ADIT LUHADIA

190911112

IT A

**LAB 3 - UNIX SHELL PROGRAMMING (SHELL SCRIPTING)**

1. Write a shell script to find whether a given file is the directory or regular file.

#!/bin/bash

echo "Enter the path of the file/directory"

read a;

if [ -d "${a}" ] ; then

    echo "$a is a directory";

elif [ -f "${a}" ]; then

    echo "${a} is a file";

else

    echo "${a} is not valid";

fi

Text

Description automatically generated

2. Write a shell script to list all files (only file names) containing the input pattern (string) in the folder entered by the user.

#!/bin/bash

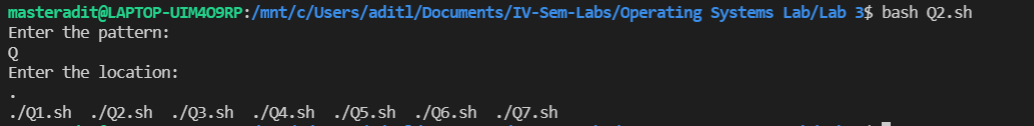
echo "Enter the pattern:"

read pattern

echo "Enter the location:"

read location

ls $location/\*$pattern\*



3. Write a shell script to replace all files with .txt extension with .text in the current directory. This has to be done recursively i.e if the current folder contains a folder “OS” with abc.txt then it has to be changed to abc.text ( Hint: use find, mv )

#!/bin/bash

echo "Replacing .txt with .text"

result=`find . -depth -name \*.txt`

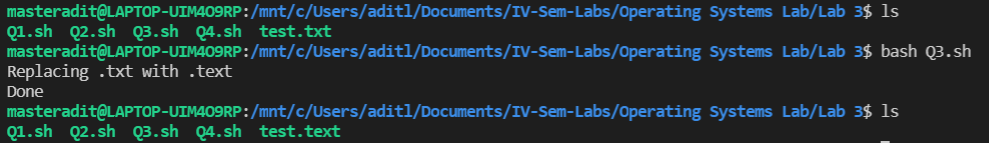
for i in $result

do

    mv $i ${i%.txt}.text

done

echo "Done"



4. Write a shell script to calculate the gross salary. GS=Basics + TA + 10% of Basics. Floating point calculations has to be performed.

#!/bin/bash

echo "Gross = Basics + TA + 10% of Basics"

echo "Enter Basics:"

read basics

echo "Enter TA"

read TA

echo `echo "$basics\*1.1 + $TA" | bc -l ` #Using Piping

echo `bc -l <<< "$basics\*1.1 + $TA"`     #Using Streams

Text

Description automatically generated

5. Write a program to copy all the files (having file extension input by the user) in the current folder to the new folder input by the user. ex: user enter .text TEXT then all files with .text should be moved to TEXT folder. This should be done only at single level. i.e if the current folder contains a folder name ABC which has .txt files then these files should not be copied to TEXT.

#!/bin/bash

echo "Enter the file extension to be moved:"

read extension

echo "Enter the folder where the files will be moved to:"

read folder

result=`find ./ -maxdepth 1 -name \*$extension`

mkdir $folder

for i in $result

do

    mv $i $folder

done

Text

Description automatically generated

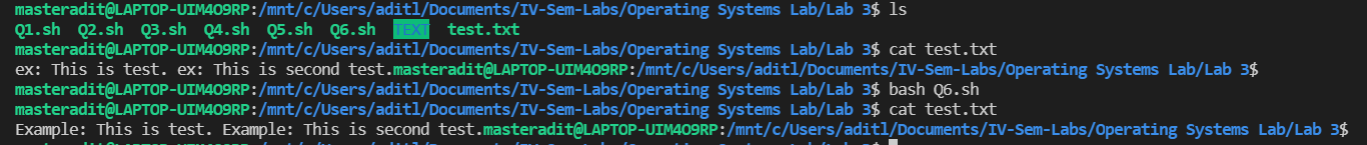
6. Write a shell script to modify all occurrences of “ex:” with “Example:” in all the files present in current folder only if “ex:” occurs at the start of the line or after a period (.). Example: if a file contains a line: “ex: this is first occurrence so should be replaced” and “second ex: should not be replaced as it occurs in the middle of the sentence.”

#!/bin/bash

# Program to replace all "ex:" with "Example:"

sed -i -e 's/^ex:/Example:/' \*.txt       #Replace the ex: in beginning of lines

sed -i -e 's/\. ex:/\. Example:/' \*.txt  #Replace the ex: occuring after .



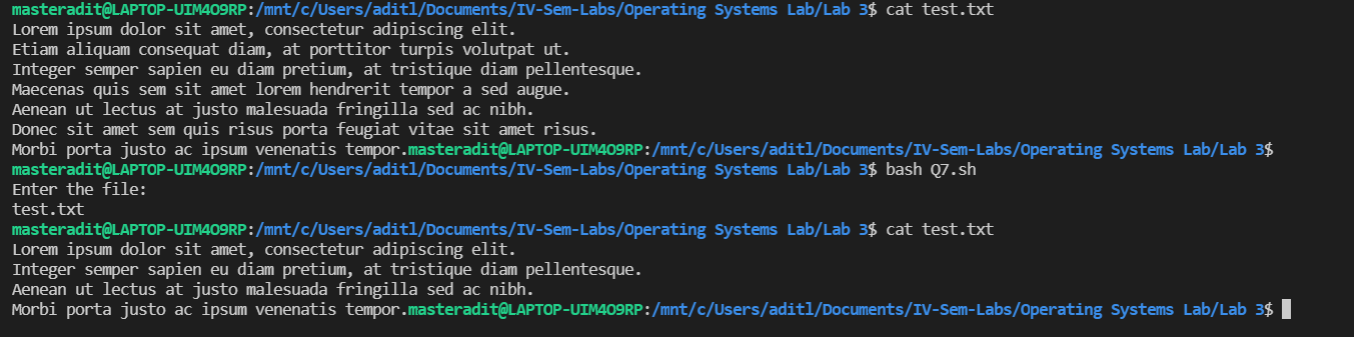
7. Write a shell script which deletes all the even numbered lines in a text file.

#!/bin/bash

echo "Enter the file:"

read file

sed -i '2~2d' $file



**LAB 4 - ADVANCED SHELL SCRIPTING**

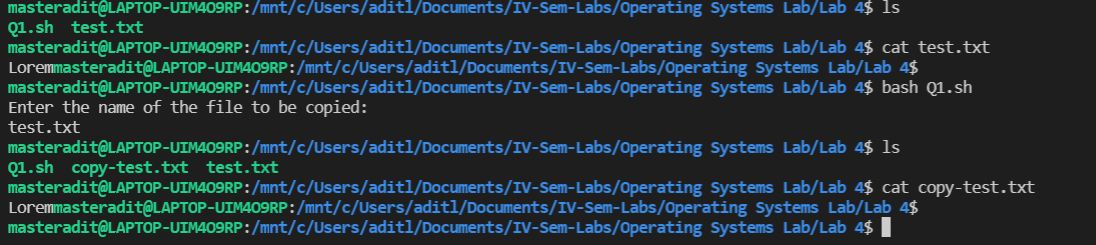
1. Write a shell script to make a duplicate copy of a specified file through command line.

#!/bin/bash

echo "Enter the name of the file to be copied:"

read file

cp $file copy-$file



2. Write a shell script to remove all files that are passed as command line arguments interactively.

#!/bin/bash

n=$#

if [ $n -eq 0 ]

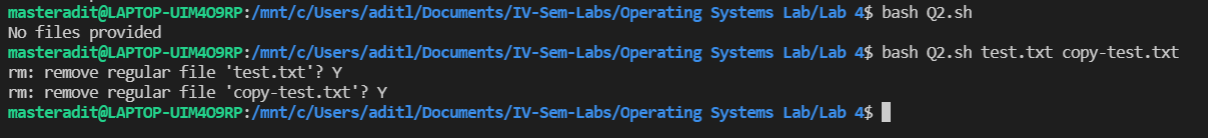
then

    echo "No files provided"

    exit 1

fi

rm -i $\*



3. Write a program to sort the strings that are passed as a command line arguments. (ex: ./script.sh “OS Lab” “Quoted strings” “Command Line” “Sort It”. The output should be “Command Line” “OS Lab” “Quoted strings” “Sort It”. ( make use of usrdefined sort function)

#!/bin/bash

sort(){

    list=("${!1}")

    for ((i=0; i<5; i++))

    do

            for ((j=0; j<5-i-1; j++))

            do

                if  [ "${list[j]}" \> "${list[$((j+1))]}" ]

                then

                        temp=${list[j]}

                        list[j]=${list[$((j+1))]}

                        list[$((j+1))]=$temp

                fi

            done

    done

    echo "Sorted Array:"

    echo ${list[\*]}

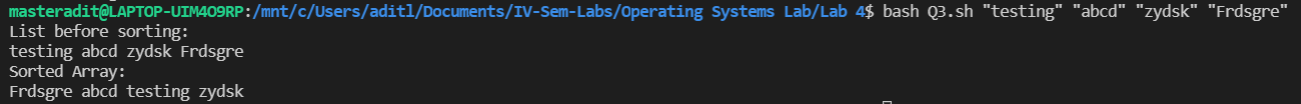
}

a=(${\*})

echo "List before sorting:"

echo ${a[\*]}

sort a[@]



4. Implement wordcount script that takes -linecount, -wordcount, -charcount options and performs accordingly, on the input file that is passed as command line argument (use case statement)

#!/bin/bash

case $1 in

    -linecount)

            wc "$2" -l;;

    -wordcount)

            wc "$2" -w;;

    -charcount)

            wc "$2" -c;;

    \*);;

esac

Graphical user interface, text

Description automatically generated

5. Write a menu driven shell script to read list of patterns as command line arguments and perform following operations.

a. Search the patterns in the given input file. Display all lines containing the pattern in the given input file.

b. Delete all occurrences of the pattern in the given input file.

c. Exit from the shell script.

#!/bin/bash

where=$1

shift       #Rotates parameters counter-clockwise

patterns=$1

for ((value=2; value<=$#; value++))

do

    patterns+="\|${!value}"

done

while (true)

do

    echo "1. Search the patterns within the file"

    echo "2. Delete the patterns from the file"

    echo "3. Exit"

    read choice

    case $choice in

        1) grep -i $patterns $where ;;

        2) for i

           do

               sed -e s/$i//g $where > text.txt

               mv text.txt $where

           done;;

        3) exit 0;;

    esac

done

Text

Description automatically generated