LAB12

Course Code: CSC 2209 Course Title: Operating Systems

Dept. of Computer Science
Faculty of Science and Technology

Lecturer No:	**12**	**Week No:**	**12**	**Semester:**	
Lecturer:	*Name & email*				

Lecture Outline

1. Function
2. Advantages of Functions in Shell Script
3. Syntax of Functions in Shell Script
4. Passing Parameters on function
5. Function Return

Function

- A function is a group of commands that are assigned a name that acts like a handle to that group of commands. To execute this group of commands defined in the function, you simply call the function by the name you provided.
- There will be cases where you need to execute a block of code that achieves a specific procedure several times in different places in your shell script. Shell functions are like subroutines, procedures, and functions in other programming languages.

Advantages of Functions in Shell Script

- It helps us to reuse the code.
- Improve the readability of the program.
- Efficient use of variables inside the program.
- Allows us to test the program part by part.
- Displays program as a bunch of sub-steps.

Syntax of Functions in Shell Script

- **Syntax1:**

```
function function_name
{
###set of commands
}
```

- **Syntax2:**

```
function_name()
{
####set of commands
}
```

- **function** is a key word that declares the function definition.
- **function_name** is the name of the declared function.

- Curly braces {} acts as delimiters that enclose the function's code.
- Set of commands are the code to be executed when the function is called.

- So, a function is declared first, and then called when needed.

Function Example-1

```
#!/bin/bash
 myfunction()
 {
 echo “Oh! Actually, it works”
 }
 myfunction
```

After creating the function myfunction, it was then invoked by calling its function name to our main routine. The main routine will be anywhere in our script that was not defined as part of our function.

Function Example-1 (cont'd)

```
#!/bin/bash
echo "For testing"
myfunction
myfunction() {
   echo "Oh! Actually, it works"
}
```

The line 3 in the code returns a command not found error. Because the function only works if it is declared before your main routine. The interpreter will return an error if you have declared your function after your main routine.

Passing Parameters on function

You can pass parameters and process those data in bash function. The code shows the procedure on how to pass parameters in shell scripting

```
#! /bin/bash
myfunction()
{
echo “First argument is as $1”
echo “Second argument is as $2”
}
myfunction “Hello” “World”
```

Note: The 1 and 2 in the above are local variables and thus, are not accessible to other parts of the script aside from the function where the parameters are being passed.

Let’s understand the code:

1. We added the values "Hello" and "World" after we called the myfunction .
2. Those values are passed to the myfunction as parameters and stored in a local variable.
3. The interpreter stores the passed values into predefined variables, which is named according to the sequence of passing the parameters, 1 as the starting name up to the order of passing.
4. Note that the "Hello" world is stored in the variable 1 and value "World" is stored in variable 2.

Function Return

```
#!/bin/bash
add()
{
sum=$(($1+$2))
return $sum
}
read “operand1” op1
read “operand2” op2
add op1 op2
echo “Summation: ” $?
```

function can return the values of a function's local variable to the main routine by using the keyword return. The returned values are then stored to the default variable **$?**

Practice Questions

- Write a function to find the Fibonacci series till nth term.
- Write a function to find the max of two number and use the function in such a way to find the max of three number.
- Write a function to find prime number.
- To create a simple calculator write four function for addition, subtraction, multiplication, and division and show use of all the functions.

Books

- Unix Shell Programming
  - Written by Yashavant P. Kanetkar