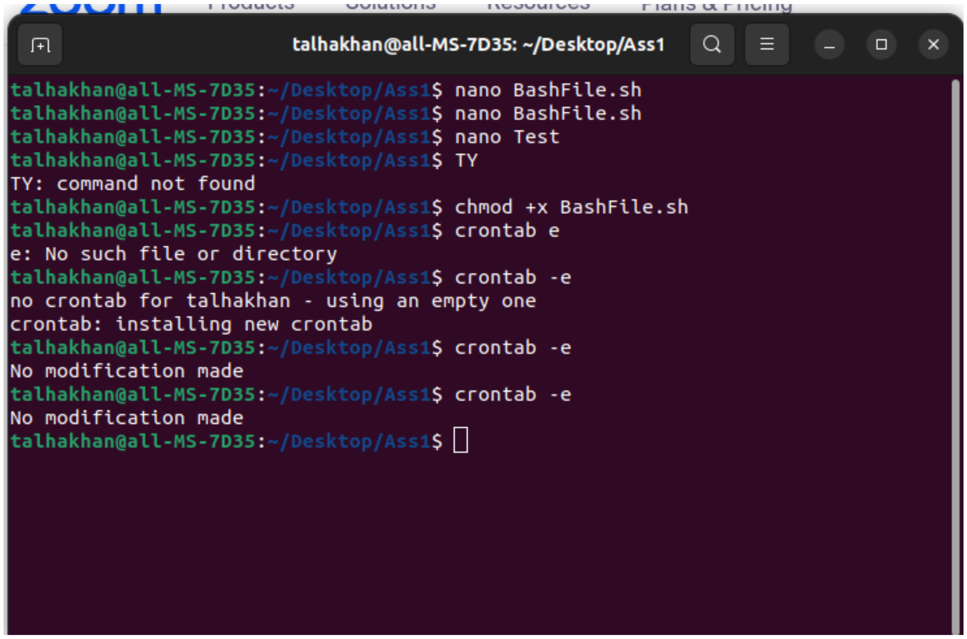
Assignment 01

Talha khan (2303.009.KHI.DEG)

Muhammad Moiz Khan (2303.022.KHI.DEG)

Task:

On a linux server setup a cron job for copying example data with rsync periodically. Ensure the copying is handled in the background and independently of the user session.

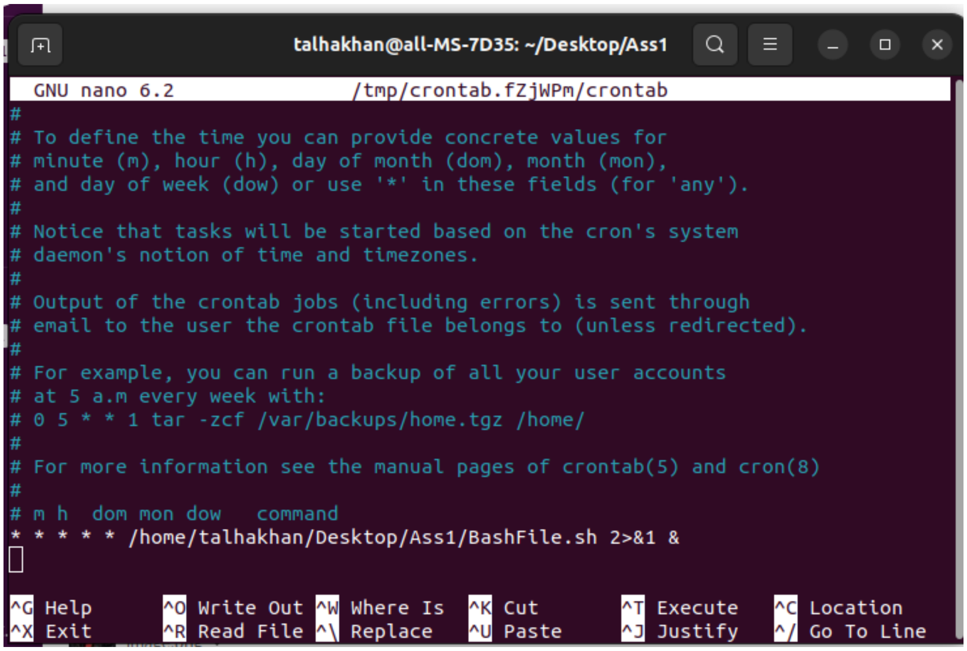


```
talhakhan@all-MS-7D35:~/Desktop/Ass1$ nano BashFile.sh
talhakhan@all-MS-7D35:~/Desktop/Ass1$ nano BashFile.sh
talhakhan@all-MS-7D35:~/Desktop/Ass1$ nano Test
talhakhan@all-MS-7D35:~/Desktop/Ass1$ TY
TY: command not found
talhakhan@all-MS-7D35:~/Desktop/Ass1$ chmod +x BashFile.sh
talhakhan@all-MS-7D35:~/Desktop/Ass1$ crontab e
e: No such file or directory
talhakhan@all-MS-7D35:~/Desktop/Ass1$ crontab -e
no crontab for talhakhan - using an empty one
crontab: installing new crontab
talhakhan@all-MS-7D35:~/Desktop/Ass1$ crontab -e
No modification made
talhakhan@all-MS-7D35:~/Desktop/Ass1$ crontab -e
No modification made
talhakhan@all-MS-7D35:~/Desktop/Ass1$ 
```

Steps:

1. Open the Linux terminal.
2. For creating a new bashfile for editing we'll run **"nano bashfile.sh"** which opens the nano text editor and creates or a file.
3. Once you run this command, the nano editor will open, and you can start editing the file.
4. You can use the keyboard to type in commands or add text to the file, and use various nano keyboard shortcuts to save, exit, and perform other operations.
5. Once you're finished editing the file directory, you can save your changes and exit nano by pressing **"Ctrl+X".**
6. Next step is to set an executable permission on the file from the respective directory which is achieved by the command **"chmod +x bashfile.sh"**.

7. The **"chmod"** command is used to change the permissions of a file or directory when it comes to Linux and Unix Operating system.
8. Secondly the "+x" option grants the execute permission to the file which executes the file as a program or script.
9. To set up a cron job for copying example data with rsync periodically on a Linux server we'll use **"crontab -e"**.

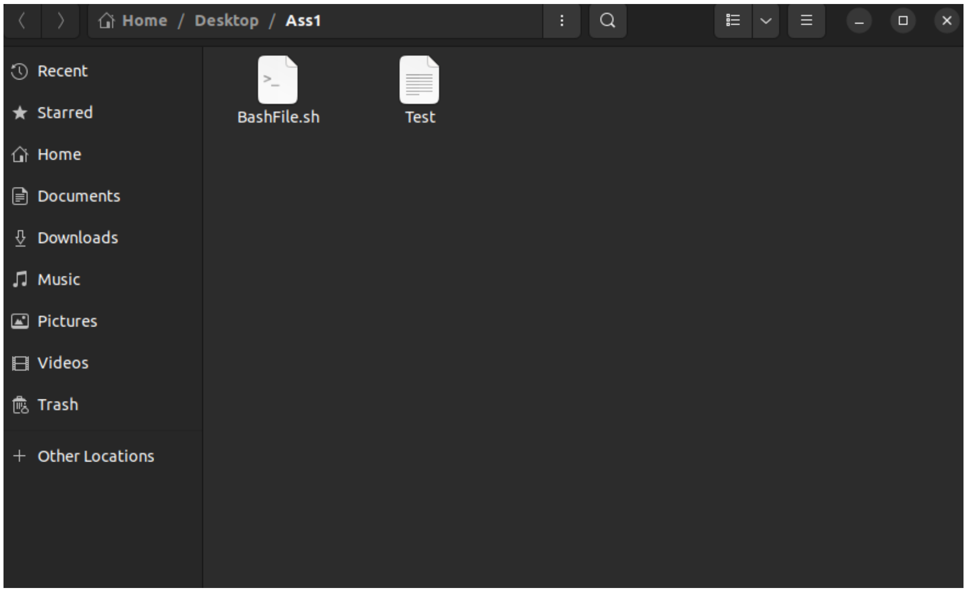


```
 GNU nano 6.2              /tmp/crontab.fZjWPm/crontab
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# 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
* * * * * /home/talhakhan/Desktop/Ass1/BashFile.sh 2>&1 &


^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

10. This command will take you to run the rsync command with the specified options every minute. You can change the **"* * * * *"** part to specify a different schedule.
11. The **"2>&1 &"** part of the command redirects output to /dev/null and runs the command in the background, independently of the user session.
12. This means that the rsync process will continue running even if you log out of the server.

Final Output:


Home / Desktop / Ass1
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
BashFile.sh
Test