OPERATING SYSTEMS
Assignment 1 Report
Threads assignment

Name: **Mostafa Nabil Mohamed Sayed Ahmed.**

ID: **54**

Problem Statement:

It required to implement two popular algorithms as multi-threaded ones 1) Matrix Multiplication 2) Merge Sort

Objectives:

1. Introducing threads concepts and POSIX threads library.
2. Implementing popular algorithms as multi-threaded ones.

Main Functions Of Merge sort:

Rread_Array:

In this method we read the input array from file “merge-sort-readme.txt” and put the values in the array.

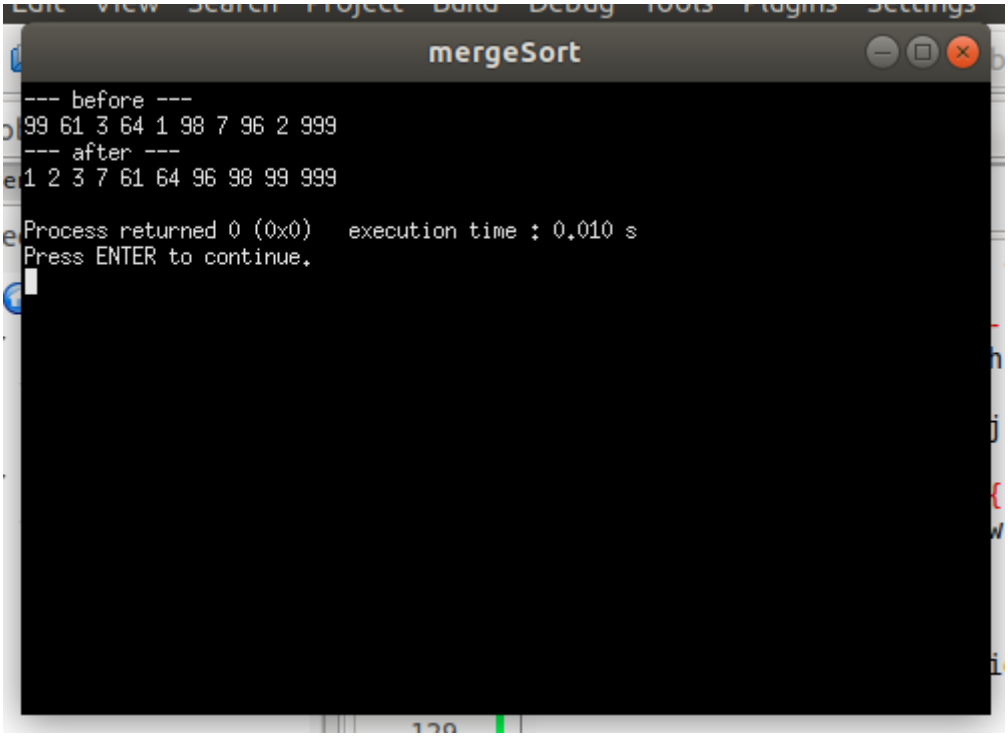
Merge_sort:

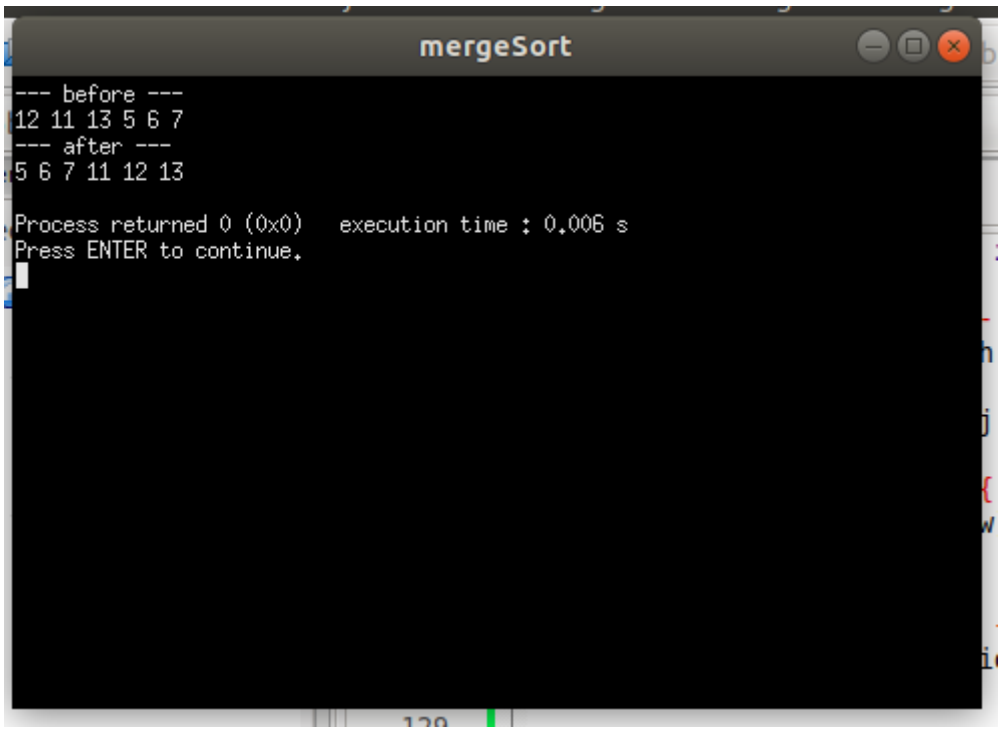
we have two functions with the same name one for the main call and the other to be called by the thread as it must be of the type (void *).
In this function we split the array into 2 halves and call the function again using threads until its only one element in the array then we merge them togeather.

Merge:

In this method we merge the two arrays we have in sorted order.

Sample Runs:


mergeSort
--- before ---
99 61 3 64 1 98 7 96 2 999
--- after ---
1 2 3 7 61 64 96 98 99 999
Process returned 0 (0x0) execution time : 0.010 s
Press ENTER to continue.


mergeSort
--- before ---
12 11 13 5 6 7
--- after ---
5 6 7 11 12 13
Process returned 0 (0x0) execution time : 0.006 s
Press ENTER to continue.

Main Functions Of Matrix multiplication:

Rread_matrix:

In this method we read the input matrices from file "Matrix-readme.txt" and put the values in matrix A and B.

matrix_multiplication_row:

In this method we create threads to calculate the value of the multiplication of each row and calculate time after it finish.

multiply_row:

In this method we calculate the multiplication of the specified row in a single thread.

matrix_multiplication_element:

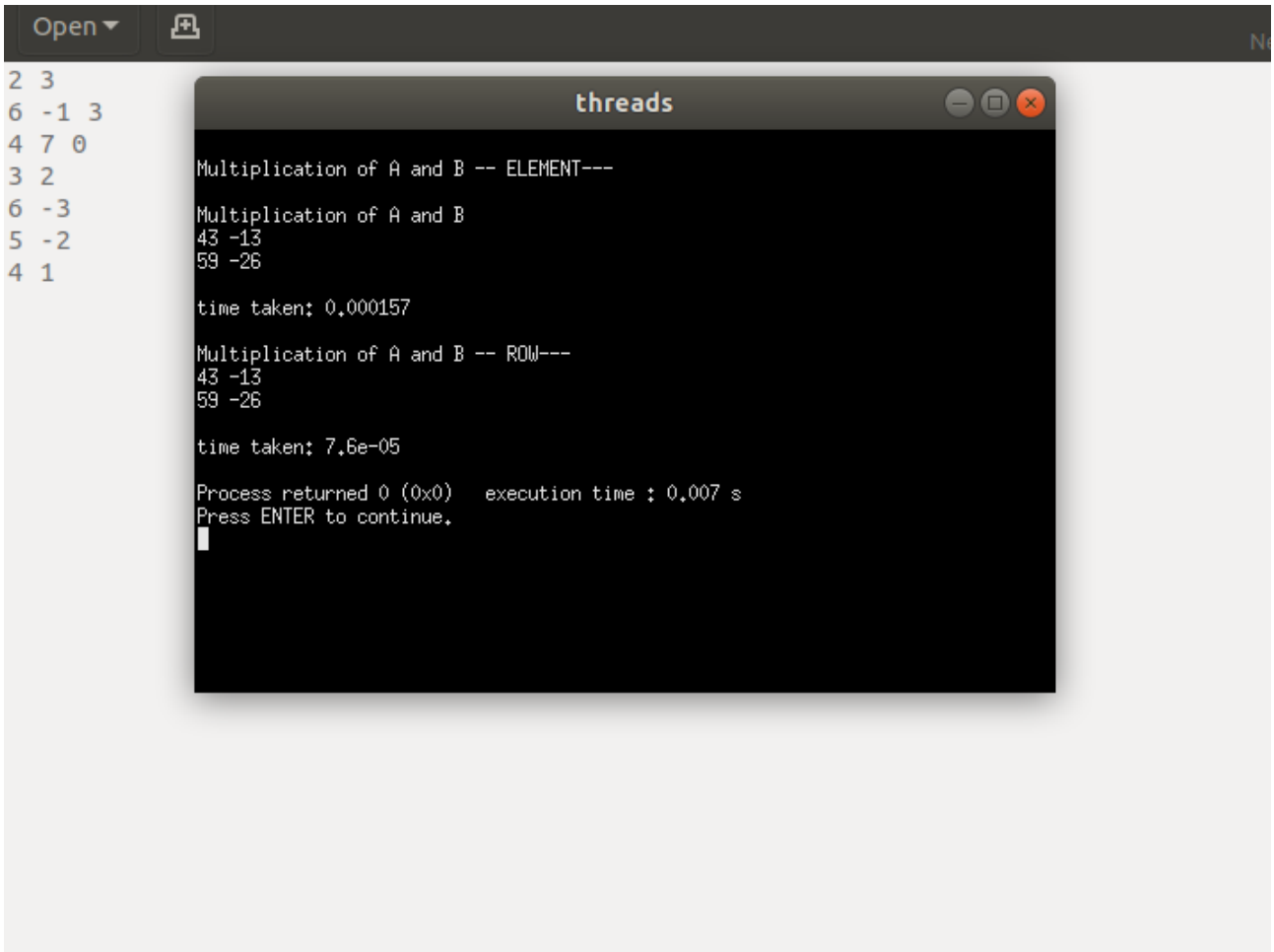
In this method we create threads to calculate the value of the multiplication of each element and calculate time after it finish.

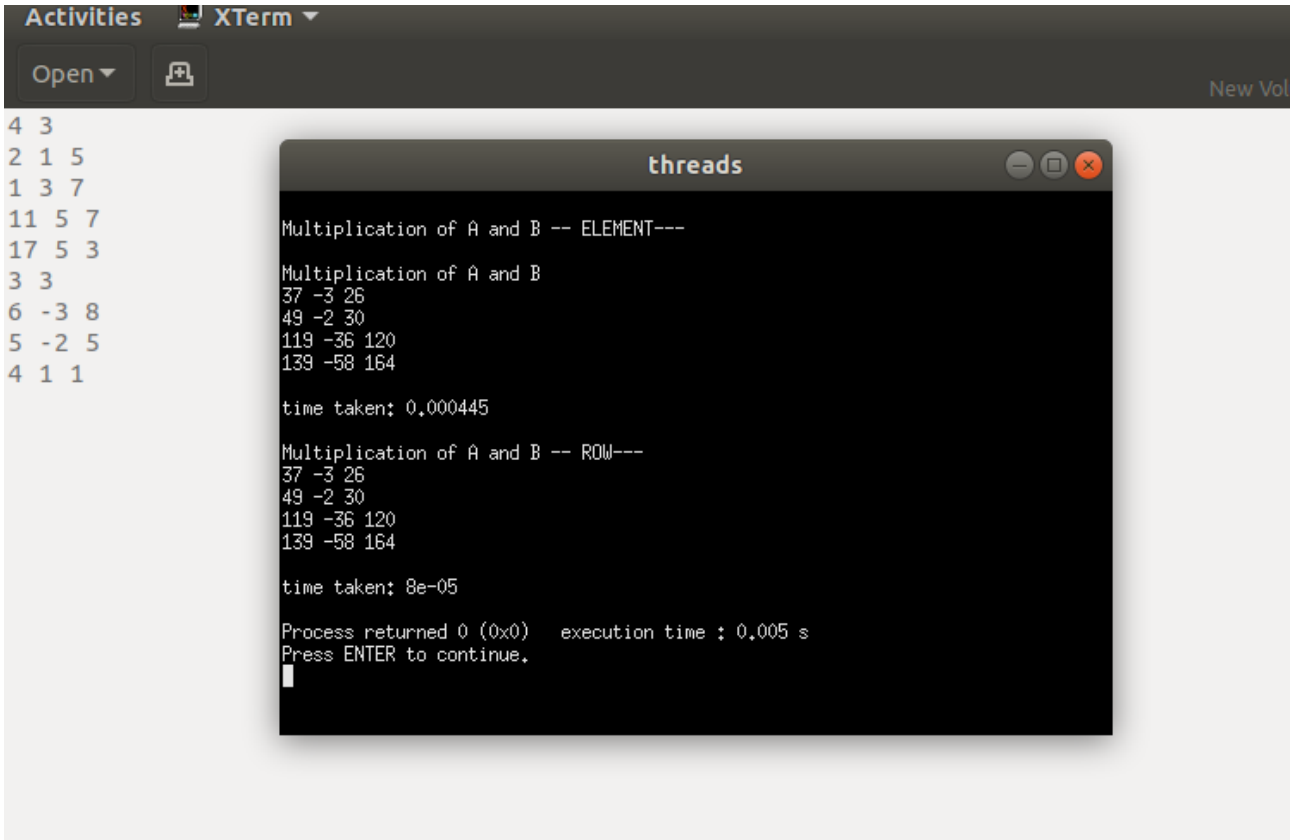
multiply_element:

In this method we calculate the multiplication of the specified element in a single thread.

→ **The time that the multiply_row takes is less than the time that the multiply_element takes because the threads that works in parallel is calculating nested for loop. But, in multiply_element the nested loop is in the main thread waiting much more time than being in parallel .**

Sample Runs:


Open
2 3
6 -1 3
4 7 0
3 2
6 -3
5 -2
4 1
threads
Multiplication of A and B -- ELEMENT---
Multiplication of A and B
43 -13
59 -26
time taken: 0.000157
Multiplication of A and B -- ROW---
43 -13
59 -26
time taken: 7.6e-05
Process returned 0 (0x0) execution time : 0.007 s
Press ENTER to continue.


Activities
XTerm
Open
4 3
2 1 5
1 3 7
11 5 7
17 5 3
3 3
6 -3 8
5 -2 5
4 1 1
threads
Multiplication of A and B -- ELEMENT---
Multiplication of A and B
37 -3 26
49 -2 30
119 -36 120
139 -58 164
time taken: 0.000445
Multiplication of A and B -- ROW---
37 -3 26
49 -2 30
119 -36 120
139 -58 164
time taken: 8e-05
Process returned 0 (0x0) execution time : 0.005 s
Press ENTER to continue.