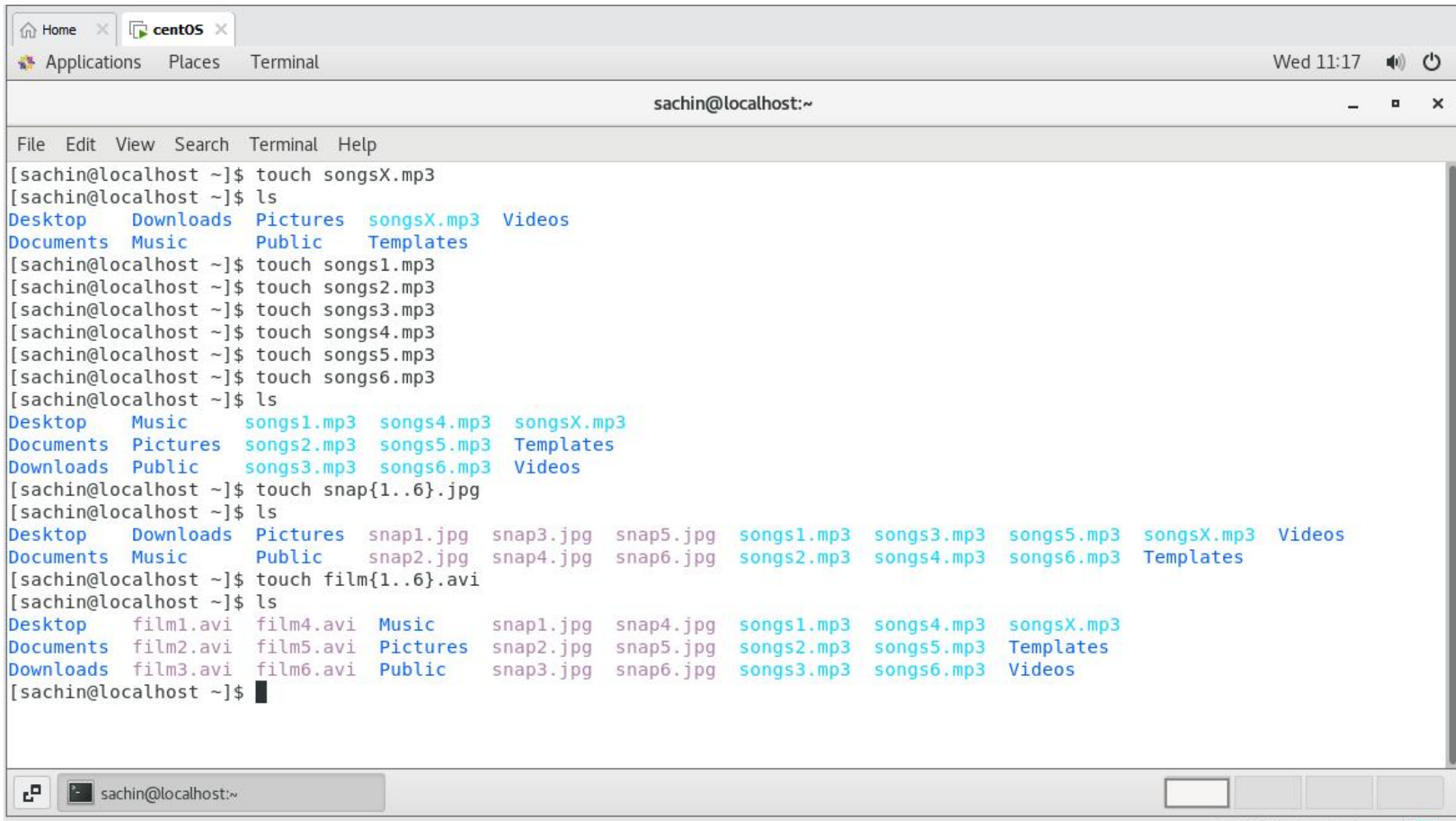


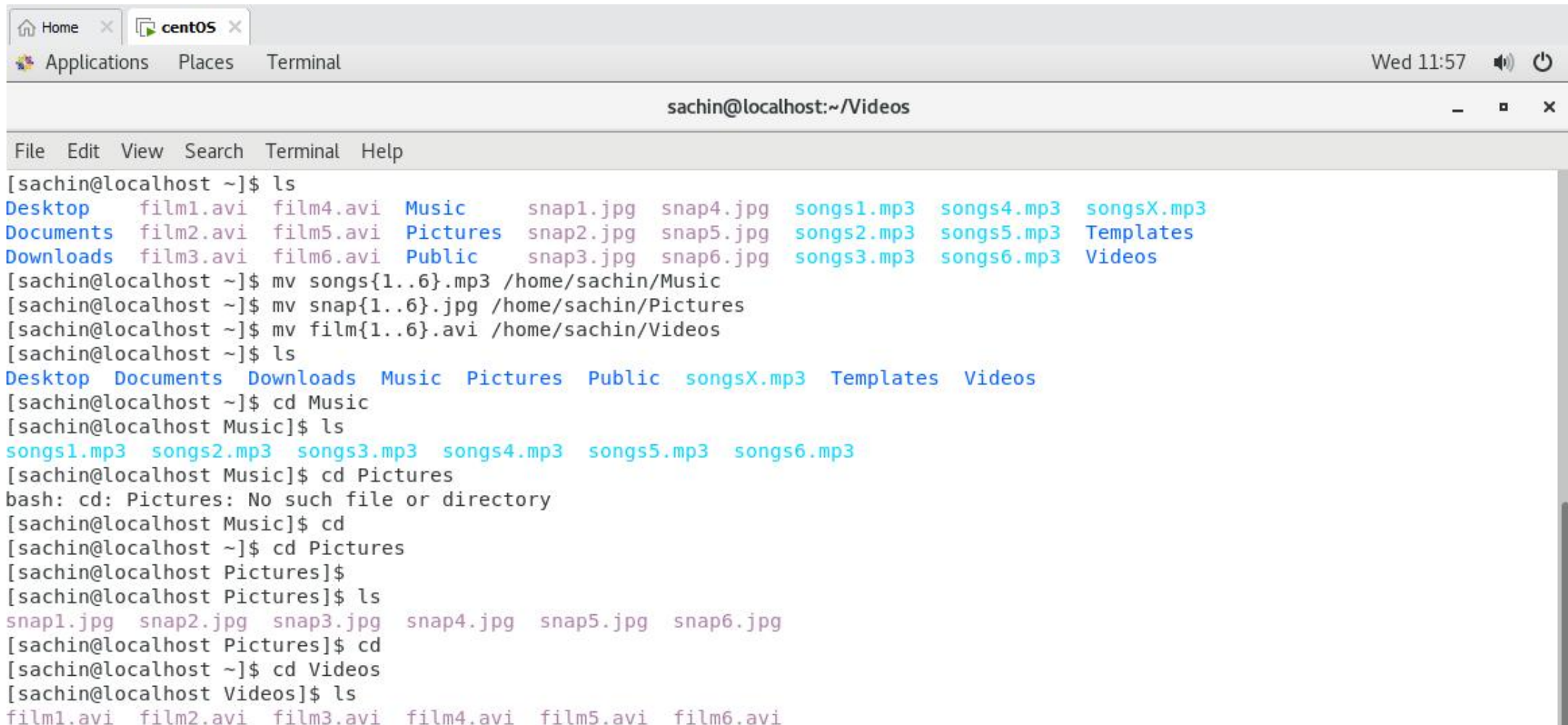
Assignment-1

1. In your home directory, create sets of empty practice files. Create 6 files with names of the form songsX.mp3. Create 6 files with names of the form snapX.jpg. Create 6 files with names of the form filmX.avi. In each set, replace X with the numbers 1 through 6.



```
sachin@localhost:~$ touch songsX.mp3
sachin@localhost:~$ ls
Desktop  Downloads  Pictures  songsX.mp3  Videos
Documents Music      Public    Templates
sachin@localhost:~$ touch songs1.mp3
sachin@localhost:~$ touch songs2.mp3
sachin@localhost:~$ touch songs3.mp3
sachin@localhost:~$ touch songs4.mp3
sachin@localhost:~$ touch songs5.mp3
sachin@localhost:~$ touch songs6.mp3
sachin@localhost:~$ ls
Desktop  Music      songs1.mp3  songs4.mp3  songsX.mp3
Documents Pictures  songs2.mp3  songs5.mp3  Templates
Downloads Public    songs3.mp3  songs6.mp3  Videos
sachin@localhost:~$ touch snap{1..6}.jpg
sachin@localhost:~$ ls
Desktop  Downloads  Pictures  snap1.jpg  snap3.jpg  snap5.jpg  songs1.mp3  songs3.mp3  songs5.mp3  songsX.mp3  Videos
Documents Music      Public    snap2.jpg  snap4.jpg  snap6.jpg  songs2.mp3  songs4.mp3  songs6.mp3  Templates
sachin@localhost:~$ touch film{1..6}.avi
sachin@localhost:~$ ls
Desktop  film1.avi  film4.avi  Music      snap1.jpg  snap4.jpg  songs1.mp3  songs4.mp3  songsX.mp3
Documents film2.avi  film5.avi  Pictures    snap2.jpg  snap5.jpg  songs2.mp3  songs5.mp3  Templates
Downloads film3.avi  film6.avi  Public      snap3.jpg  snap6.jpg  songs3.mp3  songs6.mp3  Videos
sachin@localhost:~$
```

2. From your home directory,
Move songs file into your Music subdirectory.
Move snap file into your Pictures subdirectory.
Move your movie files into Videos subdirectory



The screenshot shows a terminal window titled "sachin@localhost:~/Videos". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal output shows the following commands and results:

```
[sachin@localhost ~]$ ls
Desktop    film1.avi  film4.avi  Music      snap1.jpg  snap4.jpg  songs1.mp3  songs4.mp3  songsX.mp3
Documents  film2.avi  film5.avi  Pictures   snap2.jpg  snap5.jpg  songs2.mp3  songs5.mp3  Templates
Downloads  film3.avi  film6.avi  Public     snap3.jpg  snap6.jpg  songs3.mp3  songs6.mp3  Videos

[sachin@localhost ~]$ mv songs{1..6}.mp3 /home/sachin/Music
[sachin@localhost ~]$ mv snap{1..6}.jpg /home/sachin/Pictures
[sachin@localhost ~]$ mv film{1..6}.avi /home/sachin/Videos
[sachin@localhost ~]$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  songsX.mp3  Templates  Videos

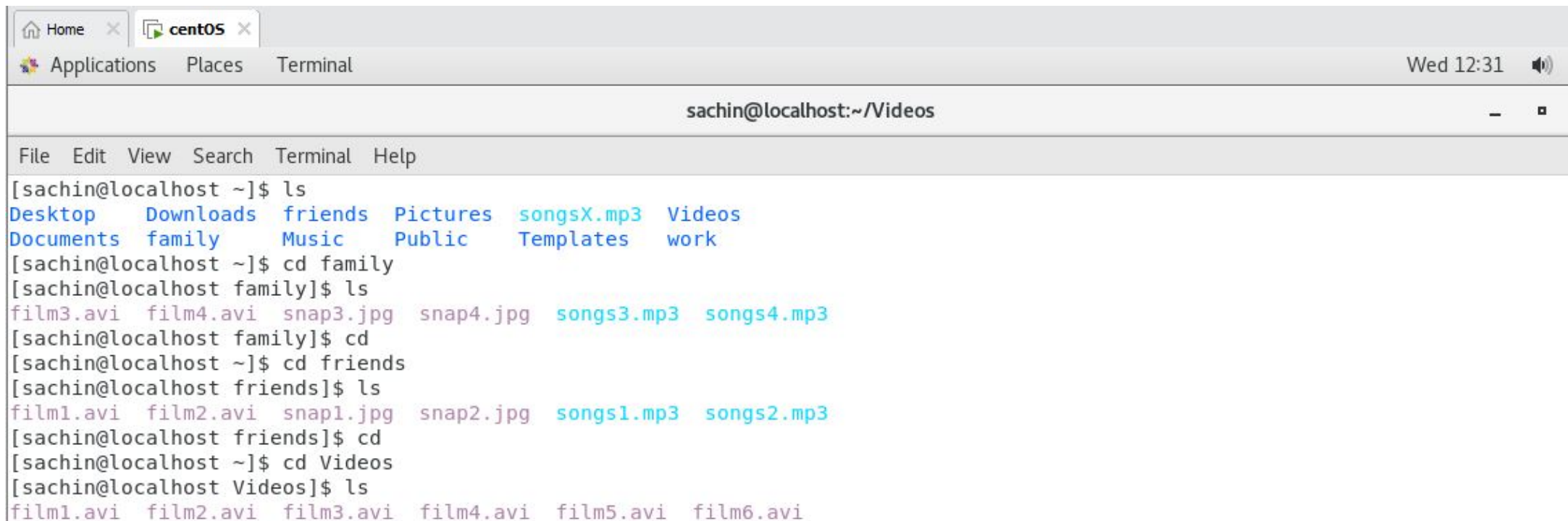
[sachin@localhost ~]$ cd Music
[sachin@localhost Music]$ ls
songs1.mp3  songs2.mp3  songs3.mp3  songs4.mp3  songs5.mp3  songs6.mp3

[sachin@localhost Music]$ cd Pictures
bash: cd: Pictures: No such file or directory

[sachin@localhost Music]$ cd
[sachin@localhost ~]$ cd Pictures
[sachin@localhost Pictures]$ ls
snap1.jpg  snap2.jpg  snap3.jpg  snap4.jpg  snap5.jpg  snap6.jpg

[sachin@localhost Pictures]$ cd
[sachin@localhost ~]$ cd Videos
[sachin@localhost Videos]$ ls
film1.avi  film2.avi  film3.avi  film4.avi  film5.avi  film6.avi
```

3. Create 3 subdirectories for organizing your files named friends,family,work
4. Copy files (all types) containing numbers 1 and 2 to the friends folder.
Copy files (all types) containing numbers 3 and 4 to the family folder.
Copy files (all types) containing numbers 5 and 6 to the work folder.



The screenshot shows a Linux desktop environment with a window titled 'centOS'. The window has tabs for 'Home', 'Applications', 'Places', and 'Terminal'. The 'Terminal' tab is active, showing the command prompt 'sachin@localhost:~/Videos'. The terminal output shows the following commands and results:

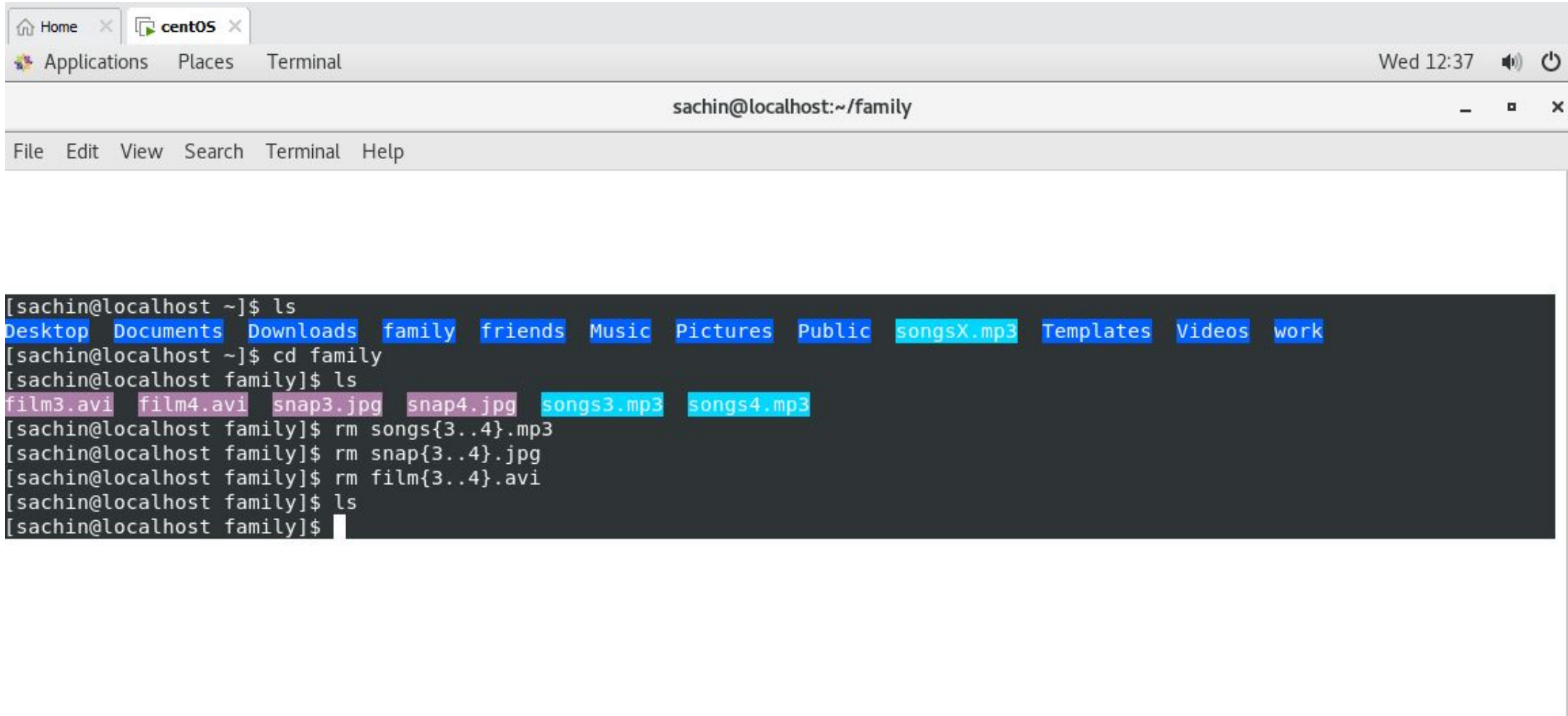
```
[sachin@localhost ~]$ ls
Desktop  Downloads  friends  Pictures  songsX.mp3  Videos
Documents  family      Music    Public    Templates  work

[sachin@localhost ~]$ cd family
[sachin@localhost family]$ ls
film3.avi  film4.avi  snap3.jpg  snap4.jpg  songs3.mp3  songs4.mp3

[sachin@localhost family]$ cd
[sachin@localhost ~]$ cd friends
[sachin@localhost friends]$ ls
film1.avi  film2.avi  snap1.jpg  snap2.jpg  songs1.mp3  songs2.mp3

[sachin@localhost friends]$ cd
[sachin@localhost ~]$ cd Videos
[sachin@localhost Videos]$ ls
film1.avi  film2.avi  film3.avi  film4.avi  film5.avi  film6.avi
```

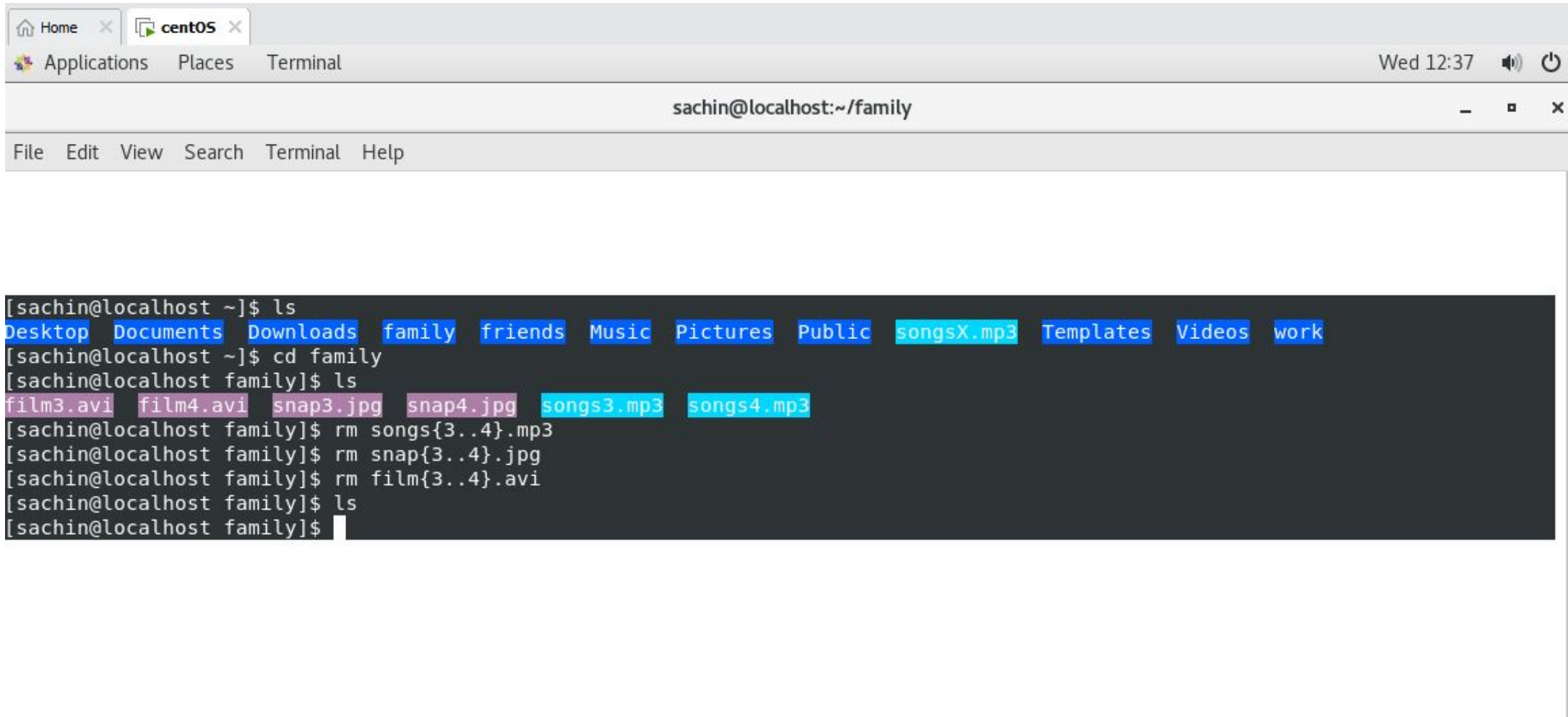
5. Delete all files in family subdirectory.



The screenshot shows a Linux desktop environment with a window titled "centOS". The window has a menu bar with "Applications", "Places", and "Terminal". The title bar also shows "sachin@localhost:~/family" and system icons for volume and power. The terminal window displays the following commands and output:

```
[sachin@localhost ~]$ ls
Desktop  Documents  Downloads  family  friends  Music  Pictures  Public  songsX.mp3  Templates  Videos  work
[sachin@localhost ~]$ cd family
[sachin@localhost family]$ ls
film3.avi  film4.avi  snap3.jpg  snap4.jpg  songs3.mp3  songs4.mp3
[sachin@localhost family]$ rm songs{3..4}.mp3
[sachin@localhost family]$ rm snap{3..4}.jpg
[sachin@localhost family]$ rm film{3..4}.avi
[sachin@localhost family]$ ls
[sachin@localhost family]$
```

6. Delete friends subdirectory



The screenshot shows a Linux desktop environment with a window titled "centOS". The window has a menu bar with "Applications", "Places", and "Terminal". The title bar also shows "sachin@localhost:~/family" and system icons for volume and power. The terminal window displays the following commands and output:

```
[sachin@localhost ~]$ ls
Desktop  Documents  Downloads  family  friends  Music  Pictures  Public  songsX.mp3  Templates  Videos  work
[sachin@localhost ~]$ cd family
[sachin@localhost family]$ ls
film3.avi  film4.avi  snap3.jpg  snap4.jpg  songs3.mp3  songs4.mp3
[sachin@localhost family]$ rm songs{3..4}.mp3
[sachin@localhost family]$ rm snap{3..4}.jpg
[sachin@localhost family]$ rm film{3..4}.avi
[sachin@localhost family]$ ls
[sachin@localhost family]$
```


7. rename all snap files in work directory to new name photoX.jpg

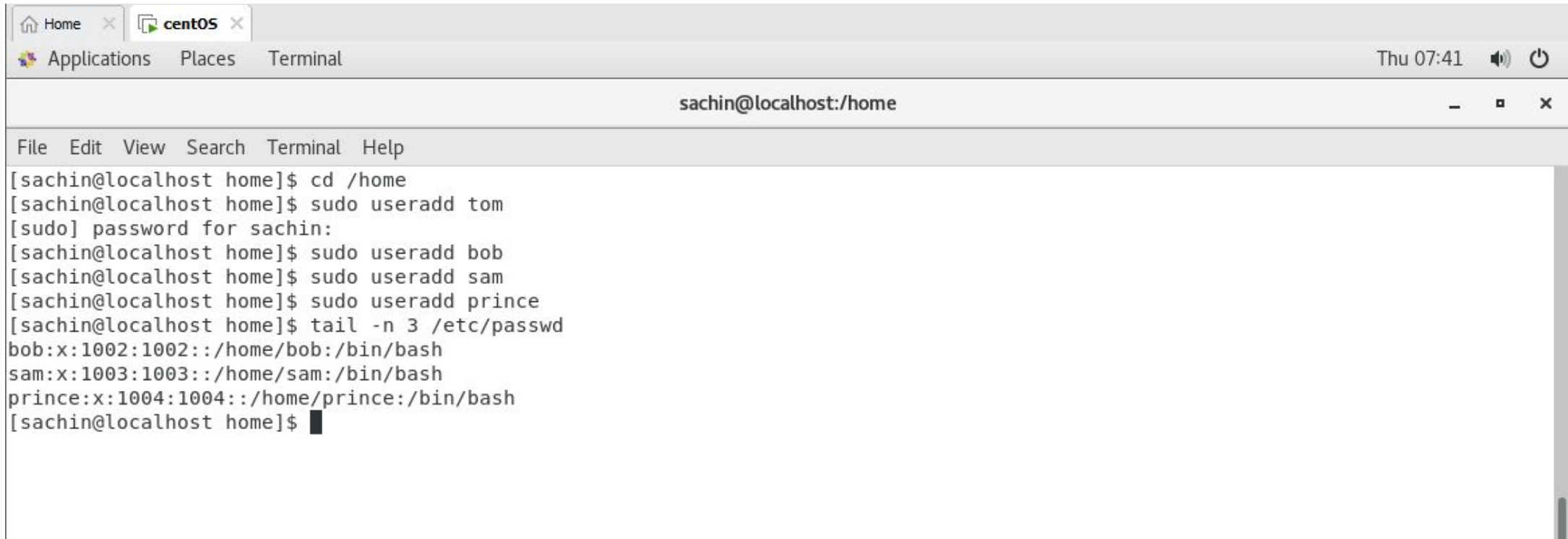


The image shows a terminal window titled 'centOS' with a menu bar containing 'Applications', 'Places', and 'Terminal'. The window title bar also shows 'sachin@localhost:~/work'. The terminal content shows the following commands and output:

```
[sachin@localhost ~]$ ls
Desktop Documents Downloads family Music Pictures Public songsX.mp3 Templates Videos work
[sachin@localhost ~]$ cd work
[sachin@localhost work]$ ls
film5.avi film6.avi snap5.jpg snap6.jpg songs5.mp3 songs6.mp3
[sachin@localhost work]$ mv snap5.jpg photo5.jpg
[sachin@localhost work]$ mv snap6.jpg photo6.jpg
[sachin@localhost work]$ ls
film5.avi film6.avi photo5.jpg photo6.jpg songs5.mp3 songs6.mp3
[sachin@localhost work]$
```

Assignment-2

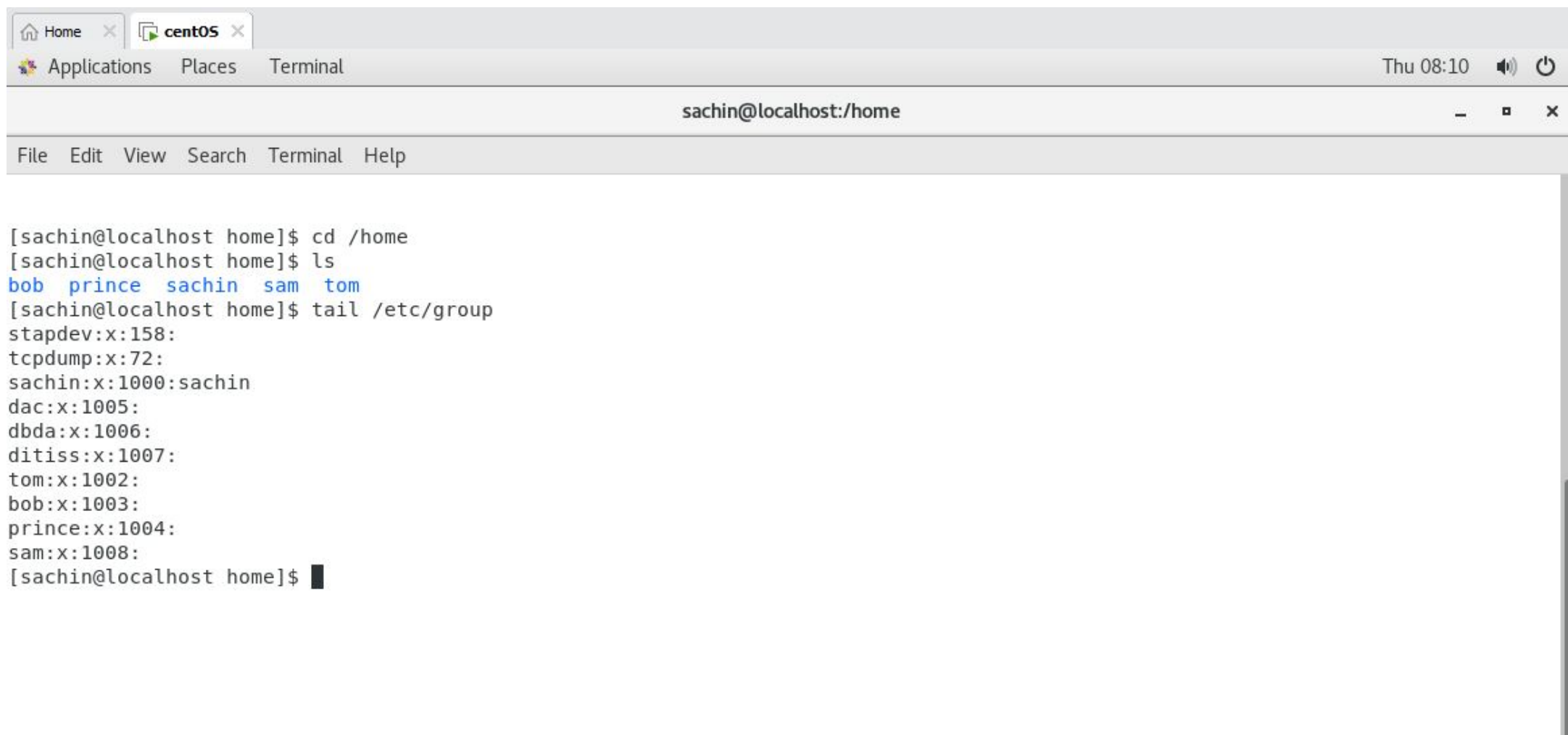
1. Create user tom , bob , sam , prince



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window has a title bar with 'Home', 'centOS', and 'Terminal' tabs. The terminal content shows the user 'sachin' at 'localhost' in the '/home' directory. The user runs 'cd /home' and then 'sudo useradd tom'. A password prompt appears, and the user enters a password. Then, the user runs 'sudo useradd bob', 'sudo useradd sam', and 'sudo useradd prince'. Finally, the user runs 'tail -n 3 /etc/passwd', which displays the entries for 'bob', 'sam', and 'prince' in the /etc/passwd file. The terminal output is as follows:

```
[sachin@localhost home]$ cd /home
[sachin@localhost home]$ sudo useradd tom
[sudo] password for sachin:
[sachin@localhost home]$ sudo useradd bob
[sachin@localhost home]$ sudo useradd sam
[sachin@localhost home]$ sudo useradd prince
[sachin@localhost home]$ tail -n 3 /etc/passwd
bob:x:1002:1002::/home/bob:/bin/bash
sam:x:1003:1003::/home/sam:/bin/bash
prince:x:1004:1004::/home/prince:/bin/bash
[sachin@localhost home]$
```

2. Create Group dac , dbda ,ditiss

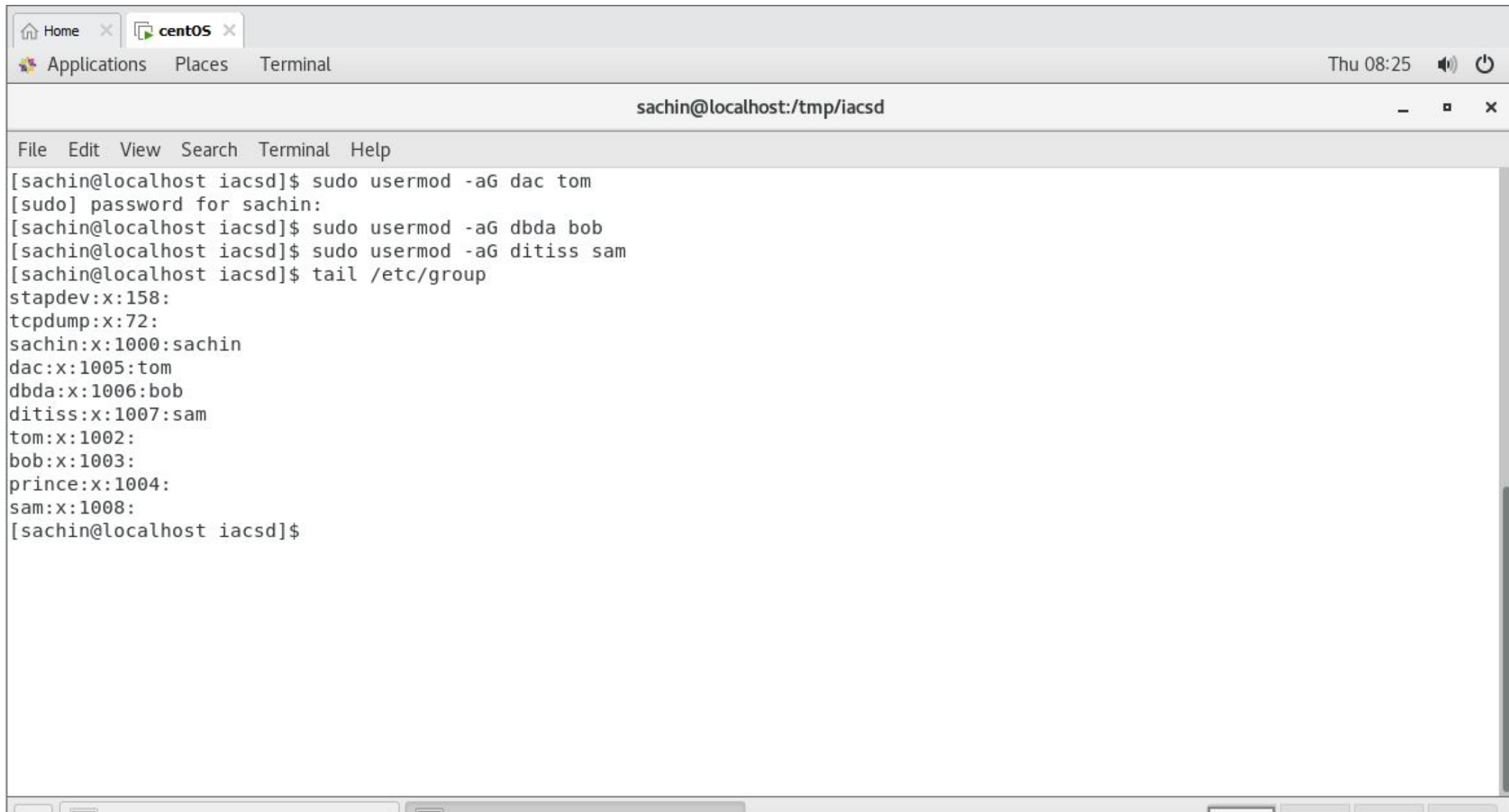


The screenshot shows a CentOS desktop environment with a terminal window open. The terminal displays the following commands and output:

```
[sachin@localhost home]$ cd /home
[sachin@localhost home]$ ls
bob prince sachin sam tom
[sachin@localhost home]$ tail /etc/group
stapdev:x:158:
tcpdump:x:72:
sachin:x:1000:sachin
dac:x:1005:
dbda:x:1006:
ditiss:x:1007:
tom:x:1002:
bob:x:1003:
prince:x:1004:
sam:x:1008:
[sachin@localhost home]$
```

The terminal window has a title bar with 'Home' and 'centOS' tabs, and a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The status bar shows 'Thu 08:10' and system icons.

3. add user Tom in dac, Bob in dbda, Sam in ditiss



A terminal window titled "centOS" with tabs for "Home", "Applications", "Places", and "Terminal". The window shows a user named "sachin" at "localhost" in the directory "/tmp/iacsd". The user has executed several commands to create users and view group information. The output shows the creation of users "tom", "bob", and "sam" with specific group IDs, followed by the tail of the /etc/group file listing various system and regular users.

```
sachin@localhost:/tmp/iacsd

File Edit View Search Terminal Help

[sachin@localhost iacsd]$ sudo usermod -aG dac tom
[sudo] password for sachin:
[sachin@localhost iacsd]$ sudo usermod -aG dbda bob
[sachin@localhost iacsd]$ sudo usermod -aG ditiss sam
[sachin@localhost iacsd]$ tail /etc/group
stapdev:x:158:
tcpdump:x:72:
sachin:x:1000:sachin
dac:x:1005:tom
dbda:x:1006:bob
ditiss:x:1007:sam
tom:x:1002:
bob:x:1003:
prince:x:1004:
sam:x:1008:
[sachin@localhost iacsd]$
```

4. login as prince and create iacsd directory in /tmp and create 4 files in iacsd with name project-1 project-2 upto 4



The image shows a terminal window titled 'centOS' with a menu bar containing 'Applications', 'Places', and 'Terminal'. The status bar at the top right shows 'Thu 08:26' and system icons. The terminal title bar indicates the user is 'prince@localhost' in the directory '/tmp/iacsd'. The terminal content shows the following sequence of commands and outputs:

```
File Edit View Search Terminal Help
[prince@localhost tmp]$ cd /home
[prince@localhost home]$ ls
bob prince sachin sam tom
[prince@localhost home]$ cd ..
[prince@localhost /]$ ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
[prince@localhost /]$ cd tmp
[prince@localhost tmp]$ cd iacsd
[prince@localhost iacsd]$ ls
project-1 project-2 project-3 project-4
[prince@localhost iacsd]$
```

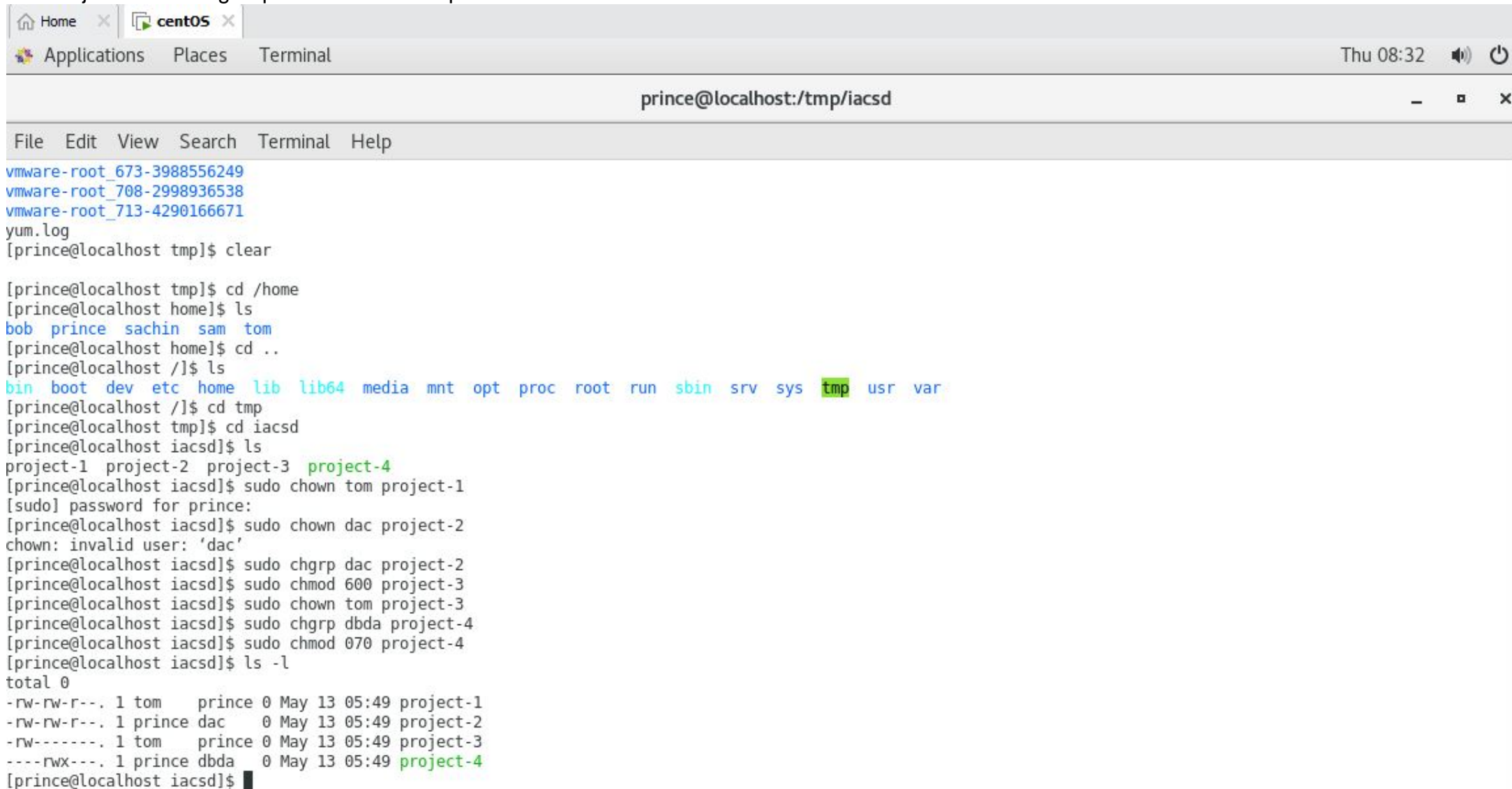
5. assign permissions to project files as below

Project-1 – tom should be owner of this

Project-2 – dac should be owner of this

Project-3 --- others should not have any permission but tom should have rw access

Project-4 – dbda group should have rwx permissions.



```
vmware-root_673-3988556249
vmware-root_708-2998936538
vmware-root_713-4290166671
yum.log
[prince@localhost tmp]$ clear

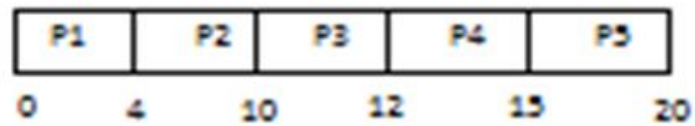
[prince@localhost tmp]$ cd /home
[prince@localhost home]$ ls
bob prince sachin sam tom
[prince@localhost home]$ cd ..
[prince@localhost /]$ ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
[prince@localhost /]$ cd tmp
[prince@localhost tmp]$ cd iacsd
[prince@localhost iacsd]$ ls
project-1 project-2 project-3 project-4
[prince@localhost iacsd]$ sudo chown tom project-1
[sudo] password for prince:
[prince@localhost iacsd]$ sudo chown dac project-2
chown: invalid user: 'dac'
[prince@localhost iacsd]$ sudo chgrp dac project-2
[prince@localhost iacsd]$ sudo chmod 600 project-3
[prince@localhost iacsd]$ sudo chown tom project-3
[prince@localhost iacsd]$ sudo chgrp dbda project-4
[prince@localhost iacsd]$ sudo chmod 070 project-4
[prince@localhost iacsd]$ ls -l
total 0
-rw-rw-r--. 1 tom   prince 0 May 13 05:49 project-1
-rw-rw-r--. 1 prince dac    0 May 13 05:49 project-2
-rw-----. 1 tom   prince 0 May 13 05:49 project-3
----rwx---. 1 prince dbda  0 May 13 05:49 project-4
[prince@localhost iacsd]$
```

Asssignment-3

1. FCFS SCHEDULING-

Process Id	Arrival time	Burst time
P1	0	4
P2	2	6
P3	3	2
P4	6	1
P5	4	3

Gnatt Chart:-

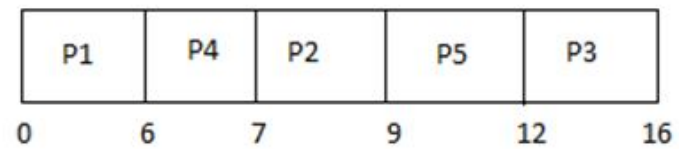


$$\begin{aligned}\text{Average waiting time} &= (0+2+7+9+8)/5 \\ &= 26/5 = 5.2\end{aligned}$$

2. SJF non-preemptive

Process Id	Arrival time	Burst time
P1	0	6
P2	3	2
P3	4	4
P4	5	1
P5	2	3

Gantt Chart:-

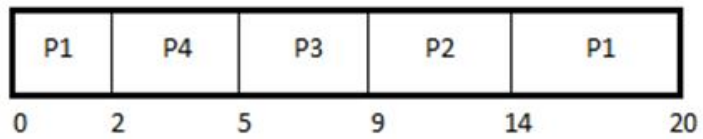


$$\begin{aligned}\text{Average waiting time} &= (0+4+8+1+7)/5 \\ &= 20/4 = 4\end{aligned}$$

3) SJF preemptive -SRTF

Process Id	Arrival time	Burst time
P1	4	6
P2	3	5
P3	2	4
P4	1	3
P5	0	2

Gantt Chart:-



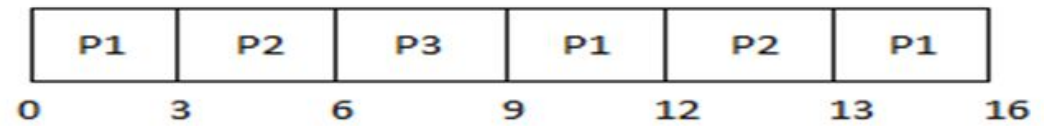
$$\begin{aligned}\text{Average waiting time} &= (14-4) + (9-3) + (5-2) + (2-1) + 0 / 5 \\ &= (10+6+3+1+0) / 5 \\ &= 20 / 5 = 4\end{aligned}$$

4) Round Robin

Q=3

Process Id	Arrival time	Burst time
P1	0	9
P2	1	4
P3	2	3

Gantt Chart:-

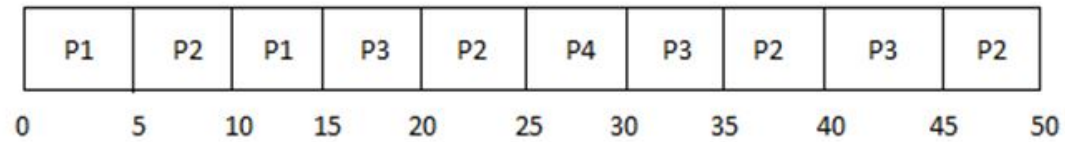


$$\begin{aligned}\text{Average waiting time} &= (13-6) + (12-1-3) + (6-2) / 3 \\ &= (7+8+4) / 3 = 19/3 = 6.33\end{aligned}$$

$Q = 5$

Process Id	Arrival time	Burst time
P1	0	10
P2	5	20
P3	10	15
P4	15	5

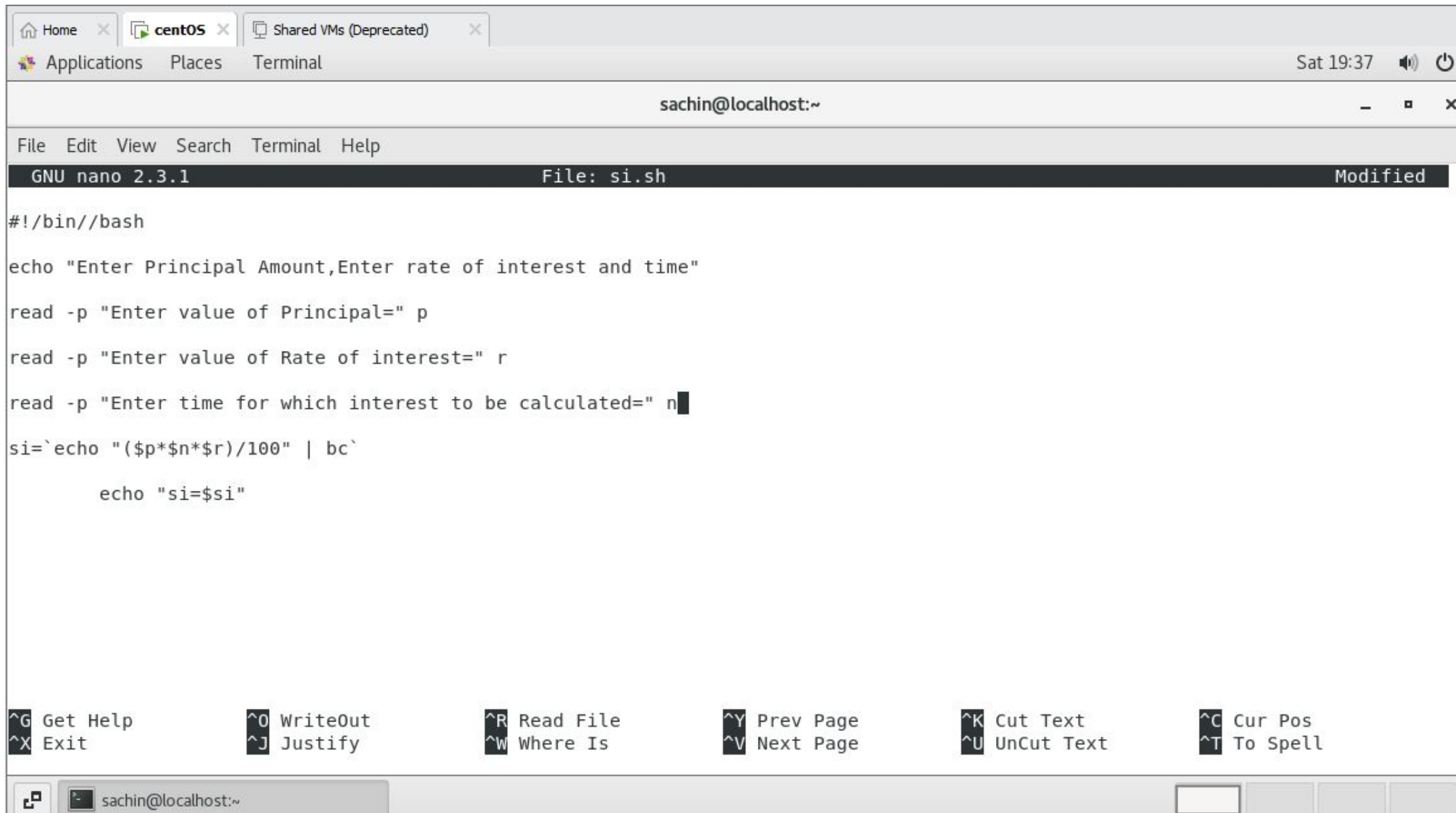
Gnatt Chart:-



$$\begin{aligned}\text{Average waiting time} &= (10-5-0) + (45-15-5) + (40-10-10) + (25-15) \\ &= 60/4 \\ &= 15\end{aligned}$$

Assignment-4

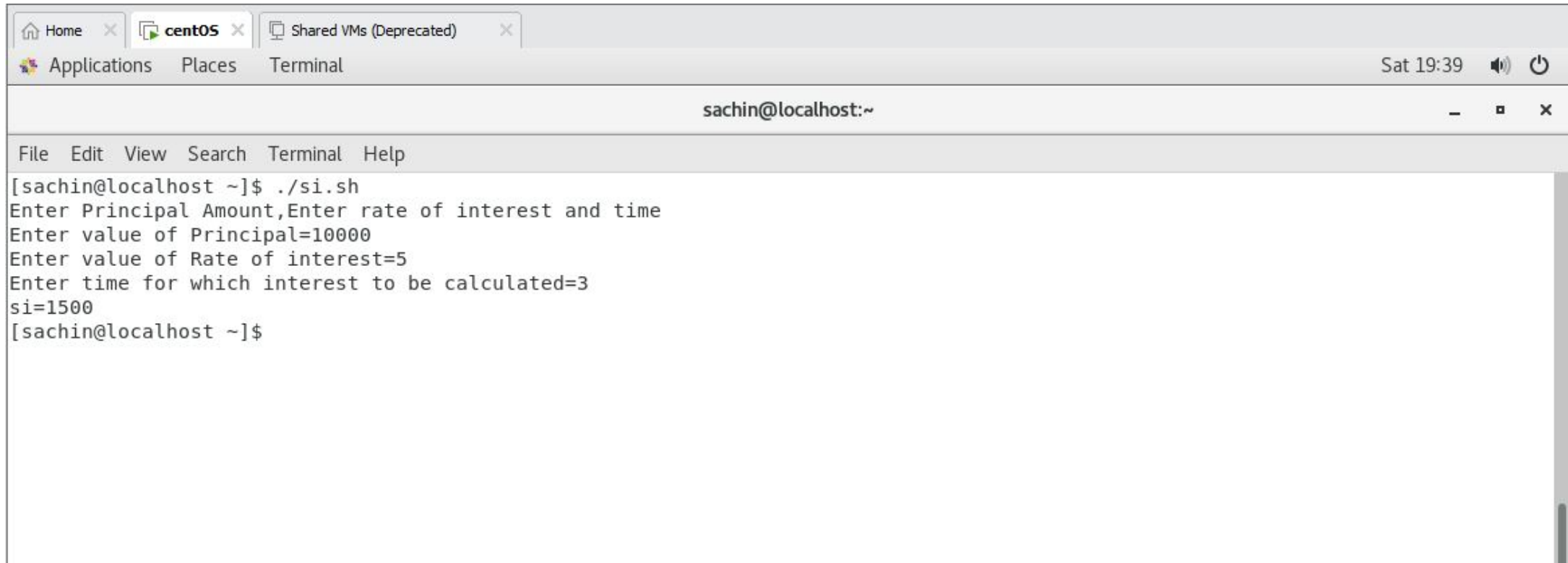
1) Write a shell script to calculate simple interest.



The screenshot shows a terminal window with a menu bar (File, Edit, View, Search, Terminal, Help) and a title bar (sachin@localhost:~). The terminal is running the GNU nano 2.3.1 editor, editing a file named si.sh. The script content is as follows:

```
#!/bin//bash
echo "Enter Principal Amount,Enter rate of interest and time"
read -p "Enter value of Principal=" p
read -p "Enter value of Rate of interest=" r
read -p "Enter time for which interest to be calculated=" n
si=`echo "($p*$n*$r)/100" | bc`
    echo "si=$si"
```

The terminal window also displays a status bar at the bottom with various keyboard shortcuts for nano editor operations, such as ^G Get Help, ^X Exit, ^O WriteOut, ^J Justify, ^R Read File, ^W Where Is, ^Y Prev Page, ^V Next Page, ^K Cut Text, ^U UnCut Text, ^C Cur Pos, and ^T To Spell.



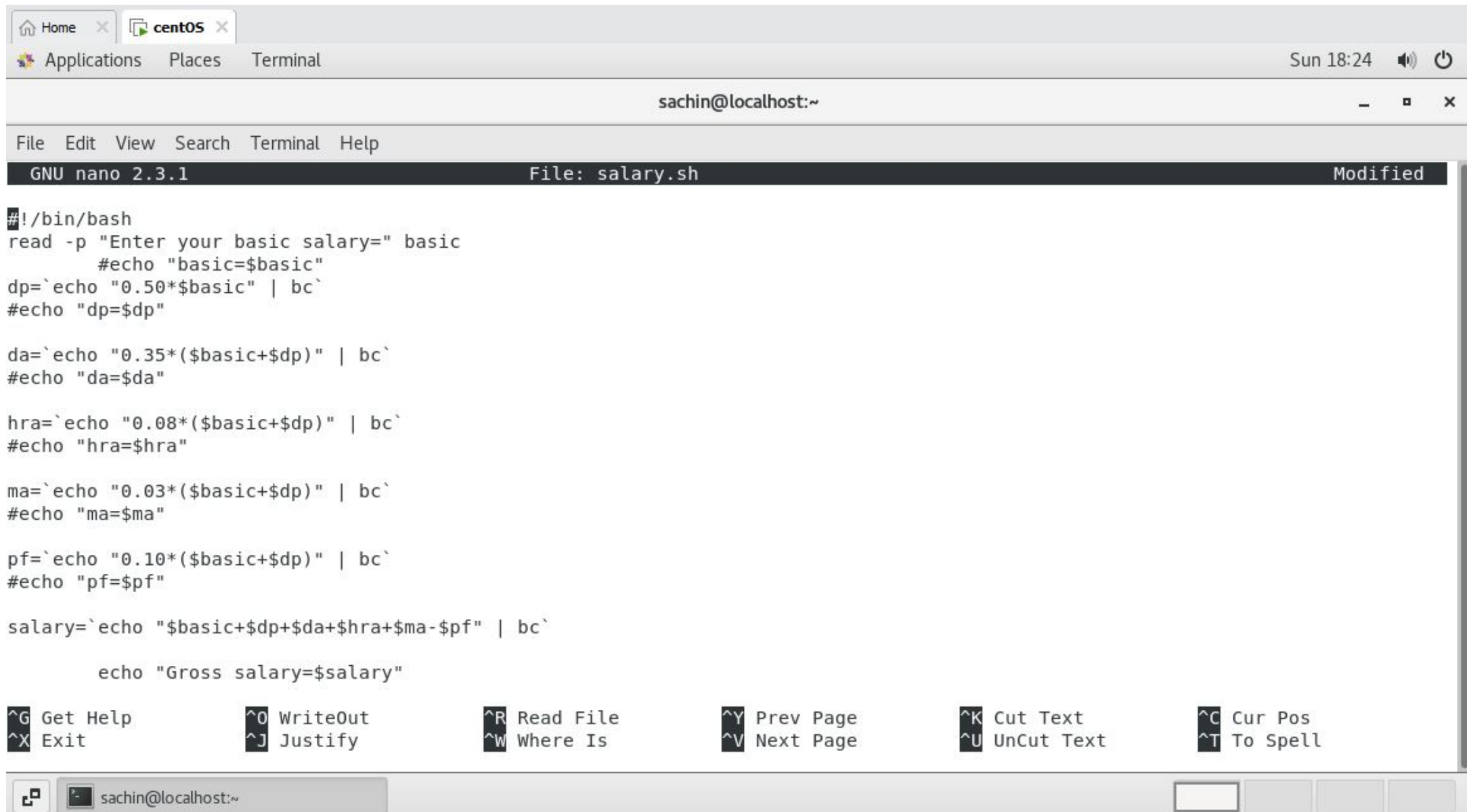
The screenshot shows a Linux desktop environment. At the top, there is a panel with three window tabs: 'Home', 'centOS', and 'Shared VMs (Deprecated)'. Below this is a menu bar with 'Applications', 'Places', and 'Terminal'. The system clock on the right indicates 'Sat 19:39'. A terminal window is open, displaying the prompt 'sachin@localhost:~'. The terminal has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal content shows the execution of a script './si.sh' which prompts for principal, rate, and time, and then outputs the calculated simple interest 'si=1500'.

```
[sachin@localhost ~]$ ./si.sh
Enter Principal Amount,Enter rate of interest and time
Enter value of Principal=10000
Enter value of Rate of interest=5
Enter time for which interest to be calculated=3
si=1500
[sachin@localhost ~]$
```

2) Write a shell script to calculate salary from given basic.

Salary = basic + dp + da +hra +ma –pf, basic – to be taken as input, dp - 50 % of basic, da - 35 % of (basic + dp)

hra - 8 % of (basic + dp), ma - 3 % of (basic + dp), pf - 10% of (basic + dp).



The screenshot shows a Linux terminal window with a window manager interface. The title bar includes tabs for 'Home' and 'centOS', and buttons for 'Applications', 'Places', and 'Terminal'. The system clock shows 'Sun 18:24'. The terminal prompt is 'sachin@localhost:~'. The terminal content shows the GNU nano 2.3.1 editor editing a file named 'salary.sh'. The script calculates the gross salary based on a basic salary input, with deductions for dp (50%), da (35%), hra (8%), ma (3%), and pf (10%). The bottom status bar of the nano editor displays various keyboard shortcuts for navigation and editing.

```
#!/bin/bash
read -p "Enter your basic salary=" basic
    #echo "basic=$basic"
dp=`echo "0.50*$basic" | bc`
#echo "dp=$dp"

da=`echo "0.35*($basic+$dp)" | bc`
#echo "da=$da"

hra=`echo "0.08*($basic+$dp)" | bc`
#echo "hra=$hra"

ma=`echo "0.03*($basic+$dp)" | bc`
#echo "ma=$ma"

pf=`echo "0.10*($basic+$dp)" | bc`
#echo "pf=$pf"

salary=`echo "$basic+$dp+$da+$hra+$ma-$pf" | bc`

    echo "Gross salary=$salary"
```

GNU nano 2.3.1 File: salary.sh Modified

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

Home × CentOS ×

Applications Places Terminal

Sun 18:23 🔊 🔌

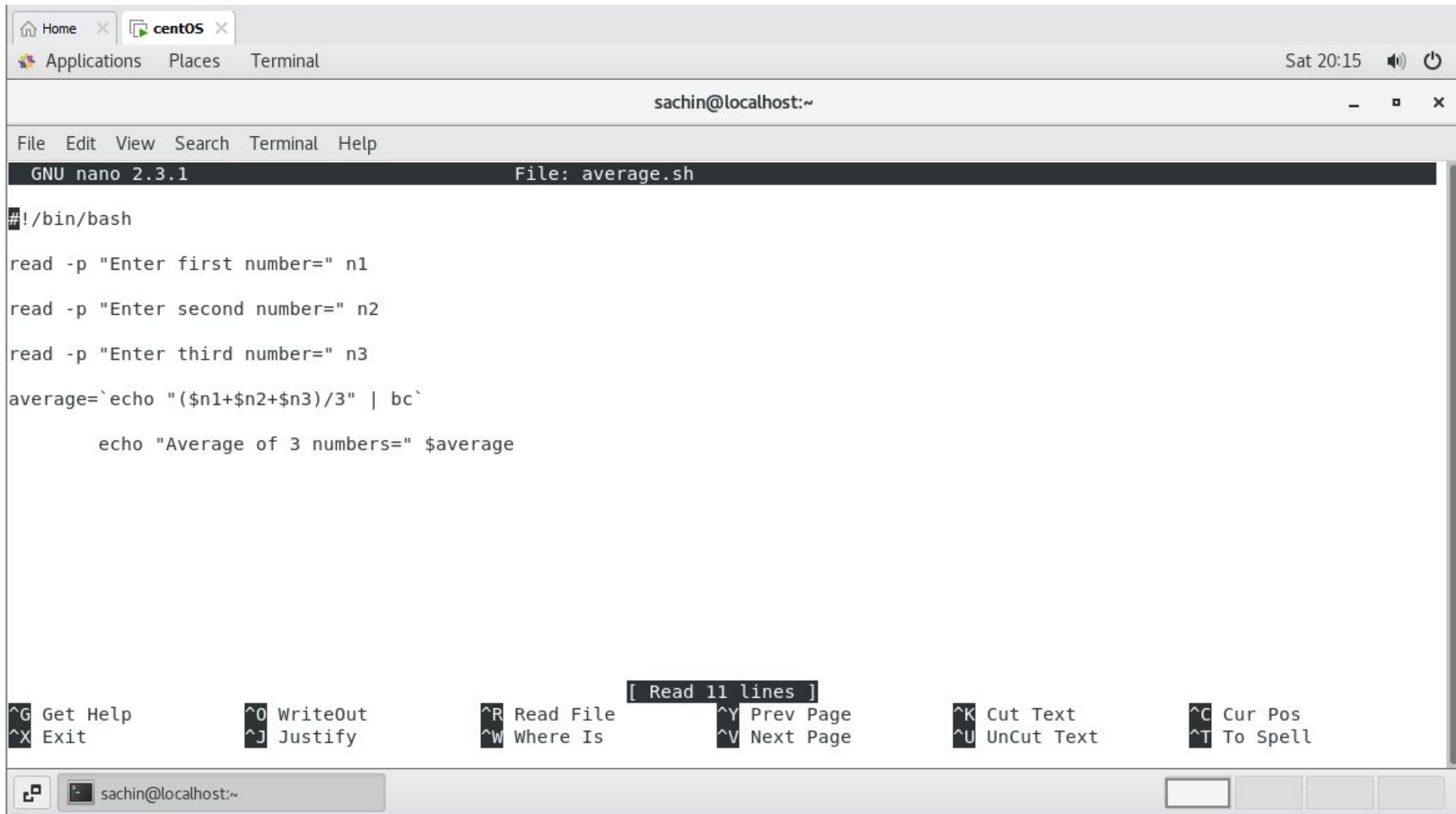
sachin@localhost:~

— □ ×

File Edit View Search Terminal Help

```
[sachin@localhost ~]$ ./salary.sh
Enter your basic salary=10000
Gross salary=20400.00
[sachin@localhost ~]$ █
```


3) Write a shell script to calculate the average of a 3 number.



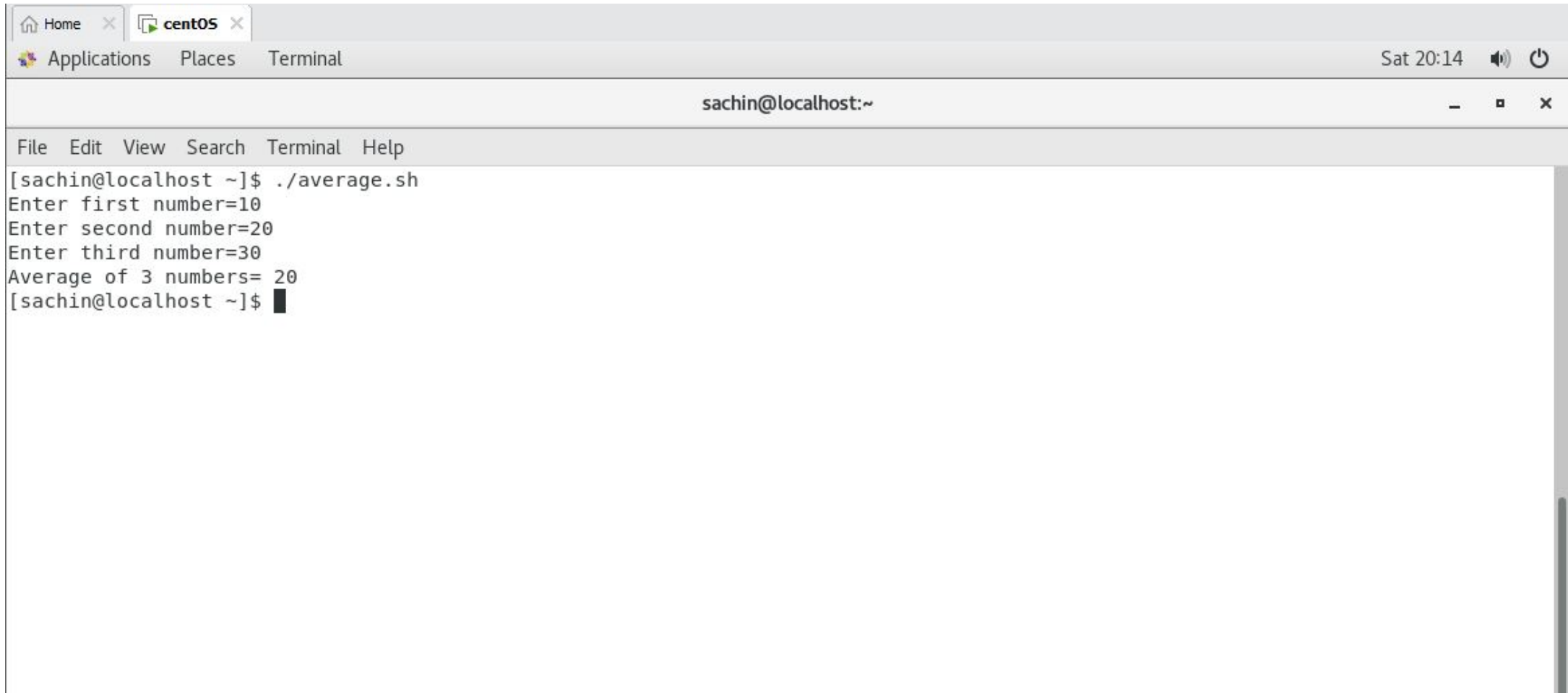
The screenshot shows a terminal window with a nano editor open, editing a file named 'average.sh'. The terminal title bar indicates the user is 'sachin@localhost' and the time is 'Sat 20:15'. The nano editor's status bar at the top shows 'GNU nano 2.3.1' and 'File: average.sh'. The script content is as follows:

```
#!/bin/bash
read -p "Enter first number=" n1
read -p "Enter second number=" n2
read -p "Enter third number=" n3
average=`echo "($n1+$n2+$n3)/3" | bc`
    echo "Average of 3 numbers=" $average
```

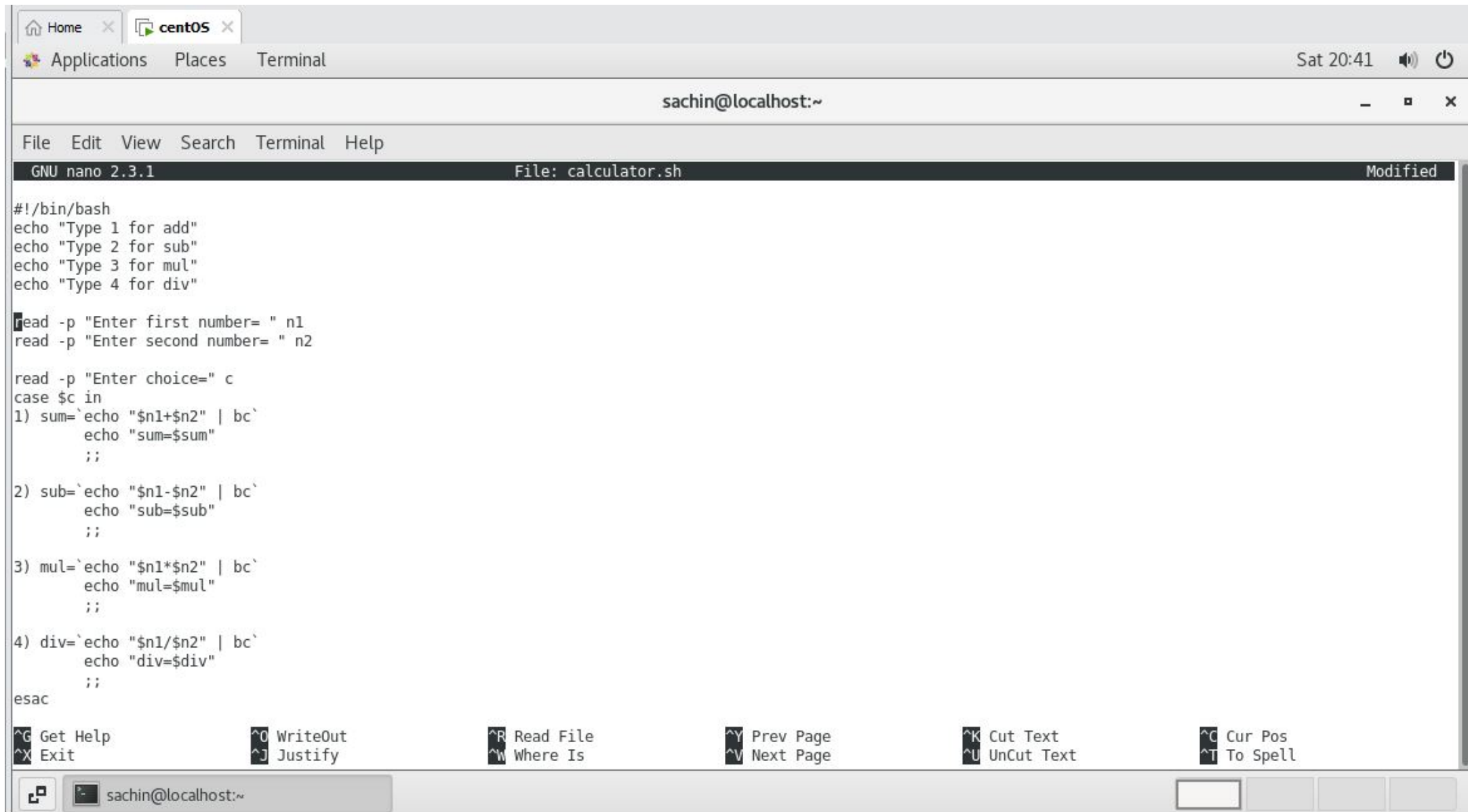
The bottom of the nano editor displays a list of keyboard shortcuts:

^G Get Help	^O WriteOut	^R Read File	^Y Prev Page	^K Cut Text	^C Cur Pos
^X Exit	^J Justify	^W Where Is	^V Next Page	^U UnCut Text	^T To Spell

A status bar at the very bottom of the terminal shows the user 'sachin@localhost' and a cursor icon.



4). Write a shell script to create a command line calculator.
e.g. input : mycal 5 + 5 Result : 10 , input : mycal 5 / 5 result : 1

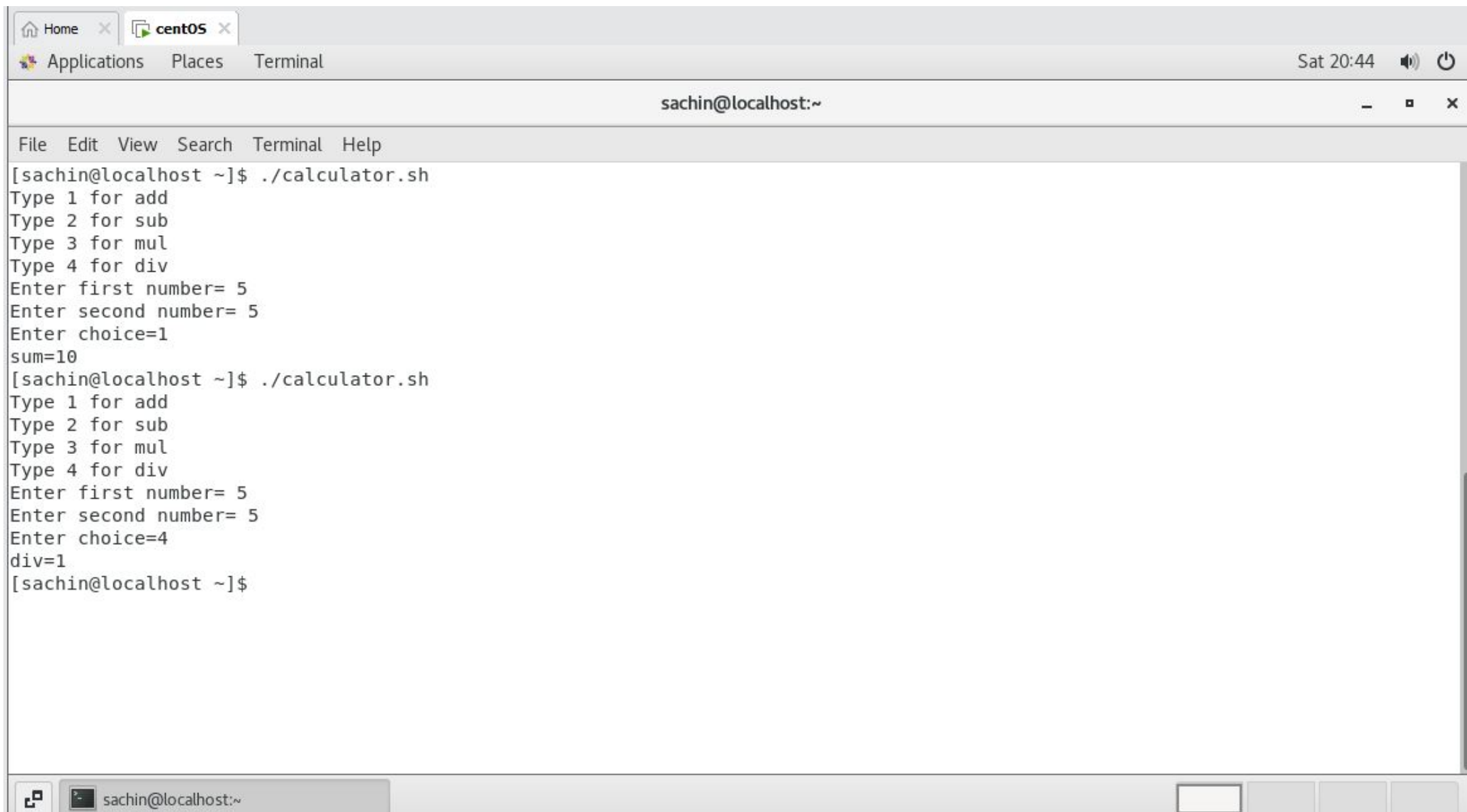


The screenshot shows a terminal window titled 'centOS' with a menu bar containing 'Applications', 'Places', and 'Terminal'. The status bar at the top right shows 'Sat 20:41' and system icons. The terminal content shows a nano editor editing a file named 'calculator.sh'. The script is a shell script that prompts the user for two numbers and a choice of operation (add, subtract, multiply, or divide). It uses 'bc' for calculations and 'echo' to display the results. The nano editor's status bar at the bottom shows 'GNU nano 2.3.1' and 'File: calculator.sh Modified'. The bottom of the terminal window has a command prompt 'sachin@localhost:~' and a row of keyboard shortcuts: ^G Get Help, ^O WriteOut, ^R Read File, ^Y Prev Page, ^K Cut Text, ^C Cur Pos; ^X Exit, ^J Justify, ^W Where Is, ^V Next Page, ^U UnCut Text, ^T To Spell.

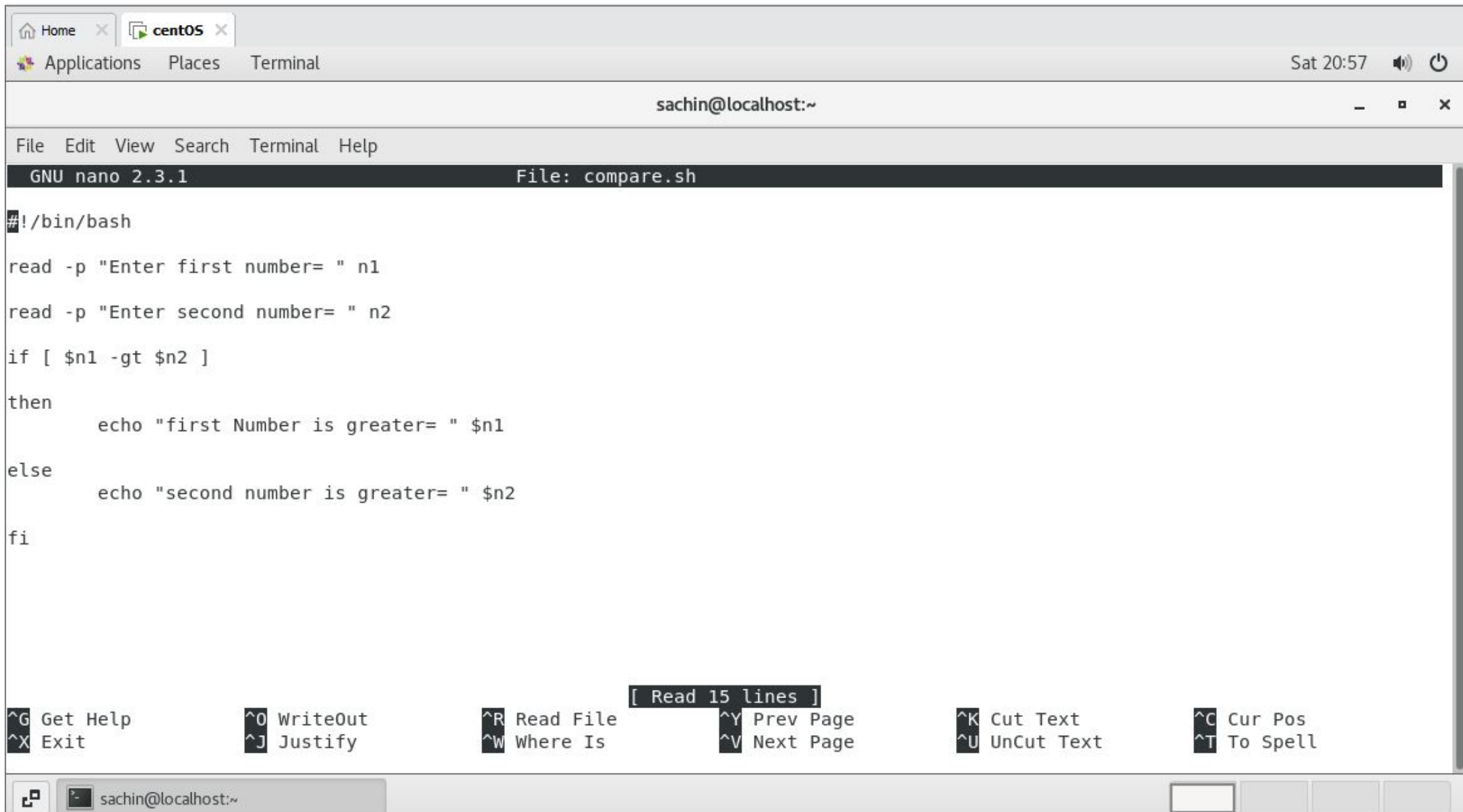
```
#!/bin/bash
echo "Type 1 for add"
echo "Type 2 for sub"
echo "Type 3 for mul"
echo "Type 4 for div"

read -p "Enter first number= " n1
read -p "Enter second number= " n2

read -p "Enter choice=" c
case $c in
1) sum=`echo "$n1+$n2" | bc`
   echo "sum=$sum"
   ;;
2) sub=`echo "$n1-$n2" | bc`
   echo "sub=$sub"
   ;;
3) mul=`echo "$n1*$n2" | bc`
   echo "mul=$mul"
   ;;
4) div=`echo "$n1/$n2" | bc`
   echo "div=$div"
   ;;
esac
```



5) Write a shell script to accept 2 numbers and display which number is greater



The screenshot shows a Linux desktop environment with a window titled 'centOS'. Inside the window is a terminal application. The terminal's title bar shows 'sachin@localhost:~'. The terminal content shows the GNU nano 2.3.1 editor editing a file named 'compare.sh'. The script is a shell script that takes two numbers as input and prints the greater one. The script content is as follows:

```
#!/bin/bash
read -p "Enter first number= " n1
read -p "Enter second number= " n2
if [ $n1 -gt $n2 ]
then
    echo "first Number is greater= " $n1
else
    echo "second number is greater= " $n2
fi
```

At the bottom of the terminal window, there is a status bar with various keyboard shortcuts for nano editor operations:

^G Get Help	^O WriteOut	^R Read File	^Y Prev Page	^K Cut Text	^C Cur Pos
^X Exit	^J Justify	^W Where Is	^V Next Page	^U UnCut Text	^T To Spell

The terminal window also shows a top bar with 'Home', 'Applications', 'Places', and 'Terminal' buttons, and a system status bar on the right showing 'Sat 20:57' and system icons.

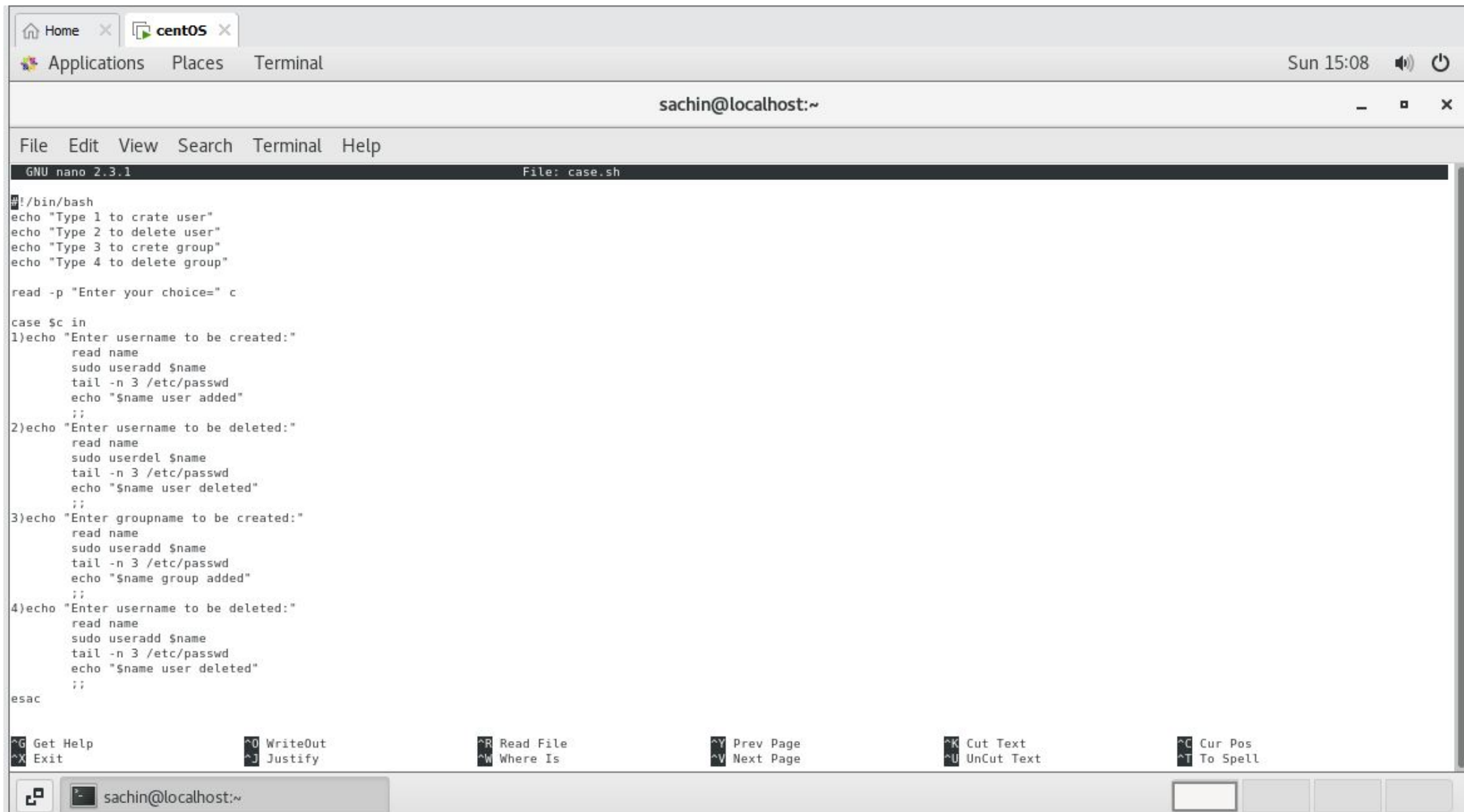
sachin@localhost:~

File Edit View Search Terminal Help

```
[sachin@localhost ~]$ ./compare.sh
Enter first number= 20
Enter second number= 10
first Number is greater= 20
[sachin@localhost ~]$
```


6) Create a script to

Create user , Delete user , Create group , delete Group using case



```
GNU nano 2.3.1 File: case.sh

#!/bin/bash
echo "Type 1 to crate user"
echo "Type 2 to delete user"
echo "Type 3 to crete group"
echo "Type 4 to delete group"

read -p "Enter your choice=" c

case $c in
1)echo "Enter username to be created:"
  read name
  sudo useradd $name
  tail -n 3 /etc/passwd
  echo "$name user added"
  ;;
2)echo "Enter username to be deleted:"
  read name
  sudo userdel $name
  tail -n 3 /etc/passwd
  echo "$name user deleted"
  ;;
3)echo "Enter groupname to be created:"
  read name
  sudo useradd $name
  tail -n 3 /etc/passwd
  echo "$name group added"
  ;;
4)echo "Enter username to be deleted:"
  read name
  sudo useradd $name
  tail -n 3 /etc/passwd
  echo "$name user deleted"
  ;;
esac

^G Get Help      ^O WriteOut      ^R Read File     ^Y Prev Page     ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify        ^W Where Is      ^V Next Page     ^U UnCut Text    ^T To Spell
```

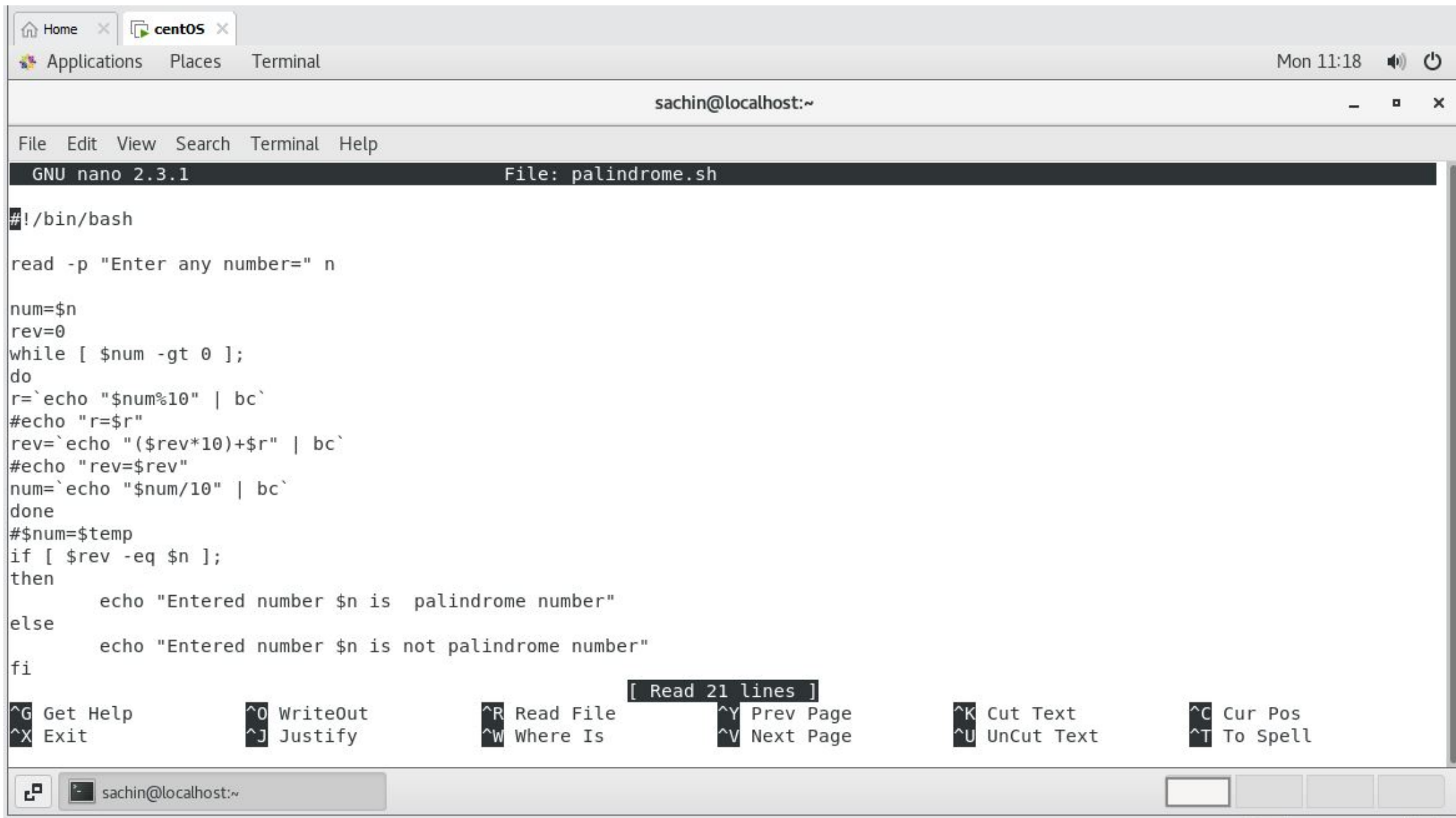
sachin@localhost:~

File Edit View Search Terminal Help

```
[sachin@localhost ~]$ nano case.sh
[sachin@localhost ~]$ ./case.sh
Type 1 to crate user
Type 2 to delete user
Type 3 to crete group
Type 4 to delete group
Enter your choice=3
Enter groupname to be created:
king
[sudo] password for sachin:
prince:x:1004:1004::/home/prince:/bin/bash
sam:x:1005:1008::/home/sam:/bin/bash
king:x:1006:1009::/home/king:/bin/bash
king group added
[sachin@localhost ~]$ ./case.sh
Type 1 to crate user
Type 2 to delete user
Type 3 to crete group
Type 4 to delete group
Enter your choice=4
Enter username to be deleted:
king
useradd: user 'king' already exists
prince:x:1004:1004::/home/prince:/bin/bash
sam:x:1005:1008::/home/sam:/bin/bash
king:x:1006:1009::/home/king:/bin/bash
king user deleted
[sachin@localhost ~]$
```

Assignment-5

1) Write a script to find out String is palindrome or not.



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window has a title bar with 'Home', 'centOS', and 'Terminal' tabs. The status bar shows 'Mon 11:18' and system icons. The terminal content shows a shell script being edited in nano. The script checks if a number is a palindrome by reversing its digits. The script is as follows:

```
#!/bin/bash
read -p "Enter any number=" n
num=$n
rev=0
while [ $num -gt 0 ];
do
r=`echo "$num%10" | bc`
#echo "r=$r"
rev=`echo "($rev*10)+$r" | bc`
#echo "rev=$rev"
num=`echo "$num/10" | bc`
done
#$num=$temp
if [ $rev -eq $n ];
then
    echo "Entered number $n is  palindrome number"
else
    echo "Entered number $n is not palindrome number"
fi
```

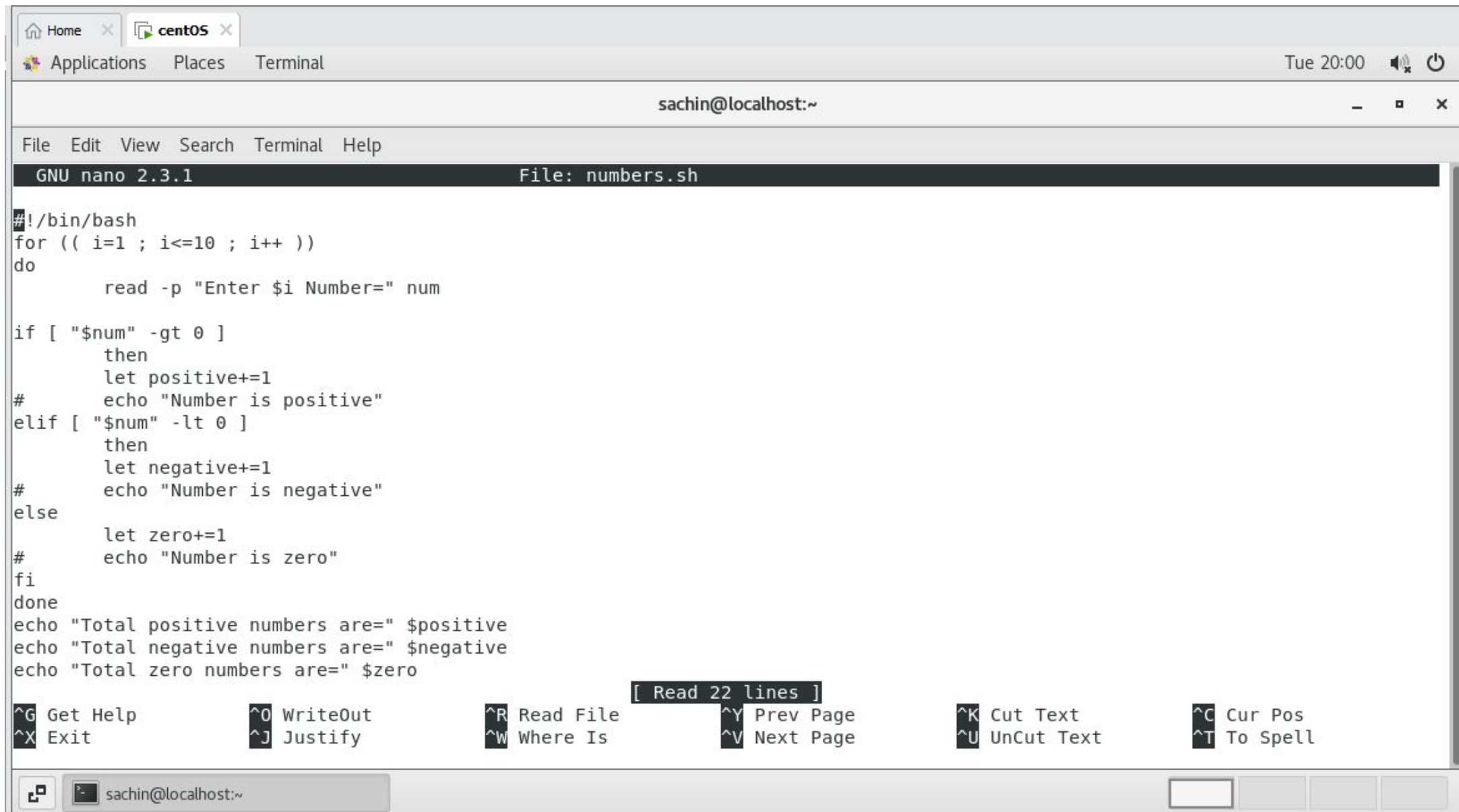
At the bottom of the terminal, there is a status bar with various shortcuts: **^G** Get Help, **^O** WriteOut, **^R** Read File, **^Y** Prev Page, **^K** Cut Text, **^C** Cur Pos, **^X** Exit, **^J** Justify, **^W** Where Is, **^V** Next Page, **^U** UnCut Text, **^T** To Spell. A message '[Read 21 lines]' is also visible.

sachin@localhost:~

File Edit View Search Terminal Help

```
[sachin@localhost ~]$ ./palindrome.sh
Enter any number=121
Entered number 121 is  palindrome number
[sachin@localhost ~]$ ./palindrome.sh
Enter any number=1234
Entered number 1234 is not palindrome number
[sachin@localhost ~]$
```

2) Write a shell script to accept 10 numbers and tell how many are +tive, -tive and zero.



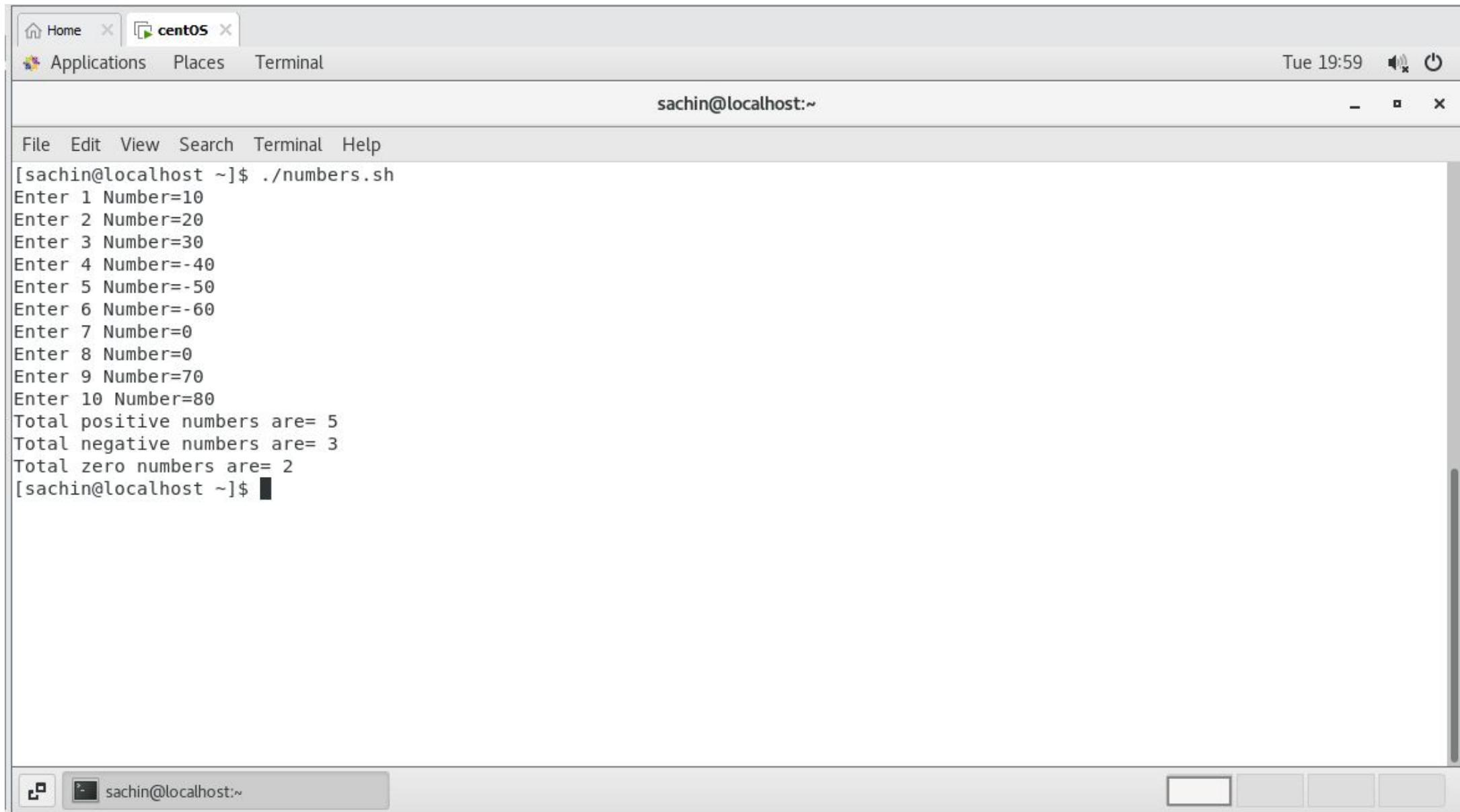
```
GNU nano 2.3.1 File: numbers.sh

#!/bin/bash
for (( i=1 ; i<=10 ; i++ ))
do
    read -p "Enter $i Number=" num

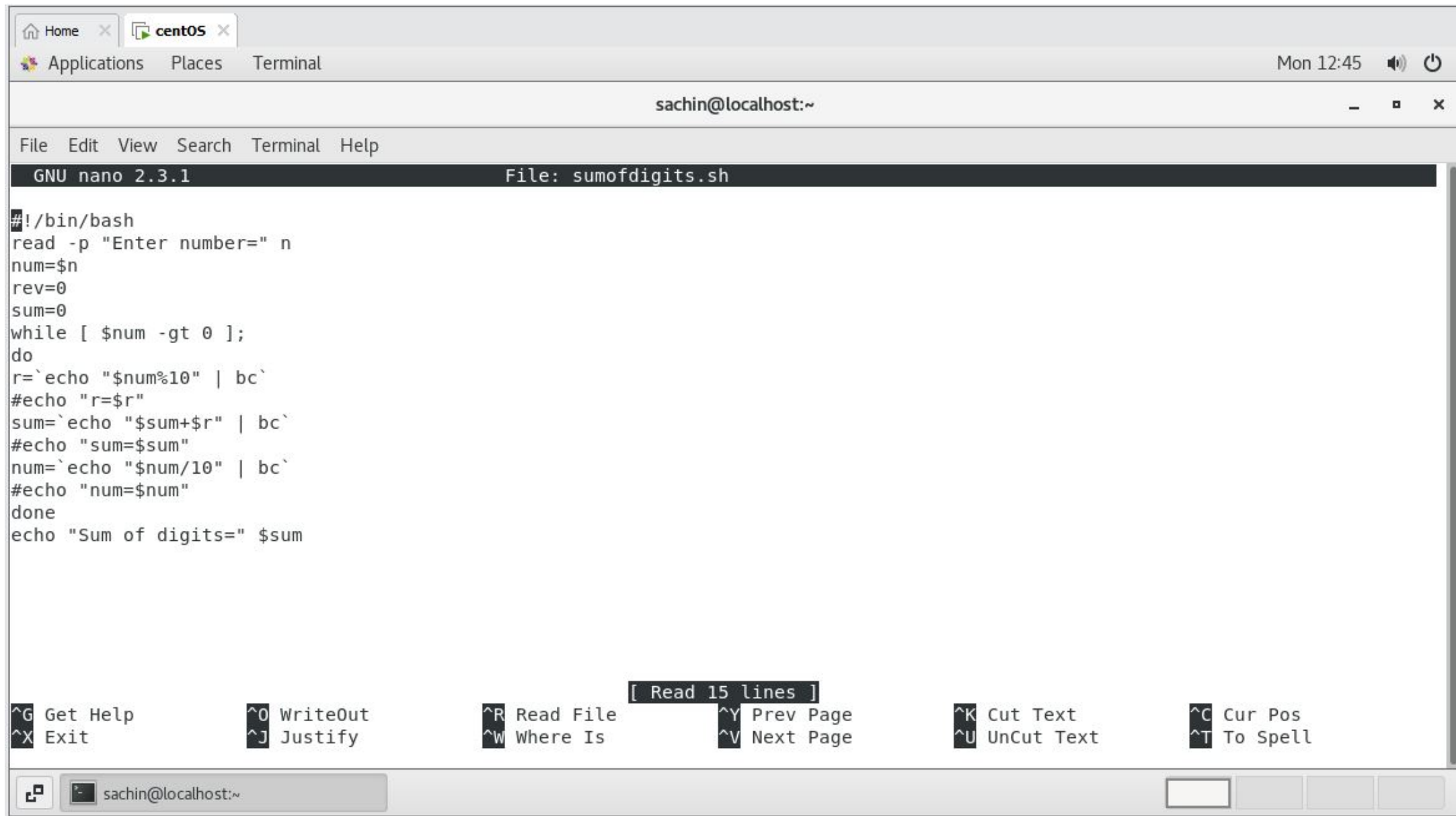
    if [ "$num" -gt 0 ]
    then
        let positive+=1
        echo "Number is positive"
    #
    elif [ "$num" -lt 0 ]
    then
        let negative+=1
        echo "Number is negative"
    #
    else
        let zero+=1
        echo "Number is zero"
    fi
done
echo "Total positive numbers are=" $positive
echo "Total negative numbers are=" $negative
echo "Total zero numbers are=" $zero

[ Read 22 lines ]

^G Get Help      ^O WriteOut      ^R Read File     ^Y Prev Page     ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify       ^W Where Is      ^V Next Page     ^U UnCut Text    ^T To Spell
```



3) Write a shell script to print given number's sum of all digits (eg. If number is 123, then it's sum of all digits will be $1+2+3=6$)



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window has a title bar with 'Home', 'centOS', and 'Terminal' tabs. The top right corner shows the date and time 'Mon 12:45'. The terminal prompt is 'sachin@localhost:~'. The terminal is running the GNU nano 2.3.1 editor, editing a file named 'sumofdigits.sh'. The script content is as follows:

```
#!/bin/bash
read -p "Enter number=" n
num=$n
rev=0
sum=0
while [ $num -gt 0 ];
do
r=`echo "$num%10" | bc`
#echo "r=$r"
sum=`echo "$sum+$r" | bc`
#echo "sum=$sum"
num=`echo "$num/10" | bc`
#echo "num=$num"
done
echo "Sum of digits=" $sum
```

At the bottom of the terminal window, there is a status bar with various keyboard shortcuts for navigation and editing:

^G Get Help	^O WriteOut	^R Read File	^Y Prev Page	^K Cut Text	^C Cur Pos
^X Exit	^J Justify	^W Where Is	^V Next Page	^U UnCut Text	^T To Spell

The status bar also includes a '[Read 15 lines]' indicator and a prompt 'sachin@localhost:~'.

sachin@localhost:~

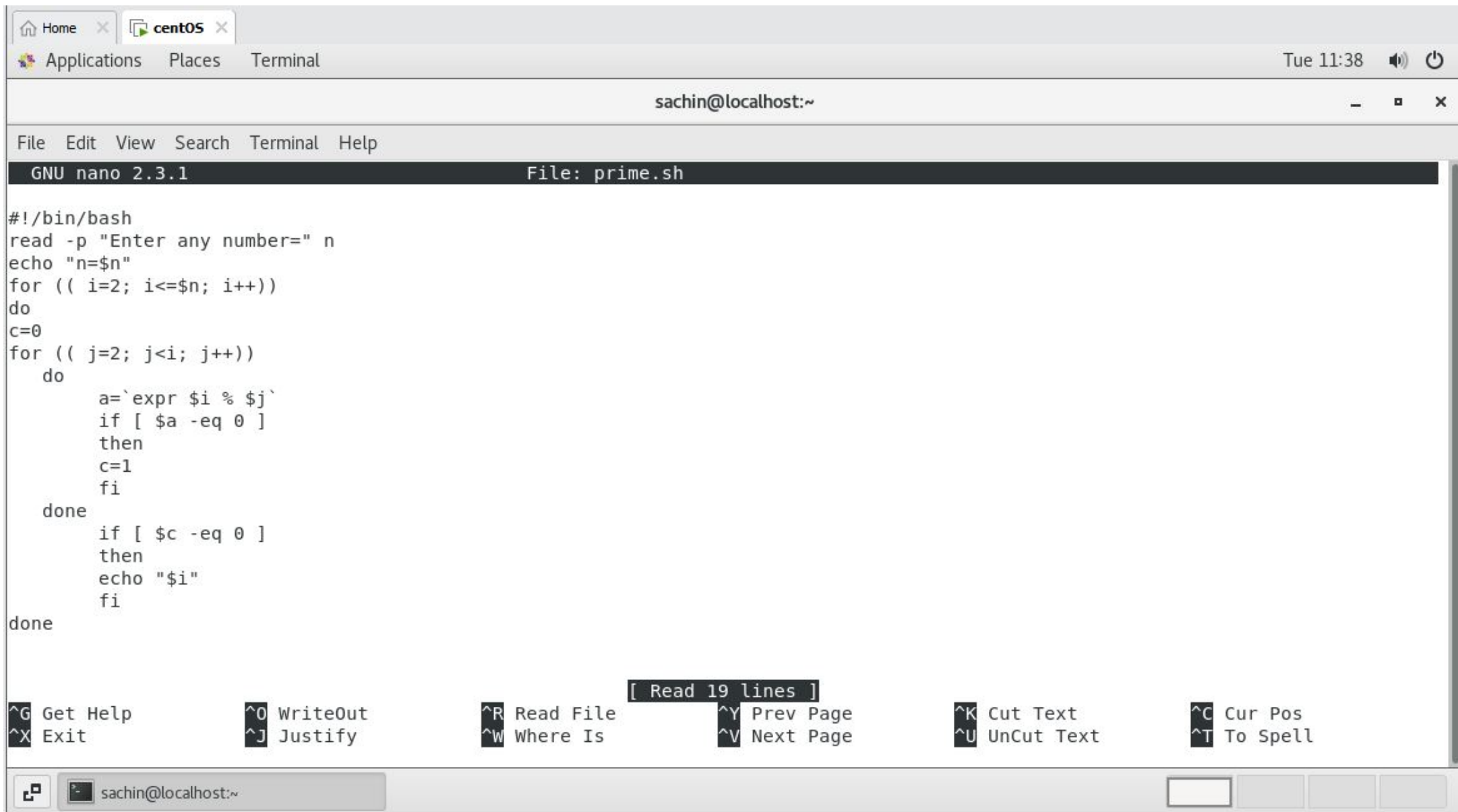
File Edit View Search Terminal Help

```
[sachin@localhost ~]$ ./sumofdigits.sh
Enter number=123
Sum of digits= 6
[sachin@localhost ~]$ ./sumofdigits.sh
Enter number=1224
Sum of digits= 9
[sachin@localhost ~]$
```



sachin@localhost:~

4)Write a shell script to display the prime numbers from 1 to n (n is a given number)



The screenshot shows a terminal window with a window manager at the top. The window has tabs for 'Home' and 'centOS'. Below the tabs are buttons for 'Applications', 'Places', and 'Terminal'. The system clock shows 'Tue 11:38'. The terminal title bar reads 'sachin@localhost:~'. The terminal content shows the GNU nano 2.3.1 editor editing a file named 'prime.sh'. The script is a bash script that takes a number 'n' as input and prints all prime numbers from 1 to 'n'. The script uses nested loops: an outer loop for 'i' from 2 to 'n', and an inner loop for 'j' from 2 to 'i-1'. It checks if 'i' is divisible by 'j'. If not, it increments a counter 'c'. After the inner loop, if 'c' is 0, it prints 'i'. The bottom of the terminal shows a status bar with various keyboard shortcuts for nano editor operations.

```
GNU nano 2.3.1 File: prime.sh

#!/bin/bash
read -p "Enter any number=" n
echo "n=$n"
for (( i=2; i<=$n; i++))
do
c=0
for (( j=2; j<i; j++))
do
a=`expr $i % $j`
if [ $a -eq 0 ]
then
c=1
fi
done
if [ $c -eq 0 ]
then
echo "$i"
fi
done
```

[Read 19 lines]

^G Get Help	^O WriteOut	^R Read File	^Y Prev Page	^K Cut Text	^C Cur Pos
^X Exit	^J Justify	^W Where Is	^V Next Page	^U UnCut Text	^T To Spell

sachin@localhost:~

File Edit View Search Terminal Help

```
[sachin@localhost ~]$ ./prime.sh
```

```
Enter any number=10
```

```
n=10
```

```
2
```

```
3
```

```
5
```

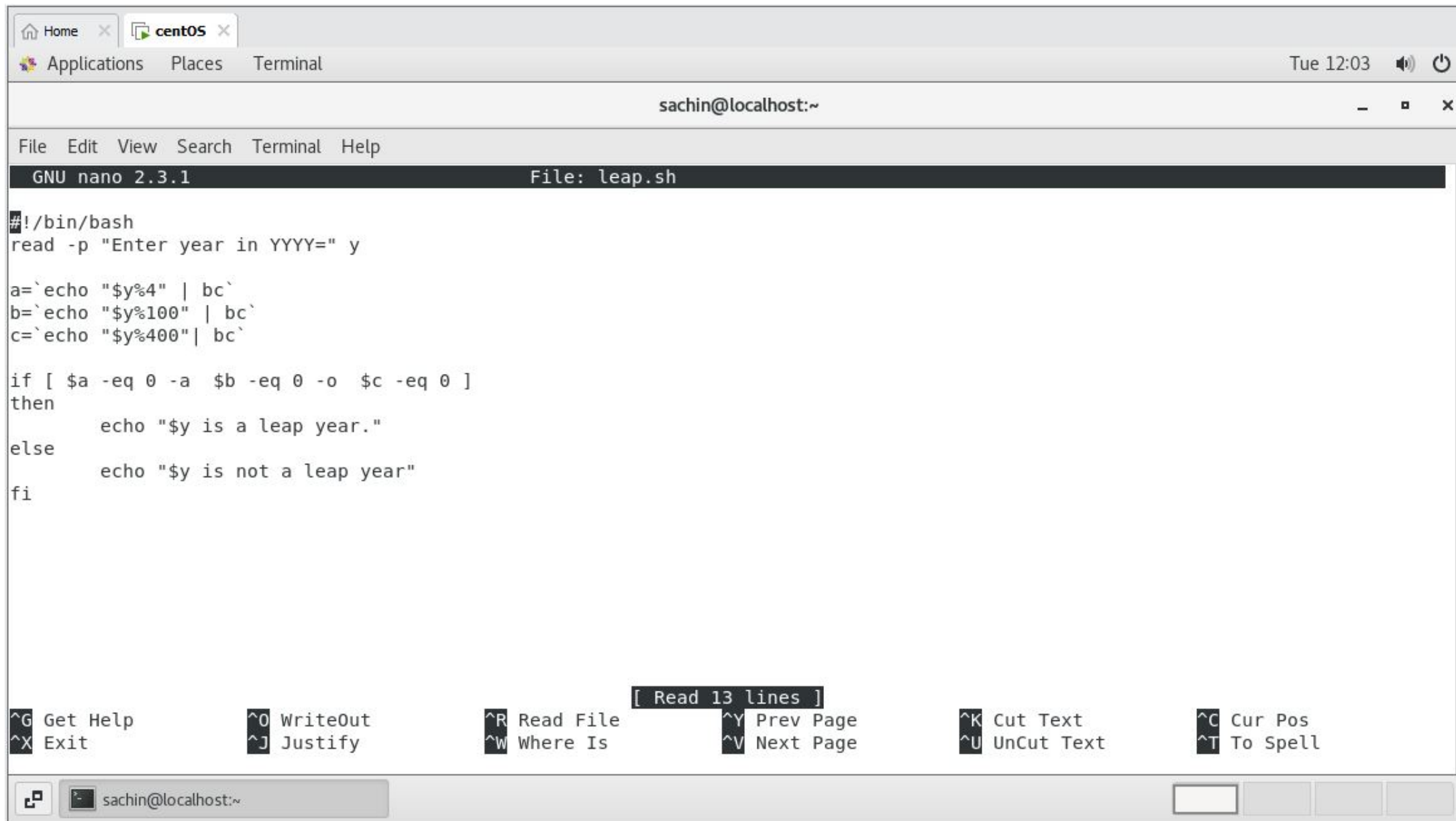
```
7
```

```
[sachin@localhost ~]$ █
```



sachin@localhost:~

5) Write a shell script to find whether a given year is leap year or not



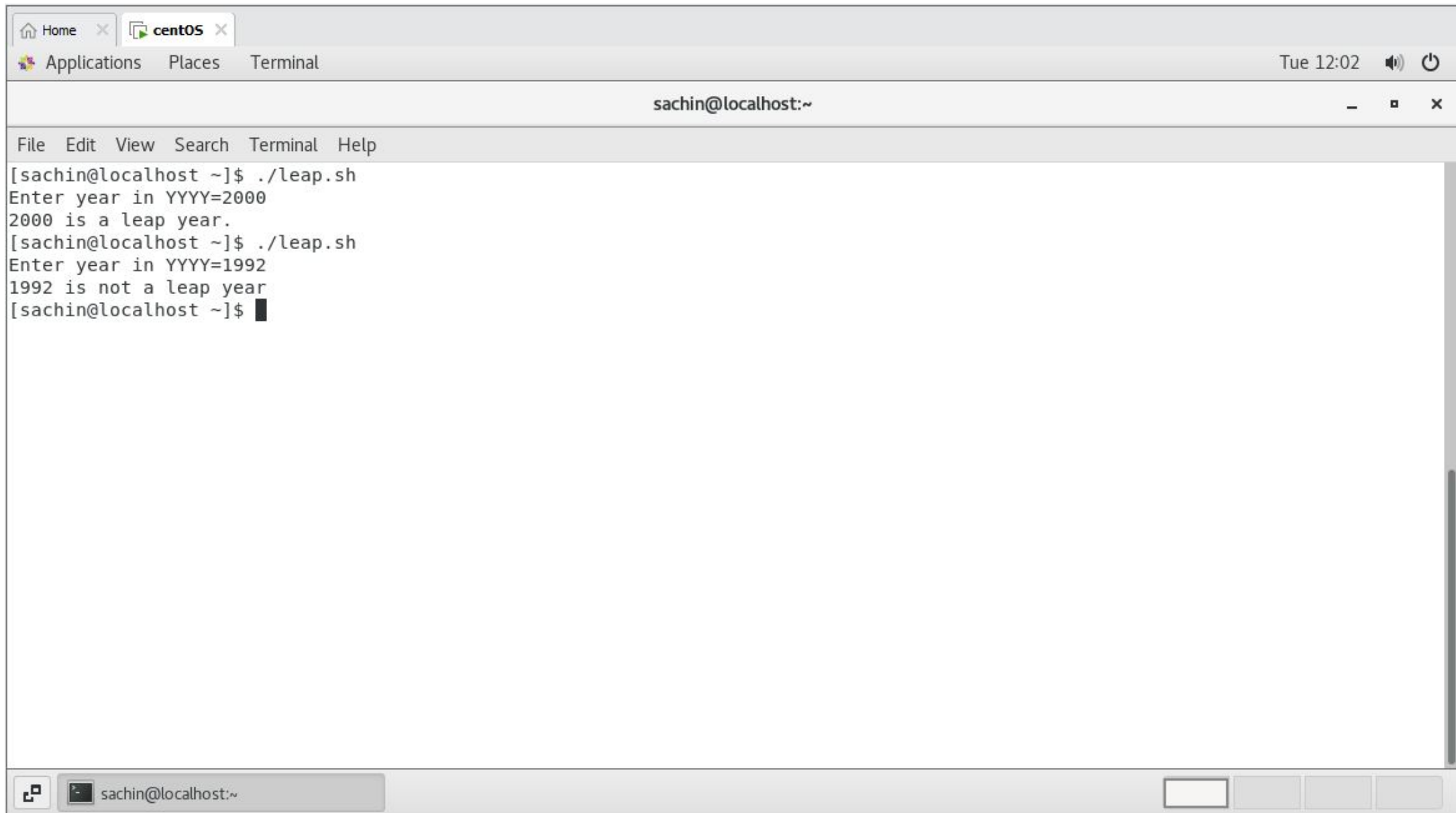
The screenshot shows a terminal window with a window manager at the top. The window has tabs for 'Home', 'centOS', and 'Terminal'. The title bar indicates the time is 'Tue 12:03'. The terminal itself has a title bar showing 'sachin@localhost:~'. Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The main content area shows the GNU nano 2.3.1 editor editing a file named 'leap.sh'. The script content is as follows:

```
#!/bin/bash
read -p "Enter year in YYYY=" y

a=`echo "$y%4" | bc`
b=`echo "$y%100" | bc`
c=`echo "$y%400" | bc`

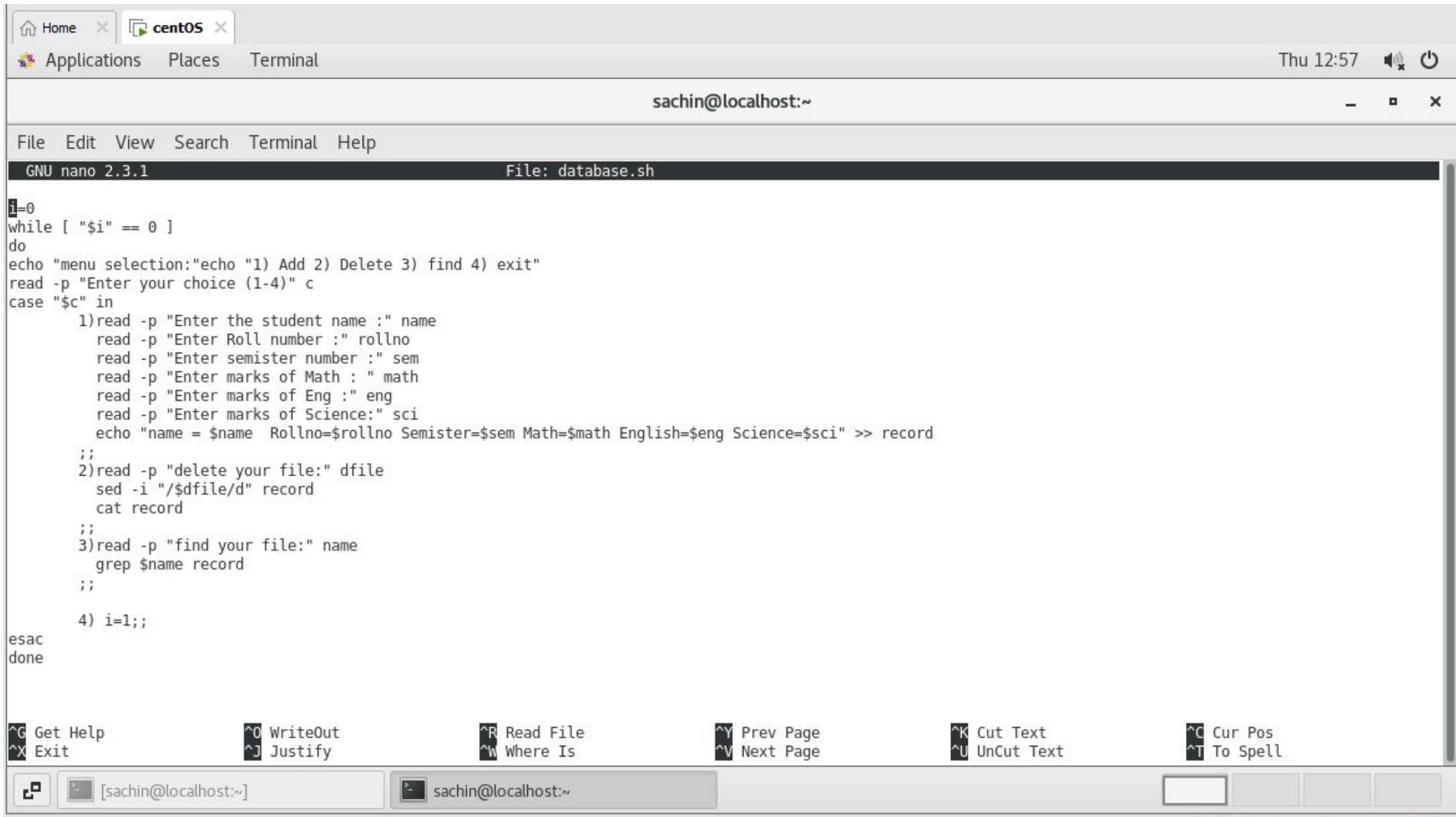
if [ $a -eq 0 -a $b -eq 0 -o $c -eq 0 ]
then
    echo "$y is a leap year."
else
    echo "$y is not a leap year"
fi
```

At the bottom of the terminal, there is a status bar with various keyboard shortcuts: ^G Get Help, ^X Exit, ^O WriteOut, ^J Justify, ^R Read File, ^W Where Is, [Read 13 lines], ^Y Prev Page, ^V Next Page, ^K Cut Text, ^U UnCut Text, ^C Cur Pos, and ^T To Spell. The bottom-most bar shows the terminal icon, the prompt 'sachin@localhost:~', and some empty input fields.



Assignment-6

1. Write a shell script to create a menu driven program for adding, deletion or finding a record in a database. Database should have the field like rollno, name, semester and marks of three subjects. Last option of the menu should be to exit the menu.



```
i=0
while [ "$i" == 0 ]
do
echo "menu selection:"echo "1) Add 2) Delete 3) find 4) exit"
read -p "Enter your choice (1-4)" c
case "$c" in
1)read -p "Enter the student name :" name
read -p "Enter Roll number :" rollno
read -p "Enter semester number :" sem
read -p "Enter marks of Math : " math
read -p "Enter marks of Eng : " eng
read -p "Enter marks of Science:" sci
echo "name = $name Rollno=$rollno Semester=$sem Math=$math English=$eng Science=$sci" >> record
;;
2)read -p "delete your file:" dfile
sed -i "$dfile/d" record
cat record
;;
3)read -p "find your file:" name
grep $name record
;;
4) i=1;;
esac
done
```

GNU nano 2.3.1 File: database.sh

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

Home x centOS x

Applications Places Terminal

Thu 12:56

sachin@localhost:~

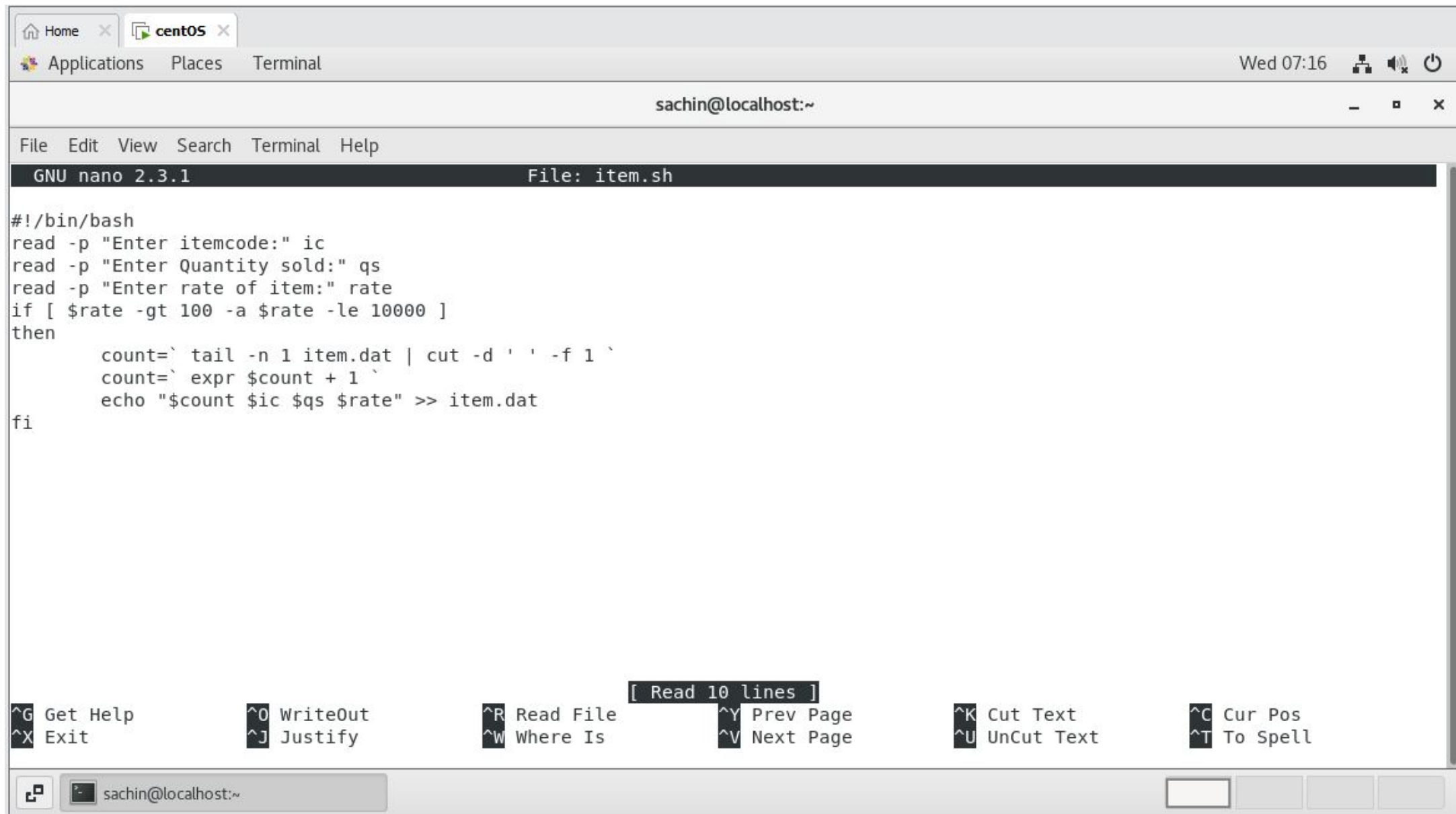
File Edit View Search Terminal Help

```
[sachin@localhost ~]$ cat record
name = sachin Rollno=1 Semester=2 Math=85 English=83 Science=86
name = palash Rollno=2 Semester=2 Math=80 English=90 Science=85
[sachin@localhost ~]$ ./database.sh
menu selection:echo 1) Add 2) Delete 3) find 4) exit
Enter your choice (1-4)1
Enter the student name :sandeep
Enter Roll number :3
Enter semester number :2
Enter marks of Math : 80
Enter marks of Eng :75
Enter marks of Science:83
menu selection:echo 1) Add 2) Delete 3) find 4) exit
Enter your choice (1-4)2
delete your file:sachin
name = palash Rollno=2 Semester=2 Math=80 English=90 Science=85
name = sandeep Rollno=3 Semester=2 Math=80 English=75 Science=83
menu selection:echo 1) Add 2) Delete 3) find 4) exit
Enter your choice (1-4)3
find your file:palash
name = palash Rollno=2 Semester=2 Math=80 English=90 Science=85
menu selection:echo 1) Add 2) Delete 3) find 4) exit
Enter your choice (1-4)
```

[sachin@localhost:~]

sachin@localhost:~

2. Write a unix shell to add records to a file called item.dat The fields being itemcode, qty, sold and rate
item_code to be generated, qty_sold should be greater than 0, rate between 100 to 10000.

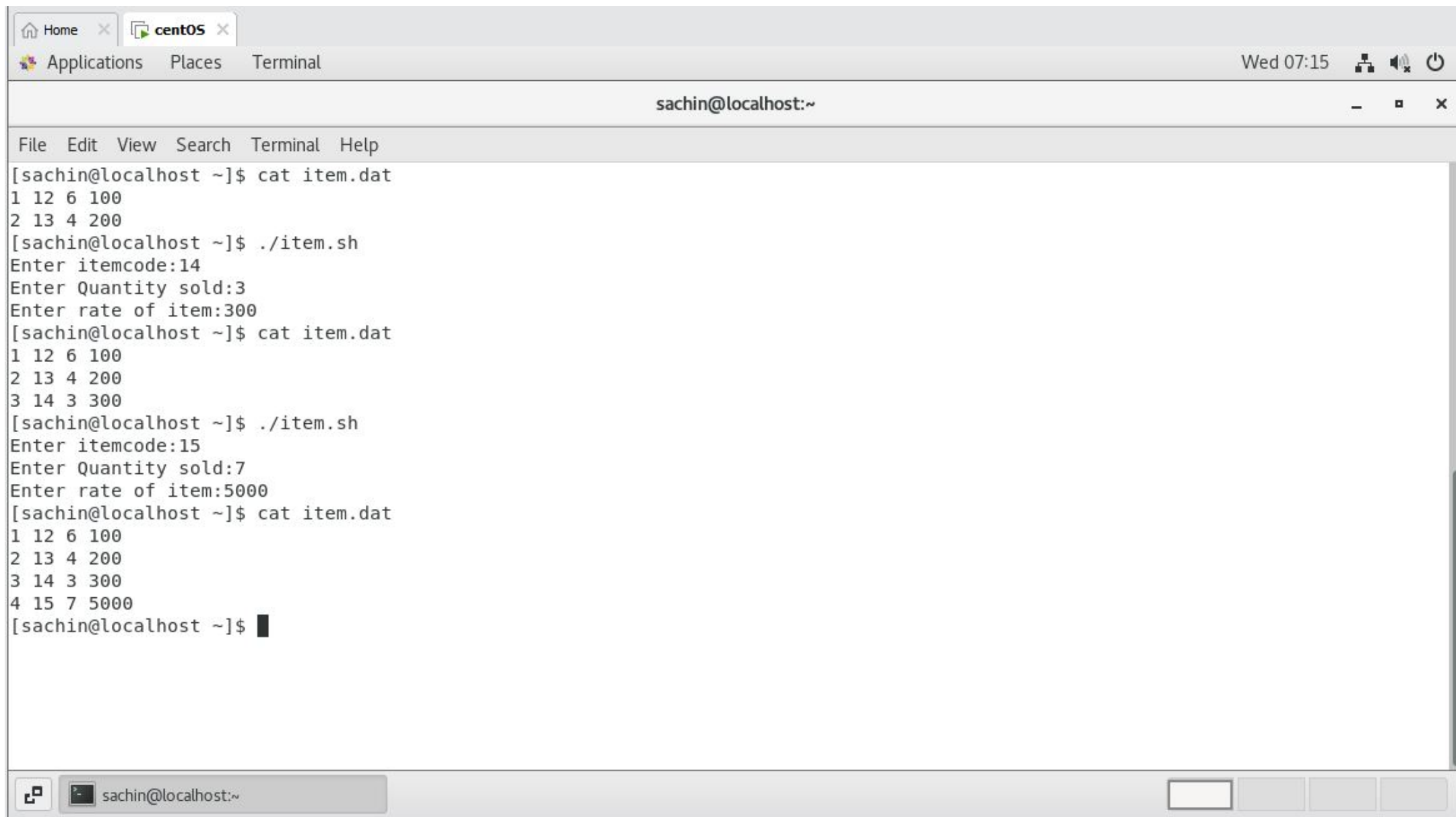


```
GNU nano 2.3.1 File: item.sh

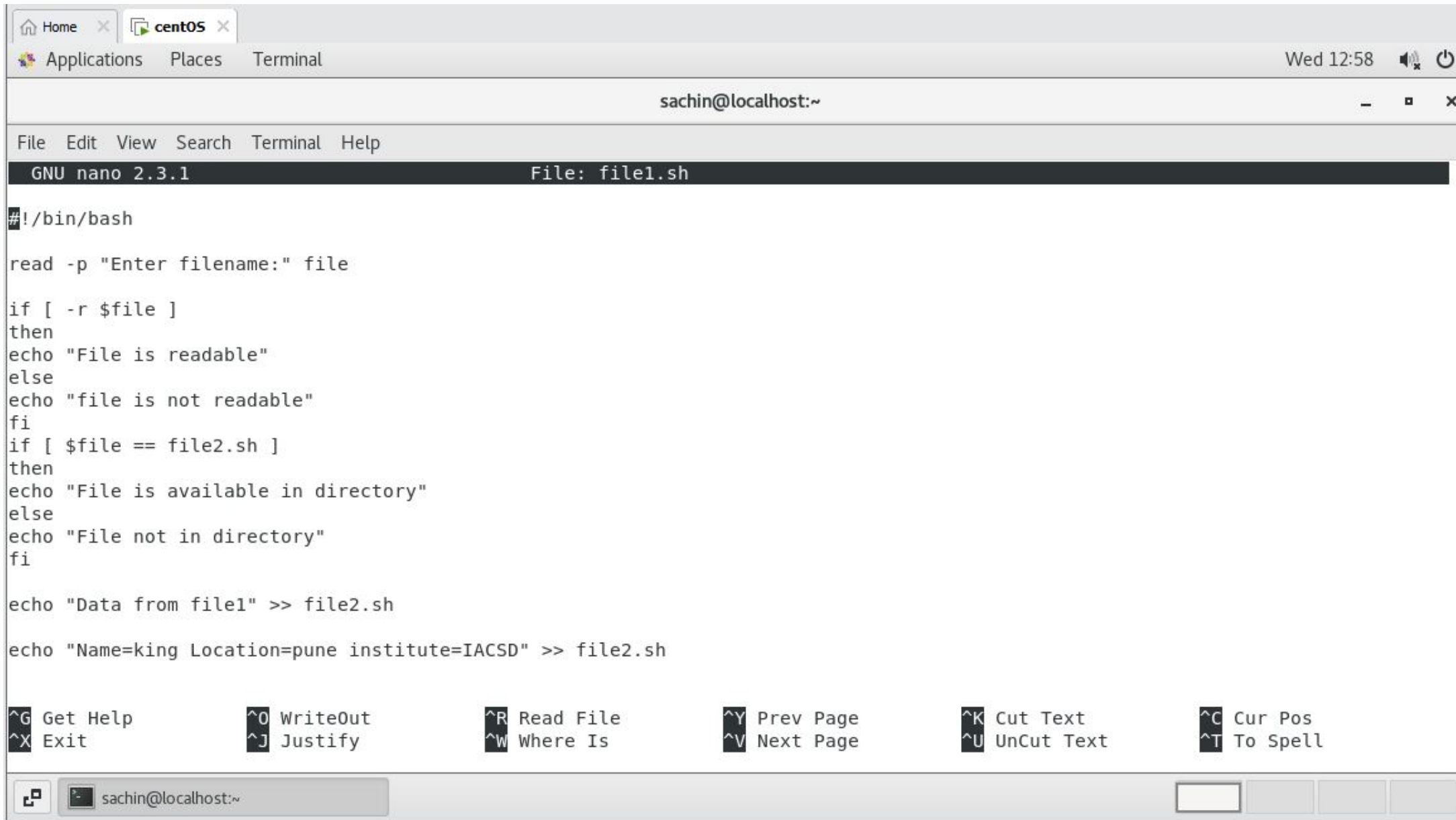
#!/bin/bash
read -p "Enter itemcode:" ic
read -p "Enter Quantity sold:" qs
read -p "Enter rate of item:" rate
if [ $rate -gt 100 -a $rate -le 10000 ]
then
    count=`tail -n 1 item.dat | cut -d ' ' -f 1`
    count=`expr $count + 1`
    echo "$count $ic $qs $rate" >> item.dat
fi
```

[Read 10 lines]

^G Get Help	^O WriteOut	^R Read File	^Y Prev Page	^K Cut Text	^C Cur Pos
^X Exit	^J Justify	^W Where Is	^V Next Page	^U UnCut Text	^T To Spell



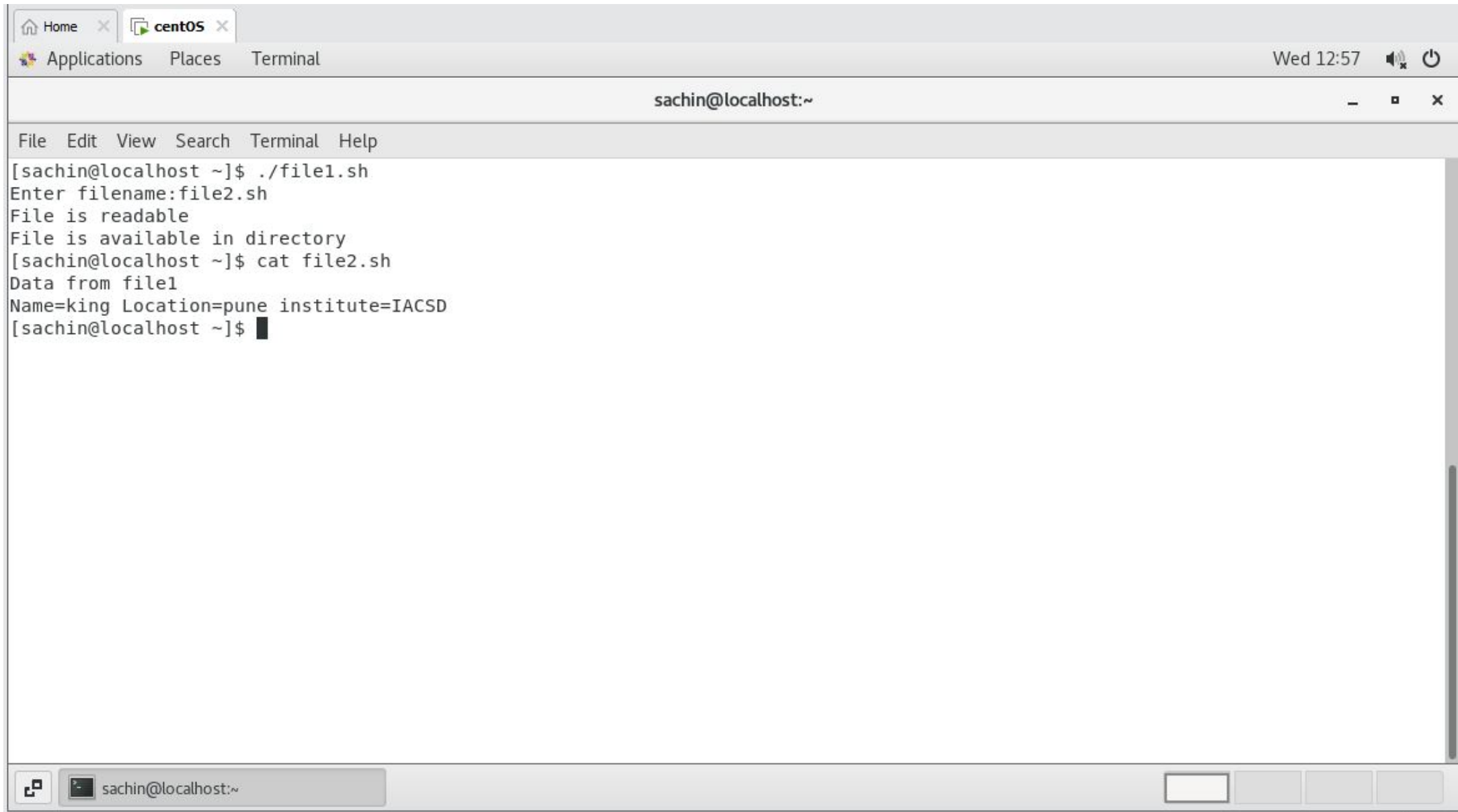
3. Write a script which copies the content of file1 to file2 without using cp command. It should check if file has a read permissions; if not, it should print an error message. If file2 exists, then it should ask the user whether he wants to overwrite it.



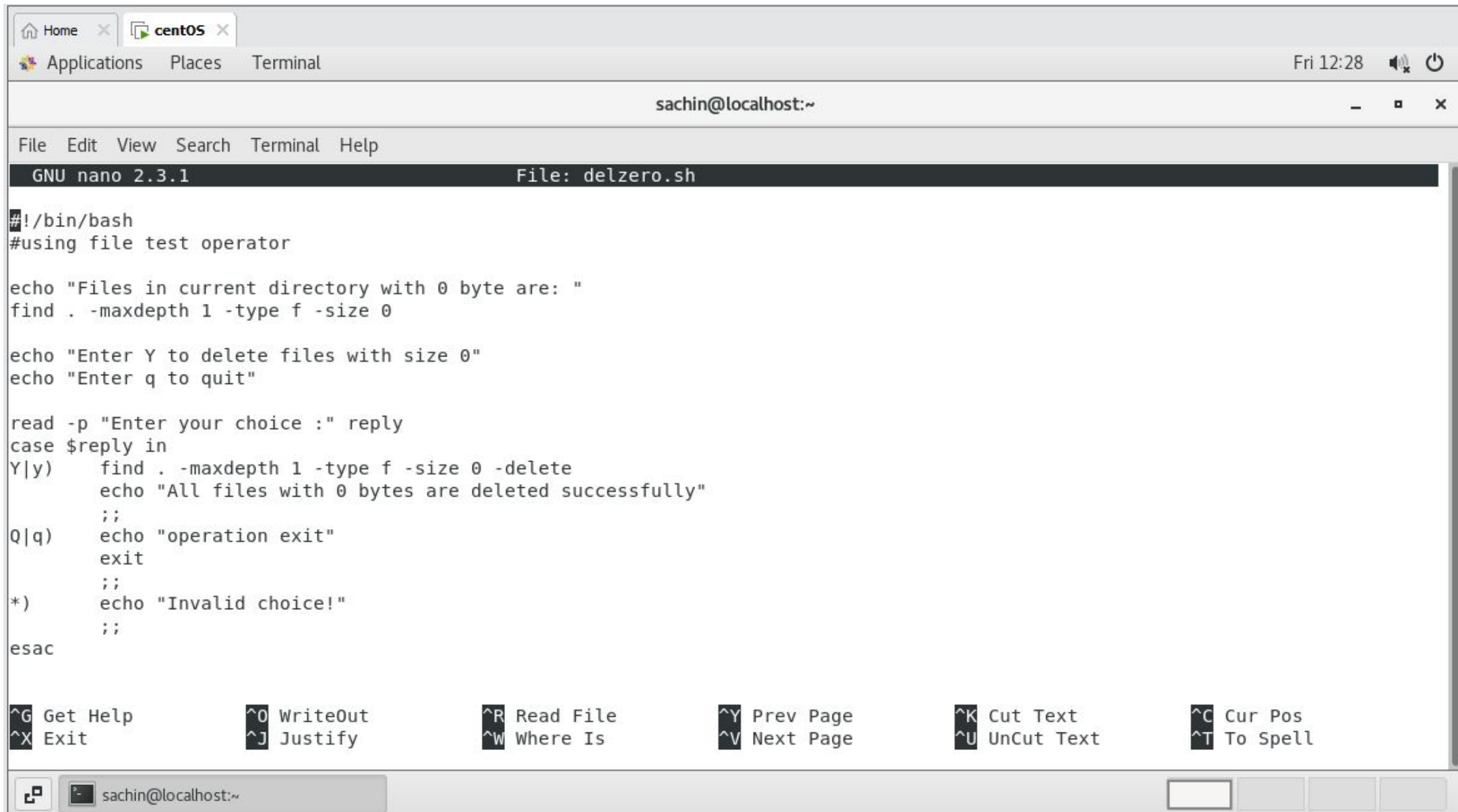
The screenshot shows a Linux desktop environment with a window titled 'centOS'. Inside the window is a terminal application. The terminal title bar shows 'sachin@localhost:~'. The terminal content shows a nano editor editing 'file1.sh'. The script in file1.sh is as follows:

```
#!/bin/bash
read -p "Enter filename:" file
if [ -r $file ]
then
echo "File is readable"
else
echo "file is not readable"
fi
if [ $file == file2.sh ]
then
echo "File is available in directory"
else
echo "File not in directory"
fi
echo "Data from file1" >> file2.sh
echo "Name=king Location=pune institute=IACSD" >> file2.sh
```

The terminal window also displays a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. At the bottom of the terminal window, there is a status bar with various shortcuts: ^G Get Help, ^X Exit, ^O WriteOut, ^J Justify, ^R Read File, ^W Where Is, ^Y Prev Page, ^V Next Page, ^K Cut Text, ^U UnCut Text, ^C Cur Pos, and ^T To Spell.



4. Write a shell scripts that delete all files in current directory with 0 byte.



```
GNU nano 2.3.1 File: delzero.sh

#!/bin/bash
#using file test operator

echo "Files in current directory with 0 byte are: "
find . -maxdepth 1 -type f -size 0

echo "Enter Y to delete files with size 0"
echo "Enter q to quit"

read -p "Enter your choice :" reply
case $reply in
Y|y)  find . -maxdepth 1 -type f -size 0 -delete
      echo "All files with 0 bytes are deleted successfully"
      ;;
Q|q)  echo "operation exit"
      exit
      ;;
*)    echo "Invalid choice!"
      ;;
esac

^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit          ^J Justify      ^W Where Is    ^V Next Page    ^U UnCut Text   ^T To Spell
```

HomecentOS

ApplicationsPlacesTerminal

Fri 12:25

sachin@localhost:~

FileEditViewSearchTerminalHelp

```
[sachin@localhost ~]$ ls -lh
total 108K
-rwxrwxr-x. 1 sachin sachin 8.3K May 20 05:58 a.out
-rwxrwxr-x. 1 sachin sachin 418 May 15 20:42 calculator.sh
-rwxrwxr-x. 1 sachin sachin 759 May 16 15:05 case.sh
-rwxrwxr-x. 1 sachin sachin 197 May 15 20:55 compare.sh
-rwxrwxr-x. 1 sachin sachin 650 May 20 12:54 database.sh
-rwxrwxr-x. 1 sachin sachin 428 May 21 12:23 delzero.sh
drwxr-xr-x. 2 sachin sachin  6 May 12 10:40 Desktop
-rwxrwxr-x. 1 sachin sachin 282 May 20 02:59 dirlisting.sh
drwxr-xr-x. 2 sachin sachin  6 May 12 10:40 Documents
drwxr-xr-x. 2 sachin sachin  6 May 12 10:40 Downloads
drwxrwxr-x. 2 sachin sachin  6 May 12 12:37 family
-rwxrwxr-x. 1 sachin sachin 329 May 19 12:53 file1.sh
-rw-rw-r--. 1 sachin sachin  56 May 19 12:57 file2.sh
-rw-rw-r--. 1 sachin sachin 235 May 20 05:25 file.c
-rw-rw-r--. 1 sachin sachin 256 May 20 05:32 fork.c
-rw-rw-r--. 1 sachin sachin  56 May 20 14:04 item.dat
-rwxrwxr-x. 1 sachin sachin 274 May 19 07:13 item.sh
-rwxrwxr-x. 1 sachin sachin 225 May 18 12:01 leap.sh
drwxr-xr-x. 2 sachin sachin 114 May 12 11:52 Music
-rwxrwxr-x. 1 sachin sachin 400 May 18 19:58 numbers.sh
-rw-rw-r--. 1 sachin sachin 158 May 20 05:58 orphan.c
-rwxrwxr-x. 1 sachin sachin 342 May 17 11:16 palindrome.sh
drwxr-xr-x. 2 sachin sachin 4.0K May 20 05:25 Pictures
-rwxrwxr-x. 1 sachin sachin 226 May 18 11:37 prime.sh
```

sachin@localhost:~

HomecentOS

ApplicationsPlacesTerminal

Fri 12:27

sachin@localhost:~

FileEditViewSearchTerminalHelp

drwxr-xr-x. 2 sachin sachin 6 May 12 10:40 Public
-rw-rw-r--. 1 sachin sachin 131 May 20 12:56 record
-rwxrwxr-x. 1 sachin sachin 407 May 16 18:22 salary.sh
-rwxrwxr-x. 1 sachin sachin 406 May 16 18:43 salary.sh.save
-rw-----. 1 sachin sachin 0 May 19 06:44 sedTYDqvt
-rw-----. 1 sachin sachin 0 May 19 06:40 sedW6naNC
-rwxrwxr-x. 1 sachin sachin 272 May 15 19:38 si.sh
-rw-rw-r--. 1 sachin sachin 0 May 12 11:08 songsX.mp3
-rwxrwxr-x. 1 sachin sachin 238 May 17 12:43 sumofdigits.sh
drwxr-xr-x. 2 sachin sachin 6 May 12 10:40 Templates
drwxr-xr-x. 2 sachin sachin 108 May 12 12:02 Videos
drwxrwxr-x. 2 sachin sachin 112 May 12 12:48 work
-rw-rw-r--. 1 sachin sachin 248 May 20 05:43 zombie.c
[sachin@localhost ~]\$./delzero.sh
Files in current directory with 0 byte are:
./songsX.mp3
./sedW6naNC
./sedTYDqvt
Enter Y to delete files with size 0
Enter q to quit
Enter your choice :y
All files with 0 bytes are deleted successfully
[sachin@localhost ~]\$ ls -lh
total 108K
-rwxrwxr-x. 1 sachin sachin 8.3K May 20 05:58 a.out
-rwxrwxr-x. 1 sachin sachin 418 May 15 20:42 calculator.sh
-rwxrwxr-x. 1 sachin sachin 759 May 16 15:05 case.sh

sachin@localhost:~

Home

centOS

ApplicationsPlacesTerminal

Fri 12:27

sachin@localhost:~

File

Edit

View

Search

Terminal

Help

-rwxrwxr-x. 1 sachin sachin 197 May 15 20:55 compare.sh

-rwxrwxr-x. 1 sachin sachin 650 May 20 12:54 database.sh

-rwxrwxr-x. 1 sachin sachin 428 May 21 12:23 delzero.sh

drwxr-xr-x. 2 sachin sachin 6 May 12 10:40 Desktop

-rwxrwxr-x. 1 sachin sachin 282 May 20 02:59 dirlisting.sh

drwxr-xr-x. 2 sachin sachin 6 May 12 10:40 Documents

drwxr-xr-x. 2 sachin sachin 6 May 12 10:40 Downloads

drwxrwxr-x. 2 sachin sachin 6 May 12 12:37 family

-rwxrwxr-x. 1 sachin sachin 329 May 19 12:53 file1.sh

-rw-rw-r--. 1 sachin sachin 56 May 19 12:57 file2.sh

-rw-rw-r--. 1 sachin sachin 235 May 20 05:25 file.c

-rw-rw-r--. 1 sachin sachin 256 May 20 05:32 fork.c

-rw-rw-r--. 1 sachin sachin 56 May 20 14:04 item.dat

-rwxrwxr-x. 1 sachin sachin 274 May 19 07:13 item.sh

-rwxrwxr-x. 1 sachin sachin 225 May 18 12:01 leap.sh

drwxr-xr-x. 2 sachin sachin 114 May 12 11:52 Music

-rwxrwxr-x. 1 sachin sachin 400 May 18 19:58 numbers.sh

-rw-rw-r--. 1 sachin sachin 158 May 20 05:58 orphan.c

-rwxrwxr-x. 1 sachin sachin 342 May 17 11:16 palindrome.sh

drwxr-xr-x. 2 sachin sachin 4.0K May 20 05:25 Pictures

-rwxrwxr-x. 1 sachin sachin 226 May 18 11:37 prime.sh

drwxr-xr-x. 2 sachin sachin 6 May 12 10:40 Public

-rw-rw-r--. 1 sachin sachin 131 May 20 12:56 record

-rwxrwxr-x. 1 sachin sachin 407 May 16 18:22 salary.sh

-rwxrwxr-x. 1 sachin sachin 406 May 16 18:43 salary.sh.save

-rwxrwxr-x. 1 sachin sachin 272 May 15 19:38 si.sh

-rwxrwxr-x. 1 sachin sachin 238 May 17 12:43 sumofdigits.sh

drwxr-xr-x. 2 sachin sachin 6 May 12 10:40 Templates

drwxr-xr-x. 2 sachin sachin 108 May 12 12:02 Videos

drwxrwxr-x. 2 sachin sachin 112 May 12 12:48 work

-rw-rw-r--. 1 sachin sachin 248 May 20 05:43 zombie.c

[sachin@localhost ~]\$ clear

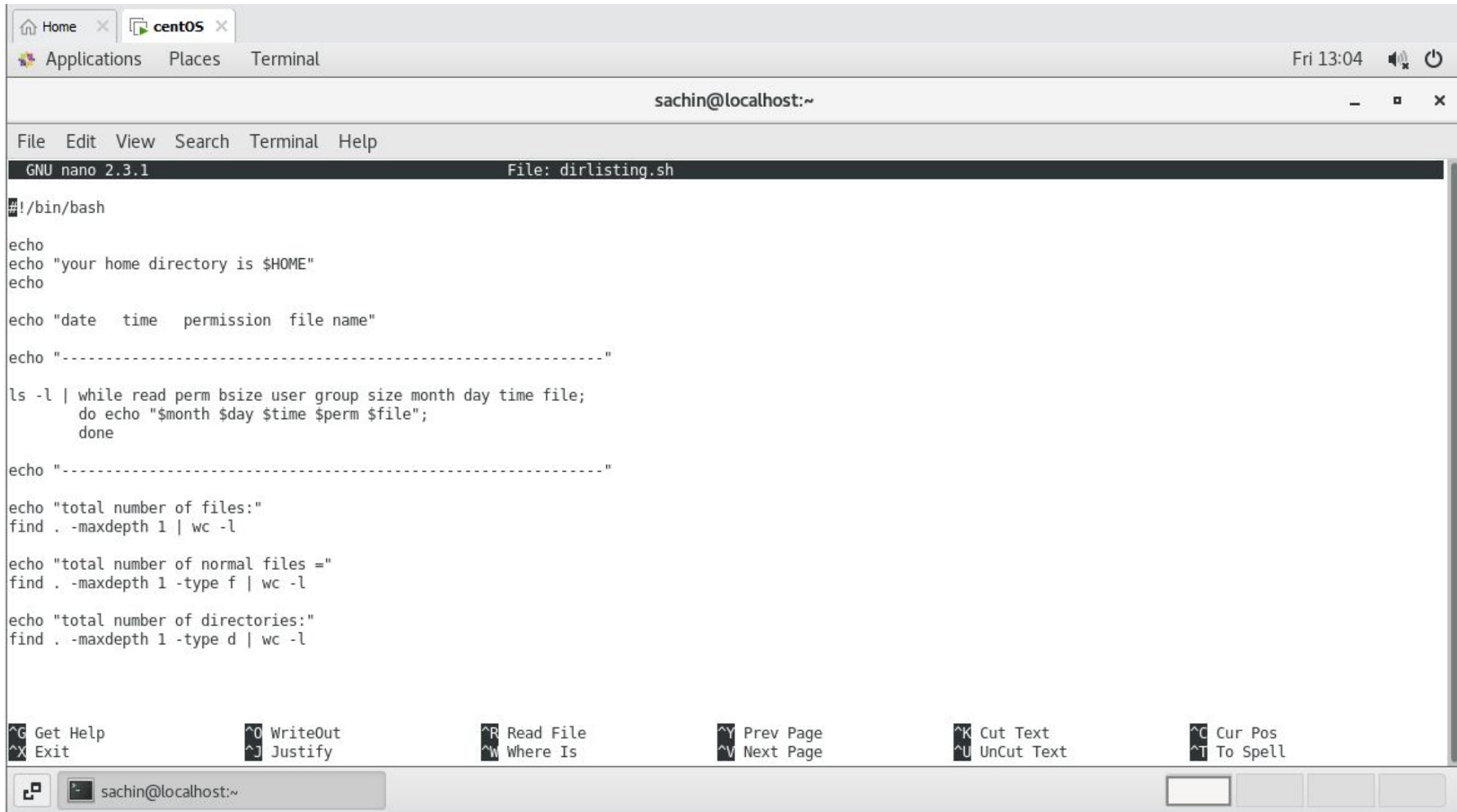
sachin@localhost:~

5. Write a shell script to display a directory listing as follows. Your home directory is <home directory name>

File name date time permission

Filename1 date time permission Filename2 date time permission Filename3 date time permission

Total no. of files : <total number> Total no of normal file : <number> Total no of directory : <number>



The screenshot shows a Linux desktop environment with a window titled 'centOS'. The window contains a terminal application running the nano text editor. The editor is editing a file named 'dirlisting.sh'. The script content is as follows:

```
#!/bin/bash

echo
echo "your home directory is $HOME"
echo

echo "date   time   permission   file name"
echo "-----"

ls -l | while read perm bsize user group size month day time file;
do echo "$month $day $time $perm $file";
done

echo "-----"

echo "total number of files:"
find . -maxdepth 1 | wc -l

echo "total number of normal files ="
find . -maxdepth 1 -type f | wc -l

echo "total number of directories:"
find . -maxdepth 1 -type d | wc -l
```

The terminal window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The status bar at the bottom shows the user 'sachin@localhost' and the current directory '~'. The nano editor's status bar at the very bottom displays various keyboard shortcuts for navigation and editing.

HomecentOS

ApplicationsPlacesTerminal

Fri 13:03

sachin@localhost:~

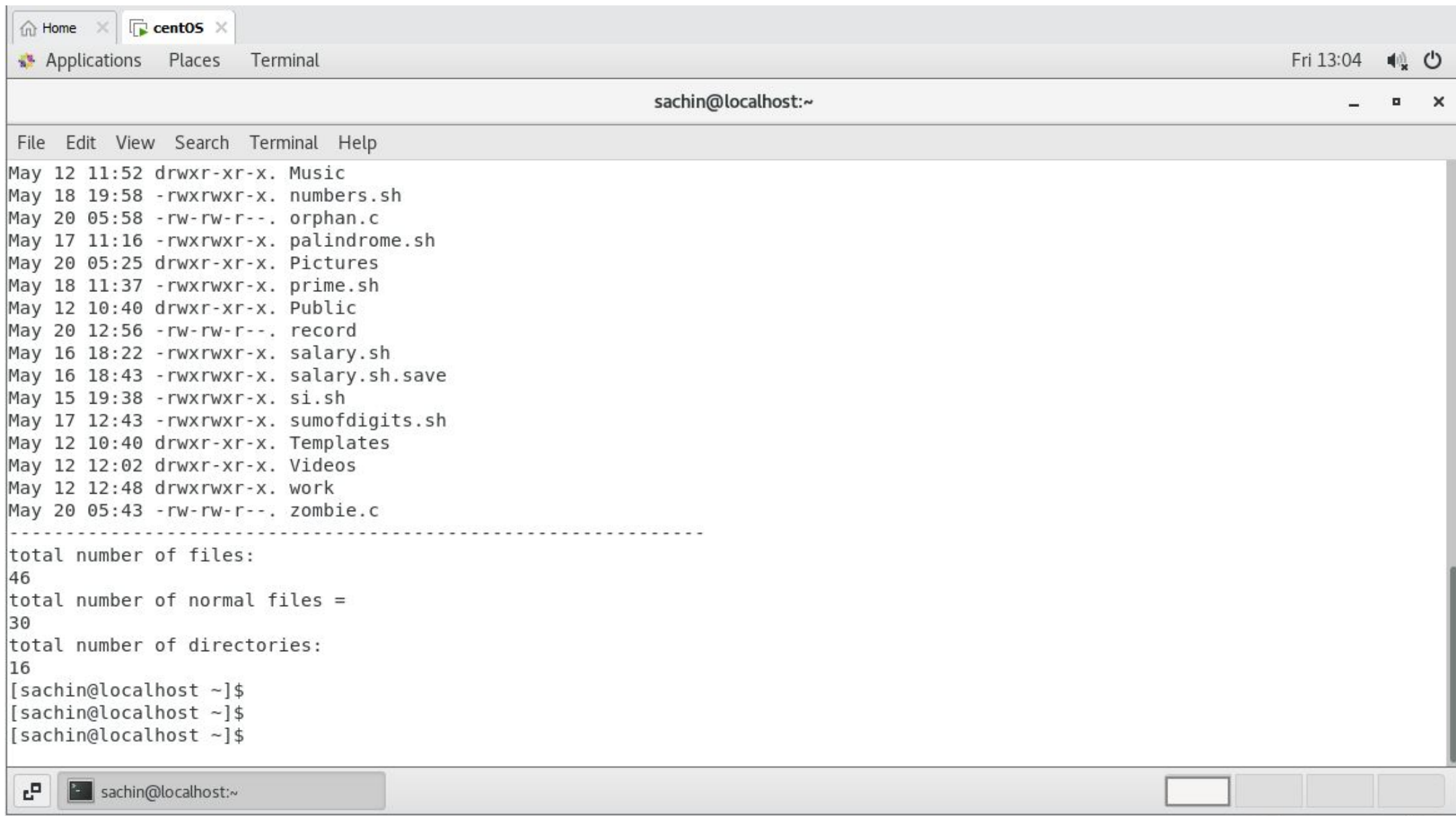
FileEditViewSearchTerminalHelp

```
[sachin@localhost ~]$ ./dirlisting.sh

your home directory is /home/sachin

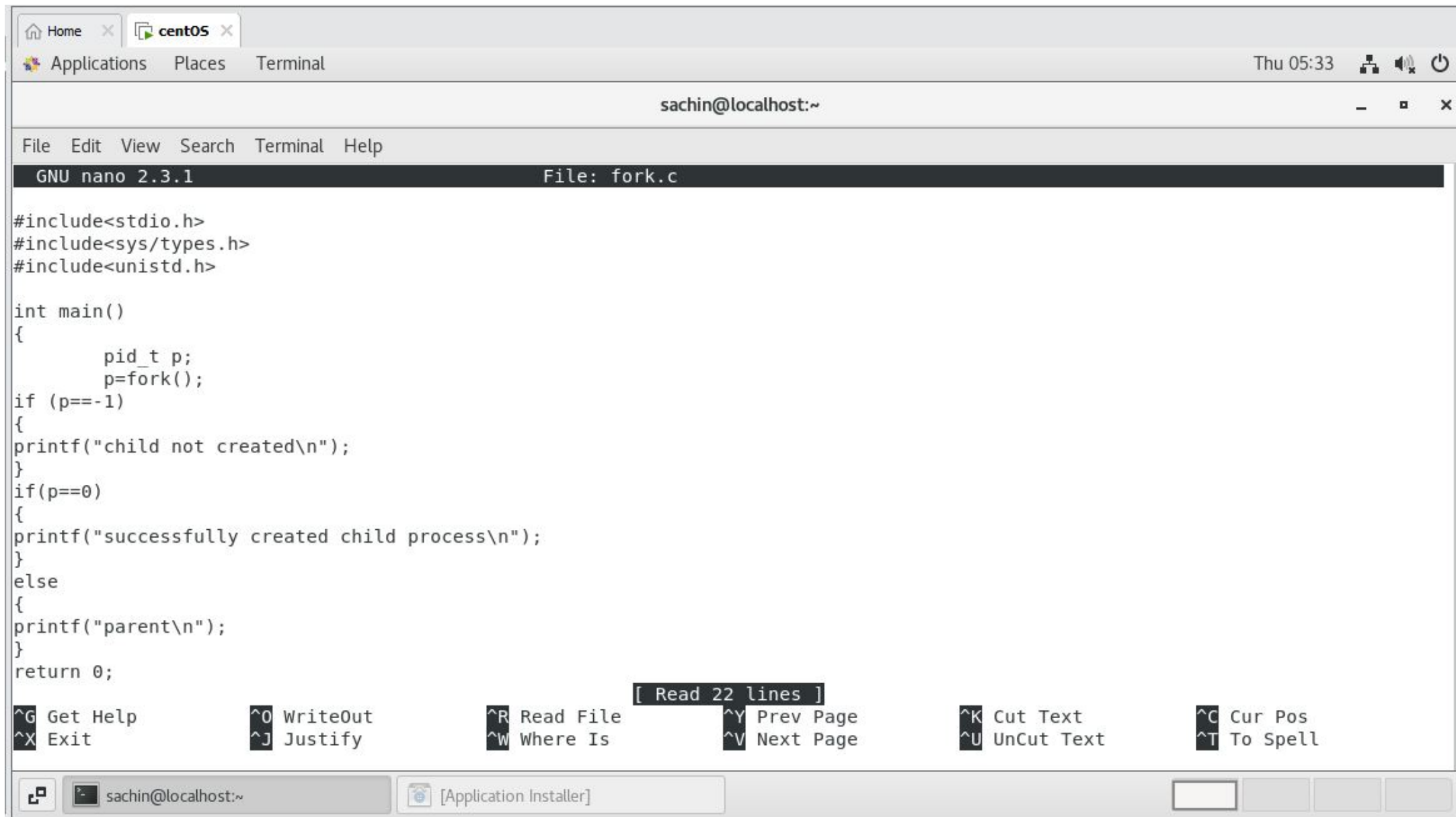
date    time    permission  file name
-----
total
May 20 05:58 -rwxrwxr-x. a.out
May 15 20:42 -rwxrwxr-x. calculator.sh
May 16 15:05 -rwxrwxr-x. case.sh
May 15 20:55 -rwxrwxr-x. compare.sh
May 20 12:54 -rwxrwxr-x. database.sh
May 21 12:23 -rwxrwxr-x. delzero.sh
May 12 10:40 drwxr-xr-x. Desktop
May 21 13:02 -rwxrwxr-x. dirlisting.sh
May 12 10:40 drwxr-xr-x. Documents
May 12 10:40 drwxr-xr-x. Downloads
May 12 12:37 drwxrwxr-x. family
May 19 12:53 -rwxrwxr-x. file1.sh
May 19 12:57 -rw-rw-r--. file2.sh
May 20 05:25 -rw-rw-r--. file.c
May 20 05:32 -rw-rw-r--. fork.c
May 20 14:04 -rw-rw-r--. item.dat
May 19 07:13 -rwxrwxr-x. item.sh
May 18 12:01 -rwxrwxr-x. leap.sh
May 12 11:52 drwxr-xr-x. Music
```

sachin@localhost:~



Assignment-7

1. Create Child process using fork()



```
GNU nano 2.3.1 File: fork.c

#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>

int main()
{
    pid_t p;
    p=fork();
    if (p==-1)
    {
        printf("child not created\n");
    }
    if(p==0)
    {
        printf("successfully created child process\n");
    }
    else
    {
        printf("parent\n");
    }
    return 0;
}
```

[Read 22 lines]

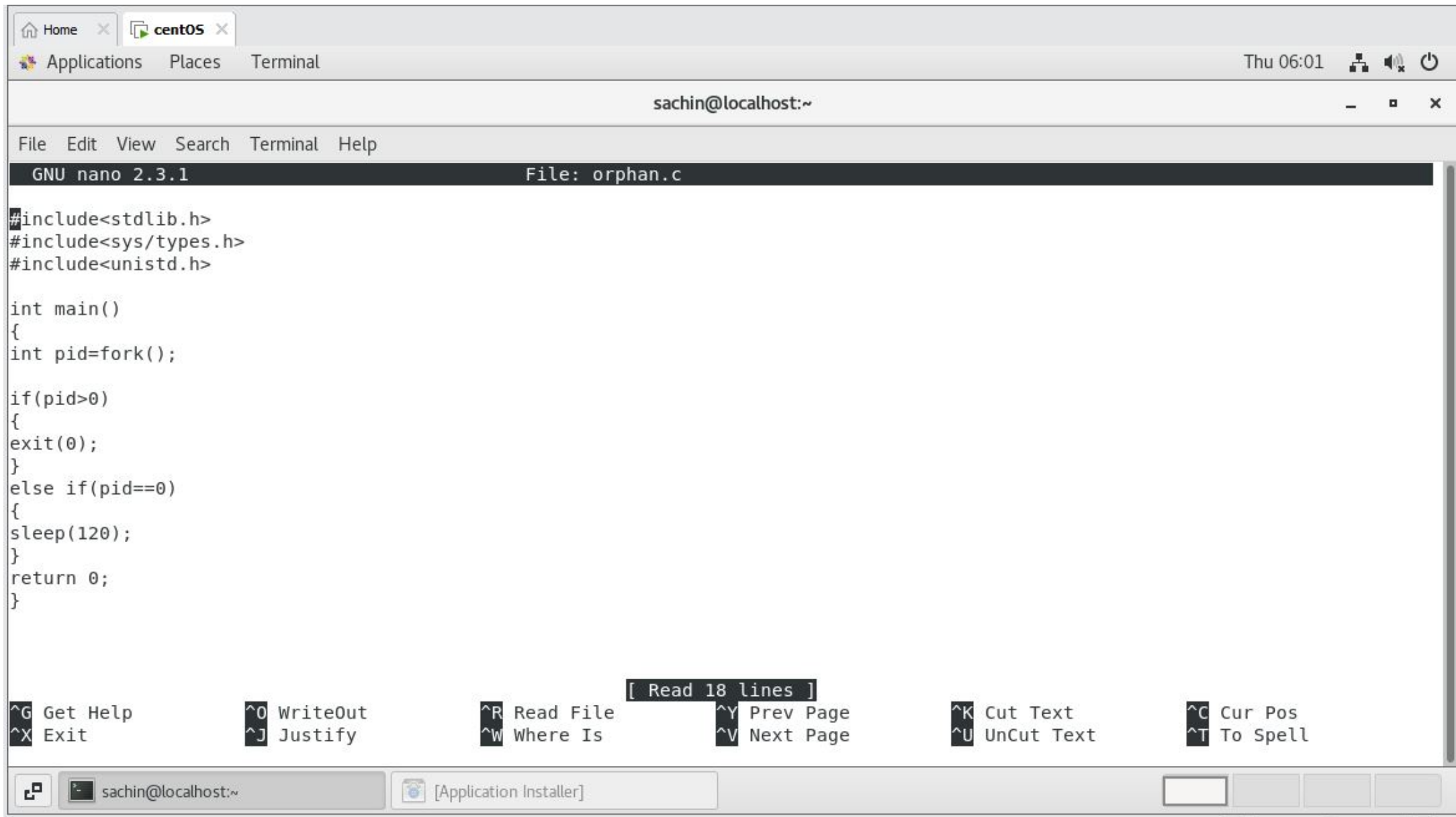
^G Get Help	^O WriteOut	^R Read File	^Y Prev Page	^K Cut Text	^C Cur Pos
^X Exit	^J Justify	^W Where Is	^V Next Page	^U UnCut Text	^T To Spell

sachin@localhost:~

File Edit View Search Terminal Help

```
[sachin@localhost ~]$ gcc fork.c
[sachin@localhost ~]$ ./a.out
parent
successfully created child process
[sachin@localhost ~]$
```

2. Create orphan process



The screenshot shows a Linux desktop environment with a window titled 'centOS'. The window contains a terminal application running the GNU nano 2.3.1 text editor. The file being edited is 'orphan.c'. The code in the file is a C program that creates an orphan process. The program includes `<stdlib.h>`, `<sys/types.h>`, and `<unistd.h>`. The `main` function calls `fork()` to create a child process. If the child process is successful (`pid > 0`), the parent process calls `exit(0)`. If the child process fails (`pid == 0`), the parent process calls `sleep(120)` before returning 0. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The status bar at the bottom shows the user 'sachin@localhost' and the application 'sachin@localhost:~'.

```
GNU nano 2.3.1 File: orphan.c

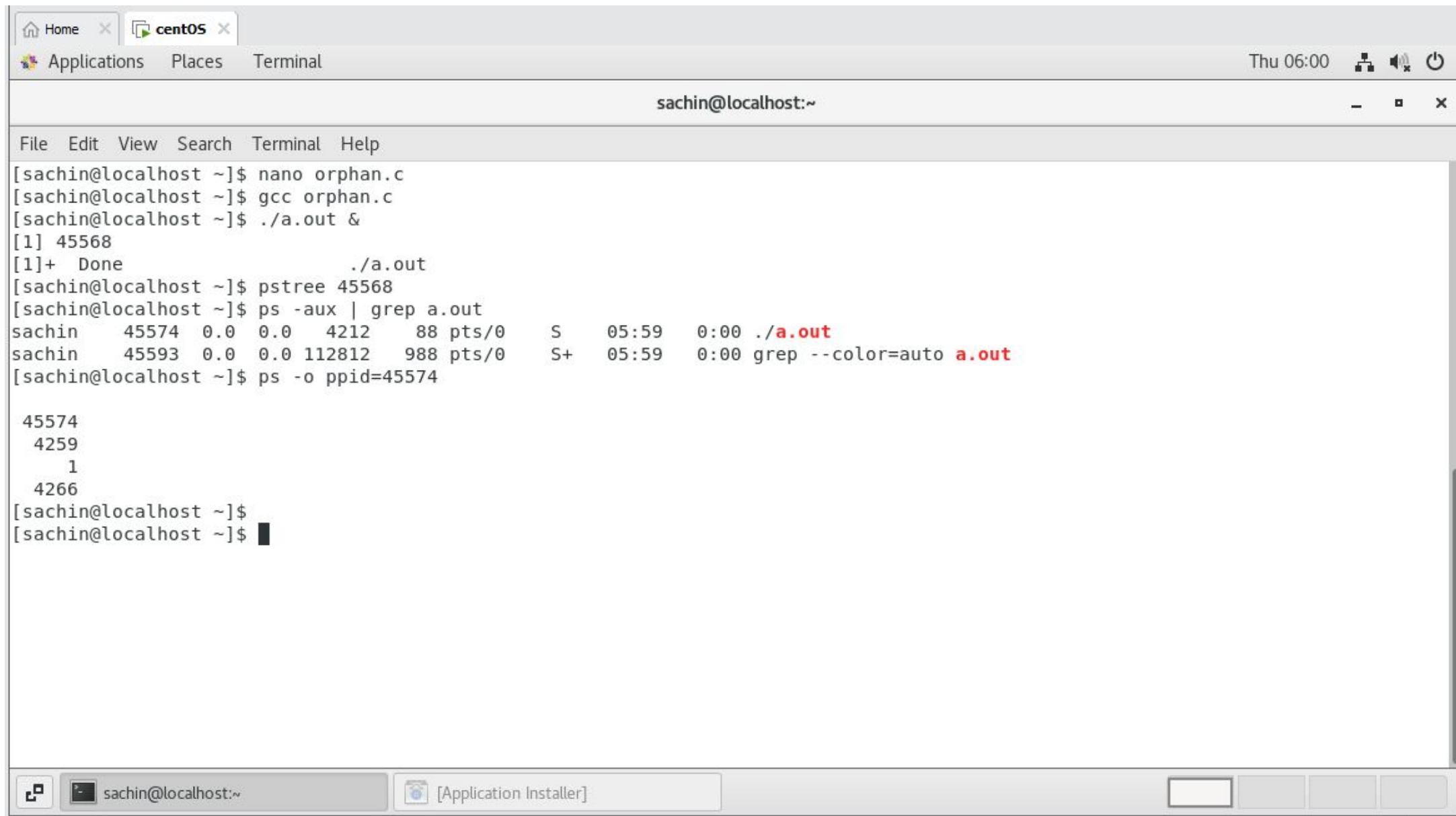
#include<stdlib.h>
#include<sys/types.h>
#include<unistd.h>

int main()
{
int pid=fork();

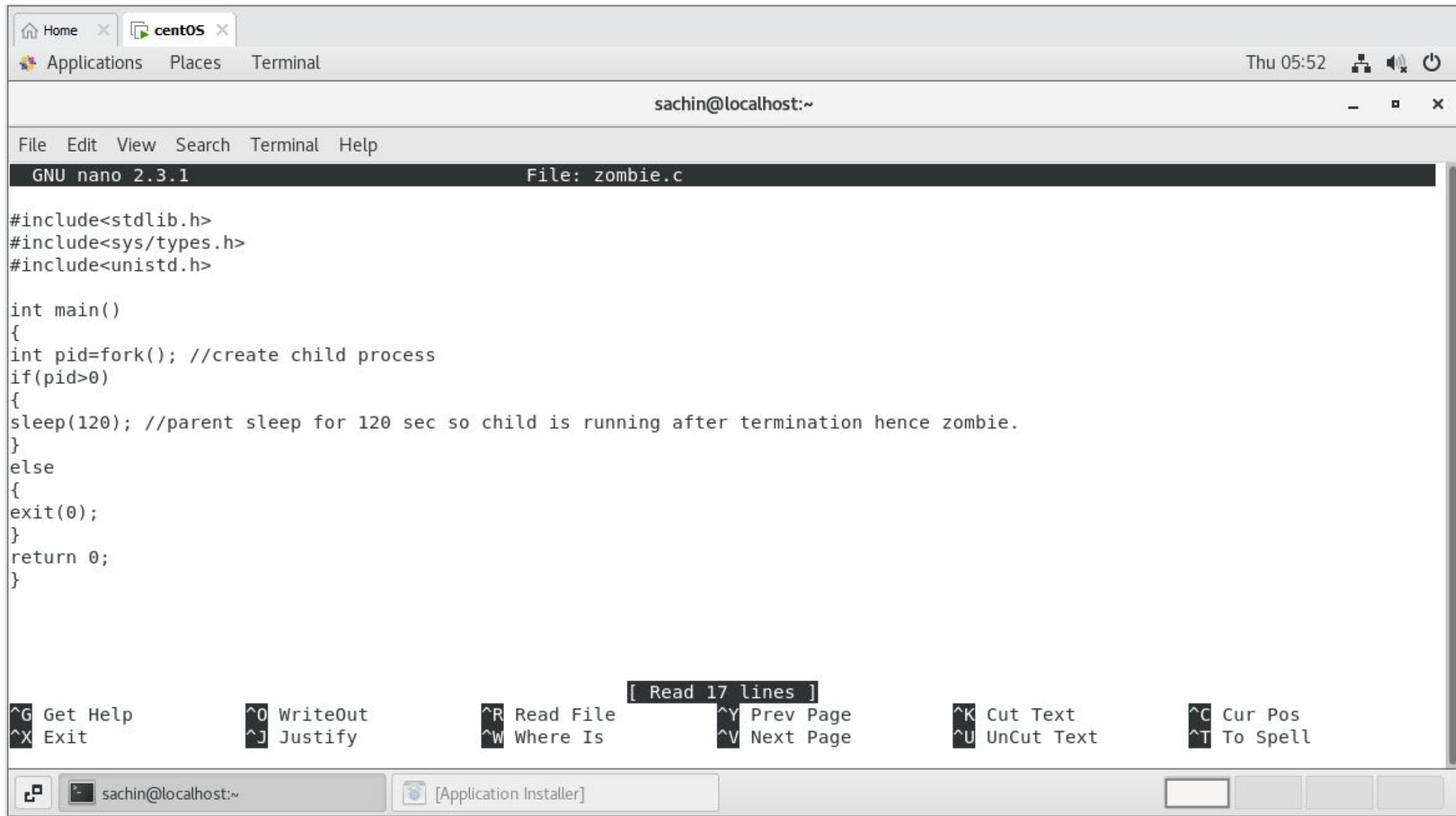
if(pid>0)
{
exit(0);
}
else if(pid==0)
{
sleep(120);
}
return 0;
}

[ Read 18 lines ]

^G Get Help      ^O WriteOut      ^R Read File     ^Y Prev Page     ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify       ^W Where Is      ^V Next Page     ^U UnCut Text    ^T To Spell
```



3. Create Zombie process



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window has a title bar with 'Home', 'centOS', and 'Terminal' tabs. The top right corner shows the time 'Thu 05:52' and system icons. The terminal prompt is 'sachin@localhost:~'. The terminal content shows the GNU nano 2.3.1 editor editing a file named 'zombie.c'. The code in the file is a C program that creates a child process using 'fork()' and sleeps for 120 seconds to create a zombie process. The bottom of the terminal window displays a status bar with various keyboard shortcuts and a '[Read 17 lines]' indicator.

```
GNU nano 2.3.1 File: zombie.c

#include<stdlib.h>
#include<sys/types.h>
#include<unistd.h>

int main()
{
int pid=fork(); //create child process
if(pid>0)
{
sleep(120); //parent sleep for 120 sec so child is running after termination hence zombie.
}
else
{
exit(0);
}
return 0;
}

[ Read 17 lines ]
^G Get Help      ^O WriteOut      ^R Read File     ^Y Prev Page     ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify       ^W Where Is     ^V Next Page     ^U UnCut Text    ^T To Spell
```