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 with `pwd`.
2. List the contents of your home directory. Then, list them again showing **all** files in the **long** list format using `ls -la`.
3. Navigate to the system log directory at `/var/log` and list its contents.
4. Get some information: find out your username with `whoami` and the current date with `date`.
5. Return to your home directory using the `cd ~` shortcut.

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` using `mkdir`.
2. Navigate inside your new `TIA` directory.
3. Create an empty file called `notes.txt` using the `touch` command.
4. Add text to your file using `echo "My first line of text." > notes.txt`. View its contents with `cat notes.txt`.
5. Make a copy of your file named `notes_backup.txt` using `cp`.
6. Rename `notes.txt` to `important_notes.txt` using the `mv` command.
7. Clean up by deleting the backup file with `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`.
2. View the file's default permissions using `ls -l`.
3. Remove all permissions for everyone with `chmod 000 secret_data.txt`.
4. Try to view the file's contents with `cat`. You should get a **"Permission denied"** error.
5. Restore read and write permission for **only yourself** (`u+rw`).
6. Create an empty script file `my_script.sh` and make it executable for yourself using `chmod u+x my_script.sh`. Check the permissions with `ls -l` to see the `x` has been added.

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. This is a crucial first step. `bash $ sudo apt update`
2. Search for a useful command-line tool called `htop`, an interactive process viewer. `bash $ apt search htop`
3. Now, install `htop` using `apt install`. You will need `sudo` for this. `bash $ sudo apt install htop` Confirm the installation when prompted.
4. Run the program you just installed! `bash $ htop` Explore the interface for a moment. You can press `q` to quit.
5. Finally, clean up by removing the package from your system. `bash $ 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. `bash $ ps aux | grep "bash"`
2. Create a log file. Use `echo` and `>` to add the first entry to `~/TIA/activity.log`. `bash $ 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. `bash $ echo "$(date): Finished exercise 5." >> ~/TIA/activity.log`
4. Verify that your log file contains both lines using `cat`.

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.
2. Scroll to the very bottom and add the following line to create a shortcut `ll` for the command `ls -alF`. bash `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 by running `source ~/.bashrc`.
5. Test your new alias by typing `ll` and pressing `Enter`.

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.

2. Add the following code. This script checks if a directory already exists and then uses a `for` loop to create subdirectories.

```bash
#!/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!"
```

3. Make the script executable and run it.

4. Use `ls -R ~/TIA/my_project` to verify that the directory and its subdirectories were created.

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 new script named `log_time.sh`. Its only job is to append the current date and time to a log file. `bash #!/bin/bash date >> $HOME/TIA/cron_log.txt`
2. **Make it Executable:** `bash $ chmod +x ~/TIA/log_time.sh`
3. **Open your Crontab:** Use the command `crontab -e`. If it's your first time, you may be asked to choose a text editor (select `nano`).
4. **Add the Cron Job:** Go to the bottom of the file and add the following line. The five asterisks mean "run every minute of every hour of every day..." **It is critical to use the full, absolute path to your script.** `cron * * * * * /home/student/TIA/log_time.sh`
5. **Save and Verify:** Save and exit the editor. Now, wait two minutes. Check your log file. You should see two timestamp entries. `bash $ cat ~/TIA/cron_log.txt`
6. **Clean Up:** It's very important to remove the cron job so it doesn't run forever. Use `crontab -r` to remove your entire crontab file. `bash $ crontab -r`