Assignment no:1.1

Faiza Gulzar Ahmed (2303.khi.deg.001)

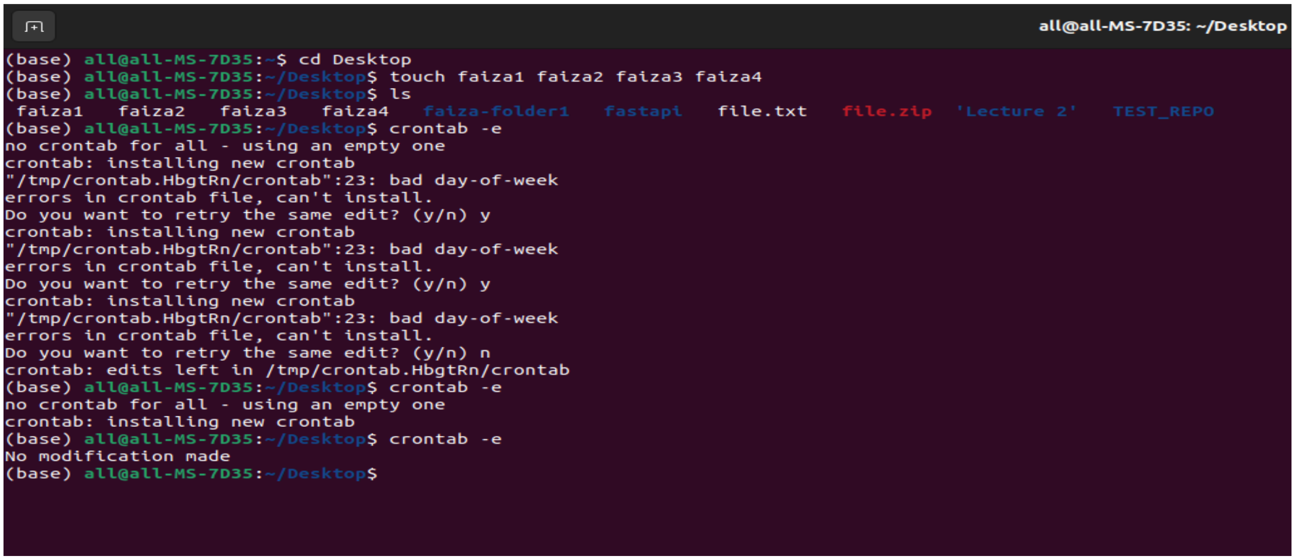
Muhammad Faizan Rafique (2303.005.KHI.DEG)

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.

Step1:

Create a Folder1:

Firstly, open a terminal and I write a cd Desktop command and create a three touch files faiza1 faiza2 faiza3 faiza4 and check the ls command show the folders. One folder faiza-folder1 to saves another folder like faiza-folder2.

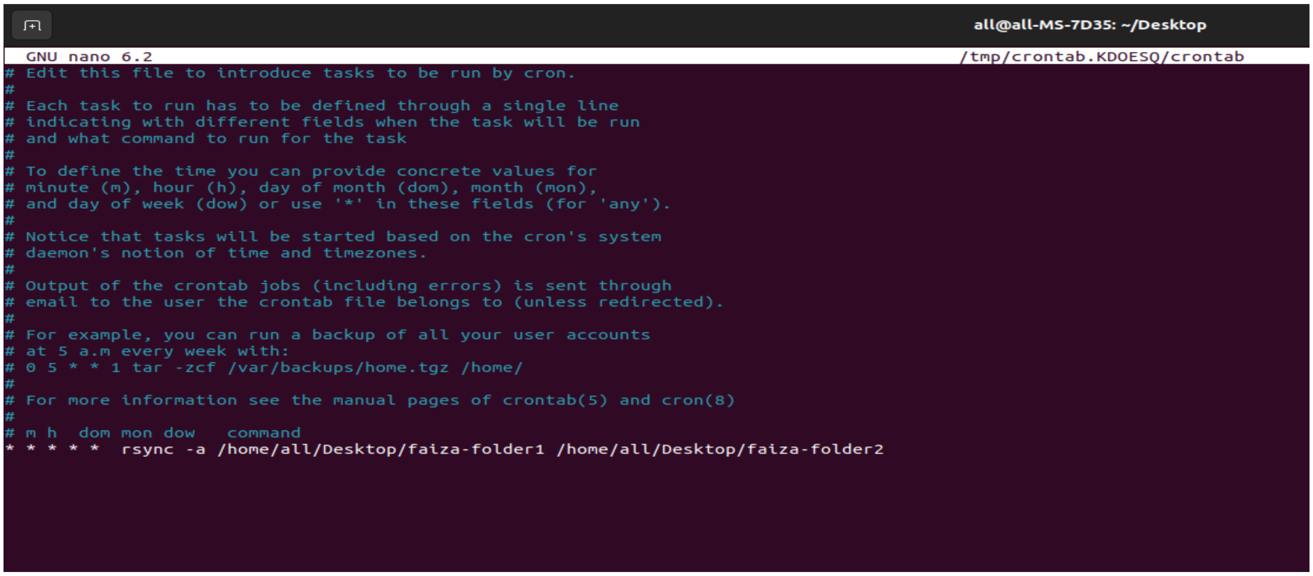


```
(base) all@all-MS-7D35:~$ cd Desktop
(base) all@all-MS-7D35:~/Desktop$ touch faiza1 faiza2 faiza3 faiza4
(base) all@all-MS-7D35:~/Desktop$ ls
 faiza1   faiza2   faiza3   faiza4   faiza-folder1   fastapi   file.txt   file.zip  'Lecture 2'   TEST_REPO
(base) all@all-MS-7D35:~/Desktop$ crontab -e
no crontab for all - using an empty one
crontab: installing new crontab
"/tmp/crontab.HbgtRn/crontab":23: bad day-of-week
errors in crontab file, can't install.
Do you want to retry the same edit? (y/n) y
crontab: installing new crontab
"/tmp/crontab.HbgtRn/crontab":23: bad day-of-week
errors in crontab file, can't install.
Do you want to retry the same edit? (y/n) y
crontab: installing new crontab
"/tmp/crontab.HbgtRn/crontab":23: bad day-of-week
errors in crontab file, can't install.
Do you want to retry the same edit? (y/n) n
crontab: edits left in /tmp/crontab.HbgtRn/crontab
(base) all@all-MS-7D35:~/Desktop$ crontab -e
no crontab for all - using an empty one
crontab: installing new crontab
(base) all@all-MS-7D35:~/Desktop$ crontab -e
No modification made
(base) all@all-MS-7D35:~/Desktop$
```

Step 2:

Open the Crontab-e

Open the crontab-e and save the file1 minutes * file save source to destination use this command rsync –a source to a destination set the path folder1 and folder2. Crontab executes the task and destination times and scheduling.

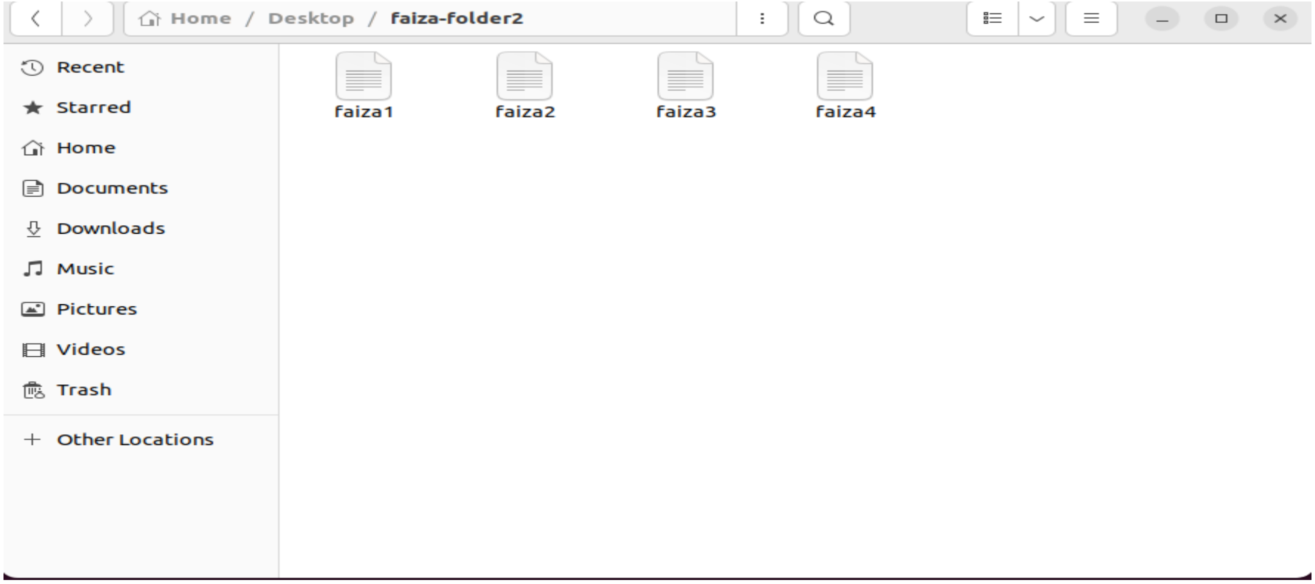


```
  GNU nano 6.2                                                              /tmp/crontab.KDOESQ/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command 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
* * * * *  rsync -a /home/all/Desktop/faiza-folder1 /home/all/Desktop/faiza-folder2
```

Step 3:

Destination Files:

Show the destination file.



Explanation of the Assignments Conclusion:

First, I made two folders named faiza_1 and faiza_2 then I saved three text files in faiza_1 and showed it by using **ls command** and faiza_2 was empty at that time shown in terminal.
Secondly, I opened crontab by command **crontab –e** after that I scheduled it for one minute and apply rsync with it to transfer files periodically by giving it source and destination path.
At the end I used **crontab –r** to stop the crontab.

Citation:
https://www.freecodecamp.org/news/cron-jobs-in-linux/