Day-6 SRE Training

Topic - Linux

Linux is an open-source, Unix-like operating system that manages hardware and software resources on a computer. It is widely used in servers, embedded systems, cloud computing, and personal computers. It offers security, stability, and flexibility, with various distributions like Ubuntu, CentOS, and Debian.

Basic Linux commands:

`pwd` (Print working directory) → Prints the current directory path.

`ls` → Lists files and directories in the current location.

`ls -l` → Long format with permissions, owner, size

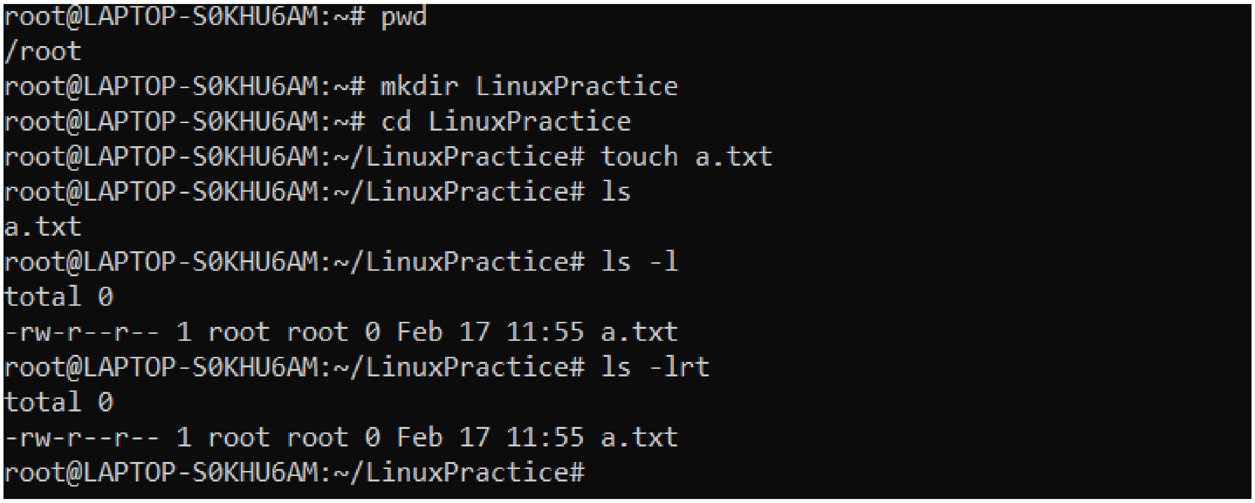
`ls -a` → Shows hidden files.

`ls -lrt` → Lists files in a directory in **long format**, **sorted by modification time (oldest first)**, and **in reverse order**.

`cd folder_name` → Changes the current directory.

`cd Documents` → Moves into the Documents folder.

`cd ..` → Moves up one directory.



```
root@LAPTOP-S0KHU6AM:~# pwd
/root
root@LAPTOP-S0KHU6AM:~# mkdir LinuxPractice
root@LAPTOP-S0KHU6AM:~# cd LinuxPractice
root@LAPTOP-S0KHU6AM:~/LinuxPractice# touch a.txt
root@LAPTOP-S0KHU6AM:~/LinuxPractice# ls
a.txt
root@LAPTOP-S0KHU6AM:~/LinuxPractice# ls -l
total 0
-rw-r--r-- 1 root root 0 Feb 17 11:55 a.txt
root@LAPTOP-S0KHU6AM:~/LinuxPractice# ls -lrt
total 0
-rw-r--r-- 1 root root 0 Feb 17 11:55 a.txt
root@LAPTOP-S0KHU6AM:~/LinuxPractice#
```

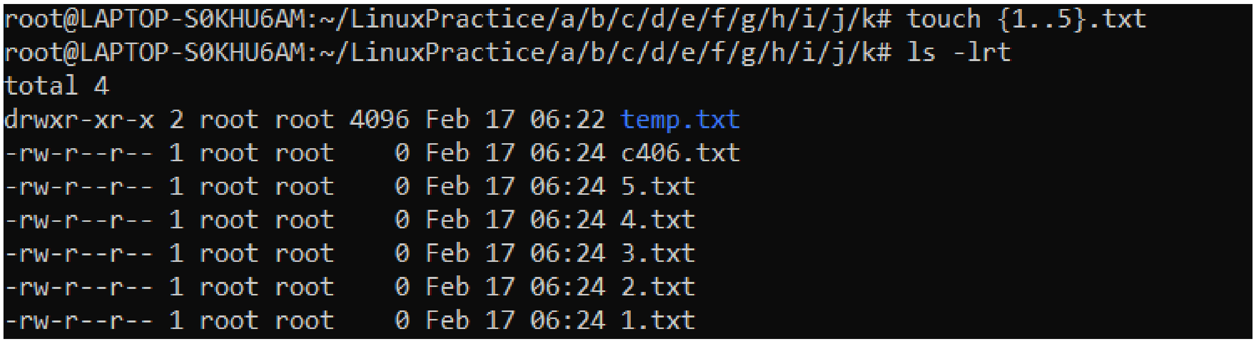
`mkdir folder_name` → Creates a new directory.

`mkdir -p` → Creates parent directories if they don’t exist.

`touch file_name` → Creates an empty file.

touch {1..5}.txt → Creates files 1.txt, 2.txt, 3.txt, 4.txt, and 5.txt.

`{1..5}` → Expands to `1 2 3 4 5` `.txt` → Appends `.txt` to each number. `touch` → Creates empty files if they don’t exist.

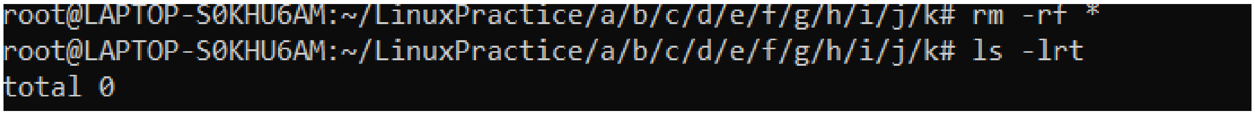


```
root@LAPTOP-S0KHU6AM:~/LinuxPractice/a/b/c/d/e/f/g/h/i/j/k# touch {1..5}.txt
root@LAPTOP-S0KHU6AM:~/LinuxPractice/a/b/c/d/e/f/g/h/i/j/k# ls -lrt
total 4
drwxr-xr-x 2 root root 4096 Feb 17 06:22 temp.txt
-rw-r--r-- 1 root root    0 Feb 17 06:24 c406.txt
-rw-r--r-- 1 root root    0 Feb 17 06:24 5.txt
-rw-r--r-- 1 root root    0 Feb 17 06:24 4.txt
-rw-r--r-- 1 root root    0 Feb 17 06:24 3.txt
-rw-r--r-- 1 root root    0 Feb 17 06:24 2.txt
-rw-r--r-- 1 root root    0 Feb 17 06:24 1.txt
```

`rm file.txt` → Deletes a specific file.

`rm -r folder` → Deletes a folder and its contents.

rm -rf * → Removes all files and directories in the current directory recursively without prompting for confirmation.



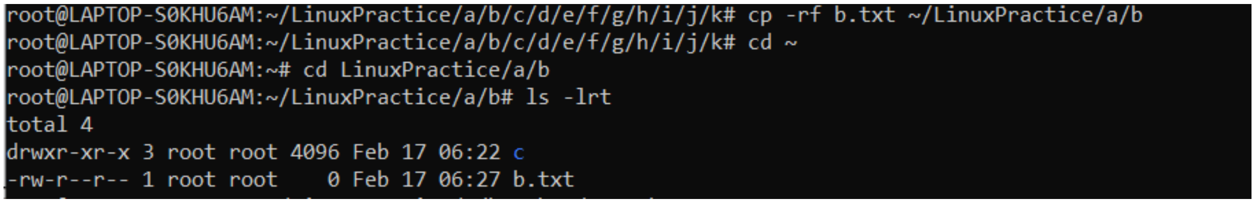
```
root@LAPTOP-S0KHU6AM:~/LinuxPractice/a/b/c/d/e/f/g/h/i/j/k# rm -rf *
root@LAPTOP-S0KHU6AM:~/LinuxPractice/a/b/c/d/e/f/g/h/i/j/k# ls -lrt
total 0
```

`cp -r folder1 folder2` → Copies a folder and its contents.

`cp file1 file2` → Copies a file.

`mv oldname.txt newname.txt` → Renames or moves a file.

`mv file.txt /home/user/Documents/` → Moves a file to another location.

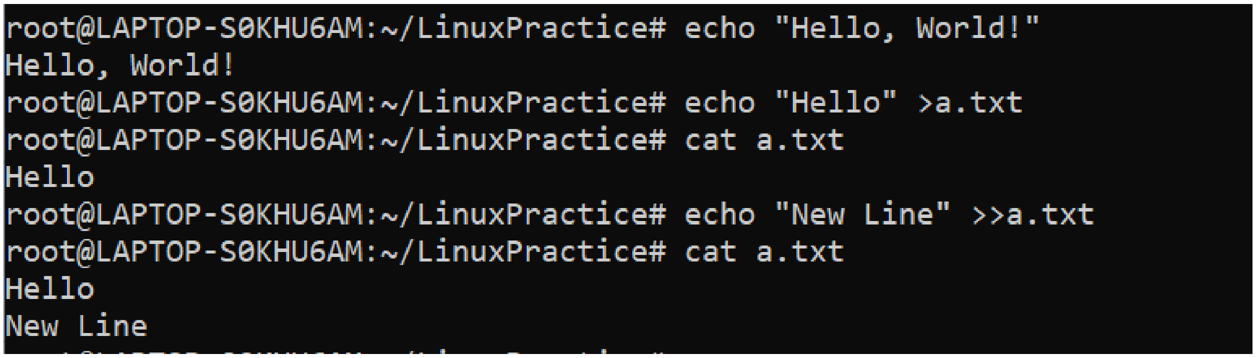


```
root@LAPTOP-S0KHU6AM:~/LinuxPractice/a/b/c/d/e/f/g/h/i/j/k# cp -rf b.txt ~/LinuxPractice/a/b
root@LAPTOP-S0KHU6AM:~/LinuxPractice/a/b/c/d/e/f/g/h/i/j/k# cd ~
root@LAPTOP-S0KHU6AM:~# cd LinuxPractice/a/b
root@LAPTOP-S0KHU6AM:~/LinuxPractice/a/b# ls -lrt
total 4
drwxr-xr-x 3 root root 4096 Feb 17 06:22 c
-rw-r--r-- 1 root root    0 Feb 17 06:27 b.txt
```

`echo "Hello, World!"` → Prints text in the terminal.

`echo "Hello" > file.txt` → Writes text to a file (overwrites content).

`echo "New line" >> file.txt` → Appends text to a file.



```
root@LAPTOP-S0KHU6AM:~/LinuxPractice# echo "Hello, World!"
Hello, World!
root@LAPTOP-S0KHU6AM:~/LinuxPractice# echo "Hello" >a.txt
root@LAPTOP-S0KHU6AM:~/LinuxPractice# cat a.txt
Hello
root@LAPTOP-S0KHU6AM:~/LinuxPractice# echo "New Line" >>a.txt
root@LAPTOP-S0KHU6AM:~/LinuxPractice# cat a.txt
Hello
New Line
```

`history` → Lists previously used commands.

`clear` → Clears the terminal screen.

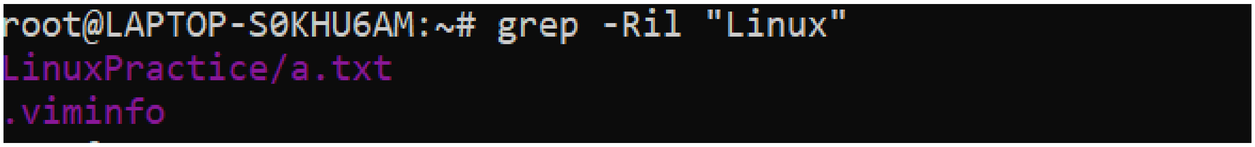
`man command_name` → Shows help documentation for a command.

`grep -Ril "error" log.txt` → Searches for specific text in a file.

`-R`: Stands for **recursive**. It tells `grep` to search through all files in a directory and its subdirectories.

`-i`: Stands for **ignore case**. It tells `grep` to ignore case (uppercase or lowercase) when searching for the pattern.

`-l`: Stands for **list**. It tells `grep` to only display the names of files containing the matching pattern, rather than the actual matching lines.



```
root@LAPTOP-S0KHU6AM:~# grep -Ril "Linux"
LinuxPractice/a.txt
.viminfo
```

There are three types of permissions:

1. **Read (r)**: Allows viewing the content of a file or listing the contents of a directory.
2. **Write (w)**: Allows modifying the content of a file or adding/removing files in a directory.
3. **Execute (x)**: Allows running a file as a program or entering a directory.

These permissions are assigned to three categories of users:

1. **User (u)**: The file's owner.
2. **Group (g)**: Other users who belong to the same group as the file.
3. **Others (o)**: All other users.

Permissions are displayed in the following format when you run `ls -l`

The first character indicates the file type (e.g., - for regular files, d for directories).

The next three characters show the owner's permissions (rwx = read, write, execute).

The next three characters show the group's permissions (r-x = read, execute).

The last three characters show the permissions for others (r-- = read).

Changing Permissions:

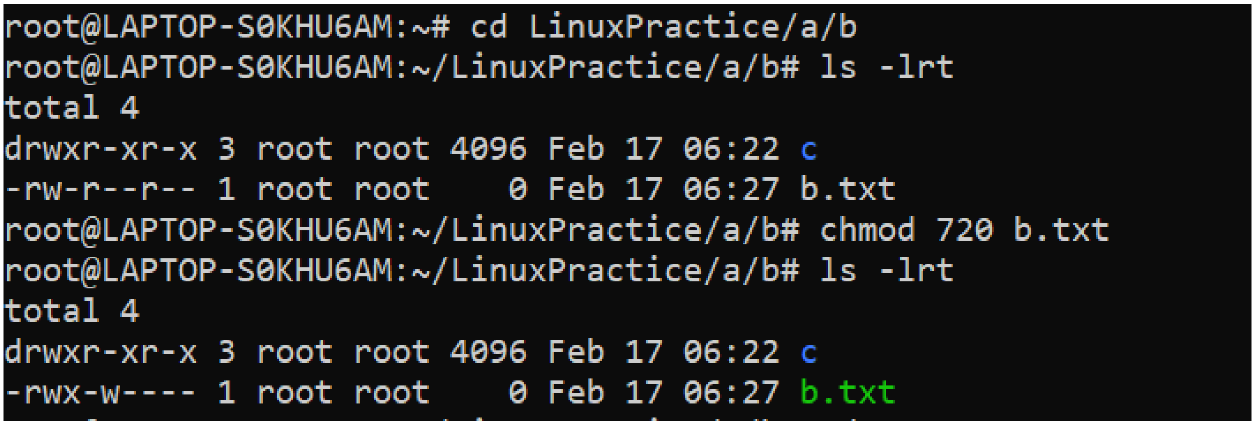
chmod: Changes file permissions.

Example: chmod 720 file.txt sets read, write, and execute for the owner, and write for the group and no permissions for others.

Numeric Representation:

Permissions can also be represented using numbers:

- **r = 4**, **w = 2**, **x = 1**
- Sum of permissions:
  - `rwx = 4 + 2 + 1 = 7`
  - `rw- = 4 + 2 = 6`
  - `r-- = 4`



```
root@LAPTOP-S0KHU6AM:~# cd LinuxPractice/a/b
root@LAPTOP-S0KHU6AM:~/LinuxPractice/a/b# ls -lrt
total 4
drwxr-xr-x 3 root root 4096 Feb 17 06:22 c
-rw-r--r-- 1 root root    0 Feb 17 06:27 b.txt
root@LAPTOP-S0KHU6AM:~/LinuxPractice/a/b# chmod 720 b.txt
root@LAPTOP-S0KHU6AM:~/LinuxPractice/a/b# ls -lrt
total 4
drwxr-xr-x 3 root root 4096 Feb 17 06:22 c
-rwx-w---- 1 root root    0 Feb 17 06:27 b.txt
```

Example: `chmod 720 file.txt` corresponds to:

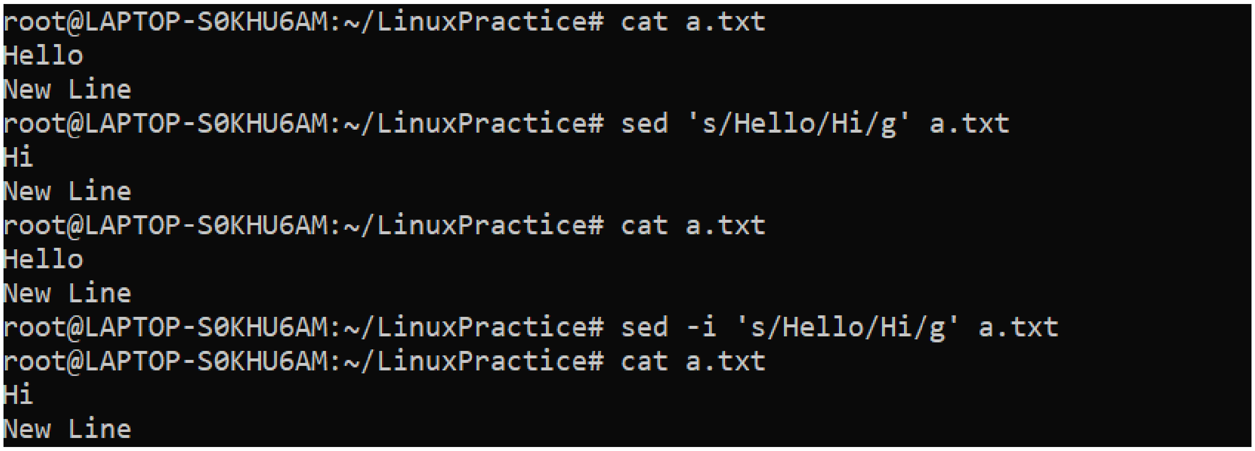
- `7` for owner (rwx),
- `2` for group (-w-),
- `0` for others (---).

`df -h` → Displays disk usage in human-readable format.

`ps aux` → Lists currently running processes.

`sed 's/old/new/g' file.txt` → Modifies file content without opening it.

`sed -i 's/old/new/g' file.txt` → Modifies and saves file content without opening it.



```
root@LAPTOP-S0KHU6AM:~/LinuxPractice# cat a.txt
Hello
New Line
root@LAPTOP-S0KHU6AM:~/LinuxPractice# sed 's/Hello/Hi/g' a.txt
Hi
New Line
root@LAPTOP-S0KHU6AM:~/LinuxPractice# cat a.txt
Hello
New Line
root@LAPTOP-S0KHU6AM:~/LinuxPractice# sed -i 's/Hello/Hi/g' a.txt
root@LAPTOP-S0KHU6AM:~/LinuxPractice# cat a.txt
Hi
New Line
```