Lab: Practicing the Unix Command Line Environment

DS2040, Satyajit Das

January 9, 2025

1 Objective

This lab aims to get comfortable navigating and manipulating files and directories in a Unix-like environment. You will learn how to:

- Navigate the directory structure
- Create, view, copy, move, and delete files/directories
- Search and filter content within files
- Use redirection and piping to connect commands
- Understand basic permissions
- Use environment variables and simple scripting

Some commands, especially the *sudo* environment, may not be available in the lab machines. Please practice these commands on your personal computers in Unix environments.

2 Prerequisites

- A Unix-like environment (Linux, macOS, or a VM/container running Linux).
- Access to a terminal or command-line interface.
- Basic understanding of what a shell is (e.g., Bash).

3 Preparations

1. **Terminal Setup:** Open a terminal window (Linux or macOS) or use an SSH client to log in to a Unix server.
2. **Home Directory:** Confirm you are in your home directory by typing:

```
cd ~
pwd
```

   This should display your home directory path, for example `/home/username`.

4 Exercises

4.1 4.1 Navigating Directories

1. **Check Current Directory**
   Command: `pwd`
   Write down the output to confirm your current location.

2. **List Files**
   Command: `ls`
   Observe any files or folders. Try `ls -l` and `ls -a` to see additional details.

3. **Create and Navigate Folders**
   Commands:

```
mkdir unix_lab
cd unix_lab
```

   Confirm you are inside the new folder with `pwd`.

4. **Make a Subdirectory**
   Command: `mkdir test_dir`
   Navigate into `test_dir` (`cd test_dir`) and back out (`cd ..`) to practice moving around.

4.2 4.2 Creating and Managing Files

1. **Create Empty Files**
   Commands:

```
touch file1.txt
touch file2.txt
```

   Verify they are created with `ls`.

2. **Write Simple Content**
   Command:

```
echo "Hello, Unix Lab!" > file1.txt
```

   Check the contents of `file1.txt` using:

```
cat file1.txt
```

3. **Copy and Rename**
   Copy `file1.txt` to a new file named `file3.txt`:

```
cp file1.txt file3.txt
```

   Rename `file2.txt` to `file2-renamed.txt`:

```
mv file2.txt file2-renamed.txt
```

   Use `ls` to confirm changes.

4. **Remove Files**
   Command:

```
rm file3.txt
```

   Confirm the file was removed.

4.3 4.3 Searching and Filtering Text

1. **Create a Multi-Line File**
   Command:

```
cat > colors.txt
```

   Then type multiple color names (e.g., red, blue, green, yellow) on separate lines. Press **Ctrl+D** to save and exit.

2. **Find Specific Lines**
   Use `grep` to search for "red":

```
grep red colors.txt
```

   Search for multiple patterns by piping `grep`:

```
grep -E "red|blue" colors.txt
```

   Experiment with `grep -n` to show line numbers.

3. **Count Lines or Words**
   Use `wc` to see the number of lines, words, and characters:

```
wc colors.txt
```

   Combine `grep` and `wc`:

```
grep "green" colors.txt | wc -l
```

   This counts how many lines contain "green."

4.4 4.4 Redirection and Piping

1. **Basic Redirection**
   Write a list of items to a file using redirection (`>`):

```
echo "apple" > fruits.txt
echo "banana" >> fruits.txt
echo "cherry" >> fruits.txt
```

   Check contents with:

```
cat fruits.txt
```

2. **Combining Commands with Pipes**
   Sort the file alphabetically and display its content:

```
cat fruits.txt | sort
```

Count lines after sorting:

```
cat fruits.txt | sort | wc -l
```

Observe how each command's output becomes the next command's input.

4.5 4.5 File Permissions

1. **Check Permissions**
   Run `ls -l` in your `unix_lab` directory to see permissions for each file. Note the format: `rwxr-xr-x`, etc.

2. **Change Permissions**
   Command:

```
chmod 600 file1.txt
```

   Now only the owner can read/write, and no one else can do anything with `file1.txt`.

3. **Restore Permissions**
   Command:

```
chmod 644 file1.txt
```

   This gives the owner read/write and everyone else read-only access.

4.6 4.6 Environment Variables

1. **Check Existing Variables**
   Command:

```
echo $HOME
echo $PATH
```

   Observe how the shell responds.

2. **Set Your Own Variable**
   Commands:

```
MYVAR="HelloUnix"
echo $MYVAR
```

   Note that `MYVAR` is only set in the current shell session.

3. **Export Variable**
   Make the variable available to sub-processes:

```
export MYVAR
```

   Launch a new subshell (`bash`), and echo `$MYVAR` again to see if it persists.

4.7 4.7 Simple Shell Scripting

1. **Create a Script**
   Command:

```
cat > script.sh
```

   Then type:

```
#!/bin/bash
echo "This is a simple script."
pwd
ls
```

   Press **Ctrl+D** to save.

2. **Make the Script Executable**
   Command:

```
chmod +x script.sh
```

3. **Run the Script**
   Command:

```
./script.sh
```

   Observe the output: a printed message, your current directory, and a file listing.

5 Challenges

1. **Directory Organization**
   Create a folder named `practice`. Inside it, create three subdirectories: `A`, `B`, `C`. Move at least one file into each subdirectory. Confirm the structure using `tree` (if installed) or `ls -R`.

2. **Data Inspection**
   Create a `data.txt` file with at least 10 lines of text. Use `head -3 data.txt` to see the first three lines, and `tail -3 data.txt` to see the last three lines. Filter lines containing a specific word (e.g., "error" or "info") and write them to `filtered.txt`.

3. **Permission Practice**
   Change permissions of a directory so that others can list files (read permission) but cannot modify them (no write permission). Try creating a file in that directory as a different user (if possible) or simulate it to see the permission denial.

4. **Script Enhancement**
   Modify `script.sh` to accept a command-line argument. For instance:

```
#!/bin/bash
echo "Argument 1 is $1"
echo "Argument 2 is $2"
```

   Then run:

```
./script.sh hello world
```

   Observe the output.

6 Lab Submission and Cleanup

1. **Take Screenshots or Save Logs:** Document key command outputs (e.g., file lists, environment variables). If required, compress the folder containing your practice files and submit it.

2. **Cleanup (Optional):** Remove any temporary files or directories you created if you no longer need them:

```
rm -r unix_lab
```

   Make sure not to remove anything important accidentally.

7 Summary

Through these exercises, you have practiced:

- Navigating directories (`pwd`, `cd`, `ls`).
- Creating, editing, and removing files (`touch`, `echo`, `cat`, `rm`).
- Using redirection (`>`, `>>`) and pipelines (`|`) for powerful command chaining.
- Inspecting file content with commands like `grep`, `wc`, `head`, and `tail`.
- Manipulating file permissions (`chmod`).
- Managing environment variables (`export`) and writing a basic script.

These skills form the foundation of working in a Unix-like terminal environment. Continue exploring more advanced topics such as process management (`ps`, `kill`), networking tools (`ping`, `curl`), and shell scripting to automate tasks.