A Beginner's Guide to Linux and Terminal Navigation

This guide was inspired by the helpful tutorials and resources from the cloud community at abcofcloudcomputing.com.

What is Linux and the Command Line?

Think of your computer as a house. On most computers, you use a mouse to click on folders and open files. This is like looking at a physical map of the house and pointing at a room.

In Linux, we often use the **Terminal** (the black window shown in the screenshots). Instead of clicking with a mouse, you type text commands to talk directly to the computer. It is like telling the computer: "Go to the kitchen, open the drawer, and get the spoon."

Here are a few key terms before we begin:

- **Directory**: This is just another word for a folder.
- **Path**: The address or route to a folder or file.
- **Root (/)**: The ultimate parent folder that contains everything on the computer. It is like the front door of your house.
- **Home (~)**: Your personal folder where your files are kept. For the administrator user (called root), this folder is located at /root.

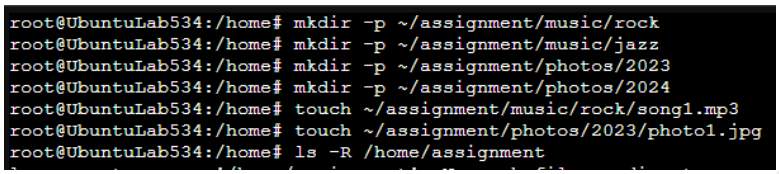
Part 1: Setting Up Your Environment (The Setup)

The first screenshot shows a series of commands used to build our folders and create dummy files.

The Commands Used:

- `mkdir -p ~/assignment/music/rock`

    - `mkdir` stands for "make directory" (create a folder).
    - `-p` is a special option that tells the computer to create any parent folders that do not exist yet. For example, it creates `assignment`, then a folder named `music` inside it, and finally a folder named `rock` inside that.
- `touch ~/assignment/music/rock/song1.mp3`
    - `touch` is used to create a blank, empty file. Here, we created an empty music file named `song1.mp3` inside the rock folder.
- `ls -R /home/assignment`
    - `ls` stands for "list" (show what is inside a folder).
    - `-R` stands for "recursive." This tells the computer to list not just the main folder, but also every single folder and file hidden deep inside it.



What the Second Screenshot Means:

The second screenshot shows the results of running `ls -R /root/assignment`. It displays a complete directory tree:

- The main folder is `assignment`.
- Inside `assignment`, there are two folders: `music` and `photos`.
- Inside `music`, there are folders named `jazz` and `rock`.
- Inside `rock`, there is the file we created: `song1.mp3`.
- Inside `photos`, there are folders named `2023` and `2024`.
- Inside `2023`, there is a file named `photo1.jpg`.

```
root@UbuntuLab534:/home# ls /root
assignment
root@UbuntuLab534:/home# ls -R /root/assignment
/root/assignment:
music  photos

/root/assignment/music:
jazz  rock

/root/assignment/music/jazz:

/root/assignment/music/rock:
song1.mp3

/root/assignment/photos:
2023  2024

/root/assignment/photos/2023:
photo1.jpg

/root/assignment/photos/2024:
root@UbuntuLab534:/home#
```

Part 2: Task 1: Absolute Navigation

Navigating your computer requires telling it where you want to go. There are two ways to do this: using an **Absolute Path** or a **Relative Path**.

An **Absolute Path** is a full, complete address that starts from the very beginning of the computer (the root `/`). It is like writing a full postal address: Country, City, Street, House Number. No matter where you are in the house, that address always points to the exact same place.

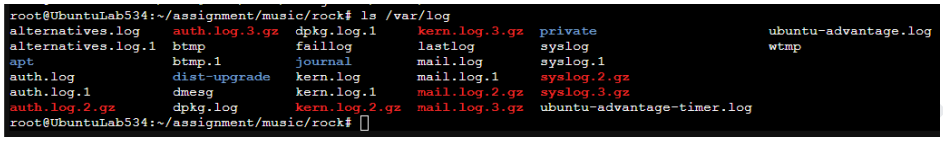
Q1. Moving to the "rock" folder

- **Command**: `cd /root/assignment/music/rock`
- **What it does**: `cd` stands for "change directory" (move to a folder). By typing the full absolute path starting with `/root`, you tell the computer to jump straight to the `rock` folder, regardless of where you currently are.

```
root@UbuntuLab534:/home# cd /root/assignment/music/rock
root@UbuntuLab534:~/assignment/music/rock#
```

Q2. Listing a distant folder

- **Command**: `ls /var/log`
- **What it does**: You do not have to move into a folder to see what is inside it. This command tells the computer to list the contents of the `/var/log` system folder (which contains computer activity logs) while you remain safely inside the `rock` folder.



Task 2: Relative Navigation

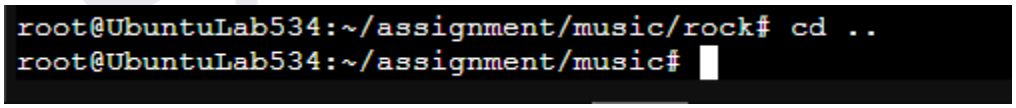
A **Relative Path** is an address that starts from where you are currently standing. It is like telling someone in your house: "Go up the stairs, then look in the first door on the right." If you are in a different room, those instructions will lead to a completely different place.

In relative navigation, we use special symbols:

- `.` (a single dot) means "the folder I am currently in."
- `..` (two dots) means "the folder directly above me" (the parent folder).

Q3. Moving up one level

- **Command**: `cd ..`
- **What it does**: Since we were inside the `rock` folder, going up one level takes us to its parent folder, which is `music`.



Q4. Moving to a sibling folder

- **Command**: `cd jazz`
- **What it does**: Since we are now in the `music` folder, and `jazz` is right there inside `music`, we can just type its name to enter it.

```
root@UbuntuLab534:~/assignment/music# cd jazz
root@UbuntuLab534:~/assignment/music/jazz#
```

Q5. Jumping across folders

- **Command**: `cd ../../photos/2023`
- **What it does**: We wanted to go from `music/jazz` to `photos/2023`.
    - The first `..` takes us up to `music`.
    - The second `..` takes us up to `assignment`.
    - From `assignment`, we go down into `photos`, and then into `2023`.

```
root@UbuntuLab534:~/assignment/music# cd jazz
root@UbuntuLab534:~/assignment/music/jazz# cd ../../photos/2023
root@UbuntuLab534:~/assignment/photos/2023#
```

Task 3: Basic File Operations

In this task, we assume we are starting inside the `~/assignment` folder.

Q6. Copying a file

- **First Command (Failed)**: `cp photos/2023/photo1.jpg`
    - **Why it failed**: `cp` stands for "copy." To copy something, you must tell the computer two things: what you want to copy (the source) and where you want to put it (the destination). Here, the destination was missing.
- **Second Command (Successful)**: `cp photos/2023/photo1.jpg ./photo1.jpg`
    - **What it does**: This tells the computer to copy the file `photo1.jpg` from the `photos/2023` folder and paste it into the current folder (represented by `./`) with the same name.

```
root@UbuntuLab534:~/assignment# cp photos/2023/photo1.jpg
cp: missing destination file operand after 'photos/2023/photo1.jpg'
Try 'cp --help' for more information.
root@UbuntuLab534:~/assignment# cp photos/2023/photo1.jpg ./photo1.jpg
music  photo1.jpg  photos
root@UbuntuLab534:~/assignment#
```

Q7. Removing a file

- **Command**: `rm /root/assignment/music/rock/song1.mp3`
- **What it does**: `rm` stands for "remove" (delete). We used the absolute path to tell the computer to delete the `song1.mp3` file. Be careful, because Linux does not have a trash can; once you delete a file with `rm`, it is gone forever.

```
root@UbuntuLab534:~# rm /root/assignment/music/rock/song1.mp3
root@UbuntuLab534:~#
```

By typing `ls -R /root/assignment` at the end of the assignment, we can confirm that the `song1.mp3` file has been successfully removed, while our newly copied `photo1.jpg` is safe in the main folder.

```
root@UbuntuLab534:~# ls -R /root/assignment
/root/assignment:
music  photo1.jpg  photos

/root/assignment/music:
jazz  rock

/root/assignment/music/jazz:

/root/assignment/music/rock:

/root/assignment/photos:
2023  2024

/root/assignment/photos/2023:
photo1.jpg

/root/assignment/photos/2024:
root@UbuntuLab534:~#
```