Exam Report: 2.7.8 Practice Questions

Date: 2/19/2020 9:50:31 pm
Time Spent: 14:20

Candidate: Garsteck, Matthew
Login: mGarsteck

Overall Performance

Your Score: 38%

Passing Score: 80%

View results by: Objective Analysis Individual Responses

Individual Responses

Question 1: Incorrect

Which of the following commands sorts the combined contents of the wordlist1 and wordlist2 files and sends the results to both the screen and a file named sortedwordlist?

- **cat /usr/wordlist1 /usr/wordlist2 | sort | tee sortedwordlist**
- **cat /usr/wordlist1 >> /usr/wordlist2 | sort sortedwordlist**
- **cat /usr/wordlist1 /usr/wordlist2 | tee sortedwordlist**
- ~~**cat /usr/wordlist1 /usr/wordlist2 | sort sortedwordlist**~~

Explanation

The **cat /usr/wordlist1 /usr/wordlist2 | sort | tee sortedwordlist command** sorts the combined contents of the wordlist1 and wordlist2 files and sends the results to both the screen and a file named sortedwordlist.

The **cat /usr/wordlist1 /usr/wordlist2 | tee sortedwordlist** command does not perform a sort.

The **cat /usr/wordlist1 /usr/wordlist2 | sort sortedwordlist** command attempts to sort a file named **sortedwordlist** that, most likely, does not exist.

The **cat /usr/wordlist1 >> /usr/wordlist2 | sort sortedwordlist** command will append the contents of the wordlist1 file to the wordlist2 file and will attempt to sort a file named sortedwordlist that, most likely, does not exist.

References

Linux Pro - 2.7 Redirection, Piping and Command Substitution
[e_redir_lp5.exam.xml Q_REDIR_PIP_F_LP5_01]

Question 2: Correct

Which of the following describes the effects of the **ls -l /usr/bin >> /tmp/list.txt** command?

- The contents of the /usr/bin directory are written to a file named /tmp/list.txt. Previous file contents are kept, and the new information is added at the beginning of the file.
- The contents of the /usr/bin directory are written to a file named /tmp/list.txt. Previous file contents are be overwritten.
- The contents of the /usr/bin directory are written to a file named /tmp/list.txt. Previous file contents are kept, and the new information is added at the end of the file.
- The contents of the /usr/bin directory are written to both the screen and to a file named /tmp/list.txt. Previous file contents are overwritten.

Explanation

The>> operator redirects the output of a command to a file, appending the new information to the end of the file.

The file contents are not overwritten.

The new information is not added to the beginning of the file.

The contents of the /usr/bin directory are not written to the screen.

References

Linux Pro - 2.7 Redirection, Piping and Command Substitution
[e_redir_lp5.exam.xml Q_REDIR_PIP_F_LP5_02]

Question 3: Incorrect

Ann, a Linux script developer, is trying to debug a shell script that has the command **ls -s** in it. She suspects that an error is occurring, and she wants to send the results of the operation and any errors to a file named **~/Friday** in order to examine it later.

Which of the following commands should she use?

- **ls -s < ~/Friday**
- **ls -s &> ~/Friday**
- **ls -s > ~/Friday**
- ~~**ls -s >> ~/Friday**~~

Explanation

The **&>** operator redirects both successful output (1) and error results (2) to the same file. The **&> ~/Friday** construct is equivalent to **1> ~/Friday 2> &1** and **1> ~/Friday 2> ~/Friday**.

The > operator only redirects the output.

The>> operator only redirects the output and appends the new information to the **~/Friday** file.

The < operator passes the contents of the ~/Friday file to the **ls** command as input. In this case, the **ls** command will ignore the input.

References

Linux Pro - 2.7 Redirection, Piping and Command Substitution
[e_redir_lp5.exam.xml Q_REDIR_PIP_F_LP5_03]

Question 4: Incorrect

Which of the following commands gives the same results as **cat < turbo**?

- **cat turbo**
- **cat &> turbo**
- ~~**cat 1> turbo**~~
- **cat 2> turbo**

Explanation

Using the **turbo** file as an argument of the **cat** command causes the **turbo** file to be used as input. The < operator results in the same action. The contents of the **turbo** file are redirected to the **cat** command as input.

The **1> turbo** construct writes the output of the **cat** command to the **turbo** file.

The 2> **turbo** construct writes any errors generated by the **cat** command to the **turbo** file.

The &> **turbo** construct writes both the output of the **cat** command and any errors generated by the **cat** command to the **turbo** file.

References

Linux Pro - 2.7 Redirection, Piping and Command Substitution
[e_redir_lp5.exam.xml Q_REDIR_PIP_F_LP5_04]

Question 5: Incorrect

Which of the following commands sorts the contents of the **wordlist1** and **wordlist2** files and sends the result to standard output?

- ~~**cat /usr/wordlist1 > /usr/wordlist2 | sort**~~
- **cat /usr/wordlist1 | /usr/wordlist2 | sort**
- **cat /usr/wordlist1 >> /usr/wordlist2 | tee**
- **cat /usr/wordlist1 /usr/wordlist2 | sort**

Explanation

The pipe operator (|) in the **cat /usr/wordlist1 /usr/wordlist2 | sort** command sends the output from the cat command (a list of the contents of the wordlist1 and wordlist2 files) as input to the sort command. The sort command sorts the input and sends the result to standard output (usually the screen).

The **cat /usr/wordlist1 > /usr/wordlist2** construct overwrites the contents of the wordlist2 file with the contents of the wordflist1 file.

The **cat /usr/wordlist1 | /usr/wordlist2 | sort** command returns an error since the **/user/wordlist** file is not an executable file.

The **cat /usr/wordlist1 >> /usr/wordlist2 | tee** command appends the contents of the **wordlist1** file to the contents of the **wordflist2** file. The **tee** command requires a filename as an argument.

References

Linux Pro - 2.7 Redirection, Piping and Command Substitution
[e_redir_lp5.exam.xml Q_REDIR_PIP_F_LP5_05]

Question 6: Correct

Which command operator pipes the output of one command as the input of another command?

Explanation

The pipe (|) operator directs the output of one command into the input of another command.

References

Linux Pro - 2.7 Redirection, Piping and Command Substitution
[e_redir_lp5.exam.xml Q_REDIR_PIP_F_LP5_06]

Question 7: Incorrect

Which command reads from standard input (stdin) and writes to both standard output (stdout) and a file?

tee

Explanation

The **tee** command reads from standard input (stdin) and writes to both standard output (stdout) and a file.

References

Linux Pro - 2.7 Redirection, Piping and Command Substitution
[e_redir_lp5.exam.xml Q_REDIR_PIP_F_LP5_07]

Question 8: Correct

Joe, a bash script developer, is trying to debug a shell script named **myscript**. Which of the following commands would record the output of the script in a text file?

- **myscript | echo | testfile.txt**
- **myscript | testfile.txt**
- **echo myscript >> testfile.txt**
- **myscript >> testfile.txt**

Explanation

The **myscript >> testfile.txt** command uses the >> operator to redirect the output of the executed myscript script to a file named testfile.txt.

The **echo myscript >> testfile.txt** command appends the contents of the myscript file to the testfile.txt file.

The **myscript | testfile.txt** command pipes the output of a command as the input of another command (or shell script). In this case, testfile.txt is not likely to be an executable shell script.

The **echo** command within the **myscript | echo | testfile.txt** command does not echo the output form the myscript script because it does not read from stdin. Additionally, testfile.txt is not likely to be an executable shell script.

References

Linux Pro - 2.7 Redirection, Piping and Command Substitution
[e_redir_lp5.exam.xml Q_REDIR_PIP_F_LP5_08]

Question 9: Incorrect

Tom, a shell script developer, knows that the **date +%A** command gives the current local weekday name (Sunday, Monday, Tuesday, ...).

Which of the following commands will copy the authentication log file, **/var/log/auth.log**, to a new file that with a filename that includes the current local weekday name?

- ~~**xargs /var/log/auth.log ~/auth$(date +%A).log**~~
- **cp /var/log/auth.log ~/auth${date +%A}.log**
- **cp /var/log/auth.log ~/auth$(date +%A).log**
- **tee /var/log/auth.log > date +%A -log**

Explanation

The **$(date +%A)** operator performs a command substitution so that the new filename will include the current local weekday name.

The **${date +%A}.log** operator attempts to substitute the contents of a variable.

The **tee** command is used to write output to both stdout and to a file.

The **xargs** command is used to divide large amounts of streamed output into smaller chunks to be used as arguments for another command.

References

Linux Pro - 2.7 Redirection, Piping and Command Substitution
[e_redir_lp5.exam.xml Q_XARGS_F_LP5_01]

Question 10: Correct

Which command overcomes the 128 KB shell command size restriction by breaking up long lists of arguments?

Explanation

The **xargs** command reads items from the standard input and breaks up long lists of arguments into smaller, usable chunks.

References

Linux Pro - 2.7 Redirection, Piping and Command Substitution
[e_redir_lp5.exam.xml Q_XARGS_F_LP5_02]

Question 11: Incorrect

Which of the following commands utilize command substitution? (Choose TWO.)

- [x] ~~**echo -e "sort -r names.txt"**~~
- [ ] **myvar=`sort -r names.txt`**
- [x] **myvar=$(sort -r names.txt)**
- [ ] **myvar="echo 'sort -r names.txt'"**

Explanation

myvar=`sort -r names.txt` and **myvar=$(sort -r names.txt)** both provide command substitution. Enclosing a command within backticks (`) performs command substitution in the same way as the $() operator.

echo -e "sort -r names.txt" echoes the command to stdout.

myvar="echo 'sort -r names.txt'" stores the following in the myvar variable: **echo -e *text stored in names.txt***

References

Linux Pro - 2.7 Redirection, Piping and Command Substitution
[e_redir_lp5.exam.xml Q_XARGS_F_LP5_BACKTICK]

Question 12: Incorrect

Which of the following commands stores the output of the **ls -a** command in a shell variable named allfiles?

- ( ) **allfiles=${ls -a}**
- ( ) **allfiles=$(ls -a)**
- ( ) **allfiles=$((ls -a))**
- (•) ~~**allfiles="exec ls -a"**~~
- ( ) **allfiles="ls -a"**

Explanation

allfiles=$(ls -a) stores the output from the**ls -a** command in the shell variable allfiles. This is command substitution.

allfiles=$((ls -a)) stores the value of zero in the variable allfiles.

allfiles="exec ls -a" stores the*ls -a* string in the variable allfiles.

allfiles="ls -a" stores the *ls -a* string in the variable allfiles.

allfiles=${ls -a} utilizes command expansion and does not produce a command output.

References

Linux Pro - 2.7 Redirection, Piping and Command Substitution
[e_redir_lp5.exam.xml Q_XARGS_F_LP5_COMMAND_SUB]