

Lab Sheet 6

Shell Programming

1. Read Statement

Use to get input (data from user) from keyboard and store (data) to variable.

Syntax:

read variable1, variable2,...variableN

Following script first ask user, name and then waits to enter name from the user via keyboard. Then user enters name from keyboard (after giving name you have to press ENTER key) and entered name through keyboard is stored (assigned) to variable fname.

Sr. No.	Command	Output
i	<pre>\$ vi sayH # #Script to read your name from key-board # echo "Your first name please:" read fname echo "Hello \$fname, Lets be friend!"</pre>	

Exercises

- I. Write a shell script that prompt user for name, address and age. Lastly display all 3 info in sequence. Script shall be written base on condition below in separate filename:
 - a. Condition A – All three read statement must be in separated lines
 - b. Condition B – All three read statement must be in one line
- II. Observe the output and note down your findings

2. Shell arithmetic

- 1) Expr
 - a) Use to perform arithmetic operations. Use this for expression evaluator to calculate on variables (as integer numbers) or compare variables (as strings of text).
 - b) *Syntax:*
expr op1 math-operator op2

Examples:
\$ expr 1 + 3

```
$ expr 2 - 1
$ expr 10 / 2
$ expr 20 % 3
$ expr 10 \* 3
$ echo `expr 6 + 3`
```

c) **Note:**

expr 20 %3 - Remainder read as 20 mod 3 and remainder is 2.

expr 10 * 3 - Multiplication use * and not * since its wild card.

- d) In shell scripts, numeric calculations are done using the command expr. This command takes a series of arguments, each of which must contain a single token from the expression to be evaluated. Each number, or symbol must thus be a separate argument.

For example, the expression (3*4)+2 is written as:

```
expr '(' '3' '*' '4' ')' '+' '2'
```

2. BC

- a) Binary calculator is to carry out more complex mathematical operations (on decimals, fractions, and unusual bases).

- b) Example; 2.45 + 3.69

```
sum=`echo "2.45+3.69" | bc -l`
```

- c) For complex calculation, for example (3*4)+2

```
sum=`echo "((3 * 4)) + 2" | bc -l`
```

- d) We can set the precision with scale parameter:

```
echo "scale=4; 2.56*2.66" |bc
6.8096
```

Sr. No.	Command	Output
1	Echo `expr 6 - 3`	3
2	echo `expr 6 % 3`	0
3	echo `expr 6 + 3`	9

Try yourself.

- i. Start your shell scripting by creating a filename 'Veg'.

- a) This script requires user to input the weight of the lettuce . The output of the script will display the total amount of price base on the weight (Assume that 1Kg is RM2.50)

- b) Sample output will be

Enter the weight of lettuce in Kg
 [User input]
 Total price is RM [Result]
 Please pay at the counter

c) Round your answer in 2 dotted decimal

ii. Start your shell scripting by creating a filename 'MyCGPA'.

a) This script requires user to input the name and last semester GPA. The output of the script will display you current CGPA (Assume that your current CGPA for 2 semester is 3.5)

b) Sample output will be:

What is your name?
 [User input]
 What is your last semester GPA?
 [User input]
 Your CGPA is [Result]

c) Round your answer in 2 dotted decimal

3. If - else Statement

If given condition is true then command1 is executed otherwise command2 is executed.

Syntax:

```
if condition
then
    condition is zero (true - 0)
    execute all commands up to else statement

else
    if condition is not true then
    execute all commands up to fi

fi
```

Sr. No.	Command	Output
i	\$ vi nested echo "1. Unix (Sun Os)" echo "2. Linux (Red Hat)" echo -n "Select your os choice [1 or 2]? "	

	<pre> read osch if [\$osch -eq 1] ; then echo "You Pick up Unix (Sun Os)" else ##### nested if i.e. if within if ##### if [\$osch -eq 2] ; then echo "You Pick up Linux (Red Hat)" else echo "What you don't like Unix/Linux OS." fi fi </pre>	
--	--	--

Try yourself.

- i. Write a shell script, which will ask you for input 'yes' or 'no' and based on that will give you what is your input,
 - a) If you enter yes as your input it will echo you have entered yes.
 - b) If you enter no as your input it will echo you have entered no.
 - c) No or Invalid input is given than gives error 'no input given'.

- ii. Write a shell script base on below requirements:
 - a) Script prompt user to enter a password
 - b) Script then checks the password entered with the given password. (Assume that the system password is “secret”)
 - c) Script will display “Your have access!!” if password is correct or “Get out from this system” if password is wrong