CIT 371 Lab 12: shell scripting

This lab can be done with SSH/PuTTY or with the Web Console. See the Student VM Access document for information on accessing your VMs.

Log into Coivcenter, start your VM and log in as Student. cd to your home directory and create a subdirectory called scripts. cd to that directory. You can use gedit or some other editor but you need to get used to vi, so I strongly recommend that you use vi. Remember before running any new script that you will need to modify its permissions to be executable. Use either 745 or 755.

1. Create a script called script1 with the following. Match the syntax precisely. Pay close attention to the use of the back tick marks on line 4. This is the character right above the tab key on your keyboard.

**#!/bin/bash**

**echo You are $USER**

**echo Your home directory is $HOME**

**echo Your home directory consists of `du -sH ~`**

Run the script by typing **./script1 <enter>**. Alter your script so that the output of each echo statement is redirected to the file info.txt. *How did you do this?*

**You can do this by directing the commands to the .txt file such as >>file.txt . This looked like**

**#!/bin/bash**

**echo You are $USER >>info.txt**

**echo Your home directory is $HOME >>info.txt**

**echo Your home directory consists of `du -sH ~` >>info.txt**

1. Create the following script, named script2 and run it.

**#!/bin/bash**

**echo What is your name?**

**read NAME**

**echo What is your username?**

**read USERNAME**

**echo Hello $NAME, your home directory contents and size:**

Complete this script by adding instructions to output this user’s home directory contents (using ls) assuming that the user’s home directory will be /home/$USERNAME as well as using the du statement from script1 to output the disk usage of the user’s home directory. When done, *add this script to your answer file*.

**#!/bin/bash**

**echo What is your name**

**read NAME**

**echo What is your username**

**read USERNAME**

**echo Your home directory consist of `du -sH /home/$USERNAME`**

**ls /home/$USERNAME**

1. Rewrite the script from #2 so that the NAME and USERNAME are supplied to the script as parameters, removing the first four instructions (the first two echo and the two read statements). Save and test your script. *When done, add this script to your answer file.*

**#!/bin/bash  
NAME=$1  
USERNAME=$2  
echo Your home directory consist of `du -sH /home/$USERNAME`  
ls /home/$USERNAME**

1. Write the following script, calling the file script4.

**#!/bin/bash**

**if [ $# -ne 2 ]; then echo Illegal input**

**elif [ $1 –gt $2 ]; then echo $1 is greater**

**else echo $2 is greater**

**fi**

Save and run the script providing it no parameters, two parameters of 5 and 10, two parameters of 10 and 5, and two parameters of 5 and 5. *What does [ $# -ne 2 ] mean? Do we need this in the script?*

**[ $# -ne 2 ] means we are saying the number of parameters that can pass through is 2. Yes, we need this because it allows us to check the parameters that are passing by which should only be 2 in this case.**

1. Modify the script from #4 so that it will also output a message if the two parameters are the same. For instance, it might output “5 equals 5”. *When done, include the modified script as the answer to this step.*

**#!/bin/bash  
if [ $# -ne 2 ]; then echo Illegal input  
elif [ $1 -gt $2 ]; then echo $1 is greater  
elif [ $2 -gt $1 ]; then echo $2 is greater  
else echo $1 equals $2  
fi**

1. Write the following script calling it script6.

**#!/bin/bash**

**read –p “Enter the number you seek ” NUM**

**for VALUE in $@; do**

**if [ $VALUE –eq $NUM ]; then COUNT=$((COUNT+1)); fi**

**done**

**echo $NUM appeared $COUNT times**

Run script6 passing it the list of numbers 5 10 6 12 5 18 10 4 19 21 5 12 18 22 and when prompted, input 5. Rerun the script inputting 18 instead. Rerun the script inputting 23 instead. *What outputs did you get for each input?*

**The outputs that I got were “ 5 appeared 3 times”, “18 appeared 2 times”, and “23 appeared times “** .

1. Rewrite the script from #6 to input two inputs from the user, instead of just NUM, and count the number of parameters that are equal to or in between the two. For instance, if the user inputs 12 and 20, we would have 5 matches (12, 18, 19, 12, 18). Assume the first input is less than the second (that is, you do not need to worry about the user inputting 20 and then 12). *Add this revised script to your answers file.*

**NUM1=10;  
NUM2=20;  
eval "t=({$NUM1..$NUM2})";  
printf -v tt "%d " ${t[\*]}; t=($tt)  
echo Array Count: ${#t[\*]}  
echo ${t[\*]}**

1. Write a script which receives a list of parameters and iterates through them using a for loop similar to what you wrote in #6. In this case, the for loop requires two if statements (or an if-then-else) to locate the smallest and the largest values of the inputs. Output both values. Assume the list of parameters contains at least one number. As an example, if the list of parameters was 5 10 6 12 5 18 10 4 19 21 5 12 18 22, it will output that 4 is the smallest and 22 is the largest. When done, *add this script to your answer file.*

**#!/bin/bash**

**read –p “Enter the number you seek ” NUM**

**max=$1**

**min=$2**

**for arg in "$@"**

**do**

**if [ "$arg" -gt "$max" ]**

**then**

**max=$arg**

**fi**

**for arg in "$@"**

**do**

**if [ "$arg" -lt "$min" ]**

**then**

**min=$arg**

**fi**

**done**

**echo "Largest value is: $max"**

**echo "Smallest value is: $min"**

Shut down your VM if desired, disconnect from the VPN if you are using it, and submit your lab report.