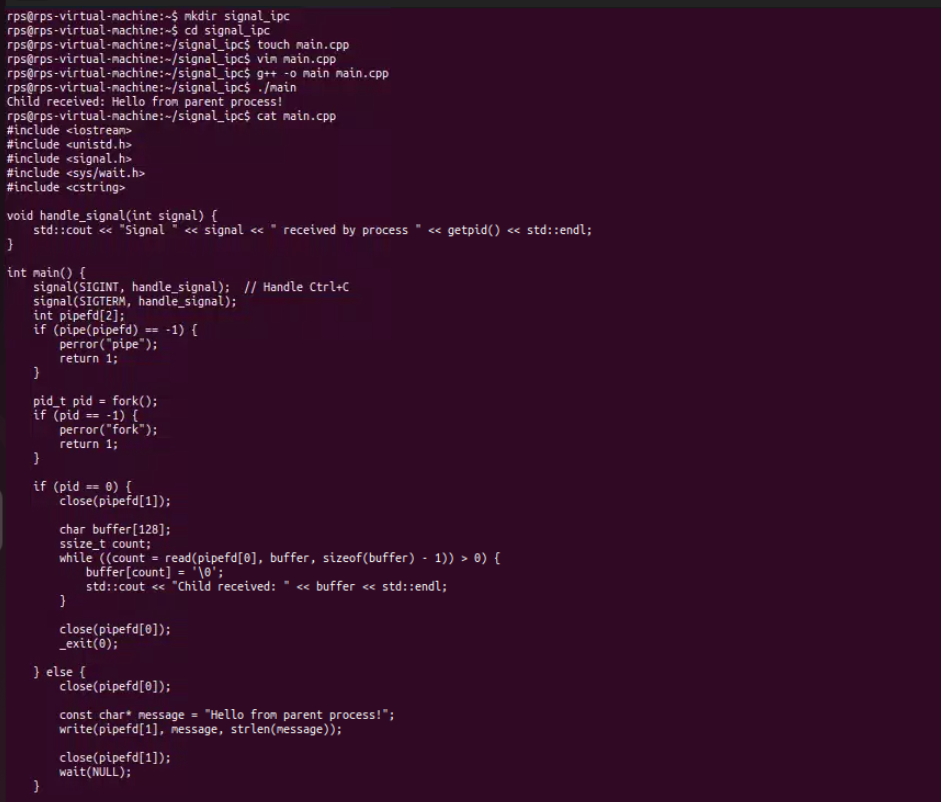
Problem Statement: Signal Handling and Inter-Process Communication using Pipes in C++

Design and implement a robust system in C++ that effectively utilizes signals to control the behavior of multiple processes and employs pipes for inter-process communication, enabling coordinated data exchange and process synchronization.



#!/bin/bash

echo "simple calculator"

sum=0

i="y"

echo "enter first number"

read n1

echo "enter second number"

read n2

while [ $i = "y" ]

do

echo "1.Addition"

echo "2.Subtraction"

echo "3.Multiplication"

echo "4.Division"

echo "Enter choice"

read ch

case $ch in

1)sum=$(echo " $n1 + $n2" | bc -l)

echo "Addition is =" $sum;;

2)sum=$(echo "$n1 - $n2" | bc -l)

echo "Sub is =" $sum;;

3)sum=$(echo "$n1 \* $n2" | bc -l)

echo "Mul is =" $sum;;

4)sum=$(echo "$n1 / $n2" | bc -l)

echo "div is =" $sum;;

\*)echo "invalid choice"

esac

echo "Do you want to continue"

read i

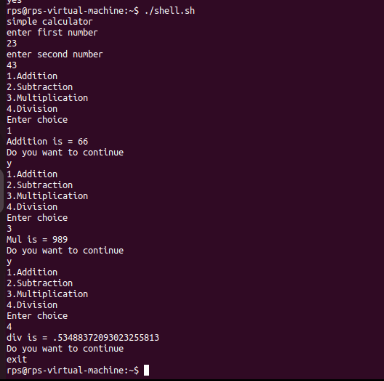
if [ $i != "y" ]

then

exit

fi

done



Assignments

1.Change File Permissions

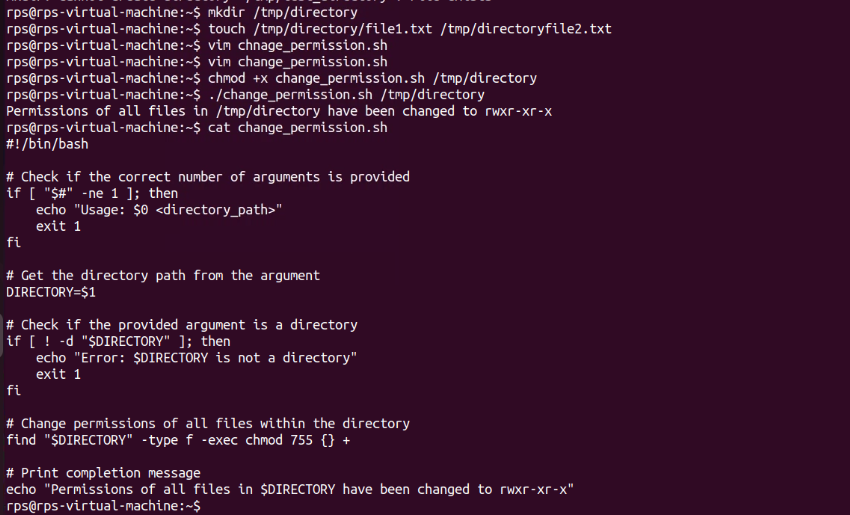
Description: Write a shell script that takes a directory path as an argument and changes the permissions of all files within that directory to read, write, and execute for the owner, and read and execute for the group and others.

Instructions:

The script should accept one argument, the directory path.

Change permissions of all files in the specified directory to rwxr-xr-x.

Print a message indicating the completion of the permission change.



Problem 2: Count Files and Directories

Description: Write a shell script that counts the number of files and directories in a given directory.

Instructions:

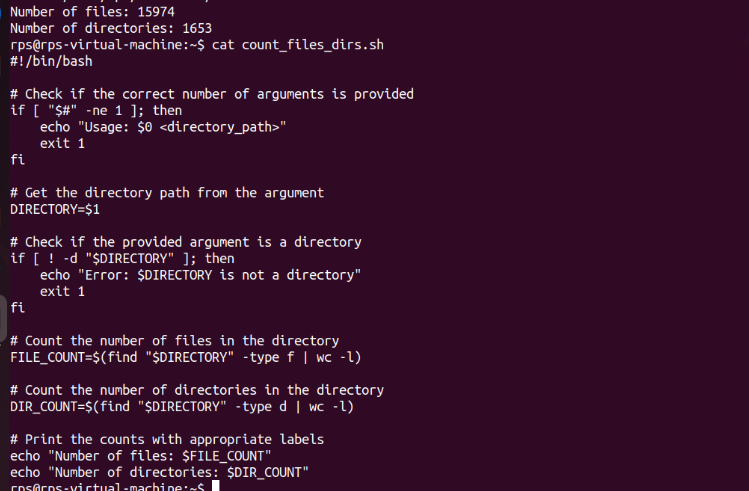
The script should accept one argument, the directory path.

Count the number of files and directories separately.

Print the counts with appropriate labels.

Sample Input:

./count\_files\_dirs.sh /path/to/directory



Problem 3: Find and Replace Text in Files

Description: Write a shell script to search for a specific text string in all files within a directory and replace it with another string.

Instructions:

The script should accept three arguments: directory path, search string, and replacement string.

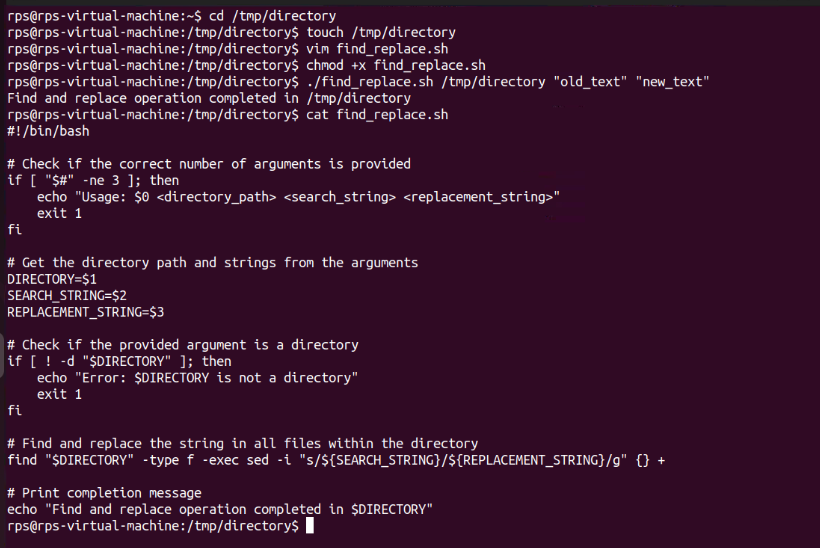
Search for the specified string in all files within the directory.

Replace the string with the given replacement string in all occurrences.

Print a message indicating the completion of the find and replace operation.

Sample Input:

./find\_replace.sh /path/to/directory "old\_text" "new\_text"



Problem 4: Disk Usage Report

Description: Write a shell script that generates a report of disk usage for a specified directory.

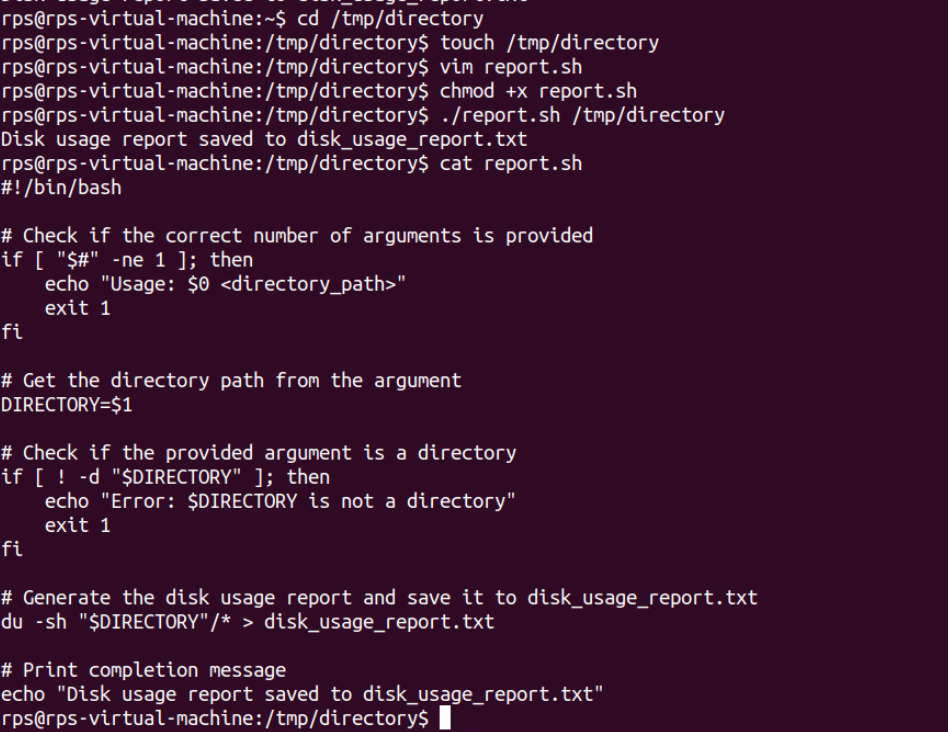
Instructions:

The script should accept one argument, the directory path.

Use the du command to generate a disk usage report for the directory.

Save the report to a file named disk\_usage\_report.txt in the current directory.

Print a message indicating where the report is saved.



Problem Statement 5: File Management Script with Functions and Arguments

Objective

Create a shell script that manages files in a specified directory. The script should include functions to perform the following tasks:

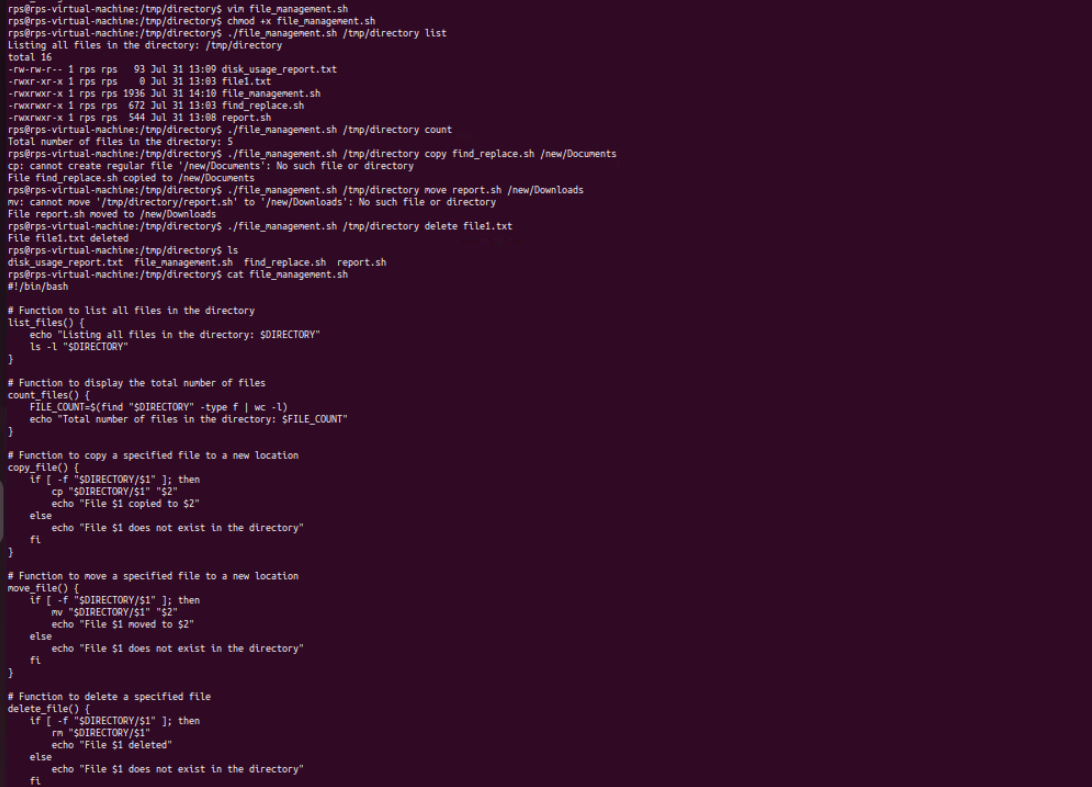
List all files in the directory.

Display the total number of files.

Copy a specified file to a new location.

Move a specified file to a new location.

Delete a specified file.



After writing the code please read from user and write on screen using read and write apis in cpp

