COMP311
Linux OS Laboratory
Lab9:Shell Scripts (I) – Introduction

By

Alaa’ Omar

Objectives

1

Create and execute simple shell scripts.

2

Use positional parameters and shifting to pass command line arguments to scripts.

How To Write A Shell Script

1) **Write a script**. Shell scripts are ordinary text files. In this lab we'll use vi editor.
2) **Make the script executable**. We need to set the script file's permissions to allow execution.
3) **Put the script somewhere the shell can find it**. The shell automatically searches certain directories for executable files when no explicit pathname is specified.

Writing your first script (steps)

1. Using the vi editor open a new file and write your script as follows:

```
vi myfirst
        echo this is my first Linux script
        echo I like it
        echo bye
:wq (save and quit)
```

2. Add execute permission to the file myfirst

3. You must make sure that you have the following line added to the end of your environment setup file (.bash_profile):
PATH=$PATH:.

4. **Run the script $myfirst**

What is the output of the script ?

Read from terminal

Follow the steps:

(1)

vi greetings

```
        echo What is your name
        read name
        echo hello $name
```

:wq

(2)

chmod +x greetings

(3)

$greetings

What do you think is the purpose of the read command?

Read from a terminal

What is the command used to delete a file from a system?
We want the user to enter a file name

(1)

```
vi delete
        echo Enter file name:
        read filename
        rm $filename
        echo File $filename has been deleted
:wq
```

(2)

chmod +x *delete*

(3)

$ *delete*

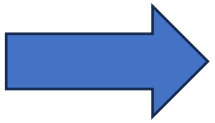
Writing scripting (practice)

Write a script called copy that asks the user to enter a source filename and a destination filename and then copies the source to the destination. Your script should work as follows

Copy (name of the script)
Enter source file name:
one (the first file name)
Enter destination file name:
two (the second file name)
File one is copied to file two

Parameters

```
vi params
    echo $1
    echo $3 $2
    echo $#
    echo $0
    echo $5
    echo $*
:wq
```



```
one
3 two
7
./params
5
One two 3 four 5 6 bye
```

$1	The first parameter
$2	The second parameter
$3	The third parameter
$#	Number of parameters
$0	The script name
$*	All parameters

$params one two 3 four 5 6 bye

Parameters (Practice)

rewrite both the delete and copy scripts above to run as follows:

delete thefile
thefile has been deleted

```
Vi delete
rm $1
echo $1 has been deleted
```

copy file1 file2
File file1 has been copied to file2
Answer:

```
Vi copy
cp $1 $2
echo File $1 has been copied to $2
```

Parameters (Practice)

Practice:
Write a script called whoisuser that takes the login name of a user as a parameter and then uses the /etc/passwd file to get and print the full name of that user as follows:
whoisuser u1122334
u1122334 = Ahmad Hamdan
hint: use variable and command substitution

```
Vi whoisuser

F_name= $(grep $1 /etc/passwd | cut –d: -f5 | tr ‘ _’ ‘ ‘ )

echo $F_name
```

Shifting parameters

To shift script command line parameters to the left, we use the shift command as follows:

- **shift** number of shifts (e.g., shift 2 for 2 shifts)
- **shift** (no number shifts one)

To understand how shift works, let us rewrite and run the params script above as follows:

vi params

```
echo $1
shift 2
echo $2 $3
echo $#
shift
echo $0
shift 3
echo $1
echo $*
```

:wq

```
one
5 4
9
./params
seven
seven 8 9 ten bye
```

$params one two three 4 5 6 seven 8 9 ten bye

Shifting parameters

```
alaa@Ubuntu: ~/Desktop
echo "$1 $2 $3 $4 $5"
shift 2
echo "$1 $2 $3 $4 $5"
shift 2
echo "$1 $2 $3 $4 $5"
shift
echo "$1 $2 $3 $4 $5"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"shiftingparams" 7L, 111B
```

Let's write the following script called shifting parameters

```
alaa@Ubuntu:~/Desktop$ ./shiftingparams one two three four five
one two three four five
two three four five
three four five
```

Shifting parameters

Comments

You can add comments to your scripts by using the # sign followed by the comment anywhere in your script. Lines that start with (#) are interpreted as comments except in one case where shells have (#!) followed by the name of a shell as the first line of a script. In that case that line is interpreted as the name of the shell to be used for executing that script.

Example:
If your script starts with the line:
#!/bin/bash

Then the script is meant to be executed using the /bin/bash shell.
Check out the following system scripts:
more /etc/rc.sysinit
more /etc/rc.local
What is the first line in those files (scripts)?

Practice

Write a script called stat, that takes pathname as parameters and print some statistics about the file:

Example: $ stat /etc/passwd
output
File /etc/passwd has 10 line, 50 words
The file owner is root
The file permission is –rw-r—r--

The End