

Assignment #1

Submission

This assignment was completed by Muzammil Mehmood entirely, due to unavailability of any peer

Problem Statement

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.

Solution

Firstly, we will break the problem it is important to first understand the concept of a cron job. Essentially, cron is a utility program that allows users to schedule tasks to be executed repeatedly at specific times. These tasks are commonly referred to as cron jobs.

In addition to understanding cron, another key aspect of this project is the utilization of a file synchronization tool known as Rsync. This tool, which is short for "remote sync," is capable of synchronizing files both locally and remotely.

Step 1. Editing the Crontab

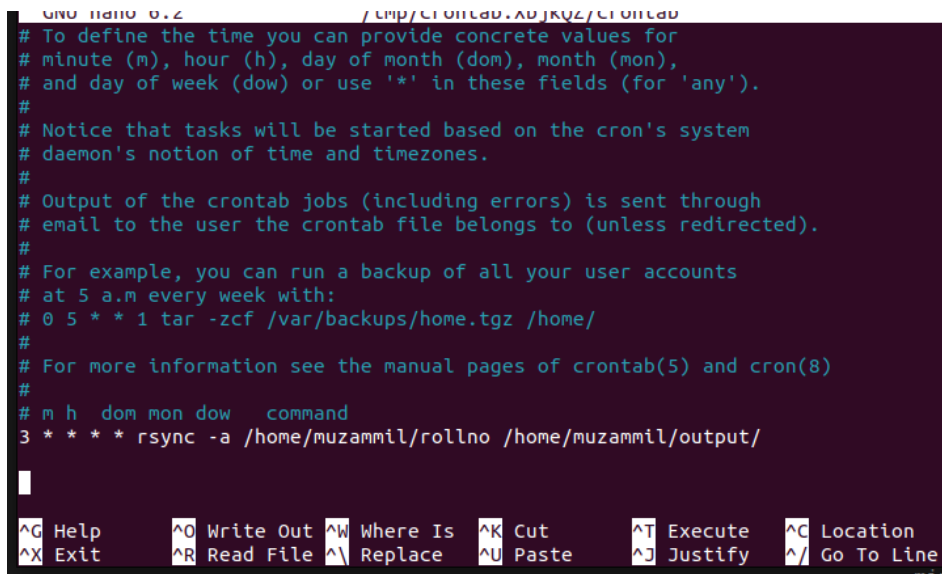
```
muzammil@all-MS-7D35:~$ crontab -e  
crontab: installing new crontab
```

-e will let you edit the content of crontab.

You can also use other command like

- -l will let you see existing crontab
- -r will let you delete the content of crontab

Step 2. Putting the Command on Crontab editor

A screenshot of the crontab editor interface. The editor shows a file named `/tmp/crontab.XDJKQZ/crontab` with the following content:

```
# 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
3 * * * * rsync -a /home/muzammil/rollno /home/muzammil/output/
```

The editor has a status bar at the bottom with various keyboard shortcuts: `^G Help`, `^O Write Out`, `^W Where Is`, `^K Cut`, `^T Execute`, `^C Location`, `^X Exit`, `^R Read File`, `^_ Replace`, `^U Paste`, `^J Justify`, and `^_ Go To Line`.

The asterisk (*) represent the schedule of cronjob you can edit this to activate this job at any time you want

First * represent minute(0-59)

Second * represent hour(0-23)

Third * represent day of month(1-31)

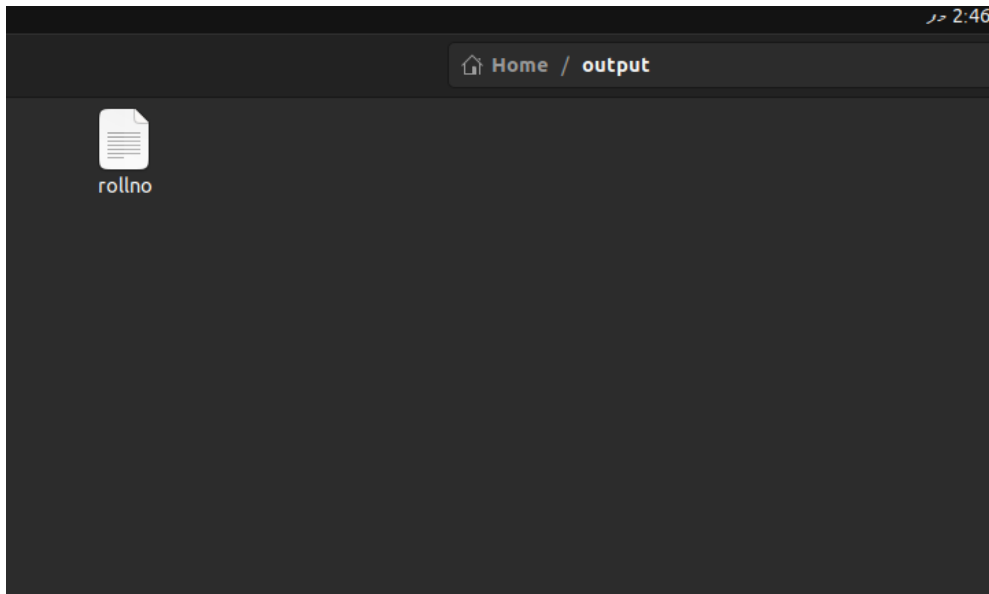
Fourth * represent month (1-12)

Fifth * represent day of week (0-6)

I have schudule my job for running at three minute using rsync and input the source of file and destination of file.

Also used nohup command this will keep the process running even after existing the terminal.

Result



This is file, I have used to show to demonstration of this cronjob, This file is currently present in Home directly and I will copy that file into folder I have created named Output folder.

