//Mohammad hlhwly  
//1191413

//Lab 9

Linux Commands General Format

1)Create a file for writing with comments inside.

2)Start writing comments

**Lab11. Shell Scripts (III)- Programming**

**( Looping Constructs)**

**Objectives**

After completing this lab, the student should be able to:

* Include programming looping constructs in shell scripts.
* Understand and use the while, until, and for loops constructs.
* Learn how to make for loops more efficient by using command outputs as lists.

**Shell Script Loops**

In order to create useful scripts that can automate real jobs, we need to learn how to include loops in those scripts. There are different loop constructs that may be used in shell scripts which include:

***while loops***

***until loops***

***for loops***

Each has its own useful features that make it useful in certain situations.

**While Loop**

Let us first start with the while loop. The structure of the while loop is as follows:

***while*** ***condition***

***do***

***statement(s)***

***done***

example:

***vi listarguments***

***while [ $# -ne 0 ]***

***do***

***echo $1***

***shift***

***done***

***:wq***

Run the above script as follows:

***listarguments a hello 7 x***

***Check the output.***

***Output:***

***[a]***

***[hello]***

***[7]***

***[x]***

***Note: Rules that apply to conditions used in selection statements are exactly the same as those that apply to conditions in loop statements.***

After making sure you understand the above example do the following:

60

Computer Science Department ( Linux OS Laboratory Manual COMP311 )

Rewrite the delete script we wrote in the last lab such that it works as follows:

***delete file1 wrong dir1 file2***

***File file1 is deleted***

***wrong: No such file or directory***

***Directory dir1 is deleted***

***File file2 is deleted***

***Answer:***

***vi*** ***delete***

if [ $# -eq 0 ]

then

echo usago : yoy must pass at leat one file or directory

else

while [ $# -ne 0 ]

do

if [ -f $1 ]

then

rm $1

echo file [$1]

elif [ -d $1 ]

then

rm -r $1

echo directory [$1] deleted

else echo no file or directory name [$1]

fi

shift

done

fi

***:wq***

Now try it with existing file and directory names as arguments. ***Does it work? \_\_\_\_\_\_\_.***

Sometimes the loop will stop executing based on the user input, as follows:

***vi findahmad***

***echo Enter name***

***read name***

***while [ “$name” != “ahmad”*** ***]***

***do***

***echo $name: wrong name. Try again.***

***echo Enter name***

***read name***

***done***

***:wq***

Now modify the ***checkusername*** script from the previous lab such that it is called ***checkusernames*** instead and works as follows:

61

Computer Science Department ( Linux OS Laboratory Manual COMP311 )

***checkusernames***

***Enter user name to check or word “enough” to stop u1112345***

***Enter user name to check or word “enough” to stop u11***

***Enter user name to check or word “enough” to stop u1123456***

***Enter user name to check or word “enough” to stop enough***

***u1112345 = Salem Hamdi***

***u11 = No such user name***

***u1123456 = Sabah Khaled***

***Answer:***

***vi checkusernames***

echo Enter use name to chec or word

read name

while [ "$name" != "enou" ]

do

output = $(cat prj\_passwd | grep "^$name:" | cut -d : -f5 | sed "s/ / /g")

if [ -z "soutput" ]

then

echo $name =no such usename else

echo $name = $output

fi

echo Enter use name to chec or word

read name

done

:wq

**Break and Continue Statements**

The programmer can use ***break*** and ***continue*** statements inside shell script loops which

mean the same as they do in the C language:

***break*** - exit the loop immediately.

***continue*** –stop running the current cycle but go back and check the condition.

In addition they can use ***break*** and ***continu***e followed by a number to specify how many loop levels they want them to work for. For example:

***break 2***

Will exit out of two nested loops if they exist.

62

Computer Science Department ( Linux OS Laboratory Manual COMP311 )

**until loop**

The until loop is similar to the while loop, but stops when the condition becomes true.

until false

do

statements

done

***Modify the above two programs such that they use the until construct instead of the while construct and try them out. Did they work? \_\_\_\_\_\_\_\_\_\_\_\_\_\_.***

**For loop**

In shell scripts, the for loop is very powerful and useful. The general structure of the for loop is as follows:

***for item in list of items***

***do***

***statement(s)***

***done***

What makes a for loop powerful is the different ways a list of items may be specified.

Let us start with a simple example:

***vi names***

***for name in ahmad hamdan subha khaled***

***do***

***echo $name***

***done***

***:wq***

Run the script names. It should display the names given in the list.

Now change the first line in script names to the following:

***for name in $\**** ( remember that $\* holds all the arguments as a list)

and run the modified script as follows:

***names ahmad subha khaled***

***What happened? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.***

***Rewrite the delete script we wrote at the beginning of this lab such that it uses a for loop instead of a while loop. Did it work? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.***

The best feature about the for loop is that we can treat the output of a command as a list of items as follows:

63

Computer Science Department ( Linux OS Laboratory Manual COMP311 )

***vi*** ***lines***

***for line in $(cat /etc/passwd)***

***do***

***echo $line***

***done***

Using a for loop, write a script called ***comp311*** that lists the full names of all the users

that are registered in the comp311 course.

Answer:

for list in $( cat prj\_passwd | cut -d: -f5 | sed "s/\_/ /")

do

echo $(echo "$list" | tr '\_' ' ')

done

Now rewrite the script ***comp311*** such that it will display only the names of the users that are currently logged in to the system. (hint: use the output of the who command) Answer:

The for loop can also be applied to a directory of files as follows:

1. ***myfiles for file in \* do***

***echo $file***

***done***

64

Computer Science Department ( Linux OS Laboratory Manual COMP311 )

Write a script called ***filetypes*** that uses a for loop to type the name and type ( file, dir, or unknown) for each file in a given directory as follows: Assume that I use the script in the following way:

filetypes /etc

then the script should display the names of all the files under directory /etc and the type of each of those files:

Answer:

for file in $1/\*

do

if [ -f $file ]

then

echo [ $file ] is a file

elif [ -d $file ]

then

echo [$file] is a directory

else

echo [$file] is unknow

fi

done

The which command displays the directory in the PATH that contains the command. Try it as follows:

***which ls***

***What is the result? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ .***

Write a script called ***mywhich*** that simulates the which command. You are not allowed to use the which command in your script. (***hint: use the for loop and the sed command***).

**Answer:**

for searchDir in $( echo $PATH | sed "s/:/ /g")

do

if [ -f $searchDir/$1 ]

then echo $searchDir

fi

done

65

3) Exit to save all lab workers