Week 2 – Assignments

1. Shell Script the following Perform Operations on a File (A/P)

a. Display the contents of the file

b. Counting characters, words & lines in the file

2. Differentiate the Batch File and Shell Scripting.

3. Write a Shell Script to accept a number and find Even or ODD

4. Write a Shell Script to accept a year and find Leap Year or Not

5. Write a Shell Script to check a given number is a prime number or Not

6. Write a Shell Script to find Factorial of a given number

7. Write a Shell Script to find Greatest of given Three numbers

8. Write a Shell Script to emulate the UNIX ls-l command. (A/P)

9. Write a Shell Script to accept numbers and print sorted numbers (A/P)

10. Write a Shell Script to accept State name and print the Capital using CASE(A/P)

11. Write a Shell Script for Arithmetic Calculator using CASE(A/P)

12. Write a Shell Script to Illustrate Logical Operators using expression evaluator expr (A/P)

13. Write a Shell Script to accept a string and display the length of the string (A/P)

14. Write a Shell Script to Illustrate the Functions (A/P)

15. Explain Unix architecture in detail with neat diagram

16. What is Shell Script in Operating System and How does it work?

Shell Script to Perform Operations on a File

Most of the time, we use shell scripting to interact with the files. Shell scripting offers some operators as well as some commands to check and perform different properties and functionalities associated with the file.

For our convenience, we create a file named 'geeks.txt' and another .sh file (or simply run on the command line) to execute different functions or operations on that file. Operations may be reading the contents of the file or testing the file type. These are being discussed below with proper examples:

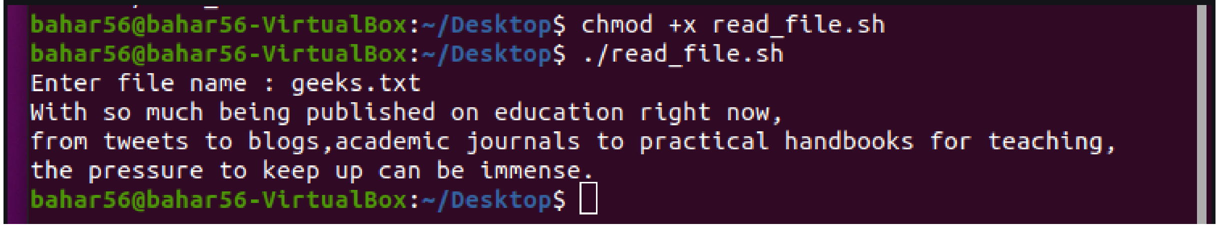
File Reading Functionalities

File reading is an interesting task in a programmer's life. Shell scripting offers some functionalities for reading the file, reversing the contents, counting words, lines, etc.

- **Reading line by line:** First, we take input using the read command then run the while loop which runs line after line.

Script:

```
#!/bin/bash

read -p "Enter file name : " filename

while read line

do

echo $line

done < $filename
```



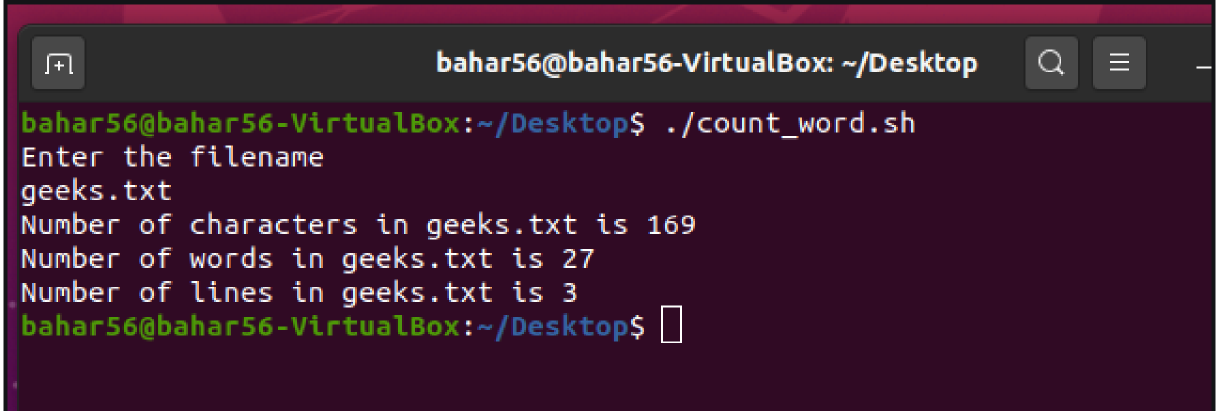
```
bahar56@bahar56-VirtualBox:~/Desktop$ chmod +x read_file.sh
bahar56@bahar56-VirtualBox:~/Desktop$ ./read_file.sh
Enter file name : geeks.txt
With so much being published on education right now,
from tweets to blogs,academic journals to practical handbooks for teaching,
the pressure to keep up can be immense.
bahar56@bahar56-VirtualBox:~/Desktop$
```

- **Counting characters, words & lines in the file:** We take three variables, one for counting characters, words, and lines respectively. We use 'wc' command, which stands for **word count** and counts the number of characters and words as well. For counting lines, we pass 'grep ' which keeps count of the lines that match a pattern. Then we print out each variable.

Script:

```
#! /bin/bash
```

```
echo Enter the filename
read file
c=`cat $file | wc -c`
w=`cat $file | wc -w`
l=`grep -c "." $file`
echo Number of characters in $file is $c
echo Number of words in $file is $w
echo Number of lines in $file is $l
```

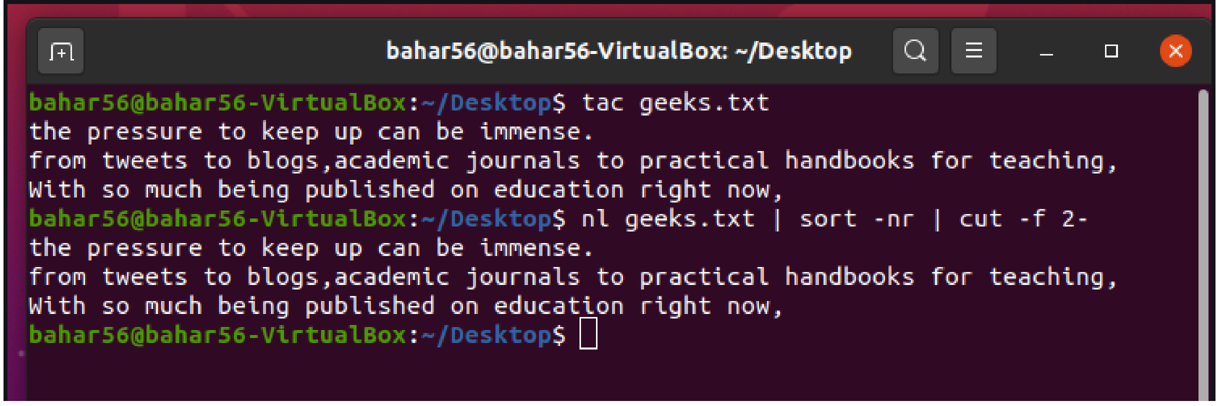


```
bahar56@bahar56-VirtualBox:~/Desktop$ ./count_word.sh
Enter the filename
geeks.txt
Number of characters in geeks.txt is 169
Number of words in geeks.txt is 27
Number of lines in geeks.txt is 3
bahar56@bahar56-VirtualBox:~/Desktop$
```

- **Display file contents in reverse:** To print the contents of any file in reverse, we use tac or nl, sort, cut commands. Tac is simply the reverse of a cat and simply prints the file in reverse order. Whereas nl commands numbers, the file contents sort the numbered file in the reverse order, and the cut command removes the number and prints the file contents.

Script:

```
$ nl geeks.txt | sort -nr | cut -f 2-
```

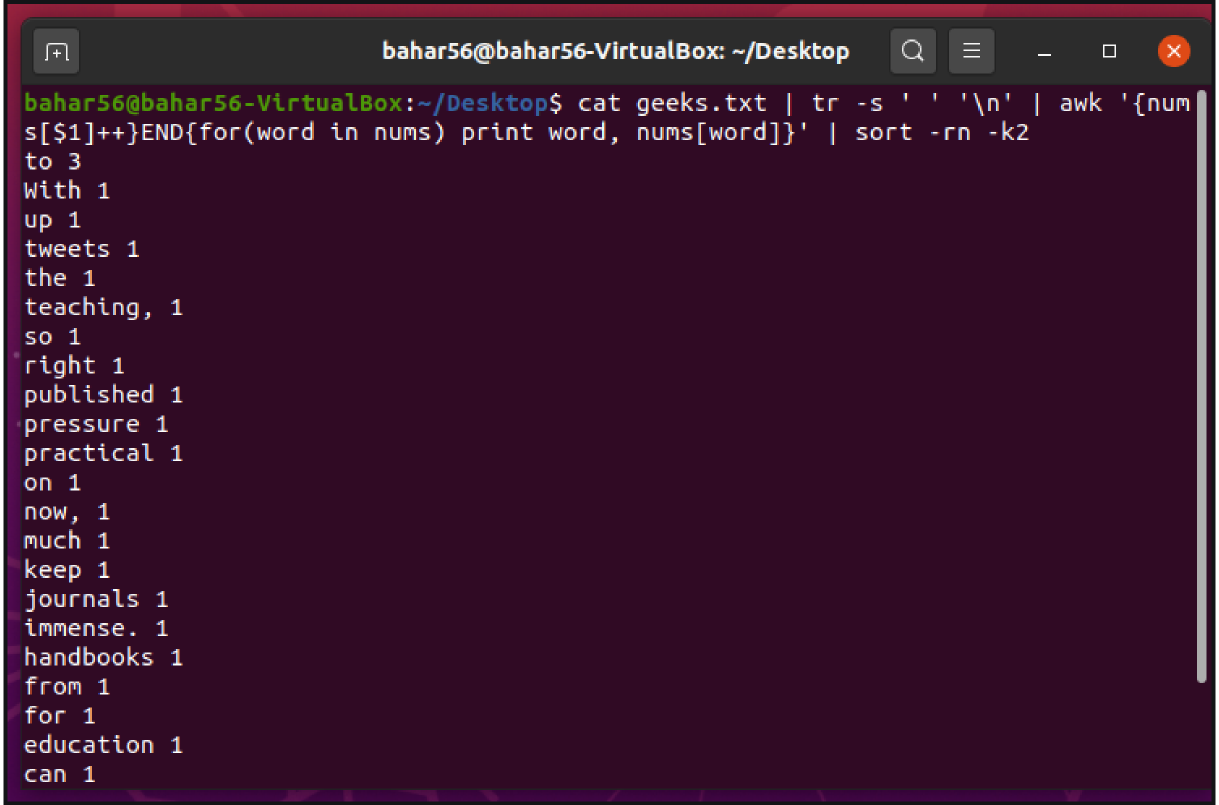


```
bahar56@bahar56-VirtualBox:~/Desktop$ tac geeks.txt
the pressure to keep up can be immense.
from tweets to blogs,academic journals to practical handbooks for teaching,
With so much being published on education right now,
bahar56@bahar56-VirtualBox:~/Desktop$ nl geeks.txt | sort -nr | cut -f 2-
the pressure to keep up can be immense.
from tweets to blogs,academic journals to practical handbooks for teaching,
With so much being published on education right now,
bahar56@bahar56-VirtualBox:~/Desktop$
```

- **Frequency of a particular word in the file:** To count the frequency of each word in a file, we use certain commands. These are xargs which applies print to every line in the output, the sort which sorts, current buffer piped to it, uniq -c displays the counts of each line in the buffer and, lastly awk, prints the 2nd column and then the 1st column based on the problem requirement.

Script:

cat geeks.txt | xargs printf "%s\n" | sort | uniq -c | sort -nr | awk '{print $2,$1}'



```
bahar56@bahar56-VirtualBox:~/Desktop$ cat geeks.txt | tr -s ' ' '\n' | awk '{num
s[$1]++}END{for(word in nums) print word, nums[word]}' | sort -rn -k2
to 3
With 1
up 1
tweets 1
the 1
teaching, 1
so 1
right 1
published 1
pressure 1
practical 1
on 1
now, 1
much 1
keep 1
journals 1
immense. 1
handbooks 1
from 1
for 1
education 1
can 1
```

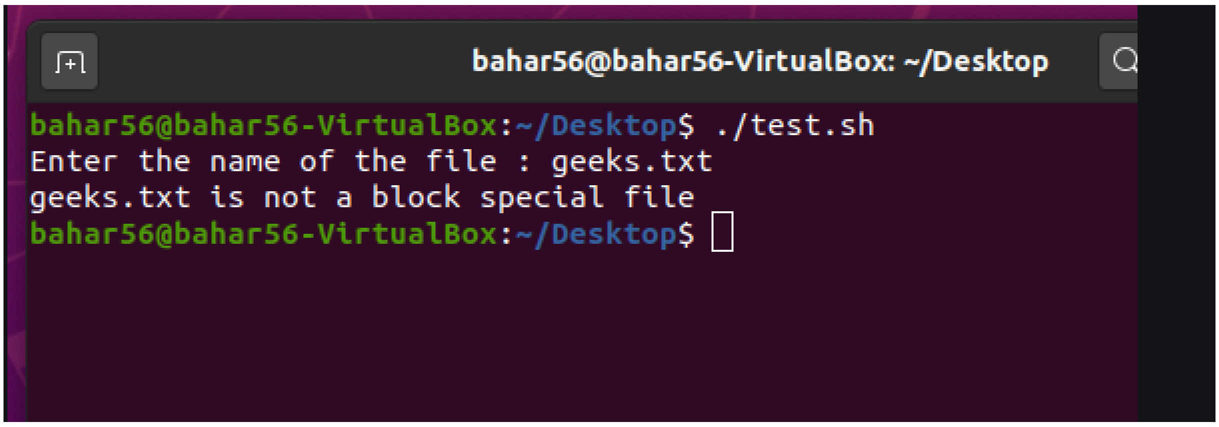
File Test Operators

- **-b file:** This operator checks if the file is a block special file or not and returns true or false subsequently. It is [-b $file] syntactically.

Script:

```
#! /bin/bash

echo -e "Enter the name of the file : \c"

read file_name


if [ -b $file_name ]

then

   echo "$file_name is a block special file"

else

   echo "$file_name is not a block special file"

fi
```

Output:



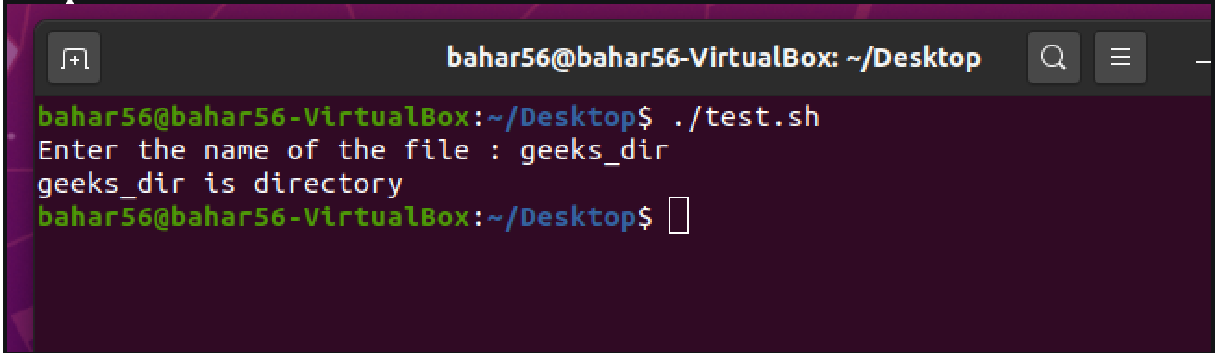
```
bahar56@bahar56-VirtualBox:~/Desktop$ ./test.sh
Enter the name of the file : geeks.txt
geeks.txt is not a block special file
bahar56@bahar56-VirtualBox:~/Desktop$
```

- **-d file:** The operator looks over if the file is present as a directory. If Yes, it returns true else false. It is [-d $file] syntactically.

Script:

```
#! /bin/bash

echo -e "Enter the name of the file : \c"

read file_name


if [ -d $file_name ]

then

   echo "$file_name is a directory"

else

   echo "$file_name is not a directory"

fi
```

Output:



```
bahar56@bahar56-VirtualBox:~/Desktop$ ./test.sh
Enter the name of the file : geeks_dir
geeks_dir is directory
bahar56@bahar56-VirtualBox:~/Desktop$
```

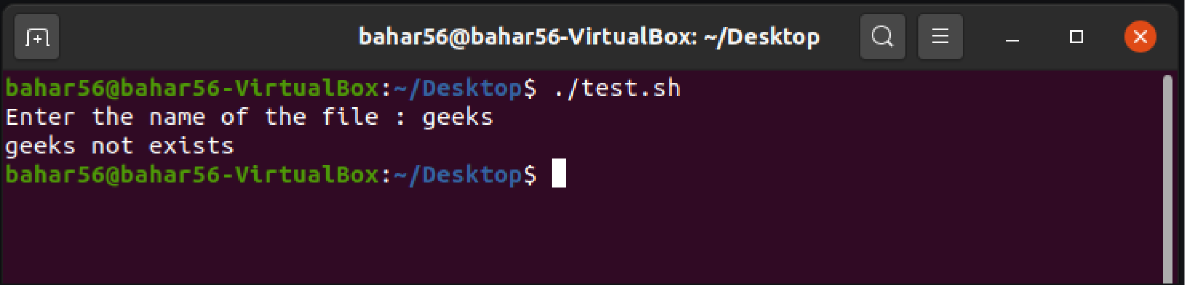
- **-e file:** The operator inspects if the file exists or not. Even if a directory is passed, it returns true if the directory exists. It is [-e $file] syntactically.

Script:

```
#! /bin/bash

echo -e "Enter the name of the file : \c"

read file_name
```

```
if [ -e $file_name ]
then
   echo "$file_name exist"
else
   echo "$file_name not exist"
fi
```

Output:



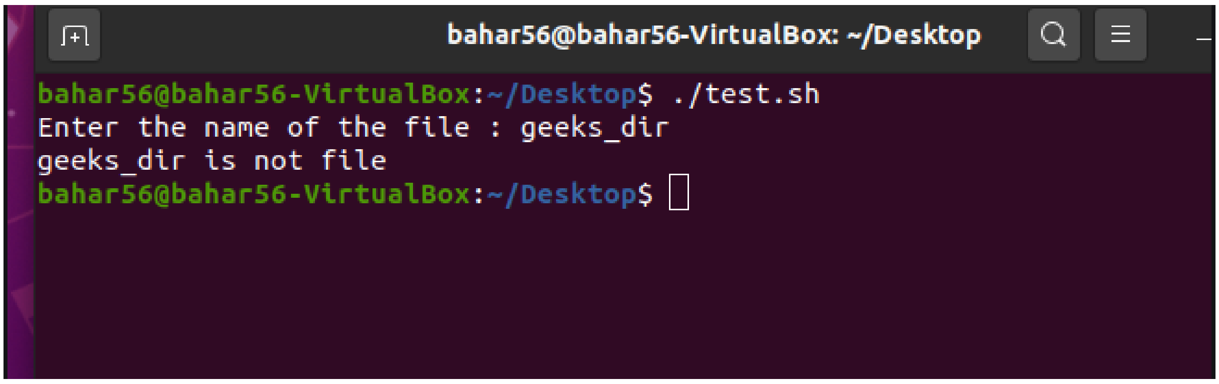
```
bahar56@bahar56-VirtualBox:~/Desktop$ ./test.sh
Enter the name of the file : geeks
geeks not exists
bahar56@bahar56-VirtualBox:~/Desktop$
```

- **-f file:** If the file is an ordinary file or special file, then it returns true else false. It is [-f $file] syntactically.

Script:

```
#! /bin/bash
echo -e "Enter the name of the file : \c"
read file_name

if [ -f $file_name ]
then
   echo "$file_name is file"
else
   echo "$file_name is not file"
fi
```

Output:



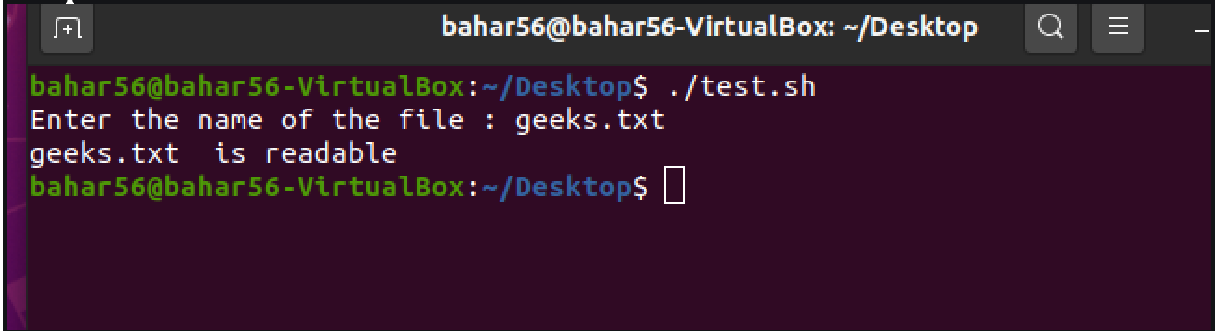
```
bahar56@bahar56-VirtualBox:~/Desktop$ ./test.sh
Enter the name of the file : geeks_dir
geeks_dir is not file
bahar56@bahar56-VirtualBox:~/Desktop$
```

- **-r file:** This checks if the file is readable. If found yes, then return true else false. It is [-r $file] syntactically.

Script:

```
#! /bin/bash

echo -e "Enter the name of the file : \c"

read file_name


if [ -r $file_name ]

then

   echo "$file_name is readable"

else

   echo "$file_name is not readable"

fi
```

Output:



```
bahar56@bahar56-VirtualBox:~/Desktop$ ./test.sh
Enter the name of the file : geeks.txt
geeks.txt  is readable
bahar56@bahar56-VirtualBox:~/Desktop$
```

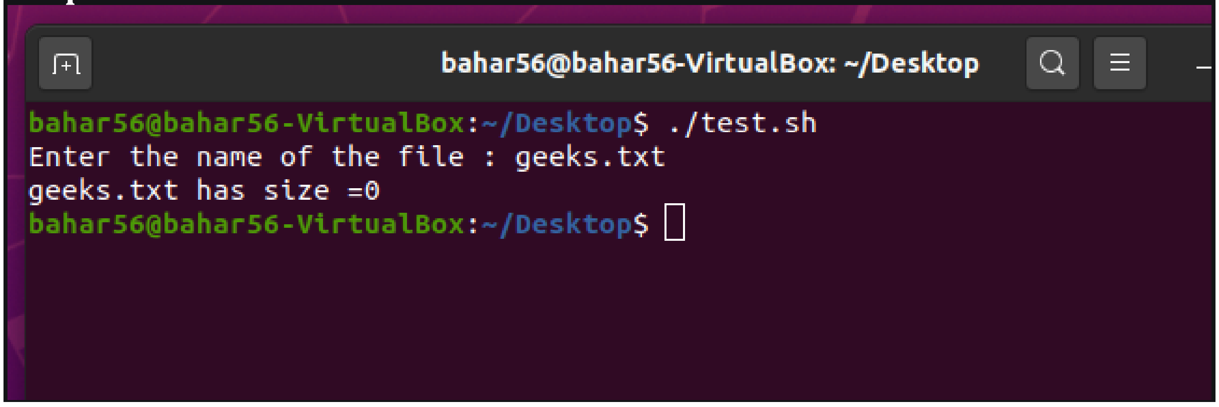
- **-s file:** This operator checks if the file has a size greater than zero or not, which returns true or false subsequently. It is [-s $file] syntactically.

Script:

```
 #! /bin/bash

echo -e "Enter the name of the file : \c"

read file_name
```

```
if [ -s $file_name ]

then

   echo "$file_name has size>0"

else

   echo "$file_name has size= 0"

fi
```

Output:



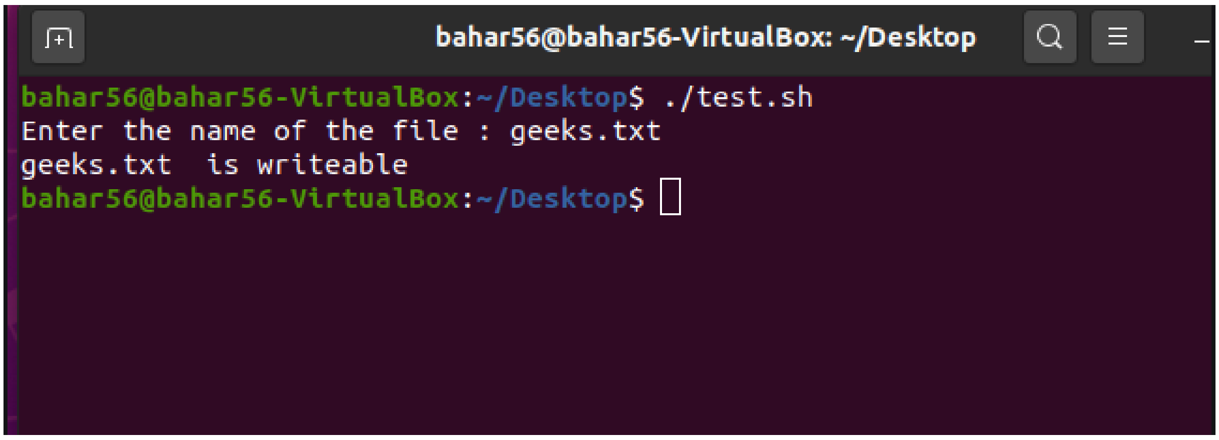
```
bahar56@bahar56-VirtualBox:~/Desktop$ ./test.sh
Enter the name of the file : geeks.txt
geeks.txt has size =0
bahar56@bahar56-VirtualBox:~/Desktop$
```

- **-w file:** If wringing is allowed over the file, then the operator returns true, and if not then false. It is [-w $file] syntactically.

Script:

```
#! /bin/bash

echo -e "Enter the name of the file : \c"

read file_name


if [ -w $file_name ]

then

   echo "$file_name is writable"

else

   echo "$file_name is not writable"

fi
```

Output:



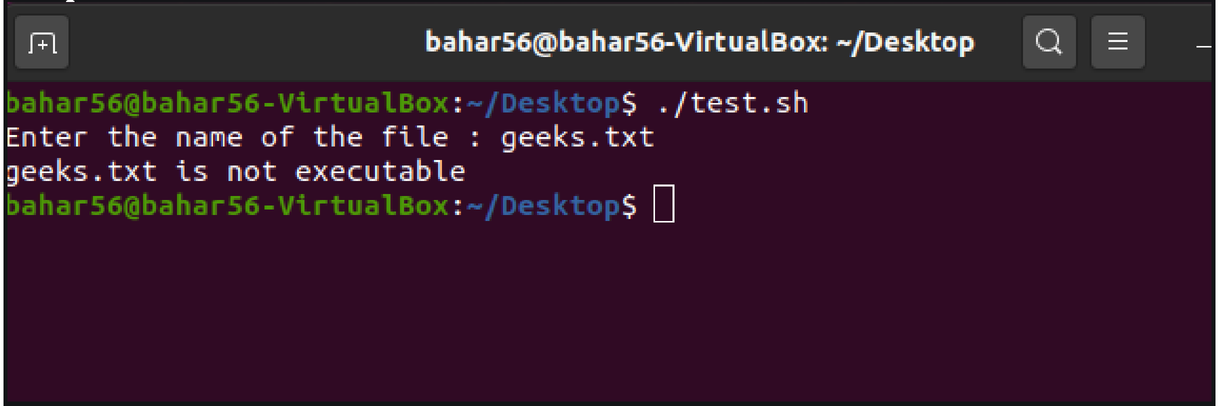
```
bahar56@bahar56-VirtualBox:~/Desktop$ ./test.sh
Enter the name of the file : geeks.txt
geeks.txt  is writeable
bahar56@bahar56-VirtualBox:~/Desktop$ 
```

- **-x file:** The operator looks over if the file is executable or not, and returns true and false subsequently. It is [-x $file] syntactically.

Script:

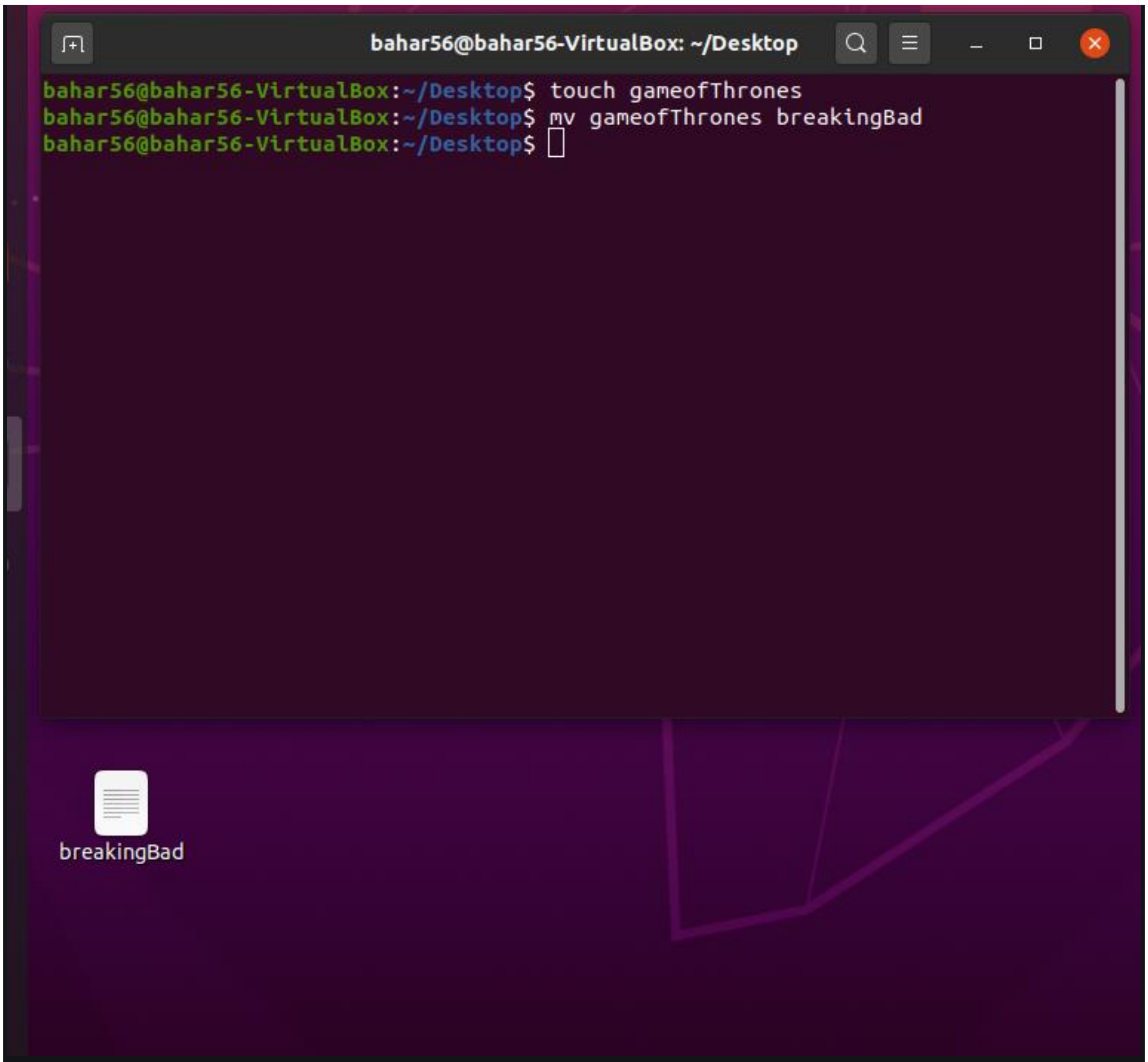
```
#! /bin/bash

echo -e "Enter the name of the file : \c"

read file_name


if [ -x $file_name ]

then

   echo "$file_name is executable"

else

   echo "$file_name is not executable"

fi
```

Output:



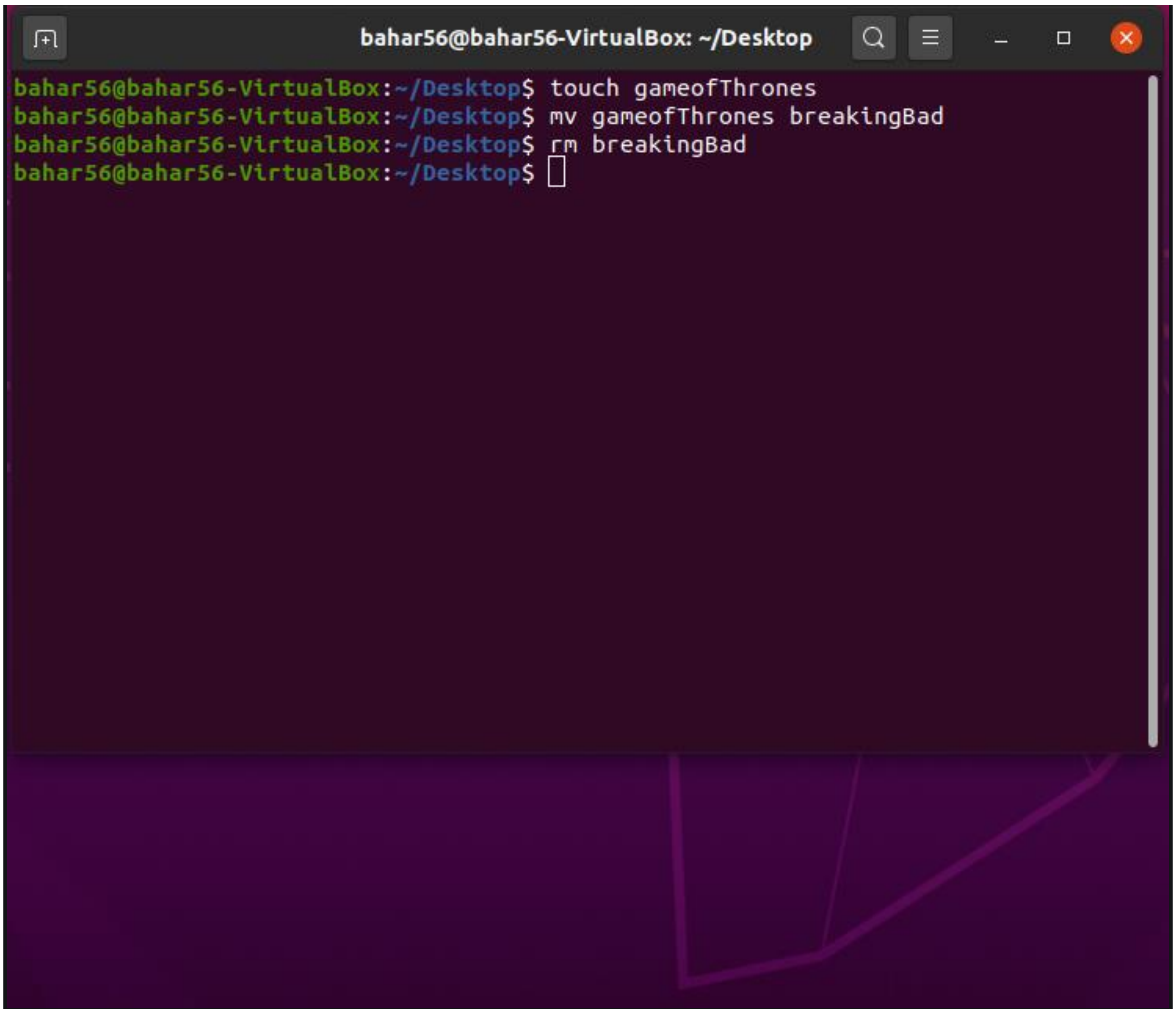
```
bahar56@bahar56-VirtualBox:~/Desktop$ ./test.sh
Enter the name of the file : geeks.txt
geeks.txt is not executable
bahar56@bahar56-VirtualBox:~/Desktop$ 
```

Rename & delete file: To rename the file, we use the ‘mv’ command which changes the name of the file and ‘rm’ to delete the file.



```
bahar56@bahar56-VirtualBox:~/Desktop$ touch gameofThrones
bahar56@bahar56-VirtualBox:~/Desktop$ mv gameofThrones breakingBad
bahar56@bahar56-VirtualBox:~/Desktop$ 
```

As we can see below the command line, breakingBad file (after renaming) and deleting it with ‘rm’ command, it is no longer there.



```
bahar56@bahar56-VirtualBox:~/Desktop$ touch gameofThrones
bahar56@bahar56-VirtualBox:~/Desktop$ mv gameofThrones breakingBad
bahar56@bahar56-VirtualBox:~/Desktop$ rm breakingBad
bahar56@bahar56-VirtualBox:~/Desktop$ 
```