Shell Programming

1. A shell program is nothing but a series of unix commands.
2. Instead of specifying one job at a time, the shell is given a to-do-list of a program that carries out an entire procedure.
3. Such programs are known as shell scripts.
4. Shell programming language incorporates most of the features that most modern day programming languages offer.

Shell variables – Rules for building shell variables are as follows:

1. A variable name is any combination of alphabets, digits and an underscore (‘_’).
2. No commas or blanks are allowed within a variable name.
3. The first character of a variable name must either be an alphabet or an underscore.
4. Variable names should be of any reasonable length.
5. Variable names are case sensitive.

Input/Output

1. Keywords for accepting input – read
2. Displaying output – echo

Assigning value to variables –

1. Values can be assigned to variables through a read statement or also by using a simple assignment operator. For ex: age=30

Note : While assigning values to variables using assignment operator, no spaces to be given on either side of it. If the variable doesn’t exist it will be created and value assigned

Note : To print or access the value of a variable use ‘$’ .

For ex: To print value of variable ‘flag‘ write - echo $flag

Arithmetic in Shell script -

1. All shell variables are string variables, hence to carry out arithmetic operations use expr command which evaluates arithmetic expressions.
2. More than one assignment can be done in a single statement.
3. Before and at the end of expr keyword use ` (back quote) sign not the (single quote i.e.') sign which is generally above TAB key.

4. Terms of the expression provided to expr must be separated by blanks. Thus expression expr 10+20 is invalid.
5. The '*" symbol must be preceded by a \ ,otherwise the shell treats it as a wildcard character for all files in the current directory.

OPERATORS USED IN SHELL SCRIPT – OPERATOR MEANING

1. –gt Greater than
2. –lt Less than
3. –ge Greater than or equal to
4. –le Less than or equal to
5. –ne Not equal to
6. –eq Equal to
7. –a Logical AND
8. –o Logical OR
9. ! Logical NOT

CONTROL INSTRUCTIONS IN SHELLS –

There are four types of control instructions in shell :

1. Sequence Control Instruction.
2. Selection or Decision Control Instruction
3. Repetition or Loop control Instruction
4. Case Control Instruction

Decision statements –

If-then-else-fi statements:

if condition then Commands else Commands fi

For statements:

for control variable in value1 value 2 value3 do

Command list done

While statements:

while control command do

Command list Done

Until statements:

until control command do

Command list done

Case statements:-

case value in

choice 1) commands;;

choice 2) commands;;

esac

Steps to write and execute a script

1. Open the terminal. Go to the directory where you want to create your script.
2. Create a file with . sh extension.
3. Write the script in the file using an editor.
4. Make the script executable with command chmod +x <fileName>.
5. Run the script using ./<fileName>.

LAB Assignment 3

Operating Systems (UCS-303)

Instructions: The instructor is required to discuss the basic syntax of shell programming; students have to implement the following programs using shell script.

1. Write a shell program to add two numbers.
2. Write a shell program to compare two strings using command prompt.
3. Write a shell program to generate Fibonacci series.
4. Write a shell program to check whether a character is VOWEL or CONSONANT using switch.
5. Write a shell program to display the total number of words and and total number of lines in a file.
6. A script that takes a filename as input and displays information about the file, such as its size, permissions, owner, and modification time.
7. Create a script that displays information about the system, such as the OS version, CPU, memory, disk usage, and network details.

ONLINE Compiler: https://www.mycompiler.io/new/bash