**day05\_107856406\_dsdipt\_sudipto\_28may2025**

**Employee Code:** 107856406

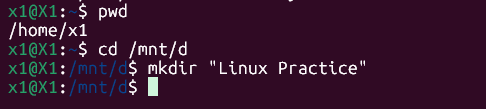
**Login ID:** dsdipt

**Email :** dsdipt@amazon.com

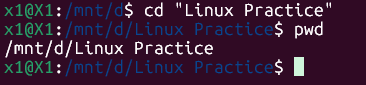
**Name:** Sudipto Das

**Date:** 28 May 2025 (Day 05)

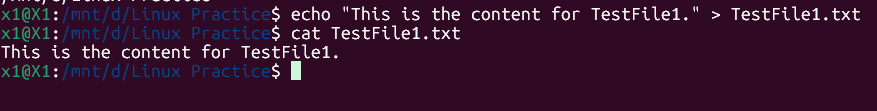
### ***Task 1: Create a Directory***

****

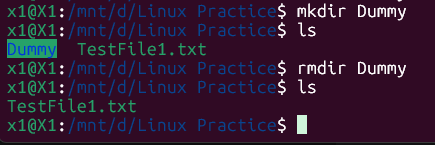
### ***Task 2: Change path to the Directory***



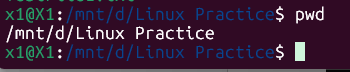
### ***Task 3: Create a file name TestFile1.txt and add the content to it.***



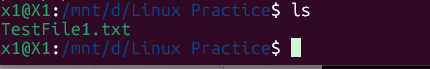
### ***Task 4: Create a Folder named Dummy and try to delete it.***



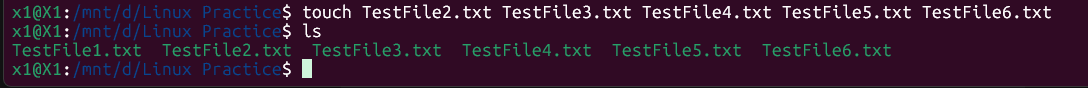
### ***Task 5: Check the working Directory***



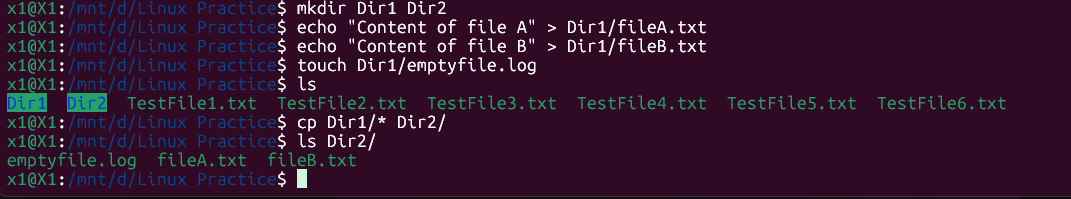
### ***Task 6: check all the files and directories in the directory.***



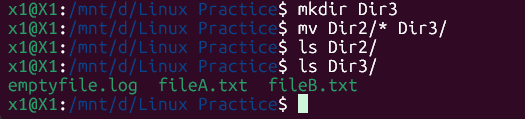
### ***Task 7: Create five files named TestFile2.txt to TestFile6.txt***

****

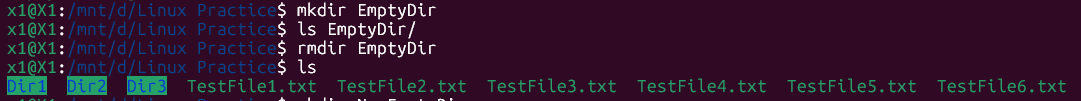
### ***Task 8: Copy all files from Dir 1 to Dir 2***



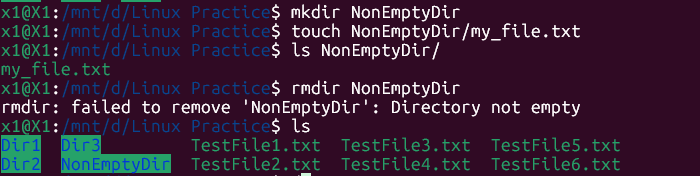
### ***Task 9: Move all files from Dir 2 to Dir 3 (Dir 2 should be empty)***

****

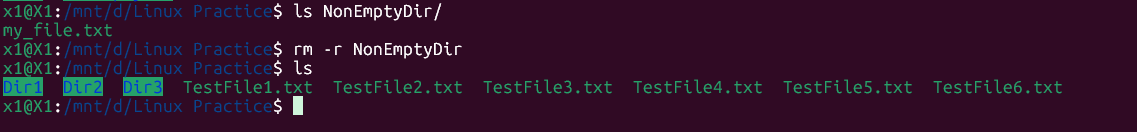
### ***Task 10: Show the difference between rm and rmdir?***



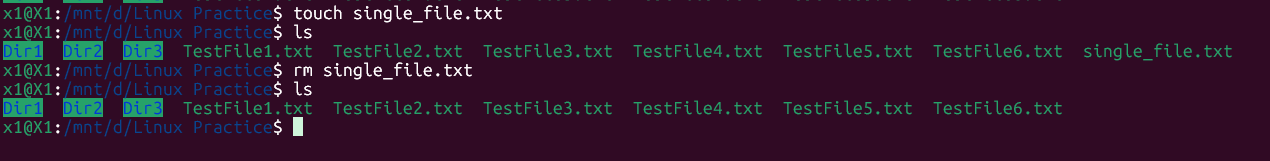
**Explanation: rmdir successfully removed EmptyDir because it was empty.**

****

**Explanation: rmdir failed because NonEmptyDir contains my\_file.txt. It will not delete a directory that has contents.**

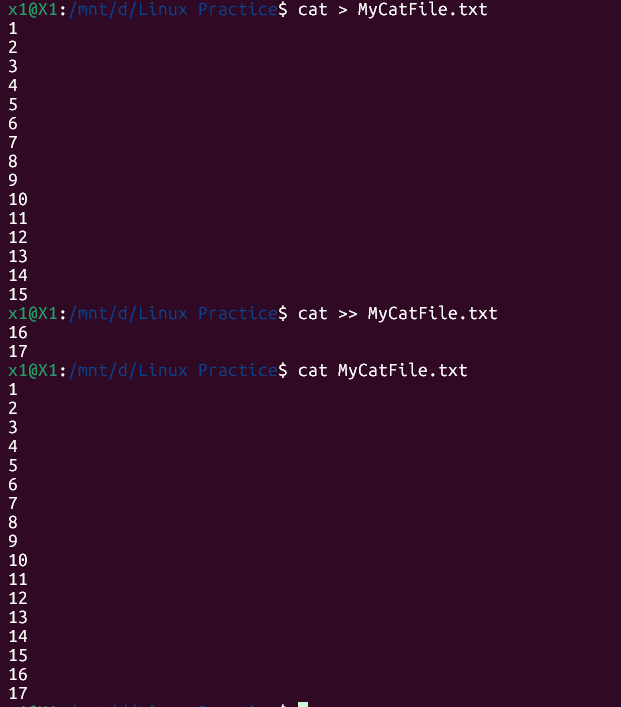
****

**Explanation: rm -r successfully removed NonEmptyDir and all its contents (my\_file.txt) because the -r (recursive) option tells rm to go into the directory and delete everything inside it, then delete the directory itself.**

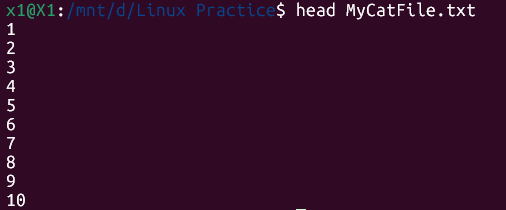
****

**Explanation: rm successfully removed single\_file.txt. This is its primary function.**

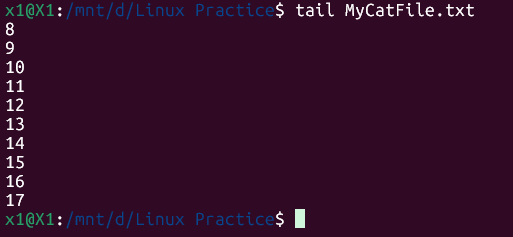
### ***Task 11: use cat command to create a file & add the dummy text***



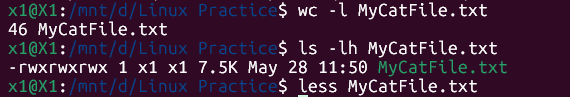
### ***Task 12: How to get only the top part of your file.***

****

### ***Task 13: How to get only the last part of your file***

****

### ***Task 14: add dummy text to the same file & show page by less command***

****

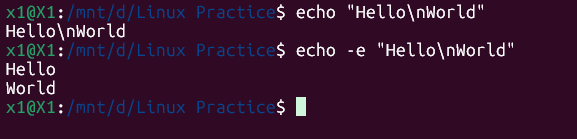
### ***Task 15: use more command & find the difference with less***



**less:** Allows you to scroll **both forward and backward** through the file. This is its most significant advantage.

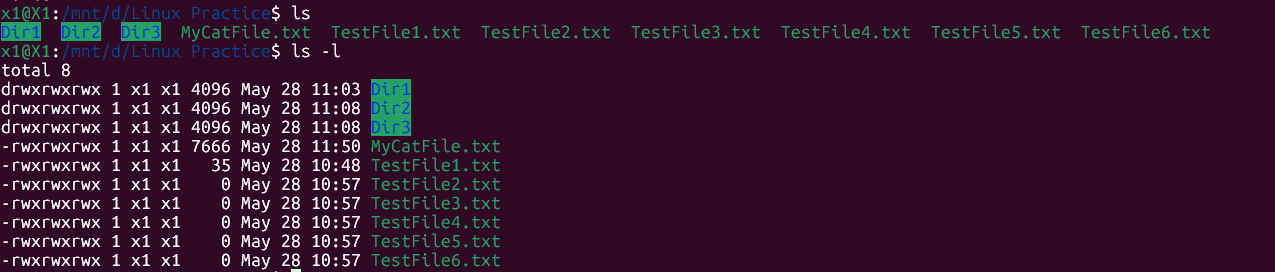
**more:** Primarily allows scrolling **only forward**. Once you've scrolled past a section, you generally cannot go back to it.

### ***Task 16: use echo command with -e and show the difference.***



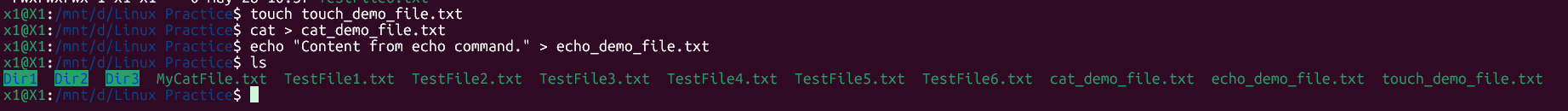
**Explanation:** With -e, \n is interpreted, and "World" is printed on a new line.

### ***Task 17: Difference between ls and ls -l command.***



**Difference:** ls gives you a **compact list of names**, while ls -l provides a **detailed, verbose listing** with crucial information about permissions, ownership, size, and modification times, which is essential for managing files and understanding their attributes.

### ***Task 18: create files using touch, cat & echo command, difference?***

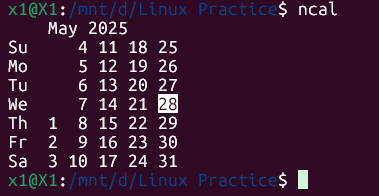


**Touch:** Primarily used to create empty files. If the file already exists, it updates its last accessed and modified timestamps without changing its content.

**Echo:** Best for creating a file with a single line of text (or outputting text to the terminal). It's typically used with output redirection (>) to place its string output into a file.

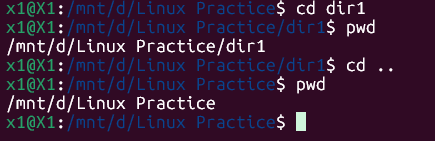
**Cat:** Versatile. When used with output redirection (>), it's ideal for creating a file by typing multiple lines of content interactively from the keyboard. It captures all lines you type until you signal the end of input (Ctrl + D).

### ***Task 19: display the calendar by using a command.***

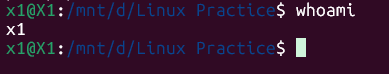
****

### 

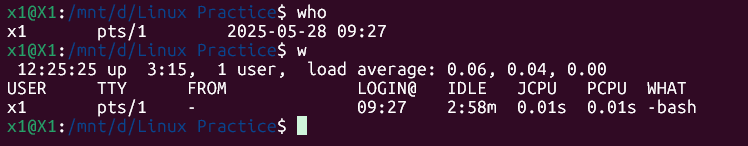
### ***Task 20: go back to 1 directory at a time***

****

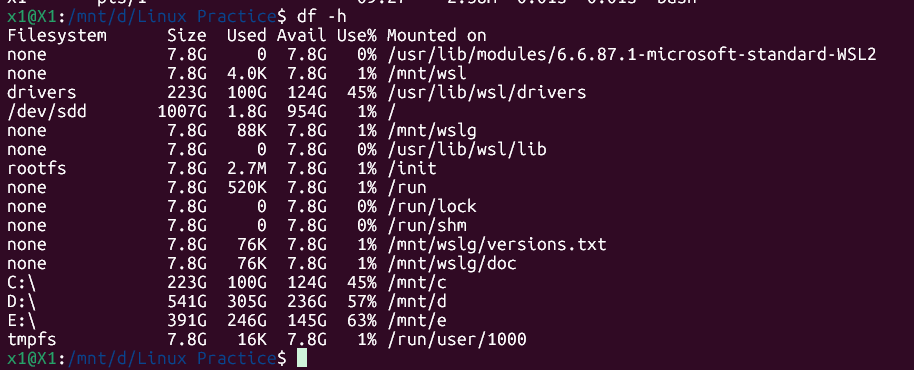
### ***Task 21: How to know who the user is?***

****

### ***Task 22: Try to find out who is peeping into your system.***



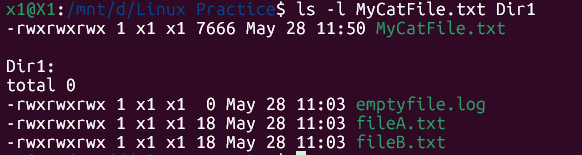
### ***Task 23: check how much disk space is consumed.***



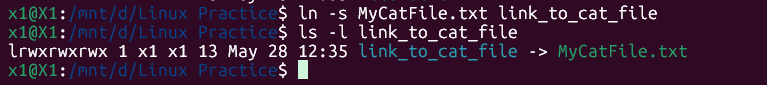
### ***Task 24: try using the below commands***

| **Prefix** | **Description** |
| --- | --- |
| **-** | **Regular file**, such as an ASCII text file, binary executable, or hard link. |
| **b** | **Block special file**. Block input/output device file such as a physical hard drive. |
| **c** | **Character special file**. Raw input/output device file such as a physical hard drive. |
| **d** | **Directory** which contains a listing of other files and directories. |
| **l** | **Symbolic link file**. Links on any regular file. |
| **p** | **Named pipe**. A mechanism for interprocess communications. |
| **s** | **Socket** which is used for interprocess communication. |

**Task 24, Part 1: Display a Regular File (-) and a Directory (d)**

****

**Task 24, Part 2: Create and Display a Symbolic Link (l)**

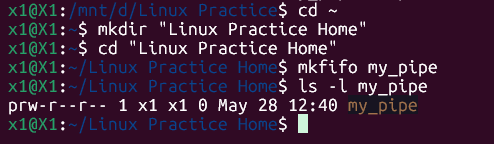
****

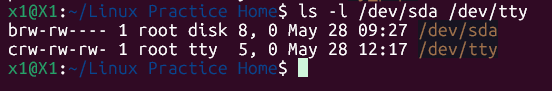
**ln -s: This is the command to create a symbolic link.**

**MyCatFile.txt: This is the original file that the link will point to.**

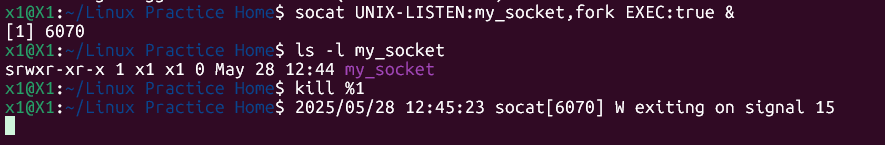
**link\_to\_cat\_file: This is the name of the new symbolic link.**

**Task 24, Part 3: Create and Display a Named Pipe (p)**

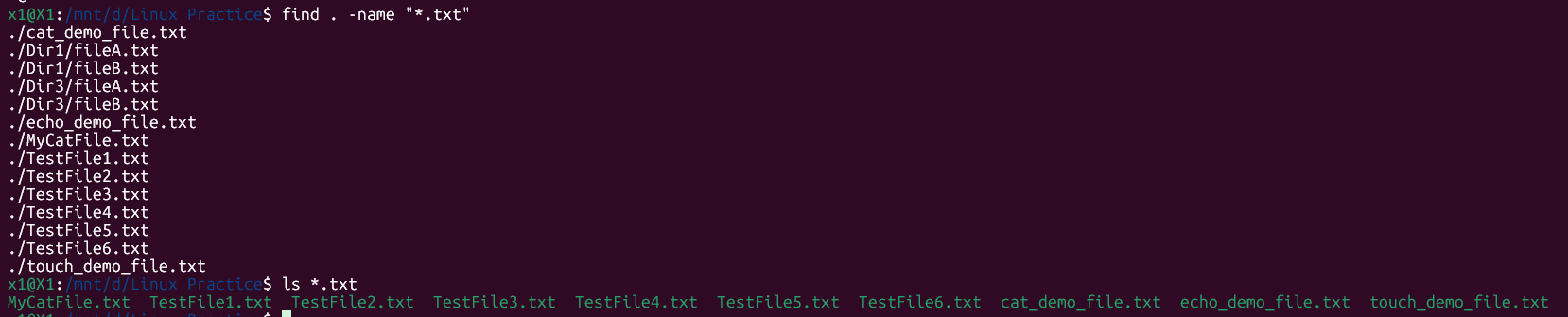
****

**Task 24, Part 4: Inspect Block (b) and Character (c) Special Files**

**Task 24, Part 5 (Optional - Socket s):**

****

### ***Task 25: Find the list of all files ending with .txt***



### ***Task 26: How to check all the hidden files in Linux.***



### 

### ***Task 27: What is the difference between (.) and (..) in linux***

* **“.” (single dot): Refers to the current working directory.**
* **“..” (double dot): Refers to the parent directory (the directory directly above the current one).**

### ***Task 28: create a file using vi editor and show the details in ss***

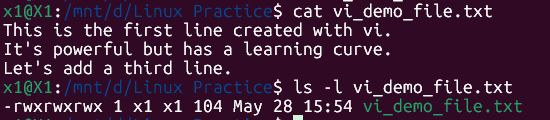
**I - to insert To move inside the file. Esc is for come out of the edit mode**

**Press Shift + ZZ or :wq to come out of the file completely.**

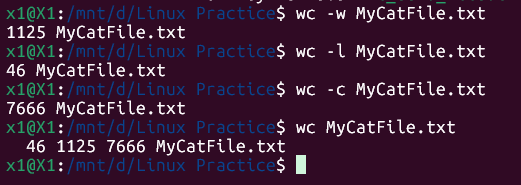
* l key to move to the right side.
* h key to move to the left side.
* k key to move upside in the file.
* j key to move downside in the file.







### ***Task 29: How to find the number of words in a file***



**wc -w MyCatFile.txt:** This option tells wc to count only the words.

**wc -l MyCatFile.txt:** Counts the number of lines.

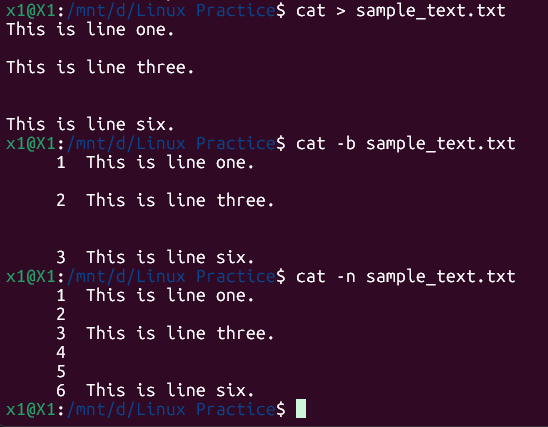
**wc -c MyCatFile.txt:** Counts the number of bytes (characters).

**wc MyCatFile.txt:** (Without any options) Shows lines, words, and bytes in that order.

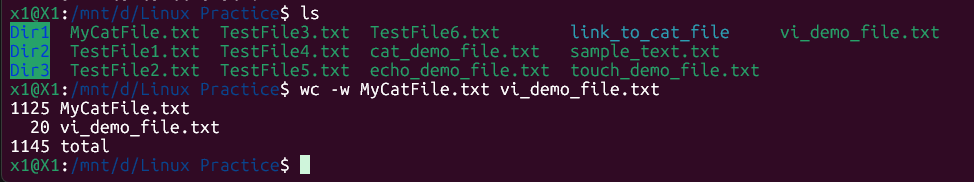
### ***Task 30: What is the use of cat -b myfilename.txt command?***

**-b**: This is an option that modifies cat's behavior. It stands for "number non-blank output lines." It tells cat to prepend line numbers to each line of output, *but it skips numbering any lines that are completely empty (blank)*.

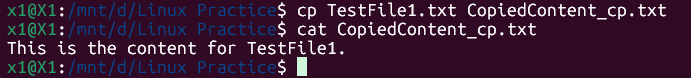
**-n:** which numbers *all* output lines, including blank ones.



### ***Task 31: use the wc command with 2 or more files.***

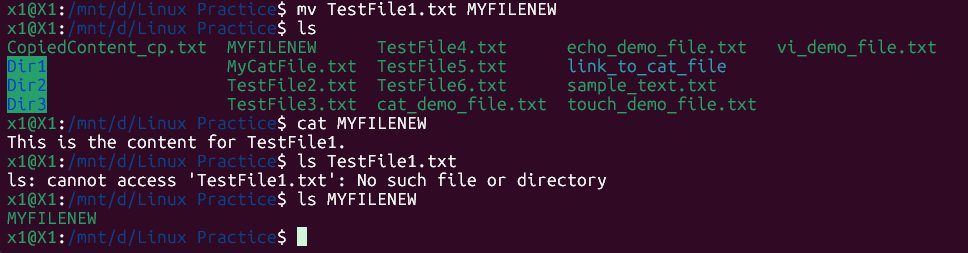


### ***Task 32: Copy content of one file to another file***

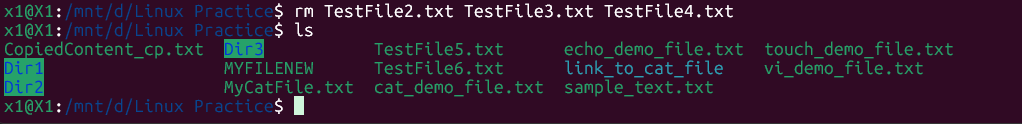


**cp is generally preferred for copying entire files** because it preserves more file attributes.

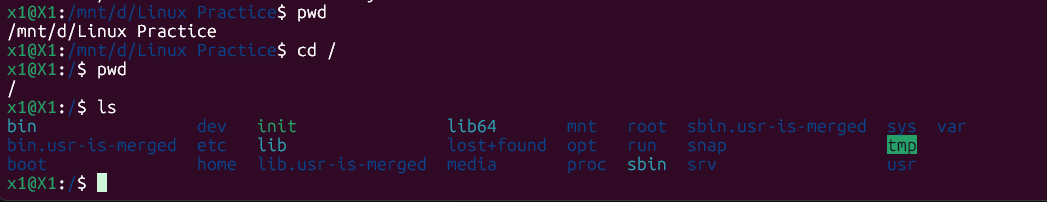
### ***Task 33: Now rename the file with MYFILENEW***



### ***Task 34: Delete multiple files in linux.***



### ***Task 35: In directory / slash is root. Command try cd /***



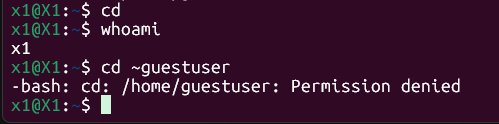
The command cd / will change the current working directory to the **root directory**.

| **Directory** | **Description** |
| --- | --- |
| **/bin** | important binary applications |
| **/boot** | boot configuration files, kernels, and other files needed at boot time. |
| **/dev** | System device files. |
| **/etc** | configuration files, startup scripts, etc. |
| **/home** | List of home directories for different users |
| **/lib** | system libraries, shared libraries |
| **/lost+found** | a lost+found system for files that exist under the root (/) directory |
| **/media** | automatically mounted (loaded) partitions on your hard drive and removable media such as CDs, digital cameras, etc. |
| **/mnt** | manually mounted filesystems on your hard drive |
| **/opt** | 3rd part applications to be installed |
| **/proc** | Maintains information about the state of the system, including currently running processes. |
| **/root** | root user's home directory. |
| **/sbin** | important system binaries |
| **/srv** | contain files that are served to other systems |
| **/sys** | system files |
| **/tmp** | temporary files |
| **/usr** | applications and files that are mostly available for all users to access |
| **/var** | variable files such as logs and databases |

### ***Task 36: What is the way to go to the home directory?***



### ***Task 37: move to different user’s home directory***



### ***Task 38: chmod***

**Every file & directory in Linux has permissions defined for three categories of users:**

* **Owner (u):** The user who owns the file.
* **Group (g):** The group that owns the file.
* **Others (o):** All other users on the system (not the owner & not in the owning group).
* **All (a):** Refers to all three (owner, group, and others).

**And three types of permissions:**

**r (read):**

* For files: Allows viewing the file's content.
* For directories: Allows listing the contents of the directory.

**w (write):**

* For files: Allows modifying or deleting the file.
* For directories: Allows creating, deleting, or renaming files within the directory.

**x (execute):**

* For files: Allows running the file as a program or script.
* For directories: Allows entering (traversing) the directory.

**Octal (Numeric) Mode:**

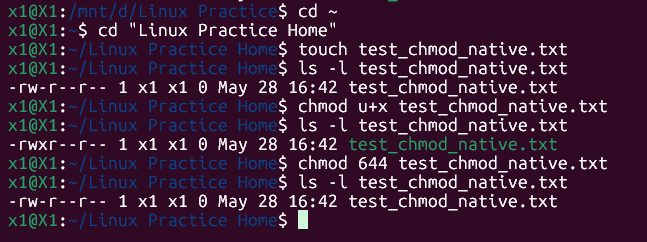
This method uses numbers to represent permissions, which is often preferred for precision and scripting. Each permission type has a numeric value:

* **r (read) = 4**
* **w (write) = 2**
* **x (execute) = 1**
* **No permission = 0**

You sum these values for each category (owner, group, others).

**Common combinations:**

* 7 (4+2+1): Read, Write, Execute (rwx)
* 6 (4+2): Read, Write (rw-)
* 5 (4+1): Read, Execute (r-x)
* 4 (4): Read only (r--)
* 0: No permissions (---)



### ***Commands***

**Basic Navigation & File Operations:**

* cd: Change directory.
* pwd: Print working directory.
* ls: List directory contents.
* mkdir: Create a new directory.
* rmdir: Remove an empty directory.
* rm: Remove files or directories.
* cp: Copy files or directories.
* mv: Move or rename files or directories.
* touch: Create an empty file.

**File Content Management:**

* cat: Concatenate and display the contents of a file.
* head: Display the first part of a file.
* tail: Display the last part of a file.
* less: Display file contents, page by page.
* more: Display file contents, page by page.

**System Information & Utility:**

* uname: Print system information.
* whoami: Display the current user's name.
* date: Display the current date and time.
* cal: Display the calendar for a specified month or year.
* df: Display disk space usage.
* du: Display disk space usage for a specific directory.
* top: Display real-time processor activity.
* man: Access the command manual.
* grep: Search for patterns within files.
* echo: Print text to the terminal.

**Other Useful Commands:**

* sudo: Execute a command with superuser privileges.
* chmod: Change file permissions.
* chown: Change file ownership.
* tar: Create archives.
* gzip: Compress or decompress files.
* unzip: Unzip archives.
* zip: Create zip archives.

=========================================================================