

Terminal Practice

Fabio Pizzati

January 2026

Goal

Practice core terminal workflows:

- navigating directories and listing files
- creating, editing, viewing, copying, moving, and deleting files/directories
- managing permissions and (optionally) ownership
- running a process, identifying it, and terminating it
- using `man` pages

Rules

- Do **not** use a GUI file explorer.
- Keep your work inside the directory you create for this lab.
- Be careful with recursive deletion.

1 Part A — Project structure

Create the following structure (exact names required):

```
terminal_lab/  
  project_1/  
    src/  
    data/  
    notes/  
  project_2/
```

Tasks:

- A1.** Verify your current location in the filesystem.
- A2.** Create the directory structure above (including nested directories).
- A3.** Move into `terminal_lab/` and verify the structure by listing directories in multiple ways (including showing hidden files and detailed listing).
- A4.** Create an empty file `project_1/notes/description.txt`.

2 Part B — Edit and view

- B1. Edit `project_1/notes/description.txt` and write **at least 15 lines** (any content. You can quit nano with Control+X).
- B2. Display the file in three different ways: full content, beginning only, and end only.

3 Part C — Copy, move, rename, delete

- C1. Create a copy of `description.txt` in the same folder.
- C2. Rename the copy to `lorem_ipsum.txt`.
- C3. Move `lorem_ipsum.txt` to the `terminal_lab/` folder.
- C4. Delete `lorem_ipsum.txt`.
- C5. Make a recursive backup copy of `project_1/` into `project_2/` (name it `project_1_copy`).

4 Part D — Remove directories safely

- D1. Create an empty directory inside `project_1/` (choose a name), then remove it using the command that only works for **empty** directories.
- D2. Create a directory `project_1/to_delete/sub/` and place a file inside it, then remove `to_delete/` recursively. **PAY ATTENTION.**

5 Part E — Permissions and the “annoying” program

Create a script file `project_1/src/annoying.sh`. Paste **exactly** this content:

```
#!/bin/bash
while true; do
    echo "I'm annoying"
    sleep 10
done
```

- E1. Try running the script from the terminal (from the correct directory). Observe what happens.
- E2. Change its permissions so it can be executed by the user.
- E3. Run it so it prints `I'm annoying` every 10 seconds.

6 Part F — Identify and kill the process

- F1.** In a second terminal window/tab, list processes.
- F2.** Find the process corresponding to the running script and identify its PID.
- F3.** Terminate it using a signal kill command.
- F4.** Confirm that it is no longer running.

7 Part G — Users, groups, and manual pages

- G1.** Display the logged-in users and your groups.
- G2.** Open the manual pages for at least two commands you used in this lab (one must be `chmod` or `ls`) and note one flag/option meaning.
- G3.** (Optional) Open the manual for `chown`. Do not run `chown` unless you understand permissions on your system.

Expected Final State

At the end of the exercise, your filesystem must have the following structure inside `terminal_lab/`:

```
terminal_lab/
|-- project_1/
|   |-- src/
|   |   '-- annoying.sh
|   |-- data/
|   '-- notes/
|-- '-- description.txt
'-- project_2/
    '-- project_1_copy/
        |-- src/
        |-- data/
        '-- notes/
        '-- description.txt
```

The file `annoying.sh` must be executable by the user and contain exactly the provided script. When executed, it must print `I'm annoying` every 10 seconds.

The file `description.txt` must contain at least 15 lines of text. All temporary files and directories created during the exercise (e.g., empty folders, backup files, or test directories) must have been removed.

Terminal Commands Cheatsheet

Navigation & Directories

<code>pwd</code>	Show current working directory
<code>ls</code>	List files in current directory
<code>ls -a</code>	List all files (including hidden)
<code>ls -l</code>	List files with details
<code>ls -la</code>	List all files with details
<code>ls <dir></code>	List contents of a specific directory
<code>cd <dir></code>	Change current directory

Viewing Files

<code>cat file.txt</code>	Show entire file content
<code>head file.txt</code>	Show beginning of file
<code>tail file.txt</code>	Show end of file

Copying, Moving, Deleting

<code>cp a b</code>	Copy file a to b
<code>cp -r dir1 dir2</code>	Copy directory recursively
<code>mv a b</code>	Move or rename file
<code>rm file</code>	Delete file
<code>rm -r dir</code>	Delete directory recursively (dangerous)

Creating & Editing

<code>mkdir dir</code>	Create directory
<code>mkdir -p dir/sub</code>	Create nested directories
<code>rmdir dir</code>	Remove empty directory
<code>touch file</code>	Create empty file
<code>nano file</code>	Edit file with nano editor

Permissions & Ownership

<code>chmod u+x file</code>	Add execute permission to user
<code>chmod g-x file</code>	Remove execute from group
<code>chmod a+rw file</code>	Give all permissions to everyone
<code>chown user file</code>	Change file owner
<code>chown :group file</code>	Change file group
<code>chown user:group file</code>	Change owner and group
<code>groups</code>	Show groups of current user
<code>users</code>	Show logged-in users

Processes

<code>ps</code>	Show processes in current shell
<code>ps aux</code>	Show all running processes
<code>ps aux grep name</code>	Filter processes by name
<code>kill -SIGKILL PID</code>	Force kill process by PID

Help

<code>man command</code>	Show manual page for a command
--------------------------	--------------------------------