Homework Week #7

(20 total points)

Using what you’ve learned so far in this course, specifically week 7, answer the following:

Question 1. (3 points) List three or more benefits of **getopt/getopts** verses traditional parsing?

1. Don’t need to pass the positional parameters through an external program.
2. Can set shell variables to use for parsing.
3. There is no need to argue with several getopt implementations.

Read the following shell script except and answer questions 2 and 3:

Line 1. read choice

Line 2. case $choice in

|  |  |
| --- | --- |
| Line 3. | 1) who;; |
| Line 4. | 2) uname -v;; |
| Line 5. | 3) ls -lS;; |
| Line 6. | 4) lscpu | head;; |
| Line 7. | 5) date;; |
| Line 8. | 6) banner "Goodbye!";; |

Line 9. \*) echo "Invalid input";;

Line 10. esac

Question 2. (2 points) Explain, in your own words, the overall function of the script except (not a line by line explanation).

This program displays system information, specifically for the CPU, based off of a user’s input.

Question 3. (2 points) Explain the output of Lines 2 and 4.

**Line 2:** Print out the kernel version of the system.

**Line 4:** Lists the CPU information.

Question 4. (3 points) The following are special variables called position parameters: (Choose all that apply.)

a. Y

1. **$1**
2. **$0**
3. $X
4. **$#**

Question 5. (3 points) List three common read command options and explain their function.

1. **-p** prints out the text that immediately follows the -p.
2. **-l** lists all the current read line commands.
3. **-s** causes the users typing to not be displayed.

Question 6. (2 points) How can data from STDIN be sent to both a file and STDOUT?

Using **tee** command can do this. This will send data from the STDIN to the destinations.

Read the following shell script and answer questions 7 and 8:

$cat script1 #!/bin/bash exec 3<$1 exec 4<$2 exec 5>$3 while read line <&3 do

echo “$line” >&5

done

while read line <&4 do

echo “$line” >&5

done

Question 7. (3 points) The script requires 3 filenames as arguments, e.g., **./script1 file1 file2 file3.** Which files are input files and which are output files?

**files1** and **file2** are input files and **file3** is an output file.

Question 8. (2 points) Inside the while-loop, why does variable “**$line**” have to be enclosed by double quotes?

This variable needs to be in double quotes because the content of the variable have to be a string. This is pretty much typecasting to a string.