CSCI 130 Linux Fundamentals - Week 11 (November 4, 2025) Assignment Submission

Assignment Details

Part 1 - Scheduling a Cron Job

Install a cron command to write the time stamp and the listing of your directory to the log file to the crontab and schedule it to run every five minutes. Save the crontab and close it. Wait a few minutes and check out the log file.

How would you schedule a command to run every 1 day of the month?

Part 1 - How to submit

Write brief explanation how you make a crontab entry. Add comments about what you did and what was the result.

Part 2 - Sending Email from Command Line

Practice sending email to the user from the command line. Create a simple script that sends result of its execution (it would be sufficient just to do a listing of the directory) to the user by email and schedule that script to run from the Crontab.

Part 2 - How to submit

Describe how you schedule the crontab job.

Part 3 - Text Editing with vi

Using the attached file as example, run a sample session of text editing with the vi. You do not need to repeat all commands exactly; it'll be sufficient to familiarize yourself with the editor.

Part 3 - How to submit

Write a brief description what you.

Assignment Submission Notes

Part 1 - Timestamp and Directory Listing Cron Job

To schedule a cron job that writes the timestamp and directory listing to a log file every five minutes, I followed these steps:

1. Opened the terminal and accessed the crontab editor by running the command (this opened in Vim for me):

   ```
   crontab -e
   ```

2. Added the following line to the crontab file to schedule the job:

   ```
   */5 * * * * echo "Timestamp: $(date)" >> /home/rodney-gauna/cron_log.txt && ls >> /home/rodney-gauna/cron_log.txt
   ```

   This command appends the current timestamp and the directory listing to `cron_log.txt` every five minutes.

3. Saved and exited the crontab editor.

4. Waited a few minutes and checked the contents of `cron_log.txt` to verify that the cron job was executing as expected.

Part 1 - Scheduling a Command Every First Day of the Month

To schedule a command to run every first day of the month, I would add the following line to the crontab file:

```
0 0 1 * * /path/to/your/command.sh
```

This command would run at midnight on the first day of every month.

Part 2 - Sending Email from Command Line Explanation

To send an email from the command line and schedule it via cron, I followed these steps:

1. Created a simple script named `send_email.sh` with the following content:

   ```
   #!/bin/bash
   ls | mail -s "Directory Listing - `date`" user@localhost
   ```

   This script lists the contents of the current directory and sends it via email with the subject containing the current date.

2. Made the script executable:

   ```
   chmod +x send_email.sh
   ```

3. Opened the crontab editor:

   ```
   crontab -e
   ```

4. Added the following line to the crontab file to schedule the email script to run every five minutes:

   ```
   */5 * * * * /path/to/send_email.sh
   ```

5. Saved and exited the crontab editor.

Part 3 - Text Editing with vi Explanation

Content in the "Examples with vi.pdf" file:

```
#inserting listing
:r!ls -l | head –10
#adding line numbering
:set nu
#assuming listing is inserted into line number 3 , going to that line
3G
#shifting 10 lines one tab to the right
10>>
#adding line numbers
3,11!awk '{print NR""$0}'
#sorting by file name in reverse
:3,11!sort -k10 –r
#remove the numbers on the left (they are now not in the order)
:3,11s/^\d\+//
#re-inserting the correct line numbers
3,11!awk '{print NR""$0}'
#replacing user name (liveuser) with nobody
3,11s/liveuser/nobody/g
#replacing user name nobody with noname, leaving group name intact
3,11s/liveuser/noname/
#flipping user name and group name
3,11s/\(\d\+\D\+\d \)\([^ ]\+\) \([^ ]\+\)/\1\3 \2/
#abbreviation
:ab LF CS130 Linux fundamentals
#Then type: "I am taking LF class this semester" to see the abbreviation
works
#emulating ^C-^V
:map ^V^C "ayy
:map ^V^V^V^V "ap
#then do ^C on some line and paste it with ^V
```

After following the steps in the provided vi example, I was able to successfully edit the text file using various vi commands, including inserting command output, adding line numbers, sorting lines, and performing substitutions. This exercise helped me become more comfortable with using vi for text editing tasks.

Here is the output of the final edited text after performing the vi commands:

```
```

```
1  -rw-rw-r--  1 nobody nobody   347765 Sep 16 19:54 all_output.txt
10  -rw-rw-r--  1 nobody nobody        0 Sep 16 19:44 file1.txt
9  -rw-rw-r--  1 nobody nobody     3497 Sep 16 18:09 env-var
8  drwxr-xr-x  2 nobody nobody     4096 Oct 21 20:13 Downloads
7  drwxr-xr-x  6 nobody nobody     4096 Sep 16 19:43 Documents
6  drwxr-xr-x  2 nobody nobody     4096 Aug 19 14:15 Desktop
5  -rw-rw-r--  1 nobody nobody    92438 Sep 16 19:53 combined_output.txt
4  -rw-rw-r--  1 nobody nobody    92455 Sep 16 19:55 both_streams.txt
2  -rw-rw-r--  1 rodney-gauna rodney-gauna   255298 Sep 16 19:52 all_txt_files.txt
I am taking CS130 Linux fundamentals class this semester
3  -rw-rw-r--  1 nobody nobody        0 Sep 16 17:09 bash_profile.bk
```