**MID-1 LP LAB RECORD**

**1.Basic Linux Commands File handling utilities, Security by file permissions, Process utilities, Disk utilities, sed, awk, grep**.

File handling Commands: : cat ,touch ,rm, mv, vi, chmod, cp, mkdir, cd, ls , find ,chown ,chgrp

Process utilities: ps , who, who am i , top

Disk Utilities: df , du

Filters: head, tail, cut, paste, sort, tr, wc, cmp

Grep command

Sed command

Awk

**Basic Linux Commands Guide**

1. File Handling Commands

- cat: Concatenate and display file contents.

Usage: `cat filename`

- touch: Create an empty file or update the modification timestamp.

Usage: `touch filename`

- rm: Remove files or directories.

Usage: `rm filename` or `rm -r directory`

- mv: Move or rename files.

Usage: `mv oldname newname` or `mv filename directory`

- vi: Open the Vi editor to edit text files.

Usage: `vi filename`

- chmod: Change file permissions.

Usage: `chmod 755 filename`

- cp: Copy files or directories.

Usage: `cp source destination`

- mkdir: Create a new directory.

Usage: `mkdir directory\_name`

- cd: Change directory.

Usage: `cd directory\_name`

- ls: List files and directories.

Usage: `ls -l` (long format)

- find: Search for files and directories.

Usage: `find /path -name filename`

- chown: Change file owner.

Usage: `chown user:group filename`

- chgrp: Change file group ownership.

Usage: `chgrp group filename`

2. Process Utilities

- ps: Display current active processes.

Usage: `ps aux`

- who: Show who is logged in.

Usage: `who`

- who am i: Show details of the current session.

Usage: `who am i`

- top: Real-time display of system processes.

Usage: `top`

3. Disk Utilities

- df: Show disk space usage.

Usage: `df -h`

- du: Estimate file or directory space usage.

Usage: `du -sh directory\_name`

4. Filters

- head: Display the first lines of a file.

Usage: `head filename`

- tail: Display the last lines of a file.

Usage: `tail filename`

- cut: Extract sections of a file.

Usage: `cut -d':' -f1 filename`

- paste: Merge lines of files.

Usage: `paste file1 file2`

- sort: Sort lines in text files.

Usage: `sort filename`

- tr: Translate characters.

Usage: `tr 'a-z' 'A-Z' < filename`

- wc: Count words, lines, or bytes.

Usage: `wc filename`

- cmp: Compare two files.

Usage: `cmp file1 file2`

5. Grep Command

- grep: Search for a specific pattern in files.

Usage: `grep 'pattern' filename`

6. Sed Command

- sed: Stream editor for filtering and transforming text.

Usage: `sed 's/old/new/g' filename`

7. Awk Command

- awk: Pattern scanning and processing language.

Usage: `awk '{print $1}' filename`

**2. Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers.**

Program

echo "enter file name"

read f

echo 'enter starting position'

read st

echo 'enter ending position'

read end

echo 'The lines between' $st 'and' $end 'from' $f

if [ $st -lt $end ]

then

n1=`expr $st + 1`

n2=`expr $end - 1`

sed -n "$n1,$n2 p" $f

elif [ $st -gt $end ]

then

n3=`expr $st - 1`

n4=`expr $end + 1`

sed -n "$n4,$n3 p" $f

fi

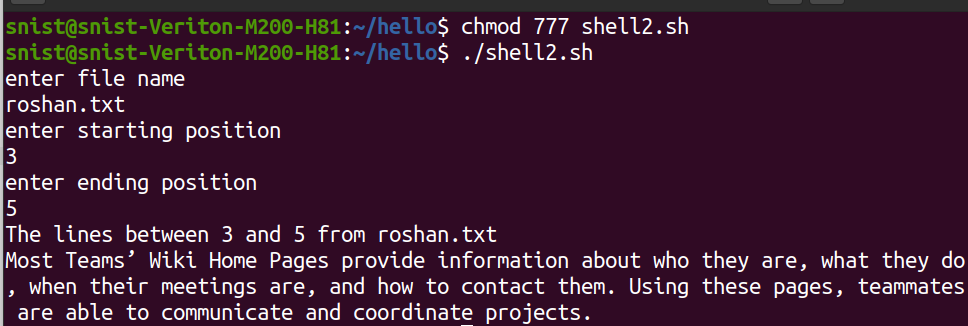
**OUTPUT:**

1.save the file with .sh extension.

2.Using chmod command give the permission to execute the file .

3. type to see the output

sh filename.sh or ./filename.sh



**3.Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.**

echo 'enter a word to be deleted'

read word

echo 'enter file name'

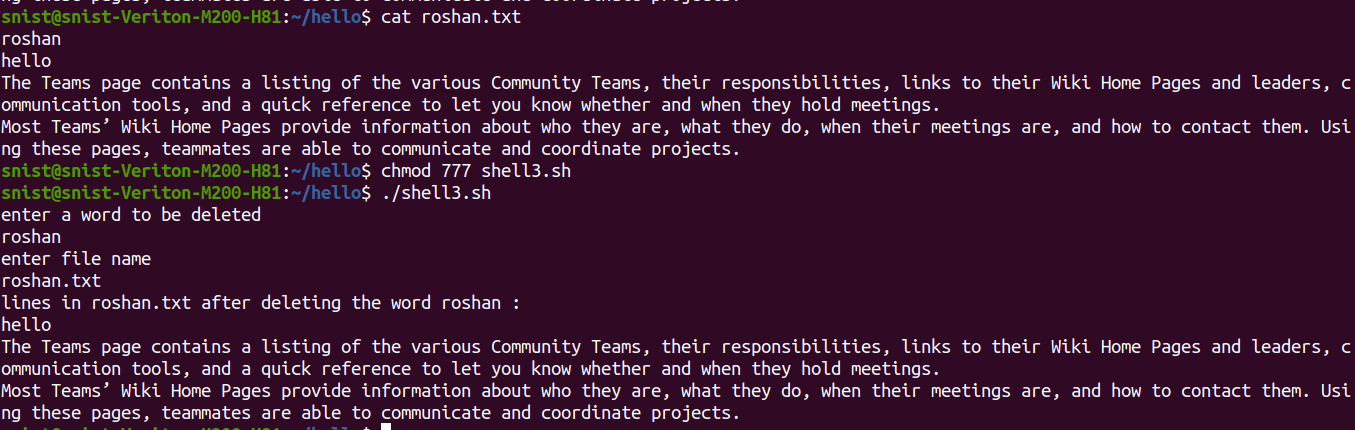
read fname

echo 'lines in' $fname 'after deleting the word' $word ':'

sed "/$word/d" $fname

**(or)**

if [ $# -eq 0 ]  
then  
echo "Please enter one or more filenames as argument"  
exit  
fi  
echo "Enter the word to be searched in files"  
read word  
for file in $\*  
do  
sed "/$word/d" $file | tee tmp  
mv tmp $file  
done



**4.Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.**

for i in \*

do

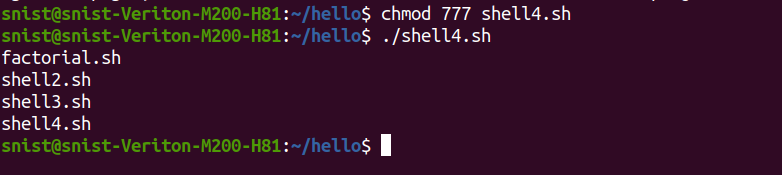
if [ -r $i -a -w $i -a -x $i ]

then

echo $i

fi

done



**6.Write a shell script that receives any number of file names as arguments checks if every argument supplied is a file or directory and reports accordingly. Whenever the argument is a file, the number of lines on it is also reported.**

for fname in $\*

do

if [ -f $fname ]

then

echo $fname 'is a file'

echo 'no.of lines in' $fname ':'

wc -l $fname

elif [ -d $fname ]

then

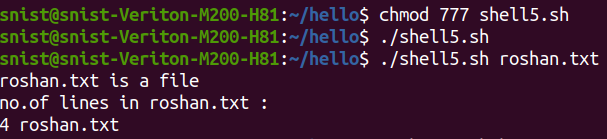
echo $fname 'is a directory'

else

echo 'Does not exist'

fi

done



7.**Write the following Shell scripts** :

**a.Write a shell script that accepts a file names as its arguments, counts and reports the occurrence of each word that is present in the file argument file in other argument files.**

**shell.sh**

if [ $# -lt 2 ]; then

echo "Usage: $0 source\_file target\_file1 [target\_file2 ...]"

exit 1

fi

source\_file=$1

shift

if [ ! -f "$source\_file" ]; then

echo "Source file '$source\_file' does not exist."

exit 1

fi

for word in $(tr -s '[:space:]' '\n' < "$source\_file"); do

total\_count=0

for target\_file in "$@"; do

if [ ! -f "$target\_file" ]; then

echo "Target file '$target\_file' does not exist."

continue

fi

count=$(grep -oiw "$word" "$target\_file" | wc -l)

total\_count=$((total\_count + count))

done

if [ $total\_count -gt 0 ]; then

echo "Word '$word' found $total\_count times in the target files."

else

echo "Word '$word' not found in any target files."

fi

done

**file1.txt**

apple

banana

orange

**file2.txt**

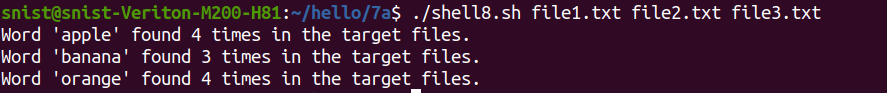
apple orange apple

banana orange banana

**file3.txt**

apple apple banana

orange orange



**b.To list all of the directory files in a directory.**

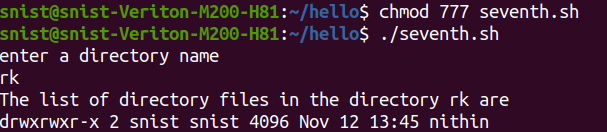
echo 'enter a directory name'

read dname

echo 'The list of directory files in the directory' $dname 'are'

cd $dname

ls -l | grep '^d'

****

**c.Write a shell script to find factorial of a given number.**

echo "enter n"

read n

i=1

fact=1

while [ $i -le $n ]

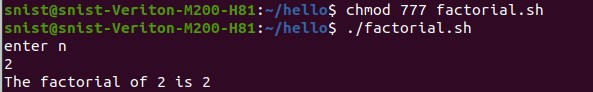
do

fact=`expr $fact \\* $i`

i=`expr $i + 1`

done

echo 'The factorial of' $n 'is' $fact



**8.a write an awk script to count number of lines in a file that does not contain vowels**

**file1.txt**

hi

how r u

this

is

hello

bajj

dfdf

hhhh

rrrr

ttt

**word.awk**

!/[aeiouAEIOU]/ {

count++

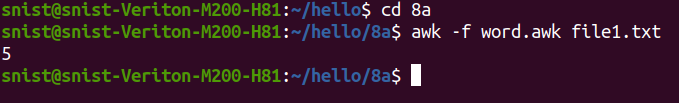
}

END {

print count

}

**output:**

****

**8.b write an awk script to find the no of characters ,words and lines in a file**

**file2.txt**

hi how r u

this is hw

**word2.awk**

{

char\_count += length($0)

word\_count += NF

line\_count++

}

END {

print "Lines:", line\_count

print "Words:", word\_count

print "Characters:", char\_count

}

