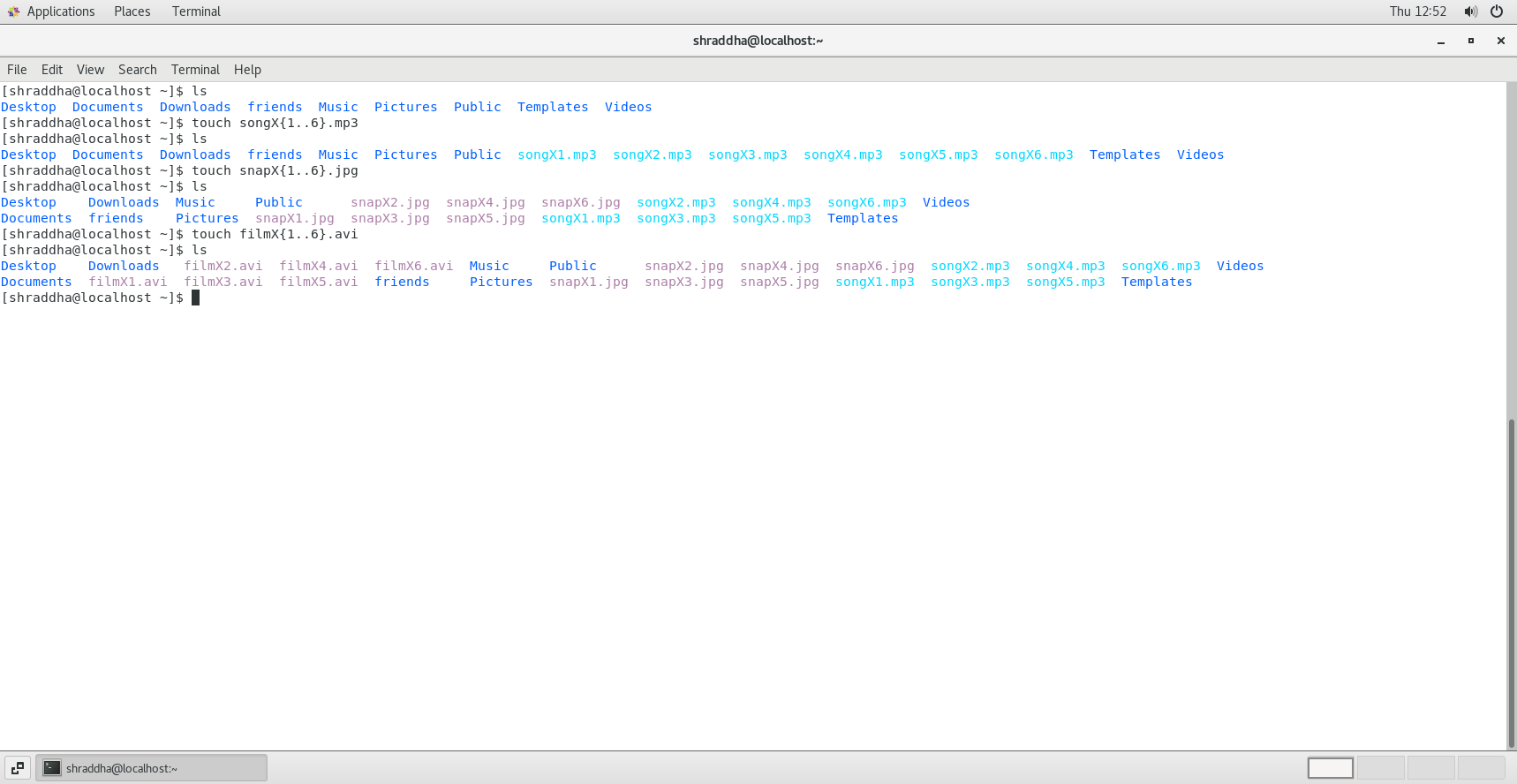
**Assignment 1**

1. In your home directory, create sets of empty practice files

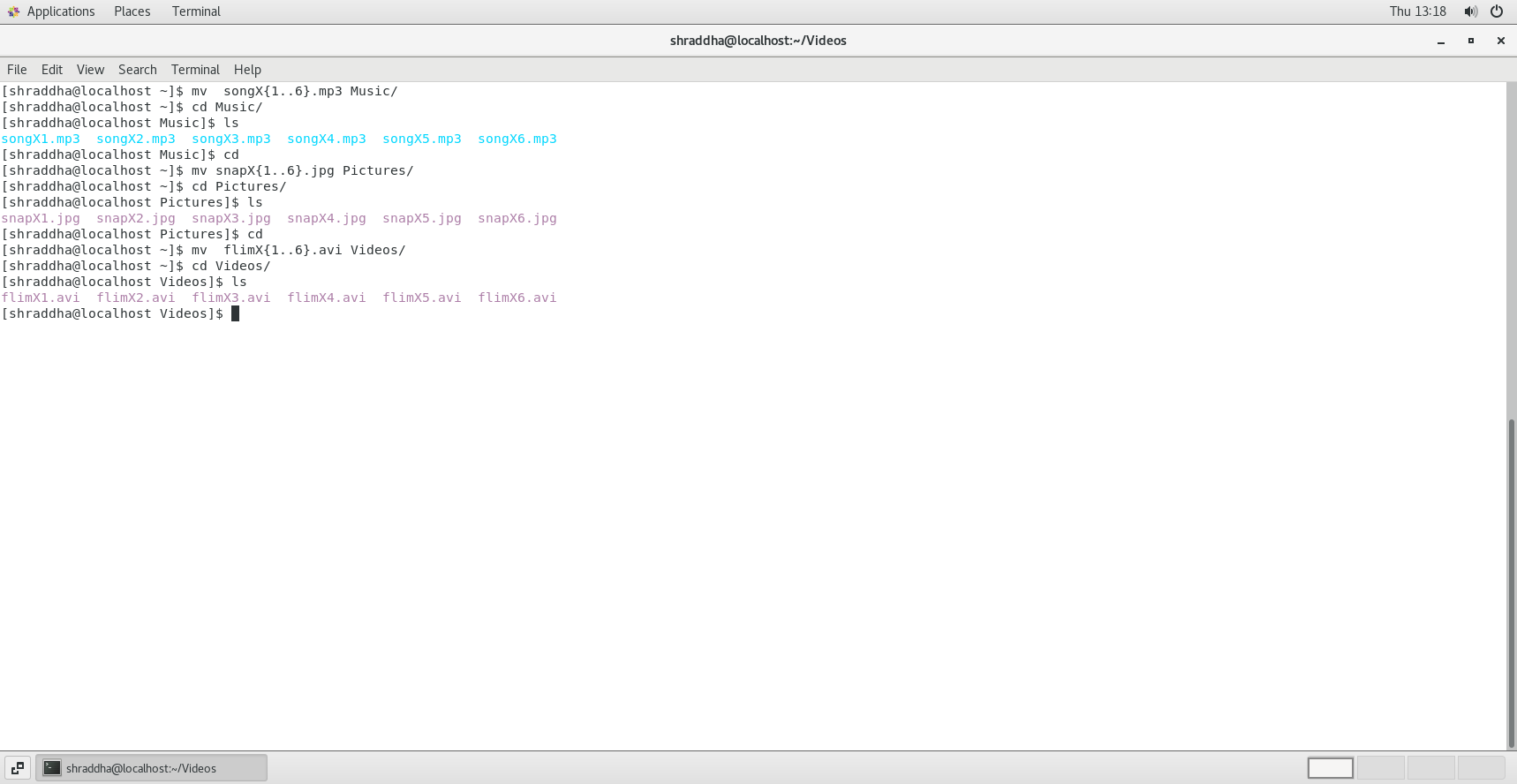
* Create 6 files with names of the forms on songsX.mp3.
* Create 6 files with names of the form snapX.jpg.
* Create 6 files with names of the form filmX.avi.

In each set, replace X with the numbers 1 through 6.

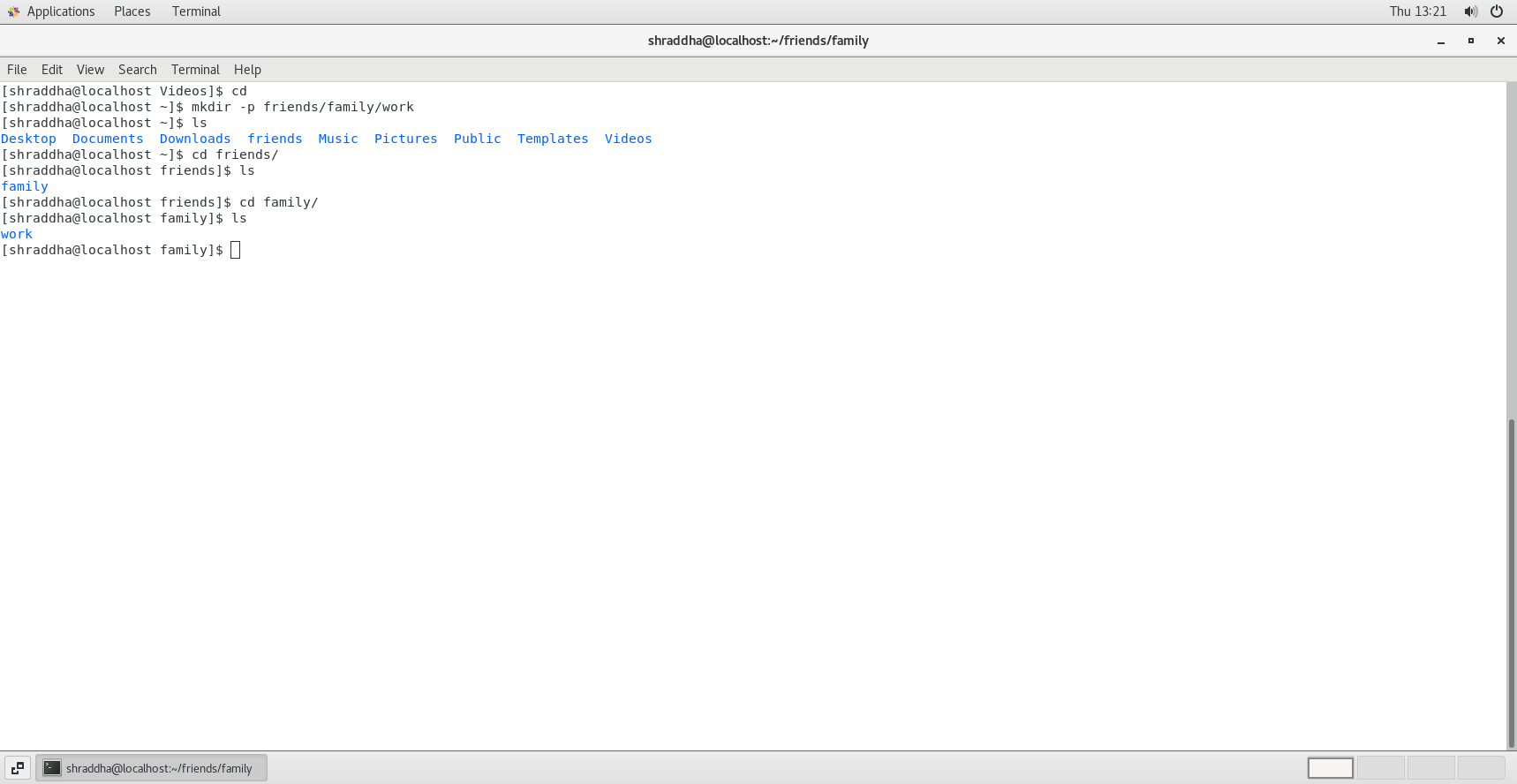


2. From your home directory,

* Move songs file into your Music subdirectory.
* Move snap file into your Pictures subdirectory.
* Move your movie files into Videos subdirectory



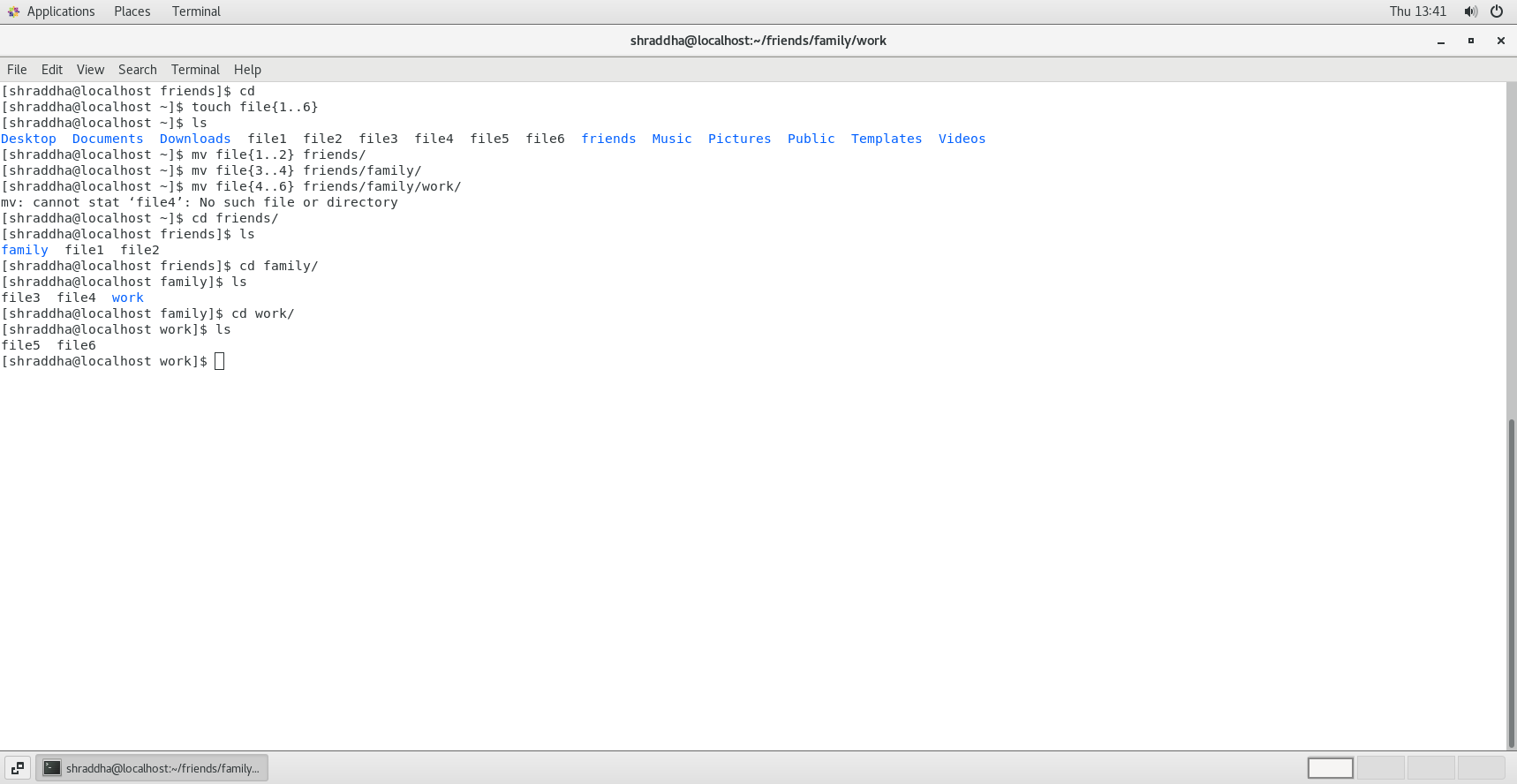
3. Create 3 subdirectories for organizing your files named friends,family,work



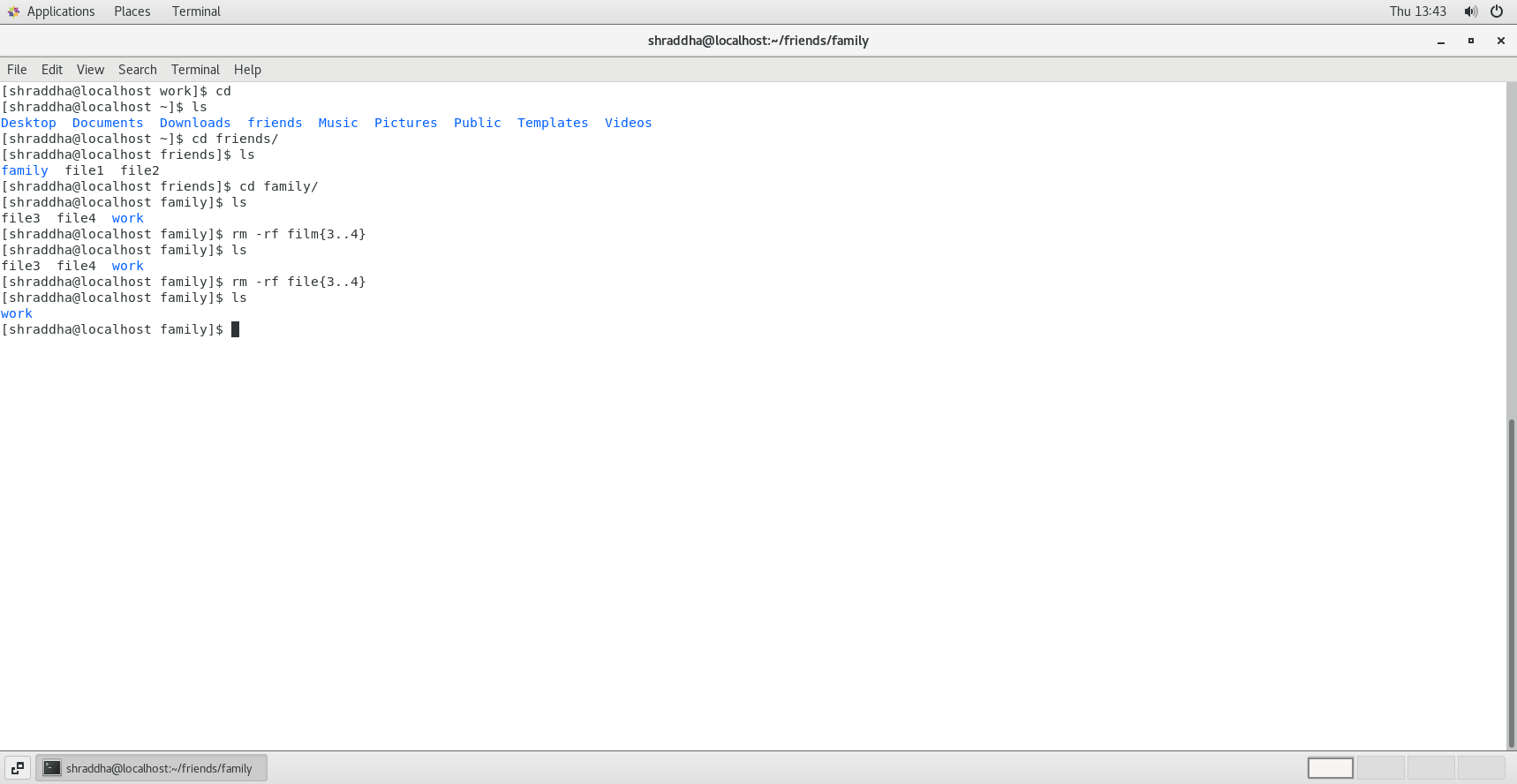
4. Copy files (all types ) containing numbers 1 and 2 to the friends folder.

    Copy files (all types) containing numbers 3 and 4 to the family folder.

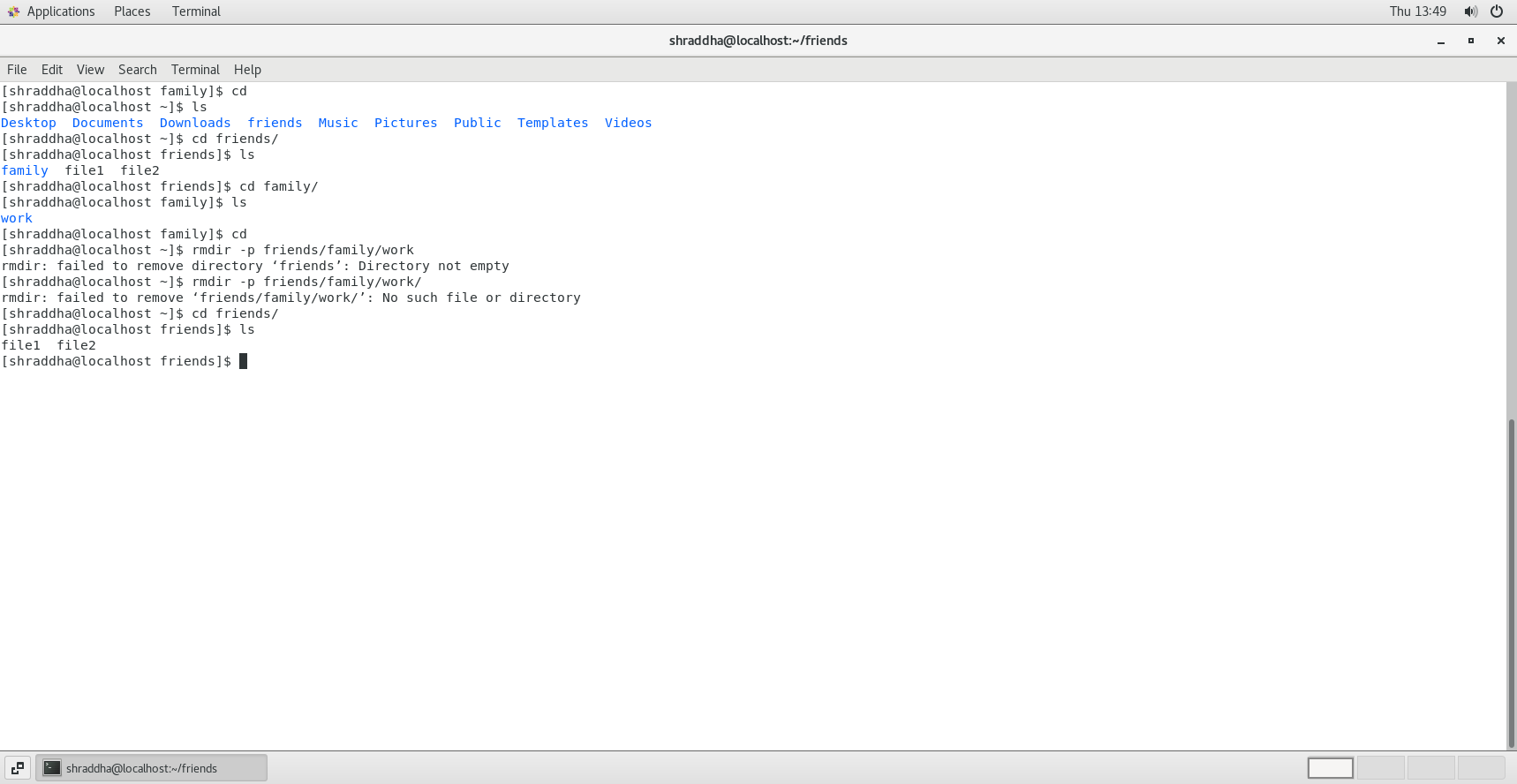
    Copy files (all types) containing numbers 5 and 6 to the work folder.



5. Delete all files in family subdirectory.

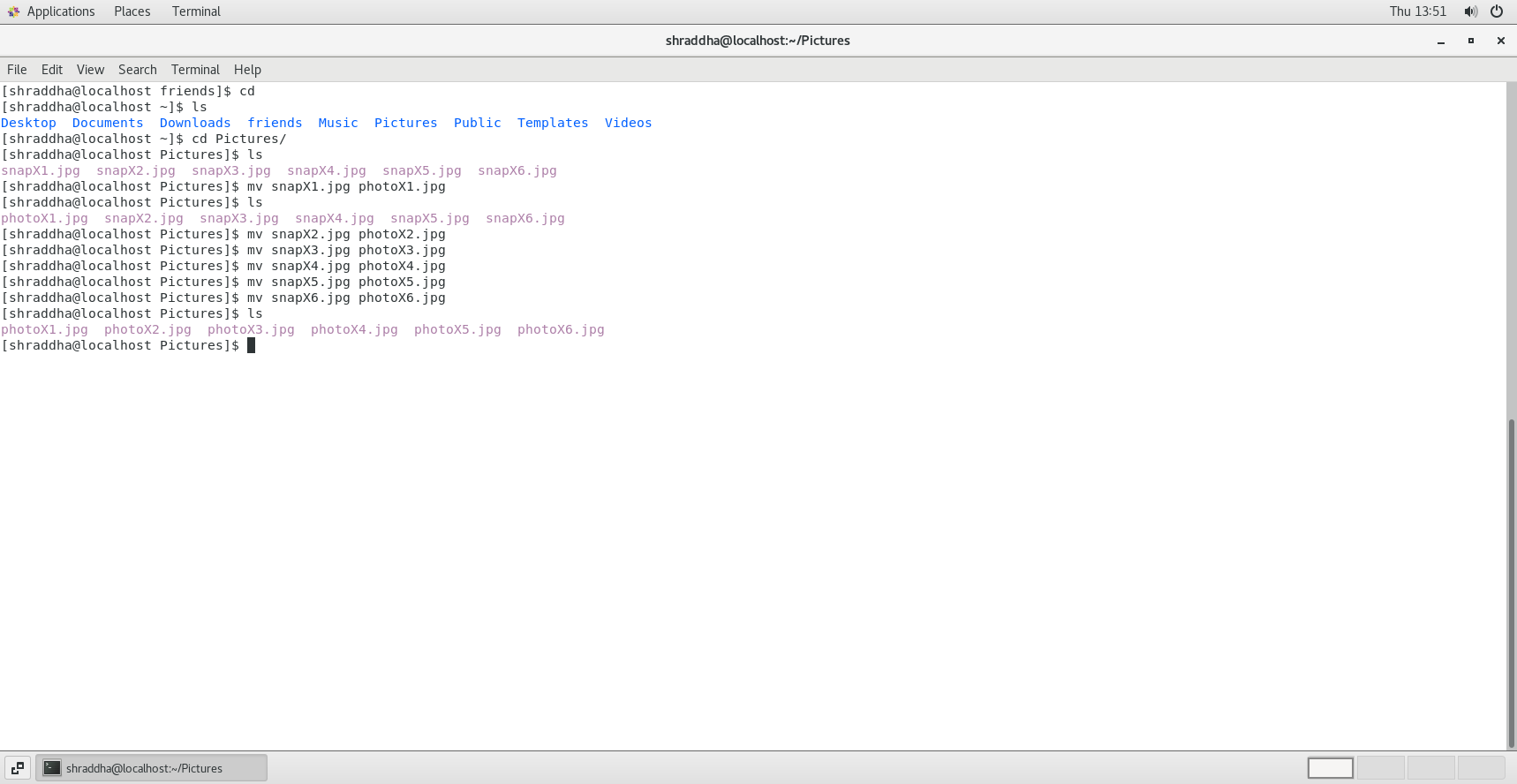


6. Delete friends subdirectory



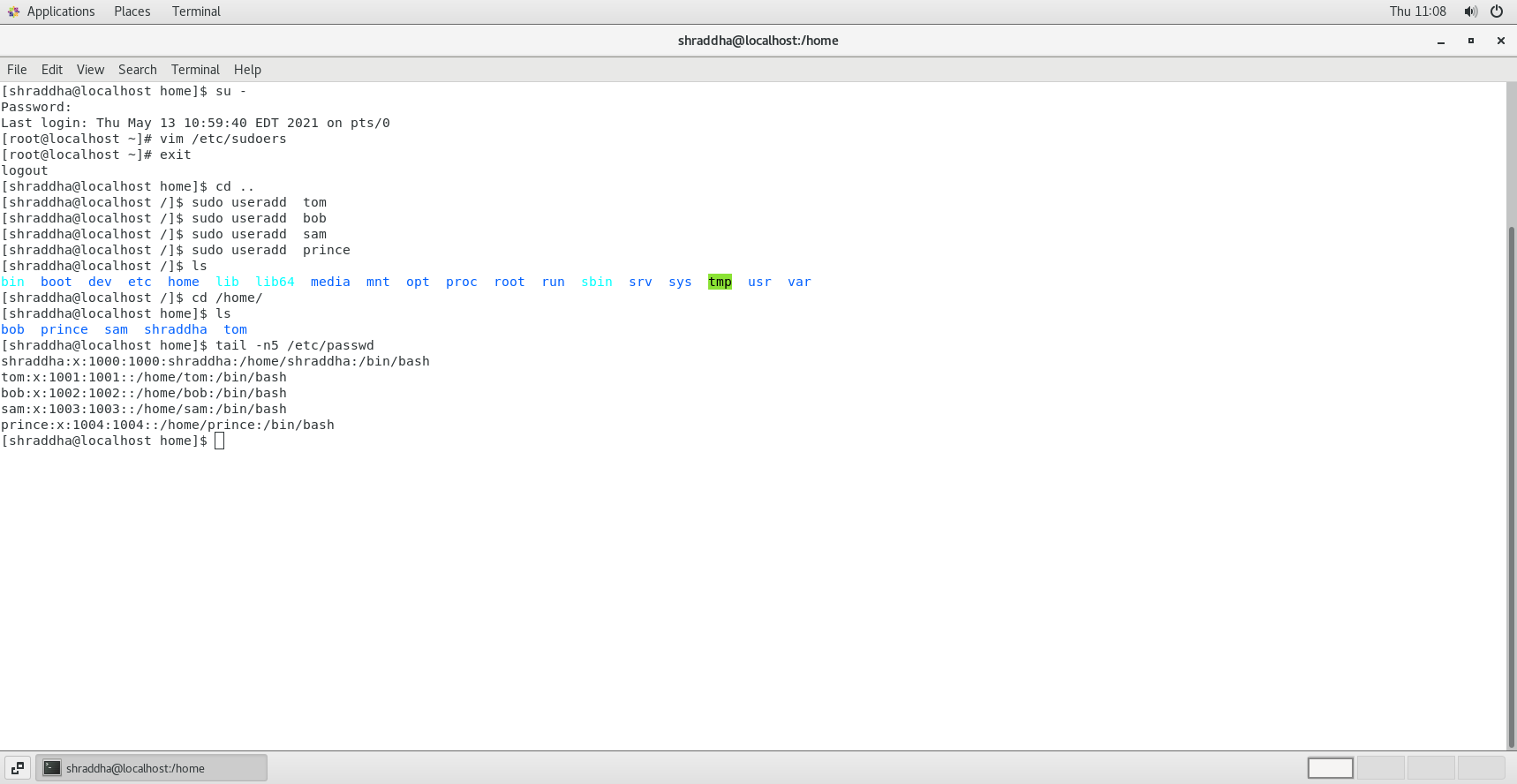
7. rename all snap files in work directory to new name photoX.jpg

X is a number

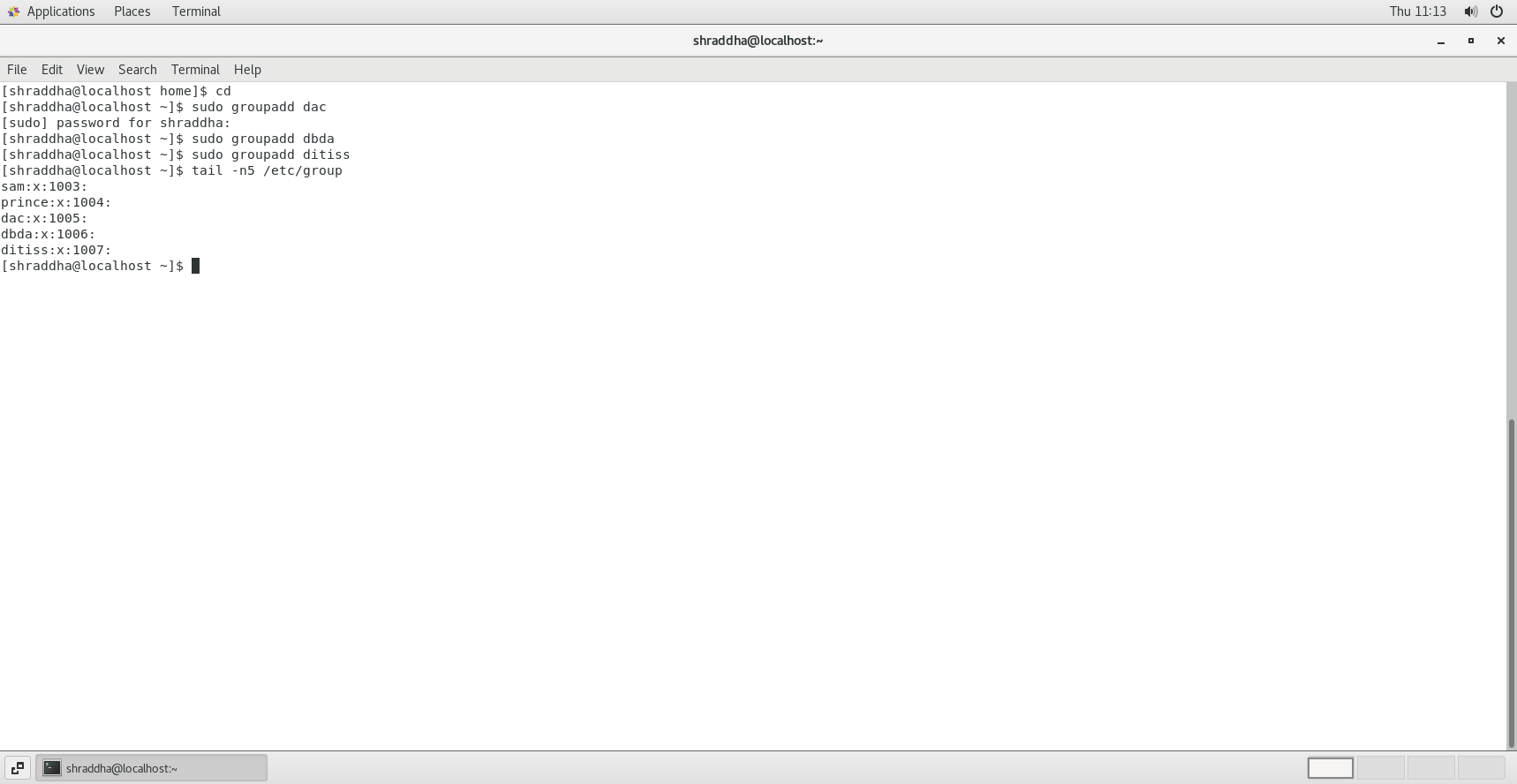


**Assignment 2**

1. Create user tom , bob , sam , prince



2. Create Group dac , dbda ,ditiss

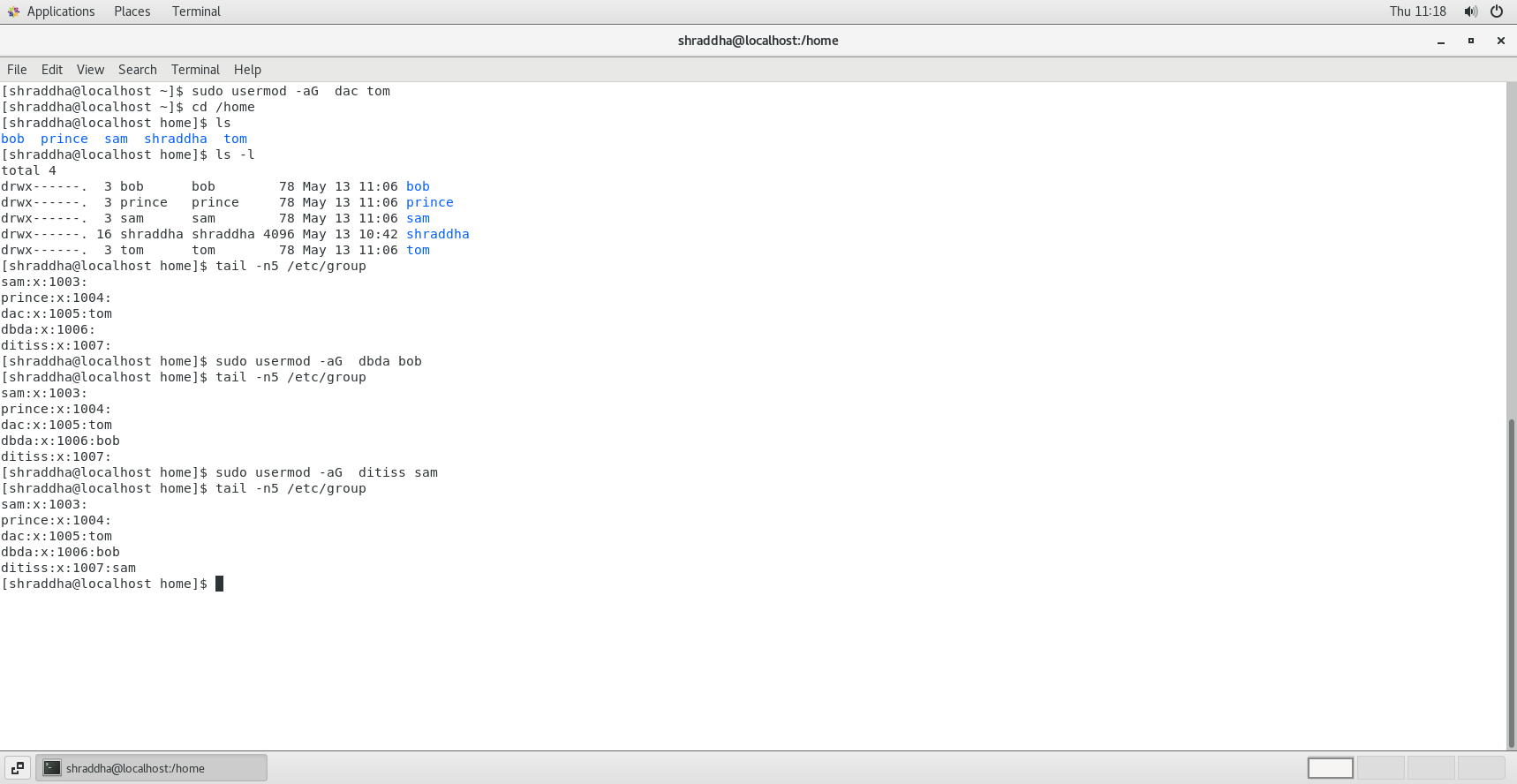


3. add user

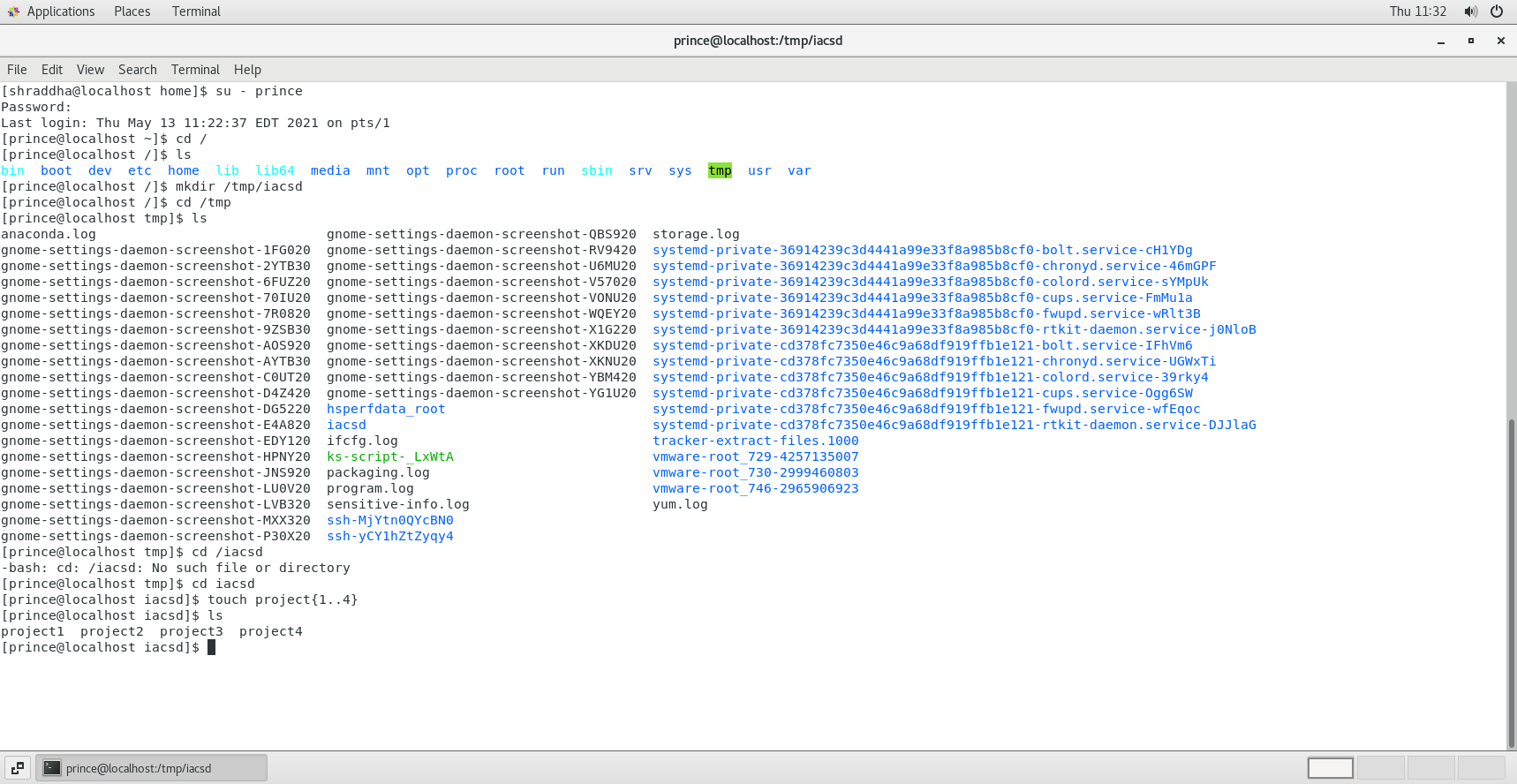
Tom in dac

Bob in dbda

Sam in ditiss



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



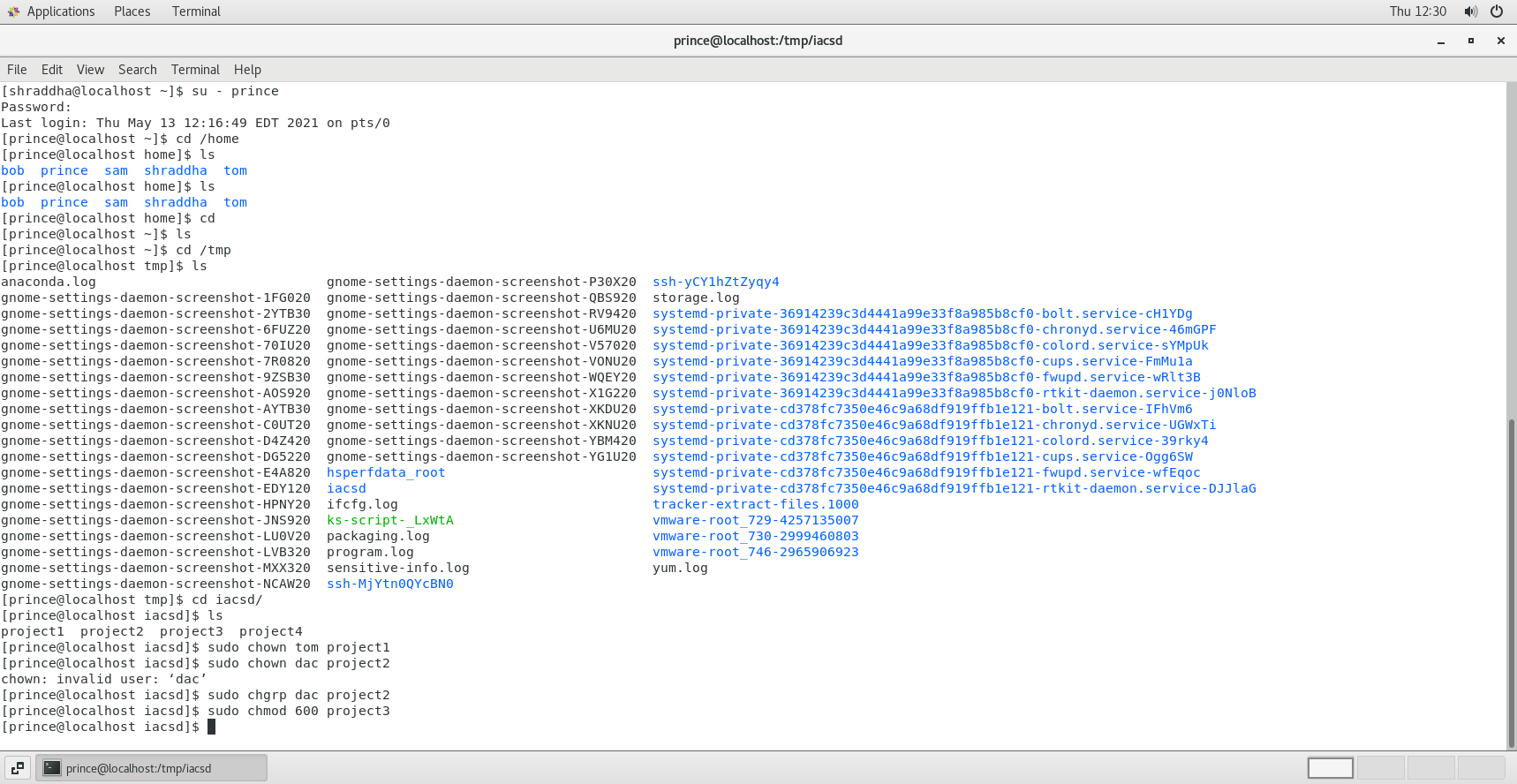
5. assign permissions to project files as below

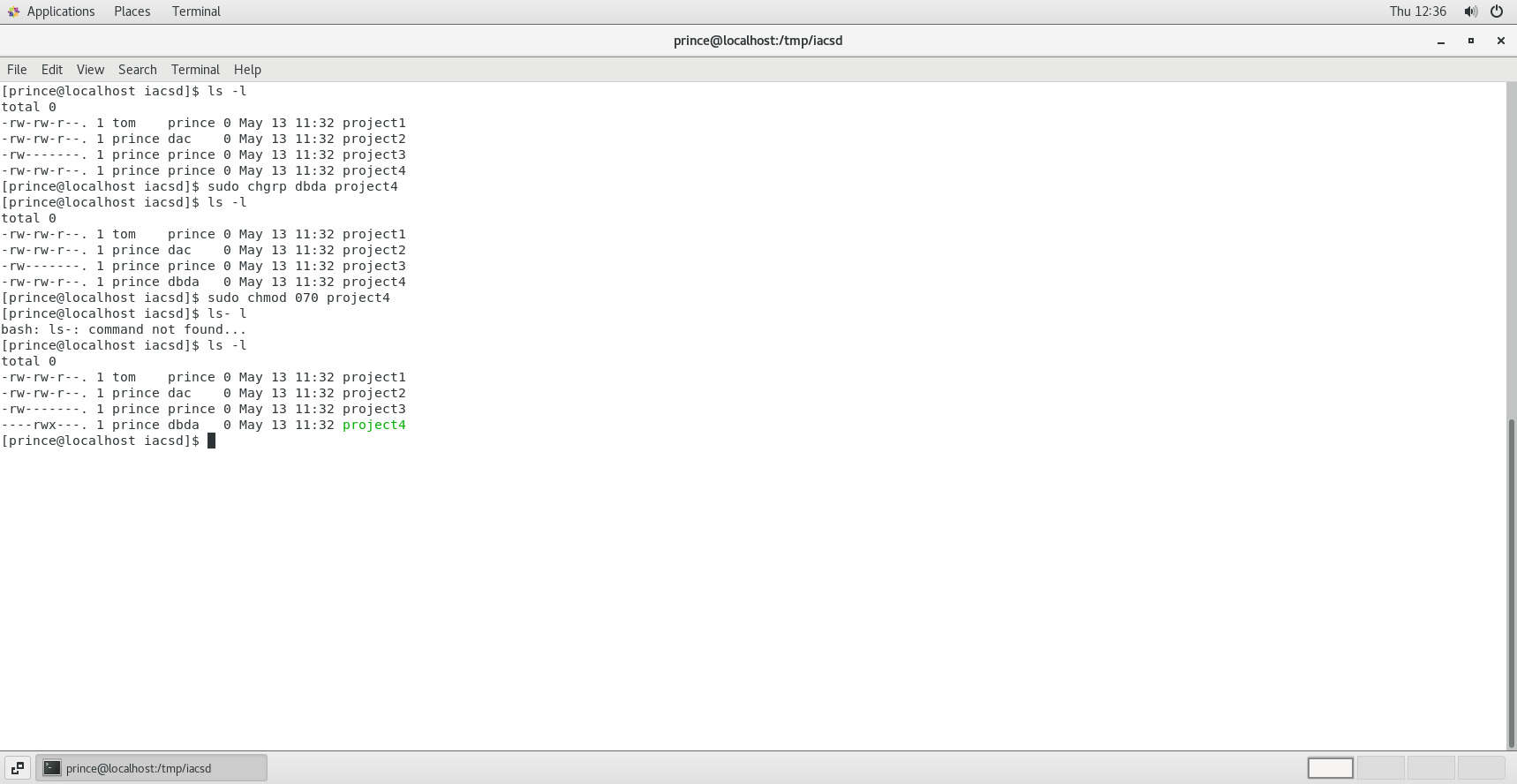
Project-1 – tom should be owner of this

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.





**Assignment 3**

Calculate the average waiting time.

1. **FCFS SCHEDULING-**

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

**Id Arrival time Burst time**

P1 0 4

P2 2 6

P3 3 2

P4 6 1

P5 4 3

**ARRIVAL ORDER: -**P1 , P2 , P3 , P5 , P4

**WAITING TIME:**

P1 = 0

P2 = 4 – 2 = 2

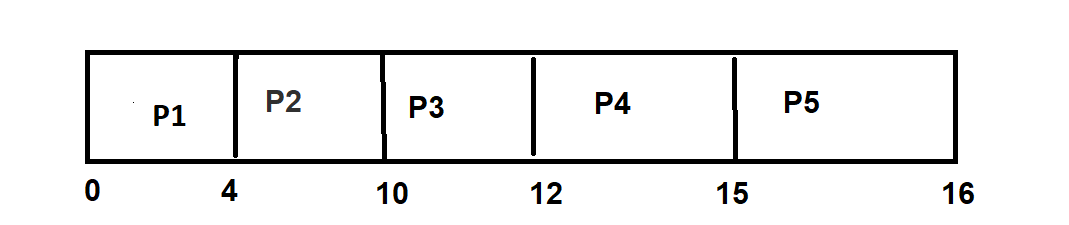
P3 = 10 – 3 = 7

P4 = 15 – 6 = 9

P5 = 12 – 4 = 8

**AVERAGE WAITING TIME: -** (0 + 2 + 7 + 9 + 8 ) / 5 = 5.2

**GNATT CHART**

****

1. **SJF non-preemptive**

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

**Process Id Arrival time Burst time**

P1 0 6

P2 3 2

P3 4 4

P4 5 1

P5 2 3

**ARRIVAL ORDER:** P1 , P5 , P2 , P3 , P4

**WAITING TIME:**

P1 = 0

P2 = (7 -3) = 4

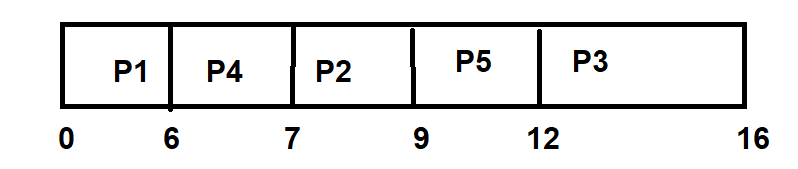
P3 = (12 – 4) = 8

P4 = (6 – 5) = 1

P5 = (9 – 2) = 7

**AVERAGE WAITING TIME:** (0 + 4 + 8 + 1 + 7 ) / 5 = 4

**GNATT CHART**



**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 |

**Process Id Arrival time Burst time**

P1 4 6

P2 3 5

P3 2 4

P4 1 3

P5 0 2

**ARRIVAL ORDER:**

P5 , P4 , P3 , P2 , P1

**WAITING TIME:**

P1 = 14 – 4 = 10

P2 = 9 – 3 = 6

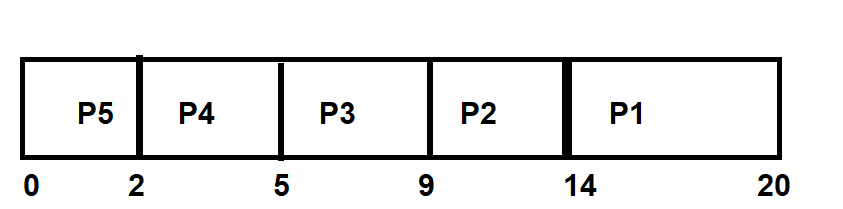
P3 = 5 – 2 = 3

P4 = 2 – 1 = 1

P5 = 0

**AVERAGE WAITING TIME:** (10 + 6 + 3 + 1 + 0 ) / 5 = 4

**GNATT CHART**

****

**4) Round Robin**

**Q=3**

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

**Q=3**

**Process Id Arrival time Burst time**

P1 0 9

P2 1 4

P3 2 3

**ARRIVAL ORDER:**

P1 , P2 , P3

**WAITING TIME:**

P1 = (13 – 6 – 0) = 7

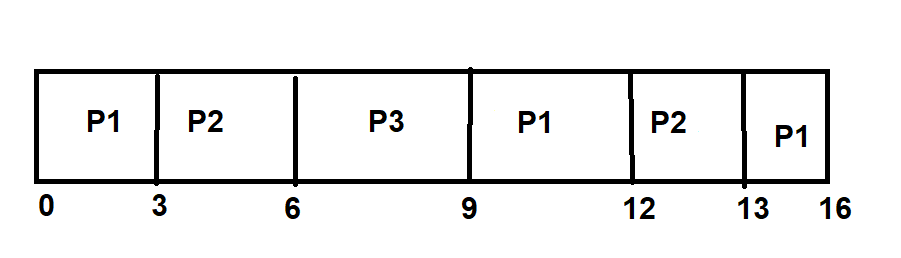
P2 = (12 – 3 – 1) = 8

P3 = (6 – 2) = 4

**AVERAGE WAITING TIME:**

( 7 + 8 + 4 ) / 3 = 6.33

**GNATT CHART**

****

**Q = 5**

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

**Process Id Arrival time Burst time**

P1 0 10

P2 5 20

P3 10 15

P4 15 5

**ARRIVAL ORDER:**

P1 , P2 , P3 , P4

**WAITING TIME:**

P1 = (10 – 5 – 0) = 5

P2 = (45 – 15 – 5) = 25

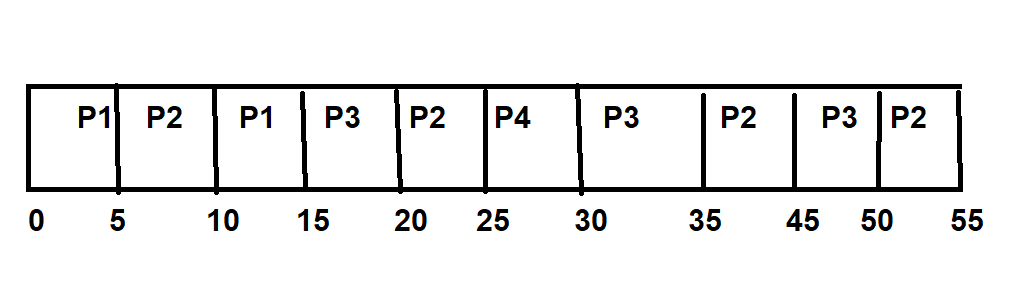
P3 = (40 – 10 - 10) = 20

P4 = 25 – 15 = 10

**AVERAGE WAITING TIME:**

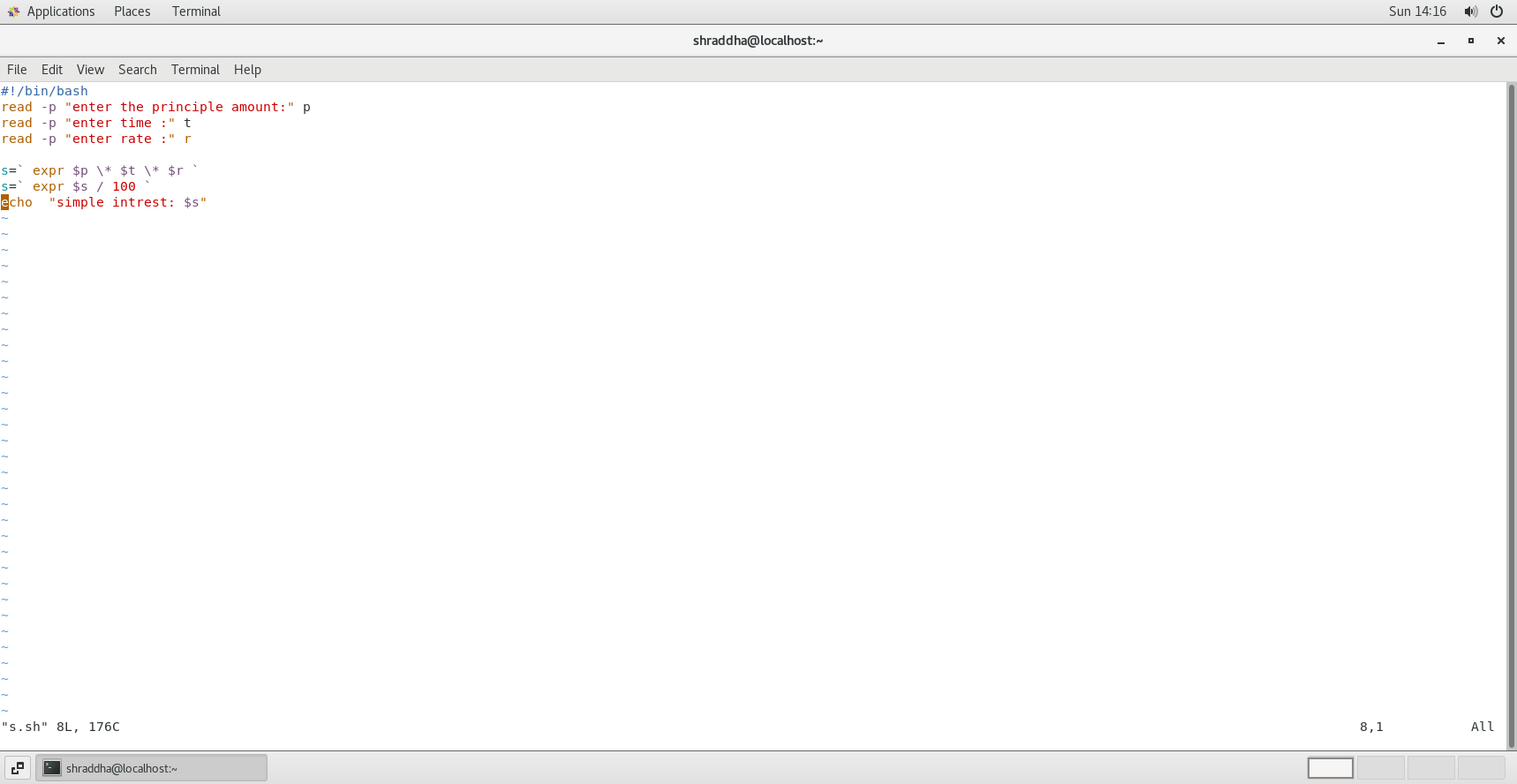
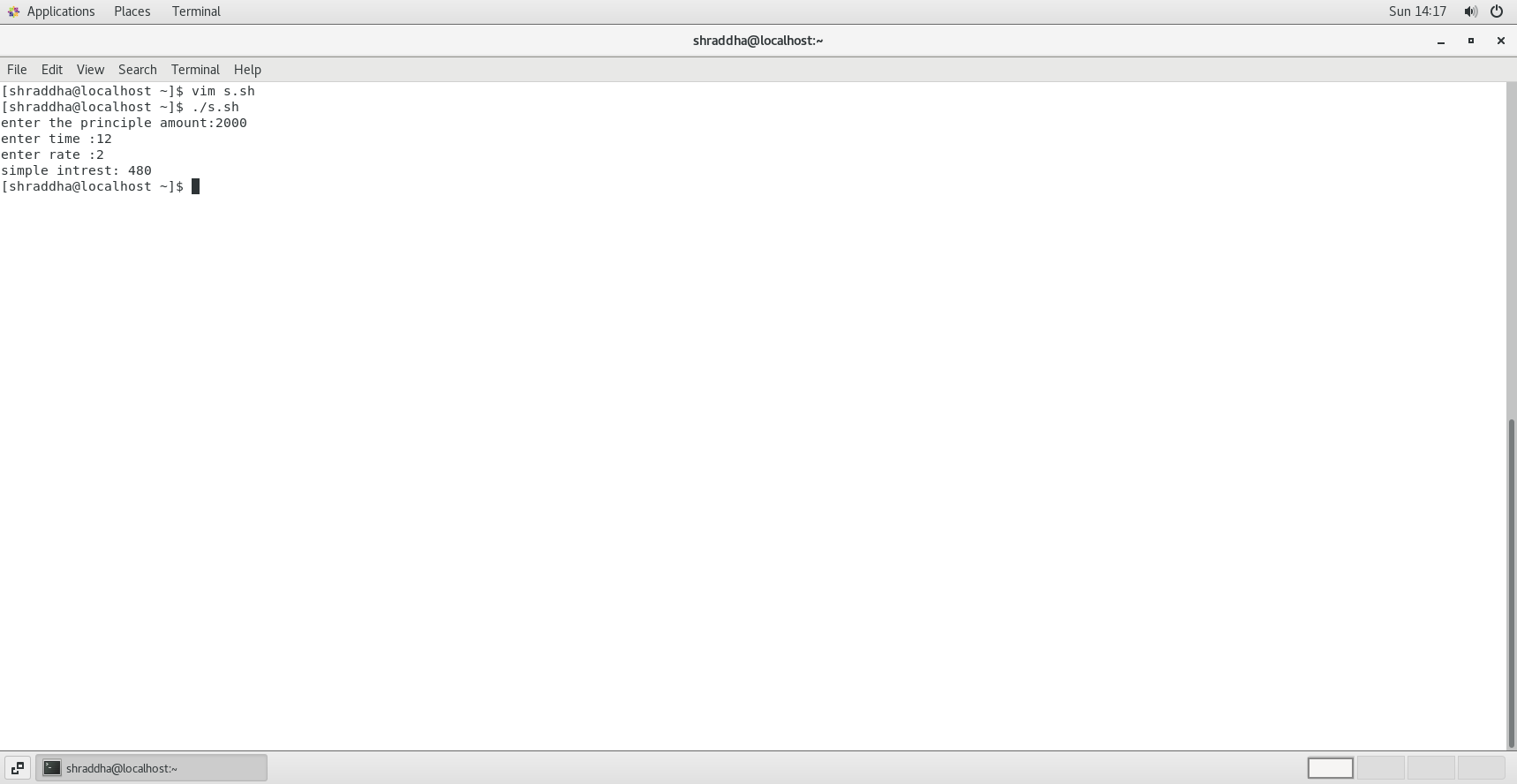
( 5 + 25 + 20 + 10 ) / 4 = 15

**GNATT CHART**



**Assignment 4**

1. Write a shell script to calculate simple interest.



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

Salary = basic + dp + da +hra +ma –pf

basic – to be taken as input

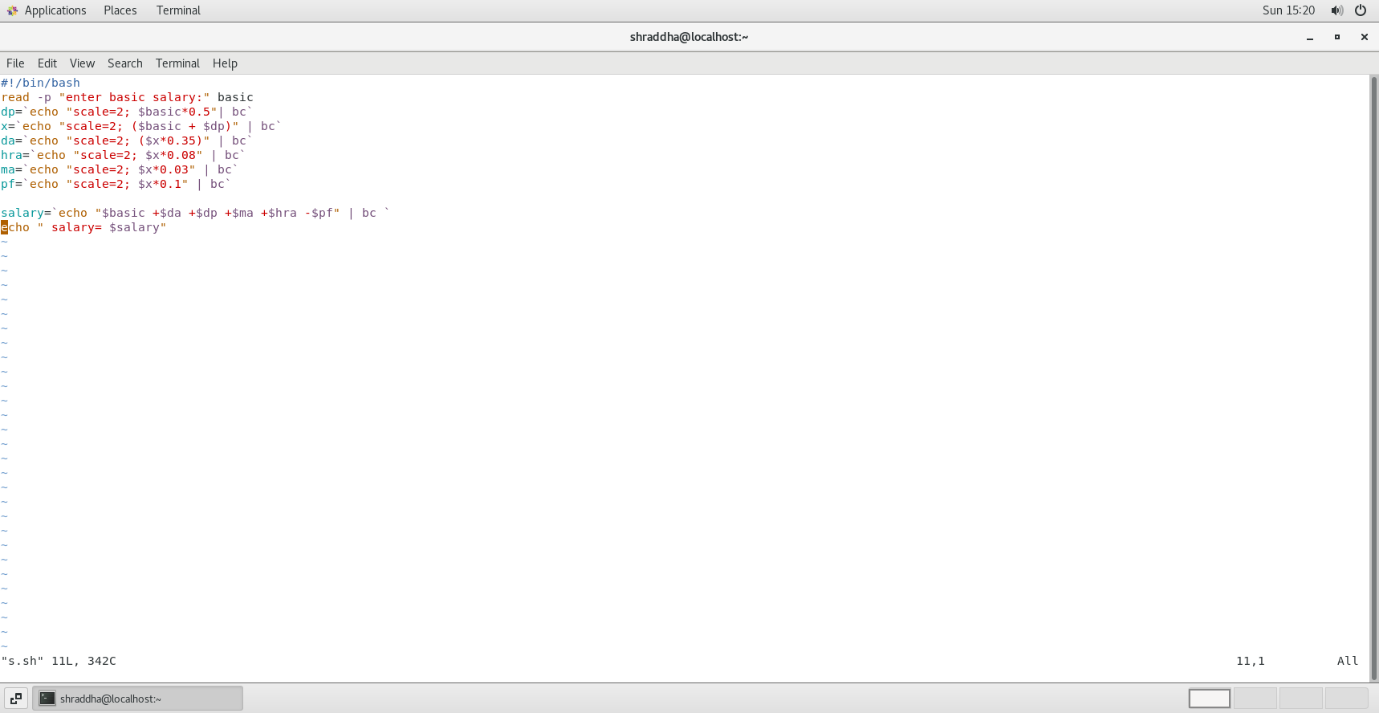
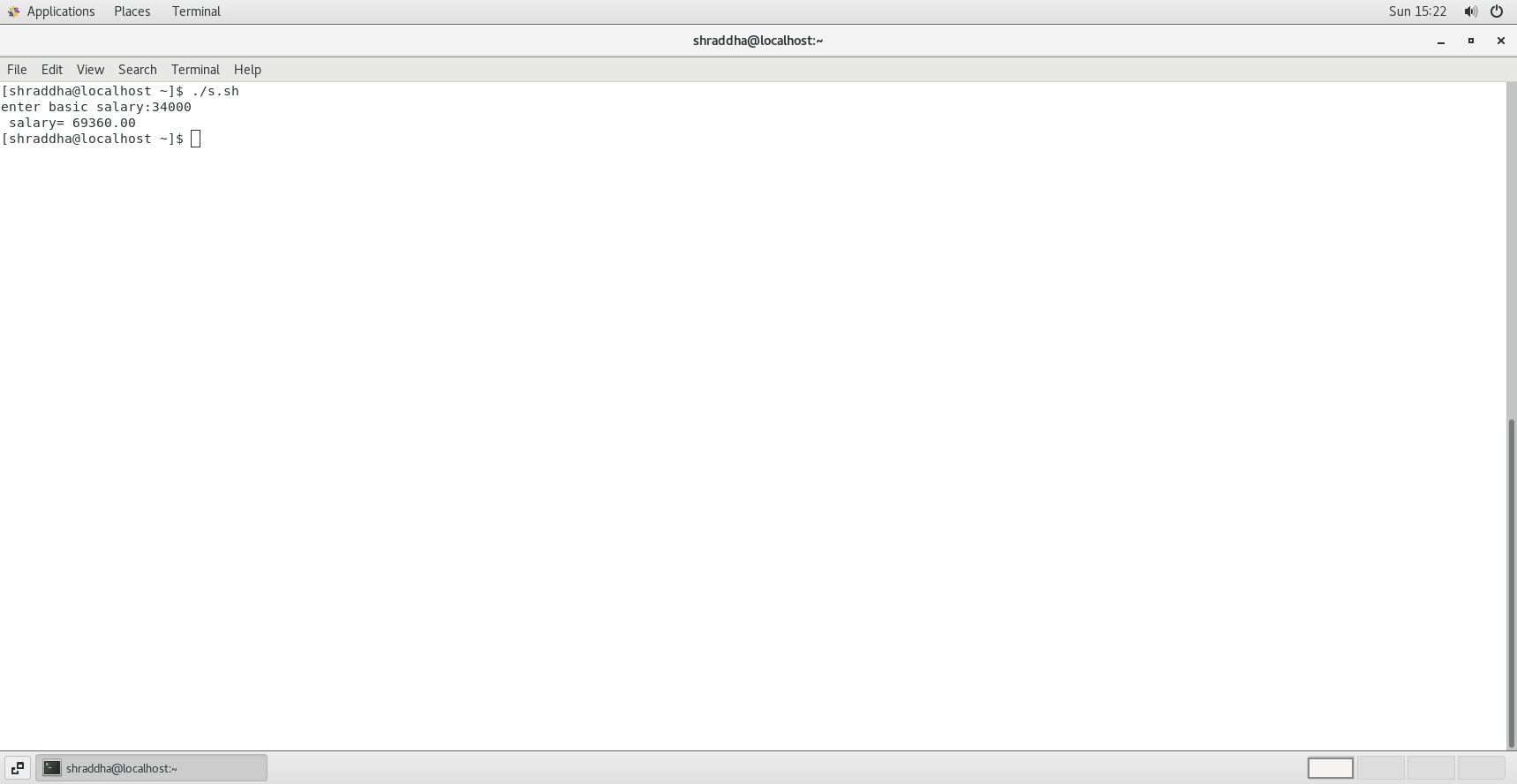
dp - 50 % of basic

da - 35 % of (basic + dp)

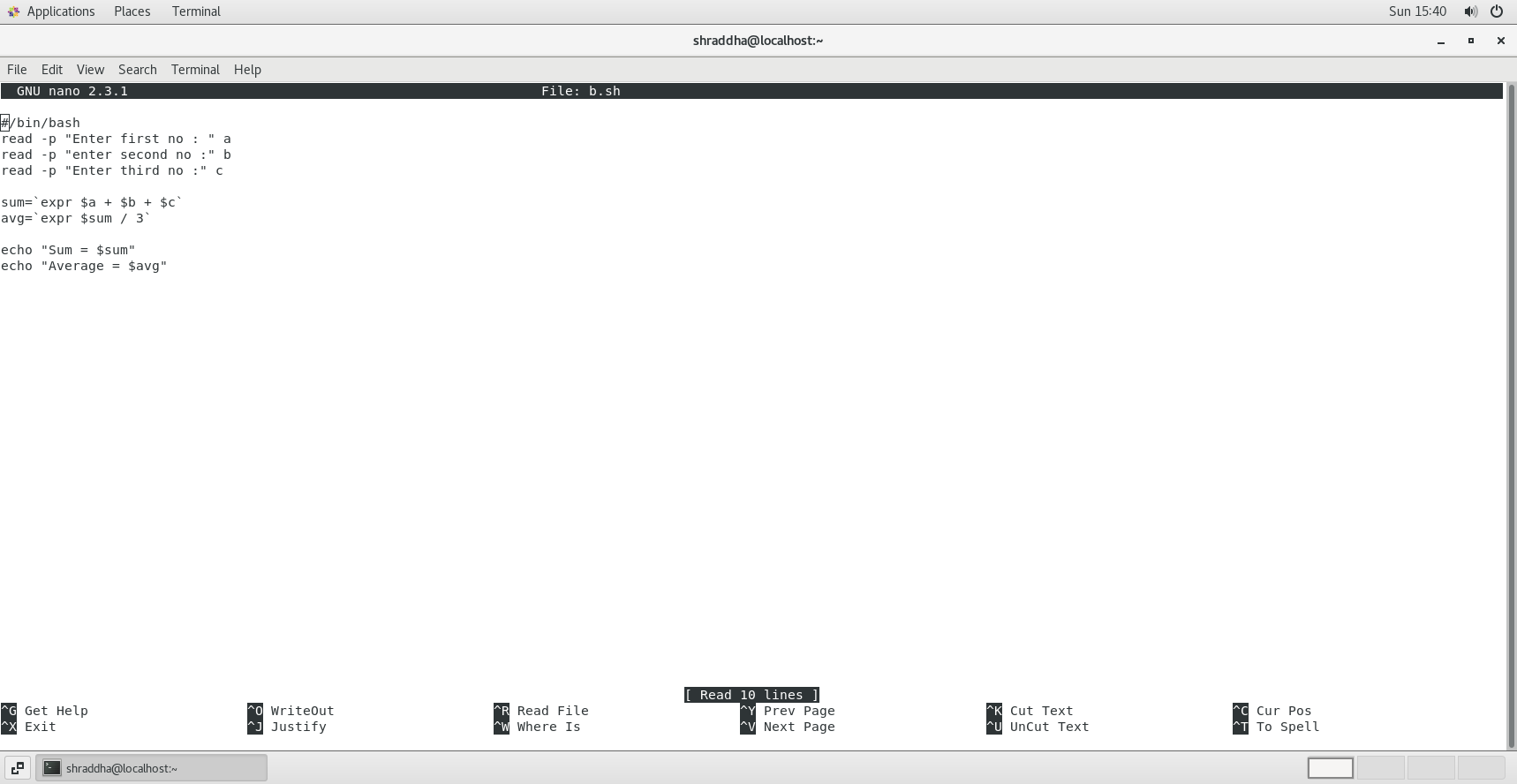
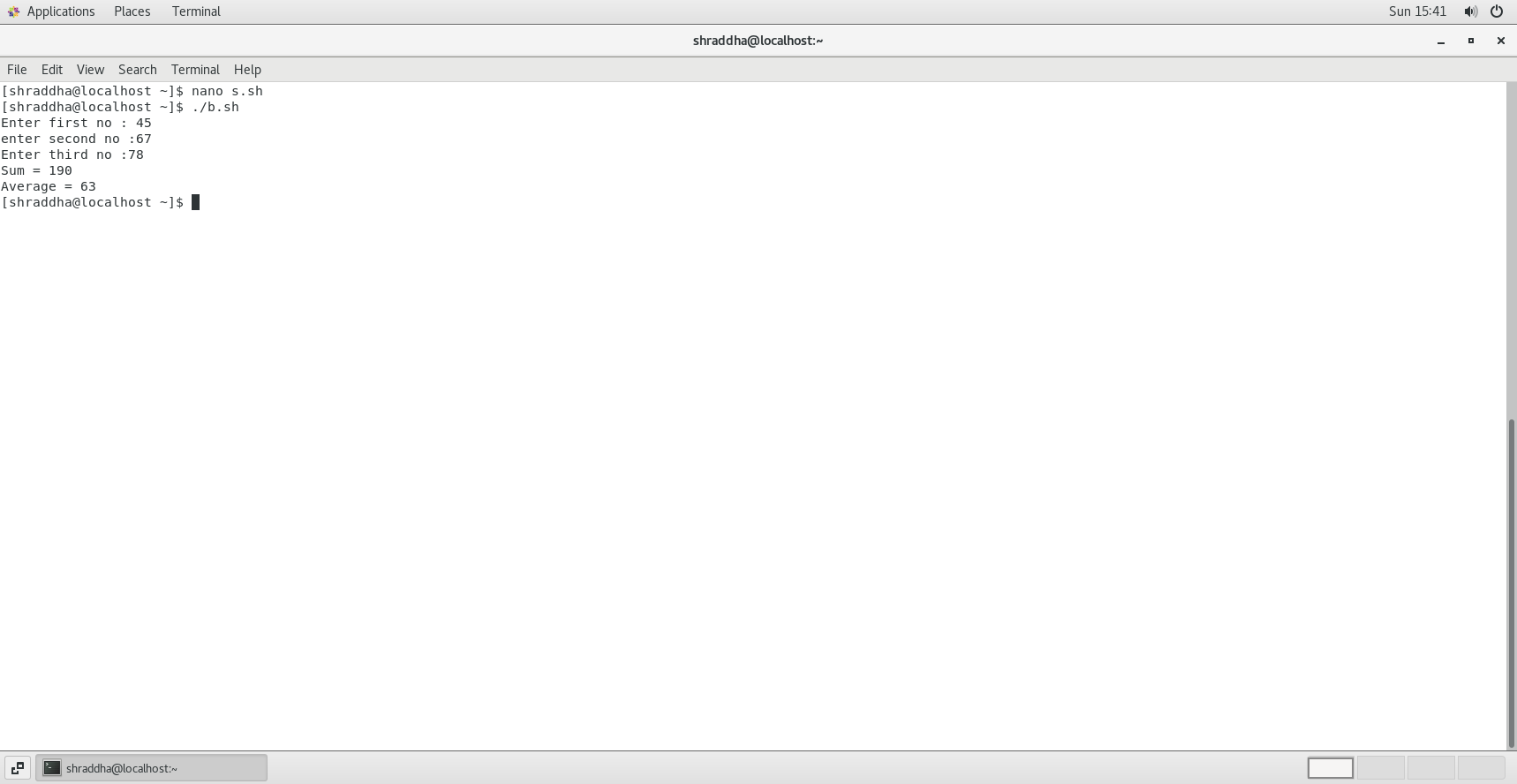
hra - 8 % of (basic + dp)

ma - 3 % of (basic + dp)

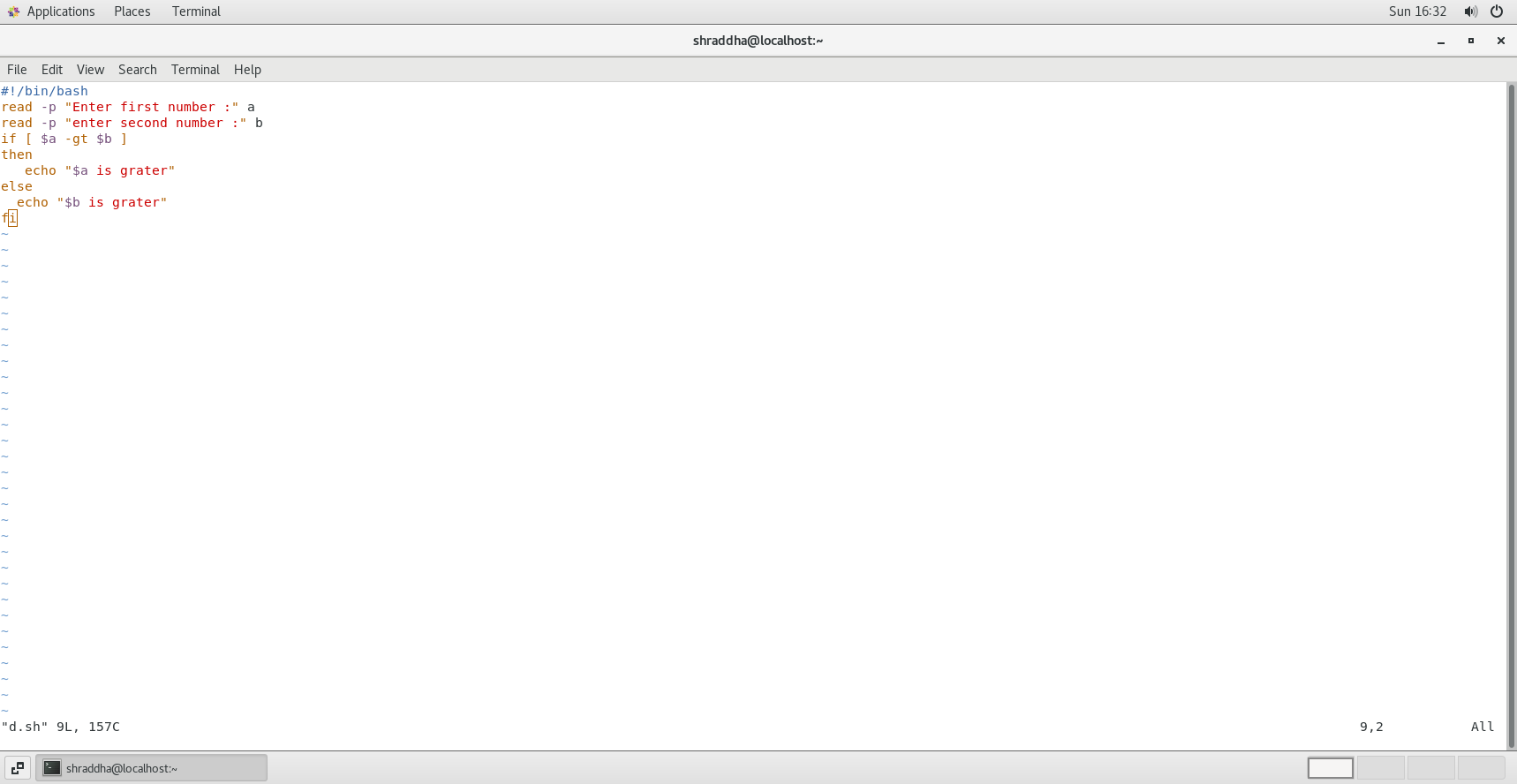
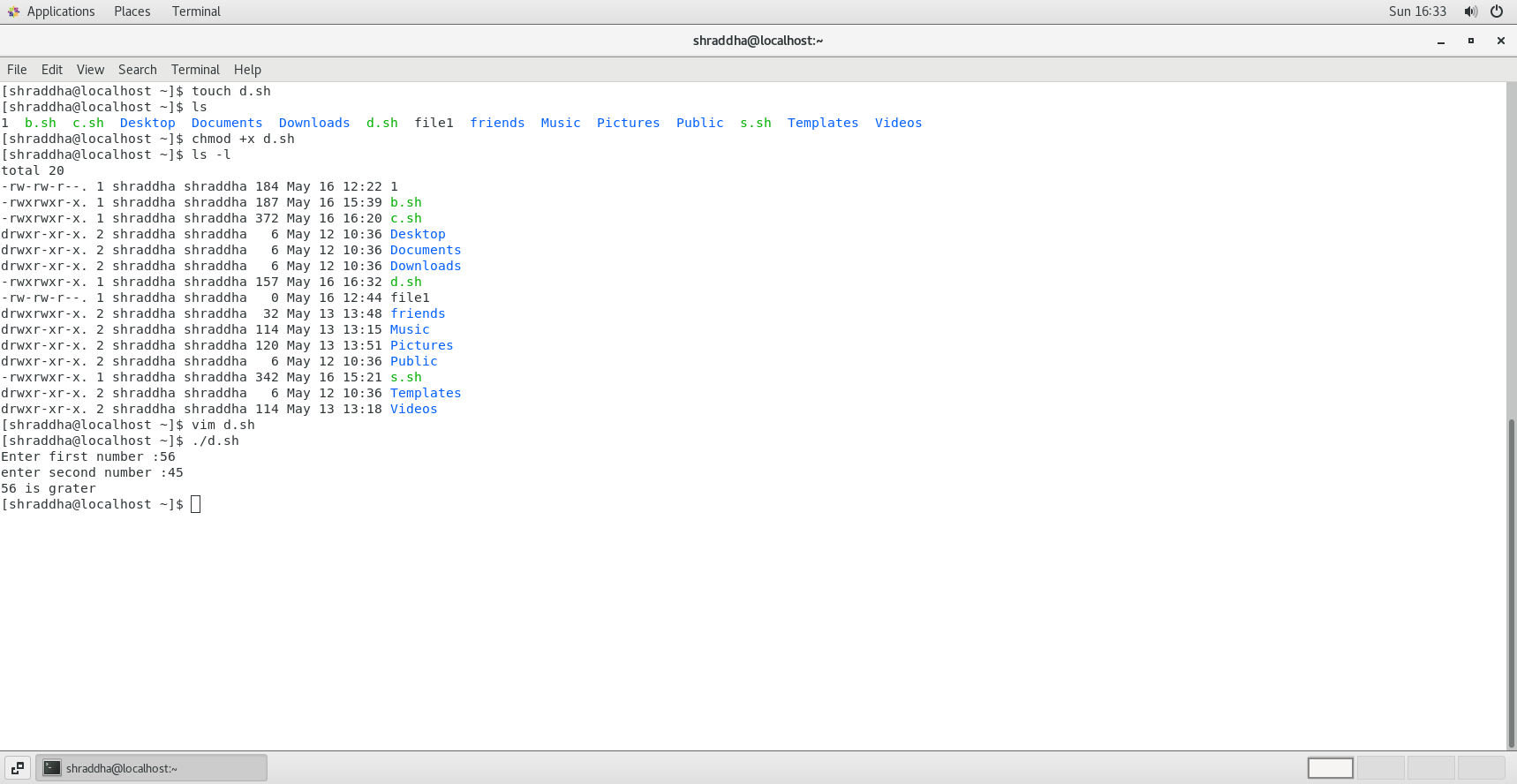
pf - 10% of (basic + dp)



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

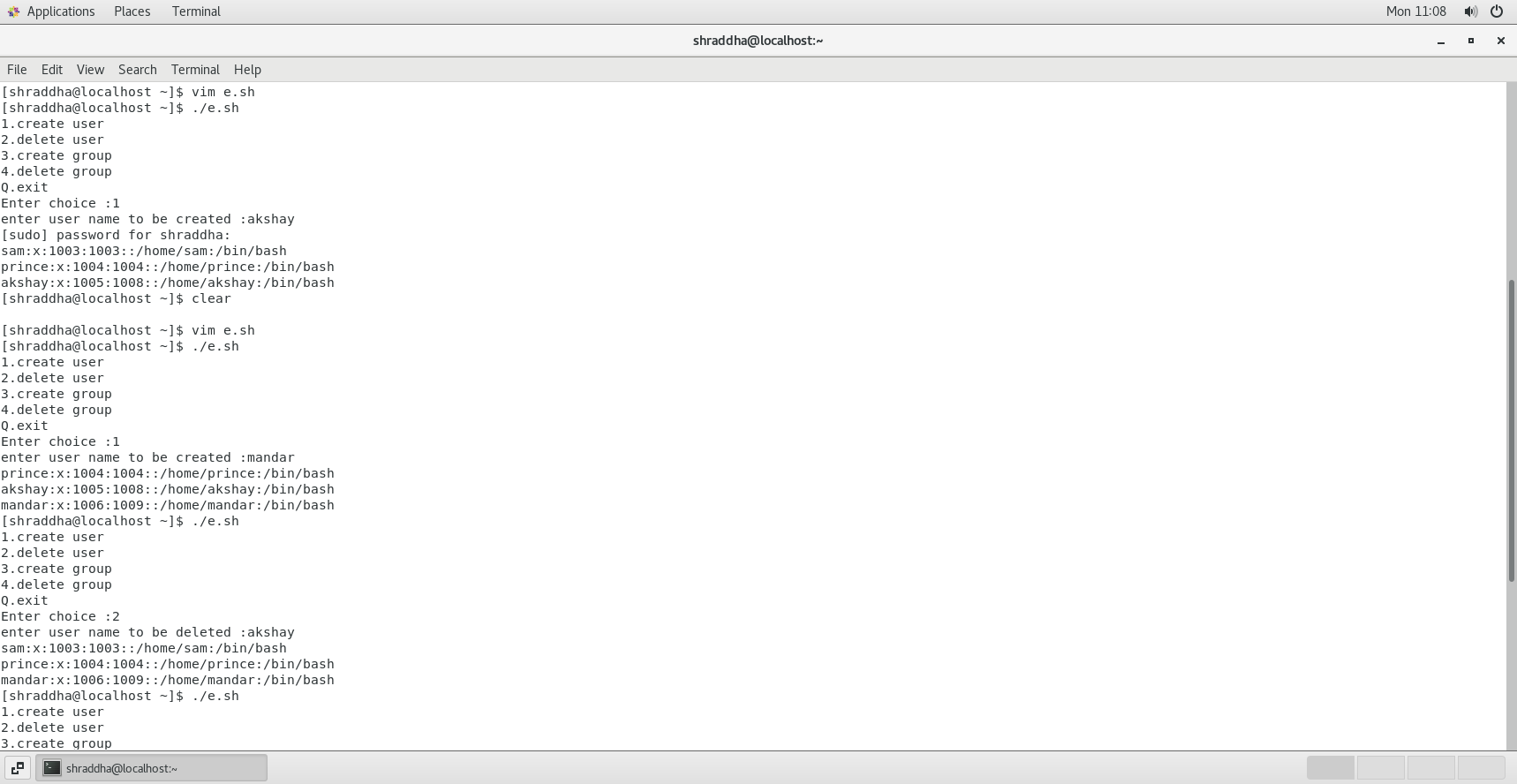
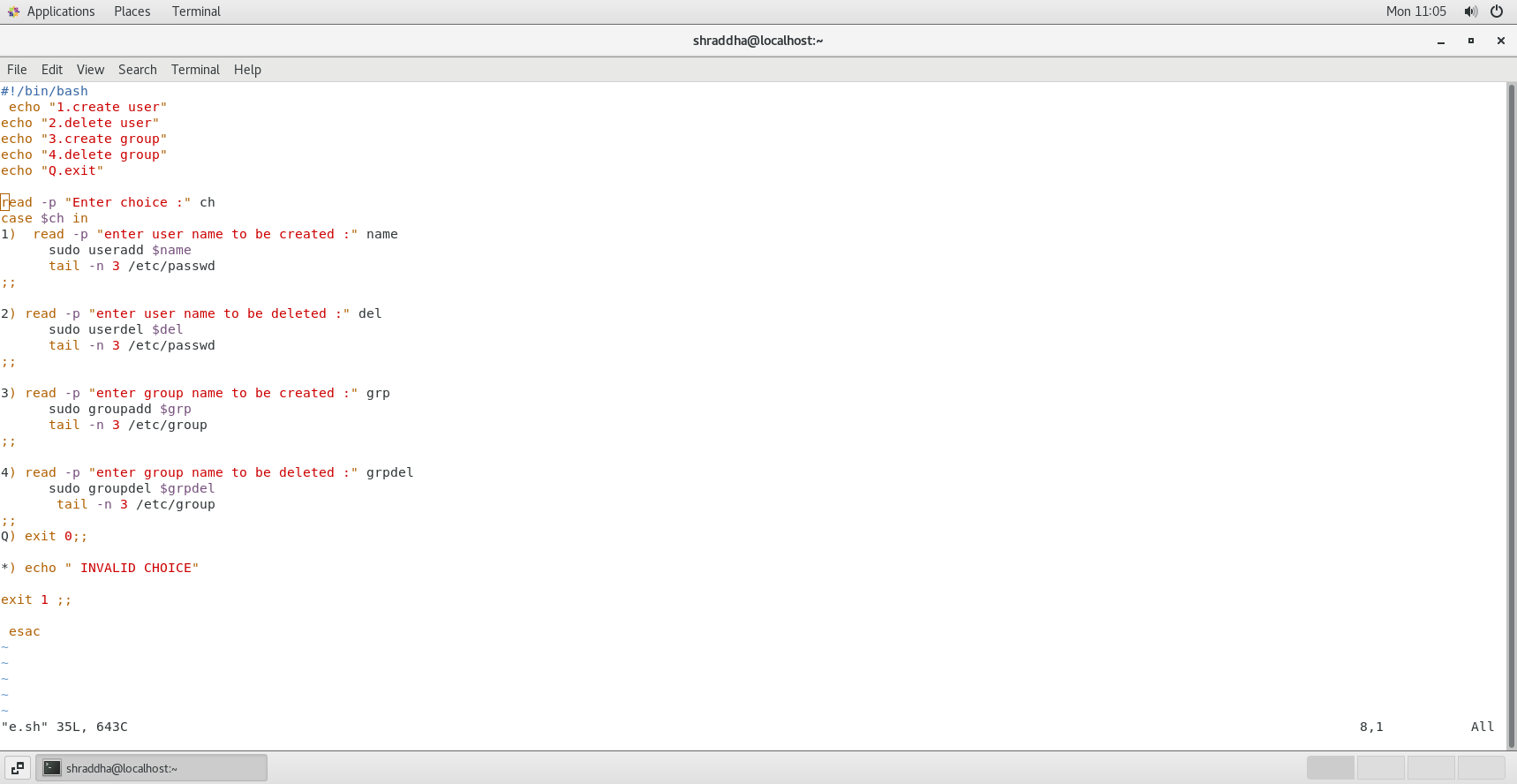


4) Write a shell script to accept 2 numbers and display which number is greater



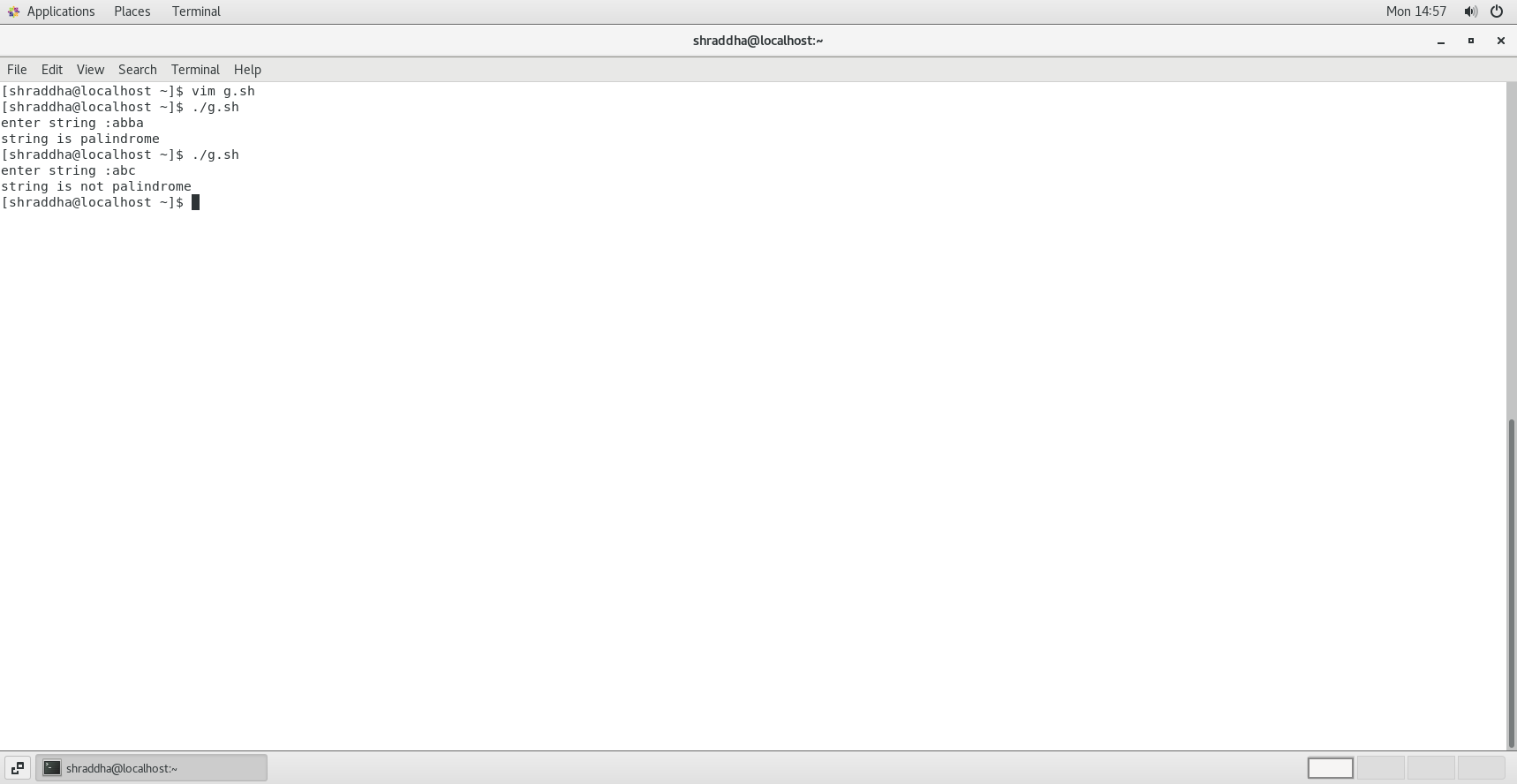
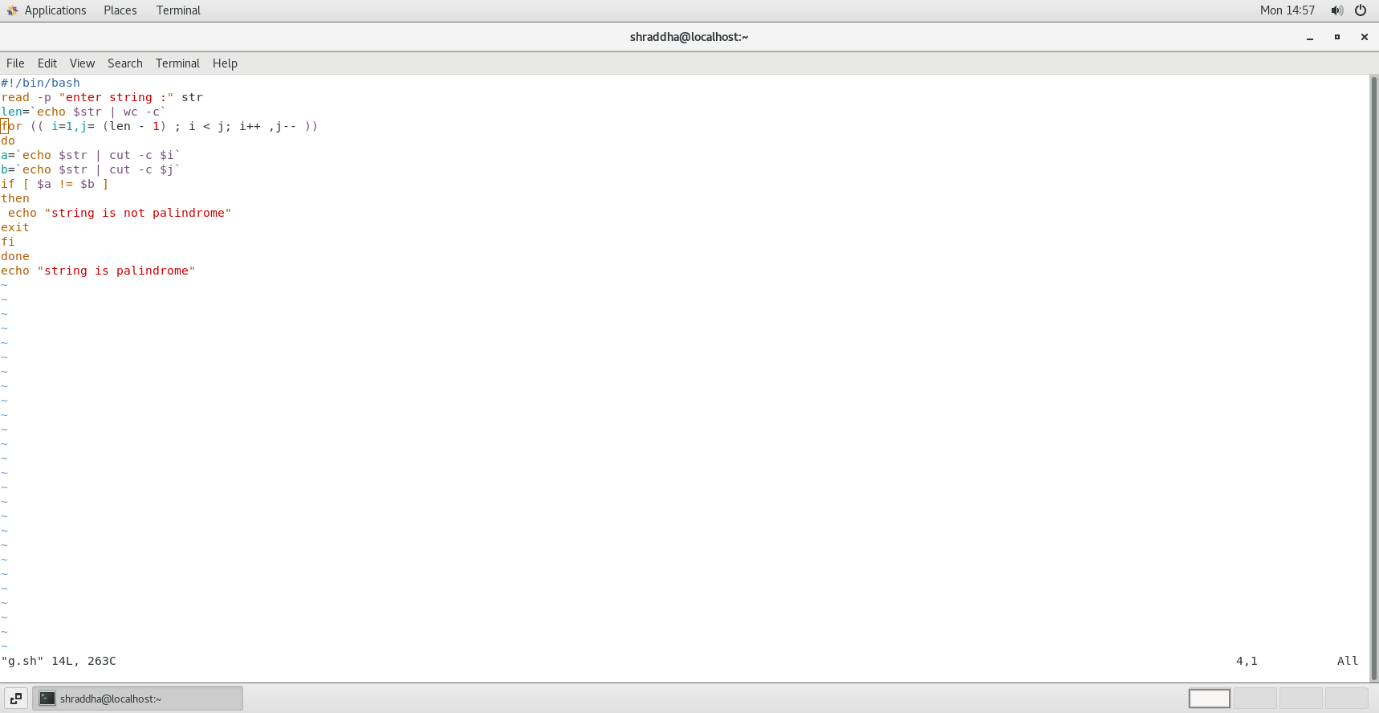
5) Create a script to

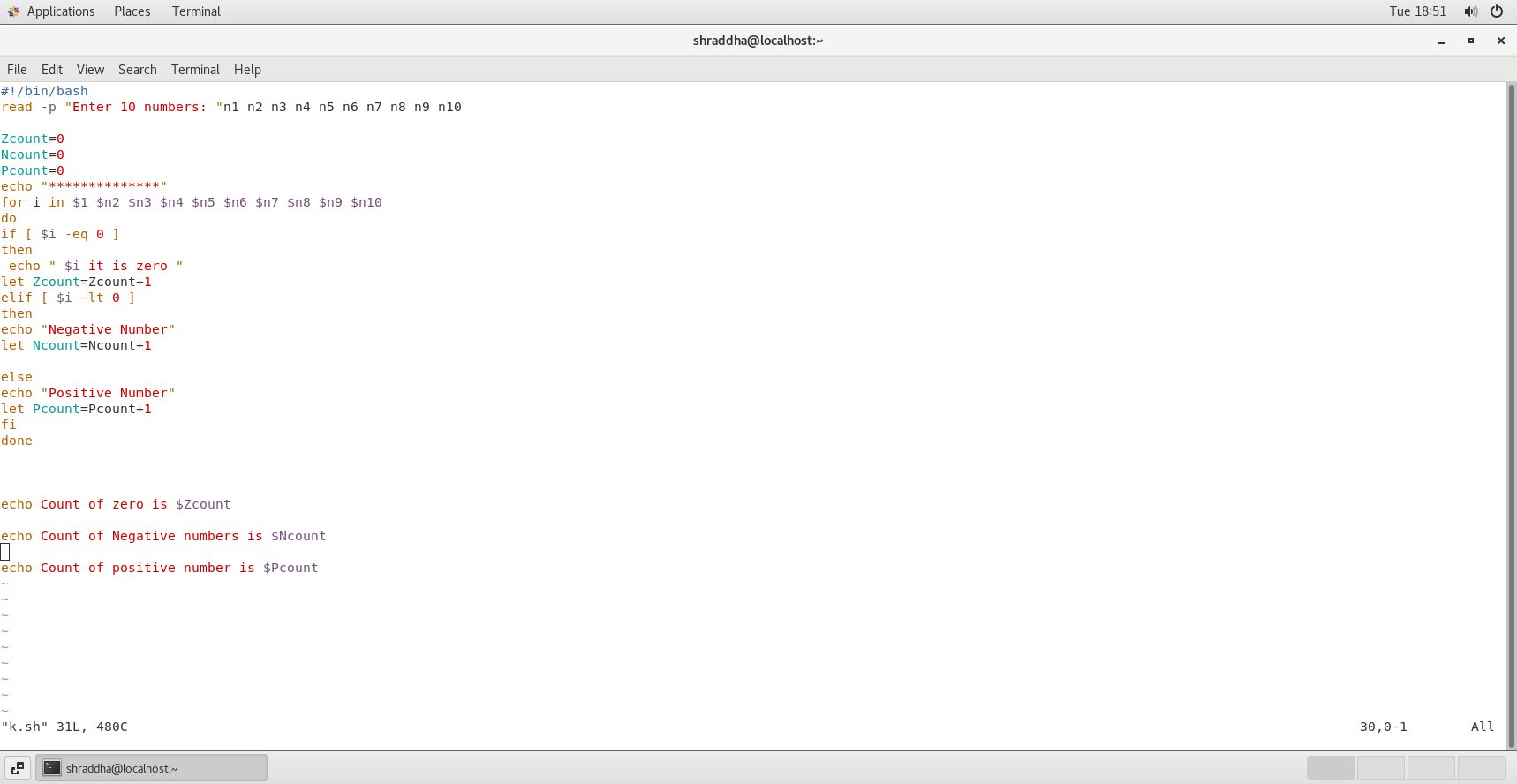
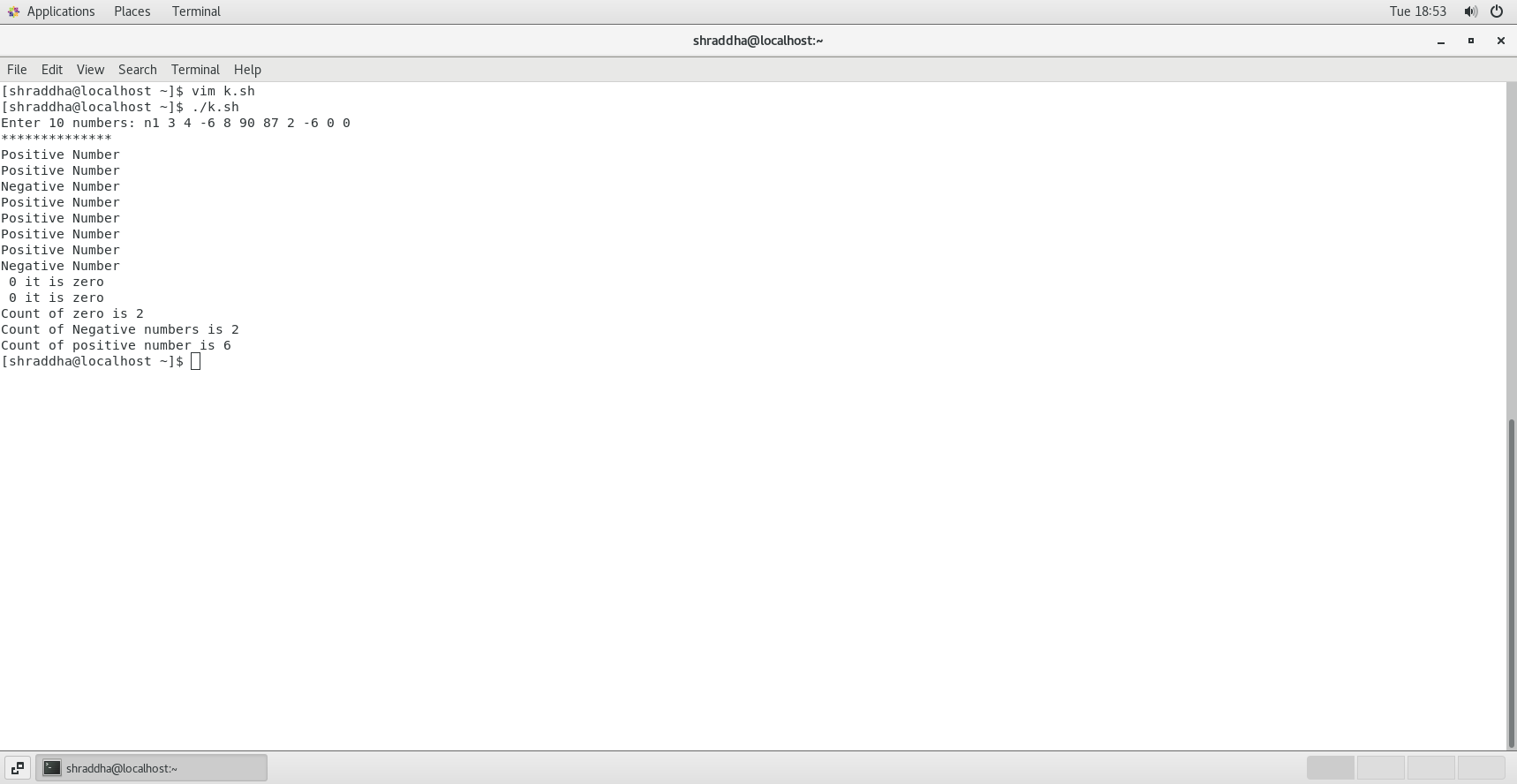
Create user , Delete user , Create group , delete Group using case



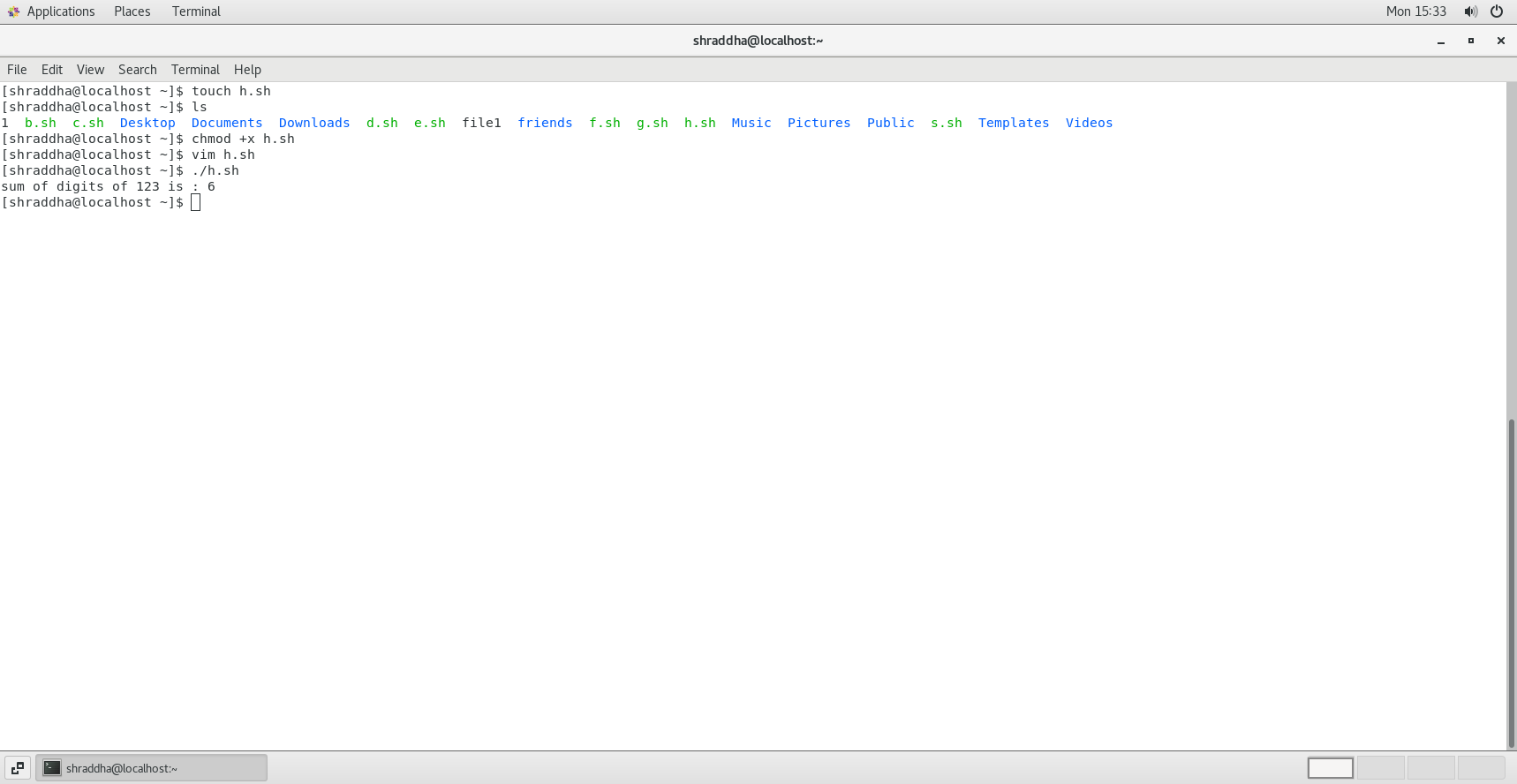
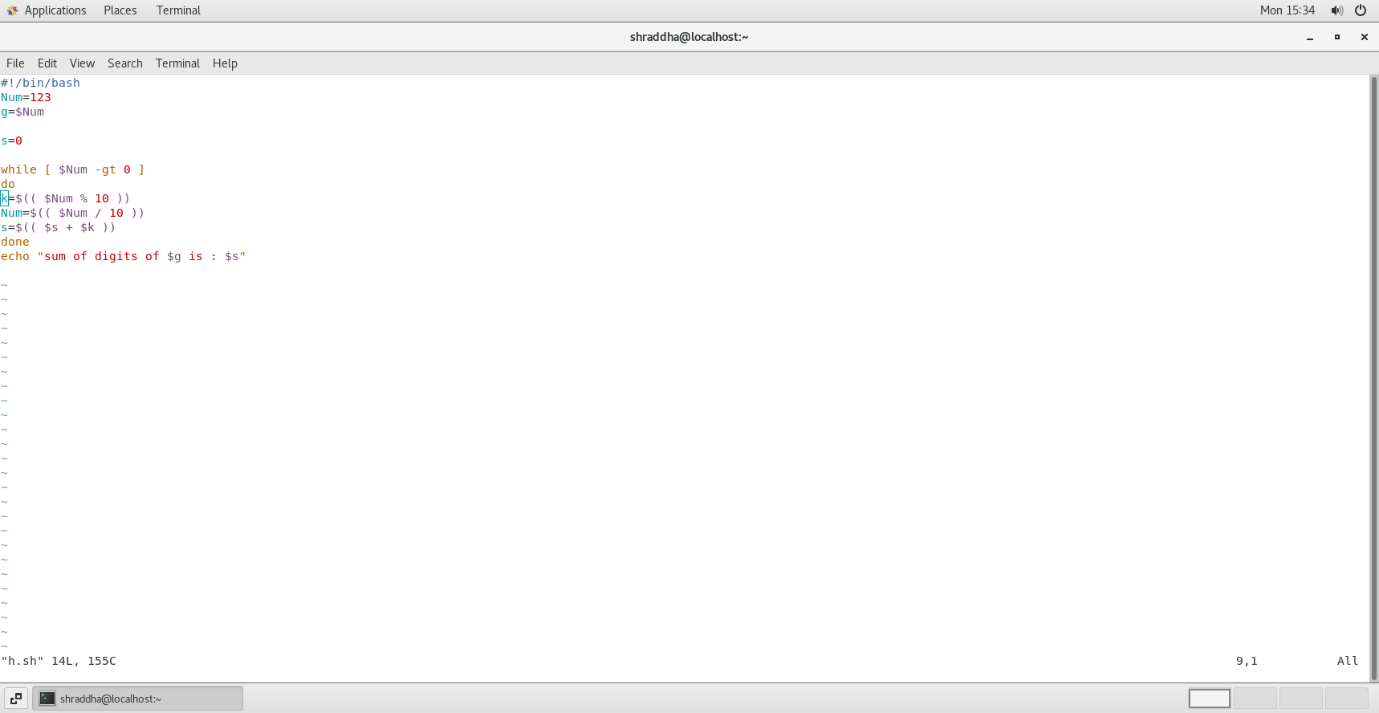
**Assignment 5**

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

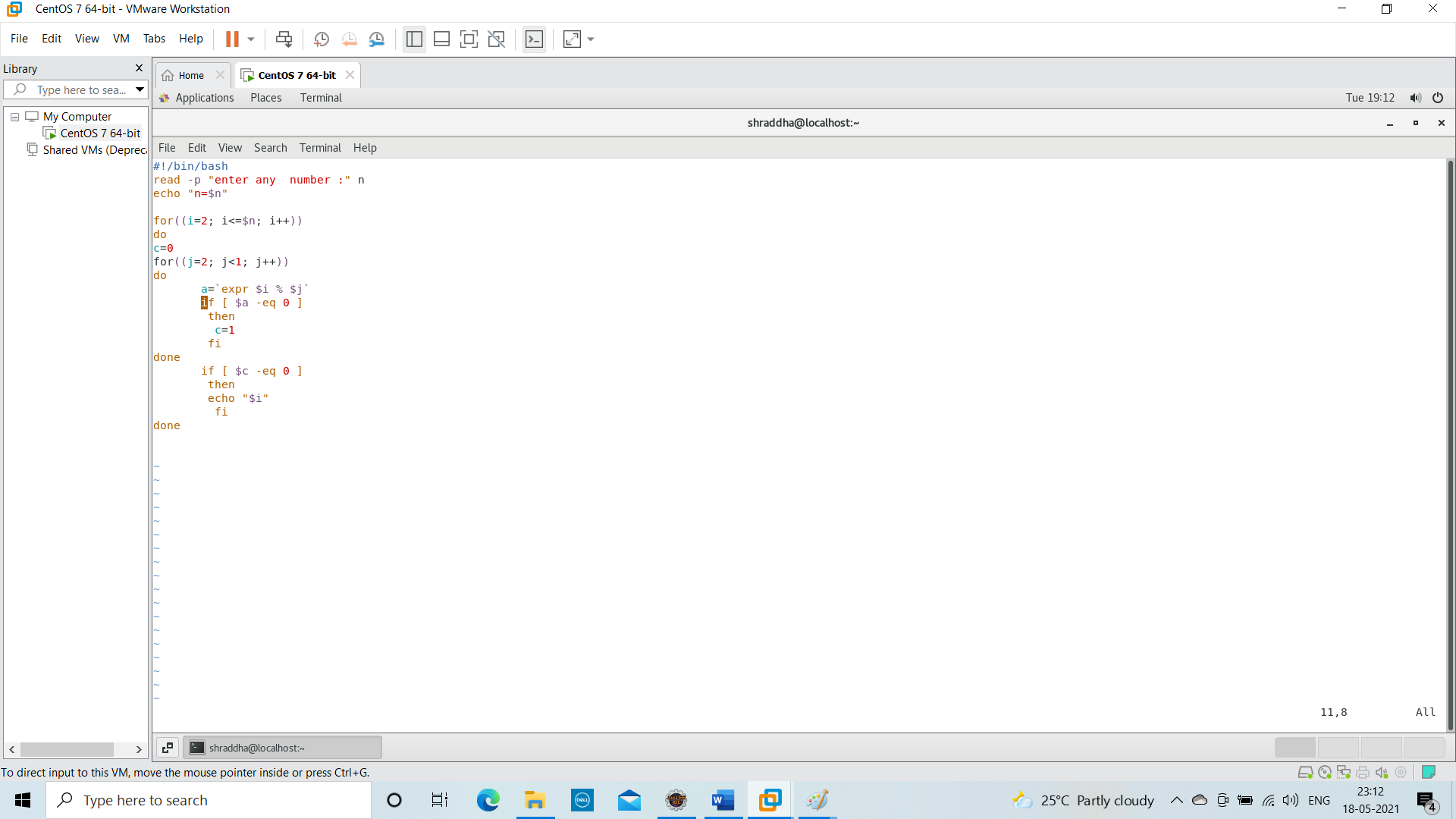
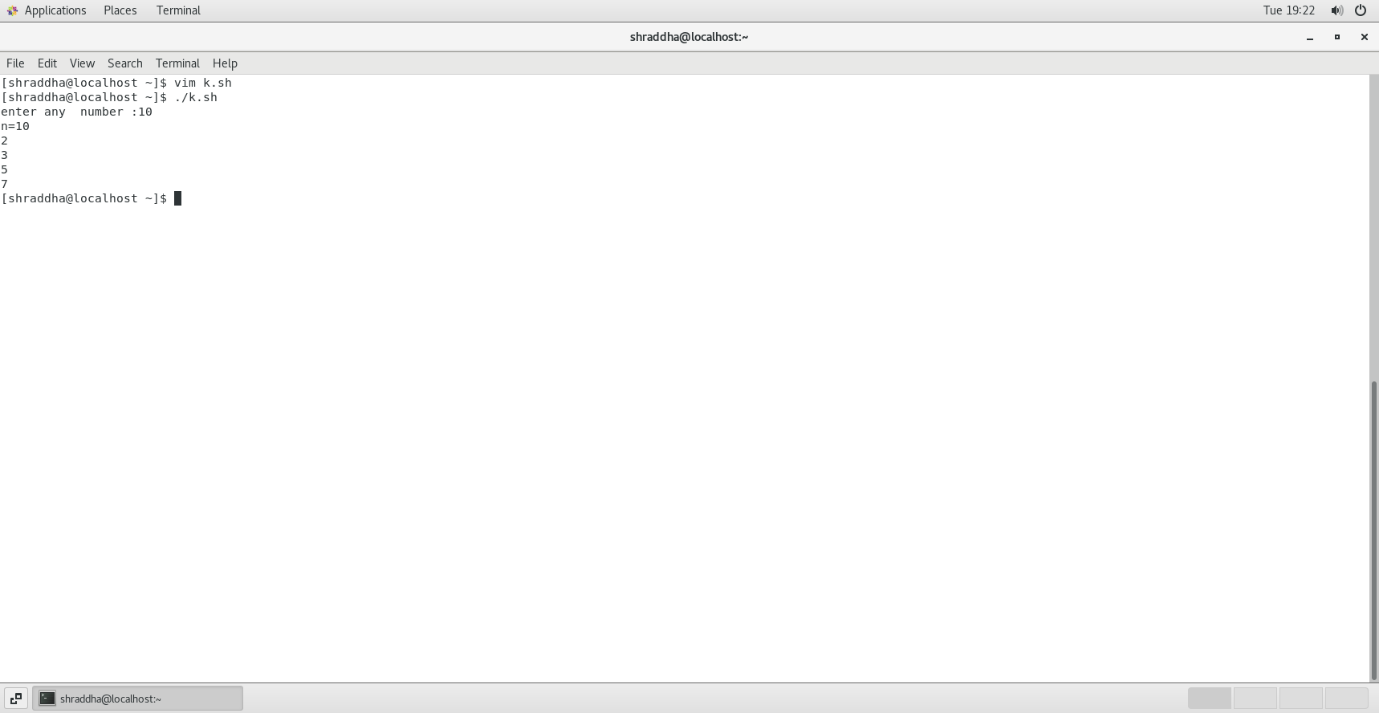


2) Write a shell script to accept 10 numbers and tell how many are +tive, -tive and zero

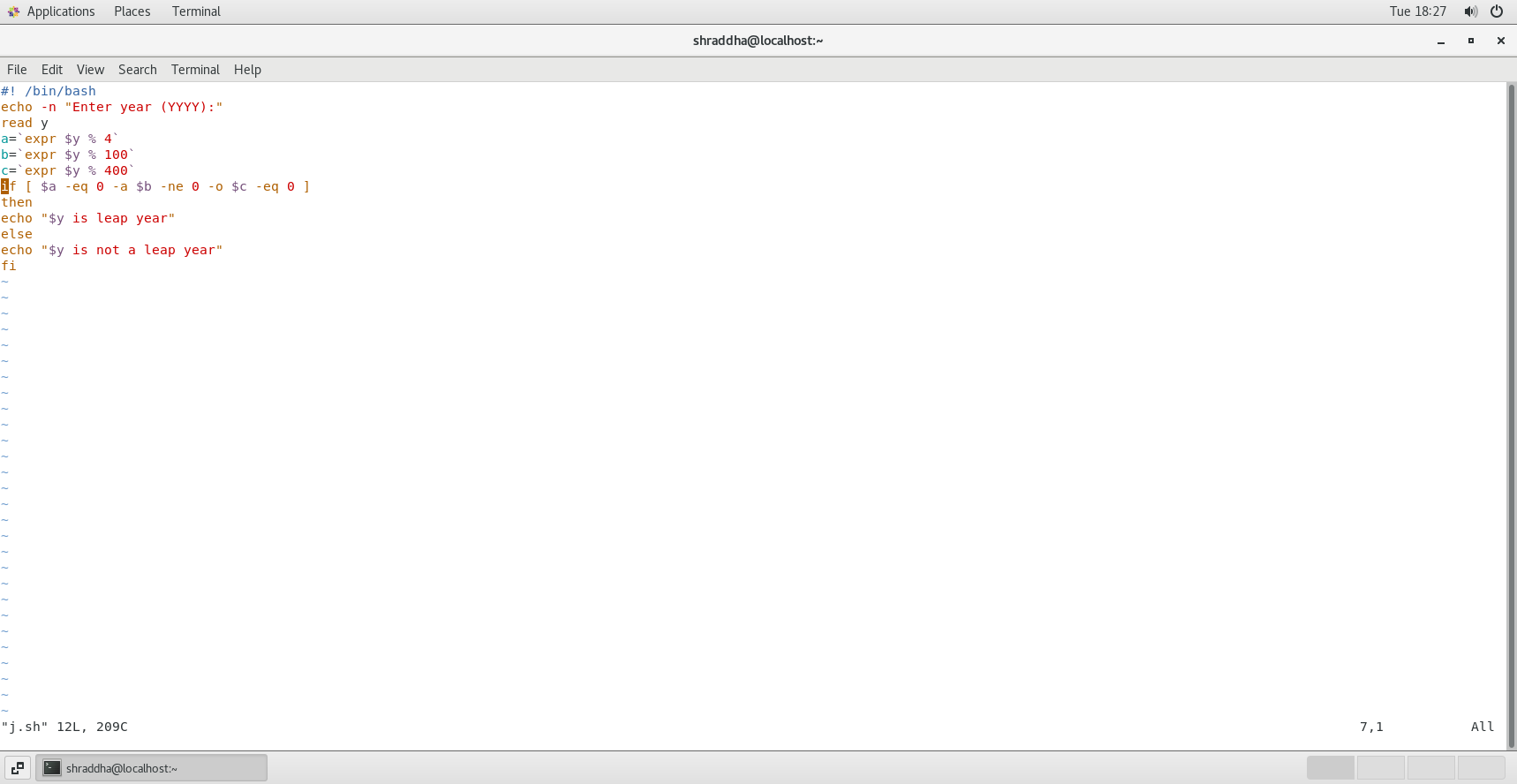
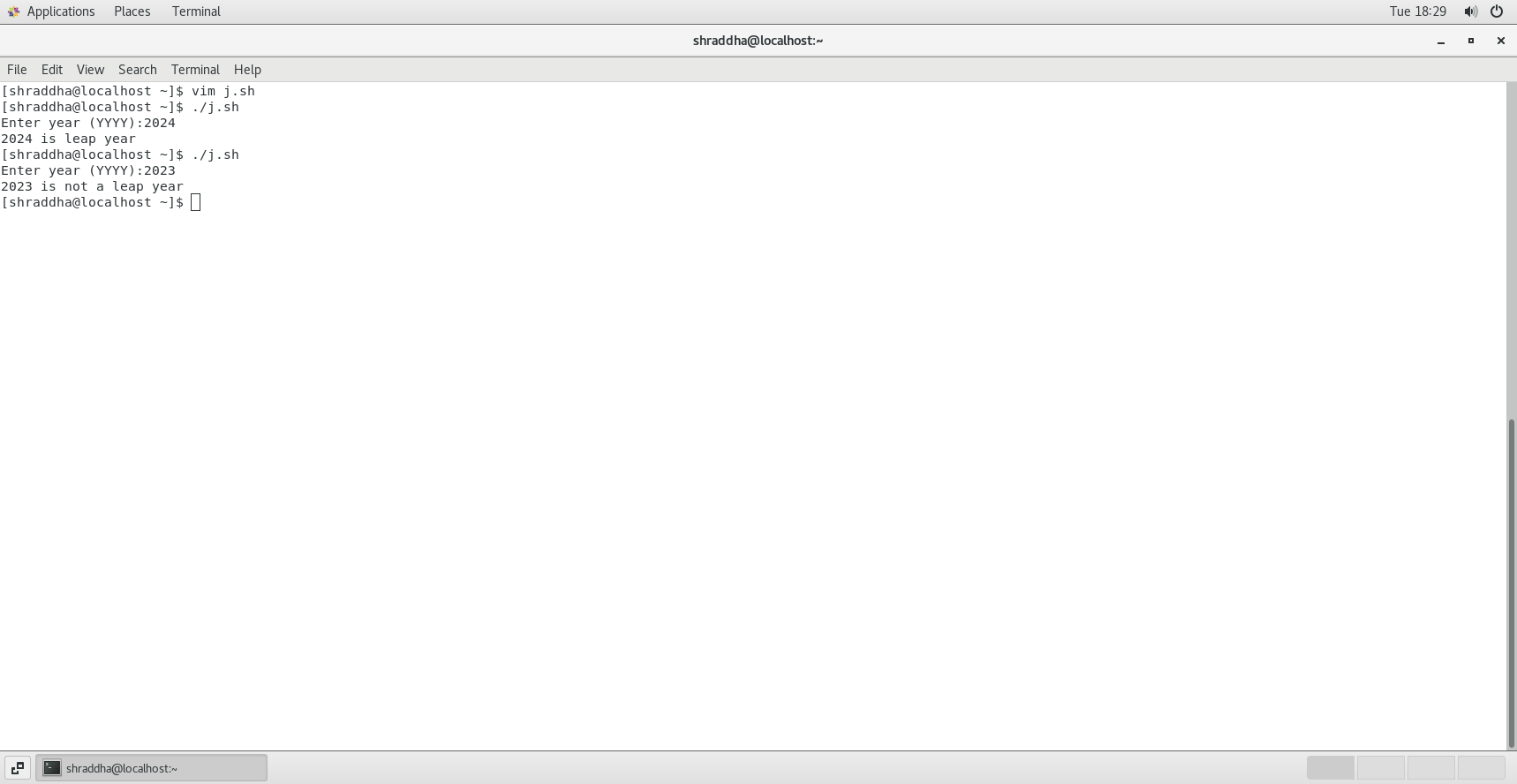
3) Write a shell script to print given number’s sum of all digits (eg. If number is 123, then it’s sum of all digits will be 1+2+3=6)



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

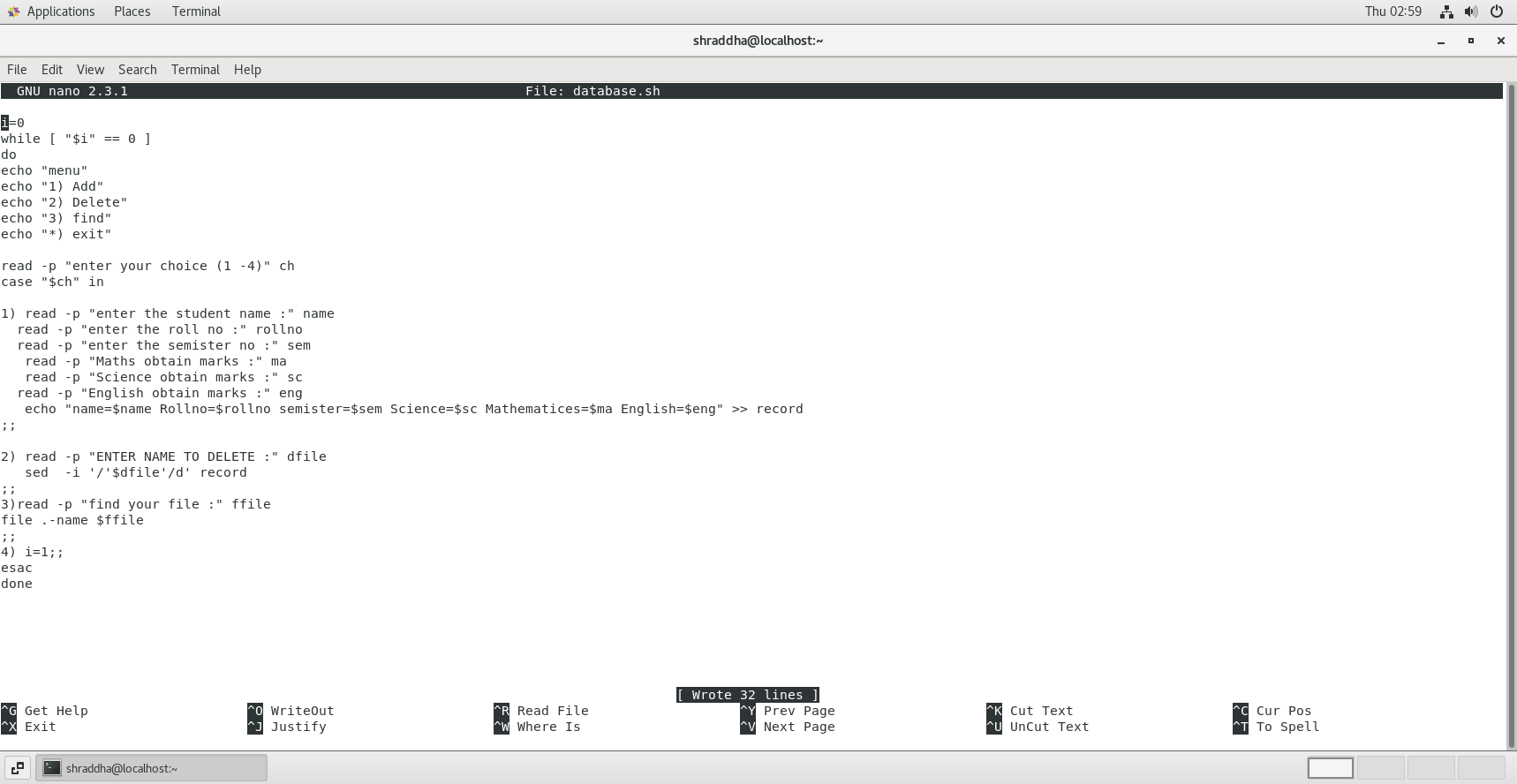
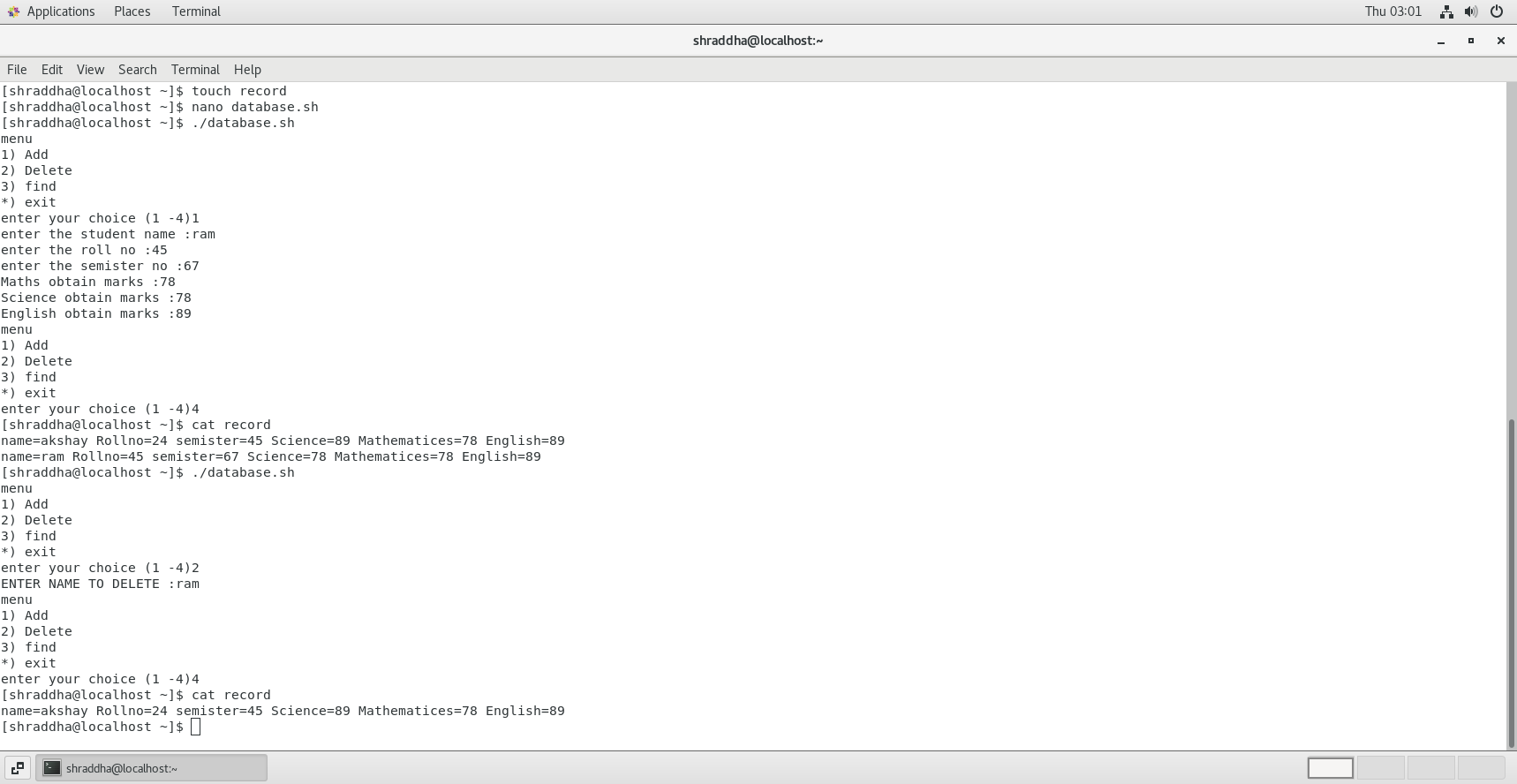


5) Write a shell script to find whether a given year is leap year or not



**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.

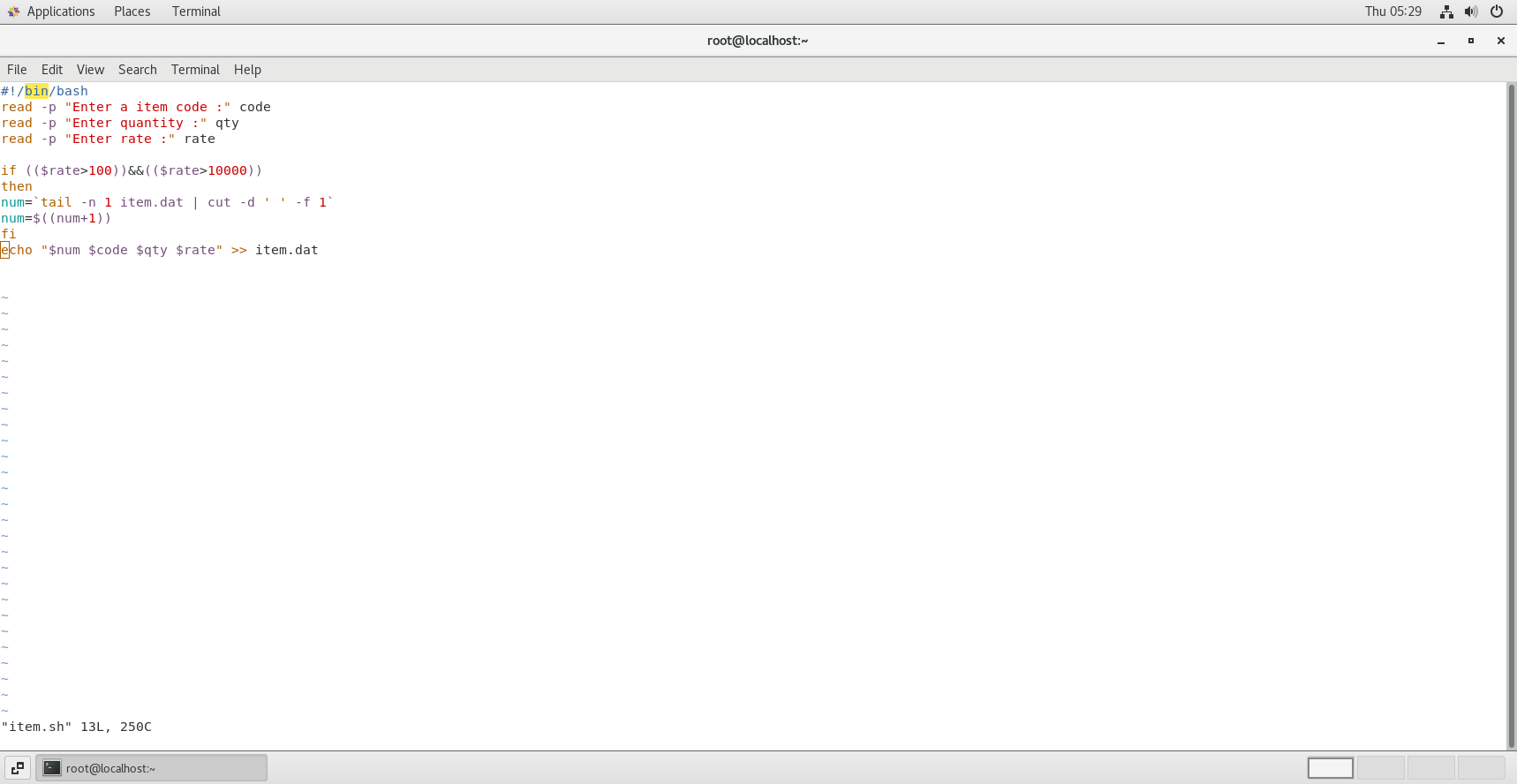
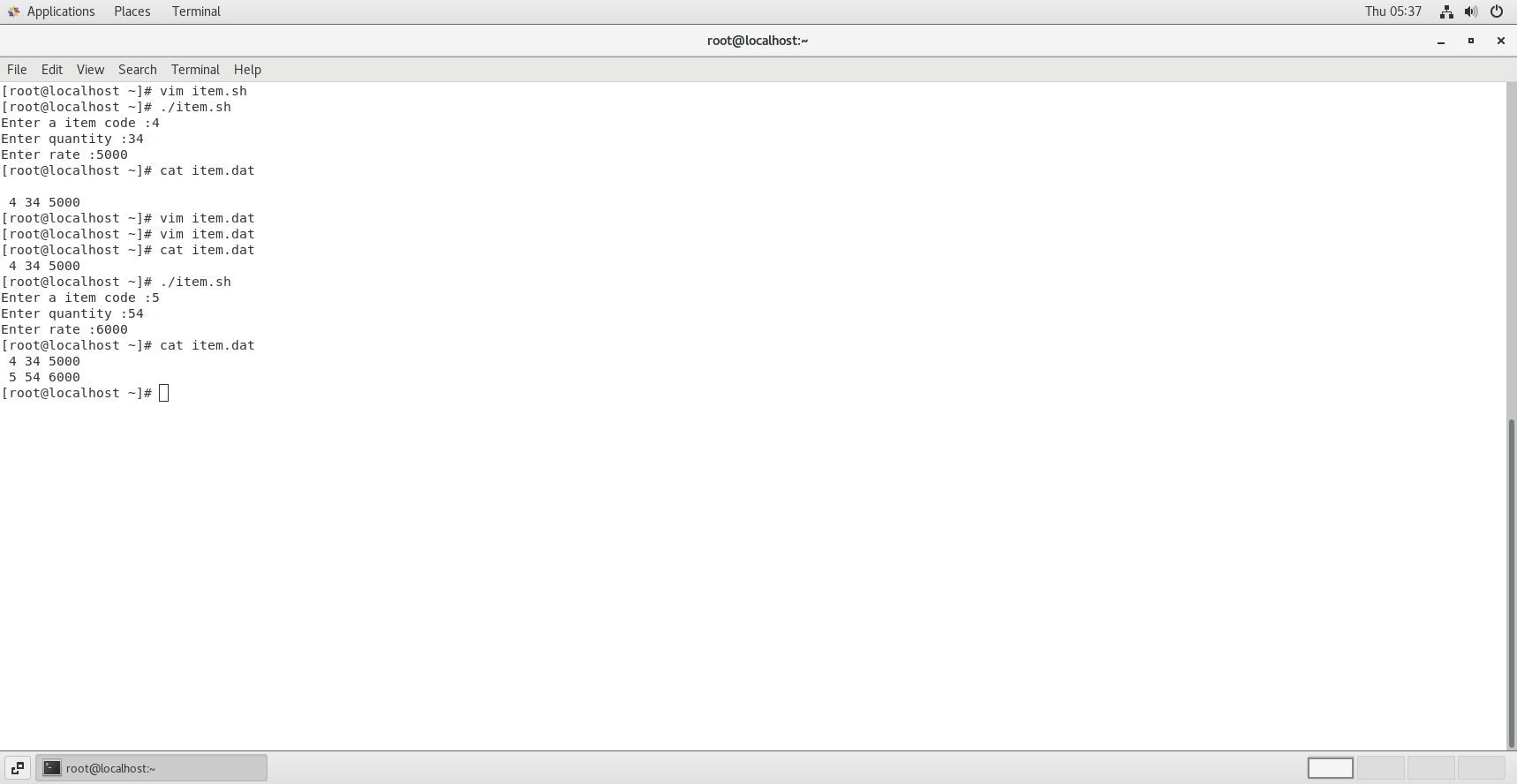


2.Write a unix shell to add records to a file called item.dat The fields being itemcode, qty, sold and rate

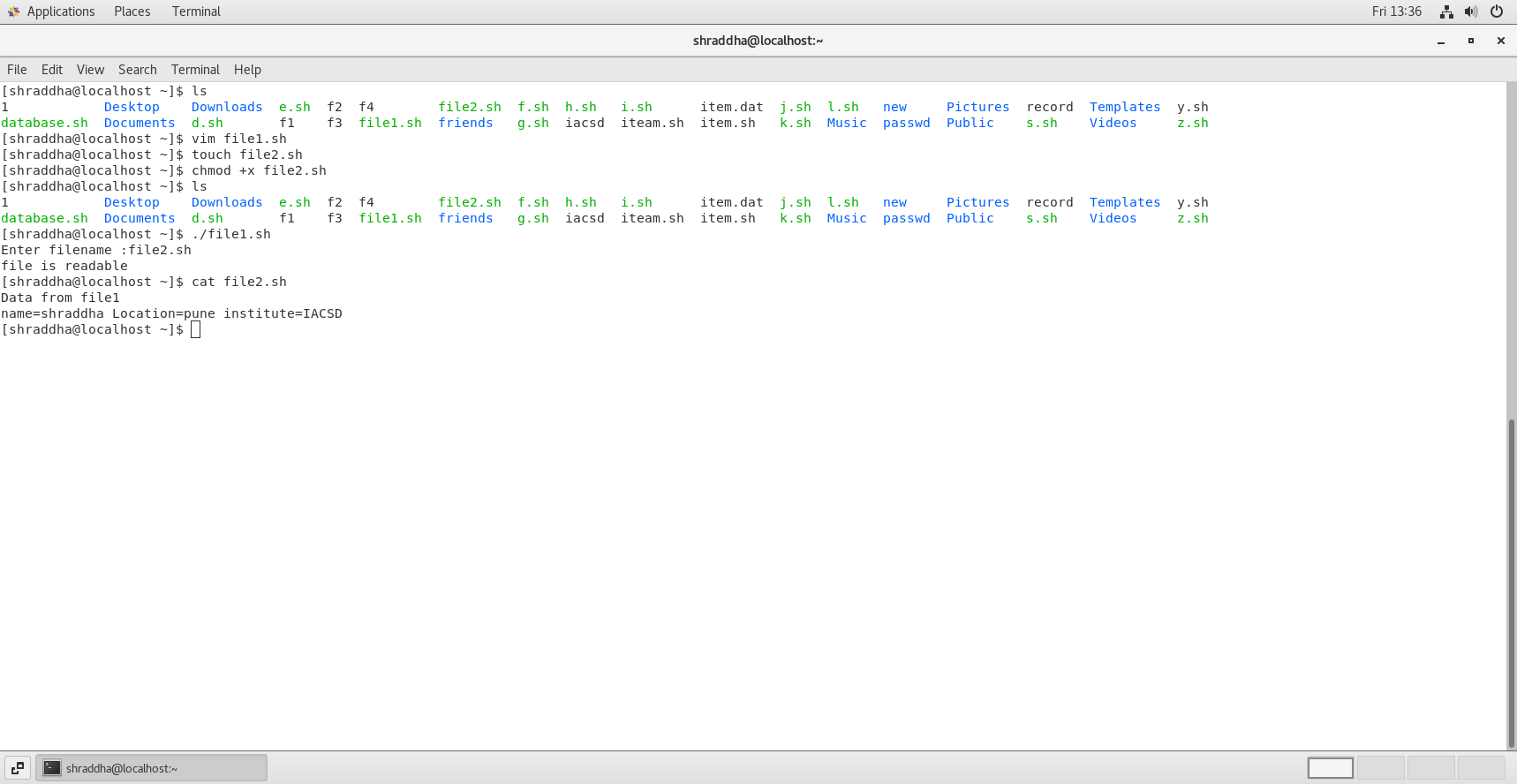
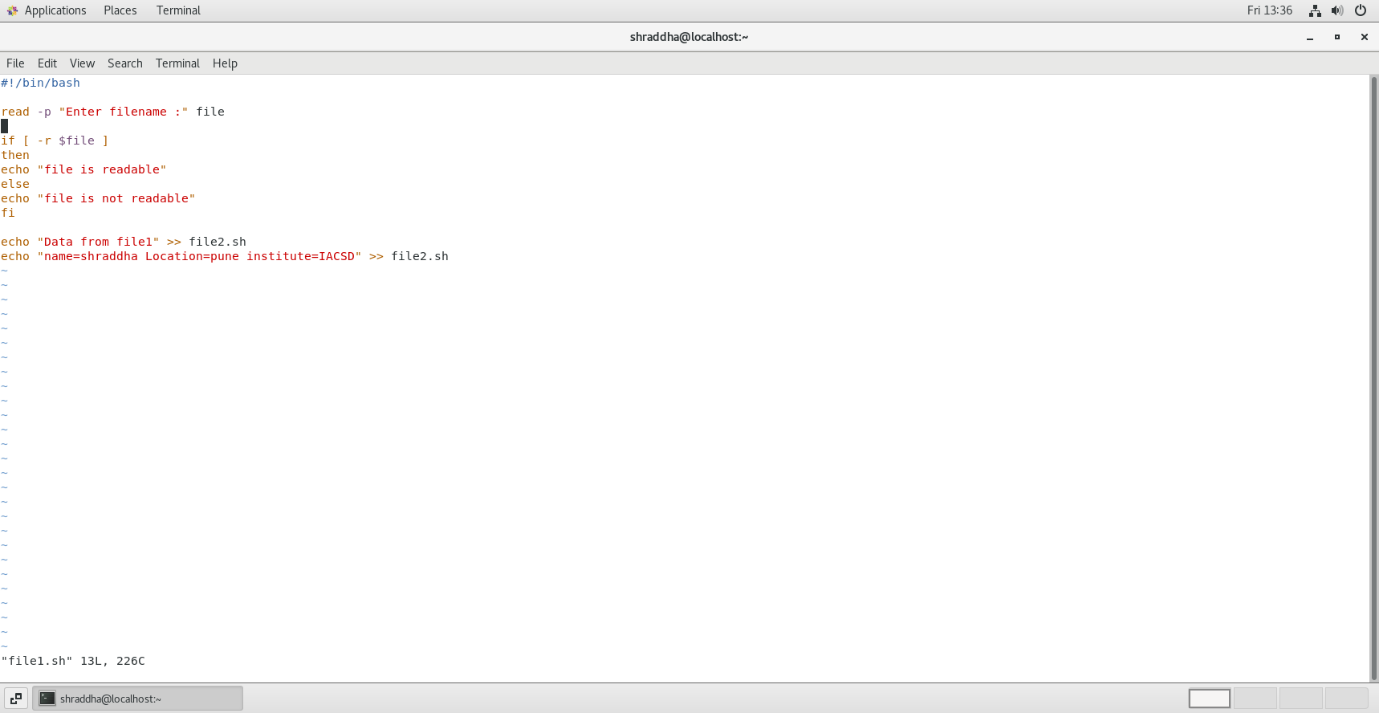
item\_code to be generated

qty\_sold should be greater than 0

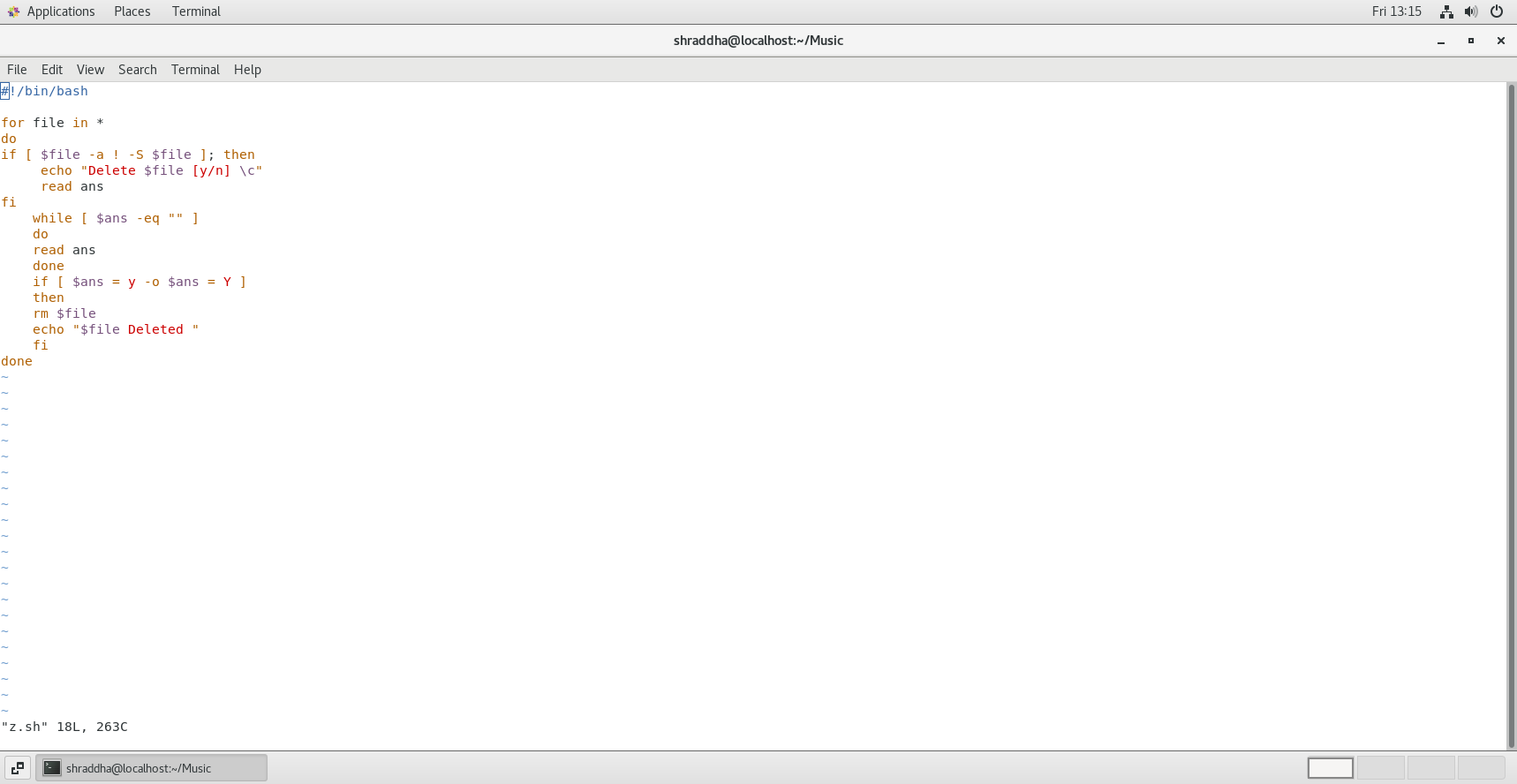
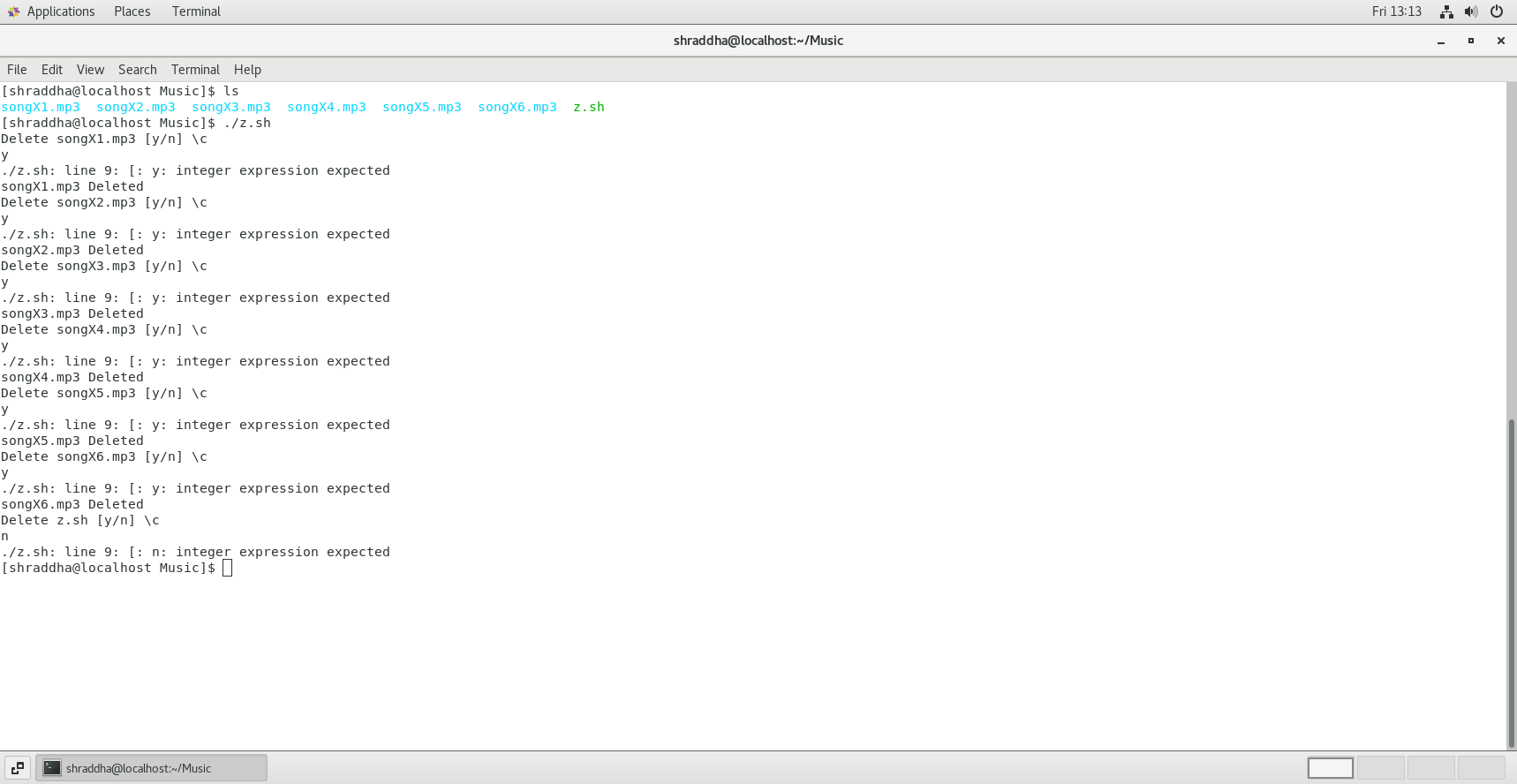
1. rate between 100 to 10000



3.Write a scripts 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 exits then it should ask the user whether he wants to overwrite it.



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



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

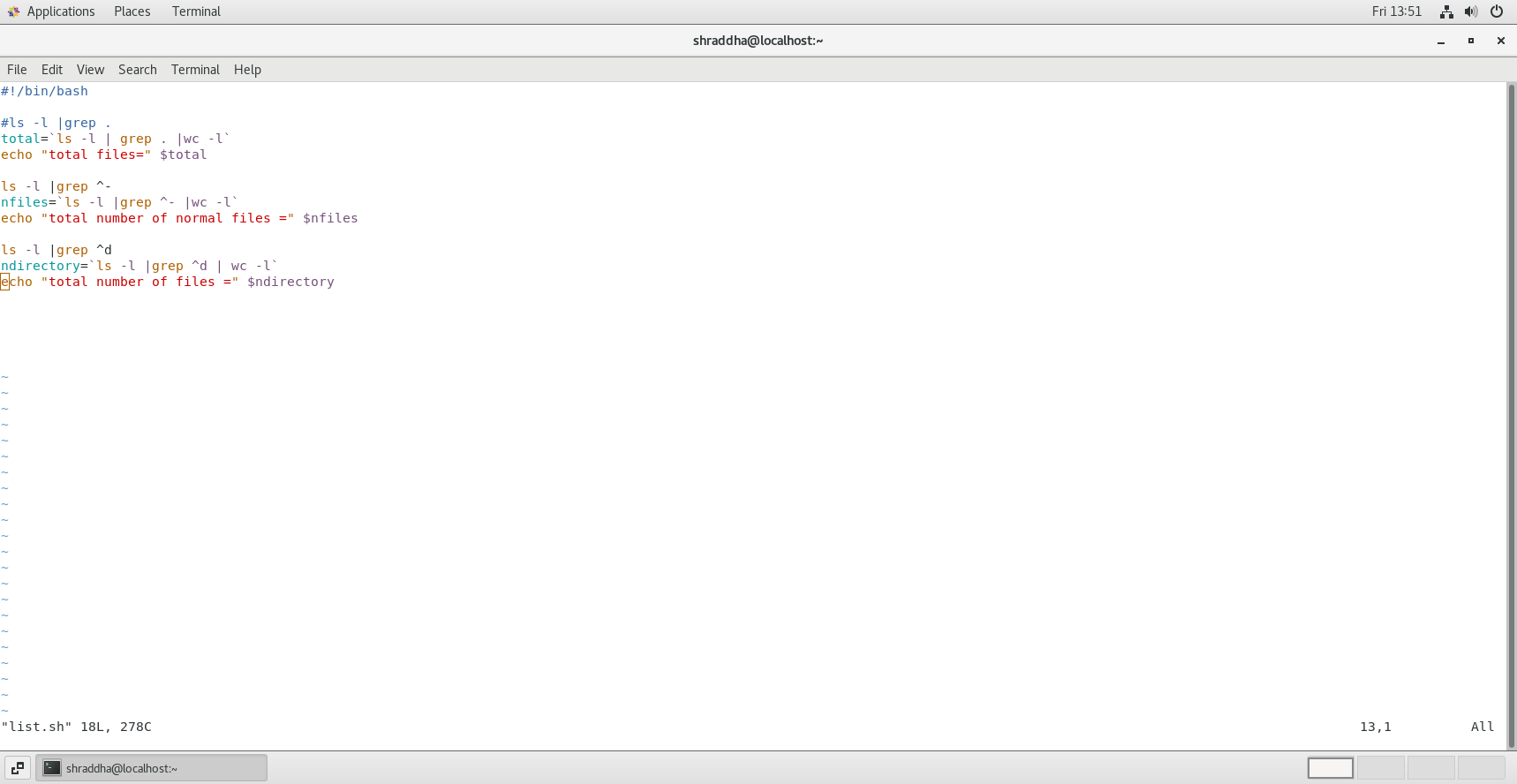
Filename3 date time permission

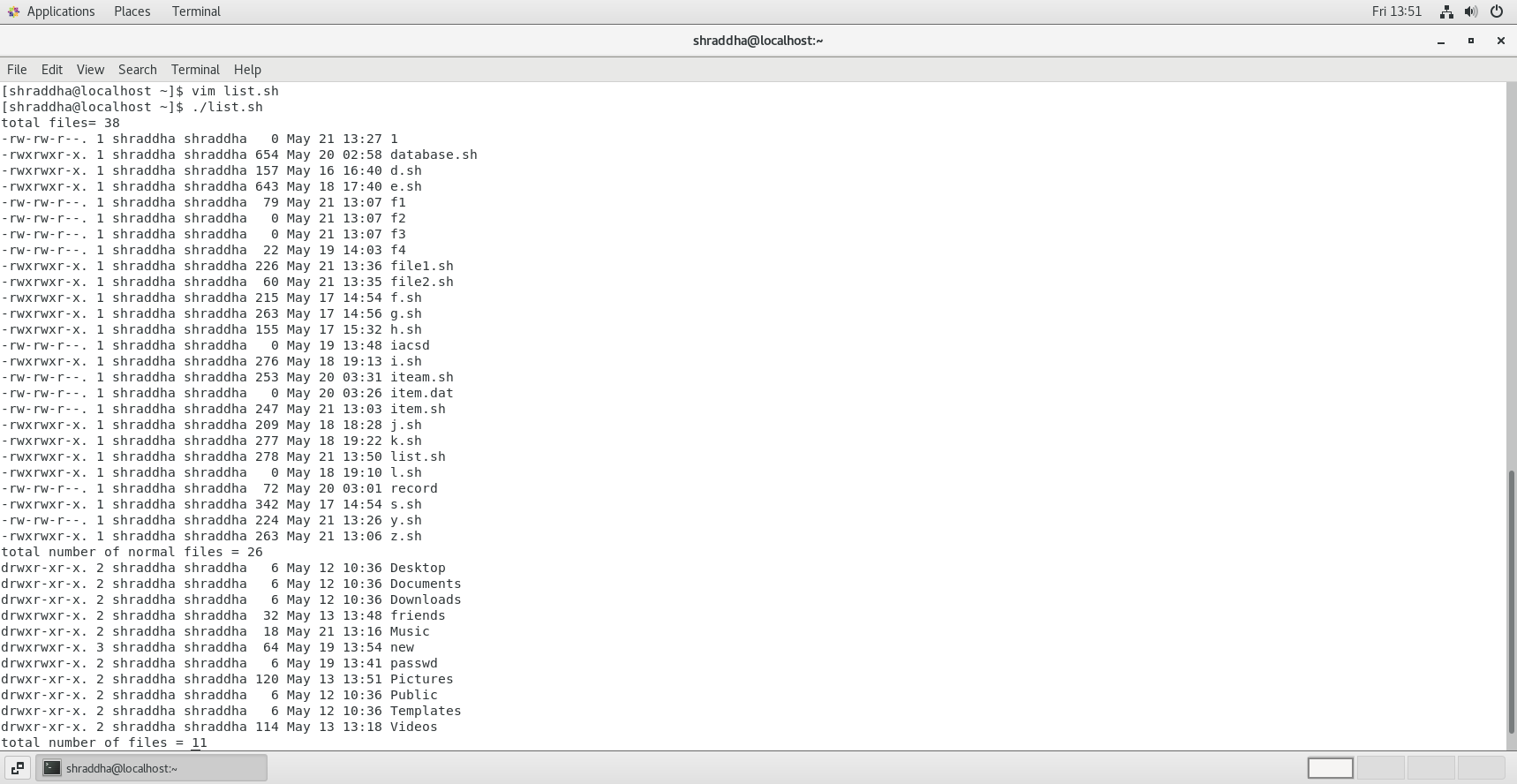
………..

………..

Total no. of files : <total number>

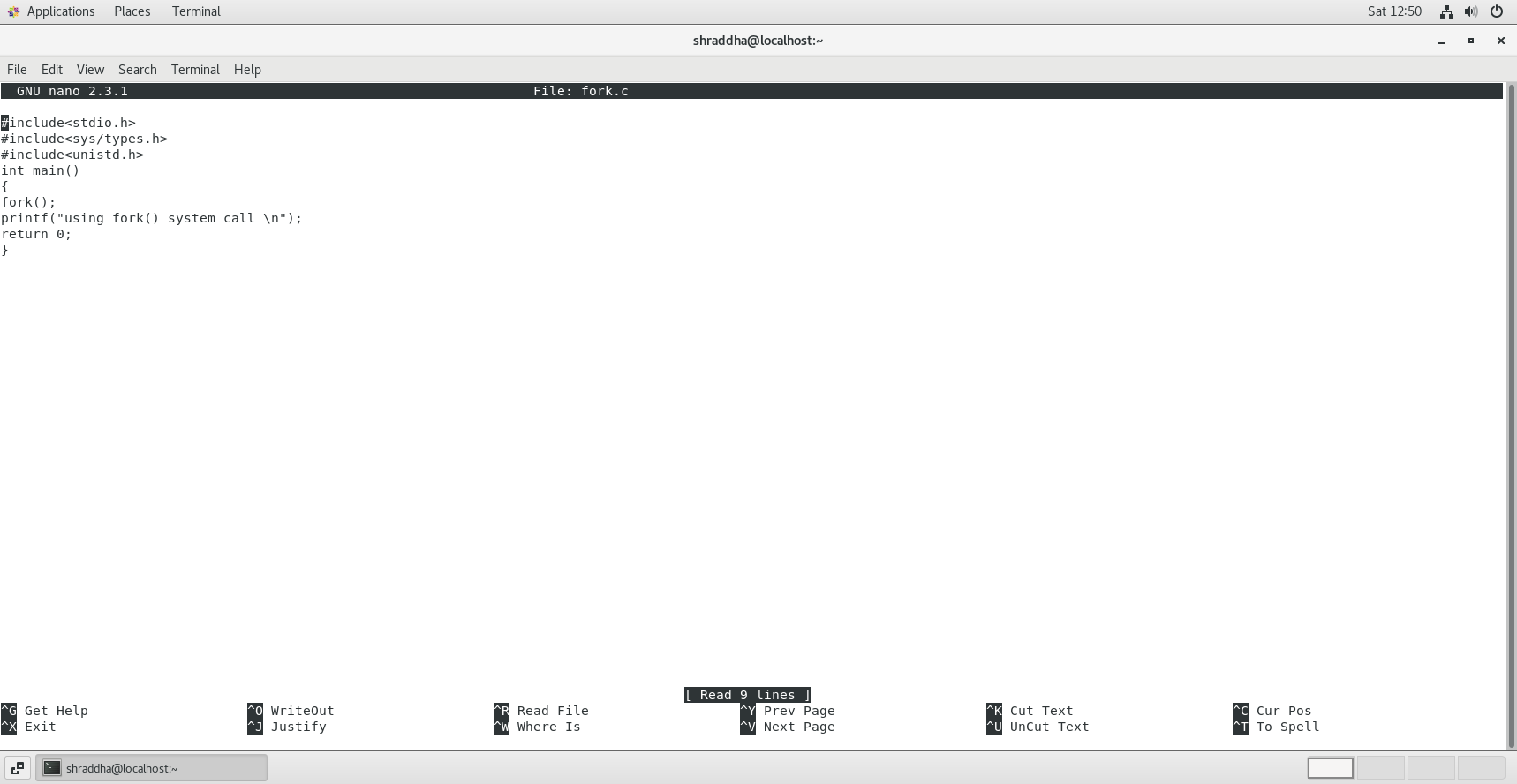
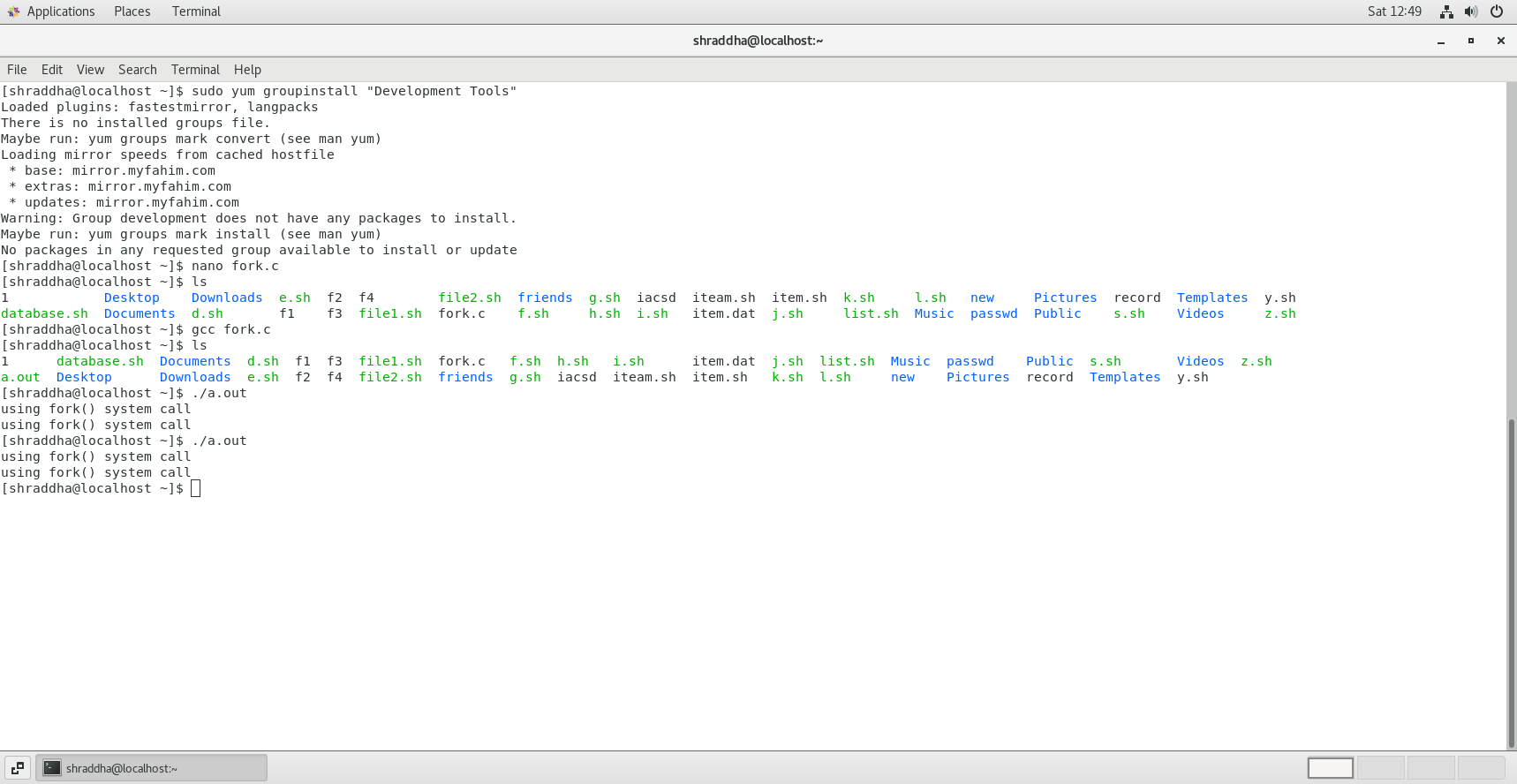
Total no of normal file : <number>

Total no of directory : <number>

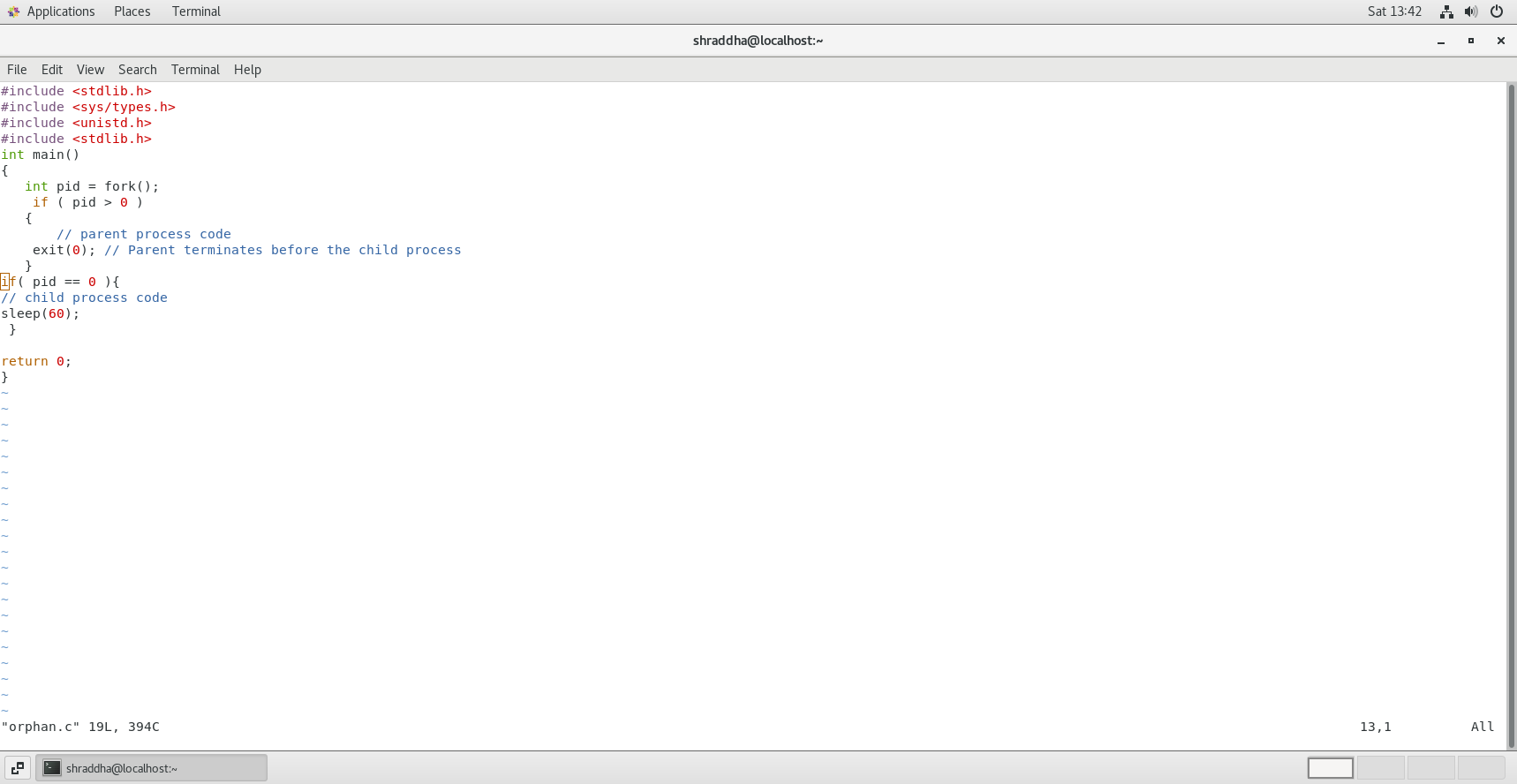
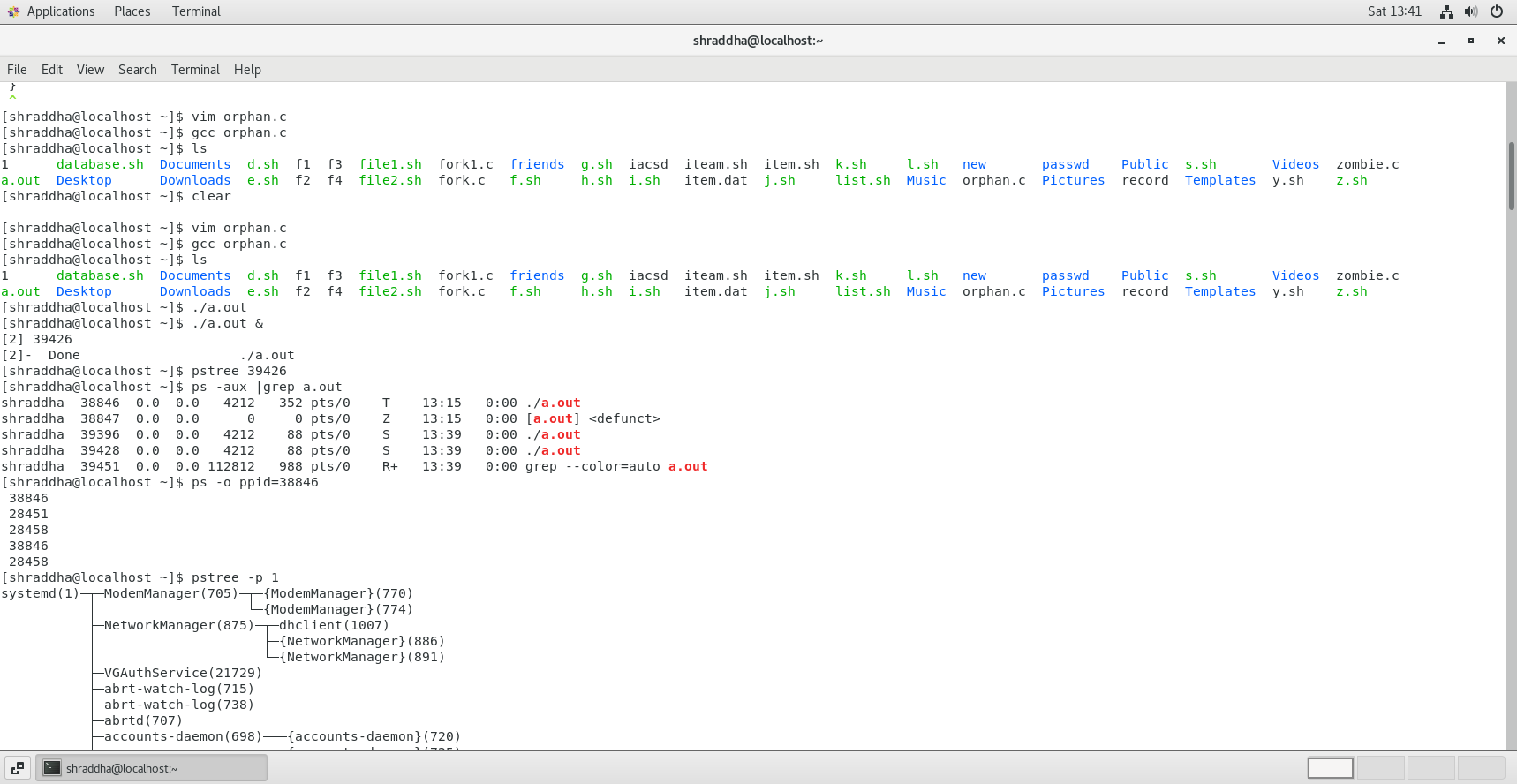


**Assignment 7**

1. Create Child process  using fork()



2. Create orphan process



3. Create Zombie process

