Topic Name:

The main aim of this lab session is to provide hands-on experience on

- Explore file structure
- File management commands
- Absolute path and Relative path
- Globbing
- Scripting

File Structure

1. Under the root directory there are many files like
   /bin , /boot , /dev , /etc , ...
   Find out the importance of those files
   Example : /etc is for user account details

S.No	Directory	Usage
1	/	Root directory.
2	/bin	Binary files .
3	/boot	Contains all the files required for the Linux Boot Process.
4	/dev	Special Device Files for all Devices.
5	/etc	Houses Configuration files for System services and Daemons.
6	/home	Personal workspace for each user.
7	/lib	Shared library images and kernel modules that are essential for booting the system and running commands in the root filesystem.
8	/proc	A virtual file system that provides information about the system's kernel and processes.
9	/sbin	System administration commands and binaries that are primarily used by the system administrator
10	/tmp	A temporary storage location for files and directories that are created and accessed during system runtime
11	/var	A standard directory that stores variable files, or files that change frequently while the system is running

2. In Linux, there are three different files
   Regular file:
   Regular files in Linux are the most common type of file and contain data that can be read and modified.

   Directory File:
   A directory file in Linux is a type of file that stores other files and directories, and contains the information the system needs to access them

Special file:

Special files, also known as device files, are associated with a computer system's hardware devices or other resources.

Block file:

Used to access and manage storage devices at the block level.

Character file:

Facilitate direct communication between user programs and hardware devices.

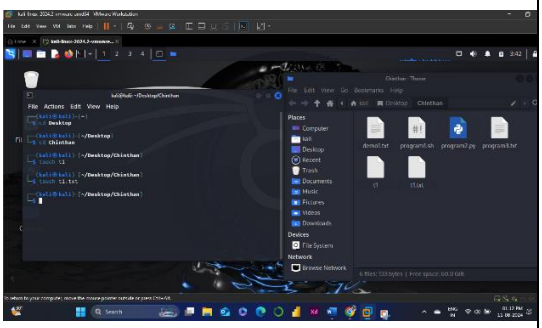
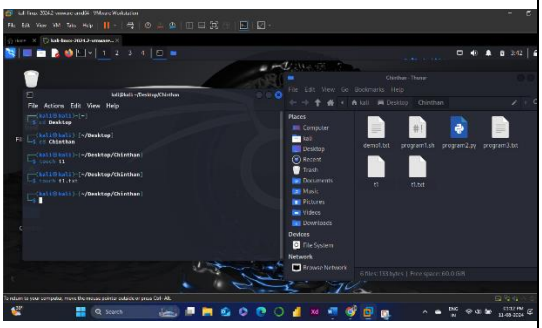
Socket file:

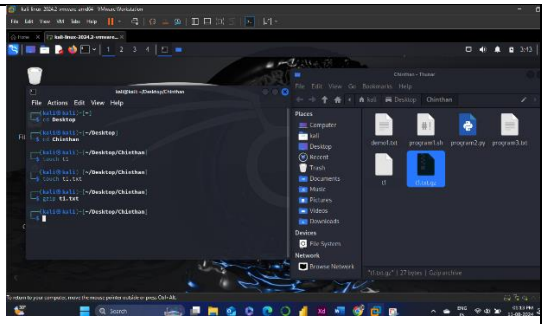
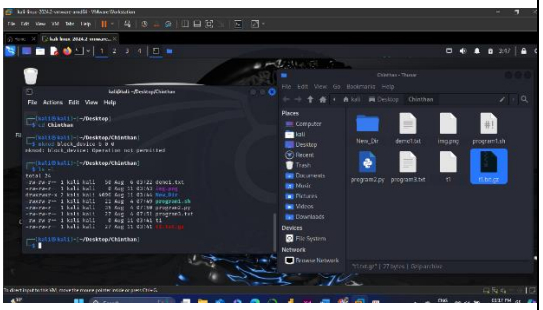
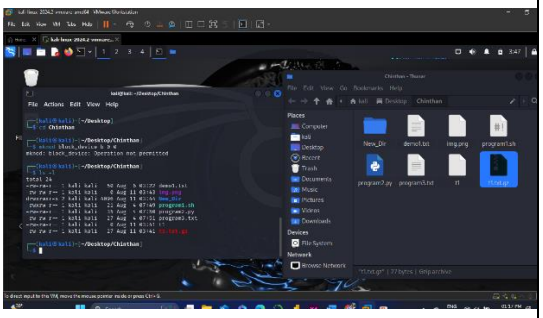
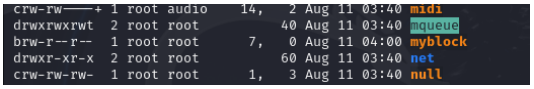
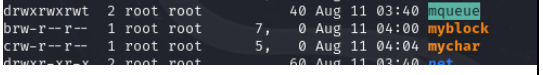
A file descriptor that acts as a communication endpoint for processes running on the device.

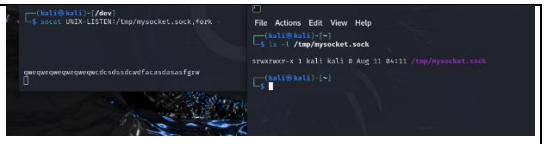
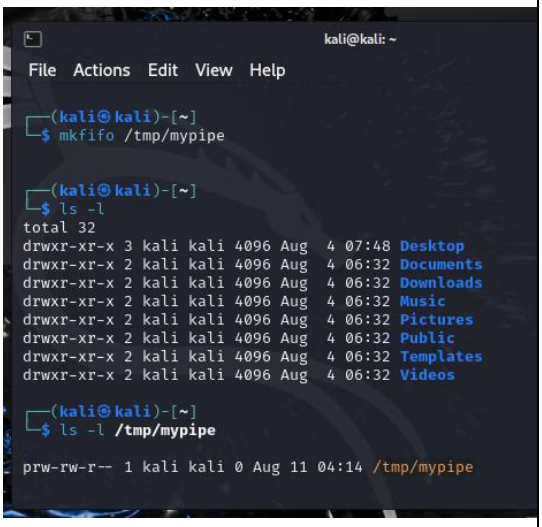
Pipe file:

Unnamed objects created to allow two processes to communicate.

Fill the below table:

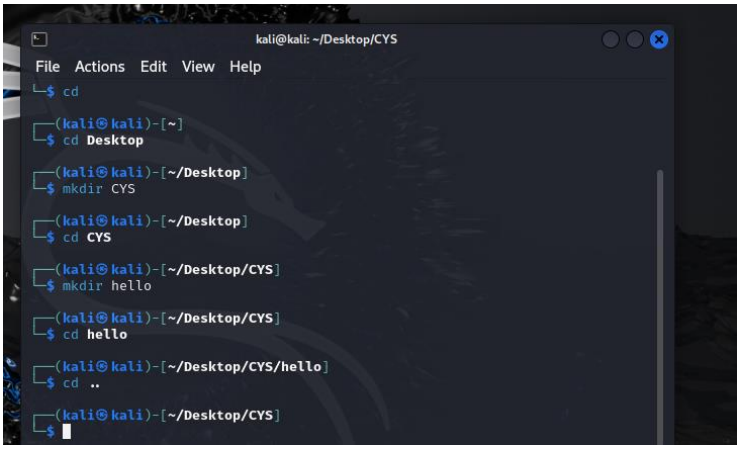
File Type	Represented by (Hintls)	Role	How to create	How to check	Location	Screen shot
Regular file	-	Stores data such as text, images, etc.	touch t1	NA	/home/kali/Desktop/Chinthan	
- Text file	-	Contains plain text	touch t1.txt	NA	/home/kali/Desktop/Chinthan	
- Compres	-	Stores data in a compres	gzip filename	NA	/home/kali/Desktop/Chinthan	

sed file		sed format				
- Image	-	Stores image data	NA	NA	/home/kali/Desktop/Chinthan	
Directory	d	Contains files and other directories	mkdir filename	NA	/home/kali/Desktop/Chinthan	
Block file	b	Represents a block device (e.g., disk)	NA	NA	/dev	 
Character file	c	Represents a character device (e.g., terminal)	NA	NA	/dev	 
Socket file	s	Provides inter-process	NA	NA	/tmp	

		communication (IPC)				
pipe file	p	Used for IPC (inter-process communication)	NA	NA	/tmp	

3. Globbing

a. Go back to CYS



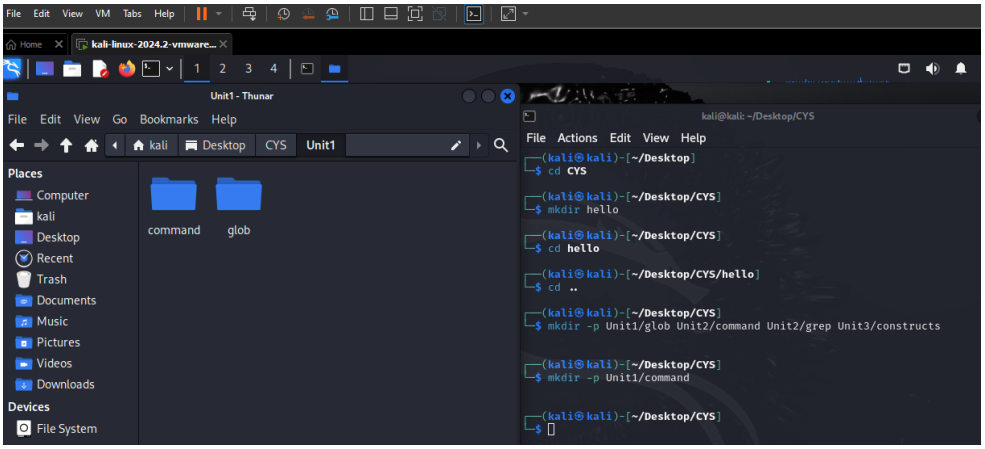
b. Create multiple subdirectories using single command

LS

Unit1

command

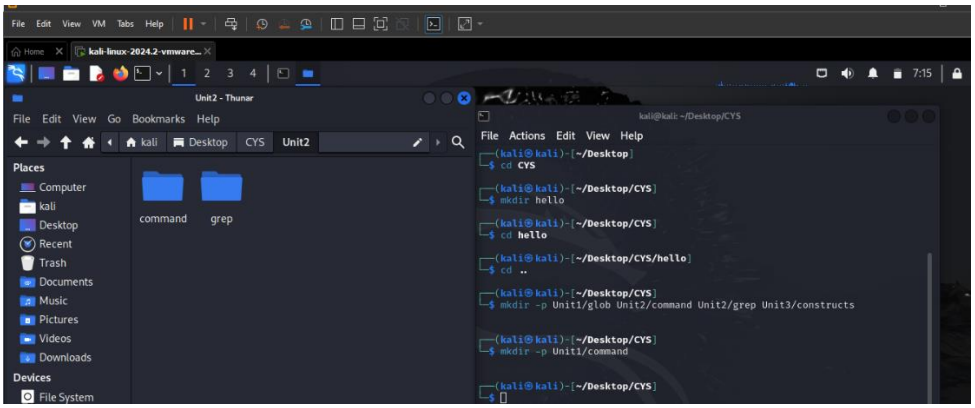
glob



Unit2

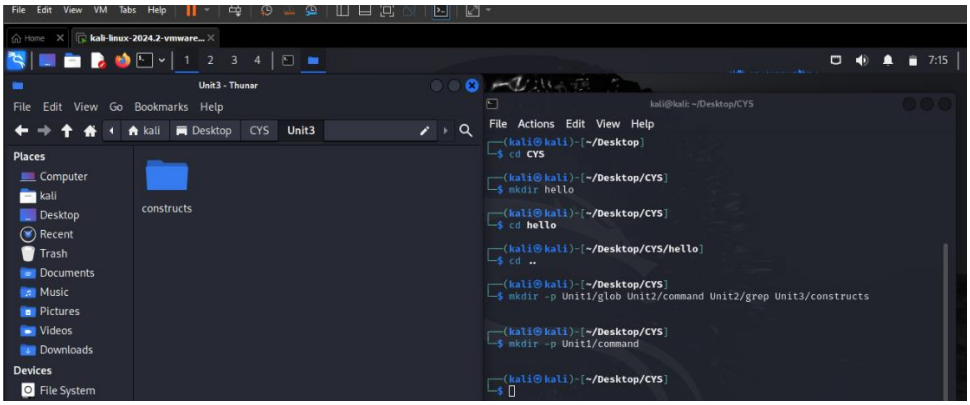
command

grep



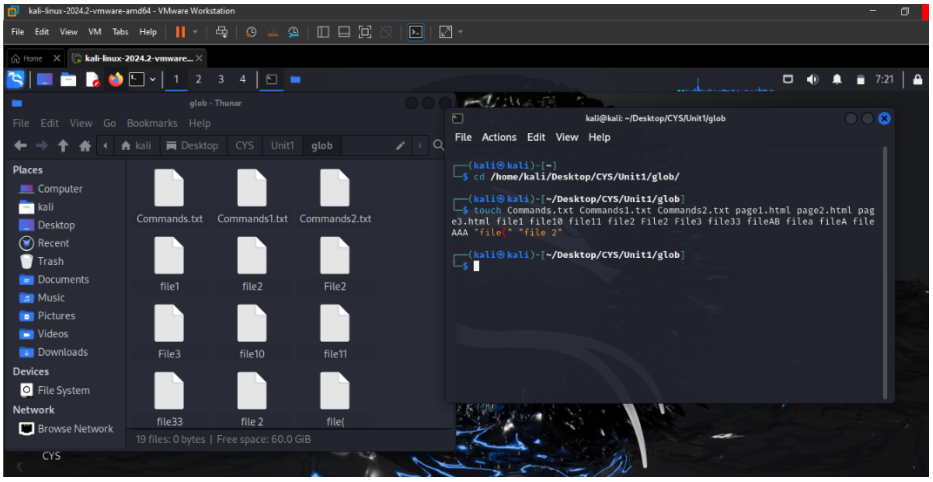
Unit3

Constructs



c. Navigate to unit1/glob

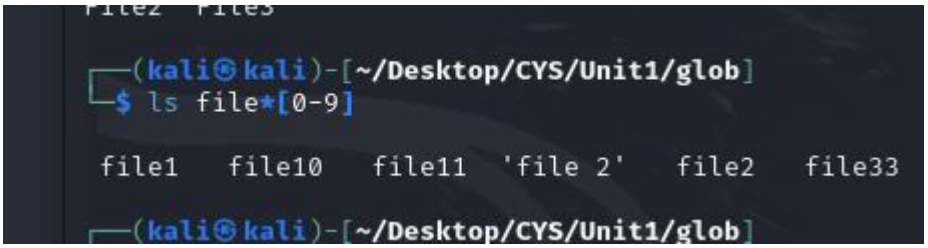
d. Create the following files:
Commands.txt
Commands1.txt
Commands2.txt
page1.html
page2.html
page3.html
`file1`
`file10`
`file11`
`file2`
`File2`
`File3`
`file33`
`fileAB`
`filea`
`fileA`
`fileAAA`
`file(`
`file2`



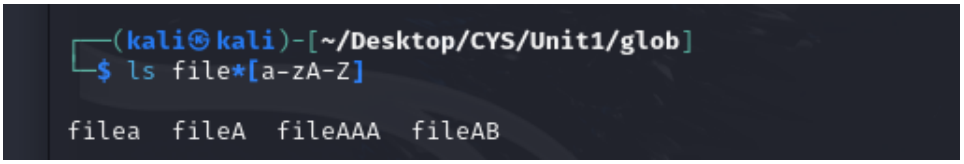
i. List all files starting with file

ii. List all files starting with File

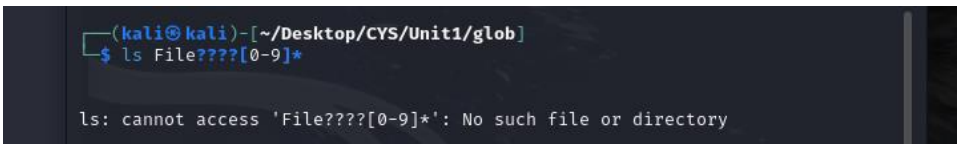
iii. List all files starting with file and ending in a number.



iv. List all files starting with file and ending with a letter



v. List all files starting with File and having a digit as fifth character.



vi. List all files starting with File and having a digit as fifth character and nothing else.

```
┌──(kali㉿kali)-[~/Desktop/CYS/Unit1/glob]
└─$ ls File????[0-9]

ls: cannot access 'File????[0-9]': No such file or directory

┌──(kali㉿kali)-[~/Desktop/CYS/Unit1/glob]
└─$
```

vii. List (with ls) all files starting with a letter and ending in a number.

```
┌──(kali㉿kali)-[~/Desktop/CYS/Unit1/glob]
└─$ ls [a-zA-Z]*[0-9]

file1   file10   file11  'file 2'   file2   File2   File3   file33
```

viii. List (with ls) all files that have exactly five characters.

```
┌──(kali㉿kali)-[~/Desktop/CYS/Unit1/glob]
└─$ ls ?????
'file('   file1   file2   File2   File3   filea   fileA
```

ix. List (with ls) all files that start with f or F and end with 3 or A.

```
┌──(kali㉿kali)-[~/Desktop/CYS/Unit1/glob]
└─$ ls [fF]*[A3]

File3  file33  fileA  fileAAA
```

x. List (with ls) all files that start with f have i or R as second character and end in a number.

```
┌──(kali㉿kali)-[~/Desktop/CYS/Unit1/glob]
└─$ ls f[iR]*[0-9]

file1   file10   file11  'file 2'   file2   file33
```

xi. List all files that do not start with the letter F.

```
┌──(kali㉿kali)-[~/Desktop/CYS/Unit1/glob]
└─$ ls [^F]*
Commands1.json  'file('   file11    file33   fileAAA
Commands2.json   file1   'file 2'   filea    fileAB
Commands.json    file10   file2     fileA
```

xii. Remove all the *.html

```
┌──(kali㉿kali)-[~/Desktop/CYS/Unit1/glob]
└─$ rm *.html

┌──(kali㉿kali)-[~/Desktop/CYS/Unit1/glob]
└─$ ls
Commands1.txt  'file('   file11    File2   filea    fileAB
Commands2.txt   file1   'file 2'   File3   fileA
Commands.txt    file10   file2     file33  fileAAA
```

xiii. Rename *.txt to *.json

```
┌──(kali㉿kali)-[~/Desktop/CYS/Unit1/glob]
└─$ for file in *.txt; do
    mv "$file" "${file%.txt}.json"
done

┌──(kali㉿kali)-[~/Desktop/CYS/Unit1/glob]
└─$ ls
Commands1.json  'file('   file11    File2    filea     fileAB
Commands2.json   file1   'file 2'   File3    fileA
Commands.json    file10   file2     file33   fileAAA
```

4. Absolute path and relative path

Use rm, mv, cp, ls with absolute path and relative path as per your choice.

- **ls:**

  Relative Path:

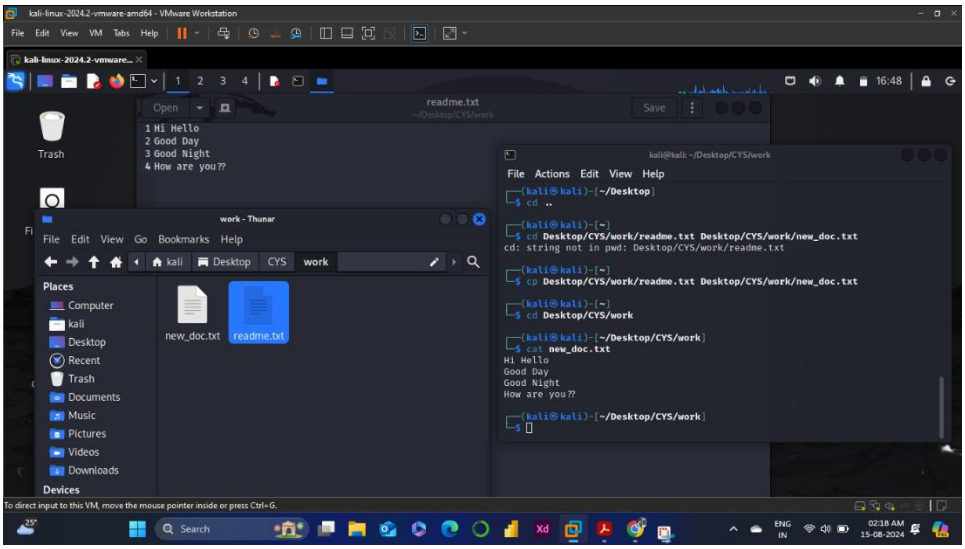
```
┌──(kali㉿kali)-[~]
└─$ ls Desktop/CYS/
chmod.exercises  File2  File4  hello       new    Unit2  work
File1            File3  File5  myfile.txt  Unit1  Unit3
```

  Absolute Path:

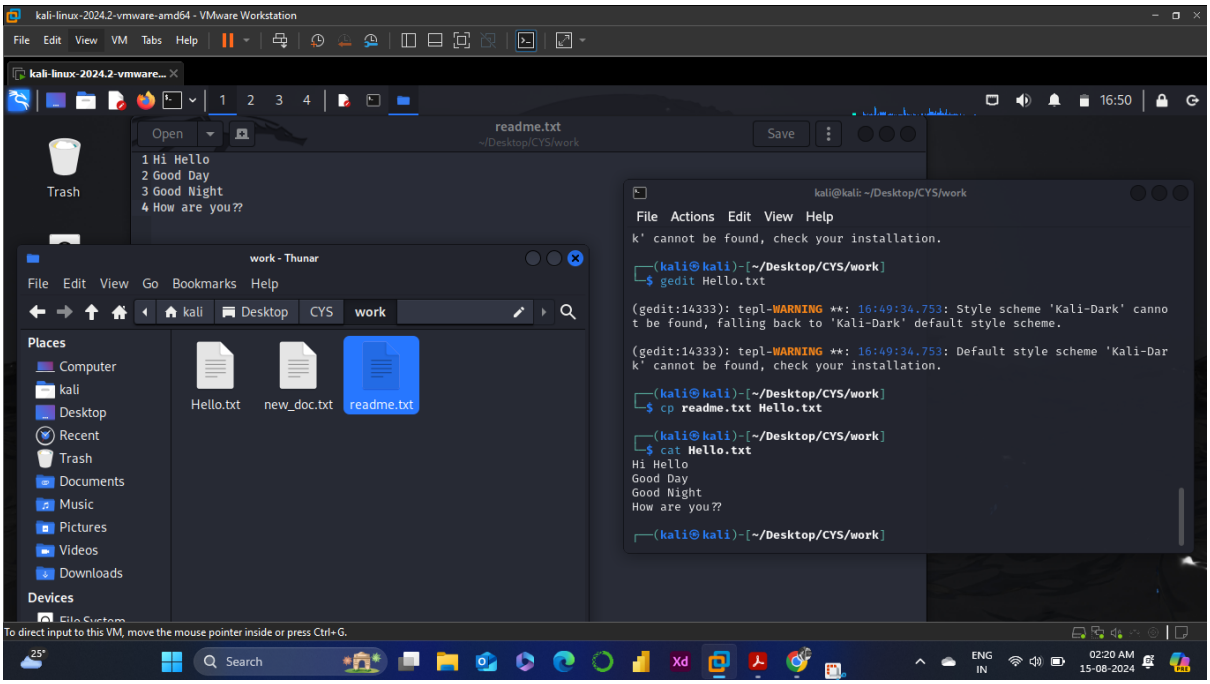
```
┌──(kali㉿kali)-[~]
└─$ ls Desktop/
Chinthan  CYS

┌──(kali㉿kali)-[~]
└─$ cd Desktop

┌──(kali㉿kali)-[~/Desktop]
└─$ ls CYS/
chmod.exercises  File2  File4  hello       new    Unit2  work
File1            File3  File5  myfile.txt  Unit1  Unit3
```

- **cp:**

  Relative Path:

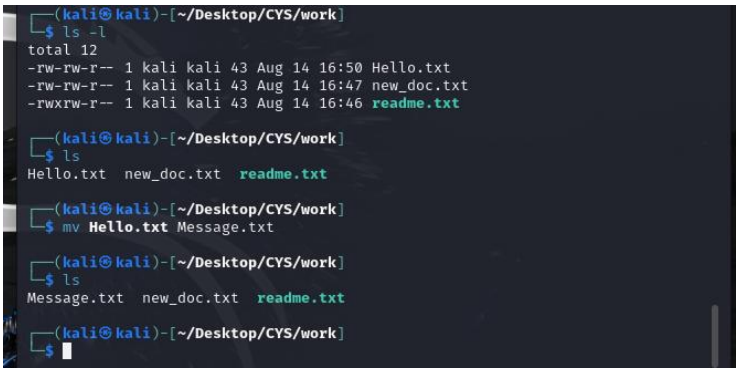


Absolute Path:



- **mv**

Relative Path:



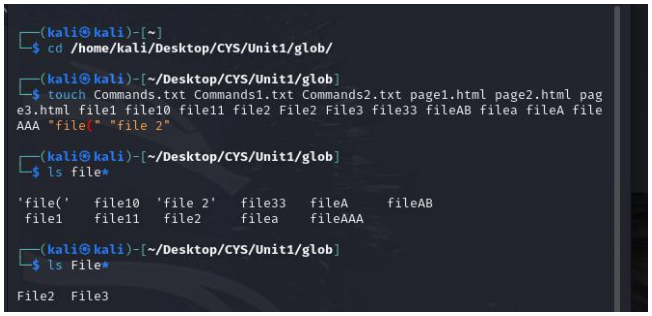
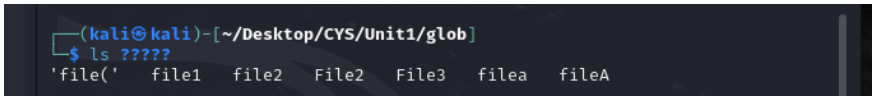
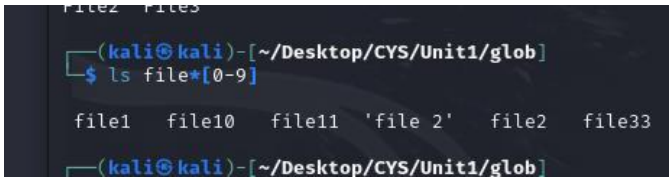
Absolute Path:

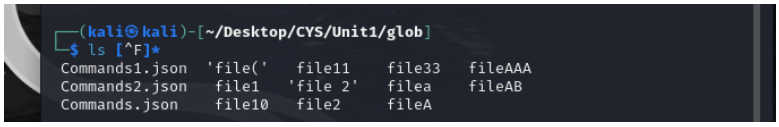
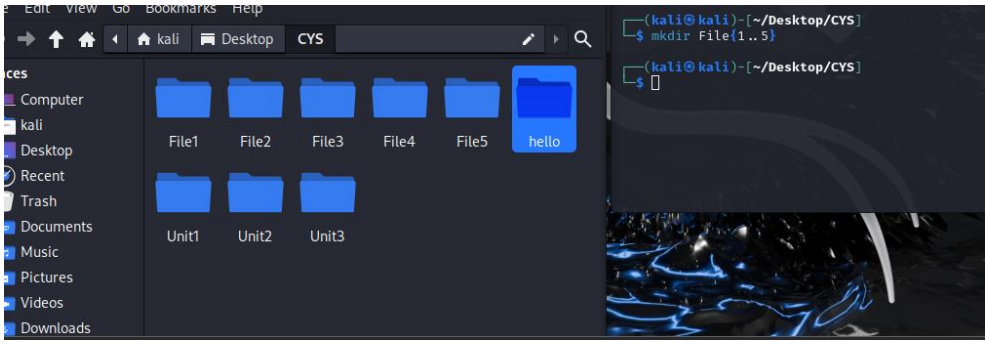
- **rm**

  Absolute path:

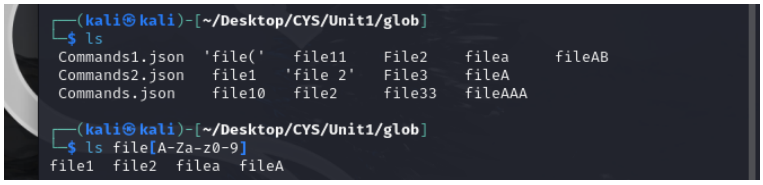
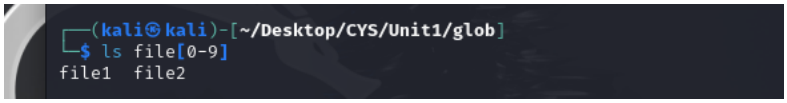
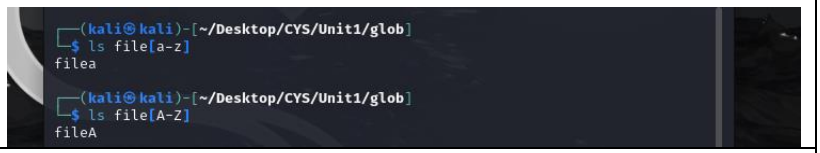
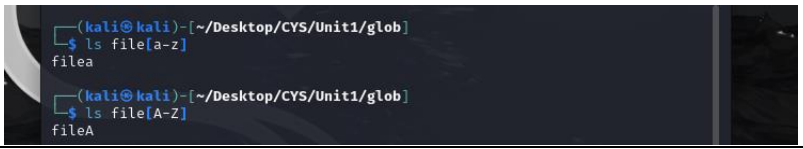
  Relative path:

5. Wildcards (Pattern Matching)

Notation	Use	Example	Screenshot
*	To match zero or more characters in file names, paths or commands	ls file*	
?	match exactly one single character in file names, paths or commands.	ls ????? (For 5 Characters)	
[ ]	Allows for flexible and specific pattern matching by defining sets or raanges	Ls file*[0-9] (File ending with digits only)	

[!] Or [^]	Matches any character that is not a member of the set characters	ls [^F]* (Fetches files which is not starting with 'F')	
{ }	Creating or manipulating multiple files or directories with varying names and numbers.	mkdir file{1..5}	

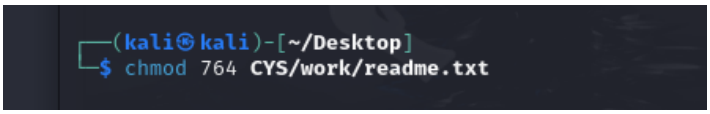
More on Character class

Notation	Use	Example	Screenshot
[:alnum:]	Matches any alphanumeric character (letters and digits).	[0-9A-Za-z]	
[:alpha:]	Matches any alphabetic character (letters only).	[A-Za-z]	
[:digit:]	Matches any digit (0-9).	[0-9]	
[:lower:]	Matches any lowercase letter (a-z)	[a-z]	
[:upper:]	Matches any uppercase letter (A-Z).	[A-Z]	

4. change permission

a) Change the permission set of /work/readme.txt so that only the user (owner) can read,write, and execute it. Use absolute mode.

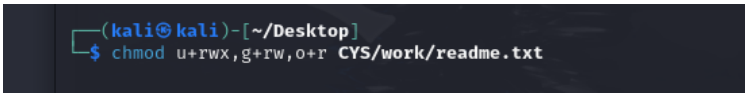
b) Change the permission set of /work/readme.txt so that any user can read it, the group can read/write to it and the user (owner) can read/write/execute it. Use absolute mode.



c) Change the permission set of /bin/bash so that only the user (owner) can read/write/ execute, group, and any user can execute it. However, whenever anyone executes it, it should run with the privileges of the owner user. Use absolute mode.

d) Change the permission set of /work/readme.txt so that only the user (owner) can read, write, and execute it. Use relative mode.

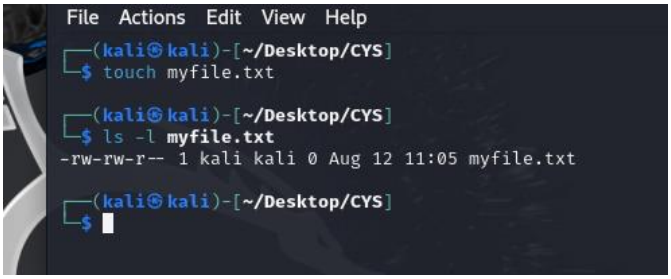
e) Change the permission set of /work/readme.txt so that any user can read it, the group can read/write to it and the user (owner) can read/write/execute it. Use relative mode.



f) Change the permission set of /work/readme.txt so that only the user (owner) can read/write/ execute, group, and any user can execute it. However, whenever anyone executes it, it should run with the privileges of the group. Use absolute mode.

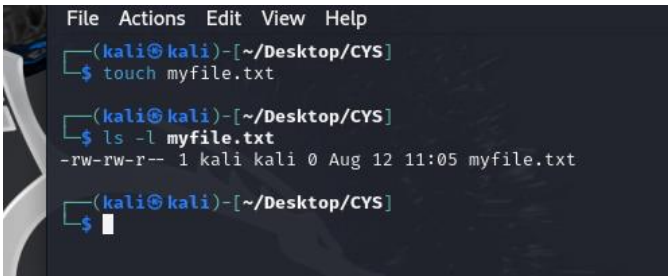
g) Change the permission set of /work/readme.txt so that only the owner can rename or delete this file while maintaining the existing permissions. Use absolute mode.

h) What are the default permissions for the new file?

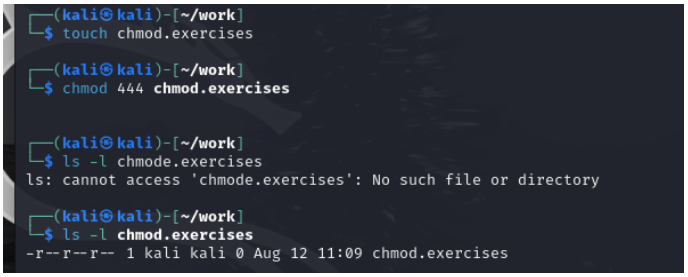


i) What was the command to view the file permissions?

ls -l <filename>



j) Change chmod.exercises permissions to -r--r--r—



k) Change the file permissions to Read only for the owner, group and all other users.

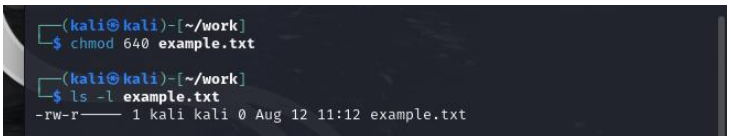
l) What was the command for changing the file permissions to -r--r--r--?

m) Change chmod.exercises permissions to -rw-r-----

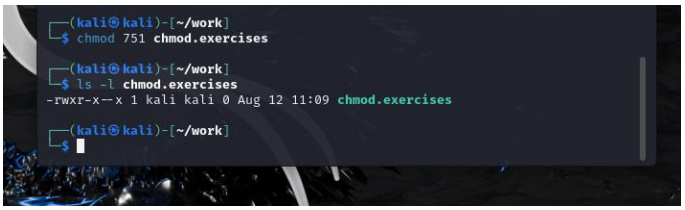
n) Change the file permissions to match the following:

a. owner: Read and Write
b. group: Read
c. other: no permissions (None)

o) What was the command for changing the file permissions to -rw-r-----?



p) Change chmod.exercises permissions to -rwxr-x—x



q) Change the file permissions to match the following:

a. owner: Read, Write and Execute
b. group: Read and Execute
c. other: Execute

r) What was the command for changing the file permissions to -rwxr-x--x?

Evaluation :

Marks : 10 (Deadline : 4 – Originality :3 – Completeness :3)

Deadline: 15.08.2024

In life there are no shortcuts. All things are connected. For success there is no fast lane. Work hard. Focus your energy, practice, remain honest, Truthful, loyal and committed.

-unknown