10.2.11 Practice Questions

Candidate: Ethan Bonavida (suborange)
Date: 12/5/2022 12:30:57 am • **Time Spent:** 05:38

Score: 80% Passing Score: 80%

Question 1: ✔ Correct

A developer calls you with a problem. He was trying to debug a new daemon and mistakenly placed it on the production machine instead of on a lab machine. To ensure that the CPU gives preference to other processes, you need to lower the daemon's priority. The PID number is 2345.

Which command would you use to decrease the daemon's priority?

- **renice -n -10 2345**
- **nice -n 0 2345**
- **renice -n 0 2345**
- ➡ **renice -n 10 2345**

Explanation

Use **renice -n 10 2345** to decrease the priority of process 2345. The **renice** command assigns a new priority to a process that has already started using the process's PID. Be aware of the following:

- Values range from 19 (lowest priority) to -20 (highest priority). The higher the number, the lower priority the job receives in the system.
- Zero (0) is the default nice value for processes not executed with the **nice** command.

Use the **nice** command to start a process with a higher or lower priority.

References

10.1.9 Process Display Facts

q_pstree_lp5_01.question.fex

Question 2: ✔ Correct

You need to determine the priority of several processes. Which command should you enter to identify the process ID and **nice** value for each process?

- **jobs**
- **renice**
- **nice**
- **top**

Explanation

top displays an interactive listing of processes with Process ID (PID), uptime, load, CPU status, memory, and **nice** value information.

jobs displays the active jobs and their job ID numbers. **nice** starts a command with a higher or lower priority. **renice** assigns a new priority to a process that has already started.

References

10.1.9 Process Display Facts

q_pstree_lp5_03.question.fex

Question 3: ✕ Incorrect

What would you enter at the command prompt to start the **gedit** process with the highest priority possible?

~~renice -20 gedit~~ nice -n -20 gedit

Explanation

nice -n -20 gedit starts the gedit process with a highest priority possible. The **nice** command starts a command with a higher or lower priority.

- Nice values range from 19 (lowest priority) to -20 (highest priority). The higher the number, the lower priority the job receives.
- Use **-n** to specify the priority value. If no value is specified, the process starts with 10 as the default.
- Zero (0) is the default nice value for processes not executed with the **nice** command.

References

10.2.5 Process Management Facts

q_pstree_lp5_04.question.fex

Question 4: ✔ Correct

What is the default nice value for a process not executed with the **nice** command?

0 ✔

Explanation

Zero (0) is the default nice value for processes not executed with the **nice** command. To start a command with a higher or lower priority, use **nice**. To assign a new priority to a process that has already started, use **renice**.

References

10.2.5 Process Management Facts

q_pstree_lp5_05.question.fex

Question 5: ✔ Correct

Which of the following commands starts the **gedit** program with a priority one above the default nice priority?

- **nice -n -1 gedit**
- **nice -n 1 gedit**
- **nice -n -20 gedit**
- **nice -n -19 gedit**

Explanation

nice -n -1 gedit starts the gedit process with a priority one above the default. Zero (0) is the default nice value for processes not executed with the **nice** command. The **nice** command starts a command with a higher or lower priority.

- Nice values range from 19 (lowest priority) to -20 (highest priority). The higher the number, the lower priority the job receives in the system.
- Use **-n** to specify the priority value. If no value is specified, the process starts with 10 as the default.

References

10.2.5 Process Management Facts

q_pstree_lp5_06.question.fex

Question 6: ✔ Correct

What command would you enter at the command prompt to start the **gedit** program in the background?

gedit &

Explanation

gedit & starts the gedit program in the background. The ampersand (&) symbol starts a process in the background, leaving the shell available for other commands.

References

- **10.2.1 Process Management**
- **10.2.2 Switching Foreground and Background Processes**
- **10.2.5 Process Management Facts**
- **10.2.6 Process Termination**
- **10.2.7 Terminating Processes**

q_pstree_lp5_08.question.fex

Question 7: ✕ Incorrect

You have several processes running in the background, as shown by the **jobs** command below:

```
[1]+ Running gedit &
[2]+ Stopped crontab e
```

Which command will bring the **gedit** program to the foreground?

- ~~**fg [1]**~~
- **fg edit**
- **bg 1**
- **fg 1**

Explanation

Use the **fg** command followed by the job ID (1, 2, 3, etc.) to bring a job to the foreground. The job ID is shown within brackets from the **jobs** command output.

Use the **bg** command to send processes to the background.

References

10.2.1 Process Management

10.2.2 Switching Foreground and Background Processes

10.2.5 Process Management Facts

q_pstree_lp5_09.question.fex

Question 8: ✔ Correct

You have a gedit process with a job ID of 6. Which command would you enter at the command prompt to send the process to the background?

bg 6 ✔

Explanation

bg 6 sends the process with job ID 6 to the background.

References

10.2.1 Process Management

10.2.2 Switching Foreground and Background Processes

10.2.5 Process Management Facts

q_pstree_lp5_10.question.fex

Question 9: ✔ Correct

While editing a script using gedit, you discover that you no longer have access to your current terminal session and need to check the values of some variables the script is using.

Which of the following commands will send the gedit process to the background so you can regain access to the current terminal?

- ➡ Ctrl+Z
- Ctrl+C
- Ctrl+Alt+F2
- **bg**

Explanation

Ctrl+Z will pause the gedit process and send it to the background, allowing access to terminal.

Ctrl+C will send a break signal to the process to terminate.

Ctrl+Alt+F2 will switch to a virtual terminal tty2. This will not accomplish the desired task.

bg will send a process to the background; however, in this case, there is no access to the terminal command line to enter the command.

References

10.2.5 Process Management Facts

q_pstree_lp5_ctrl_z.question.fex

Question 10: ✔ Correct

What is the result of the **nohup gedit &** command?

- **gedit** will start in the background and terminate after logging out of the shell.
- ➡ **gedit** will start in the background and persist in the background after logging out of the shell.
- **gedit** will start in the foreground and terminate after logging out of the shell.
- **gedit** will start after logging out of the shell.

Explanation

nohup gedit & starts the **gedit** process in the background and leaves it running after logging out of the shell. **nohup**:

- Allows a command or shell script to continue running in the background after logging out from a shell.
- Does not automatically put the command it runs in the background. Use the ampersand (&) symbol to start a process in the background.

References

10.2.6 Process Termination

10.2.8 Process Termination Facts

q_ps_kill_lp5_02.question.fex

Question 11: ✕ Incorrect

You need to manage a process in the foreground by pressing Ctrl+C on the keyboard. Which signal code is sent to the process?

- SIGTERM
- SIGKILL
- ~~SIGHUP~~
- SIGINT

Explanation

The Ctrl+C key sequence sends a SIGINT interrupt signal to the process.

SIGHUP stops and restarts the process with the same process ID number. This also causes the process to reload its configuration file. SIGKILL invokes a hard kill of the process that may not allow the process to unhook its resources. This means that RAM and other resources allocated to the process usually remain allocated to the process. SIGTERM stops the process after allowing it to unhook its resources.

References

10.2.6 Process Termination

q_ps_kill_lp5_04.question.fex

Question 12: ✔ Correct

A developer calls you with a problem. She was trying to debug a new daemon and mistakenly placed it on the production machine instead of on a lab machine. It has now entered runaway mode. The PID number is 2345. She attempted to stop the process with the standard **kill** command, but it had no effect.

Which command would you use to assure that the process will terminate?

- **kill 2345 -9**
- **kill -NOW 2345**
- **kill 2345 -NOW**
- ➡ **kill -9 2345**

Explanation

Use **kill -9 2345** to terminate PID 2345 with the strongest signal possible (-9). The other choices presented do not have the syntax required to terminate the runaway process.

References

10.2.6 Process Termination

10.2.7 Terminating Processes

15.3.2 Disable Login

q_ps_kill_lp5_05.question.fex

Question 13: ✔ Correct

A service specialist from your company calls you from a customer's site. He is attempting to install an upgrade to the software, but it will not install; instead, he receives messages stating that he cannot install the program until all non-core services are stopped. You cannot remotely access the machine, but the representative tells you that there are only four processes running, and their PID numbers are 1, 10, 100, and 1000.

With no further information available, which command would you recommend the representative run?

- **kill 1**
- **kill 10**
- **kill 100**
- ➡ **kill 1000**

Explanation

As a system starts, it assigns PID numbers to the processes that are spawned and increments the numbers. The first processes to start have the lowest PID numbers. A PID of 1000 indicates that many other processes came before it, and it probably isn't a critical application. Lower PID numbers are likely to be critical processes.

References

10.2.6 Process Termination

10.2.7 Terminating Processes

15.3.2 Disable Login

q_ps_kill_lp5_06.question.fex

Question 14: ✔ Correct

After a severe lightning strike nearby, a number of processes seem to be running on the server in runaway mode. Which utility terminates these processes by name, and not just by process ID number?

- **endall**
- **down**
- **closeproc**
- **killall**

Explanation

Use the **killall** utility to terminate processes by name.

down, **endall**, and **closeproc** are not actual Linux commands.

References

10.2.6 Process Termination

10.2.7 Terminating Processes

15.3.2 Disable Login

q_ps_kill_lp5_07.question.fex

Question 15: ✔ Correct

The /sbin/grpck process is not responding and needs to be killed. There are multiple instances that were spawned and need to be terminated.

Which of the following commands will terminate all instances of the grpck process?

- **pkill -SIGTERM -f grpck**
- **pkill -HUP 2583**
- **ps aux | grep "grpck"**
- **ps -A | grep grpck**

Explanation

pkill searches for processes that match the search criteria specified and then send them a kill signal. **pkill -SIGTERM -f grpck** will send the terminate signal to all processes named grpck.

ps -A | grep grpck displays all processes that include the name *grpck*.

pkill -HUP 2583 will force the reload of a configuration file for process ID 2583.

ps aux | grep "grpck" will list process information for any process that includes the term *grpck*.

References

10.1.9 Process Display Facts

q_ps_kill_lp5_pkill.question.fex