



Module Code & Module Title
Level 5 – Network Operating System

Assessment Type
Logbook 7
Semester 1
2023/2024 Autumn

Student Name: Dipen Khatri
London Met ID: 23056968
College ID: NP04CP4A230133

Assignment Due Date: 21 December 2024
Assignment Submission Date: 21 December 2024
Submitted To: Mr. Prasant Adhikari

I confirm that I understand my coursework needs to be submitted online via GitHub 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

<i>Introduction</i>	<i>1</i>
<i>Question no: 1.....</i>	<i>2</i>
<i>Question no: 2.....</i>	<i>2</i>
<i>Question no: 3.....</i>	<i>4</i>
<i>Question no: 4.....</i>	<i>4</i>
<i>Question no: 5.....</i>	<i>5</i>
<i>Question no: 6.....</i>	<i>5</i>
<i>Question no: 7.....</i>	<i>7</i>
<i>Question no: 8.....</i>	<i>12</i>
<i>Question no: 9.....</i>	<i>14</i>
<i>Conclusion.....</i>	<i>22</i>

Table of Figures

Figure 1 Creating Directory.....	2
Figure 2 Change to 1Level3 directory.....	2
Figure 3 Changing Directories	4
Figure 4 Creating Text File	4
Figure 5 Copying Text File.....	5
Figure 6 : Moving the file	5
Figure 7 Printing Commands	7
Figure 8 Home Directory1	9
Figure 9 Home Directory2.....	9
Figure 10 W7 Directory	10
Figure 11 W7-1 Directory.....	11
Figure 12 1Level3 Directory.....	12
Figure 13 : W7 Directory	13
Figure 14 Display access permissions	14
Figure 15 Remove all access permissions	14
Figure 16 : Display access permissions	15
Figure 17 Reading the file.....	15
Figure 18 Writing the file	16
Figure 19 : Permission	16
Figure 20 : Read Permission	16
Figure 21 Write permission	16
Figure 22 Displaying Permission	16
Figure 23 Trying Read Permission	17
Figure 24 Trying Write Permission.....	18
Figure 25 Remove & Display Permission	19
Figure 26 Trying to read & Write.....	20
Figure 27 Trying to Search 1Level3.....	20
Figure 28 Adding Permissions and displaying	20
Figure 29 Trying to read, write & Search the 1Level3	20

Introduction

Log 7 focuses on practical exercises to enhance proficiency in using UNIX/Linux utilities. The session emphasizes core skills such as directory and file management, navigating file systems, modifying access permissions, and testing the effects of these permissions. We are tasked with creating a hierarchical directory structure using the `mkdir` command, both with and without the `-p` option, and navigating between directories using relative pathnames and special symbols like `.` and `...`. These exercises help build an understanding of directory organization and movement.

File management is another key focus, involving the creation, copying, renaming, and moving of files across directories. Verification of file presence and naming conventions in target directories is included to reinforce accuracy in command execution. Additionally, we explore the use of `echo` and `printf` commands to print structured text outputs, such as formatted lines and special characters. The `ls` command is practiced with multiple options (`-a`, `-d`, `-g`, `-l`, `-R`) to observe differences in listing directory contents.

File and directory deletion is practiced with the `rm` and `rmdir` commands, using the `-i` option for interactive confirmation. A critical part of the log is understanding access permissions, where we display, modify, and test the effects of read, write, and execute permissions on files and directories. By attempting operations like reading, writing, and searching with restricted permissions, we observe system behavior and learn how to manage access effectively. Overall, this log strengthens essential skills for managing Linux systems, preparing us for advanced tasks in system administration, automation, and scripting. It bridges theoretical knowledge with practical application, fostering confidence in navigating and controlling UNIX environments.

Question no: 1

Create the directory structure presented in the figure. Use **mkdir** command and relative pathnames from your home directory. Try both: no option and **-p** option, for the command.

```
(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
```

Figure 1 Creating Directory

The **mkdir** command is used to create directories in Linux. By using relative pathnames from the home directory, we can efficiently specify directory locations. The **-p** option allows the creation of nested directories in one step, while the basic command creates a single directory and requires existing parent directories. This practice highlights the difference between both approaches.

Question no: 2

Change to the **1level3** directory by one step using a relative pathname.

```
(kali㉿kali)-[~]
$ cd W7/W7-1/1Level3/

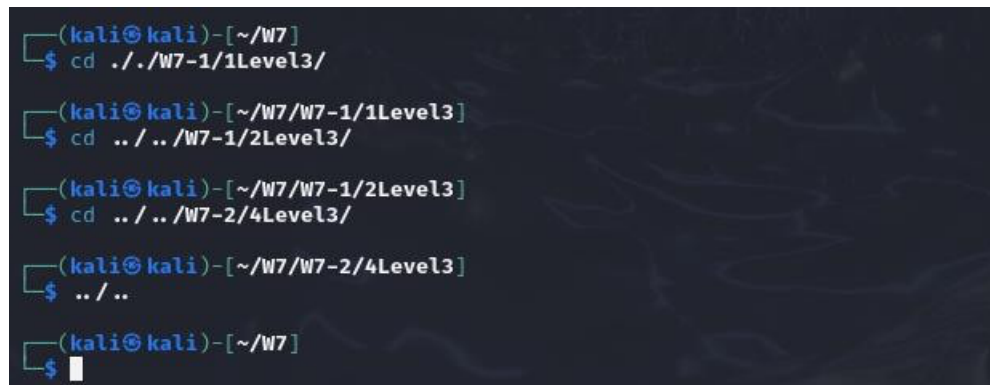
(kali㉿kali)-[~/W7/W7-1/1Level3]
$
```

Figure 2 Change to 1Level3 directory

This moves us directly into the **1level3** directory in one step, demonstrating efficient navigation within the directory structure.

Question no: 3

Practice in changing directories in your directory structure by one command using relative pathnames, e.g., from **1level3** to **2level3**, from **2level3** to **4level3**, from **4level3** to **W7**, etc. Use names of parent and child directories (‘.’ and ‘..’) as well.

A terminal window with a dark background and light blue text. It shows a series of commands and directory changes. The prompt is (kali@kali) and the current directory is shown in brackets after the prompt. The commands are: cd ../../W7-1/1Level3/, cd ../.. /W7-1/2Level3/, cd ../.. /W7-2/4Level3/, and cd ../.., returning to the initial directory ~/W7.

```
(kali@kali)~[~/W7]
$ cd ../../W7-1/1Level3/

(kali@kali)~[~/W7/W7-1/1Level3]
$ cd ../.. /W7-1/2Level3/

(kali@kali)~[~/W7/W7-1/2Level3]
$ cd ../.. /W7-2/4Level3/

(kali@kali)~[~/W7/W7-2/4Level3]
$ cd ../..

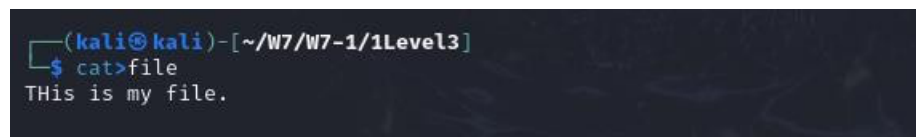
(kali@kali)~[~/W7]
$
```

Figure 3 Changing Directories

To navigate between directories using relative pathnames, we use the `cd` command. For example, to go from **1level3** to **2level3**, we use `cd ../2level3`. Similarly, `cd ../4level3` moves from **2level3** to **4level3**, and `cd ../W7` takes us from **4level3** to **W7**. Using `..` for the parent directory and `.` for the current directory simplifies navigation.

Question no: 4

Change to **1level3** and create a text file with any tool.

A terminal window with a dark background and light blue text. The prompt is (kali@kali) and the current directory is ~/W7/W7-1/1Level3. The command cat > file is entered, followed by the text 'This is my file.' on the next line.

```
(kali@kali)~[~/W7/W7-1/1Level3]
$ cat > file
This is my file.
```

Figure 4 Creating Text File

To change to the **1level3** directory, we use `cd 1level3`. To create a text file, we can use `echo "This is a sample text" > file.txt` or `cat > file.txt`, then type the text and press Ctrl + D to save. This creates file.txt in the 1level3 directory.

Question no: 5

Copy this text file from **1level3** to **1level3** (with the name **file1**), **2level3**, and to **3level3** changing its name. Show these files in corresponding directories.

```
(kali㉿kali)-[~/W7/W7-1/1Level3]
$ cp file file1

(kali㉿kali)-[~/W7/W7-1/1Level3]
$ ls
file file1

(kali㉿kali)-[~/W7/W7-1/1Level3]
$ cp file ../2Level3/

(kali㉿kali)-[~/W7/W7-1/1Level3]
$ ls ../2Level3/
file

(kali㉿kali)-[~/W7/W7-1/1Level3]
$ cp file ../../W7-2/3Level3/

(kali㉿kali)-[~/W7/W7-1/1Level3]
$ ls ../../W7-2/3Level3/
file
```

Figure 5 Copying Text File

We copied the text file from **1level3** to **1level3** with the new name **file1**, from **1level3** to **2level3** as file, and from **1level3** to **3level3** as file using the cp command.

Question no: 6

Move this file to **4level3**. Show that there is this file in **4level3** and there is not in **1level3**.

```
(kali㉿kali)-[~/W7/W7-1/1Level3]
$ mv file ../../W7-2/4Level3/

(kali㉿kali)-[~/W7/W7-1/1Level3]
$ ls ../../W7-2/4Level3/
file

(kali㉿kali)-[~/W7/W7-1/1Level3]
$ ls
file1

(kali㉿kali)-[~/W7/W7-1/1Level3]
$
```

Figure 6 : Moving the file

We moved the file from **1level3** to **4level3** using the mv command: **mv file.txt**

../4level3/. This transferred the file to **4level3** and removed it from **1level3**. To confirm, we listed the contents of both directories using **ls 1level3** and **ls ../4level3**, showing that the file is now in **4level3** and no longer present in **1level3**.

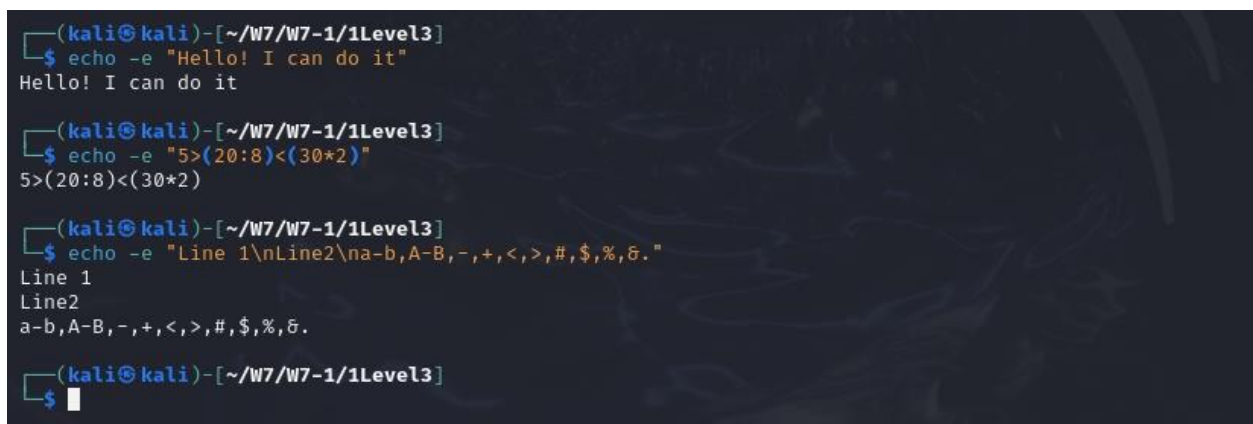
Question no: 7

Print the following texts each in one **echo** or **printf** command:

- Hello! I can do it
- $5 > (20 : 8) < (30 * 2)$
- Line 1

Line 2

a-b, A-B, -, +, <, >, #, \$, %, &.



```
(kali㉿kali)-[~/W7/W7-1/1Level3]
$ echo -e "Hello! I can do it"
Hello! I can do it

(kali㉿kali)-[~/W7/W7-1/1Level3]
$ echo -e "5>(20:8)<(30*2)"
5>(20:8)<(30*2)

(kali㉿kali)-[~/W7/W7-1/1Level3]
$ echo -e "Line 1\nLine2\na-b,A-B,-,+,<,>,#,$,%,&."
Line 1
Line2
a-b,A-B,-,+,<,>,#,$,%,&.

(kali㉿kali)-[~/W7/W7-1/1Level3]
$
```

Figure 7 Printing Commands

We can print the required texts using either the **echo** or **printf** command. For example, to print "Hello! I can do it," we can use **echo "Hello! I can do it"**. Similarly, to display the mathematical expressions or symbols like " $5 > (20 : 8) < (30 * 2)$ ", we can use **echo "5 > (20 : 8) < (30 * 2)"**. The other texts, such as "Line 1 Line 2" and "a-b, A-B, -, +, <, >, #, \$, %, &," can also be printed using **echo**. These commands are simple and effective for displaying structured text in the terminal.

Give the **ls** command (without options and with **a**, **d**, **g**, **l**, **R** options) in home directory, **w7**, **w7-1**, and **1level3** directories. Explain for yourself the results received.


```

(kali@kali)-[~]
$ ls -a
.          .gnupg          .vboxclient-draganddrop-tty7-control.pid
..         .lftauthority .vboxclient-draganddrop-tty7-service.pid
alscript   .java           .vboxclient-hostversion-tty7-control.pid
.bash_logout .local         .vboxclient-seamless-tty7-control.pid
.bashrc    .Music         .vboxclient-seamless-tty7-service.pid
.bashrc.original .Pictures     .vboxclient-vmsvga-session-tty7-control.pid
.cache     .profile      Videos
combined.txt .Public       W7
.config     Templates    .Xauthority
Desktop     test1        .xsession-errors
.dmrc       test2        .xsession-errors.oid
Documents   .vboxclient-clipboard-tty7-control.pid .zsh_history
Downloads   .vboxclient-clipboard-tty7-service.pid .zshrc
.face       .vboxclient-display-svgalib-tty7-control.pid
.face.icon  .vboxclient-display-svgalib-tty7-service.pid

(kali@kali)-[~]
$ ls -d
.
..

(kali@kali)-[~]
$ ls -lg
total 52
-rw-rw-r-- 1 kali 1192 Dec 13 10:01 alscript
-rw-rw-r-- 1 kali 86 Dec 13 09:58 combined.txt
drwxr-xr-x 2 kali 4096 Dec 13 05:03 Desktop
drwxr-xr-x 2 kali 4096 Dec 13 05:03 Documents
drwxr-xr-x 2 kali 4096 Dec 13 05:03 Downloads
drwxr-xr-x 2 kali 4096 Dec 13 05:03 Music
drwxr-xr-x 2 kali 4096 Dec 13 05:03 Pictures
drwxr-xr-x 2 kali 4096 Dec 13 05:03 Public
drwxr-xr-x 2 kali 4096 Dec 13 05:03 Templates
-rw-rw-r-- 1 kali 24 Dec 13 09:43 test1
-rw-rw-r-- 1 kali 62 Dec 13 09:50 test2
drwxr-xr-x 2 kali 4096 Dec 13 05:03 Videos
drwxr-xr-x 4 kali 4096 Dec 20 02:04 W7

(kali@kali)-[~]
$ ls -l
total 52
-rw-rw-r-- 1 kali 1192 Dec 13 10:01 alscript
-rw-rw-r-- 1 kali 86 Dec 13 09:58 combined.txt
drwxr-xr-x 2 kali 4096 Dec 13 05:03 Desktop
drwxr-xr-x 2 kali 4096 Dec 13 05:03 Documents
drwxr-xr-x 2 kali 4096 Dec 13 05:03 Downloads
drwxr-xr-x 2 kali 4096 Dec 13 05:03 Music
drwxr-xr-x 2 kali 4096 Dec 13 05:03 Pictures
drwxr-xr-x 2 kali 4096 Dec 13 05:03 Public
drwxr-xr-x 2 kali 4096 Dec 13 05:03 Templates
drwxr-xr-x 2 kali 4096 Dec 13 05:03 Templates

```

Figure 8 Home Directory1

```

drwxr-xr-x 2 kali 4096 Dec 13 05:03 templates
-rw-rw-r-- 1 kali 24 Dec 13 09:43 test1
-rw-rw-r-- 1 kali 62 Dec 13 09:50 test2
drwxr-xr-x 2 kali 4096 Dec 13 05:03 Videos
drwxr-xr-x 4 kali 4096 Dec 20 02:04 W7

(kali@kali)-[~]
$ ls -R
.:
alscript Desktop Downloads Pictures Templates test2 W7
combined.txt

./Desktop:

./Documents:

./Downloads:

./Music:

./Pictures:

./Public:

./Templates:

./Videos:

./W7:
W7-1 W7-2

./W7/W7-1:
3Level3 2Level3

./W7/W7-1/1Level3:
file1

./W7/W7-1/2Level3:
file

./W7/W7-2:
3Level3 4Level3

./W7/W7-2/3Level3:
file

./W7/W7-2/4Level3:
file

```

Figure 9 Home Directory2

In the home directory, **ls** lists the files, while **ls -a** shows hidden file. The **-d** option shows only directories, **-g** omits owner details, and **-l** provides detailed information like permissions. Using **-R** lists files and subdirectories recursively, showing the full directory structure.

In W7:

```
(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 02:04 W7-1
drwxrwxr-x 4 kali 4096 Dec 20 02:04 W7-2

(kali㉿kali)-[~/W7]
└─$ ls -l
total 8
drwxrwxr-x 4 kali kali 4096 Dec 20 02:04 W7-1
drwxrwxr-x 4 kali kali 4096 Dec 20 02:04 W7-2

(kali㉿kali)-[~/W7]
└─$ ls -R
.:
W7-1  W7-2

./W7-1:
1Level3  2Level3

./W7-1/1Level3:
file1

./W7-1/2Level3:
file

./W7-2:
3Level3  4Level3

./W7-2/3Level3:
file

./W7-2/4Level3:
file
```

Figure 10 W7 Directory

In the w7 directory, **ls** lists the files and subdirectories, while **ls -a** includes hidden files. The **-d** option shows only the directories, **-g** removes owner details, and **-l** provides detailed information like file permissions. Using **-R** lists all files and subdirectories recursively, revealing the entire structure of w7.

In w7-1:

```
(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 03:54 1Level3
drwxrwxr-x 2 kali 4096 Dec 20 02:42 2Level3

(kali㉿kali)-[~/W7/W7-1]
└─$ ls -l
total 8
drwxrwxr-x 2 kali kali 4096 Dec 20 03:54 1Level3
drwxrwxr-x 2 kali kali 4096 Dec 20 02:42 2Level3

(kali㉿kali)-[~/W7/W7-1]
└─$ ls -R
.:
1Level3  2Level3

./1Level3:
file1

./2Level3:
file
```

Figure 11 W7-1 Directory

In the w7-1 directory, **ls** lists the files and subdirectories, while **ls -a** shows hidden file. The **-d** option displays only directories, **-g** omits the owner information, and **-l** provides a detailed list with permissions and other file details. Using **-R** lists all files and subdirectories recursively, showing the full structure of w7-1.

In 1Level3:

```
(kali㉿kali)-[~/W7/W7-1/1Level3]
$ man ls

(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 29 Dec 20 02:41 file1

(kali㉿kali)-[~/W7/W7-1/1Level3]
$ ls -l
total 4
-rw-rw-r-- 1 kali kali 29 Dec 20 02:41 file1

(kali㉿kali)-[~/W7/W7-1/1Level3]
$ ls -R
.:
file1
```

Figure 12 1Level3 Directory

In the 1level3 directory, **ls** lists the files and subdirectories, while **ls -a** includes hidden files. The **-d** option shows only directories, **-g** omits the owner details, and **-l** provides a detailed view with file permissions and other information. Using **-R** lists all files and subdirectories recursively, revealing the complete structure of 1level3.

Question no: 8

Change to the **W7** directory. Remove the directory files **w7-2**, **3level-3**, **4level3** and all ordinary files in them. Use the option **-i** of the **rm** and **rmdir** commands. Show that there are not these ordinary and directory files in your file structure.

```
(kali㉿kali)-[~/W7]
$ cd

(kali㉿kali)-[~]
$ cd W7

(kali㉿kali)-[~/W7]
$ rm -r W7~2

(kali㉿kali)-[~/W7]
$ ls -l
total 4
drwxrwxr-x 4 kali kali 4096 Dec 20 02:04 W7~1
```

Figure 13 : W7 Directory

To remove the directories **w7-2**, **3level-3**, and **4level3** and their files in W7, we use **rm -i** to delete the files and **rmdir -i** for the directories. After confirming the deletions, we check with **ls** to ensure the directories and files are no longer present.

Question no: 9

Change to **w7-1**.

- Display access permissions for the file **file1** in **1level3**.
- Remove all access permissions for this file.
- Display access permissions for this file.
- Try to read this file using any utility (e.g., **cat**).
- Try to write into this file using any utility (e.g., **cat** with the sign **>>** – *append*).
- Add read and write access permissions for yourself for this file.
- Display access permissions for this file.
- Try to read this file using any utility.
- Try to write into this file using any utility.

```
(kali@kali)-[~/W7]
$ cd ../W7-1/

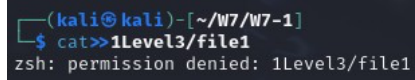
(kali@kali)-[~/W7/W7-1]
$ ls -l 1level3/
total 4
-rw-rw-r-- 1 kali kali 29 Dec 20 02:41 file1
```

Figure 14 Display access permissions

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

Figure 15 Remove all access permissions

```
(kali@kali)-[~/W7/W7-1]
$ ls -l 1level3/
total 4
----- 1 kali kali 29 Dec 20 02:41 file1
```

Figure 16 : Display access permissionsA terminal window with a dark background. The prompt is `(kali㉿kali)-[~/W7/W7-1]`. The user enters `$ cat>>1Level3/file1`. The output is `zsh: permission denied: 1Level3/file1`.

```
(kali㉿kali)-[~/W7/W7-1]
$ cat>>1Level3/file1
zsh: permission denied: 1Level3/file1
```

Figure 17 Reading the file

```
(kali㉿kali)-[~/W7/W7-1]
$ chmod u+rw 1Level3/file1
```

Figure 18 Writing the file

```
(kali㉿kali)-[~/W7/W7-1]
$ ls -l 1Level3/
total 4
-rw----- 1 kali kali 29 Dec 20 02:41 file1
```

Figure 19 : Permission

```
(kali㉿kali)-[~/W7/W7-1]
$ cat 1Level3/file1
THis is my file.
```

```
]
```

Figure 20 : Read Permission

```
(kali㉿kali)-[~/W7/W7-1]
$ cat>>1Level3/file1
>..^C
```

Figure 21 Write permission

```
(kali㉿kali)-[~/W7/W7-1]
$ ls -l
total 8
drwxrwxr-x 2 kali kali 4096 Dec 20 03:54 1Level3
drwxrwxr-x 2 kali kali 4096 Dec 20 02:42 2Level3
```

Figure 22 Displaying Permission

```
(kali㉿kali)-[~/W7/W7-1]
$ chmod -rwx 1Level3/

(kali㉿kali)-[~/W7/W7-1]
$ ls -l
total 8
d----- 2 kali kali 4096 Dec 20 03:54 1Level3
drwxrwxr-x 2 kali kali 4096 Dec 20 02:42 2Level3
```

Figure 23 Trying Read Permission

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

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

Figure 24 Trying Write Permission

In the w7-1 directory, we first display the access permissions for file1 in **1level3**. Then, we remove all access permissions using **chmod**. After checking the permissions again, we attempt to read and write to the file using utilities like **cat**. Next, we add read and write permissions for ourselves and confirm the changes. Finally, we test reading and writing to the file again to verify the updated permissions.

(Now,)

- Display access permissions for **1level3**.
- Remove all access permissions for the **1level3** directory.
- Display access permissions for **1level3**.
- Try to read a file from **1level3** using any utility.
- Try to put a file into **1level3** using any utility.
- Try to search in **1level3** using any command (e.g., the **ls** command).
- Add read, write, and execute access permissions for yourself for the **1level3** directory.
- Display access permissions for **1level3**.
- Try to read a file from **1level3** using any utility.
- Try to put a file into **1level3** using any utility.
- Try to search in **1level3** using any command (e.g., the **ls** command).

```
(kali㉿kali)-[~/W7/W7-1]
$ chmod -rwx 1Level3/

(kali㉿kali)-[~/W7/W7-1]
$ ls -l
total 8
d----- 2 kali kali 4096 Dec 20 03:54 1Level3
drwxrwxr-x 2 kali kali 4096 Dec 20 02:42 2Level3
```

Figure 25 Remove & Display Permission

```

(kali㉿kali)-[~/W7/W7-1]
$ cat 1Level3/file1
cat: 1Level3/file1: Permission denied

(kali㉿kali)-[~/W7/W7-1]
$ cat>>1Level3/file1
zsh: permission denied: 1Level3/file1

```

Figure 26 Trying to read & Write

```

(kali㉿kali)-[~/W7/W7-1]
$ ls 1Level3/
ls: cannot open directory '1Level3/': Permission denied

```

Figure 27 Trying to Search 1Level3

```

(kali㉿kali)-[~/W7/W7-1]
$ chmod u+rwX 1Level3/

(kali㉿kali)-[~/W7/W7-1]
$ ls -l
total 8
drwx----- 2 kali kali 4096 Dec 20 03:54 1Level3
drwxrwxr-x 2 kali kali 4096 Dec 20 02:42 2Level3

```

Figure 28 Adding Permissions and displaying

```

(kali㉿kali)-[~/W7/W7-1]
$ cat 1Level3/file1
This is my file.

]

(kali㉿kali)-[~/W7/W7-1]
$ cat>>1Level3/file1
^C

(kali㉿kali)-[~/W7/W7-1]
$ ls 1Level3/
file1

```

Figure 29 Trying to read, write & Search the 1Level3

In this task, we first display the access permissions for the **1level3** directory. We then remove all access permissions using **chmod**. After checking the permissions again, we try to read a file, add a file, and search within 1level3 using utilities like **cat**, **touch**, and

ls. Since we removed the permissions, these actions will fail. Next, we add read, write, and execute permissions for ourselves using `chmod`, then confirm the changes. Finally, we try the same actions again, and with the new permissions, we should be able to read, write, and search within `level3`.

Conclusion

In conclusion, Log 7 provided a comprehensive hands-on approach to mastering essential UNIX/Linux commands and utilities, which are crucial for effective system administration and file management. The practical exercises facilitated a deeper understanding of directory structure creation, file manipulation, and permission management within the Linux environment. By engaging with tasks such as using the `mkdir` command with and without the `-p` option, we explored the flexibility of directory creation, allowing for the efficient organization of files across various levels of the directory tree. The exercises in navigating between directories using relative pathnames helped reinforce the importance of efficient navigation within a file system. Through tasks like changing directories with commands like `cd` and using special symbols like `.` and `..`, we gained a clearer understanding of how to move between directories quickly and effectively. Additionally, practicing file creation, copying, renaming, and moving provided us with hands-on experience in managing files across different directories, ensuring that we can maintain an organized file structure.

Understanding and modifying file and directory permissions was a critical part of this log. We learned how to display and change access permissions using the `chmod` command and tested the effect of these changes on the ability to read, write, and execute files. This practical exercise gave insight into managing file security, ensuring that we are equipped to handle file access control in real-world scenarios.

Overall, Log 7 bridged theoretical knowledge with practical skills, equipping us with a solid foundation in UNIX/Linux systems. By working through these exercises, we not only honed our technical skills but also developed a deeper understanding of file and directory management, access control, and system administration in Linux environments, preparing us for more advanced tasks in automation, scripting, and system management.