Bilkent University
Computer Engineering

CS 342
Operating Systems

Project 1

Fuad Aghazada
21503691
section 1

14.10.2018

Part A: Processes

Notes:

- Number of characters in the input file names is restricted to 100. (MAX_SIZE_FILENAME in phistogram.c)
- Intermediate files are created or overwritten inside *handleChild* method with the name of *output* concatenated with the corresponding index of child. For example:
  For the first child name of intermediate file → output1.txt
  For the second child name of intermediate file → output2.txt and etc.

Part B: Threads

Notes:

- The same notes for Part A.
- Since it is not mentioned in the assignment about the usage of static or dynamic data structures for global data, in the program number of input files is restricted to 10000 files. (MAX_NUM_FILES in thistogram.c) It can be changed according to the test cases.

Part C: Experiments

Notes:

- Measured time includes file open, write, close, shortly all the operations that have been executed by child / parent processes (threads).

a)

Note: Experiment has been done using input files named 'part_c_input_n_m.txt'.

TABLE 1
Running time for different processes/threads for the same input.

	MULTI-PROCESS	**MULTI-THREAD**
# OF PROCESSES / THREADS	Elapsed time (μs)	Elapsed time (μs)
1	10807	2483
2	14982	4480
4	36791	4626
8	50637	5425

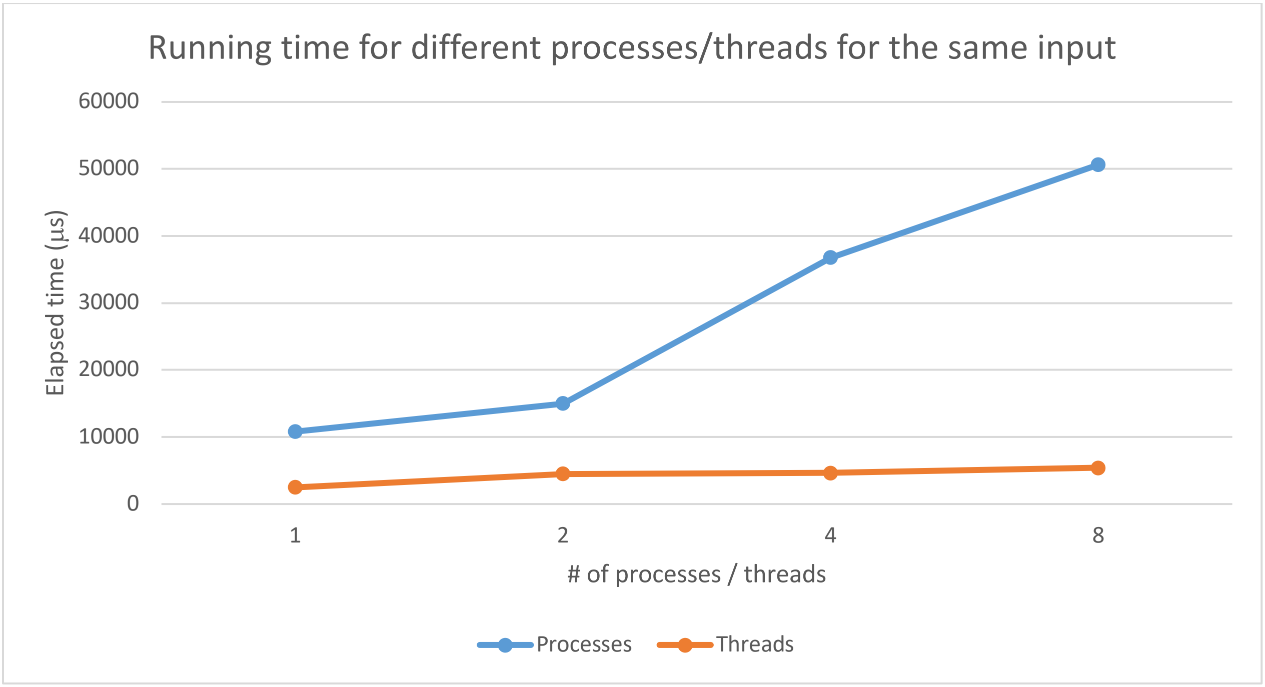

Figure 1. Relation between elapsed time and number of processes / threads.

b)

Note: Experiment has been done using input files named 'input6.txt' and 'input7.txt'

TABLE 2
Running time for 2 processes/threads for different input sizes.

.

	MULTI-PROCESS	**MULTI-THREAD**
FILE SIZE (# OF INTEGERS)	Elapsed time (µs)	Elapsed time (µs)
5	13950	3032
10	13526	3352
20	14472	4057
40	18405	9770
80	17408	2553
160	18159	4914
320	15120	11222

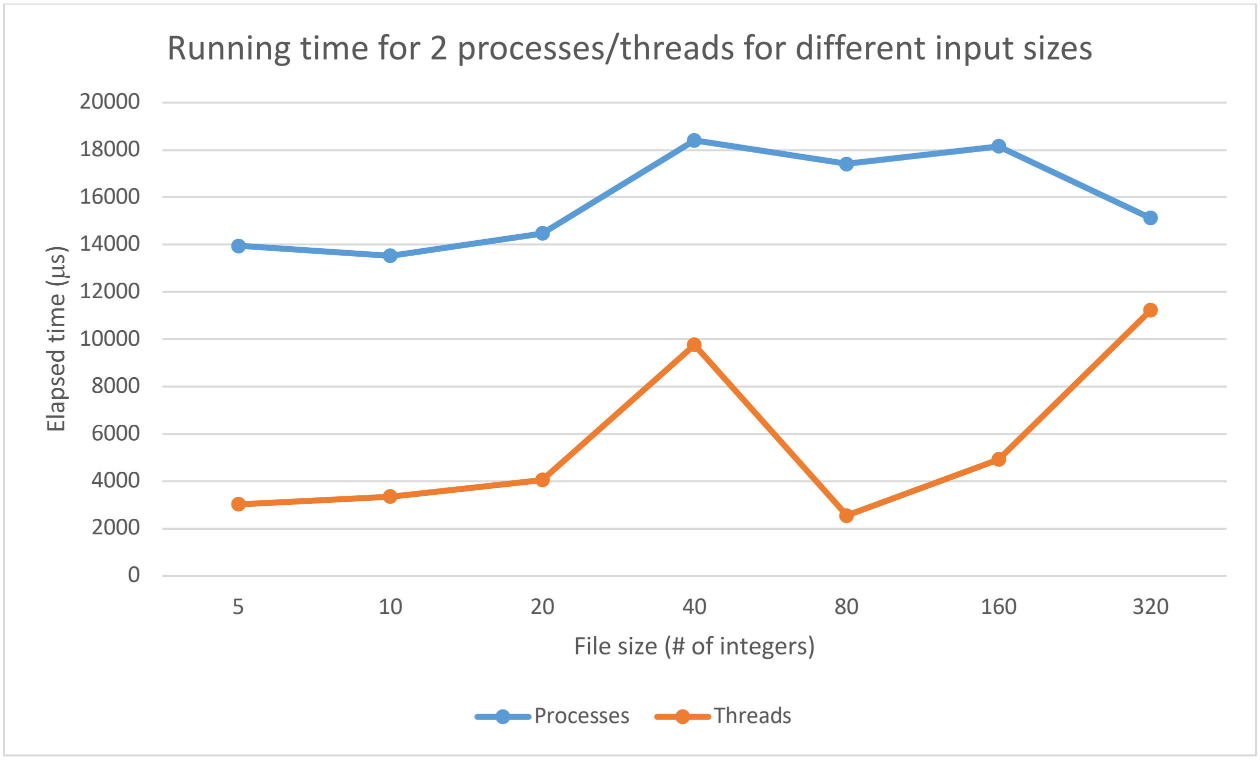

Figure 2. Relation between elapsed time and number of processes / threads.

Conclusion

From both of the experiments, it has been observed that multiprocessing version of the program works considerable slower than multithreading version of it. This should be a quite a reasonable result, because of the followings:

- Thread creation is more efficient than process creation.
- Threads does not create/write files for communication with other threads. However, we are dealing lots of file operations in the child and parent processes.