10.3.10 Practice Questions

Candidate: Ethan Bonavida (suborange)
Date: 12/5/2022 11:17:32 pm • **Time Spent:** 14:54

Score: 80% Passing Score: 80%

Question 1: ✔ Correct

Which of the following commands removes a job from the at queue? (Select TWO. Each answer is an independent solution.)

- [ ] **atq**
- [ ] **at -f**
- [ ] **at -l**
- [x] **atrm**
- [x] **at -d**

Explanation

at -d or **atrm** removes a job from the at queue. Use commas to separate multiple jobs. For example:

- **at -d 2,3** removes jobs 2 ad 3 from the at queue.
- **atrm 4** removes job 4 from the at queue.

Use **at -f** to schedule tasks in a file to run at the designated time (like a shell script, for example). Use **at -l** or **atq** to list the tasks in the at queue for the current user.

- When run as root, **atq** or **at -l** lists all the jobs in queue.
- When run as a user other than root, **at** lists only the jobs for the user.

References

10.3.1 at Task Scheduling

10.3.2 Scheduling Tasks with at

10.3.3 at Command Facts

q_at_cf_lp5_01.question.fex

Question 2: ✔ Correct

You want to keep the *gshant* user from using the **at** command. What is the full path and filename of the file you should edit?

/etc/at.deny ✔

Explanation

/etc/at.deny specifies users who cannot use the **at** command, whereas /etc/at.allow specifies users who can use the **at** command.

References

10.3.1 at Task Scheduling

10.3.2 Scheduling Tasks with at

10.3.3 at Command Facts

q_at_cf_lp5_04.question.fex

Question 3: ✔ Correct

Which of the following statements best describes the effects of having only the *gshant* user account listed in the /etc/at.allow file?

- All users but gshant can use the **at** command.
- Only gshant can use the **at** command.
- Only root can use the **at** command.
- ➡ Only gshant and root can use the **at** command.

Explanation

In this case, only gshant and root can use the **at** command. **at** uses configuration files to specify standard user accounts that can and cannot use the **at** command. /etc/at.allow specifies users who can use the **at** command. /etc/at.deny specifies users who cannot use the **at** command.

References

10.3.1 at Task Scheduling

10.3.2 Scheduling Tasks with at

10.3.3 at Command Facts

q_at_cf_lp5_05.question.fex

Question 4: ✕ Incorrect

What should you enter at the command prompt to remove tasks 2 and 3 in the **at** queue?

~~atrm 2,3~~ atrm 2 3

Explanation

Use **at -d** or **atrm** followed by the task number(s) of the jobs you want to remove from the **at** queue. Use spaces to separate multiple jobs. For example, **atrm 2 3** removes jobs 2 and 3 from the **at** queue.

References

10.3.1 at Task Scheduling

10.3.2 Scheduling Tasks with at

10.3.3 at Command Facts

q_at_cf_lp5_06.question.fex

Question 5: ✔ Correct

Which command could you use to verify whether a crontab file exists for the *thobbs* user?

- **crontab -e -u thobbs**
- **crontab -a -u thobbs**
- **crontab -r -u thobbs**
- **crontab -l -u thobbs**

Explanation

Use the **crontab -l -u** ***user*** option to see whether this file exists.

Use the **-e** option to edit the crontab, use **-r** to remove it, and use **-a** to append to the existing file.

References

10.3.4 cron Task Scheduling

10.3.5 Scheduling Tasks with cron

10.3.6 cron Facts

15.2.7 File Auditing Facts

q_schedule_lp5_01.question.fex

Question 6: ✔ Correct

After logging in as root, you need to manage the crontab files for your Linux system.

Which command should you use to edit the crontab file for the *gshant* user account?

- **crontab -ue gshant**
- **vi /etc/crontab**
- **crontab -ur gshant**
- **crontab -ul gshant**

Explanation

Use **crontab -ue gshant** to edit the crontab file for the *gshant* user account. Use crontab to manage the /var/spool/cron/username crontab file. Be aware of the following options:

- **-u *username*** specifies a user for the **-e**, **-l**, and**-r** options.
- **-e** edits the crontab file in vim for the current user.
- **-l** displays the contents of the crontab file.
- **-r** removes the crontab file.

Use vi /etc/crontab to open and edit the /etc/crontab file in Vim. The /etc/crontab file holds entries that direct commands to execute at a specific time. The/etc/crontab file is for custom task schedules that run system-wide, can only be edited by the root user, and runs each entry as the root user.

References

10.3.4 cron Task Scheduling

10.3.5 Scheduling Tasks with cron

10.3.6 cron Facts

15.2.7 File Auditing Facts

q_schedule_lp5_02.question.fex

Question 7: ✔ Correct

You are editing the crontab file and want an entry to run every hour at five minutes past the hour.

Which of the following entries will accomplish this task?

- **5 * * * * /home/emmett/example.sh**
- *** 5 * * * /home/emmett/example.sh**
- *** *5 * * /home/emmett/example.sh**
- *** * * 5 * /home/emmett/example.sh**
- *** * * * 5 /home/emmett/example.sh**

Explanation

To run the command every hour at a specific minute, place the minute value (5) in the first field. The first field of the crontab entry holds the minute specification. The second field is used for hours. The third field identifies the day of the month. The fourth field specifies the months during which to run.

References

10.3.4 cron Task Scheduling

10.3.5 Scheduling Tasks with cron

10.3.6 cron Facts

15.2.7 File Auditing Facts

q_schedule_lp5_03.question.fex

Question 8: ✕ Incorrect

What is the complete path to the directory that will hold the crontab file for the *gshant* user account?

~~/var/spool/cron/tabs~~ /var/spool/cron/

Explanation

/var/spool/cron holds a personal crontab file for specific user accounts. The *gshant* user account will have /var/spool/cron/***gshant*** as the personal crontab file. The cron daemon only checks the file of the current user.

References

10.3.4 cron Task Scheduling

10.3.5 Scheduling Tasks with cron

10.3.6 cron Facts

15.2.7 File Auditing Facts

q_schedule_lp5_05.question.fex

Question 9: ✔ Correct

Which file should you edit to schedule a task to execute every week on Saturday?

- /etc/cron.monthly
- ➡ /etc/cron.weekly
- /etc/cron.hourly
- /etc/cron.daily

Explanation

Use the /etc/cron.weekly file to execute scripts on a weekly interval. Use:

- /etc/cron.hourly to execute scripts on an hourly interval.
- /etc/cron.daily to execute scripts on an daily interval.
- /etc/cron.monthly to execute scripts on an monthly interval.

References

10.3.4 cron Task Scheduling

10.3.5 Scheduling Tasks with cron

10.3.6 cron Facts

q_schedule_lp5_06.question.fex

Question 10: ✔ Correct

You want to keep the *gshant* user from editing his respective crontab file in **/**var/spool/cron, but still allow all other users on the system to edit their respective crontab file.

What is the full path and filename of the file you should edit?

/etc/cron.deny ✔

Explanation

The /etc/cron.deny file excludes users who can edit their personal crontab file. If **/**etc/cron.deny file exists, users listed therein are not allowed to edit their personal crontab file.

References

10.3.4 cron Task Scheduling

10.3.5 Scheduling Tasks with cron

10.3.6 cron Facts

15.12.1 Security Best Practices

15.12.2 Security Best Practices Facts

q_schedule_lp5_07.question.fex

Question 11: ✔ Correct

Which file should you edit if you want to permit specific users to edit their respective crontab file, but deny all other users on the system from editing their crontab file?

- /etc/crontab
- /etc/cron.deny
- /etc/cron.permit
- ➡ /etc/cron.allow

Explanation

The /etc/cron.allow file includes users who can edit their personal crontab file. If /etc/cron.allow file exists, only users listed therein are allowed to edit /var/spool/cron/***username***.

The /etc/crontab (cron table) file holds entries that direct commands to execute at a specific time for the whole system. The cron daemon only checks the file of the current user. The /etc/cron.deny file excludes users who can edit their personal crontab file. If /etc/cron.deny file exists, users listed therein are not allowed to edit their personal crontab file. There is no /etc/cron.permit file.

References

10.3.4 cron Task Scheduling

10.3.5 Scheduling Tasks with cron

10.3.6 cron Facts

q_schedule_lp5_08.question.fex

Question 12: ✔ Correct

If a system is down at the time a recurring regularly scheduled task is supposed to run, which task scheduling service will run the task when the system is back up again?

- cron
- recron
- ➡ anacron
- at

Explanation

If a system is down at the time a recurring regularly scheduled task is supposed to run and anacron was used to schedule the task, then anacron will run the task when the system is back up again.

If a system is down at the time a recurring regularly scheduled task is supposed to run and cron was used to schedule the task, then the task will be skipped, and it won't be run until the next time it is scheduled to run.

The at daemon can only be used to schedule a single occurrence of a task to be run at a specific time in the future. If the system time is down when the task was supposed to run, it will not be run.

There is no recron service on Linux.

References

10.3.8 anacron Facts

q_anacron_lp5_01.question.fex

Question 13: ✔ Correct

What is the full path and filename of the file that is used to schedule tasks for the anacron daemon?

/etc/anacrontab ✔

Explanation

/etc/anacrontab is the full path and filename of the file that is used to schedule tasks for the anacron daemon.

References

10.3.8 anacron Facts

q_anacron_lp5_02.question.fex

Question 14: ✔ Correct

You have an anacrontab file with the following settings:

```
RANDOM_DELAY=35
START_HOURS_RANGE=17-23
#period in days delay in minutes job-identifier command
1 5 cron.daily nice run-parts /etc/cron.daily
7 20 cron.weekly nice run-parts /etc/cron.weekly
@monthly 50 cron.monthly nice run-parts /etc/cron.monthly
```

Between which hours of the day will tasks scheduled with anacron start to run?

Between 5:00 p.m. and 11:00 p.m. ✔

If the system was down during the time period a weekly task was scheduled to run, what is the minimum amount of time anacron will wait to run a task after the system is back up?

20 minutes ✔

If the delay for daily tasks is 5 minutes, how much time will anacron add to the delay of 5 minutes before it runs the scheduled daily task?

A randomly chosen number of minu... ✔

Explanation

Scheduled tasks will start to run between hour 17 (which is 5:00 p.m.) and hour 23 (which is 11:00 p.m.). Tasks will run between 5:00 p.m. and 11:00 p.m.

2. If the system was down during the time period a weekly task was scheduled to run, 20 minutes is the minimum amount of time anacron will wait to run the task after the system is back up. Remember that with a random delay set at 35 minutes, anacron will add between 0 and 35 minutes to that 20-minute delay.

3. If the delay for daily tasks is 5 minutes, anacron will add between 0 and 35 minutes to the 5-minute delay before it runs the scheduled daily task. This means the scheduled task will start between 5 and 40 minutes after the system comes back up again.

References

10.3.8 anacron Facts

q_anacron_lp5_03.question.fex

Question 15: ✕ Incorrect

anacron creates a timestamp file that tells you the last time a regularly scheduled job was run.

What is the full path and name of the directory that contains these timestamp files?

- /var/jobs/anacron/
- ~~/etc/anacron.d/~~
- /etc/cron.d/anacrontab/
- ➡ /var/spool/anacron/

Explanation

anacron creates a timestamp file in the **/var/spool/anacron/** directory every time a regularly scheduled job is run so you can see the last time a given job ran. The /var/ directory is the standard directory for log files of this nature.

References

10.3.8 anacron Facts

q_anacron_lp5_05.question.fex