Practical 6: Working with the BASH Shell and Shell Scripts

1. Which of the following files is always executed immediately after a user logs in to a Linux system and receives a BASH shell?

   a. /etc/profile b. ~/.bash_profile
   c. ~/.bash_login d. ~/.profile

   Answer: a

2. Which of the following will display the message **welcome home** if the **cd /home/user1** command is successfully executed?

   a. **cd /home/user1 && echo "welcome home"**
   b. **cat "welcome home" || cd /home/user1**
   c. **cd /home/user1 || cat "welcome home"**
   d. **echo "welcome home" && cd /home/user1**

   Answer: a

3. What would be the effect of using the **alias** command to make an alias for the **date** command named **cat** in honor of your favorite pet?

   a. It cannot be done as there already is an environment variable **cat** associated with the **cat** command.
   b. It cannot be done as there already is a command **cat** on the system.
   c. When you use the **cat** command at the command prompt with the intention of viewing a text file, the date appears instead.
   d. There is no effect until the alias is imported as it is a user-declared variable.

   Answer: c

4. Consider the following shell script:

```
echo -e "What is your favorite color?--> \c"
read REPLY
if [ "$REPLY" = "red"  –o  "$REPLY" = "blue" ]
then
  echo "The answer is red or blue."
else
  echo "The answer is not red nor blue."
fi
```

   What would be displayed if a user executes this program and answered **Blue** when prompted?

   a. The answer is red or blue.
   b. The answer is not red nor blue.
   c. The code would cause an error.
   d. The answer is red or blue. The answer is not red nor blue.

   Answer: b

5. Using sort as a filter, rewrite the following sequence of commands:

   sort list > temp
   lpr temp
   rm temp

   cat list | sort | lpr

6. What would happen if the user executed the following commands? Explain the output.

   (i) **file /usr/bin/* | grep "Again shell script" | sort -r**

   Find all the files under the /user/bin directory that are Bourne-Again shell scripts and display them in reversed order

   (ii) **tr a A < /etc/hosts | sort -r | pr -d > /etc/hosts**

   Translates all lowercase a to capital A in the file /etc/hosts then sorts all the lines in reverse order and formats the file for printing double-spaced

7. (i) Create and export a variable called **newhome** that is equivalent to the value contained in the **HOME** variable.

   export newhome = "$HOME"

   (ii) Find all files that start with the word "host" starting from /etc directory and save the stdout to a file called **file1** and the stderr to the same file.

   find /etc -name "host" >file 2>&1

   (iii) Display only the lines from the output of the set command that have the word "bash" in them. This output on the terminal screen should be sorted alphabetically.

   set | grep bash | sort

8. Write the expression that can be used to test whether:

   a. the user has read and execute permission to the /etc directory

   [-r /etc/hosts -a -x /etc/hosts]

   b. the contents of the variable **$TEST** are equal to the string "success" and the file /etc/hosts exists

   [$TEST="success" -a-f /etc/hosts]

   c. the contents of the variable **$TEST** are equal to the string "success, or the number 5 or the contents of the variable **$RESULT**

   [$TEST="success" -o $TEST -eq "5" -o $TEST="RESULT"]

9. Create the following shell scripts and trace their output.

```
     for file in `ls *.txt`
      do
         echo -n “Display $file? “
         read answer
         if [ $answer == ‘y’ ]
         then
            less $file
         fi
      done
```

10. Write a shell script that performs the following:

- Displays a list of currently logged-in users
- Display the system’s host name
- Display the time and date
- Displays the disk usage
- Display the pathname of the BASH shell

```
echo “User currently on the system: ”
who
echo -e “\nThe system’s hostname is: \e”
echo $HOSTNAME
echo -e “\nThe current date and time is: \e”
date
echo -e “\nCurrent disk usage is: ”
df
echo -e “\nThe pathname to the bash shell: \e”
echo $BASH
```

11. Write a script to display the time every 15 seconds. Read the date man page and display the time, using the **%r** field descriptor. Clear the window (using the clear command) each time before you display the time.

```
$ cat > datescript
while (true)
do
clear
date+%r
sleep15
 done
```

12. Write a script that takes the name of a directory as an argument and searches the file hierarchy rooted at that directory for zero-length files. Write the names of all zero-length files to standard output. If there is no option on the command line, have the script delete the file after displaying its name, asking the user for confirmation, and receiving positive confirmation. A **–f** (force) option on the command line indicates that the script should display the filename but not ask for confirmation before deleting the file.

```
If[ “$1” == “-f” ]
then
  find $2 -empty -print -exec rm -f {} \;(\; = im done)
else
  find $1 -empty -ok rm -f {} \;
fi
```

Extra exercise (Optional)

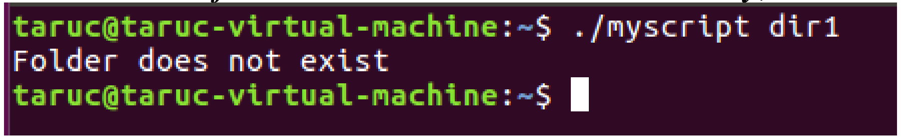
Create a shell script named “myscript” to check whether the directory name inserted by user exists beforehand under the user’s home directory.

If the directory does not exist, the system will prompt the message “Folder does not exist”. (Refer to Figure 1)

If the directory exists, the system will prompt the message “Folder <*directory_name*> exists” and the system will list/display all the contents inside such directory in a single column format. Then, the system will remove such directory together with all its content. (Refer to Figure 2)

Sample Output:

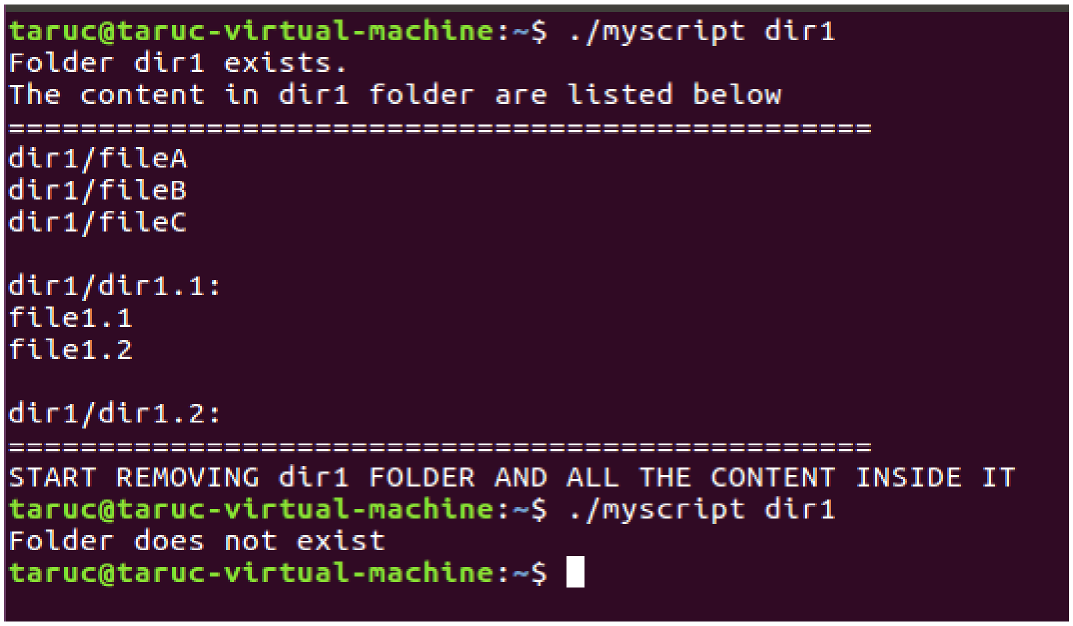
If the user types myscript in the terminal ***with parameter*** e.g. ***myscript dir1*** and the ***dir1 directory does not exist beforehand inside the user’s home directory***, the following output should be shown:



```
taruc@taruc-virtual-machine:~$ ./myscript dir1
Folder does not exist
taruc@taruc-virtual-machine:~$ 
```

Figure 1

If the user types myscript in the command prompt ***with parameter*** e.g. ***myscript dir1*** and the ***dir1 directory already exists beforehand inside the user’s home directory*** , the following output should be shown:



```
taruc@taruc-virtual-machine:~$ ./myscript dir1
Folder dir1 exists.
The content in dir1 folder are listed below
================================================
dir1/fileA
dir1/fileB
dir1/fileC

dir1/dir1.1:
file1.1
file1.2

dir1/dir1.2:
================================================
START REMOVING dir1 FOLDER AND ALL THE CONTENT INSIDE IT
taruc@taruc-virtual-machine:~$ ./myscript dir1
Folder does not exist
taruc@taruc-virtual-machine:~$ 
```

Figure 2

Answer:

```
#!bin/bash
if[ ! -d $1 ]
then
    echo “Folder does not exist”
else
    echo “Folder $1 exists”
    echo “The content in $1 folder are listed below”
    echo “==================================================”
    ls –format=single-column $1/*
    echo “==================================================”
    echo “START REMOVING $1 FOLDER AND ALL THE CONTENT INSIDE IT”
    rm -r $1
fi
```