Day2_Materials_EN

1. General Info

Date: 20.08.2025

Topic: Working with Files & Permissions.

Daily goal: To reinforce basic Linux commands, learn to work with files and folders, study file permissions, and prepare the first automation script.

---

2. Learned Material

Command	Description
`ls`	Lists directory contents.
`cd`	Changes working directory, one level up.
`pwd`	Shows the absolute path to the current directory.
`tree`	Shows directory tree.
`stat`	Displays size, permissions, and modification dates of a file.

---

3. Review + New Commands

Today we add:

- `ls` (`ls -la`) — Lists directory contents.

```
leprecha@Ubuntu-DevOps:~$ ls -la
-rw-------  1 leprecha sysadmin 8217 Aug 20 17:07 .bash_history
```

Lists directory contents.

1.) `l` — long format.

2.) `a` — include hidden files.

- `cd` — change directory.

```
leprecha@Ubuntu-DevOps:~$ cd /etc
leprecha@Ubuntu-DevOps:/etc$
```

- `pwd` — print working directory.

```
leprecha@Ubuntu-DevOps:~$ pwd
/home/leprecha
```

- `tree` — display directory structure.

`tree /etc | head -n 5`

```
leprecha@Ubuntu-DevOps:~$ tree /etc | head -n 5
/etc
├── adduser.conf
├── alsa
│   └── conf.d
│       ├── 50-pipewire.conf → /usr/share/alsa/alsa.conf.d/50-pipewire.conf
```

Shows directory tree (first 5 lines).

- `stat` — detailed file information.

`stat /etc/passwd`

```
leprecha@Ubuntu-DevOps:~$ stat /etc/passwd
  File: /etc/passwd
  Size: 2959        Blocks: 8          IO Block: 4096   regular file
Device: 259,2 Inode: 5507826     Links: 1
```

```
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2025-08-20 16:58:14.464296261 +0100
Modify: 2025-08-19 15:51:08.037927698 +0100
Change: 2025-08-19 15:51:08.038485928 +0100
 Birth: 2025-08-19 15:51:08.037485933 +0100
```

---

4. Working with nano

Nano editor — a simple console text editor in Linux. Suitable for creating and editing configuration and text files.

`nano filename.txt` — Opens the file if it exists, creates a new one if not.

Keys	Action
Ctrl + O	Save file
Ctrl + X	Exit nano
Ctrl + G	Show help
Ctrl + W	Search text
Ctrl + K	Cut current line
Ctrl + U	Paste cut text
Ctrl + C	Show cursor position
Ctrl + _	Go to line/column
Alt + ,	Switch to previous file
Alt + .	Switch to next file

Practice

1. Create a file `filename.txt` in your home directory.
2. Write 2–3 sentences about yourself in English.
3. Save changes and exit nano.

4. Copy the file to `/tmp`.

5. Check the content of the copied file using `cat`.

```
leprecha@Ubuntu-DevOps:~$ nano filename.txt
leprecha@Ubuntu-DevOps:~$ cp filename.txt /tmp/filename.txt
leprecha@Ubuntu-DevOps:~$ cat /tmp/filename.txt
How are you today? I'm fine.
```

5. Copying, moving, deleting

Commands allow you to manage files and folders — copy, move, rename, and delete.

Commands	Description
`cp file.txt backup.txt`	Copies `file.txt` to `backup.txt`
`cp file.txt /home/user/`	Copies file to the specified folder
`cp -r myfolder /home/user/`	Recursively copies a folder with all contents
`mv file.txt /home/user/`	Moves file to the specified folder
`mv oldname.txt newname.txt`	Renames a file
`mv myfolder /home/user/`	Moves a folder to the specified location
`rm file.txt`	Deletes a file
`rm file1.txt file2.txt`	Deletes multiple files
`rm -r myfolder`	Recursively deletes a folder and its contents
`rm -rf myfolder`	Deletes folder without confirmation (dangerous!)

1). Copying

In Linux, the `cp` command is used for copying.

```
leprecha@Ubuntu-DevOps:~$ cp filename.txt backup.txt
leprecha@Ubuntu-DevOps:~$ cat backup.txt
```

```
How are you today? I'm fine.
```

Copy a file to a directory:

```
leprecha@Ubuntu-DevOps:~$ tree /home/leprecha/Documents/
/home/leprecha/Documents/
└── filename.txt
```

Copy a directory with all its contents:

(`-r` or `--recursive` — recursively, required for directories).

```
leprecha@Ubuntu-DevOps:~$ cp -r Folder /home/leprecha/Documents/
leprecha@Ubuntu-DevOps:~$ tree /home/leprecha/Documents/
/home/leprecha/Documents/
├── filename.txt
└── Folder
```

---

2). Moving

The `mv` command is used for moving (and renaming).

```
leprecha@Ubuntu-DevOps:~$ mv filename.txt /home/leprecha/Documents/Fo
lder/
leprecha@Ubuntu-DevOps:~$ tree /home/leprecha/Documents/
/home/leprecha/Documents/
├── filename.txt
└── Folder
    └── filename.txt

2 directories, 2 files
leprecha@Ubuntu-DevOps:~$
```

Renaming:

```
leprecha@Ubuntu-DevOps:~$ mv filename.txt newfile.txt
leprecha@Ubuntu-DevOps:~$ ls
backup.txt  DevOps     Downloads  Music      Pictures  snap      Videos
Desktop     Documents  Folder     newfile.txt  Public    Templates
```

Moving folder:

```
leprecha@Ubuntu-DevOps:~$ mv Folder /home/leprecha/Music/
leprecha@Ubuntu-DevOps:~$ ls /home/leprecha/Music/
Folder
```

---

3). Removing

The `rm` command is used for deleting.

Delete a file:

```
leprecha@Ubuntu-DevOps:~$ rm newfile.txt
```

Delete multiple files:

```
leprecha@Ubuntu-DevOps:~$ rm backup.txt file.txt
```

Delete a directory and everything inside:

```
leprecha@Ubuntu-DevOps:~$ rm -r /home/leprecha/Music/Folder/
```

Practice

1. Create a folder `lab2_files` in your home directory.
2. Create three files in it: `file1.txt`, `file2.txt`, `file3.txt`.
3. Copy `file1.txt` to `/tmp`.
4. Move `file2.txt` to `/tmp` and rename it to `file2_moved.txt`.
5. Delete `file3.txt`.
6. Remove the `lab2_files`.

```
leprecha@Ubuntu-DevOps:~$ mkdir ~/lab2_files
leprecha@Ubuntu-DevOps:~$ touch ~/lab2_files/file1.txt ~/lab2_files/file2.txt
~/lab2_files/file3.txt
leprecha@Ubuntu-DevOps:~$ cp ~/lab2_files/file1.txt /tmp
leprecha@Ubuntu-DevOps:~$ mv ~/lab2_files/file2.txt /tmp/file2_moved.txt
leprecha@Ubuntu-DevOps:~$ rm ~/lab2_files/file3.txt
leprecha@Ubuntu-DevOps:~$ rm -r ~/lab2_files
leprecha@Ubuntu-DevOps:~$ ls
Desktop  Documents  Music    Public  Templates
DevOps   Downloads  Pictures  snap   Videos
leprecha@Ubuntu-DevOps:~$ ls /tmp
file1.txt
file2_moved.txt
```

---

6. Permissions

In Linux, permissions define who can do what with a file or directory.

1). What are file permissions.

In Linux, every file and directory has three permission groups:

1. **Owner** (user, `u`) — the user who owns the file.
2. **Group** (group, `g`) — users who belong to the same group as the owner.
3. **Others** (others, `o`) — all other users.

Each group can have three types of permissions:

- **r** (read) — permission to read the file.
- **w** (write) — permission to modify the file.
- **x** (execute) — permission to run the file as a program.

Permission format in `ls -l` .

```
leprecha@Ubuntu-DevOps:~$ ls -l
drwxr-xr-x 2 leprecha sysadmin 4096 Aug 20 17:23 Desktop
drwx------ 6 leprecha sysadmin 4096 Aug 19 21:00 snap
```

- First character: type (`-` — file, `d` — directory).

- Then three groups of 3 characters: **owner**, **group**, **others**.

- `r` — read, `w` — write, `x` — execute, `-` — no permission.

```
-rw-rw-r-- 1 leprecha sysadmin  492 Aug 20 19:31 learnlinux.spec
```

Breakdown

- The first character indicates the object type (`-` = file, `d` = directory).
- The following characters represent permissions:
  - `rw-` — owner (**read**, **write**, no execute).
  - `rw-` — group (**read**, **write**, no execute).
  - `r--` — others (**read**, no write, no execute).

---

2). How to change permissions

In Linux, each file and directory permissions can be changed using the `chmod` command.

Examples:

- `chmod u+x file` — add execute permission for the owner.

- `chmod g-w file` — remove write permission for the group.
- `chmod o+r file` — add read permission for others.
- `chmod 755 file` — set permissions using numeric (octal) notation.

```
leprecha@Ubuntu-DevOps:~$ chmod u+x learnlinux.spec
leprecha@Ubuntu-DevOps:~$ ls -l
-rwxr--r-- 1 leprecha sysadmin    0 Aug 20 19:09 learnlinux.spec
```

`rwx` — owner (**read**, **write**, **execute**).

In numeric form

Permissions are represented by numbers:

- r = 4
- w = 2
- x = 1

Summing up:

- `rwx` = 4+2+1 = **7**
- `rw-` = 4+2+0 = **6**
- `r-x` = 4+0+1 = **5**

Example: `chmod 555`

```
leprecha@Ubuntu-DevOps:~$ chmod 555 learnlinux.spec
leprecha@Ubuntu-DevOps:~$ ls -l
-r-xr-xr-x 1 leprecha sysadmin    0 Aug 20 19:09 learnlinux.spec
```

Changing permissions

Commands	Description
chmod 755 file	rwx for owner, rx for group & others

Commands	Description
chmod u+x file	Add execute to owner
chmod g-w file	Remove write from group
chmod o-r file	Remove read from others

How to change the owner

Change the file owner to the user `helpme` using `sudo chown`:

```
leprecha@Ubuntu-DevOps:~$ sudo chown helpme learnlinux.spec
leprecha@Ubuntu-DevOps:~$ ls -l learnlinux.spec
-r-xr-xr-x 1 helpme sysadmin 0 Aug 20 19:09 learnlinux.spec
```

Changing owner and group

Commands	Description
chown user file	Change file owner
chgrp group file	Change file group
chown user:group file	Change owner and group

Practice

1. Create a file `test_permissions.txt`.
2. Check permissions.
3. Give owner full access, group — read only, others — no access.
4. Check permissions.
5. Change the file owner.

```
leprecha@Ubuntu-DevOps:~$ touch test_permissions.txt
leprecha@Ubuntu-DevOps:~$ ls -l test_permissions.txt
-rw-r--r-- 1 leprecha sysadmin 0 Aug 20 19:25 test_permissions.txt
leprecha@Ubuntu-DevOps:~$ chmod 740 test_permissions.txt
leprecha@Ubuntu-DevOps:~$ sudo chown helpme test_permissions.txt
```

```
leprecha@Ubuntu-DevOps:~$ ls -l test_permissions.txt
-rwxr----- 1 helpme sysadmin 0 Aug 20 19:25 test_permissions.txt
```