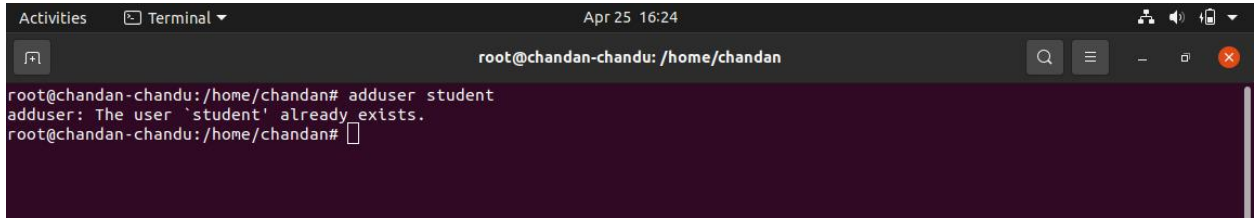


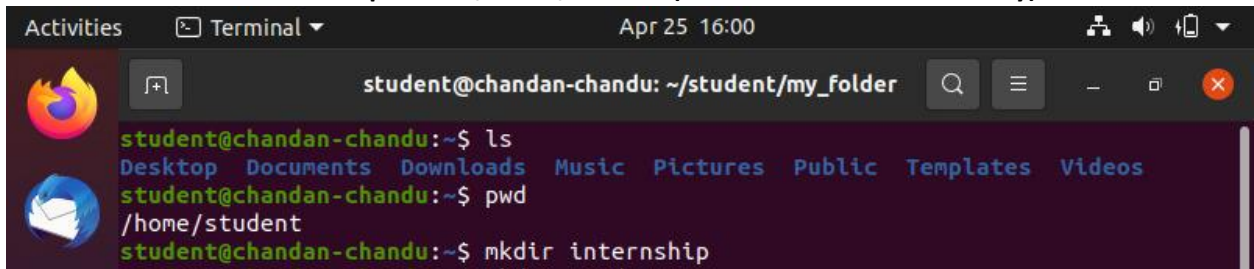
Task - 1:

1. Create a new user “student” in linux.



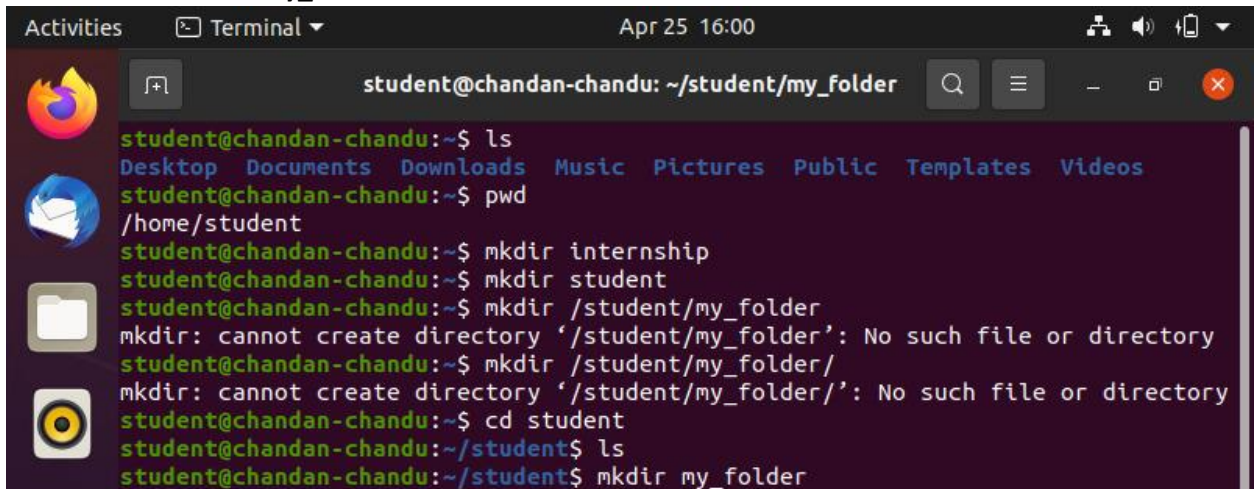
```
Activities Terminal Apr 25 16:24
root@chandan-chandu: /home/chandan
root@chandan-chandu:/home/chandan# adduser student
adduser: The user 'student' already exists.
root@chandan-chandu:/home/chandan#
```

2. Create a new folder ‘internship’ in the /home/student (student user home directory).



```
Activities Terminal Apr 25 16:00
student@chandan-chandu: ~/student/my_folder
student@chandan-chandu:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
student@chandan-chandu:~$ pwd
/home/student
student@chandan-chandu:~$ mkdir internship
```

3. Create a new folder ‘my_folder’ inside the student folder



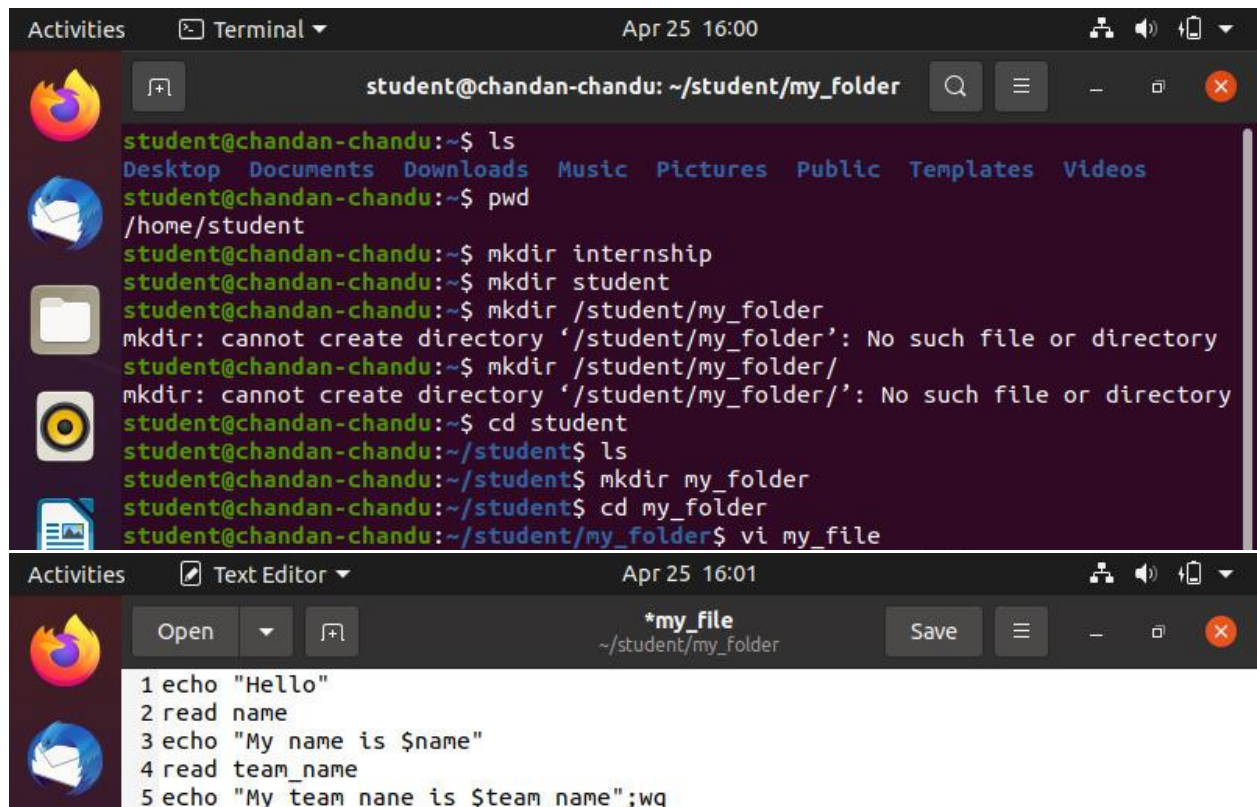
```
Activities Terminal Apr 25 16:00
student@chandan-chandu: ~/student/my_folder
student@chandan-chandu:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
student@chandan-chandu:~$ pwd
/home/student
student@chandan-chandu:~$ mkdir internship
student@chandan-chandu:~$ mkdir student
student@chandan-chandu:~$ mkdir /student/my_folder
mkdir: cannot create directory '/student/my_folder': No such file or directory
student@chandan-chandu:~$ mkdir /student/my_folder/
mkdir: cannot create directory '/student/my_folder/': No such file or directory
student@chandan-chandu:~$ cd student
student@chandan-chandu:~/student$ ls
student@chandan-chandu:~/student$ mkdir my_folder
```

4. Create a new file ‘my_file’ inside the folder student with the following contents:

Hello

My name is [your name].

My team is [your_team_name].



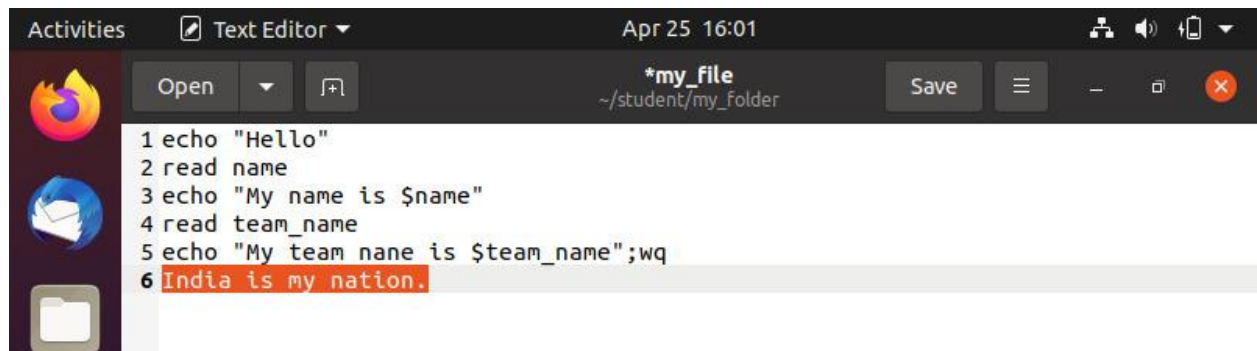
The image shows two windows from a Linux desktop environment. The top window is a terminal titled 'student@chandan-chandu: ~/student/my_folder'. It displays the following commands and output:

```
student@chandan-chandu:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
student@chandan-chandu:~$ pwd
/home/student
student@chandan-chandu:~$ mkdir internship
student@chandan-chandu:~$ mkdir student
student@chandan-chandu:~$ mkdir /student/my_folder
mkdir: cannot create directory '/student/my_folder': No such file or directory
student@chandan-chandu:~$ mkdir /student/my_folder/
mkdir: cannot create directory '/student/my_folder/': No such file or directory
student@chandan-chandu:~$ cd student
student@chandan-chandu:~/student$ ls
student@chandan-chandu:~/student$ mkdir my_folder
student@chandan-chandu:~/student$ cd my_folder
student@chandan-chandu:~/student/my_folder$ vi my_file
```

The bottom window is a text editor titled '*my_file' with the path '~/student/my_folder'. It shows the content of the file being edited:

```
1 echo "Hello"
2 read name
3 echo "My name is $name"
4 read team_name
5 echo "My team nane is $team_name";wq
```

5. Append the file my_file to add the following line at the end of the file: India is my nation.



The image shows a text editor window titled '*my_file' with the path '~/student/my_folder'. The file content is as follows:

```
1 echo "Hello"
2 read name
3 echo "My name is $name"
4 read team_name
5 echo "My team nane is $team_name";wq
6 India is my nation.
```

The line '6 India is my nation.' is highlighted in red, indicating it has been added to the end of the file.

6. Navigate to the student directory and create a new folder 'student'

```
Activities Terminal Apr 25 16:06
student@chandan-chandu: ~/student/studentdir_2
My name is chandan
Charvik_world
My team name is Charvik_world
./my_file: line 5: wq: command not found
student@chandan-chandu:~/student/my_folder$ gedit my_file
cd /
^Z
[1]+  Stopped                  gedit my_file
student@chandan-chandu:~/student/my_folder$ cd ..
student@chandan-chandu:~/student$ pwd
/home/student/student
student@chandan-chandu:~/student$ mkdir studentdir_2
```

7. Create three files with some text in the student folder.

```
student@chandan-chandu:~/student/studentdir_2$ touch myfile1.txt myfile2.txt my
file3.txt
Activities Terminal Apr 25 16:06
student@chandan-chandu: ~/student/studentdir_2
hello file3 how are you
```

8. Delete any one file from the three created files.

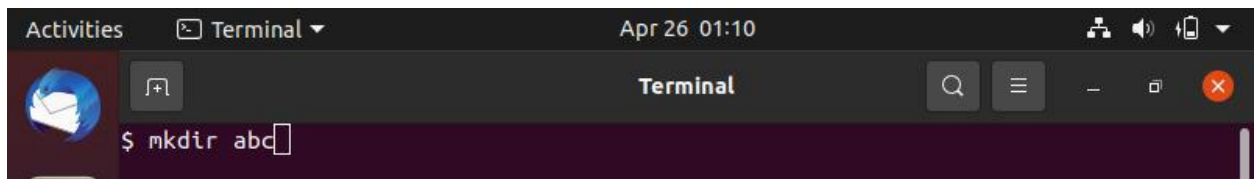
```
student@chandan-chandu:~/student/studentdir_2$
student@chandan-chandu:~/student/studentdir_2$ rm myfile3.txt
student@chandan-chandu:~/student/studentdir_2$ ls
myfile1.txt myfile2.txt
student@chandan-chandu:~/student/studentdir_2$
```

Task - 2:

1. Create two users 'student1' and 'student2'

```
root@chandan-chandu:/home# useradd -m student1
root@chandan-chandu:/home# passwd student1
New password:
Retype new password:
passwd: password updated successfully
root@chandan-chandu:/home# useradd -m student2
root@chandan-chandu:/home# passwd student2
New password:
Retype new password:
passwd: password updated successfully
root@chandan-chandu:/home#
```

2. Create a folder "abc" in the student user home directory (/home/student)



A terminal window titled "Terminal" with a search icon, a menu icon, and window control buttons. The command `$ mkdir abc` is entered at the prompt.

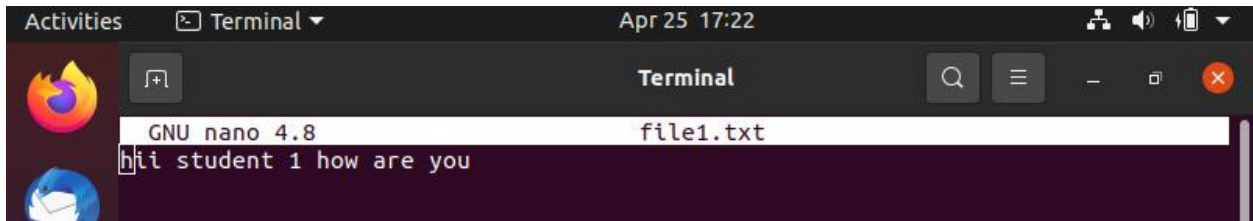
3. Create a folder "intern" inside the abc directory (/home/student)



A terminal window showing a sequence of commands: `$ cd abc`, `$ pwd` (output: `/home/student1/abc`), `$ mkdir intern`, `$ cd intern`, and `$ pwd` (output: `/home/student1/abc/intern`).

4. Create four files "file1", "file2", "file3" and "file4" with some random text inside the intern folder.


```
$ touch fil1.txt file2.txt file3.txt file4.txt
$ nano file1.txt
```



The screenshot shows a terminal window with the title 'Terminal'. The top bar indicates the date and time as 'Apr 25 17:22'. The terminal content shows the execution of 'touch' and 'nano' commands. The nano editor interface is visible, with 'GNU nano 4.8' at the top and 'file1.txt' as the filename. The text 'hii student 1 how are you' is entered on the first line.

5. Create a hidden file “my_hidden_file” inside the intern folder

```
$ pwd
/home/student1/abc/intern
$ vi .my_hideen.txt
$
```



The screenshot shows a terminal window with the following commands and output: 'pwd' returns '/home/student1/abc/intern', 'vi .my_hideen.txt' is executed, and the prompt '\$' is shown on the next line.

6. Create a new folder “test1” inside the abc folder.

```
$ pwd
/home/student1/abc
$ mkdir test1
$
```

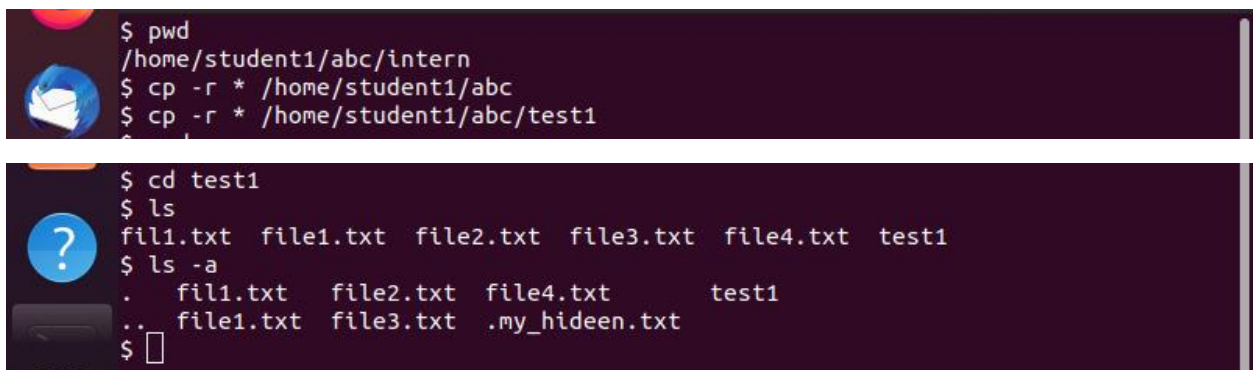


The screenshot shows a terminal window with the following commands and output: 'pwd' returns '/home/student1/abc', 'mkdir test1' is executed, and the prompt '\$' is shown on the next line.

7. Copy all the contents including the hidden file of the intern folder to the test1 folder

```
$ pwd
/home/student1/abc/intern
$ cp -r * /home/student1/abc
$ cp -r * /home/student1/abc/test1
$
```

```
$ cd test1
$ ls
fil1.txt  file1.txt  file2.txt  file3.txt  file4.txt  test1
$ ls -a
.  fil1.txt  file2.txt  file4.txt  test1
.. file1.txt  file3.txt  .my_hideen.txt
$
```



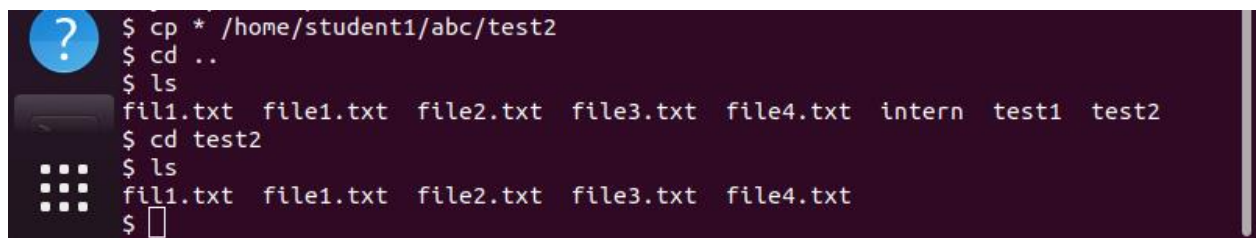
The screenshot shows two terminal windows. The top window shows the execution of 'cp -r * /home/student1/abc' and 'cp -r * /home/student1/abc/test1'. The bottom window shows the execution of 'cd test1', 'ls', and 'ls -a'. The output of 'ls' shows 'fil1.txt file1.txt file2.txt file3.txt file4.txt test1'. The output of 'ls -a' shows the hidden file '.my_hideen.txt' and other files.

8. Create a new folder “test2” inside the abc folder.



```
$ pwd
/home/student1/abc
$ mkdir test2
```

9. Copy all the contents excluding the hidden file of the intern folder to the test2 folder



```
? $ cp * /home/student1/abc/test2
$ cd ..
$ ls
file1.txt file1.txt file2.txt file3.txt file4.txt intern test1 test2
$ cd test2
$ ls
file1.txt file1.txt file2.txt file3.txt file4.txt
$
```

10. Create a file “test_file” in the abc folder with the following contents:

echo \$PWD

echo \$HOSTNAME

echo \$HOME

echo \$SHELL

echo \$PATH

echo \$USER

echo \$LANG

echo \$LOGNAME

```
Activities Terminal Apr 26 01:39
$ pwd
/home/student1
$ ls
abc Documents Music Public to_be_copied.tar.gz
Desktop Downloads Pictures Templates Videos
$ cd abc
$ pwd
/home/student1/abc
$ ls
intern test1 test2 test_file test_file.sh to_be_copied.tar.gz
$ vi test_file.sh
$ chmod +x test_file.sh
$ ./test_file.sh
/home/student1/abc

/home/student1
/bin/sh
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
student1
en_IN
student1
$
```

11. Compress the abc folder as “to_be_copied.tar.gz”.

```
$ ls
intern test1 test2 test_file test_file.sh
$ cd ..
$ ls
abc Desktop Documents Downloads Music Pictures Public Templates Videos
$ tar -zcvf abc.tar.gz /home/student1/abc
tar: Removing leading '/' from member names
/home/student1/abc/
/home/student1/abc/test1/
/home/student1/abc/test1/file4.txt
/home/student1/abc/test1/test1/
/home/student1/abc/test1/test1/file4.txt
/home/student1/abc/test1/test1/test1/
/home/student1/abc/test1/test1/test1/file4.txt
/home/student1/abc/test1/test1/test1/file1.txt
/home/student1/abc/test1/test1/test1/file3.txt
/home/student1/abc/test1/test1/test1/.my_hideen.txt
/home/student1/abc/test1/test1/test1/fil1.txt
/home/student1/abc/test1/test1/test1/file2.txt
/home/student1/abc/test1/test1/file1.txt
/home/student1/abc/test1/test1/file3.txt
/home/student1/abc/test1/test1/.my_hideen.txt
/home/student1/abc/test1/test1/fil1.txt
/home/student1/abc/test1/test1/file2.txt
```

12. Now, Login as student1 and copy the zip file to the student1 user home directory and run the test_file and save the output as “output.txt” after extracting the zip file

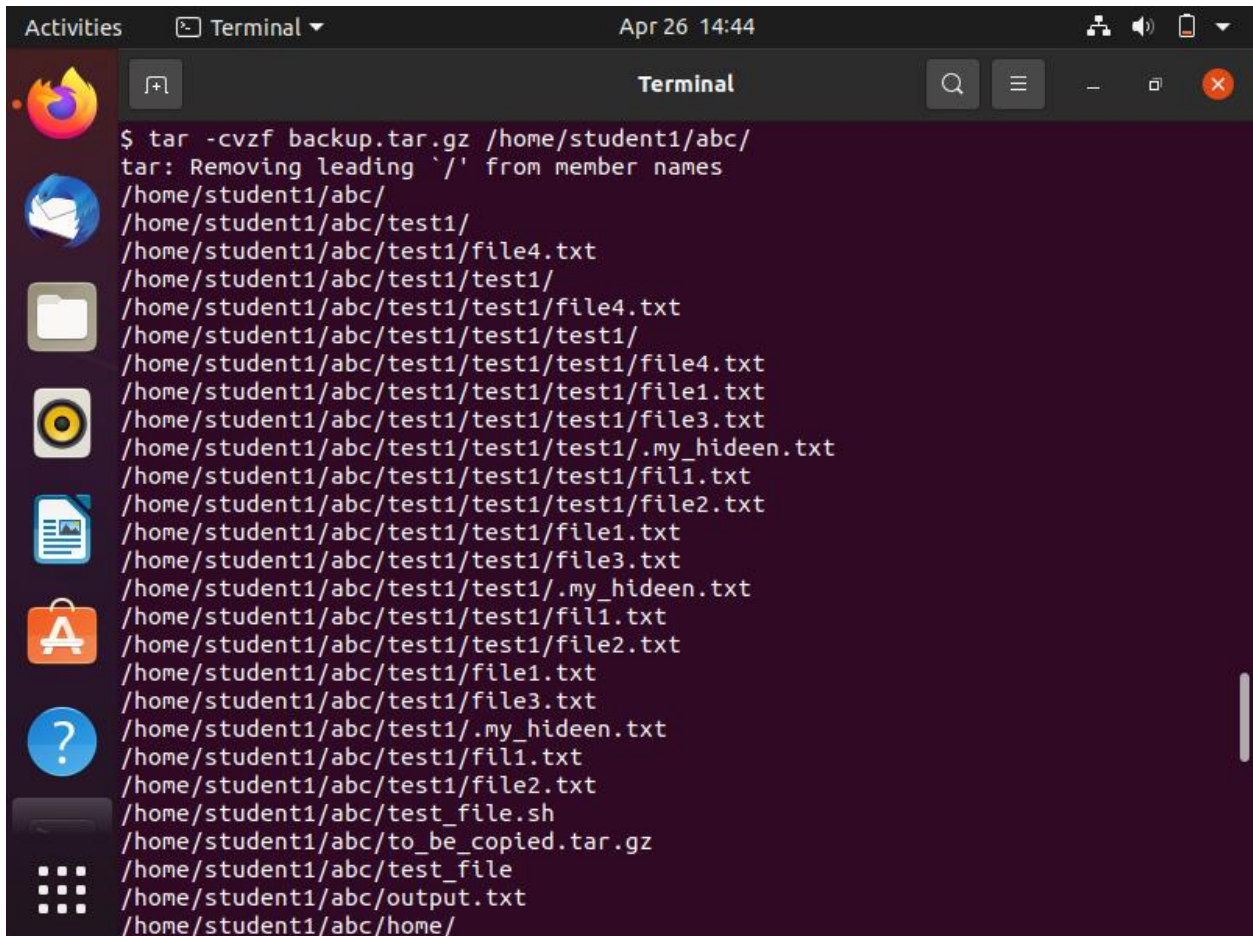
A terminal window with a dark purple background. On the left side, there are four icons: a document with a magnifying glass, an orange shopping bag, a blue circle with a white question mark, and a 3x3 grid of dots. The terminal text shows a user running a script, listing files, and displaying the script's output. The output lists files in the home directory of student1, including output.txt, test2, test_file.sh, test1, test_file, and to_be_copied.tar.gz. The user then runs 'cat output.txt' which shows the path /home/student1/abc. Finally, the user logs out of the terminal session.

```
$ ./test_file.sh > /home/student1/abc/output.txt
$ ls
home    output.txt  test2      test_file.sh
intern  test1      test_file  to_be_copied.tar.gz
$ cat output.txt
/home/student1/abc

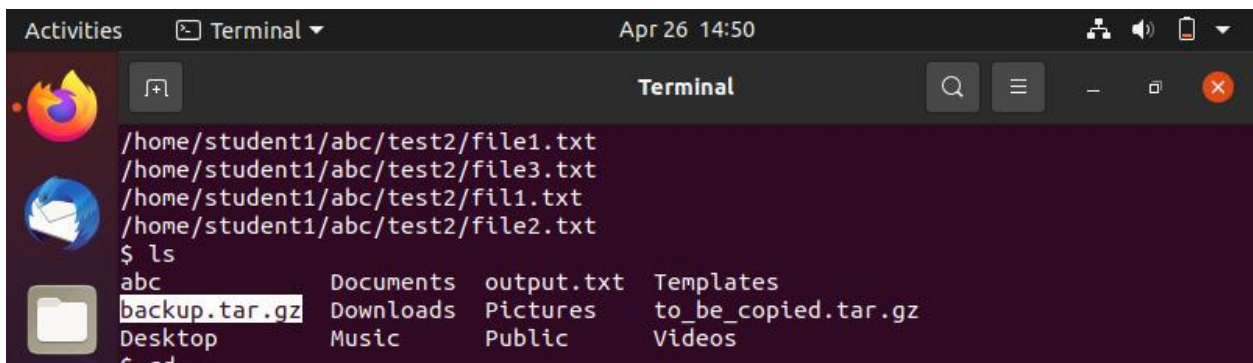
/home/student1
/bin/sh
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
student1
en_IN
student1
$
```

13. Again, login as student2 and copy the zip file to the student2 user home directory. Rename the zip file as “copy.tar.gz”. Make a copy of it as “copied.gz” and run the test_file and save the output as “output.txt” after extracting the copied.tar.gz.

14. Backup and compress the abc folder present in the student user home directory as “backup.tar.gz”.



```
Activities Terminal Apr 26 14:44
$ tar -cvzf backup.tar.gz /home/student1/abc/
tar: Removing leading '/' from member names
/home/student1/abc/
/home/student1/abc/test1/
/home/student1/abc/test1/file4.txt
/home/student1/abc/test1/test1/
/home/student1/abc/test1/test1/file4.txt
/home/student1/abc/test1/test1/test1/
/home/student1/abc/test1/test1/test1/file4.txt
/home/student1/abc/test1/test1/test1/file1.txt
/home/student1/abc/test1/test1/test1/file3.txt
/home/student1/abc/test1/test1/test1/.my_hideen.txt
/home/student1/abc/test1/test1/test1/fil1.txt
/home/student1/abc/test1/test1/test1/file2.txt
/home/student1/abc/test1/test1/file1.txt
/home/student1/abc/test1/test1/file3.txt
/home/student1/abc/test1/test1/.my_hideen.txt
/home/student1/abc/test1/test1/fil1.txt
/home/student1/abc/test1/test1/file2.txt
/home/student1/abc/test1/file1.txt
/home/student1/abc/test1/file3.txt
/home/student1/abc/test1/.my_hideen.txt
/home/student1/abc/test1/fil1.txt
/home/student1/abc/test1/file2.txt
/home/student1/abc/test_file.sh
/home/student1/abc/to_be_copied.tar.gz
/home/student1/abc/test_file
/home/student1/abc/output.txt
/home/student1/abc/home/
```



```
Activities Terminal Apr 26 14:50
/home/student1/abc/test2/file1.txt
/home/student1/abc/test2/file3.txt
/home/student1/abc/test2/fil1.txt
/home/student1/abc/test2/file2.txt
$ ls
abc          Documents  output.txt  Templates
backup.tar.gz Downloads  Pictures    to_be_copied.tar.gz
Desktop      Music      Public      Videos
$ cd
```

Task - 3:

The dataset for this assignment can be accessed from the following link

Link -: <https://drive.google.com/file/d/1OA2EYkIvry6NHLmRkQs-LMHch9dxZ6gs/view>

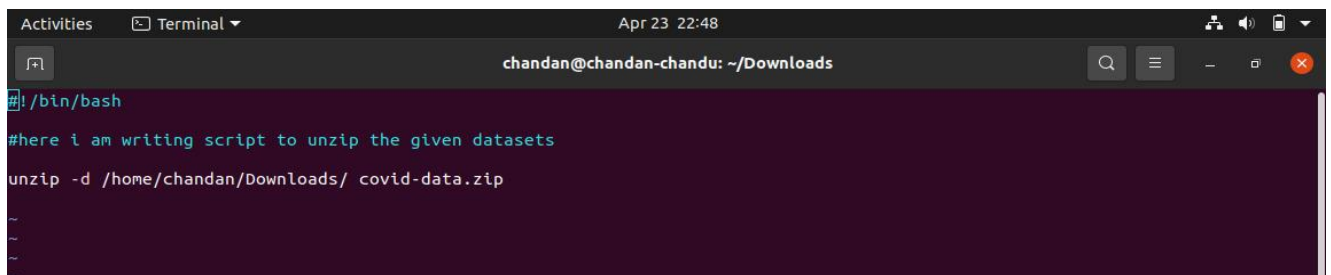
Overview:

Twitter is making it possible for developers and researchers to study the public conversation around COVID-19 in real time. This dataset includes a CSV file which contains tweets extracted from the Twitter website in March 2020. The dataset is large and thus you are initially required to manipulate it using shell scripting.

Download the file covid-data.zip from the link provided above. Use the Unix shell to manipulate the file and answer the following questions.

Questions:

1. Decompress the file. How big is it?

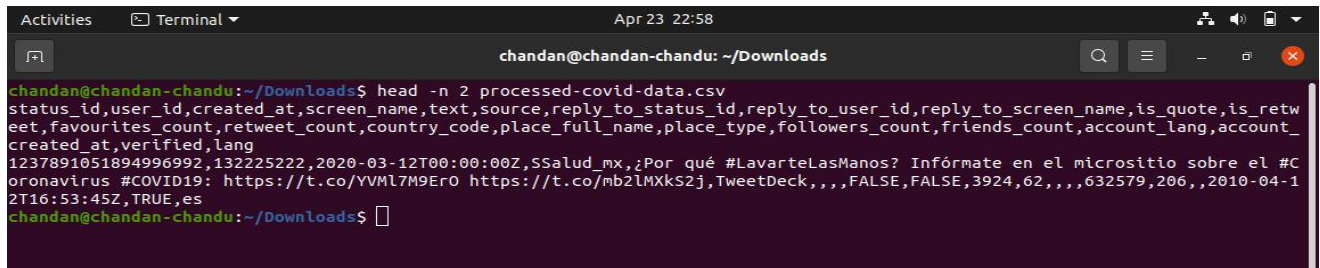
A terminal window titled 'chandan@chandan-chandu: ~/Downloads' with a timestamp of 'Apr 23 22:48'. The prompt is '#!/bin/bash'. The user has entered a comment '#here i am writing script to unzip the given datasets' and the command 'unzip -d /home/chandan/Downloads/ covid-data.zip'. The terminal shows three tilde characters '~' as output, indicating the command is still running or has just finished. The terminal window has standard Ubuntu window controls and a search bar.

Here, in above picture I have created unzip_covid.sh script to unzip the given zip file.

A terminal window titled 'chandan@chandan-chandu: ~/Downloads' with a timestamp of 'Apr 23 22:51'. The user has entered the command 'du -h processed-covid-data.csv'. The output is '2.9G processed-covid-data.csv'. The terminal window has standard Ubuntu window controls and a search bar.

Here in second snapshot there is 2.9 G is the size of the decompressed file.(peocessed-covid-data.csv)

2. What delimiter is used to separate the columns in the file? Write the code to show how many columns are there?



```
chandan@chandan-chandu: ~/Downloads
chandan@chandan-chandu:~/Downloads$ head -n 2 processed-covid-data.csv
status_id,user_id,created_at,screen_name,text,source,reply_to_status_id,reply_to_user_id,reply_to_screen_name,is_quote,is_retweet,favourites_count,retweet_count,country_code,place_full_name,place_type,followers_count,friends_count,account_lang,account_created_at,verified,lang
1237891051894996992,132225222,2020-03-12T00:00:00Z,SSalud_mx,¿Por qué #LavarteLasManos? Infórmate en el micrositio sobre el #Coronavirus #COVID19: https://t.co/YVML7M9Er0 https://t.co/mb2LMXks2j,TweetDeck,,,FALSE,FALSE,3924,62,,,632579,206,,2010-04-12T16:53:45Z,TRUE,es
chandan@chandan-chandu:~/Downloads$
```

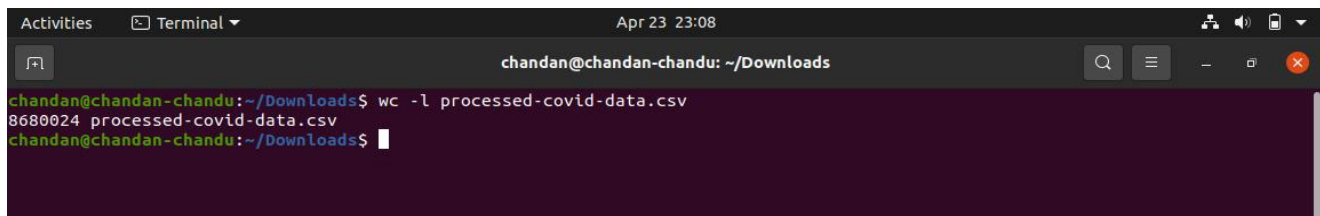
Here in the above picture “,” delimiter is used to separate the column in the file.



```
chandan@chandan-chandu:~/Downloads$ head -1 processed-covid-data.csv | sed 's/[^,]//g' | wc -c
22
chandan@chandan-chandu:~/Downloads$
```

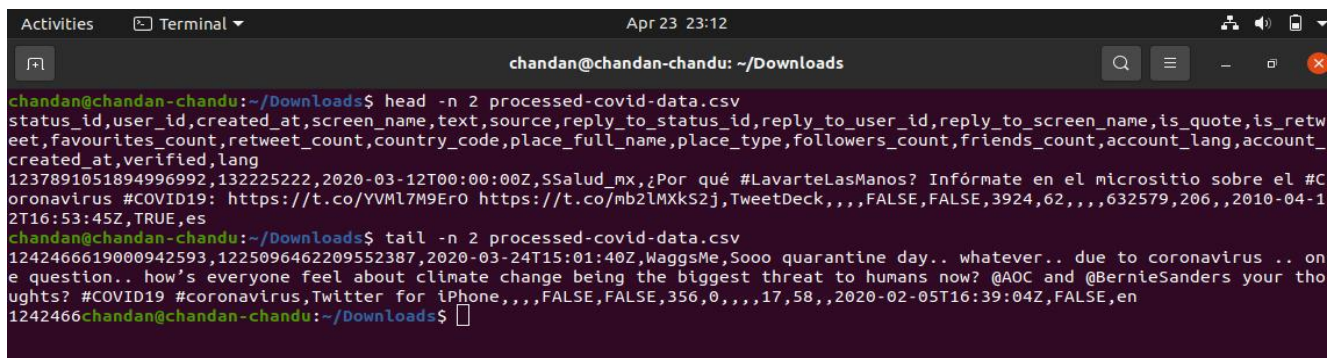
Here in the above picture the command is given to find the number of column according to given delimiter.

3. How many tweets are there in total in the file?



```
chandan@chandan-chandu:~/Downloads$ wc -l processed-covid-data.csv
8680024 processed-covid-data.csv
chandan@chandan-chandu:~/Downloads$
```

4. Assuming that the data is sorted, what is the date range of the tweets? (date of first and last tweet)



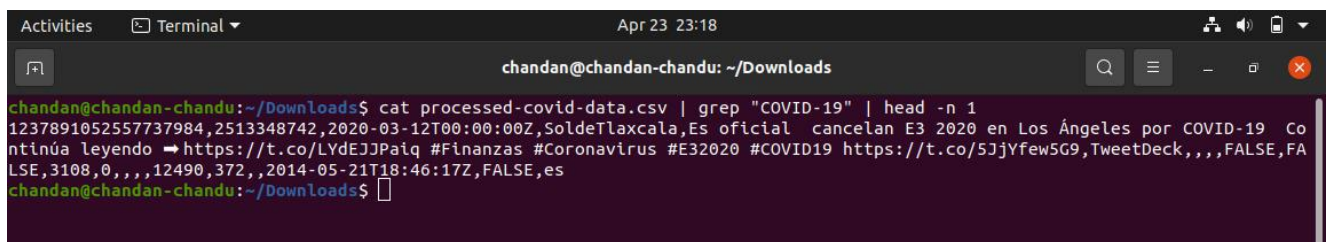
```
chandan@chandan-chandu:~/Downloads$ head -n 2 processed-covid-data.csv
status_id,user_id,created_at,screen_name,text,source,reply_to_status_id,reply_to_user_id,reply_to_screen_name,is_quote,is_retweet,favourites_count,retweet_count,country_code,place_full_name,place_type,followers_count,friends_count,account_lang,account_created_at,verified,lang
1237891051894996992,132225222,2020-03-12T00:00:00Z,SSalud_mx,¿Por qué #LavarteLasManos? Infórmate en el micrositio sobre el #Coronavirus #COVID19: https://t.co/YVML7M9Er0 https://t.co/mb2LMXks2j,TweetDeck,,,FALSE,FALSE,3924,62,,,632579,206,,2010-04-12T16:53:45Z,TRUE,es
chandan@chandan-chandu:~/Downloads$ tail -n 2 processed-covid-data.csv
1242466619000942593,1225096462209552387,2020-03-24T15:01:40Z,WaggsMe,Sooo quarantine day.. whatever.. due to coronavirus .. on the question.. how's everyone feel about climate change being the biggest threat to humans now? @AOC and @BernieSanders your thoughts? #COVID19 #coronavirus,Twitter for iPhone,,,FALSE,FALSE,356,0,,,17,58,,2020-02-05T16:39:04Z,FALSE,en
1242466chandan@chandan-chandu:~/Downloads$
```

Here two things comes out:

First tweet: 12-03-2020

Last tweet: 24-03-2020

5. When was the first mention of the term “COVID-19” in your dataset (notice that we look for COVID-19 with capital letters here)? What is the user_id, text and post date of this tweet?



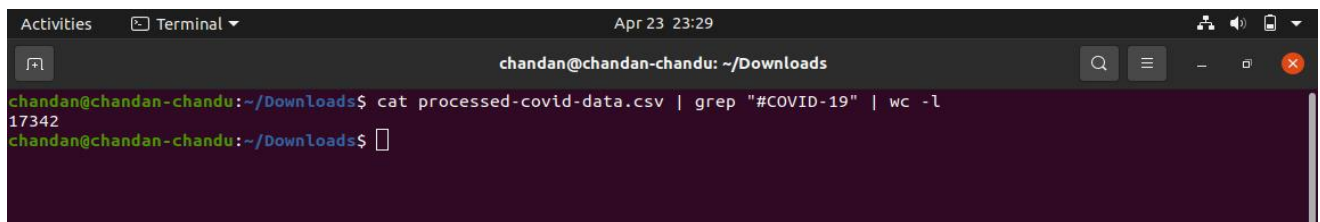
```
chandan@chandan-chandu: ~/Downloads
chandan@chandan-chandu:~/Downloads$ cat processed-covid-data.csv | grep "COVID-19" | head -n 1
1237891052557737984,2513348742,2020-03-12T00:00:00Z,SoldeTlaxcala,Es oficial cancelan E3 2020 en Los Angeles por COVID-19 Co
ntinúa leyendo ➡ https://t.co/LYdEJJPaIq #Finanzas #Coronavirus #E32020 #COVID19 https://t.co/5JjYfewSG9,TweetDeck,,,,FALSE,FA
LSE,3108,0,,,,12490,372,,2014-05-21T18:46:17Z,FALSE,es
chandan@chandan-chandu:~/Downloads$
```

In above picture we find here two things:

1st User Id:- 2513348742

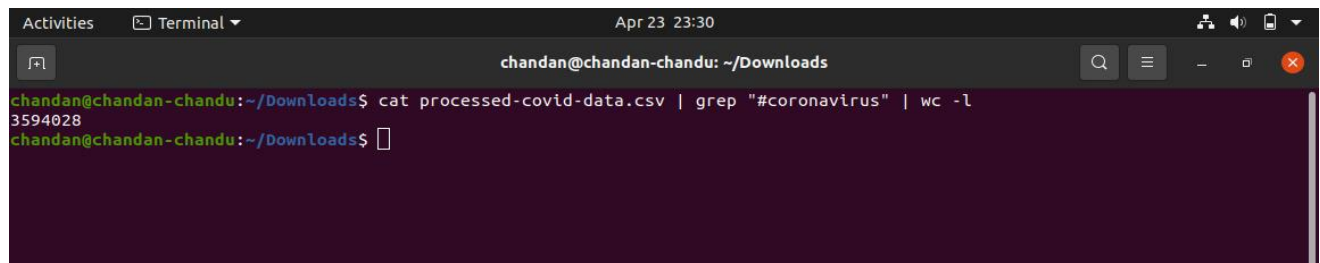
2nd Post Date:- 12-03-2020

**6. How many times did the hashtag #coronavirus or #COVID-19 appears in the file in the given form?(
If any of these words appear more than once in a line, you need to count all its occurrences to
answer this question properly)**



```
chandan@chandan-chandu: ~/Downloads
chandan@chandan-chandu:~/Downloads$ cat processed-covid-data.csv | grep "#COVID-19" | wc -l
17342
chandan@chandan-chandu:~/Downloads$
```

In the above picture #COVID-19 appears 17342 times

A terminal window titled 'Terminal' with a dark background. The prompt is 'chandan@chandan-chandu: ~/Downloads'. The command entered is 'cat processed-covid-data.csv | grep "#coronavirus" | wc -l'. The output is '3594028'.

```
chandan@chandan-chandu: ~/Downloads
chandan@chandan-chandu:~/Downloads$ cat processed-covid-data.csv | grep "#coronavirus" | wc -l
3594028
chandan@chandan-chandu:~/Downloads$
```

In the above picture #coronavirus appears 3594028 times

Task - 4

A company has set up a new site and transfer staff and visitor accounts to the new site.

Your task is to write a Bash script to create user accounts for all staff and visitors. The supplied user file `Unames.txt` is a text file containing a username and its type delimited by comma per line. There are two types of users: staff and visitors.

Example of `Unames.txt`

john,staff

alice,visitor

jonathan,visitor

bob,visitor

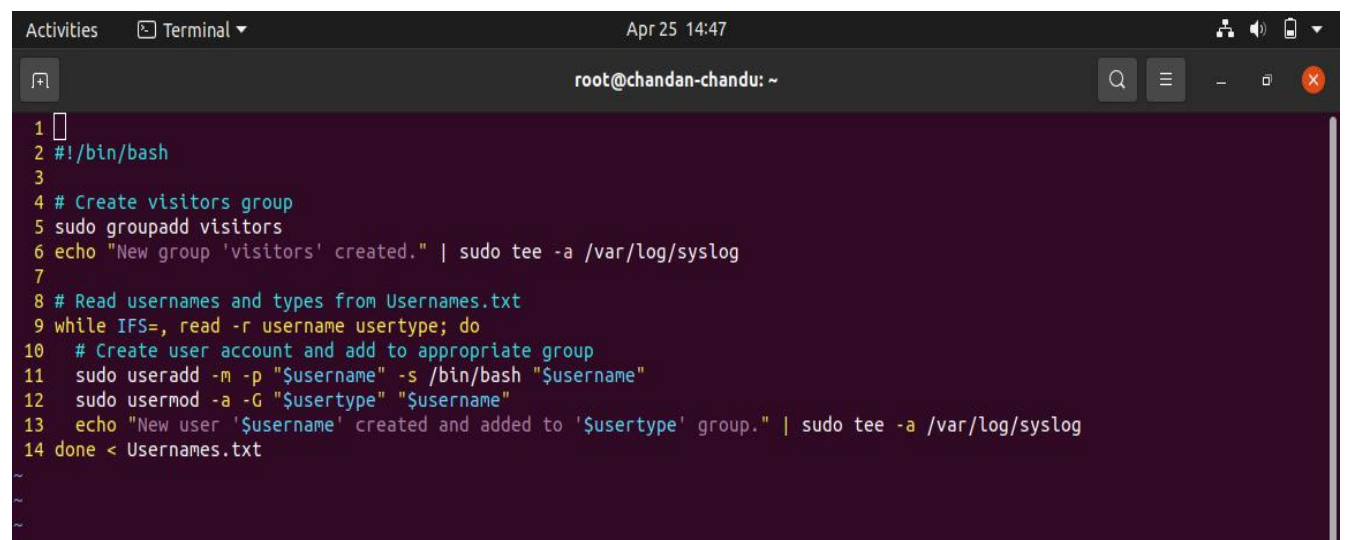
joe,staff

Staff users are added to the staff group and visitor users to the visitors group.

1. Write a Bash script, called createUsers.sh, to do the followings.

- a) Create a group called visitors;
- b) Create an account for each user and add the user to its group in one command. All user accounts are created with an initial password the same as their username; a home directory with the same name as their username in the /home directory; all accounts use Bash shell program.
- c) Write messages to syslog for all of the above events (new group, new user creation).

```
root@chandan-chandu:~# vi createuser.sh
root@chandan-chandu:~# vi createuser.sh
root@chandan-chandu:~# chmod 777 createuser.sh
root@chandan-chandu:~#
```



```
Activities Terminal Apr 25 14:47
root@chandan-chandu: ~
1
2 #!/bin/bash
3
4 # Create visitors group
5 sudo groupadd visitors
6 echo "New group 'visitors' created." | sudo tee -a /var/log/syslog
7
8 # Read usernames and types from Usernames.txt
9 while IFS=, read -r username usertype; do
10 # Create user account and add to appropriate group
11 sudo useradd -m -p "$username" -s /bin/bash "$username"
12 sudo usermod -a -G "$usertype" "$username"
13 echo "New user '$username' created and added to '$usertype' group." | sudo tee -a /var/log/syslog
14 done < Usernames.txt
~
~
~
```

```
root@chandan-chandu:~# cat Usernames.txt
chandan,staff
kundan,visitors
shabista,staff
imam,staff
aman,visitors
root@chandan-chandu:~#
```

```
Activities Terminal Apr 24 16:28
root@chandan-chandu: ~
root@chandan-chandu:~# ./createuser.sh
New group 'visitors' created.
useradd: user 'chandan' already exists
New user 'chandan' created and added to 'staff' group.
useradd: user 'kundan' already exists
New user 'kundan' created and added to 'visitors' group.
useradd: user 'shabista' already exists
New user 'shabista' created and added to 'staff' group.
useradd: user 'iman' already exists
New user 'iman' created and added to 'staff' group.
useradd: user 'aman' already exists
New user 'aman' created and added to 'visitors' group.
root@chandan-chandu:~#
```

2. Write a Bash script, called reportVisitors.sh, to report the members of visitors group to the file /tmp/visitors.txt

```
Activities Terminal Apr 24 16:51
root@chandan-chandu: ~
1 #!/bin/bash
2
3 # Report members of visitors group to /tmp/visitors.txt
4 members=$(getent group visitors | awk -F: '{print $1}')
5 echo "$members" > /tmp/visitors.txt
6 echo "Report generated at $(date)" >> /tmp/visitors.txt
~
~
~
~
```

3. Create a crontab entry to call the reportVisitors.sh script at 8:00AM and 9:00PM on every weekday

```

root@chandan-chandu:~# vi reportVisitors.sh
root@chandan-chandu:~# crontab -e
no crontab for root - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano          <---- easiest
 2. /usr/bin/vim.basic
 3. /usr/bin/vim.tiny
 4. /usr/bin/emacs
 5. /bin/ed

Choose 1-5 [1]: 1
crontab: installing new crontab
root@chandan-chandu:~# crontab -e
No modification made
root@chandan-chandu:~# ls
cpumem  cpumem.sh  createuser.sh  dir  internsctl  reportVisitors.sh  root  snap  Usernames.txt
root@chandan-chandu:~# crontab -e
crontab: installing new crontab
root@chandan-chandu:~# ./reportVisitors.sh
root@chandan-chandu:~# crontab -e
No modification made

```

```

GNU nano 4.8 /tmp/crontab.acV7Vi/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
0 8,21 * * 1-5 /reportVisitors.sh

```

Read 24 lines

Get Help Exit Write Out Read File Where Is Replace Cut Text Paste Text Justify To Spell Cur Pos Go To Line Undo Redo Mark Text Copy Text

