1. DIRECTORY MANAGEMENT COMMANDS

Aim: To familiarize with the directory management commands in Linux OS.

1. Command: ***ls***
Usage : It is used to display the files in the current working directory.
Syntax : ls [Options]
[Options] :
-l - list the files in the long format
-a - list all entries, including the hidden files
-d - list the directories only
-t - lists in order of last modification time

2. Command: ***cd***
Usage : It is used to change from the current working directory to any other directory specified in the command.
Syntax : cd [DIRECTORY]

3. Command: ***pwd***
Usage : It is used to display the full path of the current working directory, starting from the root \.
Syntax : pwd

4. Command: ***mkdir***
Usage : It is used to create a new directory.
Syntax : mkdir <directory name>

5. Command : ***rmdir***
Usage : It is used to remove/delete a directory specified in the command. The directory will be removed only if it is empty.
Syntax : rmdir <directory name>

Result:
Familiarized with the directory management commands in Linux OS.

2. FILE MANAGEMENT COMMANDS

Aim: To familiarize with the file management commands in Linux OS.

1. Command : ***cat***
Usage : This command is used to display the contents of a small file on terminal.
Syntax : cat <option> <file name>
Option : -s –Warning about non existing file.
Example: $cat sample3.txt
Output : To display the content of sample3 text file.

2. Command : ***chmod***
Usage : It is used to change the access permissions of a file by the owner, group, and others.
Syntax: chmod [OPTION] [MODE] [FILE]
Example: chmod o+x Testing.java

3. Command: ***cp***
Usage : This command is used to create duplicate copies of ordinary files.
Syntax : cp [source file] [destination file]
Example: $cp testing.java testing1.java

4. Command: ***mv***
Usage : This command is used to move or rename ordinary files and directories.
Syntax : mv [source file] [destination file]
Example: $mv testing.java test.java

5. Command: ***rm***
Usage : It is used to remove the specified file in a directory.
Syntax : rm [source file]

6. Command: ***diff***
Usage : This command is used to find difference between two files.
Syntax : diff [options] <file_name_1> <file_name_2>
Options :
-w - Ignore white space when comparing lines.
-a - Treat all files as text and compare them line-by-line.
-b - Ignore changes in amount of white space.
Example: $ diff -w file1.txt file2.txt

7. Command: ***cmp***
Usage: This command is used to compare two files and find whether they are identical or not. The two files are compared byte by byte and the location of the first mismatch is printed on the screen. If two files are identical, then it does not print anything on the screen.
Syntax : cmp <file1> <file2>

Example: $ cmp sample1.txt sample2.txt

8. Command : ***cat <file name_1> <file name_2>***
Usage : This command, when supplied with more than one filename, will concatenate the files without any header information.
Syntax : cat <file name_1> <file name_2>
Example: $ cat sample3.txt sample4.txt

Result:
Familiarized with the file management commands in Linux OS.

3. GENERAL PURPOSE COMMANDS

Aim: To familiarize with the general purpose commands in Linux OS.

1. Command: ***wc***
Usage : This command is used to count lines, words and characters, depending on the options used.
Syntax : wc [options] [file name]
Options :
-l : Number of lines
-w : Number of words
-c : Number of characters

2. Command: ***lp***
Usage : This command is used to send a file to the printer.
Syntax : lp [option] filename

3. Command : ***date***
Usage : This command is used to display the current date.
Syntax : $ date

4. Command: ***who***
Usage : This command is used to list the users currently logged on to the system.
Syntax : $ who

5. Command : ***ln***
Usage : The ln command is used to create a link to a file (or) directory. It helps to provide soft link for desired files.
Syntax : ln [options] <existing file (or directory)> <new file(or directory)>
Options:
-n - Does not overwrite existing files.
-s - Used to create soft links

Result:
Familiarized with the general purpose commands in Linux OS.

4. DEVICE PATTERN

Aim: To execute device pattern using meta character to match each of the following situation:
i. All two character filenames.
ii. All filenames consisting of two lowercase letters.
iii. All filenames ending with c.
iv. All filenames beginning with c and ending with a digit.
v. All filenames beginning with p and having p at somewhere.

Command :
i) All two character filenames:

```
$ ls ??
```

ii) All filenames consisting of two lowercase letters.

```
$ ls *[a-z]??
```

iii) All filenames ending with c.

```
$ ls *c
```

iv) All filenames beginning with c and ending with a digit

```
$ ls c*[0-9]
```

v) All filenames beginning with p and having p at somewhere

```
$ ls p*?p*
```

Result: Successfully executed various device patterns using meta characters to match different situations.

5. SHELL SCRIPT TO DECREMENT NUMERICAL VALUE OF N

Aim: To write a shell script that accepts a numerical value N, and decrement and display the value of N till it reaches 0.

Program:

```
#!bin/bash
echo "ENTER AN INTEGER VALUE: "
read n
while [ $n -ge 0 ]
do
echo "$n"
let "n = n-1"
done
```

Result: Successfully executed a shell script that accepts a numerical value N, and decremented and displayed its value it reaches 0.

Output:

```
ENTER AN INTEGER VALUE:
6

6
5
4
3
2
1
0
```

6. SHELL SCRIPT TO DISPLAY THE MESSAGE ON LOG IN

Aim: Write a shell script to say Good morning/ Afternoon /Evening as you log in to the system.

Program:

```
#!/bin/bash
hour=$(date + "%H")
if [ $hour -ge 0 -a $hour -lt 12 ]
then
greet="GOOD MORNING, $USER"
elif [ $hour -ge 12 -a $hour -lt 18 ]
then
greet="GOOD AFTERNOON,$USER"
else
greet="GOOD EVENING,$USER"
fi
echo $greet
```

Result: Successfully executed the shell script to say Good morning/Afternoon/Evening as a user logs into the system.

7. SHELL SCRIPT TO IMPLEMENT BASIC CALCULATOR

Aim: Using shell script**,** develop a basic math calculator using case statement.

Program:

```
# Implementation of Calculator application
#!bin/bash
j=1
while [ $j -eq 1 ]
do
echo "Enter the First Operand;"
read f1
echo "Enter the second operand:"
read f2
echo "1-> Addition"
echo "2-> Subtraction"
echo "3-> Multiplication"
echo "4-> Division"
echo "Enter your choice"
read n
case "$n" in
1)
echo "Addition"
f3=$((f1+f2))
echo "The result is:$f3"
;;
2)
echo "Subtraction"
let "f4=$f1 - $f2"
echo "The result is:$f4"
;;
3)
echo "Multiplication"
let "f5=$f1 * $f2"
echo "The result is:$f5"
;;
4)
echo "Division"
let "f6=$f1 / $f2"
echo "The result is:$f6"
;;
esac
echo "Do you want to continue ?(press:1 otherwise press any key
to quit)"
read j
```

```
done
```

Result: Shell script to develop a calculator application was executed successfully.

8. SHELL SCRIPT TO FIND THE LENGTH OF GIVEN STRING

Aim: To find the length of given string using shell script.

Program:

```
#!bin/bash
echo "Enter the string "
read str
len=${#str}
echo "$len"
```

Result: Shell script to find the length of a string was executed successfully.