

## Lab Sheet 5

### Simple Shell Programming

#### 1. Basic Shell introduction

Following steps are required to write shell script:

- I. Use any editor like vi or mcedit to write shell script.
- II. After writing shell script set execute permission for your script as follows

*syntax:*

chmod permission your-script-name

*Examples:*

```
$ chmod +x your-script-name
```

```
$ chmod 755 your-script-name
```

**Note:** This will set read write execute(7) permission for owner, for group and other permission is read and execute only(5).

- III. (3) Execute your script as

*syntax:*

bash your-script-name

sh your-script-name

./your-script-name

Sr. No.	Command	Output
i	vi first # # My first shell script # clear echo "Knowledge is Power"	
ii	vi second # # Script to print user information on who is the user and current date # echo "Hello \$USER" echo "Today is ";date exit 0	

Try yourself :

- a. Create a shell script with your name as the script name
- b. In that shell script
  - i. write the function to display the calendar of 2013
  - ii. write the function to display the number of user login into the system
- c. Save the file, change the permission and run the script
- d. Take note on the output

## 2. Variable in Shell

- In Linux (Shell), there are two types of variable:

**System variables** - Created and maintained by Linux itself. This type of variable defined in CAPITAL LETTERS.

**User defined variables (UDV)** - Created and maintained by user. This type of variable defined in lower letters.

You can see system variables by giving command like **\$ set**, some of the important System variables are:

Try yourself :

- i. Search for 5 types of system variable and note down the function of each

## 3. UDV

- To print or access UDV use following syntax

*Syntax:*

**\$variablename**

- I. Define variable vech and n as follows:

```
$ vech=Bus
```

```
$ n=10
```

To print contains of variable 'vech' type

```
$ echo $vech
```

It will print 'Bus',

- II. To print contains of variable 'n' type command as follows

```
$ echo $n
```

Sr. No.	Command	Output
How to Define variable x with value 10 and print it on screen.		
How to Define variable xn with value Rani and print it on screen		

Try yourself : Test your knowledge on variable.

I. Copy the coding below in a new file “error”

```
#
#
# Script to test MY knowledge about variables!
#
myname=Vivek
myos = TroubleOS
myno=5
echo "My name is $myname"
echo "My os is $myos"
echo "My number is myno, can you see this number"
```

II. Run the file

III. If there is error, write down the syntax error

IV. Analyze the error, make the necessary changers and run it back

V. Write down the output

Try yourself : Write a simple shell script to display below output. In your script, you must declare **1 System variable** and **2 UDV**

Hello fskkp

My PC is using Ubuntu

My lab session is in FSK6

#### 4. Quotes

- “ - "Double Quotes" - Anything enclosed in double quotes removed meaning of that characters (except \ and \$)
- ‘ - 'Single quotes' - Enclosed in single quotes remains unchanged.
- ` - `Back quote` - To execute command

To look at the different function of quotes, let's try using arithmetic function “expr”

Try yourself

- i. Create both files below
- ii. Run the script and observe the output. Note down the output

Sr. No.	Command	Output
1	<pre>Execute the file below “quote”  #!/bin/bash #File to show the different function of quotes  echo “expr 6 + 3” echo 'expr 6 + 3' echo `expr 6 + 3`</pre>	
2	<pre>Execute the file below “math”  #!/bin/sh # File is to look at the different between expr and bc  #Declare 2 variable x=1 y=2  ##### Here's where we have the two options: # The expr method: exprans=`expr \$x + \$y`  # The bc method: bcans=`echo \$x + \$y   bc` ##### Did you see the difference?  echo "According to expr, \$x + \$y = \$exprans" echo "According to bc, \$x + \$y = \$bcans"</pre>	