

THREADS

NAME Salah El-Din Ahmed Muhammed El-Sayed Eltenehy.

ID 19015854.

- Code organization:
 - Code mainly consists of three parts.
 - Each part for one of algorithms.
 - Thread per matrix:
 - It is forward calculate the value and store it directly in the file.
 - Thread per Row:
 - Calculate each raw and save it within the right index in a global matrix to save it later in the file.
 - Thread per element:
 - Calculate the element and send the indexes to store it correctly in the global matrix and save it later in the file.
- Main functions:
 - Thread per matrix:
 - Takes two arguments “array1 & array2”.
 - Loop iterates for all elements.
 - Call another function “getColumn” and send array2 to it to return the required column.
 - Store the value in the file and repeat the previous steps.
 - Thread per row:
 - It creates n threads “where n the first dimension of first array”
 - Each thread takes raw from the first array and the second array.
 - Calculate the result raw and save it in the global array.
 - After finishes all calculations save the global array.
 - Thread per element:
 - Same as thread per row except number of threads.
 - It creates n threads “n \rightarrow first dimn * second dimn”

- Each threads takes raw and column and calculate the result.
- Compile and run the code:
 - Open the terminal and change directory to the project file.
 - Write “make” command to build the code.
 - Write “./matMultp” command to run the code.
 - The default names for files “a b c”.
 - You can change them. e.g.”./matMultp x y z”
 - Where:
 - X the name of first file.
 - Y the name of second file.
 - Z the name of output files.
- Sample run:

```

salah@salah: /media/salah/01D769BFB6545890/College/Year projects/Semester 2/OS/Threads/Threads
gcc -pthread main.c -o matMultp
main.c:13:1: warning: useless storage class specifier in empty declaration
13 | };
    | ^
main.c: In function 'threadPerMatrix':
main.c:66:5: warning: implicit declaration of function 'strcat' [-Wimplicit-function-declaration]
66 |     strcat(fileName, "_per_Matrix.txt");
    |     ^~~~~~
main.c:66:5: warning: incompatible implicit declaration of built-in function 'strcat'
main.c:6:1: note: include <string.h> or provide a declaration of 'strcat'
5 | #include <sys/time.h>
+++ |+#include <string.h>
6 | typedef struct thread_data{
main.c: In function 'saveDataForRawAlgorithm':
main.c:128:5: warning: incompatible implicit declaration of built-in function 'strcat'
128 |     strcat(fileName, "_per_Raw.txt");
    |     ^~~~~~
main.c:128:5: note: include <string.h> or provide a declaration of 'strcat'
main.c: In function 'threadFunctionTwo':
main.c:181:5: warning: implicit declaration of function 'memset' [-Wimplicit-function-declaration]
181 |     memset(tempRaw, 0, rowsAndCols1[1]);
    |     ^~~~~~
main.c:181:5: warning: incompatible implicit declaration of built-in function 'memset'
main.c:181:5: note: include <string.h> or provide a declaration of 'memset'
main.c: In function 'threadPerRaw':
main.c:197:51: warning: format '%ld' expects argument of type 'long int', but argument 2 has type 'int' [-Wformat=]
197 |     printf("Error while creating thread %ld\n", i);
    |                                           ~~~~^
    |                                           |
    |                                           int
    |                                           long int
    |                                           %d
main.c: In function 'main':
main.c:267:22: warning: passing argument 1 of 'threadPerElement' from incompatible pointer type [-Wincompatible-pointer-types]
267 |     threadPerElement(data1, data2, output);
    |                      ^~~~~~
  
```

```
Activities Terminal 17:22 مارس 31
salah@salah: /media/salah/01D769BFB6545890/College/Year projects/Semester 2/OS/Threads/Threads

main.c:460:5: note: include '<string.h>' or provide a declaration of 'strcat'
main.c:462:5: warning: incompatible implicit declaration of built-in function 'memset'
462 |     memset(fileName, 0, 25);
    |     ^~~~~~
main.c:462:5: note: include '<string.h>' or provide a declaration of 'memset'
salah@salah: /media/salah/01D769BFB6545890/College/Year projects/Semester 2/OS/Threads/Threads$ ./matMultp
Input matrix 1: 5 x 4
1 2 3 4
6 7 8 9
11 12 13 14
16 17 18 19
1 2 3 4
-----THREADS-----
Input matrix 2: 4 x 4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
-----THREADS-----
Thread per row
Seconds taken 0
Microseconds taken: 409
-----THREADS-----
Thread per matrix
Seconds taken 0
Microseconds taken: 347
-----THREADS-----
Thread per element
Seconds taken 0
Microseconds taken: 1645
-----THREADS-----
Output matrix: 5 x 4
90 100 110 120
230 260 290 320
370 420 470 520
510 580 650 720
90 100 110 120
salah@salah: /media/salah/01D769BFB6545890/College/Year projects/Semester 2/OS/Threads/Threads$ ./matMultp x y z
```

```
Activities Terminal 17:23 مارس 31
salah@salah: /media/salah/01D769BFB6545890/College/Year projects/Semester 2/OS/Threads/Threads

510 580 650 720
90 100 110 120
salah@salah: /media/salah/01D769BFB6545890/College/Year projects/Semester 2/OS/Threads/Threads$ ./matMultp x y z
heree x
Input matrix 1: 5 x 4
1 2 3 4
6 7 8 9
11 12 13 14
16 17 18 19
1 2 3 4
-----THREADS-----
Input matrix 2: 4 x 4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
-----THREADS-----
Thread per row
Seconds taken 0
Microseconds taken: 731
-----THREADS-----
Thread per matrix
Seconds taken 0
Microseconds taken: 644
-----THREADS-----
Thread per element
Seconds taken 0
Microseconds taken: 1848
-----THREADS-----
Output matrix: 5 x 4
90 100 110 120
230 260 290 320
370 420 470 520
510 580 650 720
90 100 110 120
salah@salah: /media/salah/01D769BFB6545890/College/Year projects/Semester 2/OS/Threads/Threads$ ./matMultp A B C
heree A
Input matrix 1: 3 x 4
1 2 3 4
```

```
Activities Terminal 17:23 مارس 31
salah@salah: /media/salah/01D769BFB6545890/College/Year projects/Semester 2/OS/Threads/Threads

Microseconds taken: 1848
-----THREADS-----
Output matrix: 5 x 4
90 100 110 120
230 260 290 320
370 420 470 520
510 580 650 720
90 100 110 120
salah@salah: /media/salah/01D769BFB6545890/College/Year projects/Semester 2/OS/Threads/Threads$ ./matMultp A B C
heree A
Input matrix 1: 3 x 4
1 -2 3 4
1 2 -3 4
-1 2 3 4
-----THREADS-----
Input matrix 2: 4 x 3
-1 2 3
1 -2 3
1 2 -3
1 2 3
-----THREADS-----
Thread per raw
Seconds taken 0
Microseconds taken: 1626
-----THREADS-----
Thread per matrix
Seconds taken 0
Microseconds taken: 633
-----THREADS-----
Thread per element
Seconds taken 0
Microseconds taken: 692
-----THREADS-----
Output matrix: 3 x 3
4 20 0
2 0 30
10 8 6
salah@salah: /media/salah/01D769BFB6545890/College/Year projects/Semester 2/OS/Threads/Threads$
```

- Comparison:
 - As shown in the pictures the thread per matrix takes the minimum time.
 - It means that when we create many threads they take long time.
 - As a result the thread per element is slowest algorithm.