



International Islamic University, Islamabad

Lab Task

Mastering the Linux OS

April 15, 2024

Osama Malik | 4319-BSSE-F21-A

Operating Systems

Mr. Niaz Muhammad

All linux commands.

Table of Contents

Type chapter title (level 1)	1
Type chapter title (level 2)	2
Type chapter title (level 3)	3
Type chapter title (level 1)	4
Type chapter title (level 2)	5
Type chapter title (level 3)	6

24 Linux Commands (LAB 1)

1. echo

Description of Command: This command is used to print messages or text to the terminal.

Purpose: We are displaying "Its me Osmi"

```
-sh-4.2$ echo "Its me Osmi"  
Its me Osmi  
-sh-4.2$
```

2. Pwd

Description of Command: Prints the current working directory.

Description of Command:

```
-sh-4.2$ pwd  
/home/osmi4319
```

3. ls

Lists the files and directories in the current location.

```
-sh-4.2$ ls  
-sh-4.2$
```

4. mkdir Osmi

Description of Command: Creates a directory named "Osmi."

```
-sh-4.2$ mkdir Osmi  
-sh-4.2$ ls  
Osmi  
-sh-4.2$
```

5. cd Osmi

Description of Command: Changes the current directory to "Osmi."

```
-sh-4.2$ cd Osmi  
-sh-4.2$ pwd  
/home/osmi4319/Osmi  
-sh-4.2$
```

6. cat>myfile

Description of Command: Creates a file named "myfile" within the "Osmi" directory for text input.

```
-sh-4.2$ cat>myfile
Osama Malik 4319 BSSE/F21 A. I am undergraduate Software Engineer
```

7. cat Osmi

Description of Command: Displays the contents of the "myfile" file

```
-sh-4.2$ pwd
/home/osmi4319/Osmi
-sh-4.2$ cat myfile
Osama Malik 4319 BSSE/F21 A. I am undergraduate Software Engineer
```

8. date

Description of Command: Prints the current date and time.

```
-sh-4.2$ date
Sun Feb 25 18:53:58 UTC 2024
```

9. date +%b

Description of Command: Displays the abbreviated month name.

```
-sh-4.2$ date +%b
Feb
```

10. date +%B

Description of Command: Displays the Full month name.

```
-sh-4.2$ date +%B
February _
```

11. date +%d

Description of Command: Prints the day of the month (numeric representation).

```
-sh-4.2$ date +%d
25
```

12. date +%D

Description of Command: Prints the date in MM/DD/YY format.

```
-sh-4.2$ date +%D  
02/25/24
```

13. date +%c

Description of Command: Displays the date and time representation using the locale's date and time format.

```
-sh-4.2$ date +%c  
Sun 25 Feb 2024 07:01:44 PM UTC
```

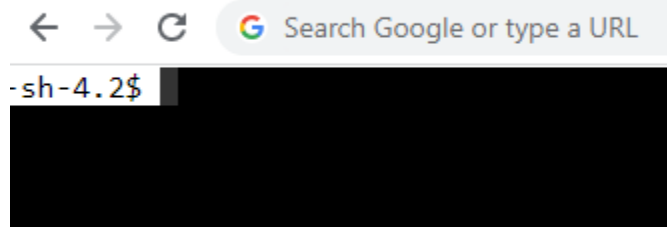
14. date +%C

Description of Command: Prints the first two digits of the four-digit year (century).

```
-sh-4.2$ date +%C  
20
```

15. clear

Description of Command: Clears the terminal screen.



```
← → ↻ Search Google or type a URL  
-sh-4.2$
```

16. cal 2024

Description of Command: Displays the calendar for the year 2024.

```
-sh-4.2$ cal 2024
```

2024																											
January							February							March													
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa							
	1	2	3	4	5	6					1	2	3						1	2							
7	8	9	10	11	12	13	4	5	6	7	8	9	10	3	4	5	6	7	8	9							
14	15	16	17	18	19	20	11	12	13	14	15	16	17	10	11	12	13	14	15	16							
21	22	23	24	25	26	27	18	19	20	21	22	23	24	17	18	19	20	21	22	23							
28	29	30	31				25	26	27	28	29			24	25	26	27	28	29	30							
														31													
April							May							June													
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa							

17. whoami

Description of Command: Prints the current username.

```
-sh-4.2$ whoami
osmi4319
```

18. free

Description of Command: Shows the amount of free and used system memory.

```
-sh-4.2$ free
```

	total	used	free	shared	buff/cache	available
Mem:	16399684	4275832	2649860	904232	9473992	10873752
Swap:	0	0	0			

19. mv myfile Osmi2

Description of Command: Moves the file "myfile" to "Osmi2" directory.

```
-sh-4.2$ cat myfile
Osama Malik 4319 BSSE/F21 A. I am undergraduate Software Engineer

cat
-sh-4.2$ mv myfile Osmi2
-sh-4.2$ cat myfile
cat: myfile: No such file or directory
-sh-4.2$
```

20. cp myfile Osmi

Description of Command: Copies the file "myfile" to "Osmi" directory.

```
-sh-4.2$ cp myfile Osmi
```

21. rm myfile

Description of Command: Removes the "myfile" file.

```
-sh-4.2$ rm myfile
-sh-4.2$ cat myfile
cat: myfile: No such file or directory
```

22. cd ..

Description of Command: Go back to the previous directory.

```
-sh-4.2$ pwd
/home/osmi4319/Osmi/Osmi2
-sh-4.2$ cd ..
-sh-4.2$ pwd
/home/osmi4319/Osmi
```

23. rmdir Osmi2

Description of Command: Removes the "Osmi2" directory

```
-sh-4.2$ ls
Osmi2
-sh-4.2$ rmdir Osmi2
-sh-4.2$ ls
-sh-4.2$
```

24. exit

Description of Command: Exits the terminal session.

```
-sh-4.2$ exit
logout
Session closed.
```


Linux Commands (LAB 2)

1. abc.txt|wc

Description of Command: This command is used to count the number of lines, words, and characters in the "abc.txt" file.

Purpose: Counting the lines, words, and characters in the file "abc.txt".

```
-sh-4.2$ cat> abc.txt
My Name is Osama Malik
I am Software Engineer.
-sh-4.2$ ls
abc.txt mnp [REDACTED]
-sh-4.2$ cat abc.txt|wc
      2      9     47
```

2. abc.txt|wc -l

Description of Command: This command is used to count the number of lines in the "abc.txt" file.

Purpose: Counting the number of lines in the file "abc.txt".

```
-sh-4.2$ ls
abc.txt mnp [REDACTED]
-sh-4.2$ cat abc.txt|wc
      2      9     47
```

3. who|wc -l

Description of Command: This command is used to count the number of lines produced by the who command, which lists the currently logged-in users.

Purpose: Counting the number of logged-in users.

```
-sh-4.2$ whoami
osmi4319
-sh-4.2$ who|wc -l
0
```

4. cat .txt|wc

Description of Command: This command is used to count the number of lines, words, and characters in all files with a ".txt" extension.

Purpose: Counting lines, words, and characters in multiple text files.

```
-sh-4.2$ ls
abcd.txt abc.txt mnp [REDACTED]
-sh-4.2$ cat *.txt|wc
      3     11     59
```

5. cat *.mp4 | wc

Description of Command: This command is used to count the number of lines, words, and characters in all files with a ".mp4" extension like we have here file.mp4.

Purpose: Counting lines, words, and characters in multiple video files (though this command might not be meaningful for binary files like videos).

```
-sh-4.2$ ls
abcd.txt  abc.txt  file.mp4  mnp
-sh-4.2$ cat *.mp4 | wc
      3      3     14
```

6. wc abc.txt

Description of Command: This command is used to count the number of lines, words, and characters in the "abc.txt" file.

Purpose: Counting lines, words, and characters in the file "abc.txt".

```
-sh-4.2$ cat abc.txt
My Name is Osama Malik
I am Software Engineer.
-sh-4.2$ wc abc.txt
 2  9 47 abc.txt
```

7. ls -l

Description of Command: This command is used to list files and directories in the current directory, with each entry on a separate line.

Purpose: Listing files and directories in a single column.

```
-sh-4.2$ ls -l
abcd.txt
abc.txt
file.mp4
mnp
```

8. ls -l abc.txt mp4.txt

Description of Command: This command is used to list the "abc.txt" and "file.mp4" files in the current directory, each on a separate line.

Purpose: Listing specific files in a single column.

```
-sh-4.2$ ls
abcd.txt  abc.txt  file.mp4  mnp
-sh-4.2$ ls -l abc.txt file.mp4
abc.txt
file.mp4
```

9. ls -l .txt

Description of Command: This command is used to list all files with a ".txt" extension in the current directory, each on a separate line.

Purpose: Listing text files in a single column.

```
file.mp4
-sh-4.2$ ls -l *.txt
abcd.txt
abc.txt
```

10. ls -l -a

Description of Command: This command is used to list files and directories in the current directory in long format, including hidden files.

Purpose: Detailed listing of all files and directories, including hidden ones.

```
-sh-4.2$ ls -l -a
total 16
drwxrwxr-x. 3 osmi4319 osmi4319 81 Mar 14 01:04
.
drwx----- 4 osmi4319 osmi4319 147 Feb 29 06:10
..
-rw-rw-r--. 1 osmi4319 osmi4319 12 Mar 14 00:56
abcd.txt
-rw-rw-r--. 1 osmi4319 osmi4319 47 Mar 14 00:50
abc.txt
-rw-rw-r--. 1 osmi4319 osmi4319 14 Mar 14 00:57
file.mp4
drwxrwxr-x. 3 osmi4319 osmi4319 17 Feb 29 06:10
mnp
-rw-rw-r--. 1 osmi4319 osmi4319 6 Mar 14 01:05
.osmi.txt
```

11. ls -la

Description of Command: This command is used to list files and directories in the current directory in long format, including hidden files.

Purpose: Detailed listing of all files and directories, including hidden ones.

```
-sh-4.2$ ls -la
total 16
drwxrwxr-x. 3 osmi4319 osmi4319 81 Mar 14 01:04
.
drwx----- 4 osmi4319 osmi4319 147 Feb 29 06:10
..
-rw-rw-r--. 1 osmi4319 osmi4319 12 Mar 14 00:56
abcd.txt
-rw-rw-r--. 1 osmi4319 osmi4319 47 Mar 14 00:50
abc.txt
-rw-rw-r--. 1 osmi4319 osmi4319 14 Mar 14 00:57
file.mp4
drwxrwxr-x. 3 osmi4319 osmi4319 17 Feb 29 06:10
mnp
-rw-rw-r--. 1 osmi4319 osmi4319 6 Mar 14 01:05
.osmi.txt
-sh-4.2$ ls -la *.txt
-rw-rw-r--. 1 osmi4319 osmi4319 12 Mar 14 00:56
abcd.txt
-rw-rw-r--. 1 osmi4319 osmi4319 47 Mar 14 00:50
abc.txt
```

12. ls -la *.txt

Description of Command: This command is used to list all files with a ".txt" extension in the current directory in long format, including hidden files.

Purpose: Detailed listing of text files, including hidden ones.

```
-sh-4.2$ ls -la *.txt
-rw-rw-r--. 1 osmi4319 osmi4319 12 Mar 14 00:56
abcd.txt
-rw-rw-r--. 1 osmi4319 osmi4319 47 Mar 14 00:50
abc.txt
```

13. files = "abc.txt H-lctxt"

Description of Command: This command creates a variable named \$files containing the filenames "abc.txt" and "H-lctxt".

Purpose: Storing file names in a variable for later use.

```
-sh-4.2$ files="abc.txt abcd.txt"
```

14. cat \$files

Description of Command: This command is used to display the contents of files stored in the variable \$files.

Purpose: Displaying the contents of specified files.

```
-sh-4.2$ cat $files
My Name is Osama Malik
I am Software Engineer.
Osama Malik
```

15. echo \$file

Description: This command prints the value stored in the \$file variable.

Purpose: To display the value stored in a variable.

```
-sh-4.2$ echo $file
-sh-4.2$
```

Linux Commands (LAB 3)

1. code: ls

Description:

The ls command is used to list the contents of a directory.

Purpose:

The purpose of ls is to display the names of files and directories in the current directory by default, or in a specified directory if provided.

screenshot:

```
-sh-4.2$ ls  
abc  mystory
```

2. code: echo \$PATH

Description:

The echo command is used to print text or values to the terminal, and \$PATH is an environment variable that stores a list of directories where executable files are located.

Purpose:

The purpose of echo \$PATH is to print the value of the PATH environment variable, which indicates the directories that the shell searches for executable files when a command is entered.

screenshot:

```
-sh-4.2$ echo $PATH  
/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin  
-sh-4.2$ cat /usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin  
cat: /usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin: No such file or directory  
-sh-4.2$ cd /usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin  
-sh: cd: /usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin: No such file or directory
```

3. code: !!s

Description:

The ! character followed by a command (in this case, ls) is used to execute the most recent command in the command history that starts with the specified string.

Purpose:

The purpose of !!s is to execute the most recent command in the command history that starts with ls. This is useful for quickly repeating a previous command without typing it out again. ls -l .txt

screenshot:


```

abc mystory [REDACTED]
-sh-4.2$ ls
ls
abc mystory [REDACTED]

```

4. **code:** for file in *.txt; do mv -v \$file \$file.old; done

Description:

This is a for loop in Bash scripting. It iterates over all files in the current directory with a .txt extension and renames each file by appending .old to its name.

Purpose:

The purpose of this loop is to rename all files with a .txt extension in the current directory by adding the .old suffix to their names.

```

-sh-4.2$ for file in *.txt; do mv -v $file $file.old; done
'abcd.txt' -> 'abcd.txt.old'
'abc.txt' -> 'abc.txt.old'

```

5. **code:**ls -la

Description: this command will show the all/all hidden .txt files and there date and time when they were created.

screenshot:

```

-sh-4.2$ ls -la
total 8
drwxrwxr-x. 3 osmi4319 osmi4319  47 Apr 18 04:51 .
drwx----- 6 osmi4319 osmi4319 189 Apr 18 04:01 ..
-rw-rw-r-- 1 osmi4319 osmi4319   9 Apr 18 04:51 abc.txt
-rw-rw-r-- 1 osmi4319 osmi4319   6 Apr 18 04:00 mnp.txt
drwxrwxr-x. 3 osmi4319 osmi4319  21 Apr 18 04:16 new [REDACTED]

```

6. **code:** ls -la *.txt

Description: this command will show the all .txt files and there date and time when they were created.

screenshot:

```

-sh-4.2$ ls -la *.txt
-rw-rw-r-- 1 osmi4319 osmi4319 9 Apr 18 04:51 abc.txt
-rw-rw-r-- 1 osmi4319 osmi4319 6 Apr 18 04:00 mnp.txt

```

7. **code:** *abc.txt | wc

Description: this command will show the .txt files data like line characters and letters but this is not showing anything so we will write cat with it.

screenshot:

```

-sh: abc.txt: command not found
0      0      0

```

8. **code:** *abc.txt | wc -l

Description:

this command will show the .txt files data like lines only but its showing nothing so we will write cat with it.

screenshot:

```
-sh: abc.txt: command not found
0      0      0
```

9. code: Command: for i in 123456; do \$i; done

Description:

This is a for loop in Bash scripting. It iterates over the sequence of characters "123456" (not individual numbers) and attempts to execute each character as a command.

Purpose:

The purpose of this loop is to display 123456

```
-sh-4.2$ for i in 123456; do echo $i; done
123456
```

10. code: cat *abc.txt | wc

Description: this command will show the .txt files data like line characters and letters

screenshot:

```
-sh-4.2$ cat *abc.txt | wc
1      3      9
```

11. code: cat *abc.txt | wc -l

Description: this command will show the .txt files data lines only.

screenshot:

```
-sh-4.2$ cat *abc.txt | wc -l
1
```

12. code: files="abc.txt mpn.txt"

Description: this command will assign a string containing multiple filenames to the variable files.

screenshot:

```
-sh-4.2$ files="abc.txt mpn.txt"
-sh-4.2$ cat $files
I am ...
Osama
```

13. code: cat \$files

Description: this command will display all the content in the above files assigned to variable through variable name.

screenshot:

```
-sh-4.2$ files="abc.txt mpn.txt"
-sh-4.2$ cat $files
I am ...
Osama
```

14. code: cat abc.txt

Description: this command will display all the content in the file.
screenshot:

```
-sh-4.2$ cat abc.txt  
I am ...  
-sh-4.2$ █
```

Linux Commands (LAB 4)

1. for i in {1..8}; do echo \$i sleep 1 done

| Loop and Print Numbers |

This command uses a for loop to iterate through numbers from 1 to 8. Inside the loop, it prints the current value of i using echo and then pauses for 1 second using sleep 1. You might use this for simple counting or creating timed delays in a script.

```
-sh-4.2$ for i in {1..8};  
> do  
> echo $i;  
> sleep 1  
> done  
1  
2  
3  
4  
5  
6  
7  
8
```

2. for file in *.txt; do echo \$file sleep 1 done

| Loop and Print Text Files |

This command iterates through all files with the .txt extension in the current directory. Inside the loop, it prints the filename using echo and then pauses for 1 second using sleep 1. This can be useful for listing text files or creating a timed process for working with them.

```
-sh-4.2$ for file in *.txt; do echo $file; sleep 1; done  
abc.txt  
mnp.txt
```

3. for file in *.mp3; do echo \$file sleep 1 done

| Loop and Print Text Files |

This command iterates through all files with the .mp3 extension in the current directory. Inside the loop, it prints the filename using echo and then pauses for 1 second using sleep 1. This can be useful for listing text files or creating a timed process for working with them.

```
-sh-4.2$ for file in *.mp3; do echo $file; sleep 1; done  
abc.mp3  
mnp.mp3
```

4. ls -l \$(locate abc.txt)

| List Details of Located File |

This command combines locate and ls -l. First, locate abc.txt searches for the file named "abc.txt" using the system's file location database. Then, the output (the file path) is captured within parentheses \$() and passed to ls -l to display detailed information about the located file, including permissions, owner, group, and size. Use this to get detailed file information after a quick search using locate.

```
-sh-4.2$ ls -l $(locate abc.txt)
-sh: locate: command not found
total 20
drwxrwxr-x. 3 osmi4319 osmi4319 89 Mar 14 05:34 abc
-rw-rw-r--. 1 osmi4319 osmi4319 3 Mar 28 03:22 abc.mp3
-rw-rw-r--. 1 osmi4319 osmi4319 3 Mar 28 03:23 abc.txt
-rw-rw-r--. 1 osmi4319 osmi4319 11 Mar 28 03:22 mnp.mp3
-rw-rw-r--. 1 osmi4319 osmi4319 17 Mar 28 03:22 mnp.txt
-rw-rw-r--. 1 osmi4319 osmi4319 134 Feb 29 05:42 mystory
```

5. find -name mnp.mp3

| Find Specific File |

This command uses find to search for a file named exactly "mnp.mp3" in the current directory and its subdirectories. It's useful for locating a specific file by its complete name.

```
-sh-4.2$ find -name abc.mp3
./Osmi/abc.mp3
./abc.mp3
```

6. find -name 'abc*'

| Find Files with Matching Patterns (Wildcard) |

This command searches for all files that start with "abc" (including any characters after "abc") in the current directory and its subdirectories. The asterisk (*) acts as a wildcard to match any characters. Use this to find groups of files with similar names.

```
-sh-4.2$ find -name 'abc*'
./Osmi/mnp/abc
./Osmi/abc.txt
./Osmi/abc.mp3
./abc
./abc/mnp/abc
./abc/abcd.txt.old
./abc/abc.txt.old
./abc.mp3
./abc.txt
```

7. find -name 'mnp*'

| Find Files with Matching Patterns (Wildcard) |

Similar to command 5, this searches for all files that start with "mnp" (including any characters after "mnp") in the current directory and its subdirectories.

```
-sh-4.2$ find -name 'mnp*' -exec cp '{}' /home/osmi4319/abc ';'
cp: omitting directory './Osmi/mnp'
cp: omitting directory './abc/mnp'
```

8. find -name 'mnp*' -exec cp /home/osmi4319 {} \;

| Find and Copy Files (Caution Advised) |

This command is more complex. It searches for files starting with "mnp*" and then attempts to copy them to the directory /home/osmi4319. The -exec option allows executing a command (cp) on each found file, represented by {}. Warning: Be cautious with this command, as incorrect paths or file permissions can lead to unintended consequences. Ensure you have the necessary permissions and the destination directory exists before using it.

```
abc abc.mp3 abc.txt mystory
-sh-4.2$ pwd
/home/osmi4319
-sh-4.2$ find -name 'abc*' -exec rm '{}' ';'
rm: cannot remove './Osmi/mnp/abc': Is a directory
rm: cannot remove './abc': Is a directory
rm: cannot remove './abc/mnp/abc': Is a directory
-sh-4.2$ ls
abc mystory
```

9. for file in *.mp3; do rm -v \$file; done

| Loop and Delete MP3 Files (Caution Advised) |

This command iterates through all .mp3 files and deletes them one by one. The -v flag with rm provides verbose output, showing each file being deleted. Warning: Use extreme caution with this command, as deleted files cannot be easily recovered. Make sure you have a backup or are absolutely certain you want to delete the files before using it.

```
-sh-4.2$ for file in *.txt;  
> do  
> rm -v $file;  
> done  
removed 'abc.txt'  
removed 'abdc.txt'  
-sh-4.2$
```

10. find -type d | Find Directories

| Find Directories |

This command searches for directories (folders) in the current directory and its subdirectories. It's helpful for listing or working with directories within your file system.

```
sh-4.2$ find -type d
```

```
/.Osmi  
/.Osmi/mnp  
/.Osmi/mnp/abc  
/.Osmi/mnp/abc/n  
/abc  
/abc/mnp  
/abc/mnp/abc  
/abc/mnp/abc/n
```

11. find -type f

| Find Files |

This command searches for regular files (not directories) in the current directory and its subdirectories. Use this to locate specific files or groups of files that are not directories.


```
./abc/mnp/abc/n
-sh-4.2$ find -type f
./bashrc
./bash_profile
./bash_logout
./magic_string.txt
./bash_history
./mystory
./Osmi/mnp/abc/n/abFn.mp3
./abc/mnp/abc/n/abFn.mp3
./abc/mnp/abc/abFn.mp3
./abc/file.mp4
./abc/.osmi.txt
-sh-4.2$
```

Linux Commands (LAB 5)

1. CODE: find -type d

DESCRIPTION: It will show the tree directories from the directory we are in but in forward direction.

SCREENSHOT:

```
-sh-4.2$ pwd
/home/osmi4319/myFiles
-sh-4.2$ ls
abc.txt  mnp.txt  new
-sh-4.2$ find -type d
.
./new
./new/myFiles
```

2. CODE: find -type f

DESCRIPTION: It will show the regular files from the directory we are in but in forward direction.

SCREENSHOT:

```
-sh-4.2$ find -type f
./mnp.txt
./abc.txt
```

9. CODE: for file in *.txt

>do

>rm -v \$file;

>done

DESCRIPTION: by using this loop command we are removing all the txt files from the directory we are in.

SCREENSHOT:

```

./mnp.txt
./abc.txt
-sh-4.2$ for file in *.txt
> do
> rm -v $file;
> done
rm: cannot remove '*.txtfor': No such file or directory
rm: cannot remove 'file': No such file or directory
rm: cannot remove 'in': No such file or directory
removed 'abc.txt'
removed 'mnp.txt'
-sh-4.2$ ls
new

```

3. CODE: find -name abc.txt

DESCRIPTION: by using this command we can find the file with the name we are entering in the command and the terminal will show us is it present or not.

SCREENSHOT:

```

-sh-4.2$ ls
abc.txt mnp.txt new
-sh-4.2$ find -name abc.txt
./abc.txt

```

4. CODE: ls -l 'locate abc.txt'

DESCRIPTION: this command not found as you told us to try this command too.

SCREENSHOT:

```

-sh-4.2$ ls -l 'locate abc.txt'
ls: cannot access 'locate: No such file or directory
ls: cannot access abc.txt': No such file or directory

```

5. CODE: ls -l \$locate abc.txt

DESCRIPTION: this command will give the information about time date and directory where the file is present.

SCREENSHOT:

```

-sh-4.2$ -l $locate abc.txt
-rw-rw-r--. 1 osmi4319 osmi4319 6 Apr 18 05:04 abc.txt

```

6. CODE: `find -name 'abc*' -exec rm '{}' \;`

DESCRIPTION: this command will find the name of the file we entered and will only remove files other than the directories.

Purpose: we used this command to delete all the files name start with abc.

SCREENSHOT:

```
-sh-4.2$ find -name 'abc*' -exec rm '{}' \;  
-sh-4.2$ LS  
-sh: LS: command not found  
-sh-4.2$ ls  
mnp.txt new
```

7. CODE: `find -name 'newfile*' -exec cp '{}' /home/osmi4319/myFiles \;`

DESCRIPTION: this command will find the name of the file we entered and will only copy the files to the directory path we entered.

SCREENSHOT:

```
-sh-4.2$ find -name 'newfile*' -exec cp '{}' /home/osmi4319/myFiles \;  
-sh-4.2$ cd myFile  
-sh: cd: myFile: No such file or directory  
-sh-4.2$ ls  
abc data mnp.txt myFiles mystory newfile.txt  
-sh-4.2$ pwd  
/home/osmi4319  
-sh-4.2$ cd myFiles  
-sh-4.2$ ls  
abc.txt mnp.txt new newfile.txt
```

8. CODE: `find -name 'newfile*' -exec mv '{}' /home/osmi4319/myFiles \;`

DESCRIPTION: this command will find the name of the file we entered and will only move the files to the directory path we entered.

Purpose: We used this command to

SCREENSHOT:

```
-sh-4.2$ find -name 'newfile*' -exec cp '{}' /home/osmi4319/myFiles ';'
-sh-4.2$ cd myFile
-sh: cd: myFile: No such file or directory
-sh-4.2$ ls
abc data mnp.txt myFiles mystory newfile.txt
-sh-4.2$ pwd
/home/osmi4319
-sh-4.2$ cd myFiles
-sh-4.2$ ls
abc.txt mnp.txt new newfile.txt
-sh-4.2$
```

Linux Commands (LAB 6)

1) Command: pushd**Description:**

The pushd command is used to push directories onto the directory stack, allowing you to change the current directory and save the previous one for later retrieval.

Purpose:

I am using this command to push my abc directory to directory stack.

```
-sh-4.2$ pushd abc
~/abc ~
-sh-4.2$ ls
file.mp4 mnp [REDACTED]
-sh-4.2$ pwd
```

2) Command: popd**Description:**

The popd command is used to remove the top directory from the directory stack and change the current directory to the one that was most recently pushed onto the stack.

Purpose:

I used this to go back to osmi4319 directory.

```
-sh-4.2$ popd
~
-sh-4.2$ ls
abc data mystory [REDACTED]
-sh-4.2$ pwd
/home/osmi4319
```

3) Command: rm - data.***Description:**

The rm command is used to remove or delete files or directories, and the - option followed by a pattern is used to specify files or directories based on a wildcard pattern. In this case, data.* represents files with any extension starting with "data."

Purpose:

The purpose of rm - data.* is to delete files matching the specified pattern (data.*) within the current directory without prompting for confirmation for each file.

```
-sh-4.2$ rm -data.*
rm: invalid option -- 'a'
Try 'rm --help' for more information.
-sh-4.2$ rm -data*
rm: invalid option -- 'a'
Try 'rm --help' for more information.
```

4) Command: `rm data.*` (delete without displaying)

Description:

Similar to the previous command, `rm` is used to delete files or directories, and `data.*` represents files with any extension starting with "data." However, without the `-` option, this command doesn't suppress the display of the files being deleted.

Purpose:

The purpose of `rm data.*` is to delete files matching the specified pattern (`data.*`) within the current directory while displaying each file's name as it is deleted. This provides a visual confirmation of the files being removed.

```
-sh-4.2$ rm -data*
rm: invalid option -- 'a'
Try 'rm --help' for more information.
-sh-4.2$ rm data.*
rm: cannot remove 'data.*': No such file or directory
-sh-4.2$ rm data*
rm: cannot remove 'data*': No such file or directory
```

5) Command: `cp -i abc.txt mnp.txt osmi4319/myFiles`

Description:

The `cp` command is used to copy files or directories, and the `-i` option prompts the user for confirmation before overwriting existing files. In this command, `abc.txt` is the source file, `mnp.txt` is the destination file, and `osmi4319/myFiles` is the destination directory.

Purpose:

The purpose of `cp -i abc.txt mnp.txt osmi4319/abc` is to copy the file `abc.txt` to the directory `osmi4319/myFiles` with the name `mnp.txt`, while prompting for confirmation if a file with the same name already exists in the destination directory.

```
-sh-4.2$ cp -i abc.txt mnp.txt osmi4319/abc
cp: target 'osmi4319/abc' is not a directory
-sh-4.2$ cp -i abc.txt mnp.txt /osmi4319/abc
cp: target '/osmi4319/abc' is not a directory
-sh-4.2$ mv -i abc.txt mnp.txt /osmi4319/abc
mv: target '/osmi4319/abc' is not a directory
-sh-4.2$ cp -i abc.txt mnp.txt /abc
cp: target '/abc' is not a directory
-sh-4.2$ pwd
/home/osmi4319
```

6) Command: mv -i abc.txt mnp.txt osmi4319/abc

Description:

The mv command is used to move or rename files or directories, and the -i option prompts the user for confirmation before overwriting existing files. In this command, abc.txt is the source file, mnp.txt is the destination file, and osmi4319/myFiles is the destination directory.

Purpose:

The purpose of mv -i abc.txt mnp.txt osmi4319/myFiles is to move the file abc.txt to the directory osmi4319/myFiles and rename it to mnp.txt, while prompting for confirmation if a file with the same name already exists in the destination directory.

```
-sh-4.2$ cp -i abc.txt mnp.txt osmi4319/abc
cp: target 'osmi4319/abc' is not a directory
-sh-4.2$ cp -i abc.txt mnp.txt /osmi4319/abc
cp: target '/osmi4319/abc' is not a directory
-sh-4.2$ mv -i abc.txt mnp.txt /osmi4319/abc
mv: target '/osmi4319/abc' is not a directory
-sh-4.2$ cp -i abc.txt mnp.txt /abc
cp: target '/abc' is not a directory
-sh-4.2$ pwd
/home/osmi4319
```

7) Command: cp -a abc abc (One directory to another directory)

Description:

The cp command is used to copy files or directories, and the -a option is used to preserve the original file attributes (such as ownership and permissions). In this command, abc is the source directory, and myFiles is the destination directory.

Purpose:

The purpose of cp -a abc myFiles is to copy the entire contents of the directory abc, including its subdirectories and files, into the directory myFiles while preserving their attributes. This is useful for creating backups or duplicating directory structures with their associated metadata intact.

```
/home/osmi4319
-sh-4.2$ cp -a abc data
-sh-4.2$ pwd
/home/osmi4319
-sh-4.2$ ls
abc data mystory [REDACTED]
-sh-4.2$ cd data
-sh-4.2$ ls
abc mydata [REDACTED]
```

8) Command: `cp -i mystory abc`

Description:

The `cp` command is used to copy files or directories, and the `-i` option prompts the user for confirmation before overwriting existing files. In this command, `mystory` is the source file.

Purpose:

The purpose of `cp -i mystory` is to copy the directory `abc`.

```
-sh-4.2$ cp -i mystory abc
-sh-4.2$ pwd
/home/osmi4319
-sh-4.2$ ls
abc data mystory [REDACTED]
-sh-4.2$ cd abc
-sh-4.2$ ls
abc.txt file.mp4 mnp mnp.txt mystory [REDACTED]
```

9) Command: `cp -a /home/osmi4319/myFiles`

Description:

The `cp` command is used to copy files or directories, and the `-a` option is used to preserve the original file attributes (such as ownership and permissions). In this command, `/home/osmi4319/myFiles` is the source directory.

Purpose:

The purpose of `cp -a /home/osmi4319/myFiles` is to copy the entire contents of the directory `/home/osmi4319/myFiles`, including its subdirectories and files, to the current directory while preserving their attributes.

```
abc.txt file.mp4 mnp mnp.txt mystory osamadir
-sh-4.2$ cd ..
-sh-4.2$ cp -a /osmi4319/osamadir
cp: missing destination file operand after '/osmi4319/osamadir'
Try 'cp --help' for more information.
-sh-4.2$ cp -a mystory /osmi4319/osamadir
cp: cannot create regular file '/osmi4319/osamadir': No such file or directory
-sh-4.2$ cp -a /osmi4319/osamadir mystory
cp: cannot stat '/osmi4319/osamadir': No such file or directory
```

10) Command: cp /osmi4319/myFiles/abc.txt mnp.txt**Description:**

The cp command is used to copy files or directories. In this command, /osmi4319/myFiles/abc.txt is the source file, and mnp.txt is the destination file.

Purpose:

The purpose of cp /osmi4319/myFiles/abc.txt mnp.txt is to copy the file abc.txt from the directory /osmi4319/myFiles to the current directory and rename it as mnp.txt.

```
-sh-4.2$ pwd
/home/osmi4319
-sh-4.2$ cp /osmi4319/myFiles/abc.txt mnp.txt
cp: cannot stat '/osmi4319/myFiles/abc.txt': No such file or directory
-sh-4.2$ cp /myFiles/abc.txt mnp.txt
cp: cannot stat '/myFiles/abc.txt': No such file or directory
-sh-4.2$ cp myFiles/abc.txt mnp.txt
-sh-4.2$ cd myFiles
-sh-4.2$ ls
abc.txt mnp.txt
```

11) Command: /osmi4319/myFiles/abc.txt /osmi4319/myFiles/mnp.txt.

This command seems to be incomplete or incorrect. It appears to list two file paths without specifying any command to operate on them.

```
-sh-4.2$ pwd
/home/osmi4319
-sh-4.2$ cp /osmi4319/myFiles/abc.txt mnp.txt
cp: cannot stat '/osmi4319/myFiles/abc.txt': No such file or directory
-sh-4.2$ cp /myFiles/abc.txt mnp.txt
cp: cannot stat '/myFiles/abc.txt': No such file or directory
-sh-4.2$ cp myFiles/abc.txt mnp.txt
-sh-4.2$ cd myFiles
-sh-4.2$ ls
abc.txt mnp.txt
```

12) Command: mv /osmi4319/myFiles/***Description:**

The mv command is used to move or rename files or directories. In this command, /osmi4319/myFiles/* represents all files and directories within the directory /osmi4319/myFiles.

Purpose:

The purpose of mv /osmi4319/myFiles/* is to move all files and directories within /osmi4319/myFiles to the current directory.

```
-sh-4.2$ pwd
/home/osmi4319/myFiles
-sh-4.2$ cd ..
-sh-4.2$ mv /osmi4319/myFiles/*
mv: missing destination file operand after '/osmi4319/myFiles/*'
Try 'mv --help' for more information.
-sh-4.2$ mv myFiles/*
-sh-4.2$ ls
abc  data  mnp.txt  myFiles  mystory
-sh-4.2$ cd myFiles
-sh-4.2$ ls
mnp.txt
-sh-4.2$ cat mnp.txt
Osama
_
```

13) Command: cp *gif myfiles/abc/iuu (myfiles, abc and iuu are directories)

This command seems to be incorrect. It appears to attempt to copy files with the extension gif to a directory structure, but it lacks the -r option to specify recursive copying, and the destination directories myfiles, abc, and iuu are not properly formatted.

```
/home/osmi4319/myFiles/abc
-sh-4.2$ cd ..
-sh-4.2$ cd ..
-sh-4.2$ pwd
/home/osmi4319
-sh-4.2$ cp *gif myfiles/abc/iuu
cp: cannot stat '*gif': No such file or directory
```

14) Command: mv ~ new (new is directory)*Description:**

The mv command is used to move or rename files or directories. In this command, *~ represents all files ending with ~.

Purpose:

The purpose of `mv *~ new` is to move all files ending with `~` to the directory `new`.

```
-sh-4.2$ mv ~ new
mv: cannot move '/home/osmi4319' to a subdirectory of itself, 'new'
-sh-4.2$ mkdir new
-sh-4.2$ ls
mnp.txt  new
-sh-4.2$ mv ~ new
mv: cannot move '/home/osmi4319' to a subdirectory of itself, 'new/osmi4319'
-sh-4.2$ mv new
```

15) Command: `rm -rf new/*`**Description:**

The `rm` command is used to remove or delete files or directories, and the `-rf` options are used to force removal without prompting and to recursively remove directories and their contents. In this command, `new/*` represents all files and directories within the directory `new`.

Purpose:

The purpose of `rm -rf new/*` is to recursively remove all files and directories within the directory `new` without prompting for confirmation.

```
-sh-4.2$ ls
abc  abc.txt
-sh-4.2$ cd ..
-sh-4.2$ rm -rf new/*
-sh-4.2$ ls
mnp.txt  new
-sh-4.2$ cd new
-sh-4.2$ ls
```

16) Command: `rm -ri new/*`**Description:**

The `rm` command is used to remove or delete files or directories, and the `-ri` options are used to prompt for confirmation before each removal and to recursively remove directories and their contents. In this command, `new/*` represents all files and directories within the directory `new`.

Purpose:

The purpose of `rm -ri new/*` is to interactively prompt for confirmation before recursively removing all files and directories within the directory `new`. But we already deleted it so no files.

```
-sh-4.2$ cd ..  
-sh-4.2$ pwd  
/home/osmi4319/myFiles  
-sh-4.2$ ls  
mnp.txt  new [REDACTED]  
-sh-4.2$ rm -ri new/*  
rm: cannot remove 'new/*': No such file or directory  
.  .  .
```

17) Command: mkdir -p new/myFiles

Description:

The mkdir command is used to create directories, and the -p option is used to create parent directories as needed. In this command, new/myFiles is the directory structure to be created.

Purpose:

The purpose of mkdir -p new/myFiles is to create the directory new if it doesn't exist, and within it, create the directory myFiles, along with any necessary parent directories. This ensures the entire directory structure is created if it doesn't already exist.

```
-sh-4.2$ pwd  
/home/osmi4319/myFiles  
-sh-4.2$ ls  
mnp.txt  new [REDACTED]  
-sh-4.2$ mkdir -p new/myFiles  
-sh-4.2$ cd new  
-sh-4.2$ ls  
myFiles [REDACTED]  
.  .  .
```

~ END ~