<Brian W. Lee>

CSC 2510: DevOps

Lab 06 – Advanced Scripting

General Instructions

Using your book and previous lecture material, fill out this assignment sheet. **Use red text to signify your answers.** Use information from your textbook and the provided lectures to answer the questions on this lab. You may utilize online resources to answer these questions if you cite them.

Submission Instructions

To submit, **change the name in the header** and save this document as a PDF. Attach your PDF document and any scripts written to the iLearn dropbox.

Lab Questions

For this lab you will write a shell script called `nethelper.sh` that does the following:

I. Reads in a list of hosts from a host file that is passed in as a parameter.
II. Repeatedly, until the user selects quit (`q`), ask the user if the user wants to perform an action from the table below. If the user selects an option that requires a host name, query the user for which host to use. Users should be able to use either uppercase or lowercase to select an option.
III. Once the user has selected an action, prompts the user for which host and then applies the action, outputting the results. For your first iteration, only implement options `N, P, H, I,` and `Q`.
IV. Add an option for `ssh` (option `S`).

Letter Option	Action
N	`nslookup $host`
P	`ping -c 1 $host`
S	`ssh $user@$host`
H	`hostname`
I	`ifconfig -a`
Q	`quit`

<Brian W. Lee>

Example output:

```
$ ./nethelper.sh hosts.txt
(P) for ping
(N) for nslookup
(Q) for quit
Select one of the above:  p
1) localhost
2) www.google.com
Enter a number to select a host: 2
ping -c 1 www.google.com
PING www.google.com (74.125.126.147) 56(84) bytes of data.
64 bytes from ik-in-f147.1e100.net (74.125.126.147): icmp_seq=1 ttl=52
time=1.72 ms

--- www.google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.722/1.722/1.722/0.000 ms
(P) for ping
(N) for nslookup
(Q) for quit
Select one of the above:  n
1) localhost
2) www.google.com
Enter a number to select a host: 2
nslookup www.google.com
Server:         172.17.0.1
Address:        172.17.0.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 74.125.124.103
Name:   www.google.com
Address: 74.125.124.105
Name:   www.google.com
Address: 74.125.124.147
Name:   www.google.com
Address: 74.125.124.104
Name:   www.google.com
Address: 74.125.124.99
Name:   www.google.com
Address: 74.125.124.106
(P) for ping
(N) for nslookup
Select one of the above:  q
$
```

<Brian W. Lee>

I. Read in Host File

Write a function called `read_hosts`. Functions in bash scripts have the following syntax:

```
function name {
 # function body goes here
}
```

You can put any bash syntax inside the function as the function body. Also, note that the parameters to a function act like the parameters passed in at the command line. In other words, the parameters to the function are place in variable named `$1`, `$2`, and so on.

So, how do you read the hosts from a file? The file will be formatted as one host per line. Therefore, you can simply use cat to print the file, capture the output, and the iterator over the output using a for loop. However, you will have to put the hosts in an array called `hosts_array`. Bash arrays are simple. You simply use the brackets to indicate an index. So, the algorithm for your function will look like the following:

1. Set variable hosts to the results of calling cat on `$1` (the name of the file passed as a parameter)
2. Initialize a `count` variable to 1 (`count` will be the index of the array)
3. For each `host` in hosts do the following:
    a. Set the `hosts_array` at the index to the `host`
    b. Increment `count` by `1`

To call the function, your script will simply execute the following code:

```
read_hosts $@
```

The `$@` passes the script parameter (the name of the file containing the hosts) to the function. In the above algorithm, `hosts_array` is a global variable, so the function's caller will be able to use the global variable to access the host names.

II. Selecting an Option

To implement this, you will need a `while` loop. In the body of the loop, print the menu, then read the user's response. Next, use a `case` statement to determine which menu item the user chose. The case statement should execute the correct command based on the user's choice. If the user inputs an invalid option, tell the user and re-prompt the user. The algorithm is as follows:

1. While not done
    a. Print the menu.
    b. Prompt for user response.
    c. Read response into variable `cmd`.
    d. Switch-case over the user input such that each option prompts the user for the necessary options and outputs the commands output.

<Brian W. Lee>

III. Executing the Selected Option

The options and their required actions are listed on the table on page 1 of this document. If the command requires a host to be selected, print the list of hosts read in from the `hosts.txt` file and prompt the user for a selection. To do this, write a function called `pick_host` that manages this. A host should be selected using a number. Any size of number should be accepted. You should also validate that the user inputted a valid option (e.g. is a number that is within the valid range). If the number is invalid, re-prompt the user for a valid host option. Store the valid host value in a variable called `which_host`. The algorithm is as follows:

1. set variable `count` to 1
2. for each host in `$1` (where `$1` is the list of host passed it - not the array)
    a. then `echo "$count $host"`
    b. add 1 to `count`
3. prompt the user to enter a number to select a host
4. read the user's response into the `which_host` variable
5. make sure that `$which_host` is greater than or equal to 1 but less than `$count` (it is a valid host in the array). If it is not, re-prompt the user for valid input.

IV. Adding an SSH Option

Add option `S` to your script to `ssh` to `shell.csc.tntech.edu` The user should be prompted for which server to use and what username should be used to login. Note that a usernames and passwords are the same as those used for other university resources such as your email and iLearn.

Additional Questions

1. (1) What is `ssh`? A program for logging into a remote machine and for executing commands on a remote machine.

2. (5) What do each of the commands used in this program do? (nslookup $host) look up information for host using the current default server or using server, if specified. If host is an Internet address and the query type is A or PTR, the name of the host is returned. If host is a name and does not have a trailing period, the search list is used to qualify the name. (ping –c 1 $host) Ping uses ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway which –c 1 $host means to stop after sending count (1) of ECHO_REQUEST packets to a specific host. (ssh $user@$host) Connects to the target host by first making a ssh connection to the jump host and then establishing a TCP forwarding to the ultimate destination from there. (hostname) show or set the system's host name. (ifconfig –a) ifconfig cofigures a network interface which –a displays all interfaces which are currently available, even if it is down.
3. (14) `nethelper.sh`