Exercise12 **– Writing Shell Scripts**

1. **Review questions**
2. Which is the purpose of the ‘hash-bang’ (**#**!**/bin/bash**) header ?
3. How can you find out if the last command succeeded or not?
4. Why would we use the **if-then-fi** construct ?
5. How can you pass information from the command line into a script?
6. **Writing a simple shell script**

In this exercise you will create a simple script, called **chkuser**.

The script will accept one argument on the command line (a valid user login name) and will print a line indicating if the user is logged in or not.

**The specification for the script is as follows:**

- The script will be executed as:

$ **echo $(whoami)**

1. First of all, create a directory called **tools** in your home directory. We will use this directory to store the script we are about to write. Move into the **tools** directory for the remainder of this exercise.
2. Using an editor of your choice (for example **vi**), write a script matching the specification above.
3. Give your script the execute permission, then run it.   
     
   On this occasion, provide the full path to your script on the command line (your **PATH** variable has not been yet adjusted to include the **tools** directory).
4. Scripts are programs on disk. In order to be able to find your scripts, the shell needs to know where you stored them. In the previous example we told the shell where to go to find the script.

Most of the time, however, we want to be able to specify the script name, without having to type it’s path.

For that reason, adjust the **PATH** variable, to include your new **tools** directory. Make your change permanent, so that you can log out and back in again and still run your script.

#!/bin/bash

echo “Please input a user name:”

read username

if [ $username == “$(whoami)” ]; then

echo “User $username is logged in."

else

echo “User $username is not logged in.”

fi

**If time allows**

This script will test your knowledge of previous bash commands combined within a script.

1. Create a script that automates the following:

* Creation of a new directory, and the creation of 2 text files within this directory
* Move one of the files within the directory to your home directory
* Change the type of the second text file to a script
* Add commands to this file to print the working directory and the current user
* Execute the second file

*Hint: Use mkdir, mv, pwd, apt-get, chmod and dot slash notation*