines.log`.
5. Append the total number of matching lines at the end of that file.

Stage 5 – Pipelines & Redirection

Demonstrate stream control and redirection.

Tasks:

1. List all files and directories (including hidden) and save to `all_files.log`.
 2. Append the current date and time at the bottom.
 3. Append total count of entries(the command used in 1.) below the date.
 4. Use `tee` in your process and explain its role in `logs/tee_explanation.txt`.
-

Stage 6 – Text Processing

Test file transformation and analysis.

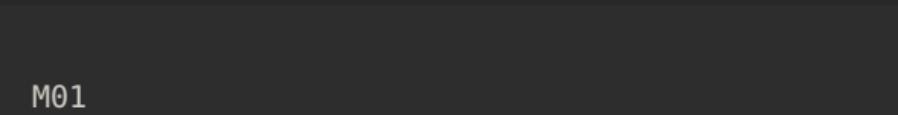
Tasks:

1. Create:
 - agents.txt



Beta
Omega
Alpha
Gamma
Delta

- missions.txt



M01

Stage 7 – Archiving & Compression

Demonstrate backup and compression.

Tasks:

1. Create a folder `backup/` and copy all the files you have in the project into the new folder.
2. Archive it as `project_backup.tar`.
3. Compress into:
 - `project_backup.tar.gz`
 - `project_backup.tar.xz`
4. Compare their sizes and record in `backup/compression_report.txt`:

GZIP size: ____
XZ size: ____
Smaller format: ____

5. Extract the `.xz` archive to `restore_test/` and verify correctness.

Stage 8 – System Directories Insight

Observe key system and virtual directories.

Tasks:

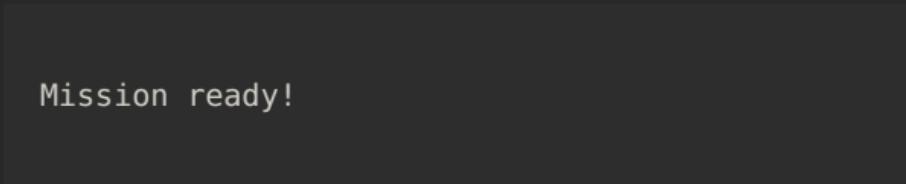
1. Copy first 3 lines of any file in /etc to reports/etc_sample.txt.
 2. Save first 5 lines of /proc/cpuinfo and /proc/meminfo to reports/sys_info.txt.
 3. Copy /etc/resolv.conf to reports/dns_info.txt.
 4. Add alias ll='ls -la' to ~/.bashrc, reload, and verify.
 5. Save first 5 lines of a log from /var/log into logs/sample_log.txt.
-

Stage 9 – Users, Groups & sudo

Apply identity and privilege management.

Tasks:

1. Create user `yaxsiqaqash` (home + bash shell).
2. Create group `qaqalar` and add the new user into it.
3. Verify and save to `logs/user_check.txt`.
4. Add him to the `sudo` group.
5. Extract sudo group rules into `logs/sudo_rules.txt`.
6. Inside `home` of the new user, create `notes.txt`:



Mission ready!

7. Fix ownership if necessary and verify.

Stage 10 – Processes & Jobs

Handle process control and job management.

Tasks:

1. Start a background process lasting over a minute.
 2. Stop, resume, and list it.
 3. Find PID and adjust its priority.
 4. Terminate it properly.
 5. Save top 5 memory-using processes to `logs/mem_top5.txt`.
-

Stage 11 – Services & Logs

Work with `systemd` services and logs.

Tasks:

1. Identify current boot target → `reports/boot_target.txt`.
 2. Restart a networking-related service.
 3. Enable NTP sync.
 4. Save full boot logs → `logs/journal_recent.log`.
 5. Save only error logs → `logs/journal_errors.log`.
 6. Save last 10 lines of `/var/log/syslog` → `logs/sys_tail.log`.
-

Stage 12 – Networking

Perform basic network analysis.

Tasks:

1. Record IP addresses and routes → `network_tests/net_info.txt`.
 2. Append listening ports to same file.
 3. Test connectivity (one IP, one domain).
 4. Trace route to `example.com`.
 5. Query MX records of `example.com`.
 6. Download its homepage via two tools, saving as:
 - `network_tests/example_curl.html`
 - `network_tests/example_wget.html`
-

Stage 13 – Package Management

Use system package managers effectively.

Tasks:

1. Update and upgrade system.
 2. Install `tree` and `htop`.
 3. Save their versions → `logs/package_versions.txt`.
 4. Remove one non-core package.
 5. Save info about one installed package → `logs/pkg_info.txt`.
 6. Save list of packages containing “python” → `python_packages.txt`.
-

Stage 14 – Disks & Filesystems

Inspect and report on storage.

Tasks:

1. Save disk and filesystem info → `reports/disk_status.txt`.
2. Save top 5 largest directories in home → `reports/home_usage.txt`.
3. Copy `/etc/fstab` → `reports/fstab_copy.txt`.
4. Record current swap usage → `reports/swap_status.txt`.
5. Create `reports/disk_summary.txt`:

```
Disk and filesystem overview completed successfully.  
All mounts, fstab entries, and swap status verified.
```

Stage 15 – Scheduling Tasks

Demonstrate automation timing.

Tasks:

1. Schedule a job appending current date to `/tmp/daily_log.txt` every minute.
2. Schedule one-time job writing One-time mission completed! into `/tmp/at_log.txt`.
3. Save verification of scheduled jobs → `logs/schedule_check.txt`.
4. After execution, copy results → `reports/schedule_results.txt`.
5. Append this explanation:

Cron is for repeated jobs, At is for one-time jobs.

Stage 16 – Bash Scripting Challenge

Integrate automation and reporting.

Tasks:

1. Create `system_summary.sh` that prints:
 - Current date & user
 - Uptime
 - Disk and memory usage
 - Top 5 CPU processes
 2. Include section titles.
 3. Save output to `summary_report.txt`.
 4. Use functions, error handling, and traps.
 5. Make executable and test.
-

Stage 17 – Customization & Environment

Adapt your environment for efficiency.

Tasks:

1. Add alias:

```
alias updateall="sudo apt update && sudo apt upgrade -y"
```

2. Add `$HOME/final_mission/scripts` to PATH.
3. Enable timestamps in command history.
4. Add:

```
export PROJECT_ROOT="$HOME/final_mission"
```

Final Submission

- ✓ Compress final_mission:

```
tar -czf final_mission_submission.tar.gz ~/final_mission
```

- ✓ Submit:

- Your compressed folder
- All **asciinema recordings** (one per stage)

Evaluation Criteria:

- Accuracy and completeness
- Proper command choice
- Standardized file contents
- Clean folder organization and documentation