VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

OPERATING SYSTEMS

LAB 1

AIM: To script and automate various tasks using BASH programming.

20BDS0405 (BIMAL PARAJULI)

8/20/2021

DESCRIPTION:

BASH stands for Bourne Again Shell. It is the default shell of UNIX and UNIX like systems i.e. MacOS and Linux.

Bash is also a very useful programming language.

We can use the bash programming language and script and automate tasks that require repetitive iterations and human interventions.

TASK1:

1. Write a shell program for the following operations:

a. Given an input file which contains 15 student names, create separate directories for each student (Directory name is same as the student's name).

b. Create a shell file inside each directory with the student's name.

c. Each shell file should contain the hashbang statement and also include a statement for printing the corresponding student name.

d. Assign file permissions(executable) to each of these shell files and execute all of them in sequence to get the final output.

e. Display the files and directories inside the parent working directory.

CODE:

```
#! /bin/bash

cat Names.txt | while read reg_no      #read the file and store each line in a variable

do

   mkdir $reg_no                  #make a folder for every registration number.

   cd $reg_no                     #move into that folder.

   touch $reg_no.sh               #make a new bash file inside that folder named corresponding Reg. No.

   echo "#!/bin/bash" >> $reg_no.sh   #Append shebang line in the bash file.

   echo "echo $reg_no" >> $reg_no.sh  #Append a echo line to print reg no in that bash file.

   chmod 777 $reg_no.sh           #make the bash file executable

   ./$reg_no.sh                   #execute that bash file

   cd ..                          #move out of that folder
done



echo "================================================"
echo "Files and folders list for verification:"

ls  *                             #display all directories and its files.
```

TASK2:

A directory consists of different types of files (.c, .txt and .sh). Write a shell script to segregate and store the name of the files according to their types, into separate files. [You have to create three separate files, for the files with .c , .txt and .sh extension].

CODE:

```
touch c_files.txt     #Creating three new files in working directory

touch txt_files.txt

touch sh_files.txt


# Writing initial line of each files

echo " ---Files having extension .c---" > c_files.txt

echo " ---Files having extension .txt ---" > txt_files.txt

echo " ---Files having extension .sh ---" > sh_files.txt


for filename in `ls $search`; #Looping through all the filenames in working directory and saving their names in a variable.

do

    if [[ $filename == *.txt ]]          #if it is .txt file, redirect it to txt_files.txt

    then

       echo $filename >> txt_files.txt

    elif [[ $filename == *.c ]]          #if it is .c file, redirect it to c_files.txt

    then

       echo $filename >> c_files.txt

    elif [[ $filename == *.sh ]]         #if it is .sh file, redirect it to sh_files.txt

    then

       echo $filename >> sh_files.txt

    fi

done


ls -l *                        #List all the files, directories and subdirectories in working directories

echo ==============================
```

```
echo

head sh_files.txt

echo ================================

echo

head txt_files.txt

echo ================================

echo

head c_files.txt

echo ================================

echo


echo finished
```

TASK3:

Write a Shell program to read a 5 digit number from user and find the number of occurrences of second digit of that number.

Sample Input: 16386

Second Digit: 6

Number of Occurrences of second digit= 2

CODE:

```
#! /bin/bash

echo -n "Enter a 5-digit number: "

read A                          #Get a 5 digit number as input from the user.



second_digit="${A:1:1}"

echo Second Digit: $second_digit        #Extract the 2nd digit from that number.



counter=0

for ((i=0; i<5; i++))

do

each_digit=${A:i:1}

if [[ $second_digit = $each_digit ]]   #Loop through and check if each digit equals
to 2nd digit.

   then

   counter=$((counter+1))

   fi

done


echo Number of Occurrences of $second_digit is: $counter
```