



GATI SHAKTI VISHWAVIDYALAYA

ASSIGNMENT 1

Linux File Handling and Directory Management

Name: AAYUSH PRAVEEN

Roll Number: 24AI002

Course: Operating Systems

Instructor: Mr. Praveen Kumar Chandaliya

February 15, 2026

Contents

1	Objective	2
2	Part 1: Creating a File and Adding Content	2
2.1	Task Description	2
2.2	Commands Used	2
2.3	Explanation	2
2.4	Screenshot	2
3	Part 2: Displaying and Counting File Content	3
3.1	Task Description	3
3.2	Commands Used	3
3.3	Explanation	3
3.4	Screenshot	3
4	Part 3: Creating Project Directory Structure	3
4.1	Task Description	3
4.2	Commands Used	4
4.3	Explanation	4
4.4	Screenshot	4
5	Part 4: Moving Files Using Wildcards	4
5.1	Task Description	4
5.2	Commands Used	5
5.3	Explanation	5
5.4	Screenshot	5
6	Conclusion	5

1 Objective

The objective of this assignment is to understand and practice basic Linux commands related to file handling and directory management. The tasks include creating files and directories, viewing file contents, counting text statistics, organizing project structures, and moving files using wildcards.

2 Part 1: Creating a File and Adding Content

2.1 Task Description

- Create a directory named `Assignment1`.
- Create a file named `notes.txt` inside it.
- Add the text “UNIX is fun!” to the file using a single command.

2.2 Commands Used

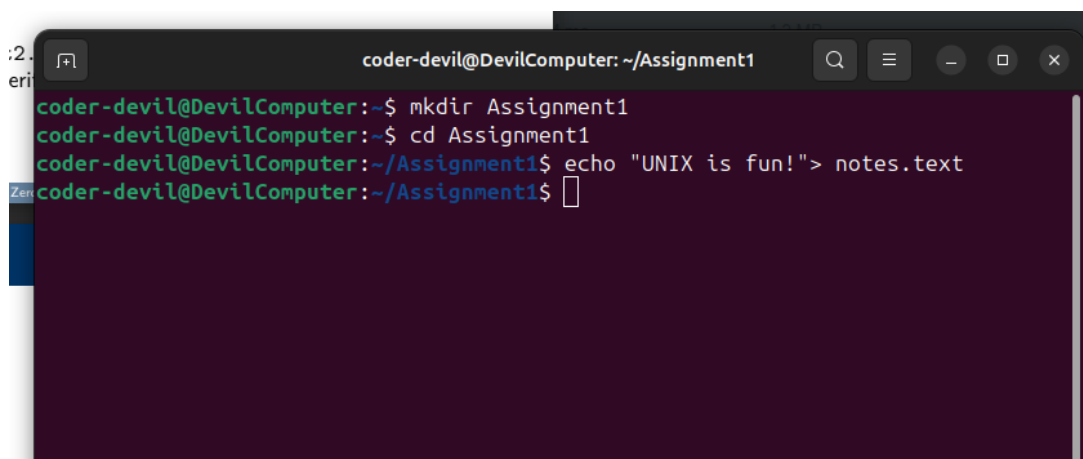
```
mkdir Assignment1
cd Assignment1
echo "UNIX is fun!" > notes.txt
cat notes.txt
```

2.3 Explanation

- `mkdir` creates a new directory.
- `cd` changes the current directory.
- `echo` prints text to the terminal.
- `>` redirects output into a file.
- `cat` displays file contents.

2.4 Screenshot

Screenshot of terminal below:

A screenshot of a terminal window with a dark background. The window title is "coder-devil@DevilComputer: ~/Assignment1". The terminal shows the following commands and their outputs:
1. `coder-devil@DevilComputer:~$ mkdir Assignment1`
2. `coder-devil@DevilComputer:~$ cd Assignment1`
3. `coder-devil@DevilComputer:~/Assignment1$ echo "UNIX is fun!"> notes.txt`
4. `coder-devil@DevilComputer:~/Assignment1$` followed by a cursor.
The terminal also shows some partial text from the previous page on the left edge:
:2.
eri
Zen

3 Part 2: Displaying and Counting File Content

3.1 Task Description

- Display the contents of `notes.txt`.
- Count the number of lines, words, and characters.

3.2 Commands Used

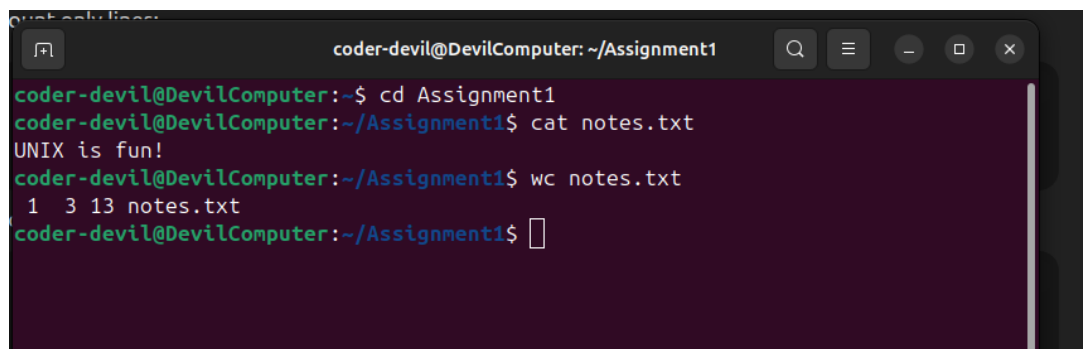
```
cat notes.txt
wc notes.txt
wc -l notes.txt
wc -w notes.txt
wc -c notes.txt
```

3.3 Explanation

- `wc` stands for word count.
- `-l` counts lines.
- `-w` counts words.
- `-c` counts characters (bytes).

3.4 Screenshot

Screenshot of terminal below:

A screenshot of a terminal window with a dark background. The window title is "coder-devil@DevilComputer: ~/Assignment1". The terminal shows the following commands and output:

```
coder-devil@DevilComputer:~$ cd Assignment1
coder-devil@DevilComputer:~/Assignment1$ cat notes.txt
UNIX is fun!
coder-devil@DevilComputer:~/Assignment1$ wc notes.txt
 1  3 13 notes.txt
coder-devil@DevilComputer:~/Assignment1$
```

4 Part 3: Creating Project Directory Structure

4.1 Task Description

Create the following structure:

```
ProjectWorkspace
|
|--Docs
|   |--README.txt
```

```
|-- Code
|   |--main.py
|-- Resources
```

4.2 Commands Used

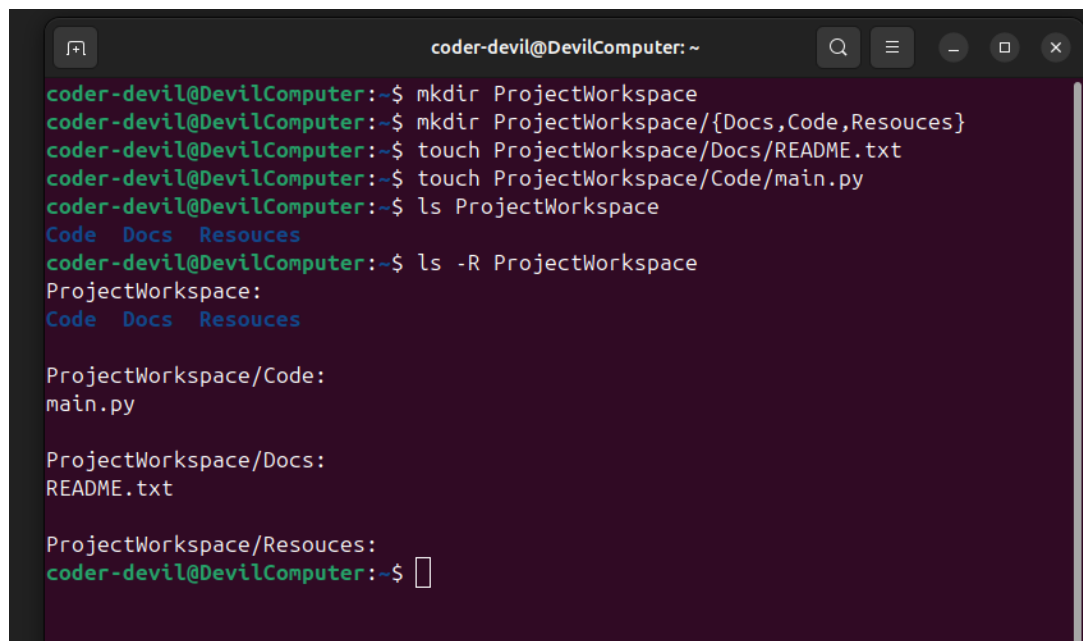
```
mkdir ProjectWorkspace
mkdir ProjectWorkspace/{Docs,Code,Resources}
touch ProjectWorkspace/Docs/README.txt
touch ProjectWorkspace/Code/main.py
ls -R ProjectWorkspace
```

4.3 Explanation

- Brace expansion allows creating multiple directories at once.
- `touch` creates empty files.
- `ls -R` lists directory contents recursively.

4.4 Screenshot

Screenshot of terminal below:

A terminal window titled 'coder-devil@DevilComputer: ~' with standard window controls. The terminal shows the following commands and output:

```
coder-devil@DevilComputer:~$ mkdir ProjectWorkspace
coder-devil@DevilComputer:~$ mkdir ProjectWorkspace/{Docs,Code,Resources}
coder-devil@DevilComputer:~$ touch ProjectWorkspace/Docs/README.txt
coder-devil@DevilComputer:~$ touch ProjectWorkspace/Code/main.py
coder-devil@DevilComputer:~$ ls ProjectWorkspace
Code  Docs  Resources
coder-devil@DevilComputer:~$ ls -R ProjectWorkspace
ProjectWorkspace:
Code  Docs  Resources

ProjectWorkspace/Code:
main.py

ProjectWorkspace/Docs:
README.txt

ProjectWorkspace/Resources:
coder-devil@DevilComputer:~$
```

5 Part 4: Moving Files Using Wildcards

5.1 Task Description

- Create files:
 - `old_report1.txt`

- old_report2.txt
- current_report.txt
- Create directory OldReports.
- Move all files starting with “old” into OldReports.

5.2 Commands Used

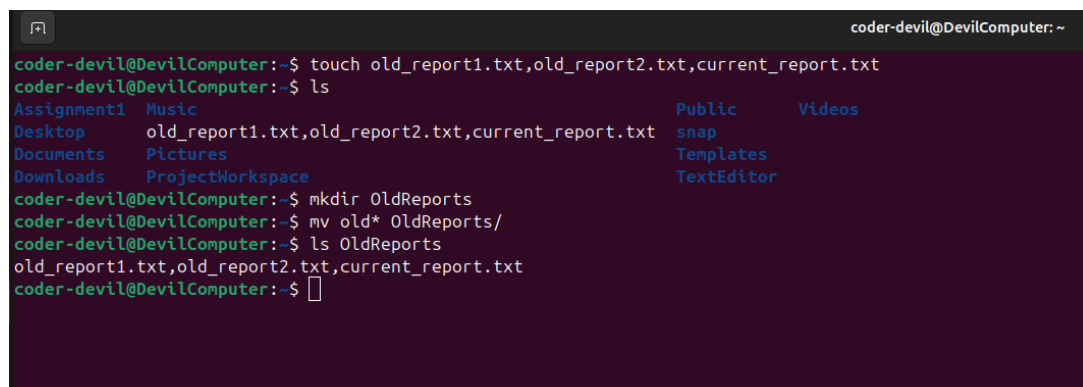
```
touch old_report1.txt old_report2.txt current_report.txt
mkdir OldReports
mv old* OldReports/
ls
ls OldReports
```

5.3 Explanation

- mv moves files.
- * is a wildcard that matches any sequence of characters.
- old* matches all files beginning with “old”.

5.4 Screenshot

Screenshot of terminal below:



```

coder-devil@DevilComputer: ~
coder-devil@DevilComputer:~$ touch old_report1.txt,old_report2.txt,current_report.txt
coder-devil@DevilComputer:~$ ls
Assignment1  Music          Public         Videos
Desktop      old_report1.txt,old_report2.txt,current_report.txt  snap
Documents    Pictures        Templates
Downloads    ProjectWorkspace  TextEditor
coder-devil@DevilComputer:~$ mkdir OldReports
coder-devil@DevilComputer:~$ mv old* OldReports/
coder-devil@DevilComputer:~$ ls OldReports
old_report1.txt,old_report2.txt,current_report.txt
coder-devil@DevilComputer:~$ 

```

6 Conclusion

In this assignment, we practiced fundamental Linux commands for directory creation, file manipulation, content viewing, file statistics analysis, and file organization using wildcards. These commands form the foundation for efficient file system management in UNIX/Linux environments.