Operating Systems Lab-3

Shell Programming (Part II)

Submitted By: Awais Saddiqui

Registration# 21pwcse1993

Section: "A"

Submitted to:

Mam Madiha Sher

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar.

Objectives:

The aim of this laboratory is to learn and practice SHELL scripts by writing small SHELL programs.
The following are the primary objectives of this lab session:

- SHELL keywords
- Arithmetic in SHELL script
- Control Structures
- Decision control
- Repetition control
- More UNIX commands
- Executing commands during login time

Handling shell variables:

The shell has several variables which are automatically set whenever you login. The values of some of these variables are stored in names which collectively are called you user environment.
Any name defined in the user environment, can be accessed from within a shell script. To include the value of a shell variable into the environment you must export it.

Example1:

Script to accept 5 numbers and display their sum.

Code:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
echo The parameter passed are :$1, $2, $3, $4, $5
echo The name of the script is : $0
echo The number of parameters passed are: $#
sum=`expr $1 + $2 + $3 + $4 + $5`
echo The sum is :$sum
```

Output:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1# ./task1.sh 2 3 4 5 6
The parameter passed are : 2 3 4 5 6
The name of the script is : ./task1.sh
The number of parameters passed are: 5
The sum is : 20
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1#
```

Example 2:

Write a script which will accept different numbers and finds their sum. The number of parameters can vary

Code:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
sum=4
while [ $# -gt 0 ]
do
        sum=`expr $sum + $1`
        shift
done
echo sum is $sum
```

Output:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1# ./task2.sh
sum is 4
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1#
```

Example 3:

Code:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
echo "arg1=$1 arg2=$2 arg3=$3"
   shift
      echo "arg1=$1 arg2=$2 arg3=$3"
         shift
            echo "arg1=$1 arg2=$2 arg3=$3"
               shift
                  echo "arg1=$1 arg2=$2 arg3=$3"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- REPLACE --                                                    1,1           All
```

Output:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1# ./task3.sh 5 6 4 5 6
 7 8 9 22 3 4 9
arg1=5 arg2=6 arg3=4
arg1=6 arg2=4 arg3=5
arg1=4 arg2=5 arg3=6
arg1=5 arg2=6 arg3=7
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1#
```

Example 4:

Code:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
echo "Enter a "
read a
echo "Enter b"
read b
if [ $a == $b ]
then
          echo "a is equal to b"
   elif [ $a -gt $b ]
   then
             echo "a is greater than b"
      elif [ $a -lt $b ]
      then
                echo "a is less than b"
         else
                   echo "None of the condition met"
fi
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                    1,1           All
```

Output:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1# vim task4.sh
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1# ./task4.sh
Enter a
5
Enter b
6
a is less than b
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1#
```

Example 5:

Code:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
if who | grep -s student > /dev/null
        then
                echo student is logged in
                  else
                        echo student is not available
fi
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- REPLACE --                                                    1,1          All
```

Output:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q
1#
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q
1# ./task5.sh
student is not available
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q
1#
```

Example 6:

Code:

Display a menu of options and depending upon the user's choice, #execute associated command

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
clear
echo "1. Date and time"
echo
echo "2. Directory listing"
echo
echo "3. Users information"
echo
echo "4. Current Directory"
echo
echo "Enter choice (1,2,3 or 4 ) :\c"
read choice
case $choice in
        1)      date;;
        2)      ls -l;;
        3)      who ;;
        4)         pwd ;;
        *)      echo wrong choice;;
esac
~
```

Output:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
1. Date and time

2. Directory listing

3. Users information

4. Current Directory

Enter choice (1,2,3 or 4 ) :\c
1
Tue Mar 14 23:21:19 PKT 2023
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q
1#
```

Example 7:

see if a number of people are logged in

Code:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
for i in $*
   do
if who | grep -s $i > /dev/null
then
echo $i is logged in
else
echo $i not available
fi
done
```

Output:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q
1# ./task7.sh awais
awais not available
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q
1#
```

Example 8:

Code:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
while echo "Please enter command"
read response
        do
             case "$response" in
          'done') break
                 ;;
               "") continue
                       ;;
               *)  eval $response
                       ;;
                esac
               done

~
~
~
~
~
~
```

Output:

```
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q
1# vim task8.sh
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q
1# ./task8.sh
Please enter command
ls
task1.sh   task11.sh  task2.sh  task4.sh  task6.sh  task8.sh
task10.sh  task12.sh  task3.sh  task5.sh  task7.sh  task9.sh
Please enter command
done
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q
1#
```

Example 9:

To show use of case statement.

Code:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
echo What kind of tree bears acorns\ ?
read responce
case $responce in
        [Oo][Aa][Kk]) echo $responce is correct ;;
        *) echo Sorry, responce  is wrong
esac
~
~
~
~
~
~
~
~
```

Output:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q
1# ./task9.sh
What kind of tree bears acorns ?
oak
oak is correct
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q
1#
```

Example 10:

Code:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
clear
echo What is the Capital of Pakistan \?
read answer
while test $answer != Islamabad
do
        echo No, Wrong please try again.
        read answer
done
~
~
~
~
~
~
~
```

Output:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
What is the Capital of Pakistan ?
Islamabad
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q
1#
```

Example 11:

Example to show use of until statement. Accept the login name from the user

Code:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
clear
echo "Please Enter the user login name: \c"
read login_name
until who | grep $login_name
do
          sleep 30
   done

~
~
~
~
```

Output:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
Please Enter the user login name: \c
Awais Saddiqui
```

Example 12:

Code:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
clear
echo "Enter the first number :\c"
read num1

echo "Enter the second number :\c"
read num2

echo "Enter the third number :\c"
read num3

if test $num1 -gt $num2
then
        if test  $num1 -gt $num3
             then
                  echo $num1 is the largest
              else
                  echo $num3 is the largest
         fi
              else
```

Output:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q1
Enter the first number :\c
2
Enter the second number :\c
4
Enter the third number :\c
8
8 is the largest
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3/q
1#
```

Assignment Problems on UNIX SHELL programming

Q_2: . Write a shell script that takes a keyword as a command line argument and lists the filenames containing the keyword

Code:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3
if [ -z "$1" ]; then
          echo "Usage: $0 keyword"
            exit 1
fi

grep -r -l "$1" .
# grep commmand is used to search files in current directory (*).
# -r option is used to search recursively in subdirectories.
# -l option is used to print only the file names.
~
~
~
~
```

Output:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3#
./Ques_2.sh ls
./Operating Systems lab03.doc
./q1/task12.sh
./q1/task4.sh
./q1/task5.sh
./q1/task6.sh
./q1/task7.sh
./Ques_3.sh
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3#
```

Q_3: Write a shell script that takes a command line argument and reports whether it is a directory, or a file or a link.

Code:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3
# Write a shell script that takes a command line argument and reports whether it is a directory,
# or a file or a link.

if [ -d "$1" ]; then
    echo "$1 is a directory"
  elif [ -f "$1" ]; then
     echo "$1 is a  file"
    elif [ -L "$1" ]; then
     echo "$1 is a  link"
    else
           echo "Enter a valid file name or path name"
     echo "$1 does not exist"
fi
```

Output:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3#
./Ques_3.sh assets
assets is a directory
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3#
```

Q_4: Write a script to find the number of sub directories in a given directory.

Code:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3
# Task 4
directory="$1"
if [ ! -d "$Directory" ]; then
                  echo "Error: $Directory is not a directory."
                            exit 1
fi

num_subdirs=$(find "$Directory" -mindepth 1 -maxdepth 1 -type d | wc -l)

echo "Number of subdirectories in $Directory is $num_subdirs subdir..."

~
~
~
```

Output:

```
root@DESKTOP-NEII4G1: /mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3#
./Ques_4.sh q1
Number of subdirectories in q1 is 0 subdir...
root@DESKTOP-NEII4G1:/mnt/e/Computer_System-Engineering/Fourth Semester/Operating System Lab/lab-3#
```