GTU

DEPARTMENT OF COMPUTER ENGINEERING

CSE 344 – Spring 2022

HOMEWORK 5 REPORT

SÜLEYMAN GÖLBOL

1801042656

1. REQUIREMENTS

NONFUNCTIONAL REQUIREMENTS

1. Portability → The application should be portable. All computers that have Linux Distro with POSIX and GCC compiler can run the program.

2. Maintainability → In case of an error occurrence, the system uses perror in order to give feedback on terminal.

3. Performance → The system should initially be able to process as many entries as possible. Each request must be processed with different terminals. The system's performance should be fast enough to show user the feedback.

FUNCTIONAL REQUIREMENTS

In order to compile the program, user have to use "make" command that uses gcc. If make or gcc is not installed user can install it via *"sudo apt-get install build-essential"* command.
Make command runs 1 command.

```
gcc -Wall *.c -lm -o hw5
```

To run, we need to use command line arguments with parameters i, j, o,n and m.

```
./hw5 -i files/file1.txt -j files/file2.txt -o output.csv
-n 4 -m 2
```

If input file exists and we have permissions to read the input file and write to the output file, the executable will run successfully.

To run, also size of each input file should be less than $(2^n)^{2.}$ Because row and column size of matrix will be 2^n. Because there are m threads, and each thread is responsible for 2^n / m columns of matrix.

So also, m should be less than 2^n. When these rules are obtained, it should be fine.

2. PROBLEM SOLUTION APPROACH AND REQUIREMENT ACHIEVEMENT

Firstly, the first big problem for to get synchronization between part1 and part2. So, I implemented a barrier. To implement it I used both mutex and conditional variable which both of them are available on pthread library.

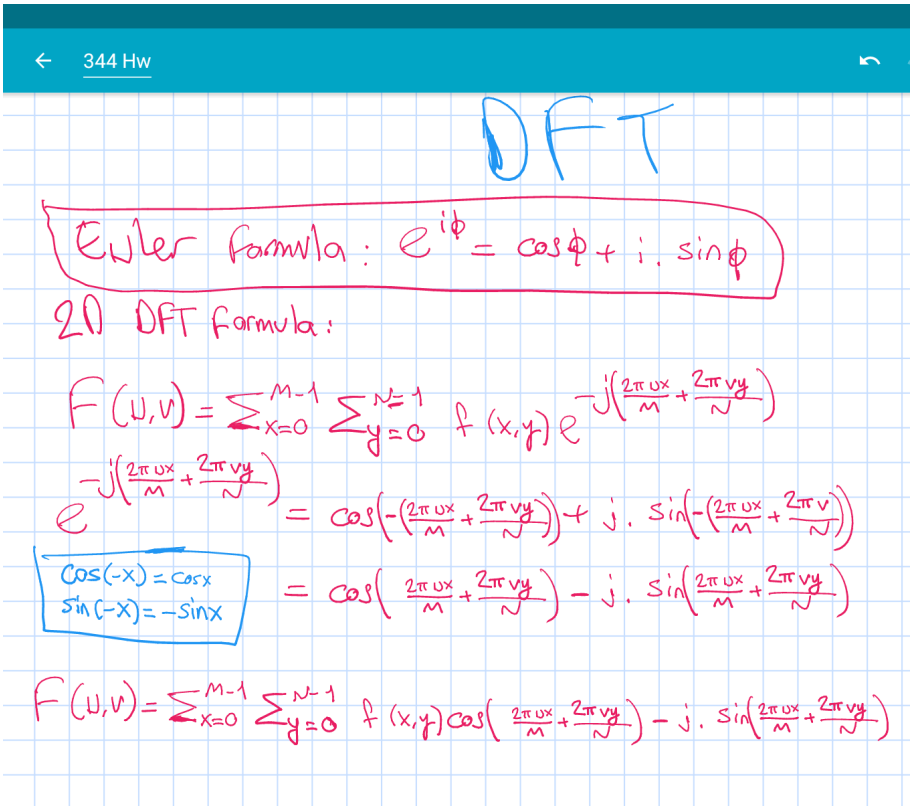
Then other problem of mine was to send more than 1 variable to thread. So I've created a struct which contains thread id index, size, number of columns to calculate.

Another problem that I have was freeing allocated memory. But I got double free error and I fixed it.

The most difficult problem that I've had was to calculate discrete fourier transform because the resources on internet are not helpful enough (for a beginner). Until last week, I didn't know about that topic.

Finally I solved it using complex numbers and so on.

This is the formula that I've used.



Last problem I encountered was printing timestamp with formatted output with same print function. For this I created a function called tprintf and to merge timestampt with formatted string I used snprintf. Then with variadic function I used variadic list and vprintf to print formatted text.

3) TEST CASES AND RESULTS

3.a) **`./hw5 -i files/file1.txt -j files/dfdffhfj.txt -o output -n 4 -m 17`**

```
sglbl@SglblPC:~/gtu/hw5$ ./hw5 -i files/file1.txt -j files/fifdfjd.txt -o output -n 4 -m 17
Error while opening the file.
```

If user tries to enter a non-existing file, it gives error.

3.b) **`./hw5 -i files/file1.txt -j files/file2.txt -o output -n 8 -m 2`**

```
sglbl@SglblPC:~/gtu/hw5$ ./hw5 -i files/file1.txt -j files/file2.txt -o output -n 8 -m 2
Error. Size of file is less than (2^n)*(2^n)
```

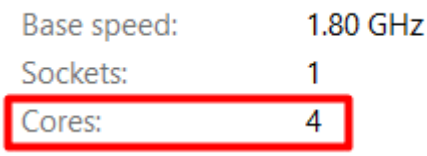
If user tries to put n as big number and if file size is less than $(2^n)^2$, it gives error.

3.c) **`./hw5 -i files/file1.txt -j files/file2.txt -o output -n 4 -m 2`**

```
sglbl@sglblPC:~/Desktop/CSE344/hw5$ ./hw5 -i files/file1.txt -j files/file2.txt -o output -n 4 -m 2
(Sat May 21 14:12:02 2022) Two matrices of size 16x16 have been read. The number of threads is 2
(Sat May 21 14:12:02 2022) Thread 1 has reached the rendezvous point in 0.00068 seconds
(Sat May 21 14:12:02 2022) Thread 0 has reached the rendezvous point in 0.00153 seconds
(Sat May 21 14:12:02 2022) Thread 0 is advancing to the second part
(Sat May 21 14:12:02 2022) Thread 1 is advancing to the second part
(Sat May 21 14:12:02 2022) Thread 0 has finished the second part in 0.036 seconds.
(Sat May 21 14:12:02 2022) Thread 1 has finished the second part in 0.036 seconds.
(Sat May 21 14:12:02 2022) The process has written the output file. The total time spent is 0.041 seconds.
sglbl@sglblPC:~/Desktop/CSE344/hw5$
```

RESULTS WITH DIFFERENT INPUTS

Number of cores in my pc is 4 cores.



SAME FILES DIFFERENT NUMBER OF THREADS

```
sglbl@sglblPC:~/Desktop/CSE344/hw5$ ./hw5 -i files/file1.txt -j files/file2.txt -o output -n 4 -m 2
(Sat May 21 14:12:02 2022) Two matrices of size 16x16 have been read. The number of threads is 2
(Sat May 21 14:12:02 2022) Thread 1 has reached the rendezvous point in 0.00068 seconds
(Sat May 21 14:12:02 2022) Thread 0 has reached the rendezvous point in 0.00153 seconds
(Sat May 21 14:12:02 2022) Thread 0 is advancing to the second part
(Sat May 21 14:12:02 2022) Thread 1 is advancing to the second part
(Sat May 21 14:12:02 2022) Thread 0 has finished the second part in 0.036 seconds.
(Sat May 21 14:12:02 2022) Thread 1 has finished the second part in 0.036 seconds.
(Sat May 21 14:12:02 2022) The process has written the output file. The total time spent is 0.041 seconds.
sglbl@sglblPC:~/Desktop/CSE344/hw5$
```

```
sglbl@sglblPC:~/Desktop/CSE344/hw5$ ./hw5 -i files/file1.txt -j files/file2.txt -o output -n 3 -m 4
(Sat May 21 14:13:10 2022) Two matrices of size 8x8 have been read. The number of threads is 4
(Sat May 21 14:13:10 2022) Thread 0 has reached the rendezvous point in 0.00011 seconds
(Sat May 21 14:13:10 2022) Thread 1 has reached the rendezvous point in 0.00009 seconds
(Sat May 21 14:13:10 2022) Thread 3 has reached the rendezvous point in 0.00002 seconds
(Sat May 21 14:13:10 2022) Thread 2 has reached the rendezvous point in 0.00002 seconds
(Sat May 21 14:13:10 2022) Thread 2 is advancing to the second part
(Sat May 21 14:13:10 2022) Thread 3 is advancing to the second part
(Sat May 21 14:13:10 2022) Thread 0 is advancing to the second part
(Sat May 21 14:13:10 2022) Thread 1 is advancing to the second part
(Sat May 21 14:13:10 2022) Thread 0 has finished the second part in 0.010 seconds.
(Sat May 21 14:13:10 2022) Thread 2 has finished the second part in 0.010 seconds.
(Sat May 21 14:13:10 2022) Thread 1 has finished the second part in 0.011 seconds.
(Sat May 21 14:13:10 2022) Thread 3 has finished the second part in 0.011 seconds.
(Sat May 21 14:13:10 2022) The process has written the output file. The total time spent is 0.016 seconds.
sglbl@sglblPC:~/Desktop/CSE344/hw5$
```

I made thread size 2 times bigger, and I made n = 3 instead of 4. Now it's much faster.

In the first one, because of thread size is 2, each thread was calculating 2^4/2 = 8 columns.

In the second one, because of thread size is 4 and n is 3, each thread is calculating 2^3/4 = 2 columns.

ANOTHER EXAMPLE

```
sglbl@sglblPC:~/Desktop/CSE344/hw5$ ./hw5 -i files/file1.txt -j files/file2.txt -o output -n 3 -m 2
(Sat May 21 14:19:25 2022) Two matrices of size 8x8 have been read. The number of threads is 2
(Sat May 21 14:19:25 2022) Thread 0 has reached the rendezvous point in 0.00005 seconds
(Sat May 21 14:19:25 2022) Thread 1 has reached the rendezvous point in 0.00003 seconds
(Sat May 21 14:19:25 2022) Thread 1 is advancing to the second part
(Sat May 21 14:19:25 2022) Thread 0 is advancing to the second part
(Sat May 21 14:19:25 2022) Thread 1 has finished the second part in 0.004 seconds.
(Sat May 21 14:19:25 2022) Thread 0 has finished the second part in 0.005 seconds.
(Sat May 21 14:19:25 2022) The process has written the output file. The total time spent is 0.009 seconds.
sglbl@sglblPC:~/Desktop/CSE344/hw5$
```

```
sglbl@sglblPC:~/Desktop/CSE344/hw5$ ./hw5 -i files/file1.txt -j files/file2.txt -o output -n 4 -m 2
(Sat May 21 14:19:51 2022) Two matrices of size 16x16 have been read. The number of threads is 2
(Sat May 21 14:19:51 2022) Thread 0 has reached the rendezvous point in 0.00017 seconds
(Sat May 21 14:19:51 2022) Thread 1 has reached the rendezvous point in 0.00022 seconds
(Sat May 21 14:19:51 2022) Thread 1 is advancing to the second part
(Sat May 21 14:19:51 2022) Thread 0 is advancing to the second part
(Sat May 21 14:19:51 2022) Thread 1 has finished the second part in 0.041 seconds.
(Sat May 21 14:19:51 2022) Thread 0 has finished the second part in 0.042 seconds.
(Sat May 21 14:19:51 2022) The process has written the output file. The total time spent is 0.044 seconds.
sglbl@sglblPC:~/Desktop/CSE344/hw5$
```

This time number of threads are same. But n is different so In the first one it calculates 2^3/2 = 4 columns per each thread. But in the second it's 2^4/2 = 8 columns per each thread. So it's much slower in the second.

VALGRIND MEMORY RESULTS

The output from valgrind about heap and leaks is like below:

```
==27006==
==27006== HEAP SUMMARY:
==27006==     in use at exit: 0 bytes in 0 blocks
==27006==   total heap usage: 126 allocs, 126 frees, 18,221 bytes allocated
==27006==
==27006== All heap blocks were freed -- no leaks are possible
==27006==
==27006== For lists of detected and suppressed errors, rerun with: -s
==27006== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
sglbl@SglblPC:~/gtu/hw5$
```

CHECKING FOR ZOMBIE PROCESSES ETC.

```
sglbl@SglblPC:~/gtu/hw5$ ps aux | grep hw5
sglbl    27235  0.0  0.0   8164   656 pts/3    S+   17:00   0:00 grep --color=auto hw5
```