



**Module Code & Module Title**

**Level 5 – CT5052, Network Operating Systems**

**Assessment Type**

**Logbook 7.**

**Semester**

**2023/24 Spring/Autumn**

**Student Name:** Asim Sapkota

**London Met ID:** 23048768

**College ID:** NP04CP4A230225

**Assignment Due Date:** 21<sup>th</sup> Dec 2024

**Assignment Submission Date:** 21<sup>th</sup> Dec 2024

**Submitted to:** Mr Prasant Adhikari

*I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order 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

---

Introduction .....	4
Objective .....	5
Required Tools and Concepts .....	5
Steps of Replicate .....	6
Conclusion.....	20
References .....	21

## Table of Figures

Figure 1: Trying to create the directory structure using mkdir without option .....	6
Figure 2: Creating the directory structure using mkdir with -p option.....	6
Figure 3: Changing to the 1level3 directory by on step using a relative pathname. ....	7
Figure 4: Practicing in changing directories in our directory.....	7
Figure 5: Changing to 1level3 and Creating a text file by using cat tool.....	7
Figure 6: Coping the text file in different directories changing its name .....	8
Figure 7: Moving the 1level3 file to 4level3.....	8
Figure 8: Printing some text in one echo command.....	9
Figure 9: Checking result using ls command without options .....	9
Figure 10: Checking result using ls command with -a options.....	10
Figure 11: Checking result using ls command with -d options .....	10
Figure 12: Checking result using ls command with -g options.....	11
Figure 13: Checking result using ls command with -i options .....	11
Figure 14: Checking result using ls command with -r options .....	11
Figure 15: Checking result using ls command without options and with a,d,g,l,r options in W7 directory .....	12
Figure 16: Checking result using ls command without options and with a,d,g,l,r options in W7-1 directory....	12
Figure 17: Checking result using ls command without options and with a,d,g,l,r options in 1level3 directory.	13
Figure 18: Changing to W7 directory and removing directory files W7-2,3level3,4level3 and all ordinary files in them and checking is it remove or not .....	13
Figure 19: Displaying access permissions for file1 in 1level3 .....	14
Figure 20: Removing all access permissions for this file .....	14
Figure 21: Displaying access permissions for this file .....	14
Figure 22: Trying to read this file using any utility(eg.cat) .....	15
Figure 23: Trying to write into this file using any utility (e.g., cat with the sign >> – append).....	15
Figure 24: Adding read and write access permissions for our self for this file. ....	15
Figure 25: Displaying access permissions for this file .....	15
Figure 26: Trying to read this file using any utility.....	16
Figure 27 : Trying to write into this file using cat utility. ....	16
Figure 28: Displaying access permissions for 1level3 .....	16
Figure 29: Removing all access permissions for the 1level3 directory .....	17
Figure 30: Displaying access permissions for 1level3 .....	17
Figure 31: Try to read a file from 1level3 using cat utility .....	17
Figure 32: Trying to put a file into 1level3 using cat utility. ....	17
Figure 33: Trying to search in 1level3 using ls command .....	18
Figure 34: Adding read, write, and execute access permissions for our self for the 1level3 directory.....	18
Figure 35: Displaying access permissions for 1level3 .....	18
Figure 36: Trying to read a file from 1level3 using cat utility. ....	18
Figure 37: Trying to put a file into 1level3 using cat utility.....	19
Figure 38: Trying to search in 1level3 using ls command .....	19

## Introduction

---

UNIX is a powerful, multi-tasking, and multi-user operating system. From the very beginning, UNIX was designed to be simple, efficient, and flexible. Its underlying philosophy is one of modularity—creating small, focused programs that each do one thing well. This approach allows users to combine these small programs to accomplish more complex tasks. The UNIX programming environment encourages programmers to write simple tools and then use them in combination to solve more complicated problems. These tools are often referred to as UNIX utilities and form the backbone of the operating system's power. (Pike, 1984)

UNIX utilities are important because they are efficient and flexible. These help us to manage files and directories, manage system permissions, or do administrative work with high accuracy. Command line utilities are different in the sense that unlike graphical user interfaces (GUIs), Flexible indeed hurts the concept of simplicity, and command line utilities are incredibly flexible and script friendly. Let's use one example — an occasion when you might need more than one step to do something, but one UNIX command can do it for you — it's just one of the things that make such tools a must for every System Administrator and Developer alike!

Here we're focusing on the simplest merging of UNIX commands like: `mkdir`, `cd`, `ls`, `rm`, and `chmod`. These are simply building blocks for further command in dealing with file systems and navigating them, and are important. In this, participants will practice creating directory structures, managing files and modifying access permissions to files. Learners who understand the base principles these utilities operate by can have a much greater awareness of the UNIX philosophy of doing one thing well.

The UNIX utility is still relevant in a world of user friendly interfaces, because most handling tasks on scale, platforms for automation, and security in the system. Accessibility is offered through alternates, which can be GUI tools but they're usually not customizable or as efficient. In honor, proficiency in UNIX commands not only make technical skills better but also make it easy to learn about the more complicated parts of operating systems and networks management.

## Objective

---

The objective of this workshop is to familiarize participants with essential UNIX utilities for effective file and directory management. Participants will practice creating and navigating directory structures using `mkdir` and `cd` commands and manage files through creation, copying, moving, and renaming operations. They will learn to modify file and directory permissions with `chmod` to ensure secure access control. The workshop also covers generating outputs using commands like `echo` and `printf` and exploring directories with the `ls` command. Additionally, participants will experiment with access permissions, practice safe file and directory removal using `rm` and `rmdir`, and address restricted access scenarios. By achieving these objectives, participants will gain foundational skills for efficient system administration and file system navigation.

## Required Tools and Concepts

---

### **Tools:**

- **UNIX-based System:** This refers to any operating system that supports UNIX-like environments (e.g., Linux, macOS).
- **Terminal/Shell:** A command-line interface for executing commands. Common types include `bash` and `zsh`.

### **Concepts:**

- **Directory Structure:** UNIX uses a hierarchical structure for organizing files and directories. Directories can contain subdirectories and files, which can be accessed using relative or absolute paths.
- **Relative Paths:** Paths that are relative to the current directory (e.g., `./` for the current directory, `../` for the parent directory).
- **Absolute Paths:** Full paths starting from the root directory (e.g., `/home/user/`).
- **File Permissions:** Files and directories have permissions defining what users can do with them (read, write, execute). Commands like `chmod` are used to manage these permissions.

- UNIX Commands: These essential commands allow users to manage files and directories:
  - a) mkdir: Creates new directories.
  - b) cd: Changes the current directory.
  - c) ls: Lists directory contents.
  - d) cp: Copies files from one location to another.
  - e) mv: Moves or renames files.
  - f) rm/rmdir: Removes files and directories.
  - g) echo/printf: Prints text or formats output.
- Redirection: This concept involves directing the output of commands to files using operators like > (overwrite) or >> (append).

## Steps of Replicate

### 1.

```
File Actions Edit View Help
(kali㉿kali)-[~]
$ mkdir W7/{W7-1/{1level3,2level3},W7-2/{3level3,4level3}}
mkdir: cannot create directory 'W7/W7-1/1level3': No such file or directory
mkdir: cannot create directory 'W7/W7-1/2level3': No such file or directory
mkdir: cannot create directory 'W7/W7-2/3level3': No such file or directory
mkdir: cannot create directory 'W7/W7-2/4level3': No such file or directory
```

Figure 1: Trying to create the directory structure using mkdir without option

```
(kali㉿kali)-[~]
$ mkdir -p W7/{W7-1/{1level3,2level3},W7-2/{3level3,4level3}}
(kali㉿kali)-[~]
$ tree W7
W7
├── W7-1
│   ├── 1level3
│   └── 2level3
└── W7-2
    ├── 3level3
    └── 4level3

7 directories, 0 files
(kali㉿kali)-[~]
$
```

Figure 2: Creating the directory structure using mkdir with -p option

2.

```
(kali㉿kali)-[~]  
$ cd W7/W7-1/1level3  
  
(kali㉿kali)-[~/W7/W7-1/1level3]  
$
```

Figure 3: Changing to the 1level3 directory by on step using a relative pathname.

3.

```
(kali㉿kali)-[~/W7/W7-1/1level3]  
$ cd ../2level3/  
  
(kali㉿kali)-[~/W7/W7-1/2level3]  
$ cd ../ ../W7-2/4level3/  
  
(kali㉿kali)-[~/W7/W7-2/4level3]  
$ cd ../ ../  
  
(kali㉿kali)-[~/W7]  
$
```

Figure 4: Practicing in changing directories in our directory

4.

```
(kali㉿kali)-[~/W7]  
$ cd W7-1/1level3  
  
(kali㉿kali)-[~/W7/W7-1/1level3]  
$ cat>file  
My name is Ashim Sapkota.  
  
(kali㉿kali)-[~/W7/W7-1/1level3]  
$ cat file  
My name is Ashim Sapkota.  
  
(kali㉿kali)-[~/W7/W7-1/1level3]  
$
```

Figure 5: Changing to 1level3 and Creating a text file by using cat tool

## 5.

```
(kali㉿kali)-[~/W7/W7-1/1level3]
$ cp file file1

(kali㉿kali)-[~/W7/W7-1/1level3]
$ cp file ../2level3/file2

(kali㉿kali)-[~/W7/W7-1/1level3]
$ cp file ../../W7-2/3level3/file3

(kali㉿kali)-[~/W7/W7-1/1level3]
$ cd

(kali㉿kali)-[~]
$ tree W7
W7
├── W7-1
│   ├── 1level3
│   │   ├── file
│   │   └── file1
│   └── 2level3
│       └── file2
└── W7-2
    ├── 3level3
    │   └── file3
    └── 4level3

7 directories, 4 files

(kali㉿kali)-[~]
$ █
```

Figure 6: Coping the text file in different directories changing its name

## 6.

```
(kali㉿kali)-[~/W7/W7-1/1level3]
$ mv file ../../W7-2/4level3/

(kali㉿kali)-[~/W7/W7-1/1level3]
$ cd

(kali㉿kali)-[~]
$ tree W7
W7
├── W7-1
│   ├── 1level3
│   │   └── file1
│   └── 2level3
│       └── file2
└── W7-2
    ├── 3level3
    │   └── file3
    └── 4level3
        └── file

7 directories, 4 files

(kali㉿kali)-[~]
$ █
```

Figure 7: Moving the 1level3 file to 4level3



## 7.

```
(kali㉿kali)-[~]  
$ echo "Hello! I can do it \n5>(20:8)<(30*2)\nLine 1 \nLine2 \na-b,A-B,-,+,<,>,#,$,%,&."  
Hello! I can do it  
5>(20:8)<(30*2)  
Line 1  
Line2  
a-b,A-B,-,+,<,>,#,$,%,&.  
  
(kali㉿kali)-[~]  
$
```

Figure 8: Printing some text in one echo command

## 8.

```
(kali㉿kali)-[~]  
$ ls  
alscript  Ashim  combined_Ashim  Documents  Music  Public  Videos  
ascript  Ashim1  Desktop  Downloads  Pictures  Templates  W7
```

Figure 9: Checking result using ls command without options

```
(kali㉿kali)-[~]
$ ls -a
.          Music
..         Pictures
alscript  .profile
ascript   Public
Ashim     .sudo_as_admin_successful
Ashim1    Templates
.bash_logout  .vboxclient-clipboard-tty7-control.pid
.bashrc       .vboxclient-clipboard-tty7-service.pid
.bashrc.original .vboxclient-display-svga-x11-tty7-control.pid
.cache        .vboxclient-display-svga-x11-tty7-service.pid
combined_Ashim .vboxclient-draganddrop-tty7-control.pid
.config       .vboxclient-draganddrop-tty7-service.pid
Desktop       .vboxclient-hostversion-tty7-control.pid
.dmrc         .vboxclient-seamless-tty7-control.pid
Documents     .vboxclient-seamless-tty7-service.pid
Downloads     .vboxclient-vmsvgasession-tty7-control.pid
.face         Videos
.face.icon   W7
.gnupg       .Xauthority
.ICEauthority .xsession-errors
.java        .xsession-errors.old
.local       .zsh_history
.mozilla     .zshrc
```

Figure 10: Checking result using ls command with -a options

```
(kali㉿kali)-[~]
$ ls -d
.
```

Figure 11: Checking result using ls command with -d options

```

(kali㉿kali)-[~]
$ ls -g
total 64
-rw-rw-r-- 1 kali 5598 Dec 13 12:46 a1script
-rw-rw-r-- 1 kali 8192 Dec 19 06:24 ascript
-rw-rw-r-- 1 kali 24 Dec 13 12:43 Ashim
-rw-rw-r-- 1 kali 62 Dec 20 02:52 Ashim1
-rw-rw-r-- 1 kali 88 Dec 13 12:45 combined_Ashim
drwxr-xr-x 2 kali 4096 Dec 13 12:33 Desktop
drwxr-xr-x 2 kali 4096 Dec 13 12:33 Documents
drwxr-xr-x 2 kali 4096 Dec 13 12:33 Downloads
drwxr-xr-x 2 kali 4096 Dec 20 12:33 Music
drwxr-xr-x 2 kali 4096 Dec 13 12:33 Pictures
drwxr-xr-x 2 kali 4096 Dec 13 12:33 Public
drwxr-xr-x 2 kali 4096 Dec 13 12:33 Templates
drwxr-xr-x 2 kali 4096 Dec 13 12:33 Videos
drwxrwxr-x 4 kali 4096 Dec 20 23:11 W7

```

Figure 12: Checking result using ls command with -g options

```

(kali㉿kali)-[~]
$ ls -i
1704038 a1script 1704074 combined_Ashim 1703990 Music 1703992 Videos
1704079 ascript 1703985 Desktop 1703991 Pictures 1704040 W7
1704077 Ashim 1703989 Documents 1703988 Public
1704213 Ashim1 1703986 Downloads 1703987 Templates

```

Figure 13: Checking result using ls command with -i options

```

(kali㉿kali)-[~]
$ ls -r
W7 Templates Pictures Downloads Desktop Ashim1 ascript
Videos Public Music Documents combined_Ashim Ashim a1script

```

Figure 14: Checking result using ls command with -r options

```
(kali㉿kali)-[~/W7]
$ ls
W7-1 W7-2

(kali㉿kali)-[~/W7]
$ ls -a
.  ..  W7-1  W7-2

(kali㉿kali)-[~/W7]
$ ls -d
.

(kali㉿kali)-[~/W7]
$ ls -g
total 8
drwxrwxr-x 4 kali 4096 Dec 20 23:11 W7-1
drwxrwxr-x 4 kali 4096 Dec 20 23:11 W7-2

(kali㉿kali)-[~/W7]
$ ls -i
1704405 W7-1 1704723 W7-2

(kali㉿kali)-[~/W7]
$ ls -r
W7-2 W7-1

(kali㉿kali)-[~/W7]
$
```

Figure 15: Checking result using ls command without options and with a,d,g,l,r options in W7 directory

```
(kali㉿kali)-[~/W7/W7-1]
$ ls
1level3 2level3

(kali㉿kali)-[~/W7/W7-1]
$ ls -a
.  ..  1level3 2level3

(kali㉿kali)-[~/W7/W7-1]
$ ls -d
.

(kali㉿kali)-[~/W7/W7-1]
$ ls -g
total 8
drwxrwxr-x 2 kali 4096 Dec 20 23:31 1level3
drwxrwxr-x 2 kali 4096 Dec 20 23:25 2level3

(kali㉿kali)-[~/W7/W7-1]
$ ls -i
1704717 1level3 1704721 2level3

(kali㉿kali)-[~/W7/W7-1]
$ ls -r
2level3 1level3

(kali㉿kali)-[~/W7/W7-1]
$
```

Figure 16: Checking result using ls command without options and with a,d,g,l,r options in W7-1 directory

```

(kali㉿kali)-[~/W7/W7-1/1level3]
$ ls
file1

(kali㉿kali)-[~/W7/W7-1/1level3]
$ ls -a
.  ..  file1

(kali㉿kali)-[~/W7/W7-1/1level3]
$ ls -d
.

(kali㉿kali)-[~/W7/W7-1/1level3]
$ ls -g
total 4
-rw-rw-r-- 1 kali 26 Dec 20 23:25 file1

(kali㉿kali)-[~/W7/W7-1/1level3]
$ ls -i
1704272 file1

(kali㉿kali)-[~/W7/W7-1/1level3]
$ ls -r
file1

(kali㉿kali)-[~/W7/W7-1/1level3]
$

```

Figure 17: Checking result using ls command without options and with a,d,g,l,r options in 1level3 directory

## 9.

```

(kali㉿kali)-[~/W7]
$ rm -i W7-2/3level3/file3
rm: remove regular file 'W7-2/3level3/file3'? y

(kali㉿kali)-[~/W7]
$ rm -i W7-2/4level3/file
rm: remove regular file 'W7-2/4level3/file'? y

(kali㉿kali)-[~/W7]
$ rm -r W7-2/{3level3,4level3}

(kali㉿kali)-[~/W7]
$ rm -r W7-2

(kali㉿kali)-[~/W7]
$ cd

(kali㉿kali)-[~]
$ tree W7
W7
├── W7-1
│   ├── 1level3
│   │   └── file1
│   └── 2level3
│       └── file2
└── W7-2

4 directories, 2 files

(kali㉿kali)-[~]
$

```

Figure 18: Changing to W7 directory and removing directory files W7-2,3level3,4level3 and all ordinary files in them and checking is it remove or not



## 10.

```
(kali㉿kali)-[~]  
$ cd W7/W7-1  
  
(kali㉿kali)-[~/W7/W7-1]  
$ ls -l 1level3/file1  
-rw-rw-r-- 1 kali kali 26 Dec 21 00:40 1level3/file1  
  
(kali㉿kali)-[~/W7/W7-1]  
$
```

Figure 19: Displaying access permissions for file1 in 1level3

```
(kali㉿kali)-[~/W7/W7-1]  
$ chmod -rwx 1level3/file1  
  
(kali㉿kali)-[~/W7/W7-1]  
$
```

Figure 20: Removing all access permissions for this file

```
(kali㉿kali)-[~/W7/W7-1]  
$ ls -l 1level3/file1  
----- 1 kali kali 26 Dec 21 00:40 1level3/file1  
  
(kali㉿kali)-[~/W7/W7-1]  
$
```

Figure 21: Displaying access permissions for this file

```
(kali㉿kali)-[~/W7/W7-1]
$ cat 1level3/file1
cat: 1level3/file1: Permission denied

(kali㉿kali)-[~/W7/W7-1]
$
```

Figure 22: Trying to read this file using any utility(eg.cat)

```
(kali㉿kali)-[~/W7/W7-1]
$ cat>>1level3/file1
zsh: permission denied: 1level3/file1

(kali㉿kali)-[~/W7/W7-1]
$
```

Figure 23: Trying to write into this file using any utility (e.g., cat with the sign >> – append).

```
(kali㉿kali)-[~/W7/W7-1]
$ chmod u+rw 1level3/file1

(kali㉿kali)-[~/W7/W7-1]
$
```

Figure 24: Adding read and write access permissions for our self for this file.

```
(kali㉿kali)-[~/W7/W7-1]
$ ls -l 1level3/file1
-rw----- 1 kali kali 26 Dec 21 00:40 1level3/file1

(kali㉿kali)-[~/W7/W7-1]
$
```

Figure 25: Displaying access permissions for this file

```
(kali㉿kali)-[~/W7/W7-1]
$ cat 1level3/file1
My name is Ashim Sapkota.

(kali㉿kali)-[~/W7/W7-1]
$
```

Figure 26: Trying to read this file using any utility

```
(kali㉿kali)-[~/W7/W7-1]
$ cat>>1level3/file1
I am from Syangja.

(kali㉿kali)-[~/W7/W7-1]
$
```

Figure 27 : Trying to write into this file using cat utility.

## 11.

```
(kali㉿kali)-[~/W7/W7-1]
$ ls -l 1level3
total 4
-rw----- 1 kali kali 45 Dec 21 01:03 file1

(kali㉿kali)-[~/W7/W7-1]
$
```

Figure 28: Displaying access permissions for 1level3



```
(kali㉿kali)-[~/W7/W7-1]
$ chmod -rwx 1level3

(kali㉿kali)-[~/W7/W7-1]
$
```

Figure 29: Removing all access permissions for the 1level3 directory

```
(kali㉿kali)-[~/W7/W7-1]
$ ls -l 1level3/file1
ls: cannot access '1level3/file1': Permission denied

(kali㉿kali)-[~/W7/W7-1]
$
```

Figure 30: Displaying access permissions for 1level3

```
(kali㉿kali)-[~/W7/W7-1]
$ cat 1level3/file1
cat: 1level3/file1: Permission denied

(kali㉿kali)-[~/W7/W7-1]
$
```

Figure 31: Try to read a file from 1level3 using cat utility

```
(kali㉿kali)-[~/W7/W7-1]
$ cat>>1level3/file1
zsh: permission denied: 1level3/file1

(kali㉿kali)-[~/W7/W7-1]
$
```

Figure 32: Trying to put a file into 1level3 using cat utility.

```
(kali㉿kali)-[~/W7/W7-1]
$ ls 1level3
ls: cannot open directory '1level3': Permission denied

(kali㉿kali)-[~/W7/W7-1]
$
```

Figure 33: Trying to search in 1level3 using ls command

```
(kali㉿kali)-[~/W7/W7-1]
$ chmod u+rwx 1level3/

(kali㉿kali)-[~/W7/W7-1]
$
```

Figure 34: Adding read, write, and execute access permissions for our self for the 1level3 directory.

```
(kali㉿kali)-[~/W7/W7-1]
$ ls -l 1level3/
total 4
-rw----- 1 kali kali 45 Dec 21 01:03 file1

(kali㉿kali)-[~/W7/W7-1]
$
```

Figure 35: Displaying access permissions for 1level3

```
(kali㉿kali)-[~/W7/W7-1]
$ cat 1level3/file1
My name is Ashim Sapkota.
I am from Syangja.

(kali㉿kali)-[~/W7/W7-1]
$
```

Figure 36: Trying to read a file from 1level3 using cat utility.

```
(kali㉿kali)-[~/W7/W7-1]
└─$ cat>>1level3/file1
I live in pokhara.

(kali㉿kali)-[~/W7/W7-1]
└─$
```

Figure 37: Trying to put a file into 1level3 using cat utility

```
(kali㉿kali)-[~/W7/W7-1]
└─$ ls 1level3/file1
1level3/file1

(kali㉿kali)-[~/W7/W7-1]
└─$
```

Figure 38: Trying to search in 1level3 using ls command

## Conclusion

---

In this workshop, we got some valuable hands on experience with the use of the essential UNIX utilities to manage files and directories with a UNIX based system. First we practiced with basic commands like `mkdir` to make directories, `cd` to change directories, and `ls` to see what's in the directories. Those basic commands let us traversal the directory structure, with relative and absolute paths, to learn how to access to files and directives in various places. We also played around with files copying (with `cp`), moving or renaming (with `mv`) and removing (with `rm`) files and directories. What we took away from this practical experience was a deeper understanding of the process of how files get managed with a UNIX environment. The other important thing that we had to work on was managing file permission using the `chmod` command. We learned to control access rights by changing file permissions, i.e. we decided, who can read, who can write or who can execute files. We also tried this out by trying to read from and write to files that we had different levels of permission to with the goal of understanding the importance of managing permissions in a multi user system.

Additionally, the workshop introduced the concept of output redirection using the `echo` and `printf` commands. We used these commands to print text and store it in files, practicing how to redirect output and append data to files. This technique is particularly useful for creating configuration files or logging system activities. These exercises not only taught us how the syntax of these commands work, meaning what they do, but also what these commands are used for in the real world. These are the skills you pick up in this workshop that are fundamental to anybody who uses, say UNIX like systems, because systems administrators, developers and people involved in how you manage files and how you manage your files on your system. In a nutshell, we were able to make proper use of tools and all appropriately introduced concepts so we are ready for more advanced utilities in UNIX system administration and other areas as well.

## References

---

Pike, B. W. (1984). Introduction to UNIX and its Philosophy. In R. P. Brian W. Kernighan, *The Unix Programming Environment* (p. 368). Prentice Hall.