

Experiment 6: Shell Scripting Part II

1. Aim:

- a. To understand the concepts of Arrays and functions using shell scripts.

2. Requirements: Linux OS

3. Related Theory:

Arrays:

An array is a systematic arrangement of data. In shell script, array is a variable that an associated value. Associative arrays provide a way to index and retrieve values contains multiple values, may be of same type or different type since by default in shell script everything is treated as a string. An array can be categorized into two:

- a) Indexed Arrays: array where elements (values) are stored and assigned with an integer number starting from 0 to N. The elements inside the array are referenced using the index number.
- b) Associative Array: a data structure for storing key-value pairs. Every key is unique and has based on corresponding keys.

There are 3 different methods of declaring an array:

- a) Indirect Declaration: The array is not declared separately. The values of the elements are assigned to the array elements index specifically.
- b) Explicit Declaration: The array is declared separately before the values are assigned to the array elements.
- c) Compound Declaration: Declare array with it's initial set of values. Additional values can be added to the array as needed.

Functions:

A function is a collection of statements that execute a specified task. Its main goal is to break down a complicated procedure into simpler subroutines that can subsequently be used to accomplish the more complex routine. Here functions can be written in the same script where it is used or can be written as a library of useful functions that can be imported in the script where it is required.

Types of functions:

- a) Functions that return a value to the calling section of the script using 'return' keyword.
- b) Functions that terminates the shell script using the 'exit' keyword.
- c) Functions that change the value of a variable/variables.
- d) Functions that echo output to the standard output.

4. Laboratory Exercise:

Write shell scripts for the above concepts.

5. Post-Experiment Exercise:

A. Conclusion:

#Summarize your experience about the skills acquired from this experiment.

B. Tasks:

1. Define an array containing various Linux OS flavor names as its elements. Get the index from user and display the array element at that index. If the index exceeds the array length, display an error message.
2. Continued from Q.1: Get the index, start location and last location from user and display the array element values at that index from start to the last location. If the index exceeds the array length or if the start and end locations are inappropriate, display an error message.
3. Continued from Q.1: Get index from the user and remove the array element at that index. Display the updated array.
4. Create two arrays with some elements. Concatenate the two arrays to create the third array. Display all three arrays.
5. Load the content of a file into an array and display the contents using for loop.
6. Create an associative array for Beatles music group, the keys should be singer, bassist, drummer, guitarist, and the values will be the names of the musicians of this group. Display all array elements using for loop. Then, delete the guitarist from this array and display the updated array.
7. Write a shell script that consists of a function that displays the number of files in the present working directory. Name this function "file_count" and call it in your script. If you use variable in your function, make it a local variable.
8. Write a shell script to find factorial of given command line arg using recursion. If the commandline argument (the number) is missing, display a message explaining the usage of this function.

9. Write a shell script to terminate the script if invoked by non-root user using function and appropriate exit codes.

10. Scope of variables in the function: Define two variables $x = 1$ and $y = 1$. Write a function called "change" that changes the value of x locally and value of y globally. Display the values of x and y before calling the function as well as after calling the function.

11. Nested functions: Write a shell script and define an outer function "Out1" which itself defines a nested function "In" which displays "Inner function #1" and "Out1" displays "Outer function #1". Further, define the second outer function "Out2" which again defines a nested function "In" which displays "Inner function #2" and "Out2" displays "Outer function #2". Call Out1, In, Out2, In. Execute the script and state your interpretation.

Submission Instructions:

For each question, students need to submit the question, screenshot of the scripts and outputs. The screenshots must cover all possible options in the menu. In case the output is big, take partial screenshot.