

**Module Code & Module Title**

Level 5 – CT5052NP

Assessment Type

Logbook Seven

Semester

2023/24 Spring/Autumn

Student Name: Subanta Poudel

London Met ID: 20048736

College ID: NP04CP4S210115

Assignment Due Date: December 21, 2024

Assignment Submission Date: December 21, 2024

Submitted To: Prasant Adhikari

Word Count (Where Required):

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded

Table of Contents

| | |
|---|---|
| Table of Figures | 3 |
| Introduction | 4 |
| Tasks and Steps | 5 |
| Task 1: Creating Directory Structures..... | 5 |
| Command: | 5 |
| Verification: | 5 |
| Task 2: Changing to the 1level3 Directory | 5 |
| Command: | 5 |
| To navigate back and move to another directory: | 5 |
| Task 3: Navigating Between Directories | 5 |
| From 1level3 to 2level3: | 5 |
| From 2level3 to 4level3: | 5 |
| Task 4: Creating a Text File in 1level3..... | 6 |
| Command: | 6 |
| Task 5: Copying Files | 6 |
| Commands:..... | 6 |
| Verification: | 6 |
| Task 6: Moving Files..... | 6 |
| Command: | 6 |
| Task 7: Printing Text Using Echo/Printf | 7 |
| Command: | 7 |
| Task 8: Using the ls Command..... | 7 |
| Commands:..... | 7 |
| Task 9: Removing Directories..... | 7 |
| Command: | 7 |
| Task 10: Managing File Permissions | 7 |
| Displaying Permissions: | 7 |
| Removing Permissions: | 7 |
| Adding Permissions: | 7 |
| Verification: | 8 |
| Task 11: Managing Directory Permissions | 8 |

| | |
|---------------------|---|
| Commands:..... | 8 |
| Verification: | 8 |
| Observations | 8 |
| Conclusion | 8 |

Table of Figures

No table of figures entries found.

Introduction

This log documents the tasks performed as part of Workshop 7, which focused on practicing various UNIX utilities. These tasks included creating directory structures, navigating directories using relative paths, managing files, modifying permissions, and working with UNIX commands like `mkdir`, `cd`, `chmod`, `ls`, and more. The primary aim was to enhance proficiency in directory and file operations in a UNIX environment. Few of the commands used:

- **mkdir**: Used to create directories. The `-p` option allows creating parent directories as needed.
- **sudo**: Executes commands with superuser privileges, essential for installing packages or modifying protected files.
- **apt**: A package management command used for installing, updating, or removing software on Debian-based systems.
- **tree**: Displays a visual representation of directory structures. Installed using `sudo apt-get install tree`.
- **cd**: Changes the current working directory. The `..` symbol refers to the parent directory.
- **cat**: Displays or creates files and appends content when used with `>>`.
- **ls**: Lists files and directories with options like:
 - o `-a`: Shows all files, including hidden ones.
 - o `-d`: Displays directories without listing their contents.
 - o `-g`: Omits owner information.
 - o `-l`: Provides detailed information.
 - o `-R`: Recursively lists files and directories.
- **cp**: Copies files or directories. Can rename files during copying.
- **mv**: Moves files or directories and can rename them.
- **chmod**: Modifies file or directory permissions. Common flags include `+r`, `+w`, and `+x` to add read, write, and execute permissions.
- **echo**: Outputs text to the terminal. The `-e` option enables interpretation of backslash escapes.
- **rm**: Removes files or directories. The `-i` option prompts before each removal.
- **rmdir**: Removes empty directories only.

This comprehensive set of commands provided the foundation for the tasks performed in this workshop, detailed below.

Tasks and Steps

Task 1: Creating Directory Structures

Create a directory structure with mkdir using relative paths.

Command:

```
sudo apt-get install tree
```

Installed the tree function to visualize the directory structure.

```
$ mkdir -p w7/{w7-1/{1level3,2level3},w7-2/{3level3,4level3}}
```

Here, the -p option ensures that parent directories are created if they do not exist.

Verification:

```
$ tree w7
```

The tree command displays the directory structure in a hierarchical format.

Task 2: Changing to the 1level3 Directory

Navigate to the 1level3 directory using the cd command with relative paths.

Command:

```
$ cd w7/w7-1/1level3/
```

To navigate back and move to another directory:

```
$ cd ../../w7-2/4level3/
```

Task 3: Navigating Between Directories

Practice changing directories using relative pathnames.

From 1level3 to 2level3:

```
$ cd ../2level3/
```

From 2level3 to 4level3:

```
$ cd ../../w7-2/4level3/
```

Back to W7:

```
$ cd ../../
```

Task 4: Creating a Text File in 1level3

We created a text file in 1level3 using the cat command.

Command:

```
$ cd ../../w7-1/1level3/
```

```
$ cat > file
```

```
This is a file.
```

```
Press Ctrl+D to save and exit.
```

Task 5: Copying Files

We copied the text file to other directories and renamed it during the copy operation.

Commands:

```
$ cp file file1
```

```
$ cp file1 ../2level3/
```

```
$ cp file1 ../../w7-2/3level3/renamed_file
```

Verification:

```
$ ls ../2level3/
```

```
$ ls ../../w7-2/3level3/
```

Task 6: Moving Files

We moved file1 to 4level3 and verified its removal from 1level3.

Command:

```
$ mv file1 ../../w7-2/4level3/
```

```
$ ls ../../w7-2/4level3/
```

```
$ ls
```

Task 7: Printing Text Using Echo/Printf

We printed multiple lines of text using echo with the -e option.

Command:

```
$ echo -e "Hello! I can do it\n5 > (20: 8) < (30 * 2)\nLine 1\nLine 2\na-b, A-B, -, +, <, >, #,  
$, %, &."
```

Task 8: Using the Is Command

We executed the ls command with various options to list contents.

Commands:

```
$ ls
```

```
$ ls -a
```

```
$ ls -d
```

```
$ ls -l
```

```
$ ls -R
```

Task 9: Removing Directories

We removed directories using rm and rmdir commands with the -i option for confirmation.

Command:

```
$ rm -ri w7-2/3level3/
```

```
$ rmdir w7-2/4level3/
```

Task 10: Managing File Permissions

We explored and modified file permissions using the chmod command.

Displaying Permissions:

```
$ ls -l 1level3/file1
```

Removing Permissions:

```
$ chmod -rw 1level3/file1
```

Adding Permissions:

```
$ chmod u+rw 1level3/file1
```

Verification:

```
$ ls -l 1level3/file1
```

Task 11: Managing Directory Permissions

We performed operations to modify and verify directory permissions.

Commands:

```
$ chmod -rwx 1level3/
```

```
$ ls -l
```

```
$ chmod u+rwx 1level3/
```

```
$ ls -l
```

Verification:

```
$ ls 1level3/
```

Observations

1. Basic Linux commands like mkdir, cd, rm, and cat were successfully used to manage files and directories.
2. The outputs of ls, ls -a, and ls -al were compared to understand hidden files and file permissions.
3. The tree command proved useful for visualizing directory structures.
4. File and directory permissions were modified using chmod, demonstrating its flexibility in user access control.

Conclusion

In this log, we explored:

1. Core Linux commands for file and directory management.
2. Creating and managing directory structures efficiently using commands like mkdir and tree.
3. Modifying and verifying permissions for files and directories.
4. Utilizing commands such as ls, cat, and chmod to understand and control file access.

This workshop provided practical experience with essential UNIX commands and concepts like directory structures, relative paths, file management, and permissions. By executing these tasks, we strengthened our understanding of how to effectively navigate and manipulate the UNIX file system.