

Practical no. – 02

AIM:- To study various file access commands.

Commands:

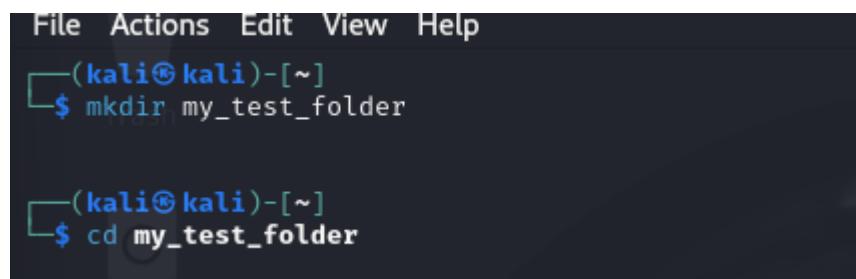
- **mkdir (Make Directory)** - The mkdir command is used to create one or more directories in the filesystem. It organizes files into separate folders.

Syntax: mkdir [options] directory_name

- ❖ -p: Creates parent directories if they do not exist.
- ❖ -v: Prints a message for each directory created.

- **cd (Change Directory)** - The cd command is used to navigate between directories. It changes the current working directory to the specified path.

Syntax: cd [directory_path]



```
File Actions Edit View Help
└─(kali㉿kali)-[~]
    $ mkdir my_test_folder

└─(kali㉿kali)-[~]
    $ cd my_test_folder
```

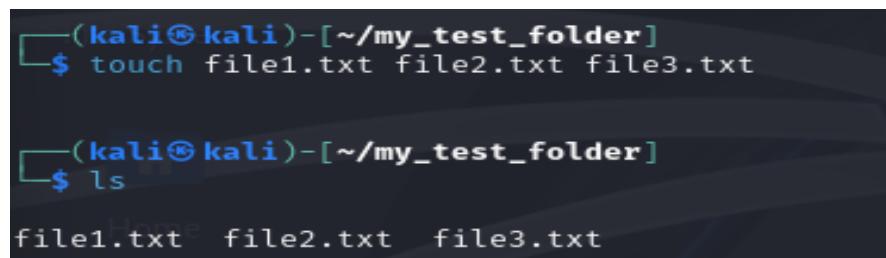
- **touch** - The touch command is used to create empty files or update the timestamps of existing files without modifying their content.

Syntax: touch [options] file_name

- **ls (List Directory Contents)** - The ls command lists the contents of a directory, including files and subdirectories. It is used to view file details and structure.

Syntax: ls [options] [directory]

- ❖ -l: Long format (shows details like permissions, owner, size).
- ❖ -a: Shows all files, including hidden files (those starting with .).
- ❖ -h: Human-readable sizes (e.g., KB, MB)



```
└─(kali㉿kali)-[~/my_test_folder]
    $ touch file1.txt file2.txt file3.txt

└─(kali㉿kali)-[~/my_test_folder]
    $ ls
file1.txt  file2.txt  file3.txt
```

- **cat (Concatenate)** - The cat command reads the content of files and outputs them to the terminal. It can also be used to create or combine files.

Syntax: cat [options] [file_name]

```
(kali㉿kali)-[~/my_test_folder]
$ cat > file1.txt
This is the content of file1.

^Z
zsh: suspended  cat > file1.txt
```

- **cp (Copy)** - The cp command copies files or directories from one location to another, creating a duplicate of the source.

Syntax: cp [options] source destination

- ❖ -r: Copies directories recursively.
- ❖ -v: Shows the files being copied.
- ❖ -i: Prompts before overwriting files.

```
(kali㉿kali)-[~/my_test_folder]
$ cat file1.txt
This is the content of file1.
```

- **mv (Move)** - The mv command is used to move files or directories to a new location or rename them within the same directory.

Syntax: mv [options] source destination

```
(kali㉿kali)-[~/my_test_folder]
$ cp file1.txt backup_file1.txt

(kali㉿kali)-[~/my_test_folder]
$ mv backup_file1.txt renamed_file1.txt
```

- **echo** - The echo command outputs text to the terminal or writes it to a file. It is often used in scripting for displaying messages or creating files with content.

Syntax: echo [options] "text" >> file_name

- **head** - The head command displays the first few lines of a file (default is 10 lines). It is useful for previewing file content.

Syntax: head [options] file_name

- **tail** - The tail command displays the last few lines of a file (default is 10 lines). It is often used to monitor logs in real-time.

Syntax: tail [options] file_name

- ❖ -n [number]: Displays the last [number] lines.

```
(kali㉿kali)-[~/my_test_folder]
$ echo "Additional content for file1." >> file1.txt

(kali㉿kali)-[~/my_test_folder]
$ head file1.txt

This is the content of file1.

Additional content for file1.
```

- **tac** - The tac command reverses the order of lines in a file, displaying the last line first and the first line last.

Syntax: tac file_name

```
(kali㉿kali)-[~/my_test_folder]
$ tac file1.txt

Additional content for file1.

This is the content of file1.
```

- **chmod (Change Mode)** - The chmod command changes the access permissions of files or directories. Permissions determine who can read, write, or execute the file.

Syntax: chmod [mode] file_name

```
(kali㉿kali)-[~/my_test_folder]
$ chmod 644 file1.txt
```

- **rm (Remove)** - The rm command deletes files or directories. With recursive options, it can delete directories and their contents

Syntax: rm [options] file_name

- ❖ -r: Deletes directories and their contents recursively.
- ❖ -i: Prompts for confirmation before deleting.

- **rmdir (Remove Directory)** - The rmdir command removes empty directories. It will fail if the directory contains any files or subdirectories.

Syntax: rmdir [options] directory_name

- ❖ -p: Removes the specified directory and its parent directories if they are empty.

```
[~] (kali㉿kali)-[~/my_test_folder]
└─$ rm renamed_file1.txt

[~] (kali㉿kali)-[~/my_test_folder]
└─$ cd ..
└─$ rm -r my_test_folder
```

- **history** - The history command shows a list of previously executed commands in the terminal, allowing users to recall or reuse them.

Syntax: history

```
[~] (kali㉿kali)-[~]
└─$ history

 1 date
 2 cal
 3 ncal
 4 sudo apt ncal
 5 cal
 6 sudo apt ncal
 7 cls
 8 clr
 9 clear
10 mkdir 'hello'
11 open hello
12 mkdir newpythonfile
13 head newpythonfile
14 python3
```