Linux Commands Assignment

📌 Objective

This document demonstrates practical usage of basic Linux command-line utilities for file handling, content inspection, text searching, file compression, downloading files, modifying permissions, and managing environment variables.

---

1. 📁 Creating and Renaming Files/Directories

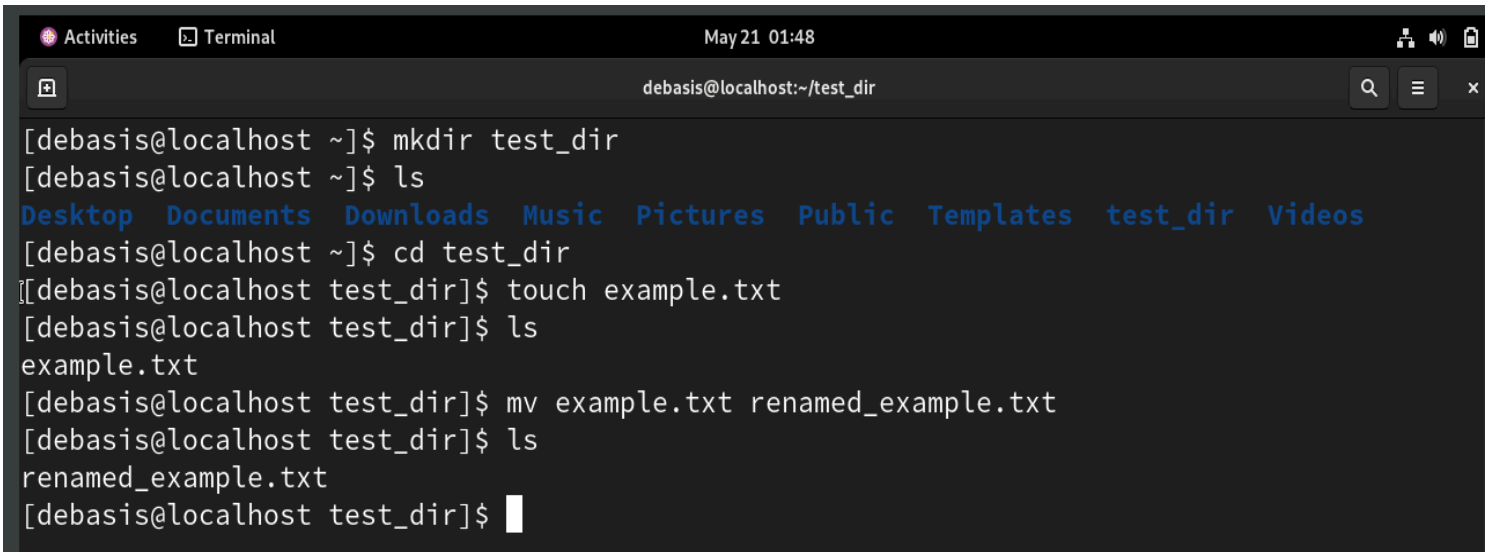
Create a directory named test_dir using mkdir.

Inside test_dir, create an empty file called example.txt.

Rename example.txt to renamed_example.txt using mv

Commands:

mkdir test_dir
cd test_dir
touch example.txt
mv example.txt renamed_example.txt



Explanation:

- mkdir test_dir: Creates a new directory named test_dir.
- cd test_dir: Navigates into the test_dir directory.
- touch example.txt: Creates an empty file named example.txt.
- mv example.txt renamed_example.txt: Renames the file to renamed_example.txt.

This operation is useful when organizing files and structuring project directories.

2. 📄 Viewing File Contents

Use cat to display the contents of /etc/passwd.

Display only the first 5 lines of /etc/passwd using head.

Display only the last 5 lines of /etc/passwd using tail.

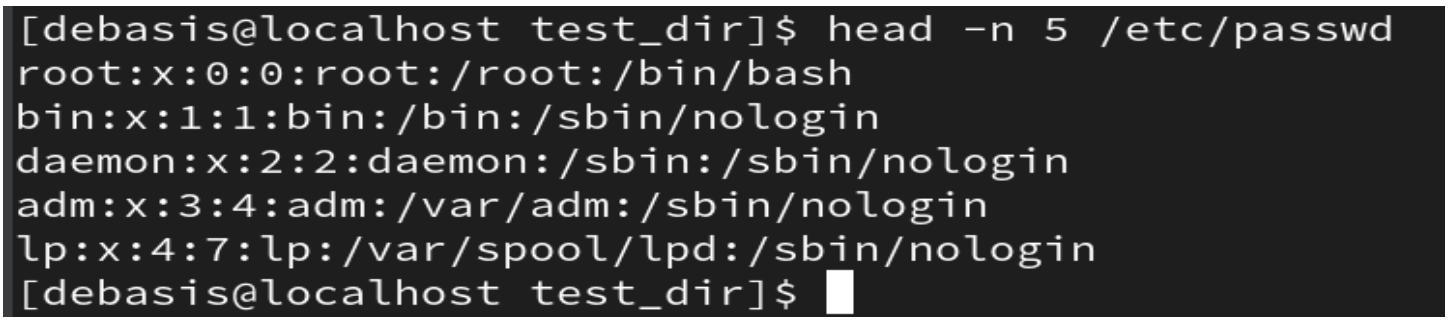
a. Display entire file:

cat /etc/passwd

- Shows complete contents of the /etc/passwd file, which stores user account details.

b. Display the first 5 lines:

head -n 5 /etc/passwd



- Useful for previewing the beginning of a large file (Shows the first 5 lines).

c. Display the last 5 lines:

tail -n 5 /etc/passwd

- Helpful when looking at recently added entries (Shows the last 5 lines)

3. 🔍 Searching for Patterns in Files

Use grep to find all lines containing the word "root" in /etc/passwd.

Command:

grep "root" /etc/passwd

```
[debasis@localhost ~]$ grep "root" /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin
[debasis@localhost ~]$
```

Explanation:

- grep is used to search for patterns in a file.
- In this example, it searches for occurrences of the string “root” within /etc/passwd.
- This is commonly used to locate specific user or configuration entries.

4. 📦 Zipping and Unzipping Files

Compress the test_dir directory into a file named test_dir.zip using zip.

Unzip test_dir.zip into a new directory named unzipped_dir.

a. Compress directory:

zip -r test_dir.zip test_dir

```
[debasis@localhost ~]$ zip -r test_dir.zip test_dir
  adding: test_dir/ (stored 0%)
  adding: test_dir/renamed_example.txt (stored 0%)
[debasis@localhost ~]$
```

b. Unzip into a new directory:

unzip test_dir.zip -d unzipped_dir

```
[debasis@localhost ~]$ unzip test_dir.zip -d unzipped_dir
Archive:  test_dir.zip
   creating: unzipped_dir/test_dir/
 extracting: unzipped_dir/test_dir/renamed_example.txt
[debasis@localhost ~]$
```

Explanation:

- zip -r compresses the test_dir directory and its contents recursively into a .zip file.
- unzip -d extracts the zip file contents into unzipped_dir.

These operations are useful for backup, sharing, and archiving.

5. 🌐 Downloading Files Using wget

Use wget to download a file from a URL (e.g., https://example.com/sample.txt).

Command:

wget https://example.com/sample.txt

```
[debasis@localhost ~]$ wget https://example.com/sample.txt
--2025-05-21 02:06:28--  https://example.com/sample.txt
Resolving example.com (example.com)... 2600:1406:3a00:21::173e:2e65, 2600:1406:3a00:21::17
3e:2e66, 2600:1406:bc00:53::b81e:94c8, ...
Connecting to example.com (example.com)|2600:1406:3a00:21::173e:2e65|:443...
```

Explanation:

- wget is a command-line utility to download files from the Internet.
- Replace the URL with a real file URL to fetch remote content.

6. 🔐 Changing File Permissions

Create a file named secure.txt and change its permissions to read-only for everyone using chmod.

Command:

touch secure.txt
chmod 444 secure.txt

```
[debasis@localhost ~]$ touch secure.txt
[debasis@localhost ~]$ ls
Desktop    Downloads  Pictures  secure.txt  test_dir      unzipped_dir
Documents  Music      Public    Templates   test_dir.zip  Videos
[debasis@localhost ~]$ chmod 444 secure.txt
[debasis@localhost ~]$ ll secure.txt
-r--r--r--. 1 debasis debasis 0 May 21 02:01 secure.txt
[debasis@localhost ~]$
```

Explanation:

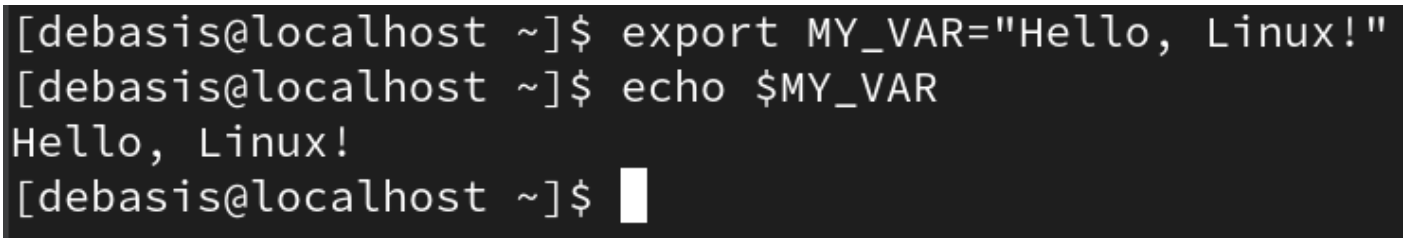
- touch creates a file named secure.txt.
- chmod 444 sets the file permission to **read-only** for all users (owner, group, others).
- This is crucial for protecting sensitive files from being modified.

7. 🧬 Working with Environment Variables

Use export to set a new environment variable called MY_VAR with the value "Hello, Linux!".

Command:

```
export MY_VAR="Hello, Linux!"
echo $MY_VAR
```



Explanation:

- export assigns a new environment variable named MY_VAR.
- echo $MY_VAR outputs its value.
- Environment variables are essential for customizing shell behavior and storing temporary configuration values.

✅ Final Notes

- Ensure that screenshots are added for each command to demonstrate execution.
- For the wget example, use an actual URL to show a successful download.

GitHub repo link – https://github.com/Debasis-21/Linux-Commands-Assignment