Assignment - 2

1. What does the command pwd, whoami, and hostname display?

Answer->

pwd

the pwd command displays full path of working directory

for example -> if we are in a document directory it shows home/document

whoami

the command whoami tells the username who has been loged in now

for example if you have logged in it shows the user name as Kumya

hostname

the command host name displays the name of machine Ur currently using

for example Ur using a machine called sonic

1. Write the command to create a directory named “project” inside the /home/Kumya folder and keep three .txt file into it. Give output snapshot ?

Answer->

┌──(kumya㉿sonic)-[~]

└─$ mkdir project

┌──(kumya㉿sonic)-[~]

└─$ cd project

┌──(kumya㉿sonic)-[~/project]

└─$ touch file1.txt

┌──(kumya㉿sonic)-[~/project]

└─$ touch file2.txt

┌──(kumya㉿sonic)-[~/project]

└─$ touch file3.txt

┌──(kumya㉿sonic)-[~/project]

└─$ ls

file1.txt file2.txt file3.txt

A screenshot of a computer

AI-generated content may be incorrect.

1. explain the difference between absolute path and relative path with proper examples.

Answer->

| **absolute Path** | **relative Path** |
| --- | --- |
| Displays the full path of file | **Describes the path of directory by a current working directory** |
| Always start from root | Starts from current working directory |
| No, its locating to same file system | Yes , it depends up on where the system has been loged in |
| begins with /. | Uses current directory or parent directory |
| It shows an exit location | when you’re working interactively inside directory |

absolute Path

┌──(kumya㉿sonic)-[~]

└─$ cat /home/kumya/project/file1.txt

Hello World

This is file1

relative Path

┌──(kumya㉿sonic)-[~]

└─$ cat project/file1.txt

Hello World

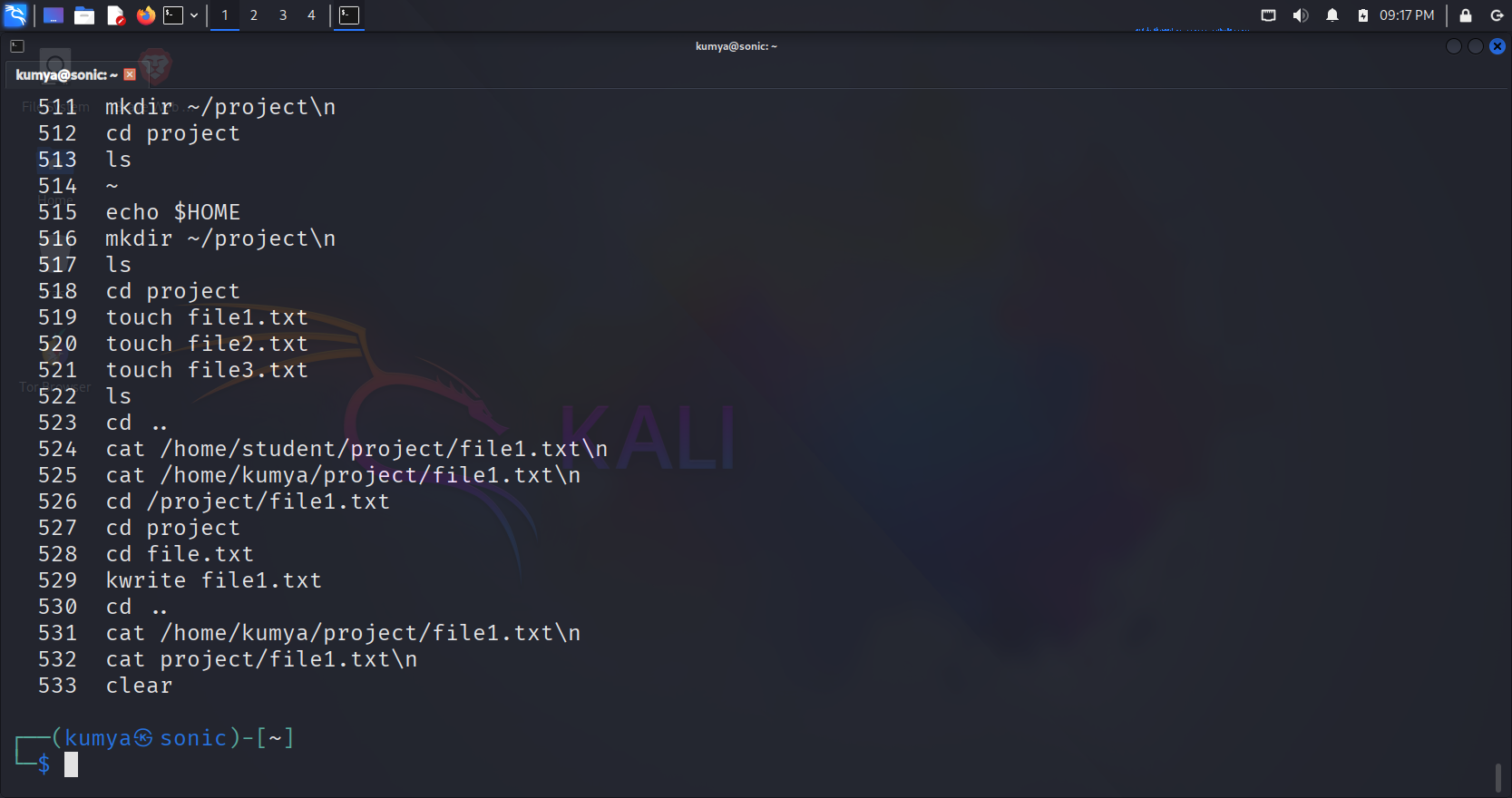
This is file1

1. what command will give you the already executed command traces in the terminal. give output snapshot.

Answer->

┌──(kumya㉿sonic)-[~]

└─$ history



You can also rerun a command from history using:

┌──(kumya㉿sonic)-[~]

└─$ !529

1. compare the working functionality of find and locate command. Which one is faster and why?

Answer->

| **find** | **locate** |
| --- | --- |
| It looks for file or directory by name ,path etc. | Looks for a file from a data base which already has been created |
| Can search any chosen directory | Search entire file system |
| Search the actual file system | Search the cached data based |
| **Slower as compared to locate** | **Faster as compared to fine command** |
| Highly flexible where it can be filter by name | Less flexible which search by file name |
| bash find /home/Kumya "file1.txt" | bash locate file1.txt |

**locate is faster** because it searches a **pre-built database** rather than scanning the filesystem in real-time like find

find

┌──(kumya㉿sonic)-[~]

└─$ find /home/kumya/project/file1.txt

/home/Kumya/project/file1.txt

locate

┌──(kumya㉿sonic)-[~]

└─$ locate file1.txt

/home/Kumya/project/file1.txt

1. which command is used to modify file permissions in Linux? Give an example.

⁕ adds **execute permission to file - file1.txt**

┌──(kumya㉿sonic)-[~/project]

└─$ chmod +x file.txt

⁕ gives full permission to owner

┌──(kumya㉿sonic)-[~/project]

└─$ chmod 777 file1.txt

1. A file has permissions -rw -r- -r- -. what does this mean?

Answer->

- → Indicates it is a regular file.

rw- (owner/user) → The owner can read and write to the file, but cannot execute the file.

r-- (group) → anyone in the group can read the file.

r-- (others) → anyone else can read the file.

1. explain the difference between Chown and chgrp with an example

Answer->

| **Chown** | **chgrp** |
| --- | --- |
| Changes the owner of file directory | Changes the group of file |
| Chown owner filename | chgrp group filename |
| Yes optically you can change both owner and group by this command | No, you can only change group name |
| This is done when a file need to be assign to new user | This is done when a file is assign to a different group |

⁕ changing file owner

┌──(kumya㉿sonic)-[~/project]

└─$ chown kumya file1.txt

⁕ changing group of file

┌──(kumya㉿sonic)-[~/project]

└─$ chgrp student file1.txt

⁕ changing both owner and group of file

┌──(kumya㉿sonic)-[~/project]

└─$ chown kumya:students file1.txt

1. A file needs to be accessible by multiple users but only writable by the owner. How will you set permissions?

Answer->

┌──(kumya㉿sonic)-[~/project]

└─$ chmod 644 file1.txt

6 ->owner: read + write (4 + 2)

4-> group: read only

4 ->others: read only

┌──(kumya㉿sonic)-[~/project]

└─$ chmod u=rw,go=r file1.txt

u-> owner read and write permission

g->group read permission

o->others read permission

┌──(kumya㉿sonic)-[~/project]

└─$ chmod 777 file1.txt

777-> gives full permission to owner

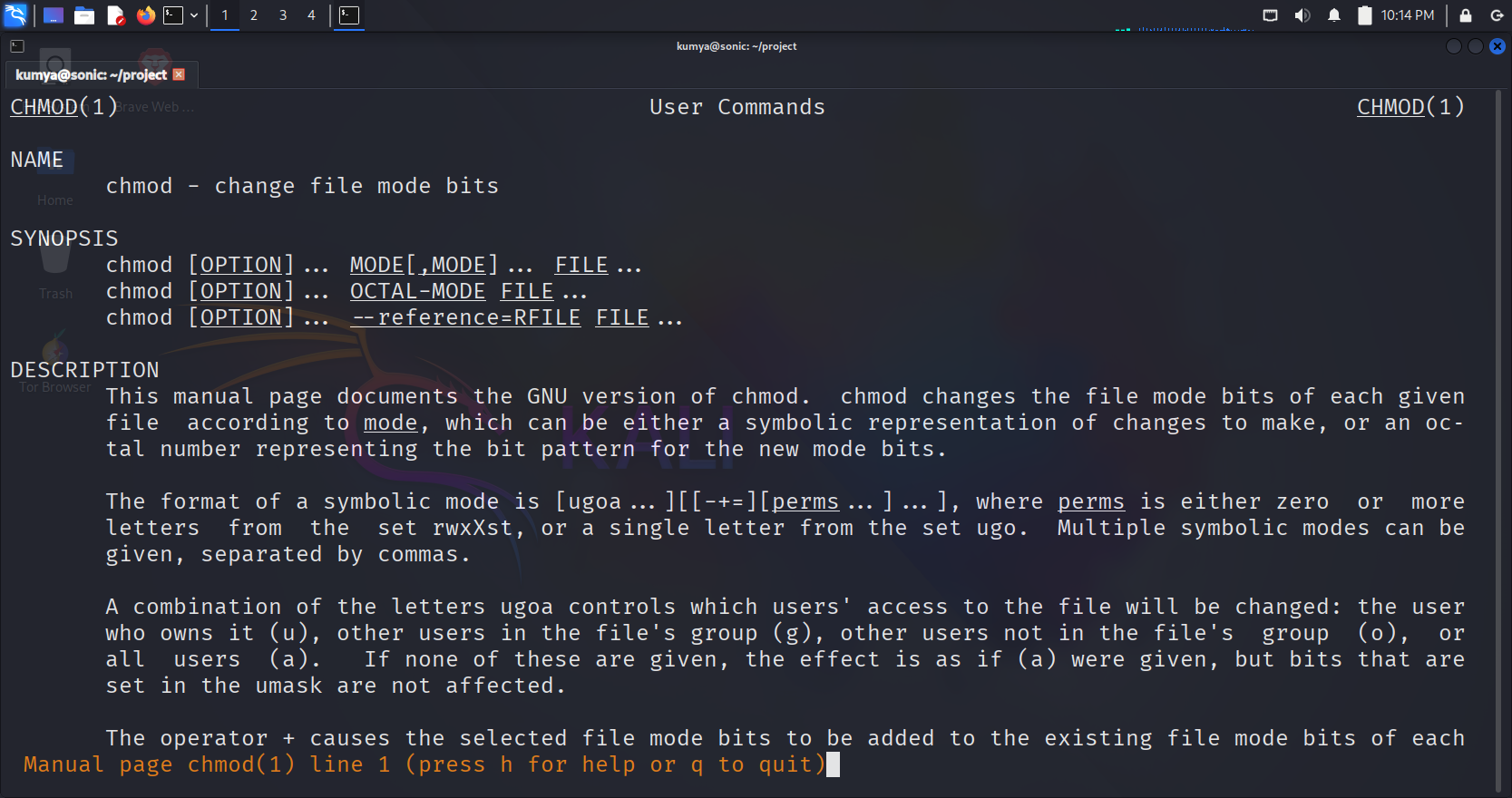
1. how do you check the manual page for any Linux commands ?

Answer->

man command name

┌──(kumya㉿sonic)-[~/project]

└─$ man chmod



-------------------------------------------END-----------------------------------------