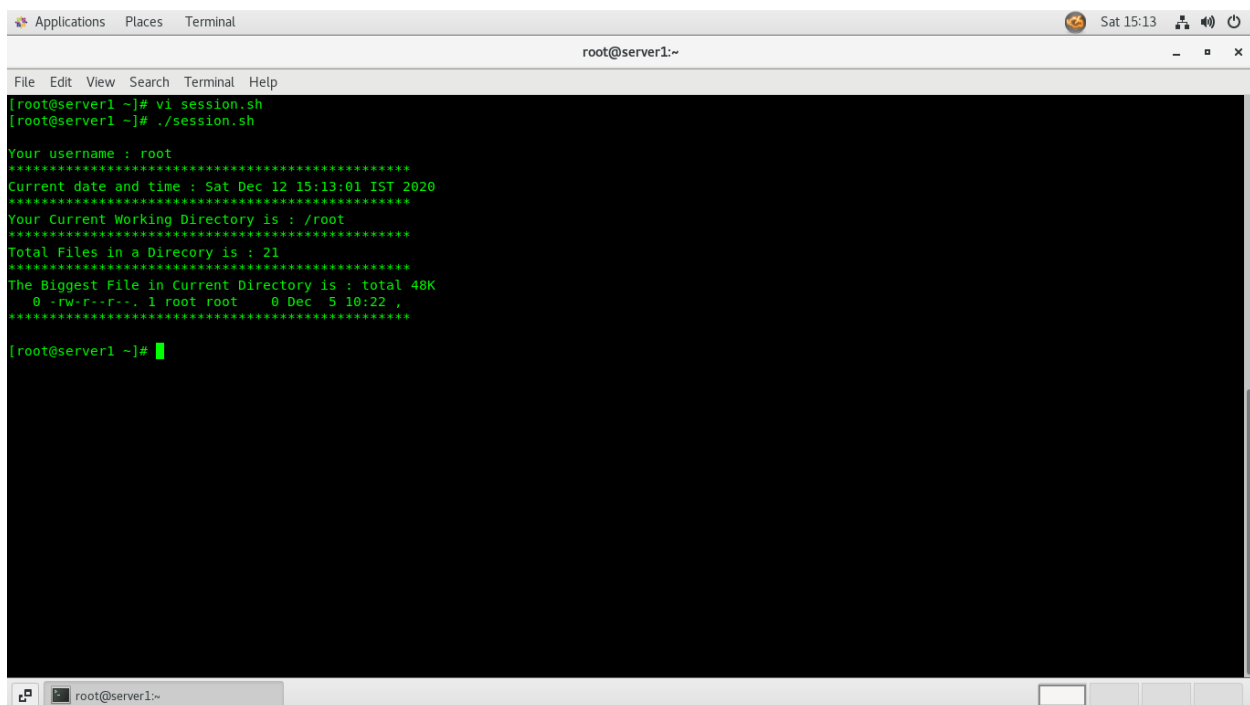


# ASSIGNMENT NO 4

## 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



The screenshot shows a terminal window titled "root@server1:~" with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal displays the following output:

```
[root@server1 ~]# vi session.sh
[root@server1 ~]# ./session.sh

Your username : root
*****
Current date and time : Sat Dec 12 15:13:01 IST 2020
*****
Your Current Working Directory is : /root
*****
Total Files in a Direcorey is : 21
*****
The Biggest File in Current Directory is : total 48K
  0 -rw-r--r-- 1 root root   0 Dec  5 10:22 ,
*****

[root@server1 ~]#
```

The terminal window has a status bar at the bottom showing "root@server1:~" and a scroll bar on the right.

# ASSIGNMENT NO 4

The screenshot shows a terminal window with the title bar 'Applications Places Terminal'. The terminal prompt is 'root@server1:~'. The user has executed a script that outputs the following information:

```

root@server1:~# ./bin/bash
# Write a shell script called hello which output the following:
# + Your username
# + The time and date
function line(){
    echo "*****"
}
echo " "
echo "Your username : $(echo $USER)"
line # call function
echo "Current date and time : $(date)"
line
echo "Your Current Working Directory is : $PWD"
line
echo "Total Files in a Direcorey is : $( ls -l | wc -l )"
line
echo "The Biggest File in Current Directory is : $(ls -lsh | head -2 )"
line
echo " "

```

At the bottom of the terminal, the command 'session.sh' is shown with its output: '19L, 508C'.

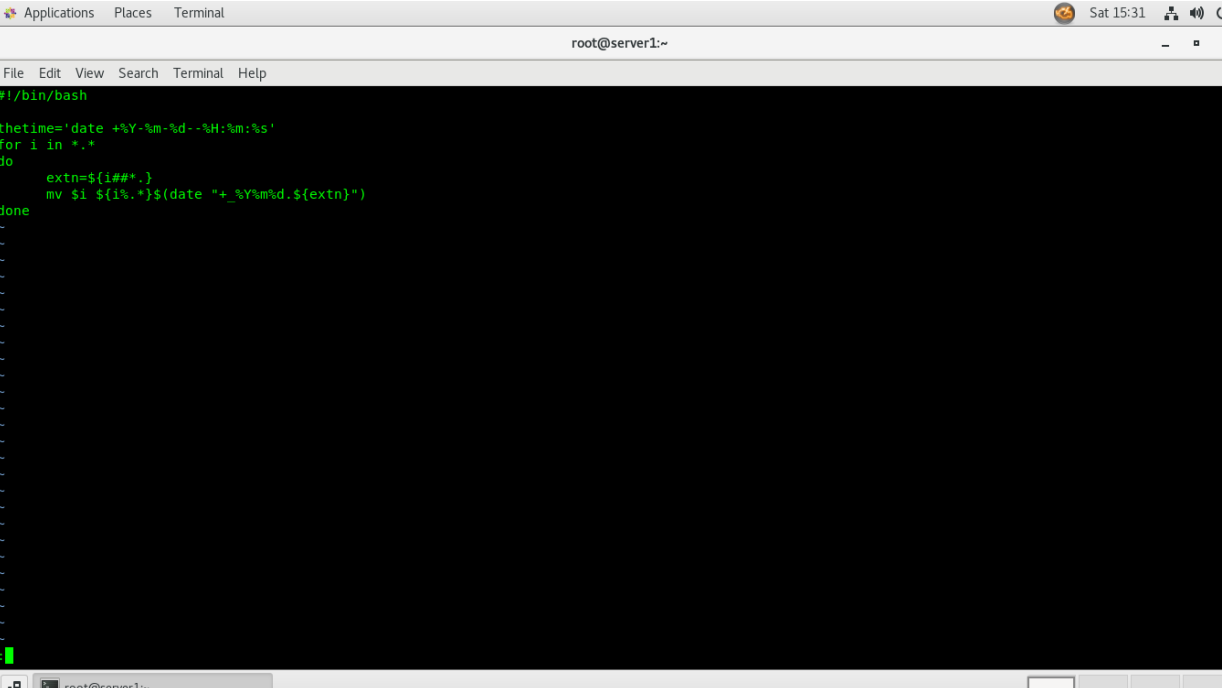
# Assignment 1

Create a directory with a few test files in it (the files can be empty). Now write a script that for

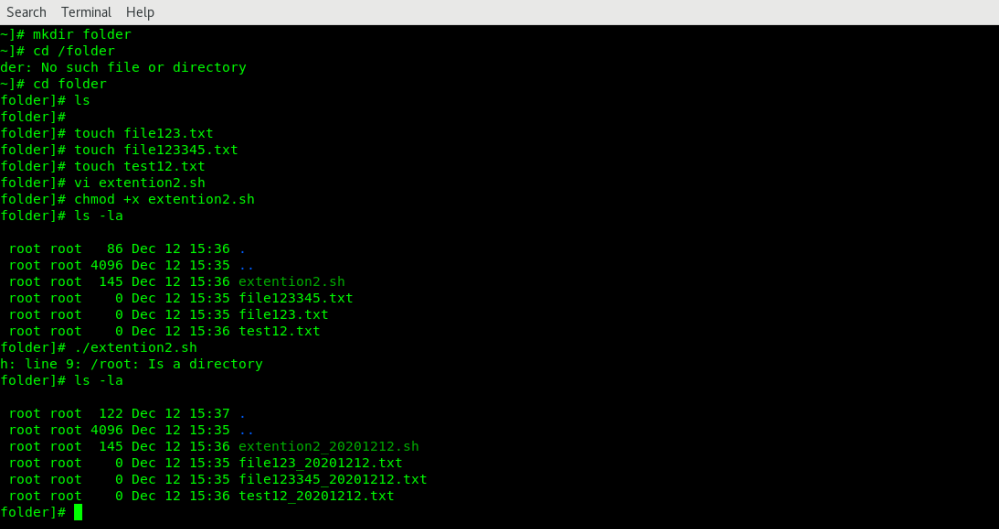
every file in that directory you rename it to have an extension of today's date in YYYYMMDD

format.

# ASSIGNMENT NO 4

A screenshot of a Linux terminal window. The title bar at the top shows 'Applications', 'Places', and 'Terminal'. The system status bar on the right indicates 'Sat 15:31' and includes icons for network, volume, and power. The terminal's command prompt is 'root@server1:~'. The menu bar contains 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal content shows a shell script being executed: 

```
#!/bin/bash
thetime='date +%Y-%m-%d--%H:%m:%s'
for i in *.*
do
    extn=${i##*.}
    mv $i ${i%.*}${(date "+_Y%m%d.${extn}")}
done
```

 A green cursor is visible at the bottom left of the terminal window. The taskbar at the bottom shows the terminal icon and the active window title 'root@server1:~'.

The screenshot shows a terminal window with the following content:

```

root@server1:~/folder

File Edit View Search Terminal Help

[root@server1 ~]# mkdir folder
[root@server1 ~]# cd /folder
bash: cd: /folder: No such file or directory
[root@server1 ~]# cd folder
[root@server1 folder]# ls
[root@server1 folder]#
[root@server1 folder]# touch file123.txt
[root@server1 folder]# touch file123345.txt
[root@server1 folder]# touch test12.txt
[root@server1 folder]# vi extention2.sh
[root@server1 folder]# chmod +x extention2.sh
[root@server1 folder]# ls -la
total 8
drwxr-xr-x. 2 root root 86 Dec 12 15:36 .
dr-xr-x---. 19 root root 4096 Dec 12 15:35 ..
-rwxr-xr-x. 1 root root 145 Dec 12 15:36 extention2.sh
-rw-r--r--. 1 root root 0 Dec 12 15:35 file123345.txt
-rw-r--r--. 1 root root 0 Dec 12 15:35 file123.txt
-rw-r--r--. 1 root root 0 Dec 12 15:36 test12.txt
[root@server1 folder]# ./extention2.sh
./extention2.sh: line 9: /root: Is a directory
[root@server1 folder]# ls -la
total 8
drwxr-xr-x. 2 root root 122 Dec 12 15:37 .
dr-xr-x---. 19 root root 4096 Dec 12 15:35 ..
-rwxr-xr-x. 1 root root 145 Dec 12 15:36 extention2_20201212.sh
-rw-r--r--. 1 root root 0 Dec 12 15:35 file123_20201212.txt
-rw-r--r--. 1 root root 0 Dec 12 15:35 file123345_20201212.txt
-rw-r--r--. 1 root root 0 Dec 12 15:36 test12_20201212.txt
[root@server1 folder]#

```

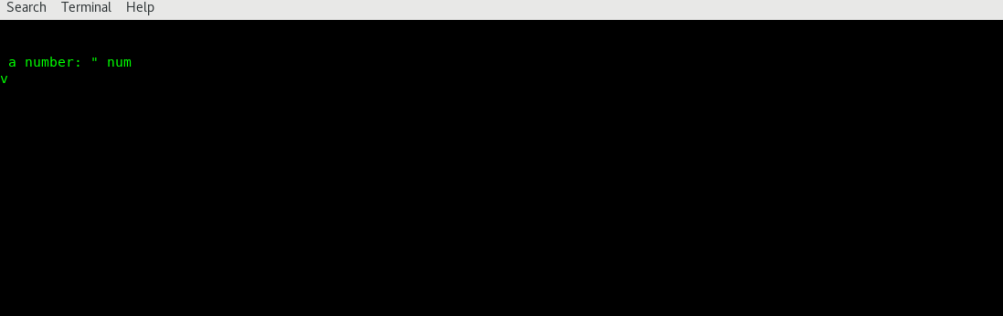
The terminal window has a title bar with "Applications", "Places", and "Terminal" buttons. The top right corner shows system status icons and the time "Sat 15:38". The bottom of the window shows a taskbar with a "root@server1:~/folder" window icon and several empty window slots.

# ASSIGNMENT NO 4

## 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.

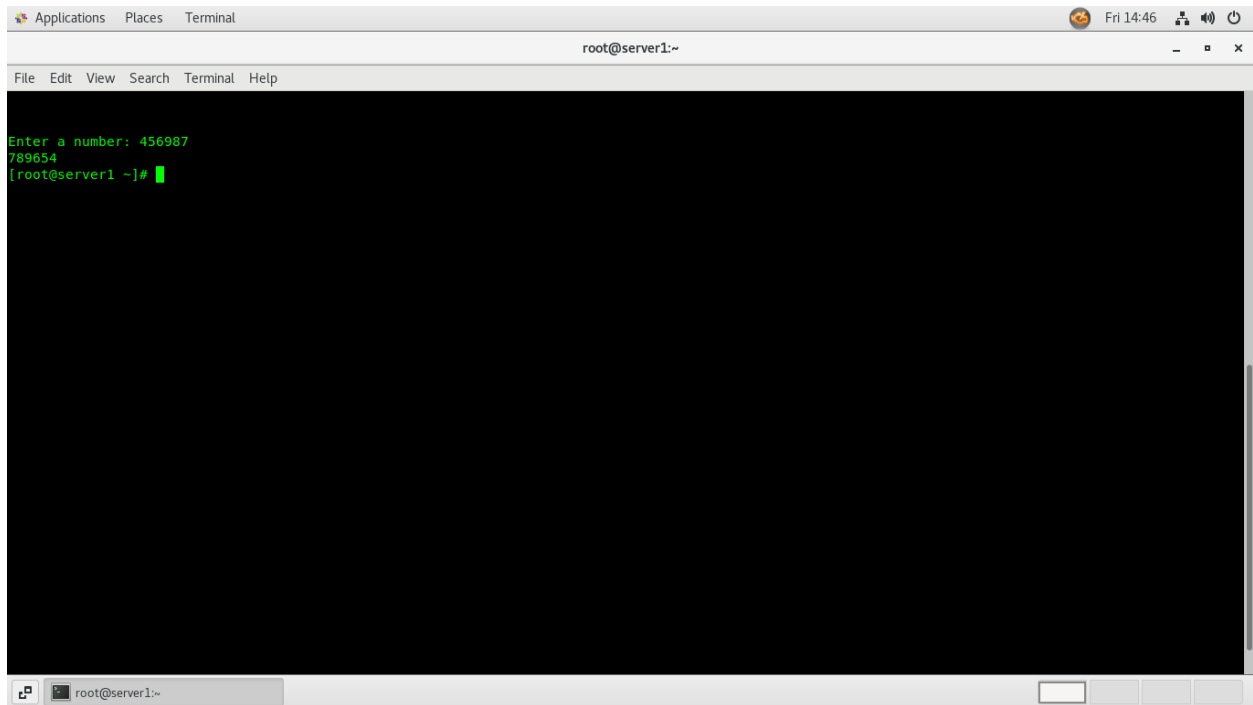


The screenshot shows a terminal window titled "root@server1:~". The terminal content is as follows:

```
#!/bin/bash
clear
read -p "Enter a number: " num
echo $num | rev
```

The prompt "-- INSERT --" is visible at the bottom of the terminal. The window's title bar includes "Applications", "Places", and "Terminal". The system status bar at the top right shows "Fri 14:45" and icons for network, volume, and power.

# ASSIGNMENT NO 4



The screenshot shows a terminal window titled 'root@server1:~'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal output is as follows:

```
Enter a number: 456987
789654
[root@server1 ~]#
```

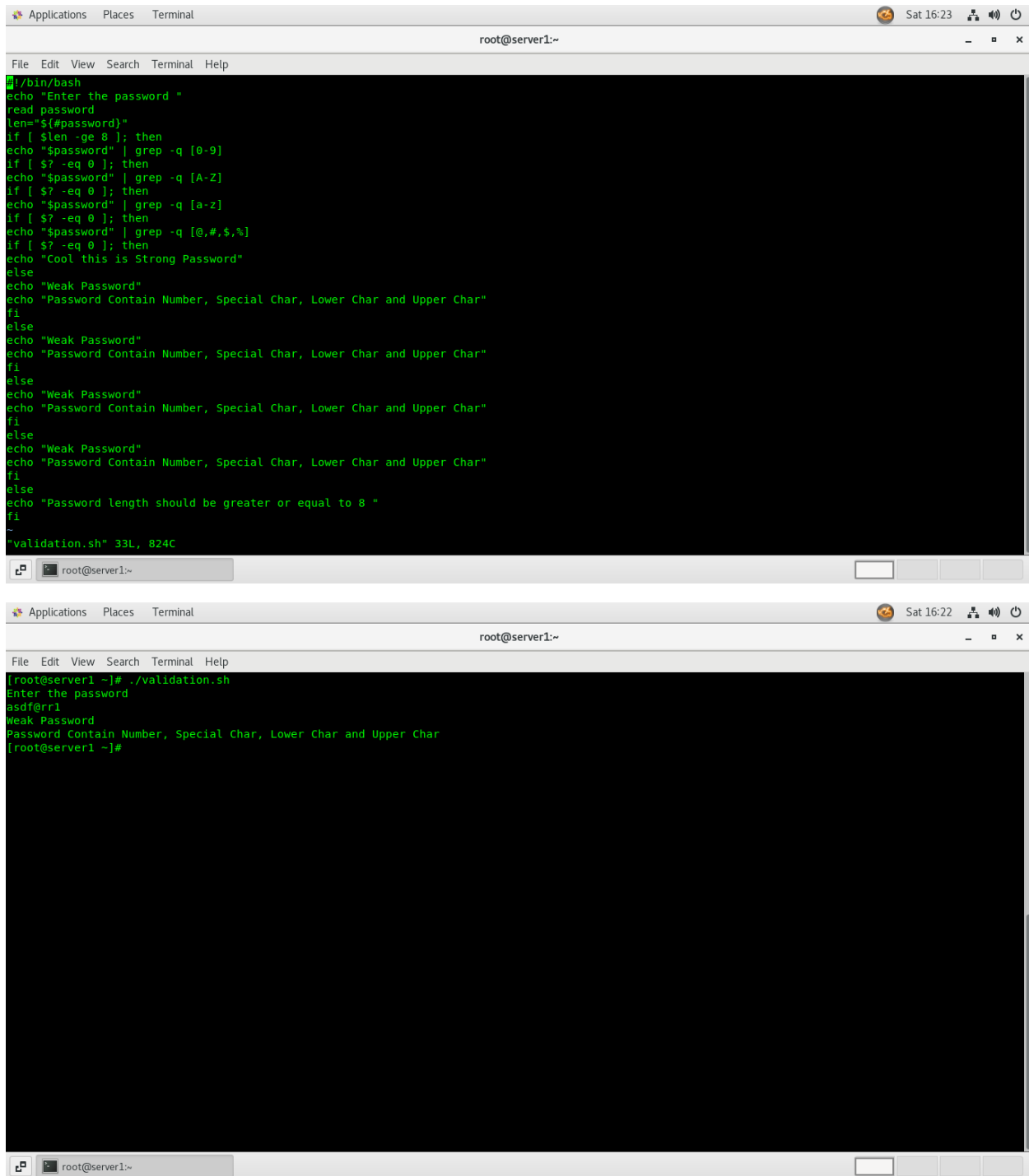
The prompt is green, and the input and output are also green. The terminal window is part of a desktop environment with a top bar showing 'Applications', 'Places', and 'Terminal'. The system clock shows 'Fri 14:46'.

## 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

# ASSIGNMENT NO 4



The image displays two screenshots of a Linux terminal window. The top screenshot shows the execution of a script named `validation.sh`. The script prompts the user to enter a password and then checks it against several criteria: length (must be at least 8 characters), uppercase letters (must contain at least one), lowercase letters (must contain at least one), and special characters (must contain at least one). If the password meets all these criteria, it is considered "Cool this is Strong Password". Otherwise, it is labeled "Weak Password" and the user is prompted to contain at least one of the missing character types. The bottom screenshot shows the user entering the password `asdf@rr1`, which fails the validation checks because it lacks uppercase letters and special characters.

```
root@server1:~  
File Edit View Search Terminal Help  
~/bin/bash  
echo "Enter the password "  
read password  
len=${#password}  
if [ $len -ge 8 ]; then  
echo "$password" | grep -q [0-9]  
if [ $? -eq 0 ]; then  
echo "$password" | grep -q [A-Z]  
if [ $? -eq 0 ]; then  
echo "$password" | grep -q [a-z]  
if [ $? -eq 0 ]; then  
echo "$password" | grep -q [!@#$%^&*~<.>|:;"]  
if [ $? -eq 0 ]; then  
echo "Cool this is Strong Password"  
else  
echo "Weak Password"  
echo "Password Contain Number, Special Char, Lower Char and Upper Char"  
fi  
else  
echo "Weak Password"  
echo "Password Contain Number, Special Char, Lower Char and Upper Char"  
fi  
else  
echo "Weak Password"  
echo "Password Contain Number, Special Char, Lower Char and Upper Char"  
fi  
else  
echo "Weak Password"  
echo "Password Contain Number, Special Char, Lower Char and Upper Char"  
fi  
else  
echo "Password length should be greater or equal to 8 "  
fi  
fi  
~  
"validation.sh" 33L, 824C  
root@server1:~  
Applications Places Terminal Sat 16:23
```

```
root@server1:~  
File Edit View Search Terminal Help  
[root@server1 ~]# ./validation.sh  
Enter the password  
asdf@rr1  
Weak Password  
Password Contain Number, Special Char, Lower Char and Upper Char  
[root@server1 ~]#  
root@server1:~  
Applications Places Terminal Sat 16:22
```