2.1.7 Practice Questions

Candidate: Ethan Bonavida (suborange)
Date: 11/2/2022 3:30:47 pm • **Time Spent:** 00:29

Score: 100% Passing Score: 80%

Question 1: ✔ Correct

Which of the following is the standard shell for MOST Linux computers?

- tcsh
- ➡ Bourne-again shell (bash)
- C-shell
- Bourne shell
- Korn

Explanation

The Bourne-again shell (bash) is the standard shell used in most Linux computers. It uses commands similar to a UNIX shell. Bash includes features such as:

- Command completion when pressing the tab key
- Command history
- Improved arithmetic functions

The Bourne shell is an earlier version of the Bash shell. It is similar in many ways.

Sh is the original shell created by Steve Bourne.

The Korn shell was developed by David Korn. Ksh has scripting features not found in bash.

The C-shell uses syntax similar to syntax used in the C programming language.

The tcsh shell is an improved version of csh. It offers command line editing and completion features not available with csh.

References

2.5.1 Environment Variables

2.5.2 Manage Environment Variables

2.5.3 Environment Variable Facts

14.1.1 Bash Scripting Overview

14.1.2 Bash Script Execution

14.2.1 Bash Shell Environments and Shell Variables

14.2.2 Bash Shell Parameters, User Variables and Expansions

14.2.3 Bash Shell Variables and Parameters

14.2.4 User Variables and Shell Arithmetic

14.2.5 Arrays and Expansions

14.2.6 Shell Environments, Bash Variables and Parameters Facts

q_shell_lp5_01.question.fex

Question 2: ✔ Correct

What would you enter at the command prompt to start a new Bourne-again shell (bash) session?

bash ✔

Explanation

The **bash** command opens a Bourne-again shell (bash) session. The Bourne-again shell (bash) is the standard shell used in most Linux computers. It uses commands similar to a UNIX shell. Bash includes features such as:

- Command completion when pressing the tab key
- Command history
- Improved arithmetic functions

References

2.5.1 Environment Variables

2.5.2 Manage Environment Variables

2.5.3 Environment Variable Facts

14.1.1 Bash Scripting Overview

14.1.2 Bash Script Execution

14.2.1 Bash Shell Environments and Shell Variables

14.2.2 Bash Shell Parameters, User Variables and Expansions

14.2.3 Bash Shell Variables and Parameters

14.2.4 User Variables and Shell Arithmetic

14.2.5 Arrays and Expansions

14.2.6 Shell Environments, Bash Variables and Parameters Facts

q_shell_lp5_02.question.fex

Question 3: ✔ Correct

A Linux user has an executable file named *ni* that can save a snapshot of network information with the date and time in a log file. The executable ni file is in the /root directory, and /root is the current working directory. Which of the following commands would run the executable file? (Select TWO).

- [ ] **source ni**
- [ ] **exec ni**
- [x] **./ni**
- [ ] **ni**
- [x] **/root/ni**

Explanation

To run an executable, you can either change to the directory where the script is held and type **./ni** or type the absolute path **/root/ni** to run the script from any directory.

Typing just the file name, **ni,** will not work because the current working directory, /root, is not typically contained in the PATH variable.

The **source ni** command is typically used within a shell script to read and execute commands within the ni file. In this case, the ni file would not be found, since the current working directory, /root, is typically not contained in the PATH variable.

The **exec ni** command is used to execute a command that completely replaces the current process. In this case, the ni file would not be found since the current working directory, /root, is typically not contained in the PATH variable.

References

14.1.2 Bash Script Execution

14.1.3 Executing and Sourcing a Script

14.1.4 Scripting Facts

q_shell_intro_lp5_01.question.fex

Question 4: ✔ Correct

Which of the following commands should a Linux user enter to see a list of all the commands the user recently ran at the command prompt?

- uname
- clear
- chsh
- **history**

Explanation

Use **history** to see all commands in the history queue. The **-c** option clears the history list. History command queues are separate for each user. A command typed as one user cannot be used after using the **su** command to switch to another user.

The **clear** command clears the shell screen, but does not clear the command history.

The **chsh** command changes the default shell. For example, **chsh -s /bin/ksh** changes the default shell for the user to the Korn shell if it is installed on the computer.

The **uname** command prints system information.

References

2.1.6 Linux Shell Command Facts

q_shell_intro_lp5_02.question.fex

Question 5: ✔ Correct

As a Linux user, you have access to an executable file named **myapp**. It's found in the current directory, but not in the command path. What would you enter at the command prompt to start the **myapp** file and replace the shell with **myapp** process?

exec ./myapp

Explanation

Use **exec ./myapp** to start the **myapp** executable file and replace the shell with **myapp** process. The **exec** command executes an executable not found in the command path. It also replaces the shell with the new process created by the executable file. **./** indicates that the executable is in the current directory.

References

14.1.2 Bash Script Execution

14.1.3 Executing and Sourcing a Script

14.1.4 Scripting Facts

q_shell_intro_lp5_03.question.fex

Question 6: ✔ Correct

Which of the following is displayed when the **uname -a** command is run?

- The names of files and directories in the current directory
- The current working directory
- ➡ All system information
- The current username

Explanation

The **uname -a** command displays all system information.

The **pwd** command displays the present working directory.

The **whoami** command displays the current username.

The **ls** command displays names of files and directories in the current directory.

References

2.1.6 Linux Shell Command Facts

q_shell_intro_lp5_04.question.fex

Question 7: ✔ Correct

What would you enter at the command prompt to display the current working directory?

pwd ✔

Explanation

Use the **pwd** command to show the current working directory.

References

- 2.8.1 Directory Navigation
- 2.8.2 Navigate Directories
- 2.8.3 Directory Management
- 2.8.4 Manage Directories
- 2.8.5 Directory Management Facts
- 2.9.2 File Management
- 2.9.4 Manage Files
- 2.9.5 File Management Facts
- 2.10.1 Links
- 2.10.2 Create Links
- 2.10.3 Link Facts
- 2.12.2 Finding Linux Commands
- 2.12.3 Finding Files
- 2.12.4 File Search Facts
- 2.12.5 Content Search Utilities
- 2.12.6 Find File Content
- 2.12.7 Content Search Facts

q_shell_intro_lp5_05.question.fex