Name: Jimmy Battistoni

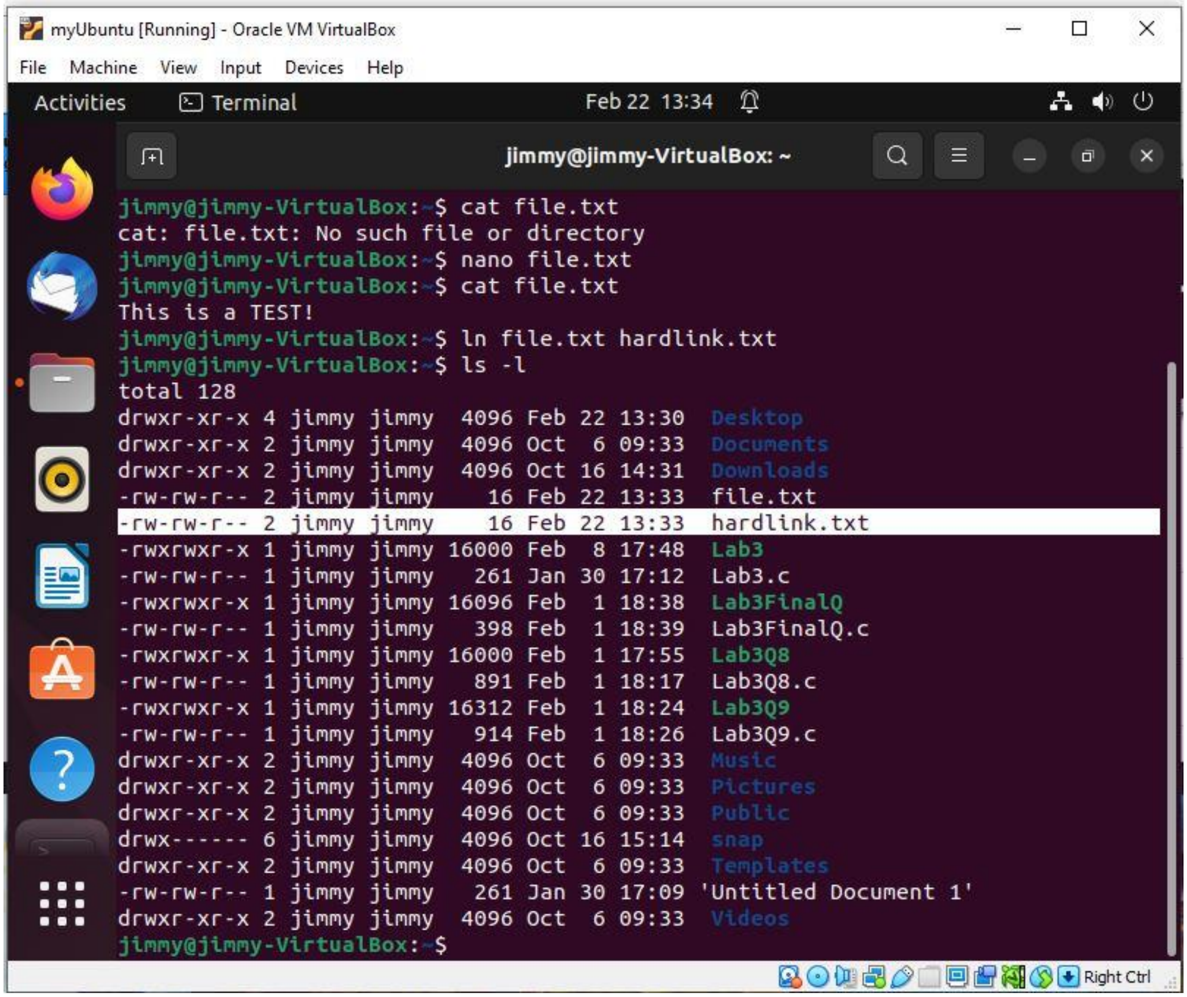
Date: 02/22/2023

Course: CS470 Operating System

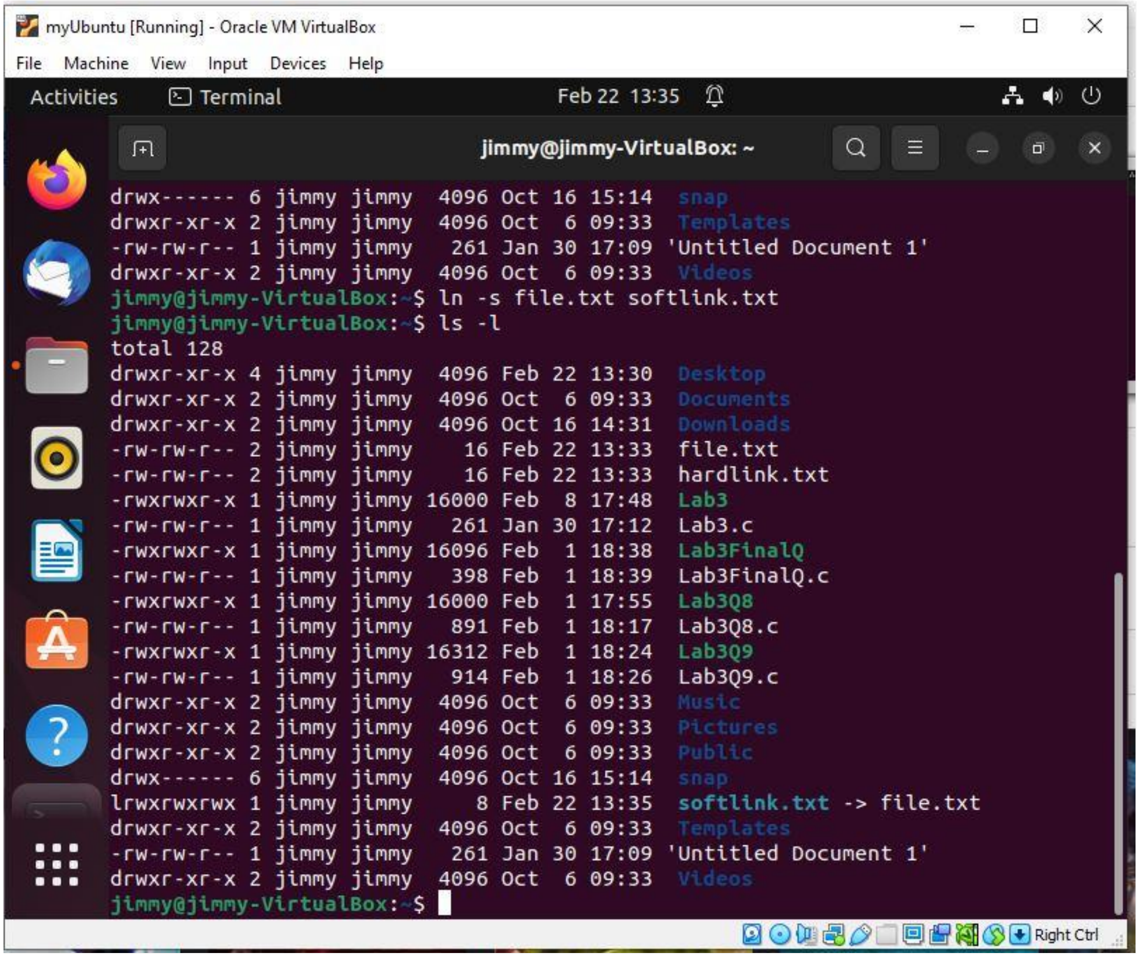
Assignment: Lab 4

1. Using the ln and ln -s commands, create hard and soft links

The following commands will create a hard link between the file.txt to create a hard link to the hardlink.txt file. Below is a screenshot that contains the original file known as file.txt that now has been hard linked with the hardlink.txt file. As seen in the screen shot below verifies that the hard link between file.txt and hardlink.txt:


myUbuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Feb 22 13:34
jimmy@jimmy-VirtualBox: ~
Right Ctrl

```
jimmy@jimmy-VirtualBox:~$ cat file.txt
cat: file.txt: No such file or directory
jimmy@jimmy-VirtualBox:~$ nano file.txt
jimmy@jimmy-VirtualBox:~$ cat file.txt
This is a TEST!
jimmy@jimmy-VirtualBox:~$ ln file.txt hardlink.txt
jimmy@jimmy-VirtualBox:~$ ls -l
total 128
drwxr-xr-x 4 jimmy jimmy  4096 Feb 22 13:30  Desktop
drwxr-xr-x 2 jimmy jimmy  4096 Oct  6 09:33  Documents
drwxr-xr-x 2 jimmy jimmy  4096 Oct 16 14:31  Downloads
-rw-rw-r-- 2 jimmy jimmy    16 Feb 22 13:33  file.txt
-rw-rw-r-- 2 jimmy jimmy    16 Feb 22 13:33  hardlink.txt
-rwxrwxr-x 1 jimmy jimmy 16000 Feb  8 17:48  Lab3
-rw-rw-r-- 1 jimmy jimmy   261 Jan 30 17:12  Lab3.c
-rwxrwxr-x 1 jimmy jimmy 16096 Feb  1 18:38  Lab3FinalQ
-rw-rw-r-- 1 jimmy jimmy   398 Feb  1 18:39  Lab3FinalQ.c
-rwxrwxr-x 1 jimmy jimmy 16000 Feb  1 17:55  Lab3Q8
-rw-rw-r-- 1 jimmy jimmy   891 Feb  1 18:17  Lab3Q8.c
-rwxrwxr-x 1 jimmy jimmy 16312 Feb  1 18:24  Lab3Q9
-rw-rw-r-- 1 jimmy jimmy   914 Feb  1 18:26  Lab3Q9.c
drwxr-xr-x 2 jimmy jimmy  4096 Oct  6 09:33  Music
drwxr-xr-x 2 jimmy jimmy  4096 Oct  6 09:33  Pictures
drwxr-xr-x 2 jimmy jimmy  4096 Oct  6 09:33  Public
drwx------ 6 jimmy jimmy  4096 Oct 16 15:14  snap
drwxr-xr-x 2 jimmy jimmy  4096 Oct  6 09:33  Templates
-rw-rw-r-- 1 jimmy jimmy   261 Jan 30 17:09 'Untitled Document 1'
drwxr-xr-x 2 jimmy jimmy  4096 Oct  6 09:33  Videos
jimmy@jimmy-VirtualBox:~$
```

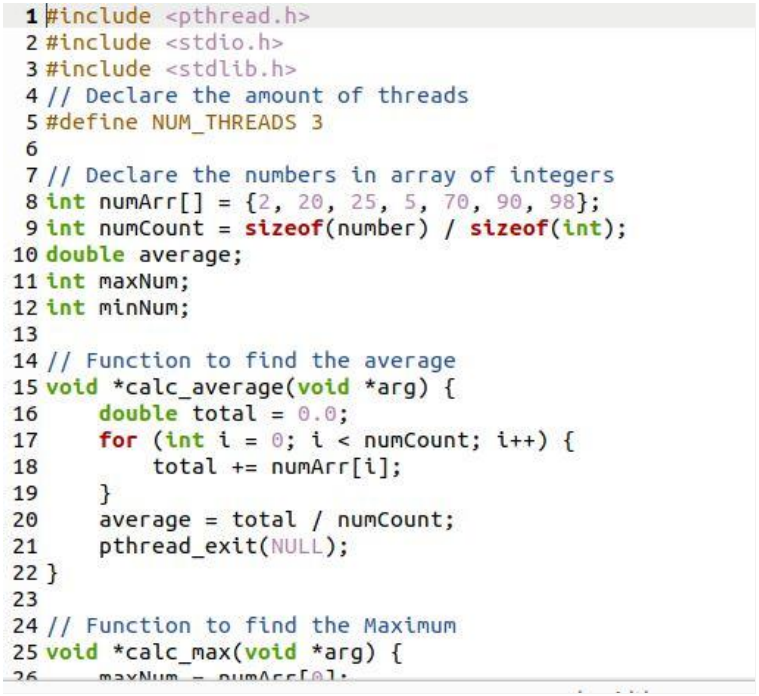

myUbuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Feb 22 13:35
jimmy@jimmy-VirtualBox: ~
Right Ctrl

```
drwx------ 6 jimmy jimmy  4096 Oct 16 15:14  snap
drwxr-xr-x 2 jimmy jimmy  4096 Oct  6 09:33  Templates
-rw-rw-r-- 1 jimmy jimmy   261 Jan 30 17:09 'Untitled Document 1'
drwxr-xr-x 2 jimmy jimmy  4096 Oct  6 09:33  Videos
jimmy@jimmy-VirtualBox:~$ ln -s file.txt softlink.txt
jimmy@jimmy-VirtualBox:~$ ls -l
total 128
drwxr-xr-x 4 jimmy jimmy  4096 Feb 22 13:30  Desktop
drwxr-xr-x 2 jimmy jimmy  4096 Oct  6 09:33  Documents
drwxr-xr-x 2 jimmy jimmy  4096 Oct 16 14:31  Downloads
-rw-rw-r-- 2 jimmy jimmy    16 Feb 22 13:33  file.txt
-rw-rw-r-- 2 jimmy jimmy    16 Feb 22 13:33  hardlink.txt
-rwxrwxr-x 1 jimmy jimmy 16000 Feb  8 17:48  Lab3
-rw-rw-r-- 1 jimmy jimmy   261 Jan 30 17:12  Lab3.c
-rwxrwxr-x 1 jimmy jimmy 16096 Feb  1 18:38  Lab3FinalQ
-rw-rw-r-- 1 jimmy jimmy   398 Feb  1 18:39  Lab3FinalQ.c
-rwxrwxr-x 1 jimmy jimmy 16000 Feb  1 17:55  Lab3Q8
-rw-rw-r-- 1 jimmy jimmy   891 Feb  1 18:17  Lab3Q8.c
-rwxrwxr-x 1 jimmy jimmy 16312 Feb  1 18:24  Lab3Q9
-rw-rw-r-- 1 jimmy jimmy   914 Feb  1 18:26  Lab3Q9.c
drwxr-xr-x 2 jimmy jimmy  4096 Oct  6 09:33  Music
drwxr-xr-x 2 jimmy jimmy  4096 Oct  6 09:33  Pictures
drwxr-xr-x 2 jimmy jimmy  4096 Oct  6 09:33  Public
drwx------ 6 jimmy jimmy  4096 Oct 16 15:14  snap
lrwxrwxrwx 1 jimmy jimmy     8 Feb 22 13:35  softlink.txt -> file.txt
drwxr-xr-x 2 jimmy jimmy  4096 Oct  6 09:33  Templates
-rw-rw-r-- 1 jimmy jimmy   261 Jan 30 17:09 'Untitled Document 1'
drwxr-xr-x 2 jimmy jimmy  4096 Oct  6 09:33  Videos
jimmy@jimmy-VirtualBox:~$
```

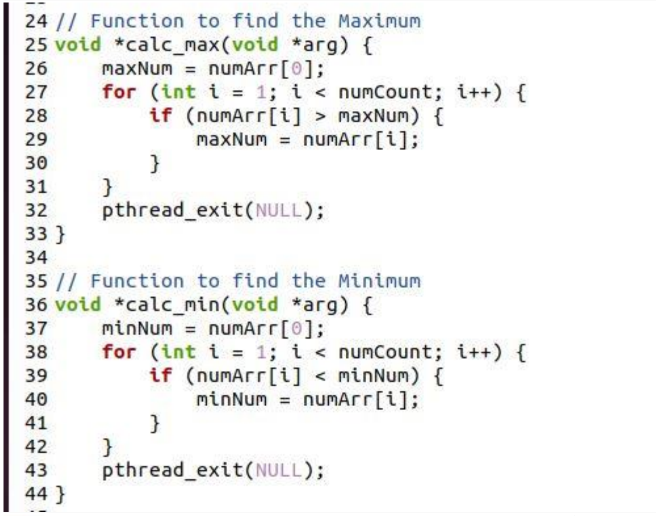
The following commands in the screenshot below will create a soft link between the file.txt file and the softlink.txt file. As seen in the screenshot the make the soft with the commands below. We can verify that the soft link has happened by using the command ls -l and seeing the softlink.txt. As seen in the screenshot below

2. Create a multithreaded program that computes different statistical values for a set of numbers. When given a series of numbers on the command line, this application will start two independent worker threads. One thread will compute the greatest value, and the next will compute the minimum value. Assume your program is given a list of integers. (The array of numbers must be provided as a parameter to the threads, and the thread must return the calculated value to the main thread.)?

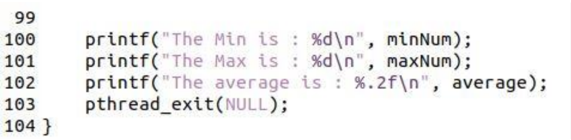
Each of the function handle a thread below and then are joined using a for loop. As seen in the screenshot below is my code for the program with the functions being threaded and the print statement being located in the main and prints out the result of the min and max and average.



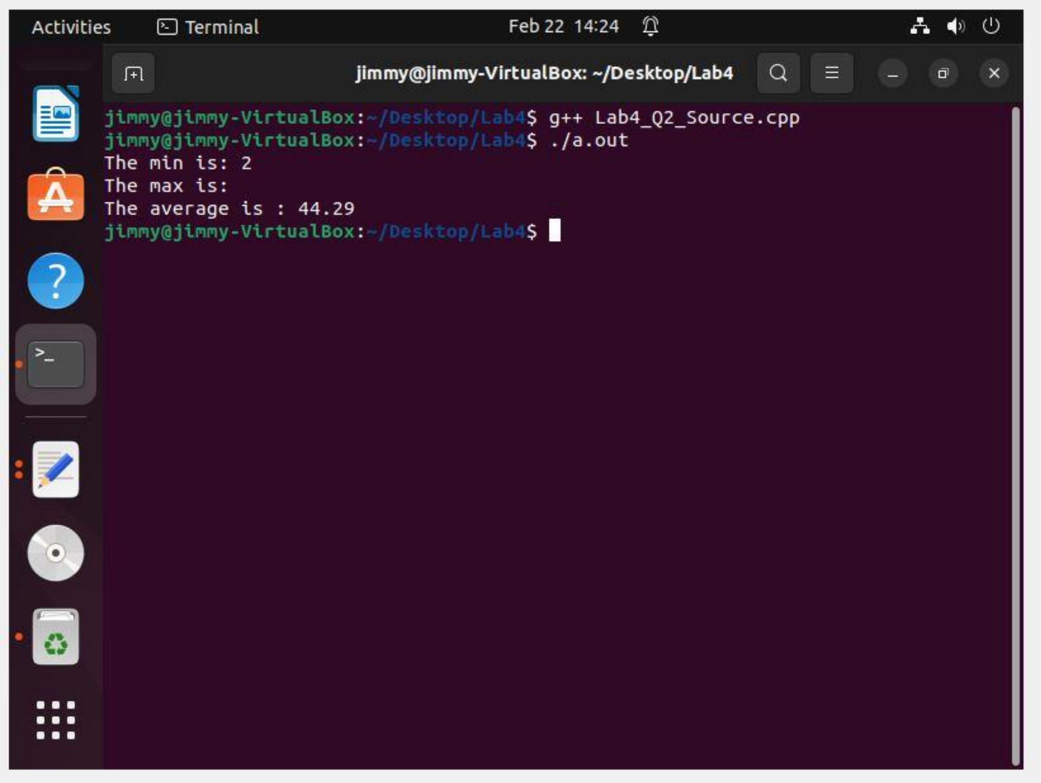
```c
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
// Declare the amount of threads
#define NUM_THREADS 3

// Declare the numbers in array of integers
int numArr[] = {2, 20, 25, 5, 70, 90, 98};
int numCount = sizeof(number) / sizeof(int);
double average;
int maxNum;
int minNum;

// Function to find the average
void *calc_average(void *arg) {
    double total = 0.0;
    for (int i = 0; i < numCount; i++) {
        total += numArr[i];
    }
    average = total / numCount;
    pthread_exit(NULL);
}

// Function to find the Maximum
void *calc_max(void *arg) {
    maxNum = numArr[0];
```



```c
// Function to find the Maximum
void *calc_max(void *arg) {
    maxNum = numArr[0];
    for (int i = 1; i < numCount; i++) {
        if (numArr[i] > maxNum) {
            maxNum = numArr[i];
        }
    }
    pthread_exit(NULL);
}

// Function to find the Minimum
void *calc_min(void *arg) {
    minNum = numArr[0];
    for (int i = 1; i < numCount; i++) {
        if (numArr[i] < minNum) {
            minNum = numArr[i];
        }
    }
    pthread_exit(NULL);
}
```



```c

    printf("The Min is : %d\n", minNum);
    printf("The Max is : %d\n", maxNum);
    printf("The average is : %.2f\n", average);
    pthread_exit(NULL);
}
```

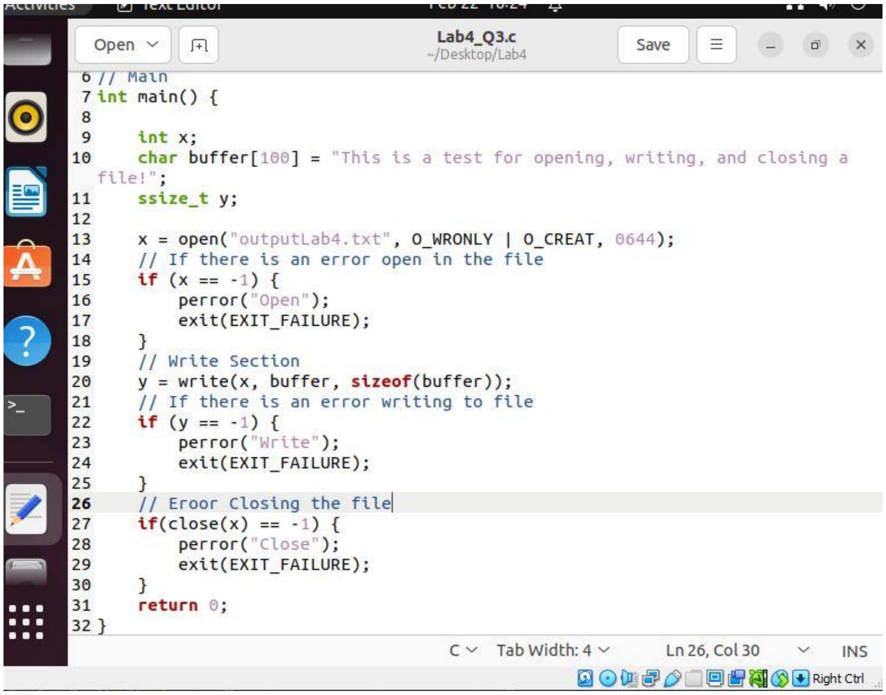


```
jimmy@jimmy-VirtualBox:~/Desktop/Lab4$ g++ Lab4_Q2_Source.cpp
jimmy@jimmy-VirtualBox:~/Desktop/Lab4$ ./a.out
The min is: 2
The max is:
The average is : 44.29
jimmy@jimmy-VirtualBox:~/Desktop/Lab4$ 
```

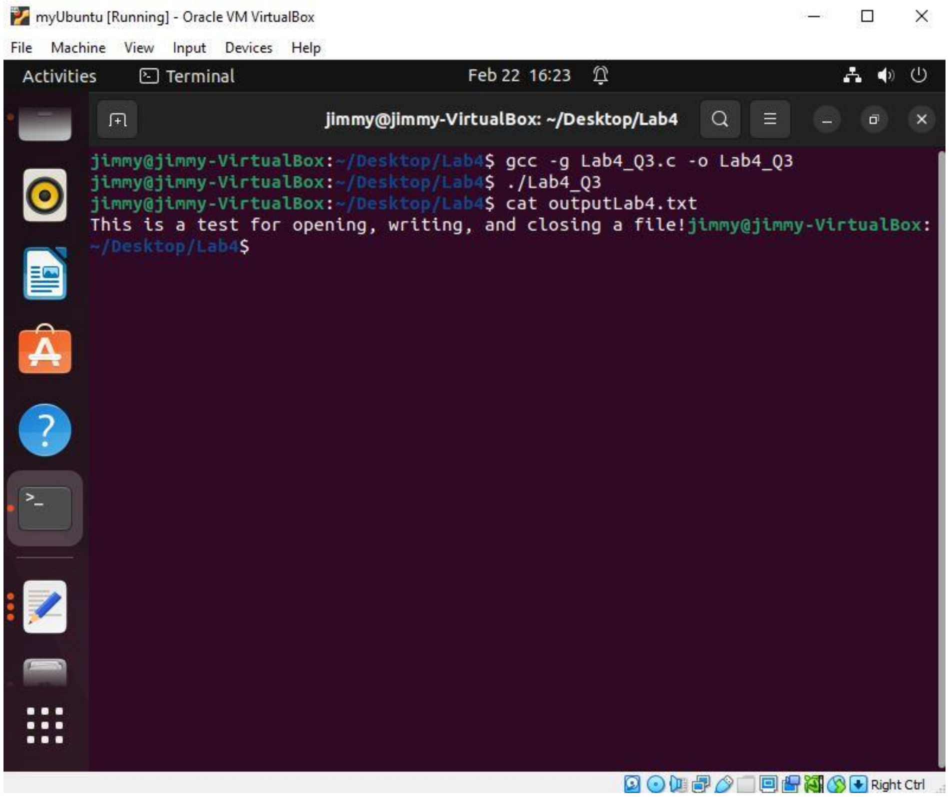
As seen above is a screenshot of the output of the terminal of the min which we can see is 2, followed by the maximum which is 98 and the average of the data set being 44.29 as the average of the data set. This can bee seen in the screenshot above of the terminal.

3. Write a C program that opens the file "outputLab4.txt" for writing and appends the phrase "This is a test for opening, writing, and closing a file!"?

Below is a screenshot of the C program that opens or creates a file called outputLab4.txt that will be opened to write to the phrase “This is a test for opening, writing, and closing a file!”?



```c
// Main
int main() {

    int x;
    char buffer[100] = "This is a test for opening, writing, and closing a file!";
    ssize_t y;

    x = open("outputLab4.txt", O_WRONLY | O_CREAT, 0644);
    // If there is an error open in the file
    if (x == -1) {
        perror("Open");
        exit(EXIT_FAILURE);
    }
    // Write Section
    y = write(x, buffer, sizeof(buffer));
    // If there is an error writing to file
    if (y == -1) {
        perror("Write");
        exit(EXIT_FAILURE);
    }
    // Eroor Closing the file
    if(close(x) == -1) {
        perror("Close");
        exit(EXIT_FAILURE);
    }
    return 0;
}
```

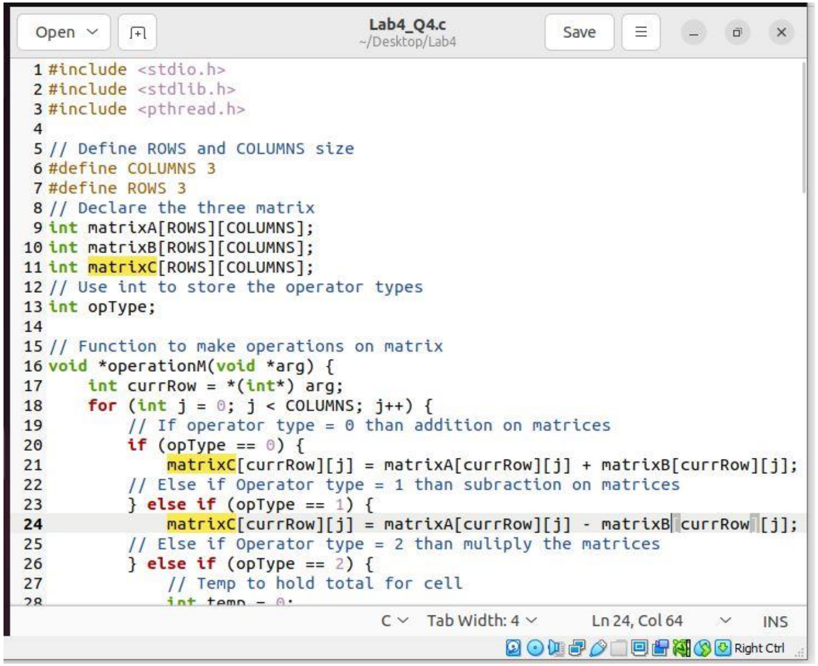

myUbuntu [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Activities
Terminal
Feb 22 16:23
jimmy@jimmy-VirtualBox: ~/Desktop/Lab4
Right Ctrl

```
jimmy@jimmy-VirtualBox:~/Desktop/Lab4$ gcc -g Lab4_Q3.c -o Lab4_Q3
jimmy@jimmy-VirtualBox:~/Desktop/Lab4$ ./Lab4_Q3
jimmy@jimmy-VirtualBox:~/Desktop/Lab4$ cat outputLab4.txt
This is a test for opening, writing, and closing a file!jimmy@jimmy-VirtualBox:
~/Desktop/Lab4$
```

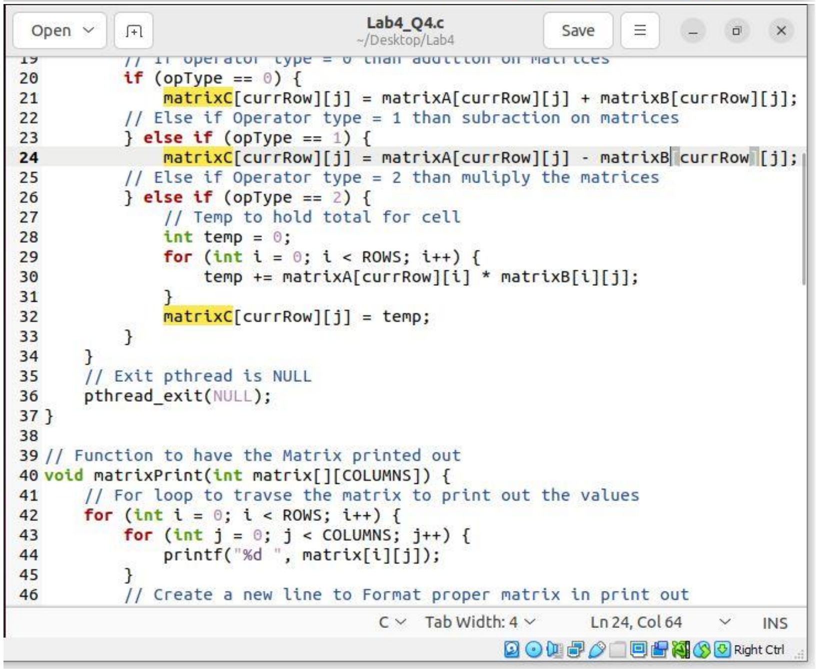
Above is a screenshot of the following code output in the terminal that opens or creates if the file is not existent. The commands above compile and run the code. I use the command cat to see the contents of the outputLab4.txt to ensure that the program is working correctly.

4. Write a program for matrix addition, subtraction and multiplication using multithreading?

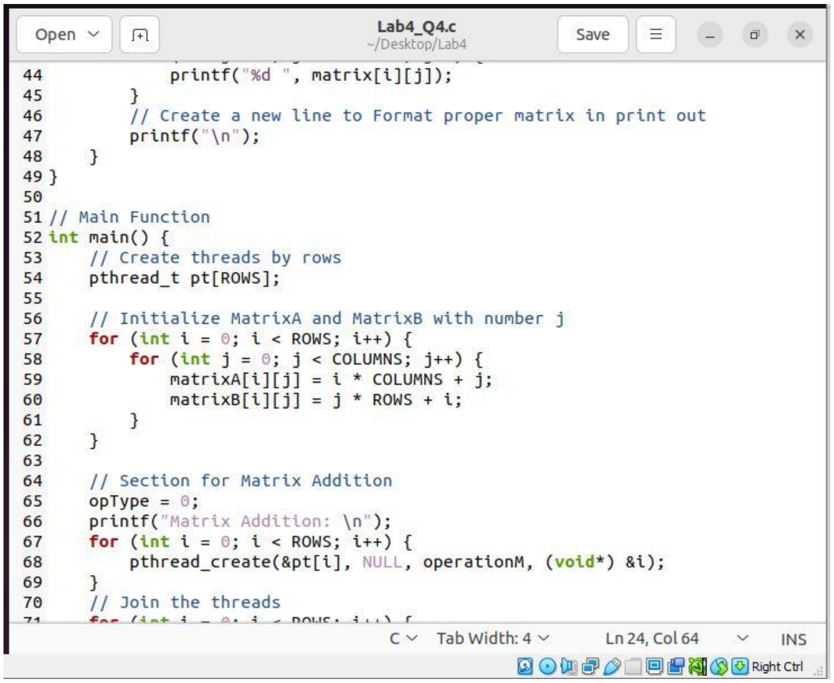
Below is a screenshot of the C program that will take a matrix that is 3x3 and compute the addition, subtraction and multiplication using multithreading. I use a int to know what case to thread by setting up the variable opType as an integer. If opType is equal to 0 than it is addition, if opType is equal to 1, than it is subtraction, if opType is equal to 2 than it is multiplication.



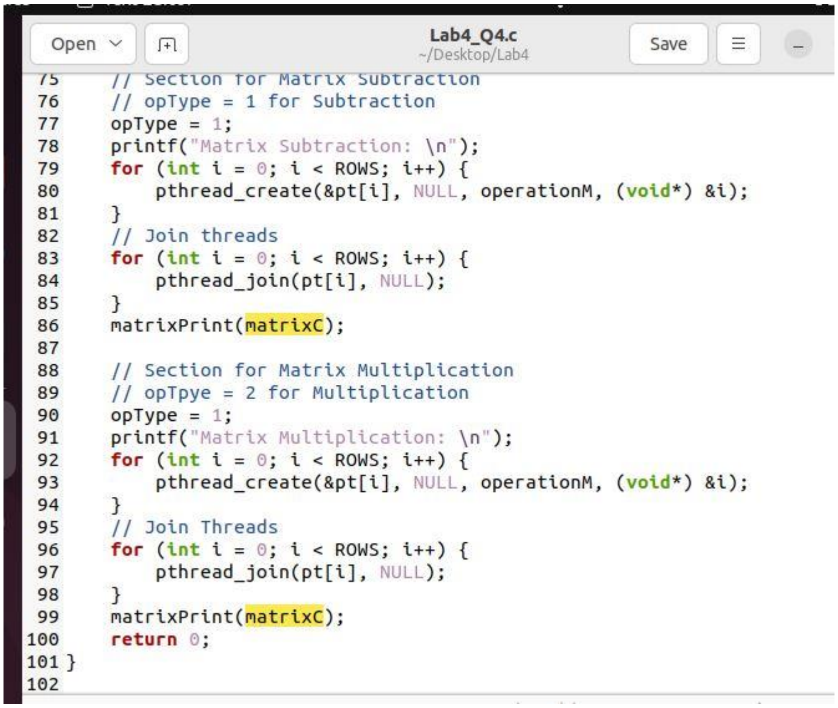
```c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

// Define ROWS and COLUMNS size
#define COLUMNS 3
#define ROWS 3
// Declare the three matrix
int matrixA[ROWS][COLUMNS];
int matrixB[ROWS][COLUMNS];
int matrixC[ROWS][COLUMNS];
// Use int to store the operator types
int opType;

// Function to make operations on matrix
void *operationM(void *arg) {
    int currRow = *(int*) arg;
    for (int j = 0; j < COLUMNS; j++) {
        // If operator type = 0 than addition on matrices
        if (opType == 0) {
            matrixC[currRow][j] = matrixA[currRow][j] + matrixB[currRow][j];
        // Else if Operator type = 1 than subraction on matrices
        } else if (opType == 1) {
            matrixC[currRow][j] = matrixA[currRow][j] - matrixB[currRow][j];
        // Else if Operator type = 2 than muliply the matrices
        } else if (opType == 2) {
            // Temp to hold total for cell
            int temp = 0;
```



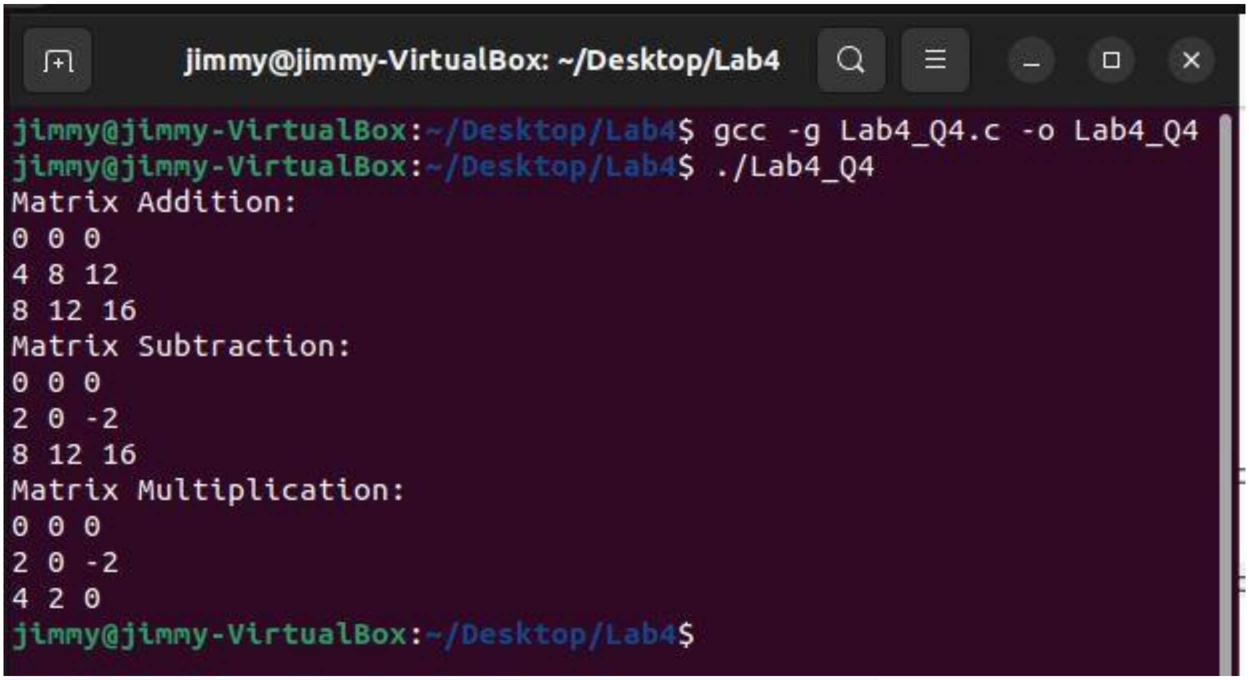
```c
        // If operator type = 0 than addition on matrices
        if (opType == 0) {
            matrixC[currRow][j] = matrixA[currRow][j] + matrixB[currRow][j];
        // Else if Operator type = 1 than subraction on matrices
        } else if (opType == 1) {
            matrixC[currRow][j] = matrixA[currRow][j] - matrixB[currRow][j];
        // Else if Operator type = 2 than muliply the matrices
        } else if (opType == 2) {
            // Temp to hold total for cell
            int temp = 0;
            for (int i = 0; i < ROWS; i++) {
                temp += matrixA[currRow][i] * matrixB[i][j];
            }
            matrixC[currRow][j] = temp;
        }
    }
    // Exit pthread is NULL
    pthread_exit(NULL);
}

// Function to have the Matrix printed out
void matrixPrint(int matrix[][COLUMNS]) {
    // For loop to travse the matrix to print out the values
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLUMNS; j++) {
            printf("%d ", matrix[i][j]);
        }
        // Create a new line to Format proper matrix in print out
```



```
            printf("%d ", matrix[i][j]);
        }
        // Create a new line to Format proper matrix in print out
        printf("\n");
    }
}

// Main Function
int main() {
    // Create threads by rows
    pthread_t pt[ROWS];

    // Initialize MatrixA and MatrixB with number j
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLUMNS; j++) {
            matrixA[i][j] = i * COLUMNS + j;
            matrixB[i][j] = j * ROWS + i;
        }
    }

    // Section for Matrix Addition
    opType = 0;
    printf("Matrix Addition: \n");
    for (int i = 0; i < ROWS; i++) {
        pthread_create(&pt[i], NULL, operationM, (void*) &i);
    }
    // Join the threads
    for (int i = 0; i < ROWS; i++) {
```



```
    // Section for Matrix Subtraction
    // opType = 1 for Subtraction
    opType = 1;
    printf("Matrix Subtraction: \n");
    for (int i = 0; i < ROWS; i++) {
        pthread_create(&pt[i], NULL, operationM, (void*) &i);
    }
    // Join threads
    for (int i = 0; i < ROWS; i++) {
        pthread_join(pt[i], NULL);
    }
    matrixPrint(matrixC);

    // Section for Matrix Multiplication
    // opTpye = 2 for Multiplication
    opType = 1;
    printf("Matrix Multiplication: \n");
    for (int i = 0; i < ROWS; i++) {
        pthread_create(&pt[i], NULL, operationM, (void*) &i);
    }
    // Join Threads
    for (int i = 0; i < ROWS; i++) {
        pthread_join(pt[i], NULL);
    }
    matrixPrint(matrixC);
    return 0;
}
```



```
jimmy@jimmy-VirtualBox:~/Desktop/Lab4$ gcc -g Lab4_Q4.c -o Lab4_Q4
jimmy@jimmy-VirtualBox:~/Desktop/Lab4$ ./Lab4_Q4
Matrix Addition:
0 0 0
4 8 12
8 12 16
Matrix Subtraction:
0 0 0
2 0 -2
8 12 16
Matrix Multiplication:
0 0 0
2 0 -2
4 2 0
jimmy@jimmy-VirtualBox:~/Desktop/Lab4$
```

Above is a screenshot of my terminal that prints out the matrixes for addition, followed by subtraction, and concluded by the multiplication matrix. As seen above we can see that the code is correct for addition, subtraction, and multiplication of matrices.

https://github.com/JimmyBattis/CS-OS-470-Labs