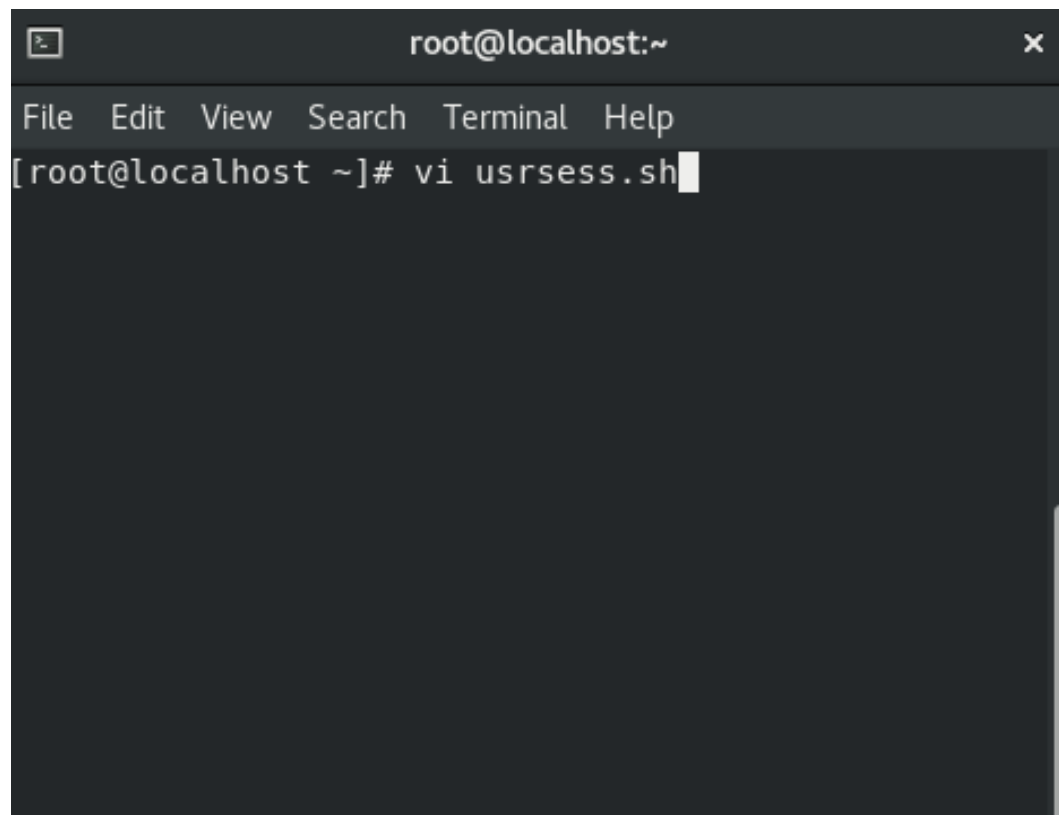


Assignment 0

1. Create a simple shell script to tell the user about their session – they need to know:

- What their username is
- What the current date is
- What the time is
- What their current working directory is
- How many files they have in that directory
- What is the biggest file in their current directory

Ans.



```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# vi usrsess.sh
```

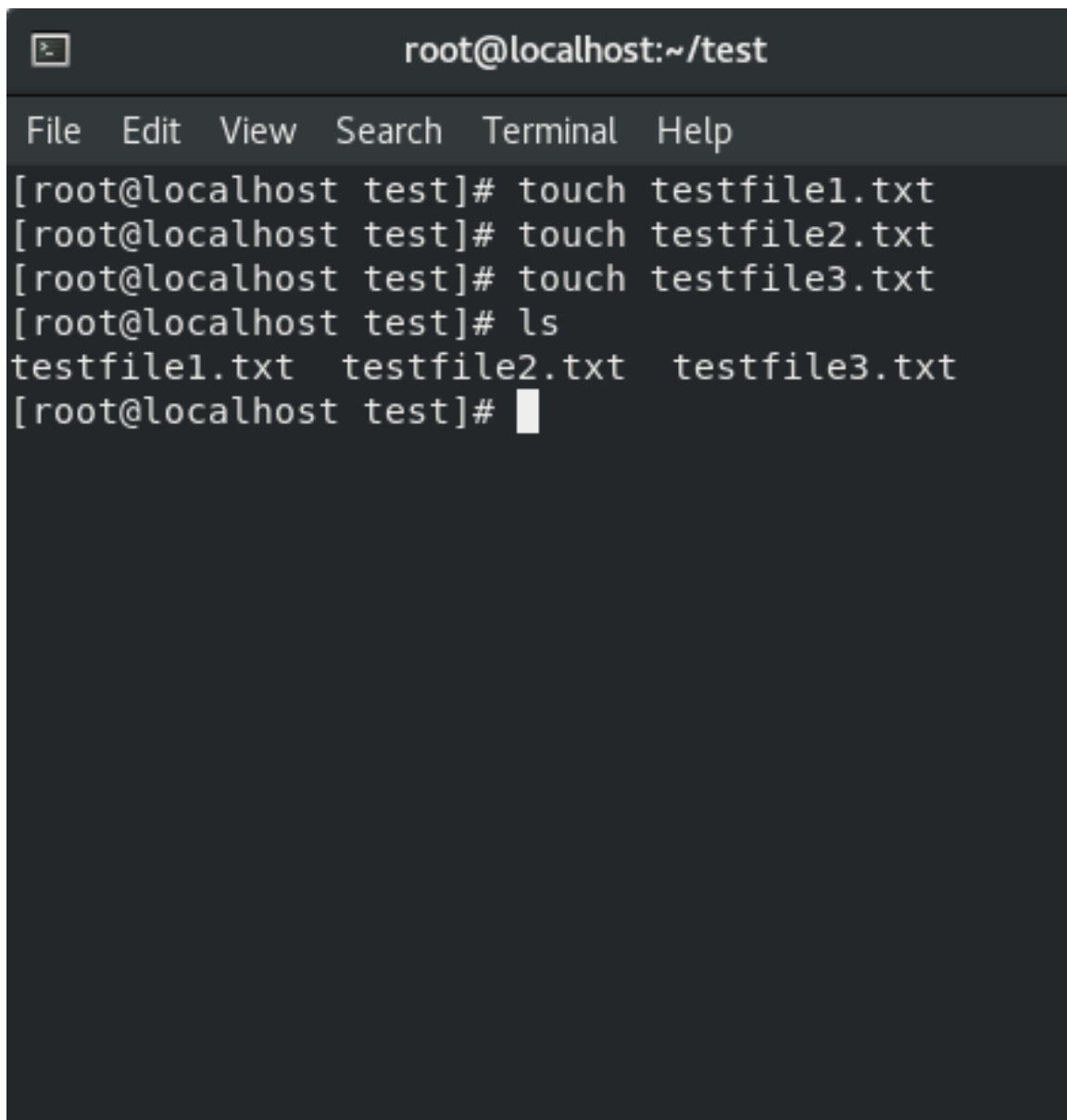
```
root@localhost:~  
File Edit View Search Terminal Help  
#To print User name  
echo "User name is $USER"  
  
#To print current date and Time  
echo "Today's Date is `date +%d/%m/%y`"  
echo "Time is `date +%r`"  
  
#To print current working directory  
echo "Your current working directory is $PWD"  
  
#To find the number of files and directories present in current directory  
if [ -d "$@" ]; then  
echo "Number of files is $(find "$@" -type f | wc -l)"  
else  
echo "[ERROR] please provide a directory."  
exit 1  
fi  
  
#To find biggest directory among present directory  
echo "Biggest file in current directory is."  
find . -printf '%s %p\n' |sort -nr|sed 1q  
  
~  
:wq
```

```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# vi usrseess.sh  
[root@localhost ~]# ./usrseess.sh  
User name is root  
Today's Date is 12/12/20  
Time is 05:38:19 PM  
Your current working directory is /root  
Number of files is 76  
Biggest file in current directory is.  
3637248 ./cache/tracker/meta.db  
[root@localhost ~]#
```

***NOTE:-** In case you are running this command for the first time kindly use “ `chmod +x usrseess.sh` ”

Assignment 1

Create a directory with a few test files in it (the files can be empty). Now write a script that for every file in that directory you rename it to have an extension of today's date in YYYYMMDD format. Assignment

A terminal window titled 'root@localhost:~/test' with a menu bar containing 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal shows the following commands and output:

```
[root@localhost test]# touch testfile1.txt
[root@localhost test]# touch testfile2.txt
[root@localhost test]# touch testfile3.txt
[root@localhost test]# ls
testfile1.txt  testfile2.txt  testfile3.txt
[root@localhost test]#
```

```
root@localhost:~/test
File Edit View Search Terminal Help
[root@localhost test]# touch testfile1.txt
[root@localhost test]# touch testfile2.txt
[root@localhost test]# touch testfile3.txt
[root@localhost test]# ls
testfile1.txt testfile2.txt testfile3.txt
[root@localhost test]# vi varchanger.sh
```

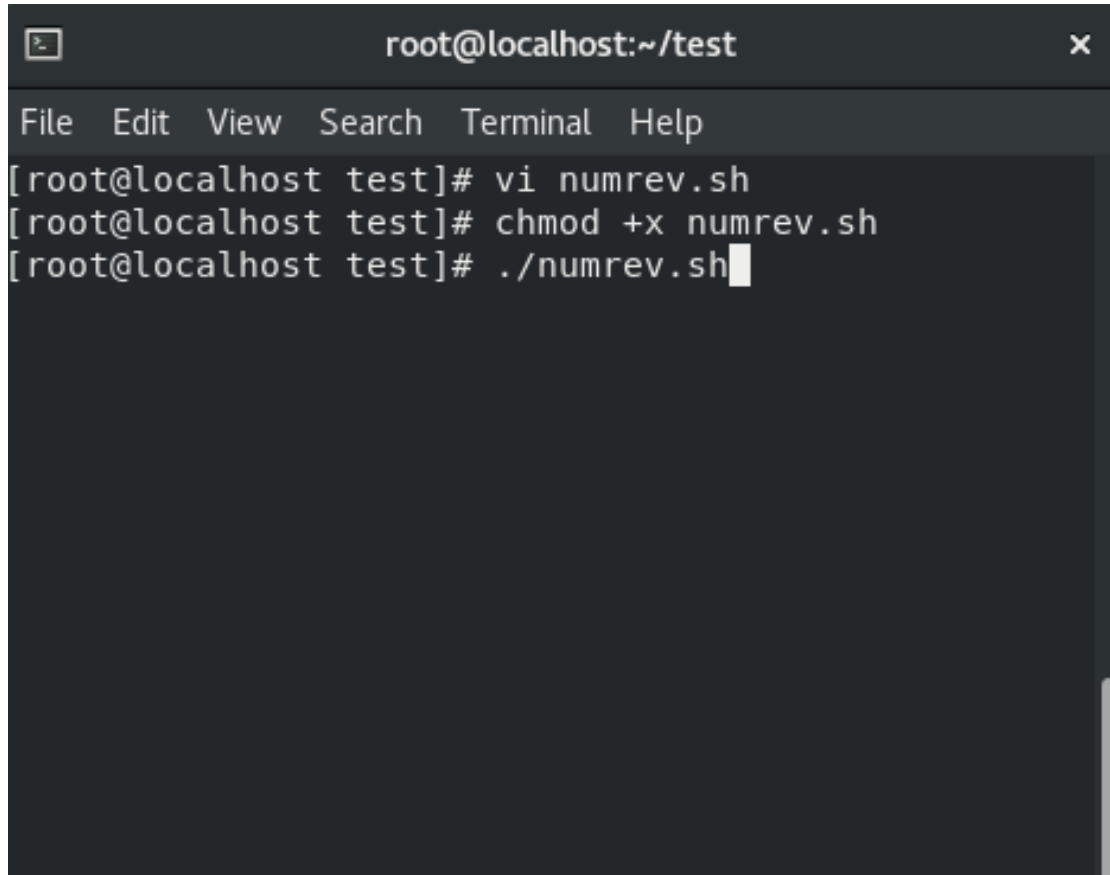
```
root@localhost:~/test
File Edit View Search Terminal Help
#!/bin/bash
DAY=$(date +%F)
cd /$PWD
for FILE in *.txt
do
    mv $FILE ${FILE}-${DAY}
done
~
~
~
~
~
~
~
~
~
```

```
root@localhost:~/test
File Edit View Search Terminal Help
[root@localhost test]# chmod +x varchanger.sh
[root@localhost test]# ./varchanger.sh
```

```
root@localhost:~/test
File Edit View Search Terminal Help
[root@localhost test]# ls
testfile1.txt-2020-12-12  testfile3.txt-2020-12-12
testfile2.txt-2020-12-12  varchanger.sh
[root@localhost test]#
```

Assignment 2

Write a script that takes a number as an input and reverses it out to the user.
For example, if the original number is 74985, the output should be 58947.

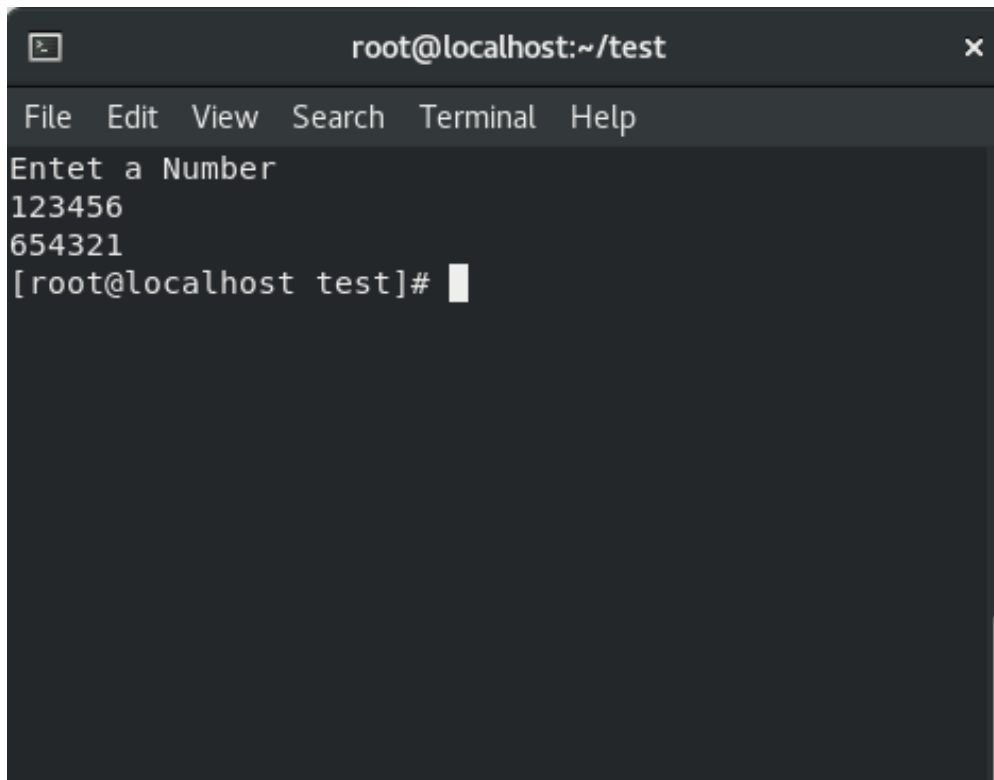
A terminal window titled 'root@localhost:~/test' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows three commands being executed: 'vi numrev.sh', 'chmod +x numrev.sh', and './numrev.sh'. The cursor is at the end of the third command.

```
root@localhost:~/test
File Edit View Search Terminal Help
[root@localhost test]# vi numrev.sh
[root@localhost test]# chmod +x numrev.sh
[root@localhost test]# ./numrev.sh
```

```
root@localhost:~/test
File Edit View Search Terminal Help
clear
echo "Entet a Number"
read x

if [[ $x =~ ^[0-9]+$ ]]; then
echo $x|rev
else
echo "Wrong Number"
fi
~
~
~
~
~
```

```
root@localhost:~/test
File Edit View Search Terminal Help
[root@localhost test]# vi numrev.sh
[root@localhost test]# chmod +x numrev.sh
[root@localhost test]# ./numrev.sh
```

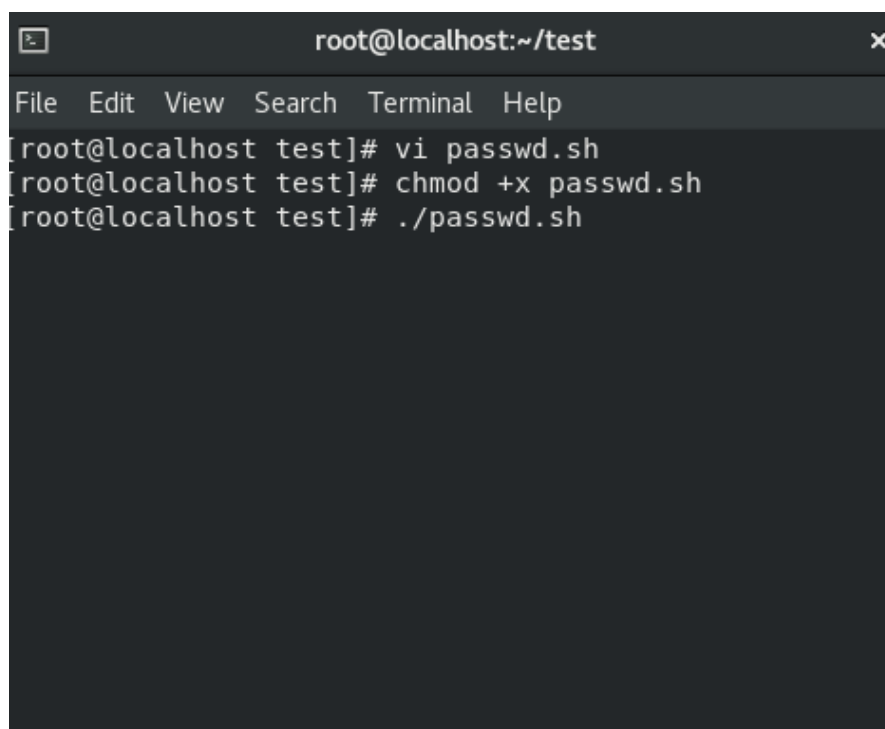
A terminal window titled 'root@localhost:~/test' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the execution of a script named 'passwd.sh'. The script prompts 'Entet a Number' and takes two inputs: '123456' and '654321'. The prompt '[root@localhost test]#' is visible at the end of the second input line.

```
root@localhost:~/test
File Edit View Search Terminal Help
Entet a Number
123456
654321
[root@localhost test]#
```

Assignment 3

Write a script to validate how secure someone's password is. Things you would care about:

- Length should be 8 or more characters
- The password should contain numbers and letters
- There should be both uppercase and lowercase letters

A terminal window titled 'root@localhost:~/test' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the creation and execution of a script named 'passwd.sh'. The user enters 'vi passwd.sh' to create the file, 'chmod +x passwd.sh' to make it executable, and './passwd.sh' to run it.

```
root@localhost:~/test
File Edit View Search Terminal Help
[root@localhost test]# vi passwd.sh
[root@localhost test]# chmod +x passwd.sh
[root@localhost test]# ./passwd.sh
```



```
root@localhost:~/test
File Edit View Search Terminal Help
clear
echo "Enter Your Password"
read pass
len=`expr length "$pass"`
if(($len<8))
then
echo "Your Password must contain at least 8 characters"
else
if [[ $pass =~ ^[:alnum:]+$ ]];
then
echo $pass | grep "[A-Z]" |grep "[a-z]" | grep "[0-9]"
if [[ $? -ne 0 ]];then
echo "Your Password should contain both Uppercase and Lowercase character"
else
echo "your Password is OK"
fi
else
echo "your Password should have only one number and letter"
fi
fi
~
~
~
~
~
~
:wg
```

```
root@localhost:~/test
File Edit View Search Terminal Help
Enter Your Password
```