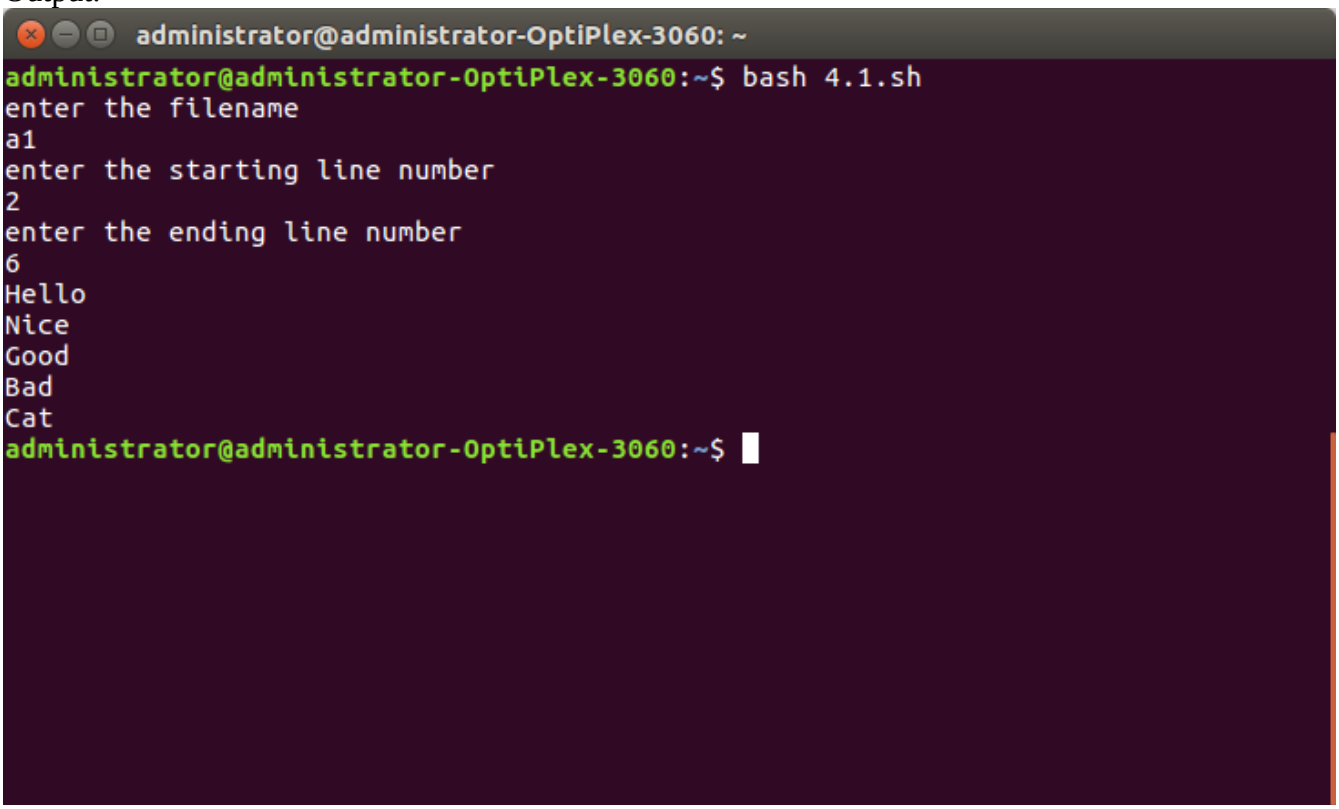
Assignment - 4 (Operating System)

U20CS110
Krishna Pandey

1. 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.
Ans.
Code.

```
echo "enter the filename"
read fname
echo "enter the starting line number"
read s
echo "enter the ending line number"
read n
sed -n $s,$n\p $fname | cat > newline
cat newline
```

Output:

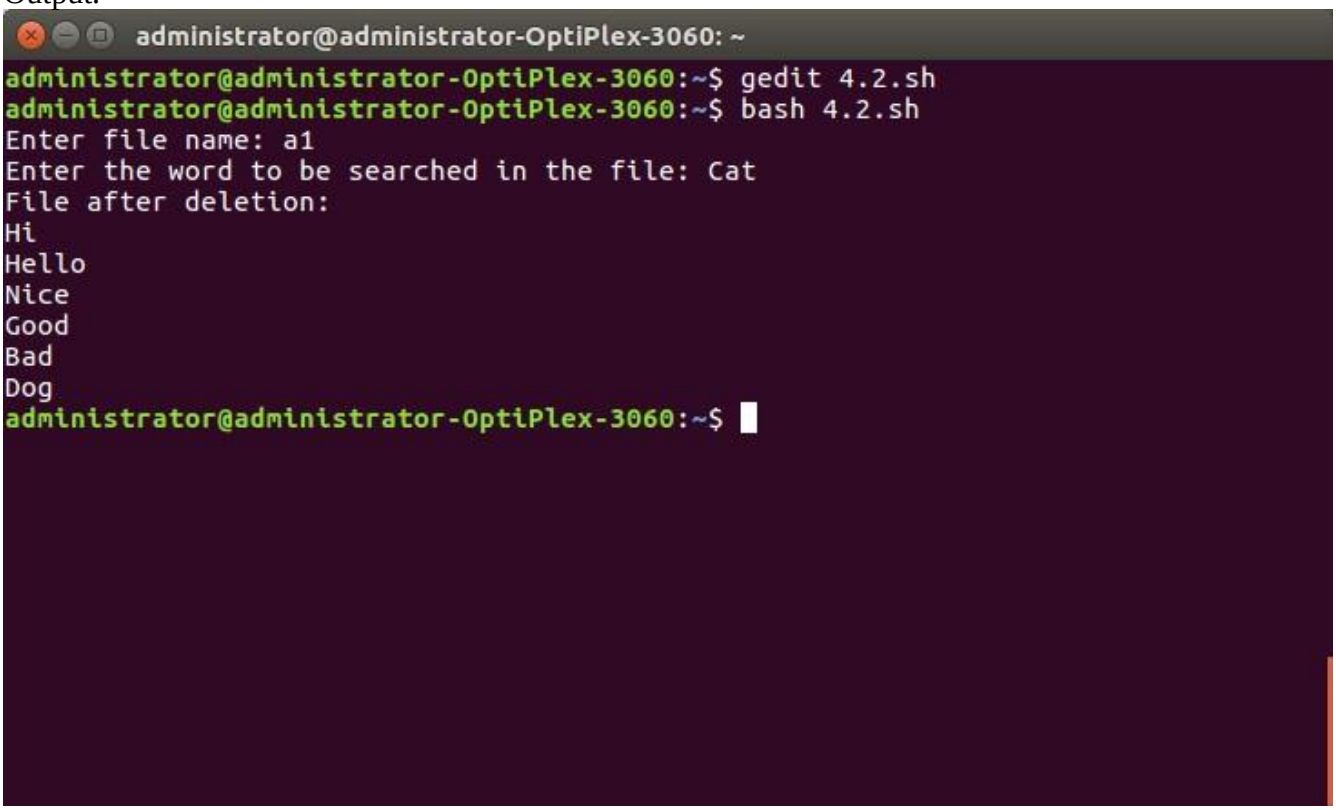


2. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
Ans.
Code.

```
echo -n "Enter file name: "
read FILE
```

```
echo -n "Enter the word to be searched in the file: "
read word
echo "File after deletion: "
for file in FILE
do
sed "/$word/d" $FILE | tee tmp
mv tmp $file
done
```

Output.



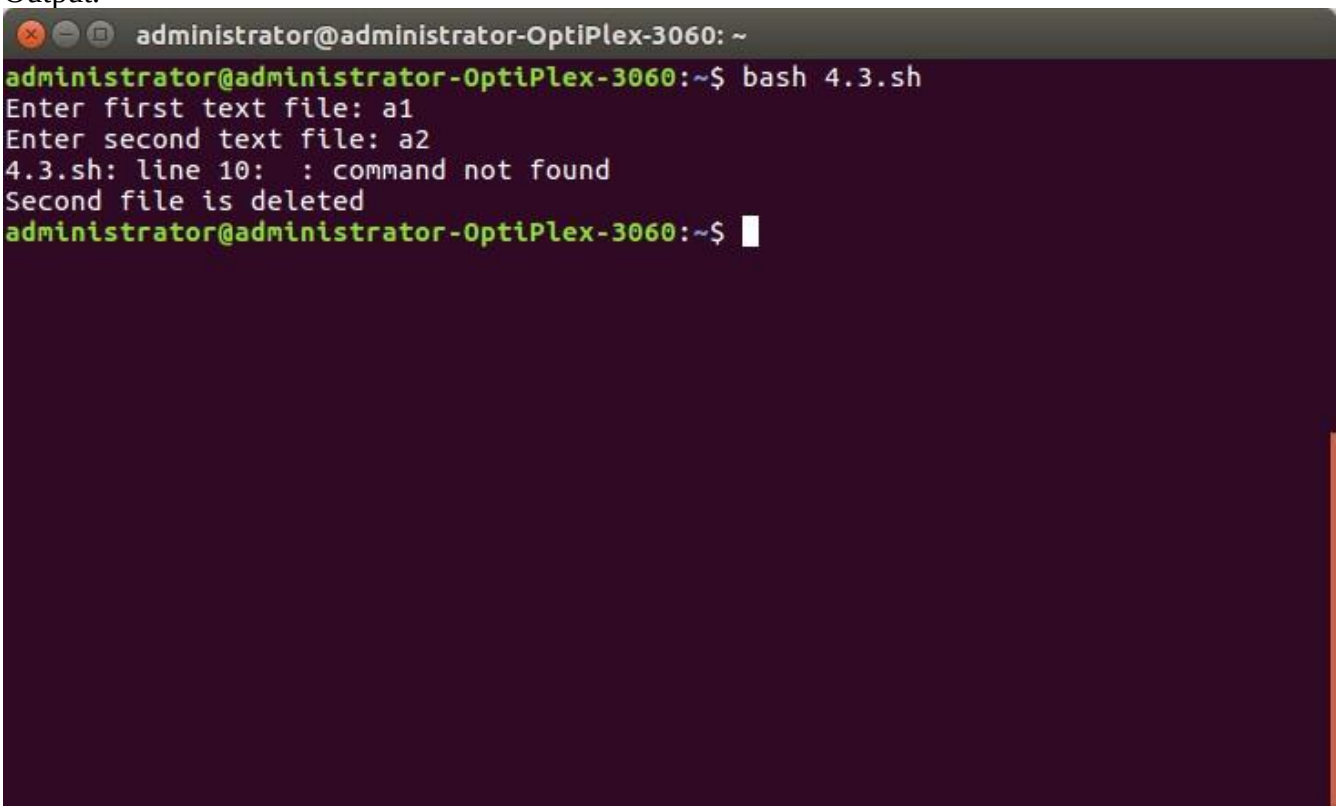
3. Write a shell script which receives two file names as argu
ments. It should check whether the two file contents are the same or not. If they are the same then the second file should be deleted.

Ans.

Code.

```
echo -n "Enter first text file: "
read text1
echo -n "Enter second text file: "
read text2
if cmp -s -- "$text1" "$text2";
then
        rm -i -- "$text2"
echo "Second file is deleted"
fi
```

Output.



4. Write a shell script which takes filename as argument and checks whether file is regular file,directory,block special file, character special file, named pipe, symbolic link, socket, device file etc.

Ans.

Code.

file a1

Output.

```
administrator@administrator-OptiPlex-3060: ~
administrator@administrator-OptiPlex-3060:~$ file a1
a1: ASCII text
administrator@administrator-OptiPlex-3060:~$
```

5. Write a shell script which will take file name as argument and check whether the file name is a dir or not and then proceed further only if it is a dir, else give usage message. The script should then print in the tabular format, name of each sub-dir (within the argument dir) and a count of the number of top level files in that sub-dir. Modify the program to work with multiple numbers of arguments, too.

Ans.

Code.

Output.

```
administrator@administrator-OptiPlex-3060: ~
administrator@administrator-OptiPlex-3060:~$ bash 4.5.sh
Enter first text file: tanisha.txt
The given argument does not exist on the file system.
administrator@administrator-OptiPlex-3060:~$
```

6. Write a script that will search for a specific word in all the files in the current directory and then prompt with the file name in which word is found.

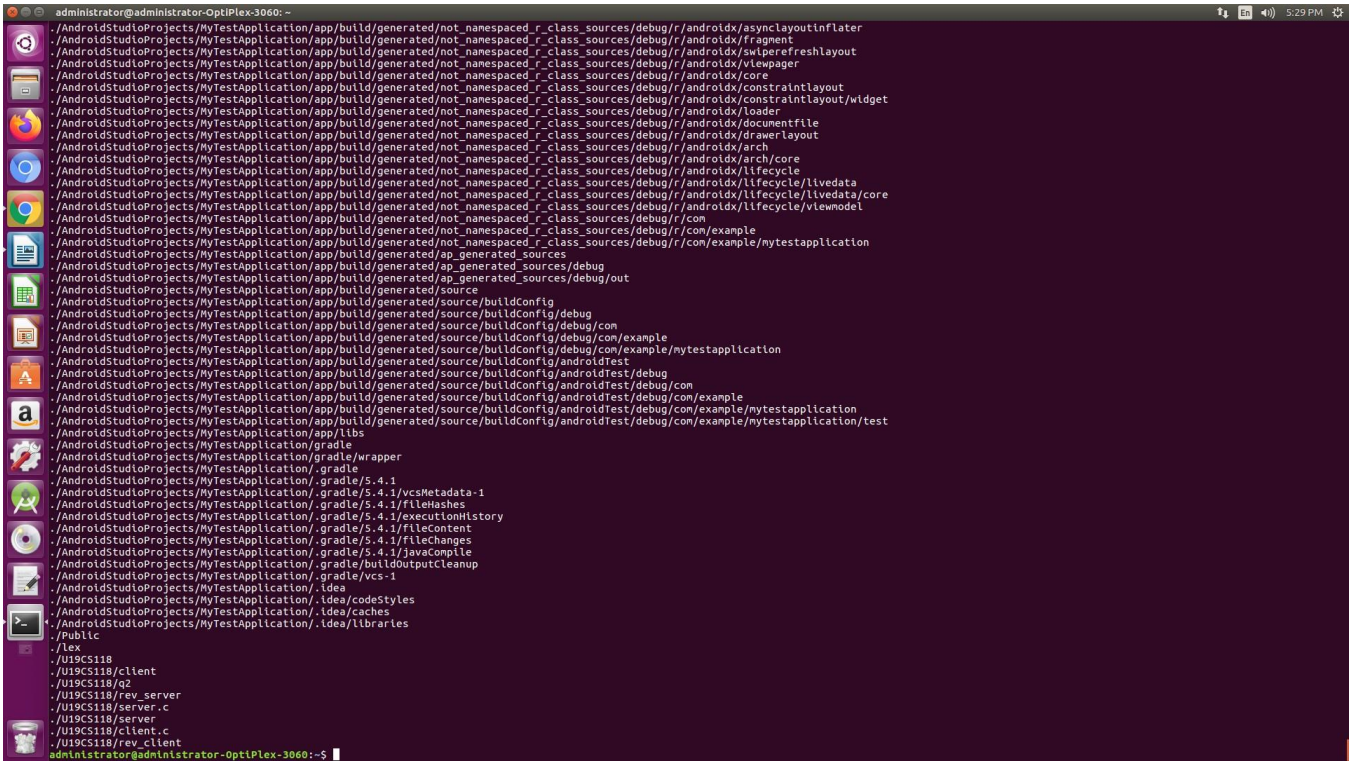
Ans.
Code.
grep -r -l "Krishna Pandey"

output:

7. Write a script to print only the number of executable files in each sub-dir of the argument directory specified.
Ans.
Code.

find . -executable

Output.



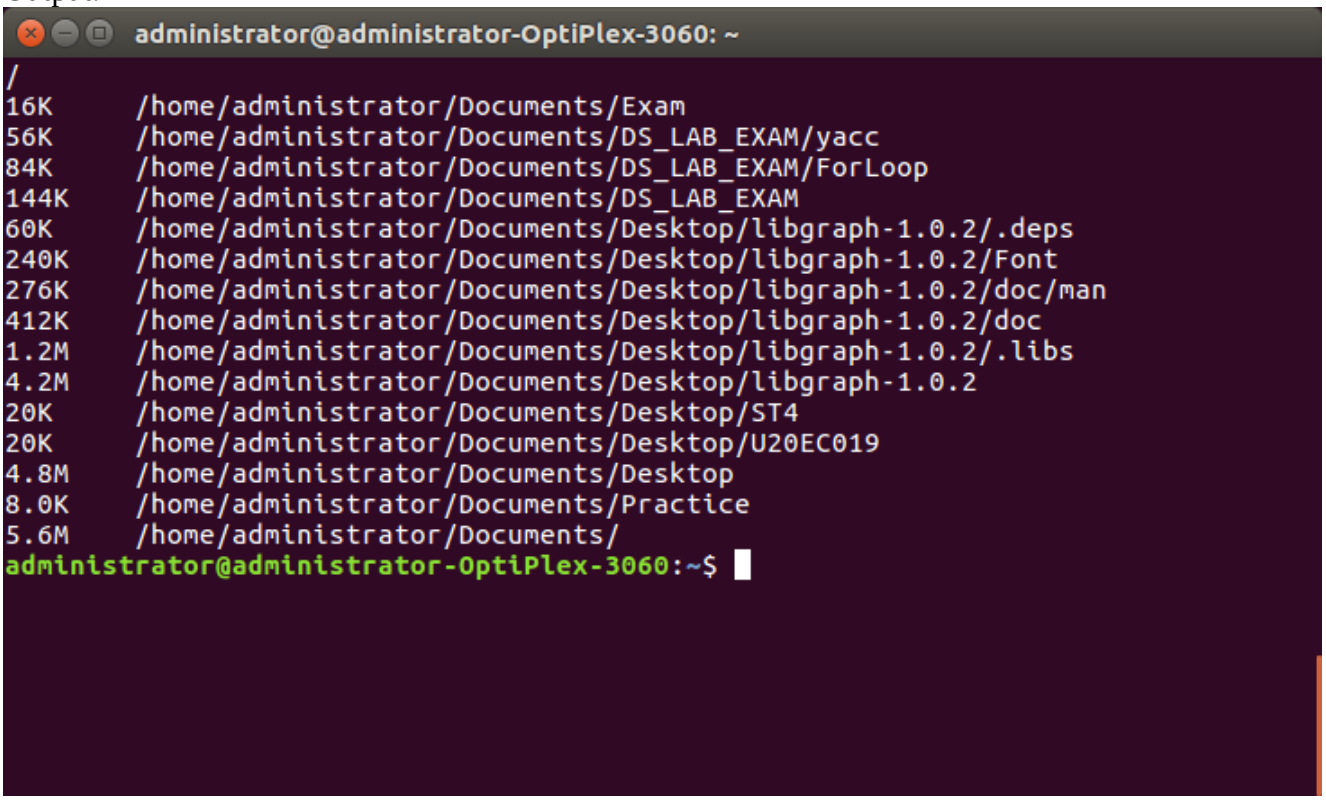
8. Write a non-interactive script that takes in any no. of directory name as argument and calculates total no. of blocks of disk space occupied by the ordinary files in all the directories.

Ans.

Code.

du -h /home/administrator/Documents/

Output.



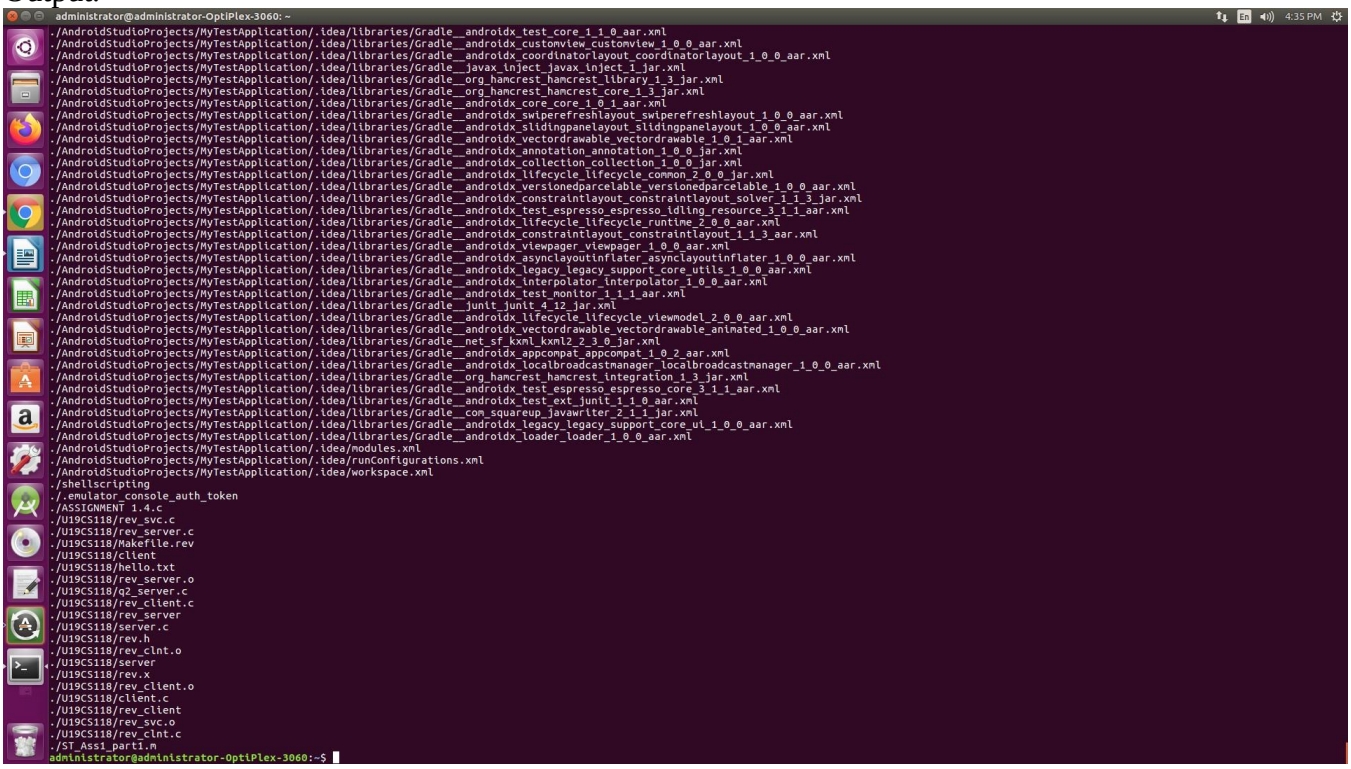
9. Write a shell script file named exercise2.sh that makes a list of files in your home directory that were changed less than 24 hours ago, but leave out directories.

Ans.

Code.

```
find -type f -mtime +1
```

Output.



10. Write the script that renames files based on the file extension. Next,It should ask the user what prefix to prepend to the file name(s). By default, the prefix should be the current date in YYYY-MM-DD format. If the user simply press enter,the current date will be used. Otherwise,whatever the user entered will be used as the prefix. Next,it should display the original file name and new name of the file. Finally,it should rename the file.

Ans.

Code.

```
read -p "Please enter a file extension: " EXTENSION
if [ ${#EXTENSION} -eq 0 ]
then
    echo "No file extension provided. Aborting."
    exit 1
fi
PREFIX=$(date +%Y%m%d)
read -p "Please enter a file prefix: (Press ENTER for ${PREFIX}). " NEW_PREFIX
if [ ${#NEW_PREFIX} -ne 0 ]
then
    PREFIX="${NEW_PREFIX}"
fi
shopt -s nullglob
```

```
for FILENAME in *.${EXTENSION}
do
   NEW_FILENAME="${PREFIX}${FILENAME}"
   echo "Renaming ${FILENAME} to ${NEW_FILENAME}."
   mv "${FILENAME}" "${NEW_FILENAME}"
done
        shopt -u nullglob
```

Output.