Shell Scripting Exercises *(2)*

by Sanka Dharmarathna

Exercise_1 - Write a shell script that accepts a file or directory name as an argument. Have the script report if it is regular file, a directory, or another type of file. If it is a directory, exit with a 1 exit status. If it is some other type of file, exit with a 2 exit status.

```
#!/bin/bash

FILE=$1

if [ -f $FILE ]
  then
    echo “It is reguler File”
    exit 0

elif [ -d $FILE ]
   then
     echo “It is directory”
     exit 1

 else
    echo “Another type”
    exit 2
fi
```

Exercise_2 - Write a script that executes the command “cat/etc/shadow”. If the command return a 0 exit status, report “command succeeded” and exit with a 0 exit status. If the command returns a non-zero exit status, report “Command failed” and exit with a 1 exit status.

```
#!/bin/bash

cat /etc/shadow

if [ “$?” -eq “0” ]
  then
    echo “Command succeeded”
    exit 0

  else
    echo “Command failed”
    exit 1
fi
```

Exercise_3 - 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, remember to make it a local variable.

```
#!/bin/bash

function file_count()
 {
   local NUMBER_OF_FILE=$(ls -l | wc -l)
    echo "$NUMBER_OF_FILE"
 }
```

Exercise_4 - Modify the script from the previous exercise. Make the “file_count” function accept a directory as an argument. Next, have the function display the name of the directory followed by a colon. Finally display the number of files to the screen on the next line. Call the function three times. First on the “/etc” directory, next on the “/var” directory and finally on the “/usr/bin” directory.

```
#!/bin/bash

function file_count()
 {
   local Directory=$1
   COUNT_FILE=$(ls $Directory|wc -l)
   echo "$Directory"
   echo "$COUNT_FILE"
 }

file_count /etc
file_count /var
file_count /usr/bin
```