Experiment [2]: [Linux file systems permissions and essential commands]

Name: vinit kumar Roll.: 590029353 Date: 2025-17-05

AIM:

- [To Learn linux file systems permissions and essential commands]

Requirements:

- [Any Linux Distro, any kind of text editor (vs code, vim, notepad, nano, etc,]

Theory:

- [Basic Linux file systems permissions and essential commands]

Procedure & Observations

TASK 1: [Directory Navigation]

Task Statement:

- [Create a directory called test_project in your home directory, then create subdirectories docs, scripts, and data inside it. Navigate to the scripts directory and display your current path.]

Explanation:

- [Use mkdir to create the wanted directory we can use cd to navigate and use pwd to show current path]

Command(s):

''' mkdir test_project cd test_project mkdir docs scripts data cd scripts pwd '''

Output:

TASK 2: [FIle Creation and Content]

Task Statement:

- [Create three files in the docs directory: readme.txt, notes.txt, and todo.txt. Add the text "Project documentation" to readme.txt and "Important notes" to notes.txt. Display the contents of both files.]

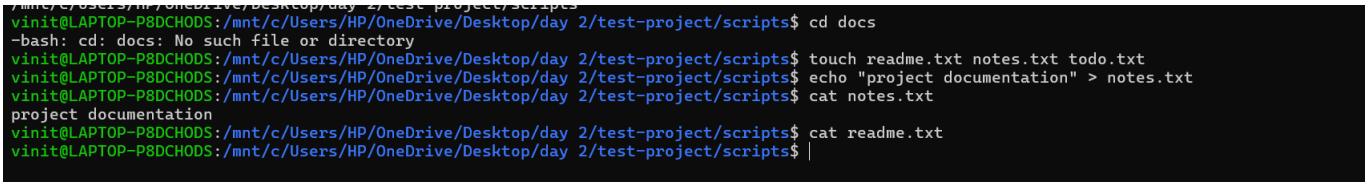
Explanation:

- [We can use touch to create empty files and using echo "text" > file.txt to add content to a file and using cat to display file contents]

Command(s):

```
cd docs
touch readme.txt notes.txt todo.txt
echo "Project documentation" > readme.txt
echo "Important notes" > notes.txt
cat notes.txt
cat readme.txt
```

Output:



TASK 3: [FIle Operations]

Task Statement:

- [Copy readme.txt to the data directory and rename the copy to project_info.txt. Then move todo.txt from docs to scripts directory.]

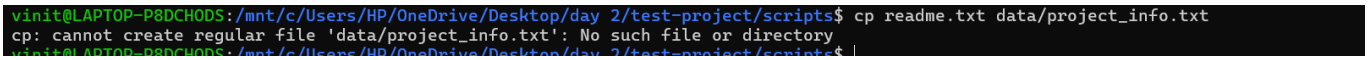
Explanation:

- [- We can use the cp source destination to copy files and using the mv oldname newname to rename files also using the same command mv file directory/ to move files to another directory we can also combine copy and rename: cp file.txt newdir/newname.txt]

Command(s):

```
cp readme.txt data/project_info.txt
```

Output:



TASK 4: [FIle Permissions]

Task Statement:

- [Create a shell script file called backup.sh in the scripts directory. Add the content #!/bin/bash and echo "Backup complete" to it. Make the file executable only for the owner.]

Explanation:

- [Using chmod u+x filename we can make the file executable for user only using ls -l to check for permissions also script files typically need executable permission to run]

Command(s):

```
cd scripts
touch backup.sh > echo "Backup complete"
chmod u+x backup.sh
```

Output:

```
vinit@LAPTOP-P8DCHODS:/mnt/c/Users/HP/OneDrive/Desktop/day 2/test-project/scripts$ cp readme.txt data/project_info.txt
cp: cannot create regular file 'data/project_info.txt': No such file or directory
```


```
vinit@LAPTOP-P8DCHODS:/mnt/c/Users/HP/OneDrive/Desktop/day 2/test-project/scripts$ cp readme.txt data/project_info.txt
cp: cannot create regular file 'data/project_info.txt': No such file or directory
```


TASK 5: [FIle Viewing]

Task Statement:

- [Create a file called numbers.txt with numbers 1 to 20 (each on a new line). Display only the first 5 lines, then only the last 3 lines, then search for lines containing the number "1".]

Explanation:

- [I can quickly generate a list of numbers by running seq 1 20 > numbers.txt. To check the first few numbers, I use head -n 5 to see the first 5 lines, and tail -n 3 to see the last 3 lines. If I want to find all numbers containing a “1”, I can use grep "1". Alternatively, I could create the list manually by using multiple echo commands.]

Command(s):

```
seq 1 20 > numbers.txt
head -n 5
tail -n 3
grep "1"
```

Output:


```
vinit@LAPTOP-P8DCHODS:/mnt/c/Users/HP/OneDrive/Desktop/day 2/test-project/scripts$ cp readme.txt data/project_info.txt
cp: cannot create regular file 'data/project_info.txt': No such file or directory
```


TASK 6: [Text Editing]

Task Statement:

- [Using nano, create a file called config.txt with the following content:

Database=localhost Port=5432 Username=admin

```
```

Save the file and then display its contents.]

Explanation:

- [I open a file in Nano using nano filename.txt and type my content normally. Once I'm done, I press Ctrl+O to save the file and Ctrl+X to exit Nano. After that, I use cat to check the contents and make sure everything was saved correctly.]

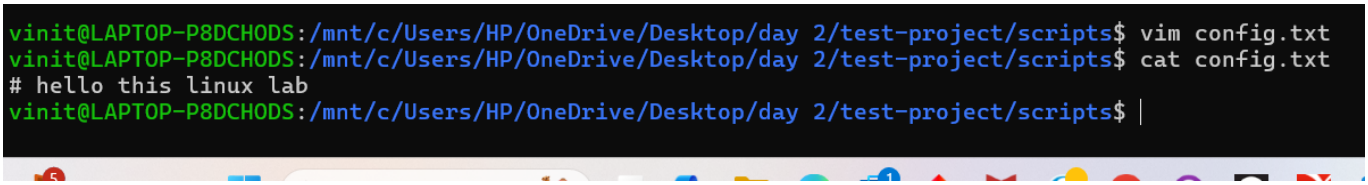
Command(s):

```
vim config.txt
cat config.txt
```

Alternatively

```
nano config.txt
cat config.txt
```

Output:



TASK 7: [System Information]

Task Statement:

- [Create a file called system_info.txt that contains: your username, current date, your current directory, and disk usage information in human-readable format.]

Explanation:

- [I can use whoami to check my username, date to see the current date, and pwd to know my current directory. To check disk usage, I use df -h. I can save the output of any command to a file by using redirection like command >> filename.txt. If I want to add labels, I use echo like this: echo "Username:" >> file.txt.]

Command(s):

```
cd scripts
touch system_info.txt
echo "Username:" >> system_info.txt
whoami >> system_info.txt
echo "Date:" >> system_info.txt
date >> system_info.txt
echo "Current Directory:" >> system_info.txt
pwd >> system_info.txt
echo "Disk Usage:" >> system_info.txt
df -h >> system_info.txt
```

Output:

```
vinit@LAPTOP-P8DCHODS:/mnt/c/Users/HP/OneDrive/Desktop/day 2/test-project/scripts$ vim config.txt
vinit@LAPTOP-P8DCHODS:/mnt/c/Users/HP/OneDrive/Desktop/day 2/test-project/scripts$ cat config.txt
# hello this linux lab
vinit@LAPTOP-P8DCHODS:/mnt/c/Users/HP/OneDrive/Desktop/day 2/test-project/scripts$
```

TASK 8: [File Organisation]

Task Statement:

- [In your test_project directory, create a backup folder. Copy all .txt files from all subdirectories into this backup folder. Then list all files in the backup folder with detailed information.]

Explanation:

- [I can use find . -name "*.txt" to locate all .txt files. Alternatively, I can navigate to each directory and copy files manually. To copy multiple files at once, I use cp file1.txt file2.txt destination/. If I want detailed information about the files, I use ls -la. The wildcard *.txt helps me match all files that end with .txt.]

Command(s):

```
cp test_project/data/project_info.txt    test_project/docs/notes.txt
test_project/docs/readme.txt    test_project/docs/todo.txt
test_project/scripts/config.txt    test_project/scripts/numbers.txt
test_project/scripts/system_info.txt    test_project/scripts/todo.txt    backup/
```

Output:

```
vinit@LAPTOP-P8DCHODS:/mnt/c/Users/HP/OneDrive/Desktop/day 2/test-project/scripts$ cp test-project/data/project_info.txt
cp: missing destination file operand after 'test-project/data/project_info.txt'
Try 'cp --help' for more information.
vinit@LAPTOP-P8DCHODS:/mnt/c/Users/HP/OneDrive/Desktop/day 2/test-project/scripts$ cp test-project/data/project_info.txt  test_project/docs/readme.txt  test
_project/scripts/todo.txt  test_project/scripts/config.txt
cp: target 'test_project/scripts/config.txt': No such file or directory
vinit@LAPTOP-P8DCHODS:/mnt/c/Users/HP/OneDrive/Desktop/day 2/test-project/scripts$
```