

Day1_Materials_EN

1. General Info

Date: 13.08.2025.

Topic: Environment Setup.

Daily goal: Prepare environment for Linux learning.

2. Learned Material

Commands and what they do.

1. | `pwd` — Show path to current directory. |

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

`pwd` In Linux, this is a command that shows the current working directory.
(*print working directory*)

2. | `ls -la` — List all files (including hidden) with details. |

```
leprecha@Ubuntu-DevOps:~$ ls -la
total 88
-rw----- 1 leprecha sysadmin 2791 Aug 19 15:54 .bash_history
drwxr-xr-x 2 leprecha sysadmin 4096 Aug 19 15:19 Desktop
```

`ls -la` — This is a command for detailed viewing of a folder's contents in Linux.

- `rw - - - - -` — access permissions (d = directory, - = file).

1. 1 — Number of hard link.

2. leprecha — file owner.

3. sysadmin — group.
4. 2791 — size in bytes.
5. Aug 19 16:04 — date of last modification.
6. .bash_history — file or directory name.

3. | `cd /etc` — Change to `/etc` directory.

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

4. | `mkdir` demo — create a directory named demo. |

- `mkdir` — *make directory*.
- `demo` — name of the new folder.
- `ls -la` — check that the folder has appeared.

```
drwxrwxr-x  2 leprecha sysadmin 4096 Aug 19 16:32 demo
```

5. | `touch` demo/file.txt — creates file file.txt in demo. |

```
leprecha@Ubuntu-DevOps:~$ touch demo/file.txt
```

- `touch` — creates an empty file if it does not exist.
- `demo/file.txt` — the path where this file will be created.

```
leprecha@Ubuntu-DevOps:~$ ls -la demo
total 8
drwxrwxr-x  2 leprecha sysadmin 4096 Aug 19 16:34 .
```

```
drwxr-x--- 18 leprecha sysadmin 4096 Aug 19 16:32 ..
-rw-rw-r-- 1 leprecha sysadmin  0 Aug 19 16:34 file.txt
```

6. | `cp` demo/file.txt demo/file.bak — making a **copy** of the file **file.txt** with the name **file.bak** in the same **demo** folder.

```
leprecha@Ubuntu-DevOps:~$ cp demo/file.txt demo/file.bak
leprecha@Ubuntu-DevOps:~$ ls -la demo
total 8
-rw-r--r-- 1 leprecha sysadmin  0 Aug 19 16:46 file.bak
-rw-rw-r-- 1 leprecha sysadmin  0 Aug 19 16:34 file.txt
```

- `cp` — *copy*.
- The first argument is what we copy.
- The second is where we copy it to.

7. | `mv` demo/file.bak demo/file.old — rename a file **file.bak** in **file.old** in folder **demo** . |

```
leprecha@Ubuntu-DevOps:~$ mv demo/file.bak demo/file.old
leprecha@Ubuntu-DevOps:~$ ls -la demo
total 8
drwxrwxr-x 2 leprecha sysadmin 4096 Aug 19 16:54 .
drwxr-x--- 18 leprecha sysadmin 4096 Aug 19 16:32 ..
-rw-r--r-- 1 leprecha sysadmin  0 Aug 19 16:46 file.old
-rw-rw-r-- 1 leprecha sysadmin  0 Aug 19 16:34 file.txt
```

- `mv` — *move*, but if the path stays the same, it's just a rename.
- The first argument is what we move/rename.
- The second is the new name or path.

8. | `rm` demo/file.old — remove file **file.old** from **demo** . |

```
leprecha@Ubuntu-DevOps:~$ rm demo/file.old
leprecha@Ubuntu-DevOps:~$ ls -la demo
total 8
drwxrwxr-x  2 leprecha sysadmin 4096 Aug 19 16:58 .
drwxr-x--- 18 leprecha sysadmin 4096 Aug 19 16:32 ..
-rw-rw-r--  1 leprecha sysadmin   0 Aug 19 16:34 file.txt
```

- `rm` — removes a file.
- `rm -r` — removes a directory and everything inside it.
- `rm -ri` — (`-i` = interactive → asks before deleting each file).

9. | `man ls` — manual page for `ls`. |

10. | `whoami` — shows which user I am currently logged in as in the system. |

```
leprecha@Ubuntu-DevOps:~$ whoami
leprecha
```

11. | `hostname` — shows the `hostname` — that is, the name of the computer (node) in network. |

```
leprecha@Ubuntu-DevOps:~$ hostname
Ubuntu-DevOps
```

12. | `date` — shows the current date and time in the system. |

```
leprecha@Ubuntu-DevOps:~$ date
Tue Aug 19 09:04:25 PM IST 2025
```

13. | `clear` — clears the terminal screen, removing all previous output. |

14. | `uname -a` — показывает подробную информацию о системе и ядре Linux. |

```
leprecha@Ubuntu-DevOps:~$ uname -a
Linux Ubuntu-DevOps 6.14.0-28-generic #28~24.04.1-Ubuntu SMP PREEMPT_
DYNAMIC Fri Jul 25 10:47:01 UTC 2 ×86_64 ×86_64 ×86_64 GNU/Linux
```

15. | `exit` — closes the current terminal session. |

Command	Description (EN)
<code>pwd</code>	Show path to current directory
<code>ls -la</code>	List all files (including hidden) with details
<code>cd /etc</code>	Change to <code>/etc</code> directory
<code>mkdir demo</code>	Create <code>demo</code> directory
<code>touch demo/file.txt</code>	Create empty file <code>file.txt</code> in <code>demo</code>
<code>cp demo/file.txt demo/file.bak</code>	Copy file with new name <code>file.bak</code>
<code>mv demo/file.bak demo/file.old</code>	Rename file
<code>rm demo/file.old</code>	Remove file
<code>man ls</code>	Open manual page for <code>ls</code>
<code>whoami</code>	Show current username
<code>hostname</code>	Show system hostname
<code>date</code>	Show current date and time
<code>clear</code>	Clear terminal screen
<code>uname -a</code>	Show system and kernel info
<code>exit</code>	Exit terminal or session

Working with nano and the Filesystem

- Create `hello.txt` and edit in nano.

```
leprecha@Ubuntu-DevOps:~$ mkdir practice
leprecha@Ubuntu-DevOps:~$ cd practice
leprecha@Ubuntu-DevOps:~/practice$ nano hello.txt
leprecha@Ubuntu-DevOps:~/practice$ cat hello.txt
Hello world!
```

1. Create a test folder — **mkdir** practice.
2. Go to the folder **cd** practice.
3. Create and open the file **hello.txt**, then write a greeting, save with **Ctrl+O**, and close with **Ctrl+X**.
4. Check the contents of the file using **cat** hello.txt.

Practice copying, renaming, and deleting files.

Copying (**cp**).

```
leprecha@Ubuntu-DevOps:~/practice$ cp hello.txt hello_new.txt
leprecha@Ubuntu-DevOps:~/practice$ ls -la
total 16
drwxr-xr-x  2 leprecha sysadmin 4096 Aug 19 21:13 .
drwxr-x--- 19 leprecha sysadmin 4096 Aug 19 21:08 ..
-rw-r--r--  1 leprecha sysadmin  13 Aug 19 21:13 hello_new.txt
-rw-r--r--  1 leprecha sysadmin  13 Aug 19 21:08 hello.txt
```

Copy hello.txt in hello_new.txt.

Renaming (**mv**).

```
leprecha@Ubuntu-DevOps:~/practice$ mv hello_new.txt renamed.txt
leprecha@Ubuntu-DevOps:~/practice$ ls -la
total 16
```

```
drwxr-xr-x  2 leprecha sysadmin 4096 Aug 19 21:14 .
drwxr-x--- 19 leprecha sysadmin 4096 Aug 19 21:08 ..
-rw-r--r--  1 leprecha sysadmin  13 Aug 19 21:08 hello.txt
-rw-r--r--  1 leprecha sysadmin  13 Aug 19 21:13 renamed.txt
```

Renaming hello_new.txt in renamed.txt.

Deleting (rm).

```
leprecha@Ubuntu-DevOps:~/practice$ rm hello.txt
leprecha@Ubuntu-DevOps:~/practice$ ls -la
total 12
drwxr-xr-x  2 leprecha sysadmin 4096 Aug 19 21:15 .
drwxr-x--- 19 leprecha sysadmin 4096 Aug 19 21:08 ..
-rw-r--r--  1 leprecha sysadmin  13 Aug 19 21:13 renamed.txt
```

Deleting file hello.txt.

Learn basic FHS structure (`/etc` , `/var` , `/usr` , `/home`).

1. `/etc`

System and service configuration files.

Contains settings for everything: network (`hosts` , `hostname`), users (`passwd` , `shadow`), services (`ssh/sshd_config` , `cron.d`).

2. `/var`

Variable data that changes frequently. Logs, queues, databases, caches.

Examples:

- `/var/log` — system and application logs.
- `/var/spool` — job queues (printing, mail).

- `/var/cache` — program caches.
-

3. `/usr`

Programs and files installed for all users.

- `/usr/bin` — executables (commands).
 - `/usr/lib` — libraries.
 - `/usr/share` — shared data (icons, docs).
-

4. `/home`

User home directories. Each contains personal files, settings, and work data.

Example: `/home/sysadmin`

```
/          → Root of the system
├── etc/    → System & service configs
│   ├── hosts → Local DNS
│   ├── passwd → Users
│   └── ssh/   → SSH settings
├── var/    → Variable data
│   ├── log/  → Logs
│   ├── cache/ → Caches
│   └── spool/ → Job queues
├── usr/    → Programs & libraries
│   ├── bin/ → Executables
│   ├── lib/ → Libraries
│   └── share/ → Shared data
├── home/   → User home dirs
│   ├── user1/ → Personal files of user1
│   └── user2/ → Personal files of user2
├── tmp/    → Temporary files
```



```
|
|— bin/      → Essential commands
|— sbin/     → System utilities (root)
|— root/     → Root's home directory
```

Remember:

- `/etc` — settings.
- `/var` — frequently changing data.
- `/usr` — programs.
- `/home` — personal data.

3. Practice

1. Creating directory structure.

- Create a folder `projects` with three subfolders: `scripts`, `configs`, `logs`.

```
leprecha@Ubuntu-DevOps:~$ mkdir -p ~/projects/{scripts,configs,logs}
leprecha@Ubuntu-DevOps:~$ cd projects
leprecha@Ubuntu-DevOps:~/projects$ la -ls
total 12
4 drwxr-xr-x 2 leprecha sysadmin 4096 Aug 19 21:22 configs
4 drwxr-xr-x 2 leprecha sysadmin 4096 Aug 19 21:22 logs
4 drwxr-xr-x 2 leprecha sysadmin 4096 Aug 19 21:22 scripts
```

`mkdir -p ~/projects/{scripts,configs,logs}` - create folder `projects` with three subfolders: `scripts`, `configs`, `logs`.

- `mkdir` — create a directory.
- `p` — create all missing parent directories.
- `~` — my home directory (`/home/my_name`).

- `projects/{scripts,configs,logs}` — Brace expansion in bash will create three subfolders `scripts,configs,logs` .

2. Working with files.

- Create two empty files in `configs` , add text to `startup.log` .

```
leprecha@Ubuntu-DevOps:~$ touch ~/projects/configs/{nginx.conf,ssh_conf
g}
leprecha@Ubuntu-DevOps:~$ cd projects/configs
leprecha@Ubuntu-DevOps:~/projects/configs$ la -ls
total 0
0 -rw-r--r-- 1 leprecha sysadmin 0 Aug 19 21:26 nginx.conf
0 -rw-r--r-- 1 leprecha sysadmin 0 Aug 19 21:26 ssh_config
```

```
touch ~/projects/configs/{nginx.conf,ssh_config}
```

- `touch` — creates empty files (or updates the modification time).
- `~/projects/configs/` — path to `configs` in home directory.
- `{nginx.conf,ssh_config}` — brace expansion creates two files at once.

```
leprecha@Ubuntu-DevOps:~$ echo "Hello DevOps" > ~/projects/logs/startup.
log
leprecha@Ubuntu-DevOps:~$ cd projects/logs
leprecha@Ubuntu-DevOps:~/projects/logs$ cat startup.log
Hello DevOps
```

```
echo "Hello DevOps" > ~/projects/logs/startup.log
```

- `echo` — prints text or the value of a variable to the terminal.
- `>` — redirects output into a file, overwriting it.
- `>>` — redirects output to the end of a file.
- `cat` — prints the contents of a file to the terminal.

3. Copying and backups.

- Make a copy of `startup.log` .

```
leprecha@Ubuntu-DevOps:~$ cp ~/projects/logs/startup.log ~/projects/logs/
startup.log.bak
leprecha@Ubuntu-DevOps:~$ cd projects/logs
leprecha@Ubuntu-DevOps:~/projects/logs$ la -ls
total 8
4 -rw-r--r-- 1 leprecha sysadmin 13 Aug 19 21:28 startup.log
4 -rw-r--r-- 1 leprecha sysadmin 13 Aug 19 21:36 startup.log.bak
```

```
cp ~/projects/logs/startup.log ~/projects/logs/startup.log.bak
```

- `cp` — copy files and directories.
- The first argument is the source file (`startup.log`).
- The second is the new file (`startup.log.bak`).
- `r` — recursive copy (needed for directories).
- `i` — interactive, asks before overwriting files.

4. Searching files.

- Find all `.conf` files in `projects` .

```
leprecha@Ubuntu-DevOps:~$ find ~/projects -name "*.conf"
/home/leprecha/projects/configs/nginx.conf
```

```
find ~/projects -name "*.conf"
```

- `find` — searches for files and directories.
- `~/projects` — where to search (here it's your `projects` directory).
- `name "*.conf"` — condition: the file name must end with `.conf` .
 - — any sequence of characters.

- `.conf` — the file extension itself.

5. Permissions.

- Give the file owner read/write permissions only for `ssh_config`.

```
leprecha@Ubuntu-DevOps:~$ chmod 600 ~/projects/configs/ssh_config
```

```
chmod 600 ~/projects/configs/ssh_config
```

- `chmod` — *change mode*, modifies file permissions.
- `600` — octal representation of permissions:
 - `6` — `rw-` (read + write) for the owner.
 - `0` — `--` (no access) for the group.
 - `0` — `--` (no access) for others.

```
ls -l ~/projects/configs/ssh_config - example output.
```

```
leprecha@Ubuntu-DevOps:~$ ls -l ~/projects/configs/ssh_config
-rw----- 1 leprecha sysadmin 0 Aug 19 21:26 /home/leprecha/projects/configs/ssh_config
```

- `rw-----` — file permissions.
- `1` — number of links.
- `leprecha` — owner of the file.
- `sysadmin` — owner's group.
- `0` — file size in bytes.
- `Aug 19 21:26` — last modification date.
- `ssh_config` — file name.
- `ls -R ~/projects` — shows all contents of the `projects` directory recursively.

4. Daily Summary

What learned: Basic Linux commands, working with nano, filesystem structure

What was hard: nothing

What to repeat: File permissions commands

Ideas for automation/projects: Script to auto-create folder structure