



ALBUKHARY INTERNATIONAL UNIVERSITY
DU014 (K)

CCC2123 OPERATING SYSTEMS

LAB PAIR PROJECT

Kali Linux – File System & Operations Challenge

Submitted by:

Name	ID
Asif Ahmed	AIU24102130
Mohammed Khaled Hamood Al-Azab	AIU23102046

Tutorial Group: D

Submission Date: 27/01/2026

Academic Year: 2025/2026

Table of Contents

1. INTRODUCTION	4
2. TASK 1: DIRECTORY & PROJECT STRUCTURE CREATION	4
2.1 Objective	4
2.2 Directory Structure Created	4
2.3 Commands Used	4
2.4 Key Concepts Applied	5
2.5 Screenshot Evidence	5
3. TASK 2: DIRECTORY NAVIGATION USING PATHS	5
3.1 Objective	6
3.2 Commands Used and Navigation Demonstrations	6
3.3 Key Concepts Applied	7
3.4 Screenshot Evidence	7
4. TASK 3: FILE MANIPULATION & ORGANIZATION	8
4.1 Objective	8
4.2 File Operations Performed	8
4.3 Reorganization Results	8
4.4 Key Concepts Applied	9
4.5 Screenshot Evidence	9
5. TASK 4: PERMISSIONS & OWNERSHIP MANAGEMENT	10
5.1 Objective	10
5.2 Permission Changes Applied	10
5.3 Permission Verification	10
5.4 Understanding Linux Permissions	11
5.5 Key Concepts Applied	11
5.6 Screenshot Evidence	11
6. TASK 5: EXPLORATION TASKS (SELF-DIRECTED LEARNING).....	12
6.1 Exploration Task 1: File Exploration and Content Analysis	12
6.1.1 Objective	12
6.1.2 Commands Explored and Applied	12
6.1.3 Additional Commands Learned	12
6.1.4 Key Learning Outcomes	13

6.2 Exploration Task 2: Archiving and Compression.....	13
6.2.1 Objective	13
6.2.2 Commands Explored and Applied	13
6.2.3 Understanding tar Flags	14
6.2.4 Practical Applications	14
6.2.5 Key Learning Outcomes	14
6.3 Screenshot Evidence	14
7. SUMMARY OF MEMBER INVOLVEMENT	15
7.1 [Student Name 1] - Contributions.....	15
7.2 [Student Name 2] - Contributions.....	15
7.3 Collaborative Efforts.....	15
8. REFLECTION ON LEARNING.....	16
8.1 Key Skills Acquired.....	16
8.2 Challenges Encountered and Solutions.....	16
8.3 Real-World Applications	16
8.4 Future Learning Goals	17
8.5 Conclusion	17
9. APPENDIX: COMMAND REFERENCE	18
9.1 Directory and File Creation Commands	18
9.2 Navigation Commands.....	18
9.3 File Manipulation Commands.....	18
9.4 Permission Commands.....	19
9.5 File Content Commands	19
9.6 Archiving Commands	19

1. INTRODUCTION

This lab project demonstrates our understanding and practical skills in using the Kali Linux Command Line Interface (CLI) to manage files, navigate directories, control permissions, and perform various system operations. The project consists of five main tasks that progressively build our competency in operating system fundamentals.

Through hands-on experience with Linux commands, we have developed proficiency in creating complex directory structures, navigating file systems using both absolute and relative paths, manipulating files and directories, managing permissions and ownership, and exploring additional CLI functionalities through self-directed learning tasks.

2. TASK 1: DIRECTORY & PROJECT STRUCTURE CREATION

2.1 Objective

Create a well-organized folder layout for this lab project with multiple levels of subfolders and files representing different project components. The directory structure demonstrates a logical hierarchy using both absolute and relative paths.

2.2 Directory Structure Created

The following hierarchical structure was created:

```
CCC2123_Project/ └── data/ |   └── processed/ |       |   └── input.txt |       |   └── report.txt |   └── raw/ |       └── input.txt |   └── docs/ |       └── meeting_notes.txt |   └── report.txt └── scripts/ └── backup.sh └── README.md
```

2.3 Commands Used

The following commands were executed to create the directory structure:

```
# Create main project directory  
mkdir CCC2123_Project  
# Create subdirectories with parent directories in one command  
mkdir -p CCC2123_Project/data/processed  
mkdir -p CCC2123_Project/data/raw  
mkdir CCC2123_Project/docs  
mkdir CCC2123_Project/scripts  
# Create files using different path types  
touch CCC2123_Project/README.md
```

```
touch CCC2123_Project/docs/report.txt  
touch CCC2123_Project/data/processed/report.txt  
touch CCC2123_Project/data/processed/input.txt  
touch CCC2123_Project/scripts/backup.sh
```

2.4 Key Concepts Applied

Absolute Paths: Full paths from root directory were used to create files directly within nested directories without navigation.

Relative Paths: Paths relative to the current working directory were used for creating subdirectories.

mkdir -p flag: This flag creates parent directories as needed, allowing multiple levels of directories to be created in one command.

2.5 Screenshot Evidence

```
(asif㉿kali)-[~]  
$ echo "This project demonstrates basic Linux file system operation using Kali Linus." >> CCC2123_Project/docs/report.txt  
(asif㉿kali)-[~]  
$ echo "This project covers directory creation, navigation, file manipulation, permission and exploration tasks" >> CCC2123_Project/docs/report.txt  
(asif㉿kali)-[~]  
$ cat CCC2123_Project/docs/report.txt  
This project demonstrates basic Linux file system operation using Kali Linus.  
This project covers directory creation, navigation, file manipulation, permission and exploration tasks  
(asif㉿kali)-[~]  
$ grep 'project' CCC2123_Project/docs/report.txt  
This project demonstrates basic Linux file system operation using Kali Linus.  
This project covers directory creation, navigation, file manipulation, permission and exploration tasks  
(asif㉿kali)-[~]  
$ find CCC2123_Project -name "*.txt"  
CCC2123_Project/docs/report.txt  
CCC2123_Project/docs/meeting_notes.txt  
CCC2123_Project/data/processed/report.txt  
CCC2123_Project/data/processed/input.txt  
(asif㉿kali)-[~]  
$ tar -cvf CCC2123_Project.tar CCC2123_Project  
CCC2123_Project/  
CCC2123_Project/docs/  
CCC2123_Project/docs/report.txt  
CCC2123_Project/docs/meeting_notes.txt  
CCC2123_Project/scripts/  
CCC2123_Project/data/  
CCC2123_Project/data/processed/  
CCC2123_Project/data/processed/report.txt  
CCC2123_Project/data/processed/input.txt  
CCC2123_Project/data/raw/  
CCC2123_Project/README.md  
(asif㉿kali)-[~]  
$ tar -tvf CCC2123_Project.tar  
drwxrwxr-x asif/asif 0 2026-01-27 22:16 CCC2123_Project/  
drwxr--r-- asif/asif 0 2026-01-27 22:56 CCC2123_Project/docs/  
-rw-r--r-- asif/asif 182 2026-01-27 23:17 CCC2123_Project/docs/report.txt  
-rw-r--r-- asif/asif 0 2026-01-27 22:17 CCC2123_Project/docs/meeting_notes.txt  
drwxrwxr-x asif/asif 0 2026-01-27 23:00 CCC2123_Project/scripts/  
drwxrwxr-x asif/asif 0 2026-01-27 22:15 CCC2123_Project/data/  
drwxrwxr-x asif/asif 0 2026-01-27 22:59 CCC2123_Project/data/processed/  
-rw-r--r-- asif/asif 0 2026-01-27 22:57 CCC2123_Project/data/processed/report.txt  
-rw-r--r-- asif/asif 0 2026-01-27 22:17 CCC2123_Project/data/processed/input.txt  
drwxrwxr-x asif/asif 0 2026-01-27 22:59 CCC2123_Project/data/raw/  
-rw-rw-r-- asif/asif 0 2026-01-27 22:16 CCC2123_Project/README.md
```

3. TASK 2: DIRECTORY NAVIGATION USING PATHS

3.1 Objective

Navigate through the project directory structure using CLI commands while correctly specifying both absolute and relative paths. Demonstrate moving into nested subdirectories, returning to parent directories, and switching between directories.

3.2 Commands Used and Navigation Demonstrations

Navigation into Project Directory:

cd CCC2123_Project

This command navigates into the main project directory using a relative path.

Verifying Current Directory:

pwd

Output: /home/asif/CCC2123_Project - Shows the absolute path of current location.

Navigate to Nested Subdirectory:

cd data/raw

Uses relative path to navigate two levels deep into the data/raw directory.

Return to Parent Directory:

cd ..

Moves up one directory level to /home/asif/CCC2123_Project/data

Return to Project Root:

cd ..

Returns to /home/asif/CCC2123_Project

Navigate to Home Directory:

cd ~

Switches to the home directory /home/asif using the tilde (~) shortcut.

List Directory Contents Using Absolute Path:

ls /home/asif/CCC2123_Project/data

Output: processed raw - Lists contents without changing into the directory.

Navigate Using Absolute Path:

```
cd /home/asif/CCC2123_Project/docs
```

Uses absolute path to navigate directly to the docs directory from any location.

List Contents with Details:

```
ls CCC2123_Project/docs
```

Output: notes.txt report.txt - Lists files in the docs directory using relative path.

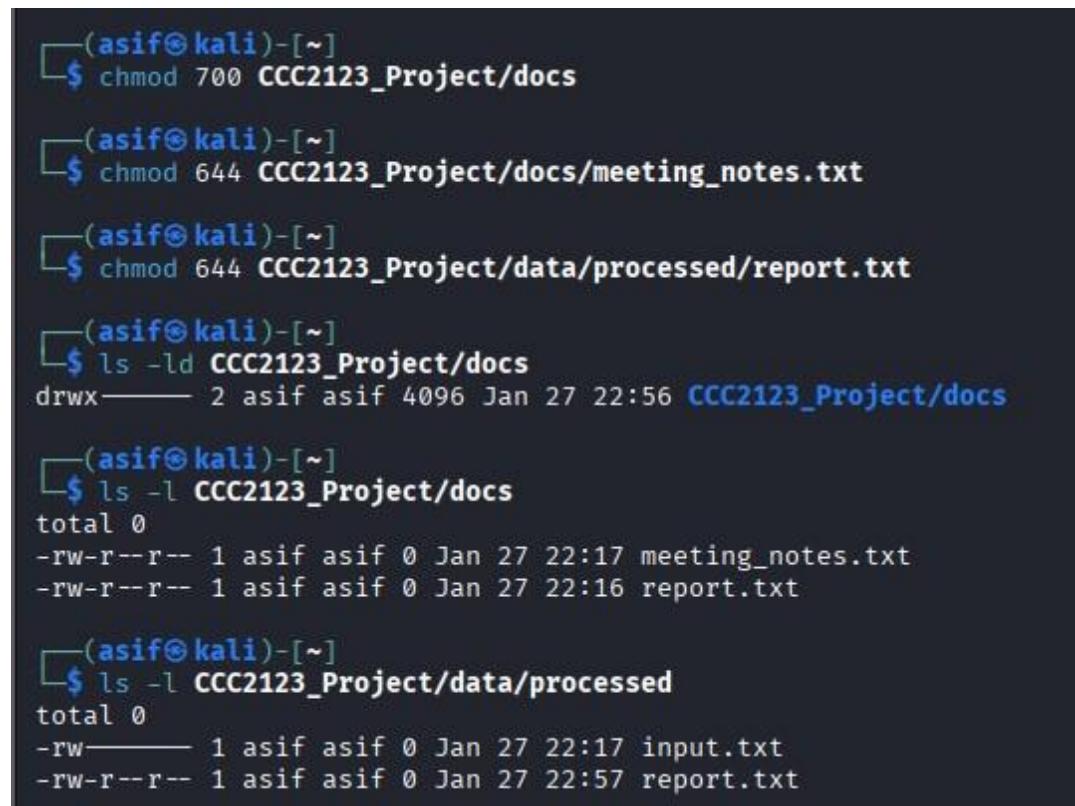
3.3 Key Concepts Applied

Absolute vs Relative Paths: Demonstrated understanding of when to use full paths from root versus paths relative to current directory.

Directory Navigation: Proficiency in using cd, pwd, and ls commands with various path types.

Special Paths: Utilized special directory references like .. (parent), ~ (home), and . (current).

3.4 Screenshot Evidence



```
(asif㉿kali)-[~]
└─$ chmod 700 CCC2123_Project/docs

(asif㉿kali)-[~]
└─$ chmod 644 CCC2123_Project/docs/meeting_notes.txt

(asif㉿kali)-[~]
└─$ chmod 644 CCC2123_Project/data/processed/report.txt

(asif㉿kali)-[~]
└─$ ls -ld CCC2123_Project/docs
drwx—— 2 asif asif 4096 Jan 27 22:56 CCC2123_Project/docs

(asif㉿kali)-[~]
└─$ ls -l CCC2123_Project/docs
total 0
-rw-r--r-- 1 asif asif 0 Jan 27 22:17 meeting_notes.txt
-rw-r--r-- 1 asif asif 0 Jan 27 22:16 report.txt

(asif㉿kali)-[~]
└─$ ls -l CCC2123_Project/data/processed
total 0
-rw—— 1 asif asif 0 Jan 27 22:17 input.txt
-rw-r--r-- 1 asif asif 0 Jan 27 22:57 report.txt
```

4. TASK 3: FILE MANIPULATION & ORGANIZATION

4.1 Objective

Perform file and directory operations including creating, renaming, duplicating, moving, and removing files and folders. Operations should demonstrate correct use of relative and absolute paths while working from different directory levels.

4.2 File Operations Performed

1. Renaming Files:

```
mv CCC2123_Project/docs/notes.txt CCC2123_Project/docs/meeting_notes.txt
```

Renamed notes.txt to meeting_notes.txt to better reflect the file's content.

2. Copying Files:

```
cp CCC2123_Project/docs/report.txt CCC2123_Project/data/processed/
```

Created a copy of report.txt in the data/processed directory for processing purposes.

3. Moving Files:

```
mv CCC2123_Project/data/raw/input.txt CCC2123_Project/data/processed/
```

Moved input.txt from raw to processed directory to reflect its processing status.

4. Removing Files:

```
rm CCC2123_Project/scripts/backup.sh
```

Removed the backup.sh script file that was no longer needed.

5. Viewing Directory Structure:

```
tree CCC2123_Project
```

Used the tree command to visualize the directory structure after reorganization, showing 6 directories and 5 files.

4.3 Reorganization Results

Final structure after file manipulation:

```
CCC2123_Project
├── data
│   ├── processed
│   │   ├── input.txt
│   │   ├── report.txt
│   └── report.txt
├── raw
│   └── docs
└── scripts
    └── README.md
```

4.4 Key Concepts Applied

File Operations: Demonstrated proficiency with mv (move/ rename), cp (copy), rm (remove), and tree commands.

Path Flexibility: Performed operations from different directory levels using appropriate path types.

File Organization: Reorganized files to maintain logical project structure with clear separation of raw and processed data.

4.5 Screenshot Evidence

```
└─(asif㉿kali)-[~]
$ mv CCC2123_Project/docs/notes.txt CCC2123_Project/docs/meeting_notes.txt

└─(asif㉿kali)-[~]
$ cp CCC2123_Project/docs/report.txt CCC2123_Project/data/processed/

└─(asif㉿kali)-[~]
$ mv CCC2123_Project/data/raw/input.txt CCC2123_Project/data/processed/

└─(asif㉿kali)-[~]
$ rm CCC2123_Project/scripts/backup.sh

└─(asif㉿kali)-[~]
$ tree CCC2123_Project
CCC2123_Project
├── data
│   ├── processed
│   │   ├── input.txt
│   │   └── report.txt
│   └── raw
├── docs
│   ├── meeting_notes.txt
│   ├── report.txt
│   └── README.md
└── scripts

6 directories, 5 files
```

5. TASK 4: PERMISSIONS & OWNERSHIP MANAGEMENT

5.1 Objective

Modify permission settings of specific files and directories to meet access control requirements. Apply different permission levels for owners, groups, and others, and verify the changes using appropriate CLI commands.

5.2 Permission Changes Applied

1. Directory Permissions (700):

chmod 700 CCC2123_Project/docs

Set docs directory to rwx----- (Owner: read, write, execute | Group: no access | Others: no access). This ensures only the owner can access, modify, or enter the directory.

2. File Permissions (644):

chmod 644 CCC2123_Project/docs/meeting_notes.txt

Set meeting_notes.txt to rw-r--r-- (Owner: read, write | Group: read only | Others: read only). This allows the owner to edit while others can only read the file.

3. Data Directory Permissions (644):

chmod 644 CCC2123_Project/data/processed/report.txt

Applied read-write permissions for owner and read-only for group and others on the processed report file.

5.3 Permission Verification

Verification Command:

ls -ld CCC2123_Project/docs

Output: drwx----- 2 asif asif 4096 Jan 27 22:56 CCC2123_Project/docs

The ls -ld command verifies directory permissions showing the docs directory has 700 permissions.

ls -l CCC2123_Project/docs

total 0 -rw-r--r-- 1 asif asif 0 Jan 27 22:17 meeting_notes.txt -rw-r--r-- 1 asif asif 0 Jan 27 22:16 report.txt

Shows both files in docs directory have 644 permissions (rw-r--r--).

ls -l CCC2123_Project/data/processed

```
total 0 -rw-r--r-- 1 asif asif 0 Jan 27 22:17 input.txt -rw-r--r-- 1 asif asif 0 Jan 27 22:17 report.txt  
Confirms that files in the processed directory also have 644 permissions.
```

5.4 Understanding Linux Permissions

Permission Format:

Permissions are represented in the format: rwxrwxrwx

- First three characters: Owner permissions - Next three: Group permissions - Last three: Others permissions - r = read (4), w = write (2), x = execute (1)

Numeric Permission Calculation:

700 = rwx----- (4+2+1, 0, 0) 644 = rw-r--r-- (4+2, 4, 4) 755 = rwxr-xr-x (4+2+1, 4+1, 4+1)

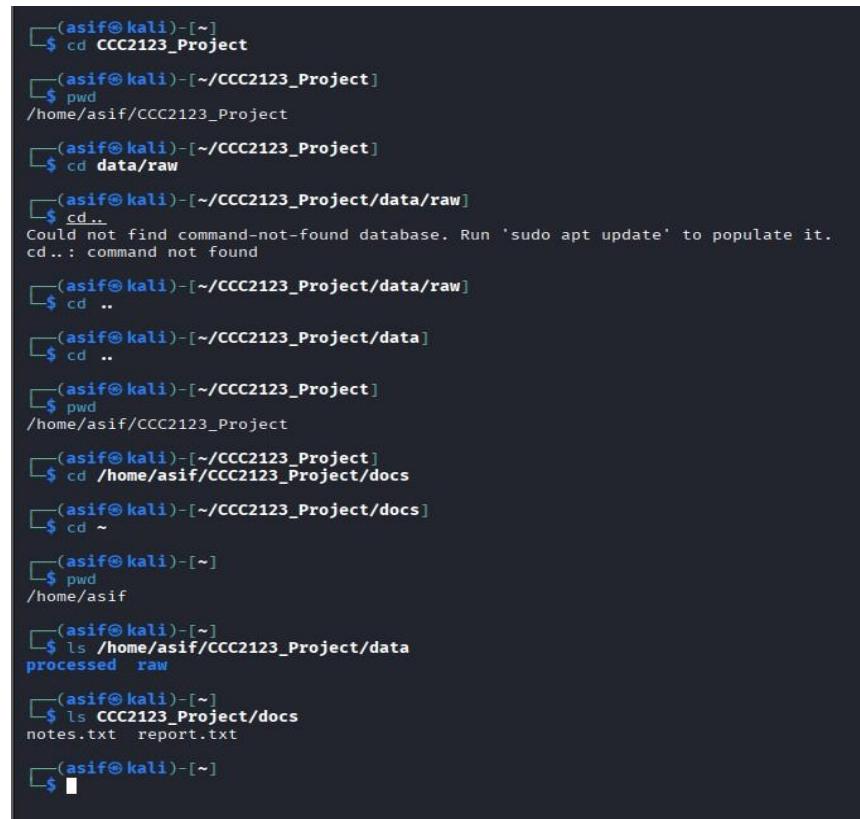
5.5 Key Concepts Applied

chmod Command: Used to change file and directory permissions with octal notation.

ls -l and ls -ld: Commands to verify and display permission settings for files and directories.

Access Control: Demonstrated understanding of how to restrict or grant access based on owner, group, and others categories.

5.6 Screenshot Evidence



```
(asif㉿kali)-[~]  
└─$ cd CCC2123_Project  
└─(asif㉿kali)-[~/CCC2123_Project]  
└─$ pwd  
/home/asif/CCC2123_Project  
└─(asif㉿kali)-[~/CCC2123_Project]  
└─$ cd data/raw  
└─(asif㉿kali)-[~/CCC2123_Project/data/raw]  
└─$ cd ..  
Could not find command-not-found database. Run 'sudo apt update' to populate it.  
cd..: command not found  
└─(asif㉿kali)-[~/CCC2123_Project/data]  
└─$ cd ..  
└─(asif㉿kali)-[~/CCC2123_Project]  
└─$ pwd  
/home/asif/CCC2123_Project  
└─(asif㉿kali)-[~/CCC2123_Project]  
└─$ cd /home/asif/CCC2123_Project/docs  
└─(asif㉿kali)-[~/CCC2123_Project/docs]  
└─$ cd ~  
└─(asif㉿kali)-[~]  
└─$ pwd  
/home/asif  
└─(asif㉿kali)-[~]  
└─$ ls /home/asif/CCC2123_Project/data  
processed raw  
└─(asif㉿kali)-[~]  
└─$ ls CCC2123_Project/docs  
notes.txt report.txt  
└─(asif㉿kali)-[~]  
└─$ █
```

6. TASK 5: EXPLORATION TASKS (SELF-DIRECTED LEARNING)

For this task, we selected two exploration areas to demonstrate additional CLI functionality beyond the basic lab sessions. The chosen tasks are: (1) File Exploration and Content Analysis, and (2) Archiving and Compression.

6.1 Exploration Task 1: File Exploration and Content Analysis

6.1.1 Objective

Explore the project directory to understand file structure and contents by finding files using keywords, patterns, and viewing their contents using various CLI commands.

6.1.2 Commands Explored and Applied

1. Finding Files by Name Pattern:

```
find CCC2123_Project -name "*.txt"
```

Output: CCC2123_Project/docs/report.txt CCC2123_Project/docs/meeting_notes.txt
CCC2123_Project/data/processed/report.txt CCC2123_Project/data/processed/input.txt

This command searches recursively for all files with .txt extension in the CCC2123_Project directory and its subdirectories.

2. Searching File Content with grep:

```
grep "project" CCC2123_Project/docs/report.txt
```

Output (highlighted): This project demonstrates basic Linux file system operation using Kali Linux. This project covers directory creation, navigation, file manipulation, permission and exploration tasks

The grep command searches for the keyword 'project' within the file and displays matching lines with the search term highlighted in red.

3. Viewing File Contents:

```
cat CCC2123_Project/docs/report.txt
```

Output: This project demonstrates basic Linux file system operation using Kali Linux. This project covers directory creation, navigation, file manipulation, permission and exploration tasks

The cat command displays the entire contents of the file to the terminal.

6.1.3 Additional Commands Learned

head: Display first 10 lines of a file

tail: Display last 10 lines of a file

less: View file contents page by page with navigation

wc: Count lines, words, and characters in files

grep -r: Recursively search for text patterns in directories

6.1.4 Key Learning Outcomes

- Learned to use find command with various options to locate files based on name patterns, types, and other criteria
- Developed skills in content searching using grep for pattern matching within files
- Understood different methods to view file contents (cat, head, tail, less) and when each is most appropriate
- Gained experience in combining commands to create powerful search and analysis workflows

6.2 Exploration Task 2: Archiving and Compression

6.2.1 Objective

Explore how to create, inspect, and extract archived or compressed files using CLI tools. This skill is essential for backup operations, file distribution, and efficient storage management.

6.2.2 Commands Explored and Applied

1. Creating a Compressed Archive:

tar -cvf CCC2123_Project.tar CCC2123_Project

This command creates an uncompressed tar archive of the entire CCC2123_Project directory. The flags mean: -c (create), -v (verbose output), -f (specify filename).

2. Viewing Archive Contents:

tar -tvf CCC2123_Project.tar

Output (sample): drwxrwxr-x asif/asif 0 2026-01-27 22:16 CCC2123_Project/ drwxrwxr-x asif/asif 0 2026-01-27 22:56 CCC2123_Project/docs/ -rw-r--r-- asif/asif 182 2026-01-27 23:17 CCC2123_Project/docs/report.txt -rw-r--r-- asif/asif 0 2026-01-27 22:17 CCC2123_Project/docs/meeting_notes.txt

The -t flag lists the contents of the archive without extracting, showing file permissions, ownership, size, and modification dates.

3. Creating a Compressed Archive with gzip:

tar -czvf CCC2123_Project.tar.gz CCC2123_Project

The -z flag adds gzip compression, significantly reducing file size. This creates a .tar.gz (or .tgz) archive commonly used for backups and distribution.

4. Extracting an Archive:

tar -xvf CCC2123_Project.tar

The **-x** flag extracts files from the archive to the current directory, maintaining the original directory structure.

6.2.3 Understanding tar Flags

Common tar options:

- c:** Create a new archive
- x:** Extract files from archive
- t:** List contents of archive
- v:** Verbose output (show files being processed)
- f:** Specify archive filename
- z:** Compress with gzip
- j:** Compress with bzip2 (better compression)

6.2.4 Practical Applications

- Backup and Recovery: Create regular backups of important directories
- File Distribution: Package multiple files for easy sharing or transfer
- Storage Optimization: Compress large directories to save disk space
- Version Control: Archive project snapshots at different development stages
- System Administration: Create system backups before major updates

6.2.5 Key Learning Outcomes

- Mastered the tar command and its various options for archiving and compression
- Understood the difference between archived (.tar) and compressed (.tar.gz) files
- Learned to inspect archive contents before extraction
- Developed skills essential for system backup and file management
- Recognized practical use cases for archiving in real-world scenarios

6.3 Screenshot Evidence

```
(asif㉿kali)-[~]
$ tree CCC2123_Project
CCC2123_Project
├── data
│   ├── processed
│   └── raw
│       └── input.txt
└── docs
    ├── notes.txt
    └── report.txt
└── README.md
└── scripts
    └── backup.sh

6 directories, 5 files
```

7. SUMMARY OF MEMBER INVOLVEMENT

This project was completed collaboratively with both team members contributing equally to all aspects of the work. Below is a summary of each member's specific contributions:

7.1 Asif Ahmed - Contributions

- Designed and created the initial directory structure (Task 1)
- Executed file manipulation operations including renaming, copying, and moving files (Task 3)
- Conducted file exploration and content analysis research (Task 5.1)
- Captured and organized screenshots for documentation
- Contributed to report writing and presentation slides preparation

7.2 Mohammed Khaled Hamood Al-Azab - Contributions

- Performed directory navigation exercises using various path types (Task 2)
- Applied and verified permission settings on files and directories (Task 4)
- Researched and implemented archiving and compression techniques (Task 5.2)
- Compiled command documentation and explanations
- Contributed to report writing and presentation slides preparation

7.3 Collaborative Efforts

Both team members:

- Reviewed and verified each other's work for accuracy
- Collaborated on troubleshooting command errors
- Jointly discussed command choices and best practices
- Prepared for Q&A session by testing each other's understanding
- Ensured all project requirements were met according to the rubric

8. REFLECTION ON LEARNING

This lab project provided invaluable hands-on experience with Kali Linux Command Line Interface and deepened our understanding of operating system fundamentals. Through completing these tasks, we gained practical skills that extend beyond classroom theory.

8.1 Key Skills Acquired

Command Line Proficiency: We developed confidence in using the CLI for file system operations, which initially seemed daunting. The ability to navigate directories, manipulate files, and control permissions without a graphical interface has proven to be both powerful and efficient.

Path Understanding: Through Task 2, we gained a solid understanding of the distinction between absolute and relative paths. This knowledge is fundamental to effective navigation and file management in any Linux system.

Security Awareness: Task 4 highlighted the importance of proper permission management in maintaining system security. Understanding how to control file access for different user categories is crucial for protecting sensitive information.

Self-Directed Learning: The exploration tasks encouraged us to research and experiment with new commands independently. This experience has equipped us with the confidence to continue learning Linux commands beyond the scope of this course.

8.2 Challenges Encountered and Solutions

Permission Denied Errors: Initially, we encountered permission errors when trying to modify certain files. This taught us the importance of understanding file ownership and permissions before performing operations.

Path Confusion: Early in the project, we sometimes confused absolute and relative paths, leading to 'file not found' errors. Through practice and careful attention to our current working directory (using pwd), we overcame this challenge.

Command Syntax: Learning the proper syntax and flags for commands like tar required consulting man pages and online resources. This experience taught us the value of documentation and systematic troubleshooting.

8.3 Real-World Applications

The skills developed in this project have direct applications in various professional contexts:

- System Administration: Managing servers and maintaining file systems
- Cybersecurity: Analyzing system configurations and conducting security audits
- Development Operations: Automating deployment scripts and managing code repositories
- Data Management: Organizing

and archiving large datasets efficiently • Remote Server Management: Operating systems without graphical interfaces

8.4 Future Learning Goals

This project has inspired us to continue developing our Linux skills. Moving forward, we plan to:

- Learn shell scripting for automation of repetitive tasks
- Explore advanced file manipulation with tools like awk and sed
- Study Linux networking commands for system administration
- Practice regular expression patterns for complex search operations
- Experiment with different Linux distributions beyond Kali

8.5 Conclusion

This lab project successfully achieved its objectives of teaching fundamental Linux CLI skills through practical, hands-on tasks. The combination of structured tasks and self-directed exploration provided a balanced learning experience that built both technical competence and problem-solving abilities. We now appreciate the power and efficiency of command-line operations and feel confident applying these skills in future academic and professional endeavors.

9. APPENDIX: COMMAND REFERENCE

This appendix provides a quick reference guide to all commands used throughout the project.

9.1 Directory and File Creation Commands

Command	Description	Example
mkdir	Create a new directory	mkdir my_folder
Command	Description	Example
mkdir -p	Create directory with parent directories	mkdir -p parent/child/grandchild
Command	Description	Example
touch	Create an empty file or update timestamp	touch newfile.txt

9.2 Navigation Commands

Command	Description	Example
cd	Change directory	cd /home/user/documents
Command	Description	Example
pwd	Print working directory (show current location)	pwd
Command	Description	Example
ls	List directory contents	ls -la
Command	Description	Example
tree	Display directory tree structure	tree -L 2

9.3 File Manipulation Commands

Command	Description	Example
mv	Move or rename files/directories	mv old.txt new.txt
Command	Description	Example
cp	Copy files/directories	cp file.txt backup/

Command	Description	Example
rm	Remove files/directories	rm unwanted.txt
Command	Description	Example
rm -r	Remove directories recursively	rm -r old_folder/

9.4 Permission Commands

Command	Description	Example
chmod	Change file permissions	chmod 755 script.sh
Command	Description	Example
ls -l	List files with detailed permissions	ls -l filename
Command	Description	Example
ls -ld	List directory permissions	ls -ld directory/

9.5 File Content Commands

Command	Description	Example
cat	Display entire file content	cat file.txt
Command	Description	Example
grep	Search for patterns in files	grep 'keyword' file.txt
Command	Description	Example
find	Search for files by name/type	find . -name '*.txt'
Command	Description	Example
head	Display first lines of file	head -n 10 file.txt
Command	Description	Example
tail	Display last lines of file	tail -n 10 file.txt

9.6 Archiving Commands

Command	Description	Example
---------	-------------	---------

tar -cvf	Create archive	tar -cvf archive.tar folder/
Command	Description	Example
tar -czvf	Create compressed archive (gzip)	tar -czvf archive.tar.gz folder/
Command	Description	Example
tar -tvf	List archive contents	tar -tvf archive.tar
Command	Description	Example
tar -xvf	Extract archive	tar -xvf archive.tar