Linux terminal

Introdução Engenharia Informática

Mário Antunes

September 22, 2025

Exercise 1: Finding Your Way Around 🧭

This exercise covers `pwd`, `ls`, `cd`, and basic information commands.

1. Open your terminal. Verify your starting location (your home directory) by printing the working directory.

   ```
   $ pwd
   ```

2. List the contents of your home directory. Then, list them again showing **all** files in the **long** list format.

   ```
   $ ls
   $ ls -la
   ```

3. Navigate to the system log directory at `/var/log` and list its contents.

   ```
   $ cd /var/log
   $ ls
   ```

4. Get some information: find out your username and the current date.

   ```
   $ whoami
   $ date
   ```

5. Return to your home directory using the quickest shortcut.

   ```
   $ cd ~
   ```

Exercise 2: Creating and Managing Files 📂

In this exercise, you'll create, copy, move, and delete files and directories.

1. From your home directory, create a new directory called `TIA`.

```
$ mkdir TIA
```

2. Navigate inside your new `TIA` directory.

```
$ cd TIA
```

3. Create an empty file called `notes.txt`.

```
$ touch notes.txt
```

4. Add text to your file and then view its contents.

```
$ echo "My first line of text." > notes.txt
$ cat notes.txt
```

5. Make a copy of your file named `notes_backup.txt`.

```
$ cp notes.txt notes_backup.txt
```

6. Rename `notes.txt` to `important_notes.txt`.

```
$ mv notes.txt important_notes.txt
```

7. Clean up by deleting the backup file.

```
$ rm notes_backup.txt
```

Exercise 3: Understanding Permissions 🔐

This exercise focuses on reading and changing file permissions with `chmod`.

1. Inside your `~/TIA` directory, create a new file called `secret_data.txt`.

```
$ touch secret_data.txt
```

2. View the file's default permissions.

```
$ ls -l secret_data.txt
```

3. Remove all permissions for everyone.

```
$ chmod 000 secret_data.txt
```

4. Try to view the file's contents. You should get a **"Permission denied"** error.

```
$ cat secret_data.txt
```

5. Restore read and write permission for **only yourself**.

```
$ chmod u+rw secret_data.txt
```

6. Create an empty script file `my_script.sh` and make it executable for yourself. Check the permissions afterward to see the change.

   ```
   $ touch my_script.sh
   $ chmod u+x my_script.sh
   $ ls -l my_script.sh
   ```

Exercise 4: Managing Software with APT 📦

Let’s install and remove a program using the **APT** package manager.

1. First, synchronize your system’s package list with the software repositories.

   ```
   $ sudo apt update
   ```

2. Search for a useful command-line tool called `htop`.

   ```
   $ apt search htop
   ```

3. Now, install `htop`. You will need to confirm the installation when prompted.

   ```
   $ sudo apt install htop
   ```

4. Run the program you just installed. Press `q` to quit.

   ```
   $ htop
   ```

5. Finally, clean up by removing the package from your system.

   ```
   $ sudo apt remove htop
   ```

Exercise 5: Combining Commands 🔗

Let’s explore the power of the **pipe (|)** and **redirection (>>)**.

1. The command `ps aux` lists all running processes. Use the pipe (|) to send this output to `grep` to find your own “bash” process.

   ```
   $ ps aux | grep "bash"
   ```

2. Create a log file with one entry.

   ```
   $ echo "$(date): Starting my work." > ~/TIA/activity.log
   ```

3. Use the append operator (>>) to add a second line to the file without deleting the first one.

```
$ echo "$(date): Finished exercise 5." >> ~/TIA/activity.log
```

4. Verify that your log file contains both lines.

```
$ cat ~/TIA/activity.log
```

Exercise 6: Customizing Your Environment ✨

Time to edit your `.bashrc` file to create a handy shortcut (an alias).

1. Open your `~/.bashrc` file using the `nano` editor.

```
$ nano ~/.bashrc
```

2. Scroll to the very bottom and add the following line to create a shortcut `ll` for the command `ls -alF`.

```
alias ll='ls -alF'
```

3. Save the file and exit `nano` (`Ctrl+X`, then `Y`, then `Enter`).
4. Load the changes into your current session.

```
$ source ~/.bashrc
```

5. Test your new alias.

```
$ ll
```

Exercise 7: Scripting Challenge 🚀

Let's create a script that automates creating a project structure.

1. Create and open a new file named `setup_project.sh` in your `~/TIA` directory. Add the following code, then save and close the file.

```
#!/bin/bash
PROJECT_DIR="$HOME/TIA/my_project"

if [ -d "$PROJECT_DIR" ]; then
  echo "Error: Directory '$PROJECT_DIR' already exists."
  exit 1
fi

mkdir "$PROJECT_DIR"
echo "Directory '$PROJECT_DIR' created."
```

```
for folder in assets source docs
do
  mkdir "$PROJECT_DIR/$folder"
  echo "-> Created subfolder: $folder"
done

echo "Project setup complete!"
```

2. Make the script executable and then run it.

```
$ chmod +x ~/TIA/setup_project.sh
$ ~/TIA/setup_project.sh
```

3. Verify that the directory and its subdirectories were created.

```
$ ls -R ~/TIA/my_project
```

Exercise 8: Scheduling a Task with `cron` 🕒

Let's create a simple script and schedule it to run automatically every minute.

1. **Create the Script:** In your `~/TIA` directory, create a script named `log_time.sh` with the following content.

```
#!/bin/bash
date >> $HOME/TIA/cron_log.txt
```

2. **Make it Executable:**

```
$ chmod +x ~/TIA/log_time.sh
```

3. **Open your Crontab:** This will open a text editor.

```
$ crontab -e
```

4. **Add the Cron Job:** Go to the bottom of the file and add the following line. You must use the full, absolute path to your script.

```
* * * * * /home/student/TIA/log_time.sh
```

5. **Save and Verify:** Save and exit the editor. Wait two minutes, then check your log file. You should see two timestamp entries.

```
$ cat ~/TIA/cron_log.txt
```

6. **Clean Up:** It's very important to remove the cron job so it doesn't run forever. This command removes your entire crontab file.

```
$ crontab -r
```