Michael Flanagan

CIS 140U Lab 8

Chemeketa CC Online

Summer 2017

**2. Variable Questions**

**1. Determine and use a command that will store the number 12 into the variable num. Write the command in your assignment, and explain the importance of the command.**

num=12

**2. What command will allow you to check the value of the variable num? Write the command in your assignment.**

echo $num

**3. Start a subshell by typing sh. What is the current value for the variable you just created? Why is that so and what can be done to remedy this? Write down the command, along with your answer as to why the variable did not show in the subshell.**

The current value doesn’t exist, in the subshell, so we see a blank space. It is currently only defined locally in home/mflanagan. To make it a global variable which will be seen across other shells we can simply export it. Since I’m currently in the subshell I have to get back to the shell I defined it in in I can get there by typing **exit**. Than once back in home/mflanagan I can type.

**export num**

than if I type

**sh**

**echo $num**

I can see that **num** is set to 12.

3. Executing a Script

**4. For a script file to run in a Linux shell, what permissions must the file have? Write down the command and options needed in your assignment.**

**Read and execute**

**chmod 755 scriptname**

**5. Once a script has the permissions set, what are the three separate ways that would allow you to run that script? Which way allows you to debug your script?**

move it to /usr/local/bin (as *root*, of course), to make the script available to yourself and all other users as a systemwide executable.

You could navigate to the directory of the script and type

**./scriptname**

You could also invoke it by using **sh scriptname**, however this isn’t recommended because using this command will effectively disable stdin

4. Scripting – **Develop scripts using the following problem sets. Be sure to use comments in your scripts. It may be helpful to first write out the step by step process that you need to solve the problem BEFORE attempting to write the scripts. There are examples in the PowerPoint lecture.**

**6. Create a script that asks the user for a file and then appends the date to the file the user has entered. Let the user know if the file can or cannot be written too. Also, if the file does not exist create the file and write the date in the newly created file. Lastly display to the user the file, including a statement letting them know that the file has changed. (Hint: remember the – f and – w when evaluating a file in an if statement)**

#!/bin/bash

echo "Please input the filename of the file you would like to append the date to: "

read fileName #getting user input

if [ -f $fileName ] #checking if input is a file and if the file exists

then #fileName is a file and exists

echo "The file $fileName can be written to. " #prints to console the file can be written to

else #since we got this far. This means that file didn't exist so we will create one with the name that user inputed

echo "The file $fileName does not exist. Creating an empty file and calling it $fileName" #prints to console the file DNE so we will create one and append the date in it

fi

""date"" >> $fileName #this line appends the date to the end of the file

echo "The file $fileName has been modified to include the date appended at the bottom" #prints to console the file has been modified

echo "The contents of $fileName are displayed below"

cat $fileName

**7. Create a script that takes two numbers as input from the user, and then asks the user if they would like to add or subtract the two numbers (use a case statement). Then print out the result in a friendly message to the screen. Be sure to let the user know if they choose an option that was not on the menu. (Hint: when evaluating the inputted values from the user, they must be represented as numerical data types. )**

#!/bin/bash

echo "Please input number 1"

read num1 #getting number 1

echo "Please input number 2"

read num2 #getting number 2

echo "Would you like to Add or Subtract these numbers?"

echo "Type a to add"

echo "Type s to subtract"

read option

case $option in

[a])

number=`expr $num1 + $num2`

echo "$num1 + $num2 is $number"

;;

[s])

number=`expr $num1 - $num2`

echo "$num1 - $num2 is $number";

;;

\*)echo "You chose an option that is not on the menu. Please run

the script again to get your answer. ";;

esac

**Now modify your case statement so the user can use either an upper or lower case letter in the menu. That is if the user types a or A both will allow the case statement to run the add selection. (Hint: use | as the or operator)**

#!/bin/bash

echo "Please input number 1"

read num1 #getting number 1

echo "Please input number 2"

read num2 #getting number 2

echo "Would you like to Add or Subtract these numbers?"

echo "Type a to add"

echo "Type s to subtract"

read option

case $option in

[aA])

number=`expr $num1 + $num2`

echo "$num1 + $num2 is $number"

;;

[sS])

number=`expr $num1 - $num2`

echo "$num1 - $num2 is $number";

;;

\*)echo "You chose an option that is not on the menu. Please run

the script again to get your answer. ";;

esac