

Threading Assignment Report

PART 1:

How to compile my program:

```
gcc Threads_Part1.c -lpthread  
./a.out shakespeare.txt
```

1. Explain the problem and identify evaluation metrics for experiments

Given two-character strings s_1 and s_2 , write a parallel program using a threaded library to find out the number of substrings, in string s_1 , that was exactly the same as s_2 . We had to assume that $n_1 \bmod \text{NUM_THREADS} = 0$ and $n_2 < n_1/\text{NUM_THREADS}$. An evaluation metric will be the time each program takes to calculate the number of substrings.

2. Explain your choice of threading libraries

I picked the pthread library in C because I was familiar with it since I used it in a previous class.

3. Explain the design of the experiment and develop programs for evaluation

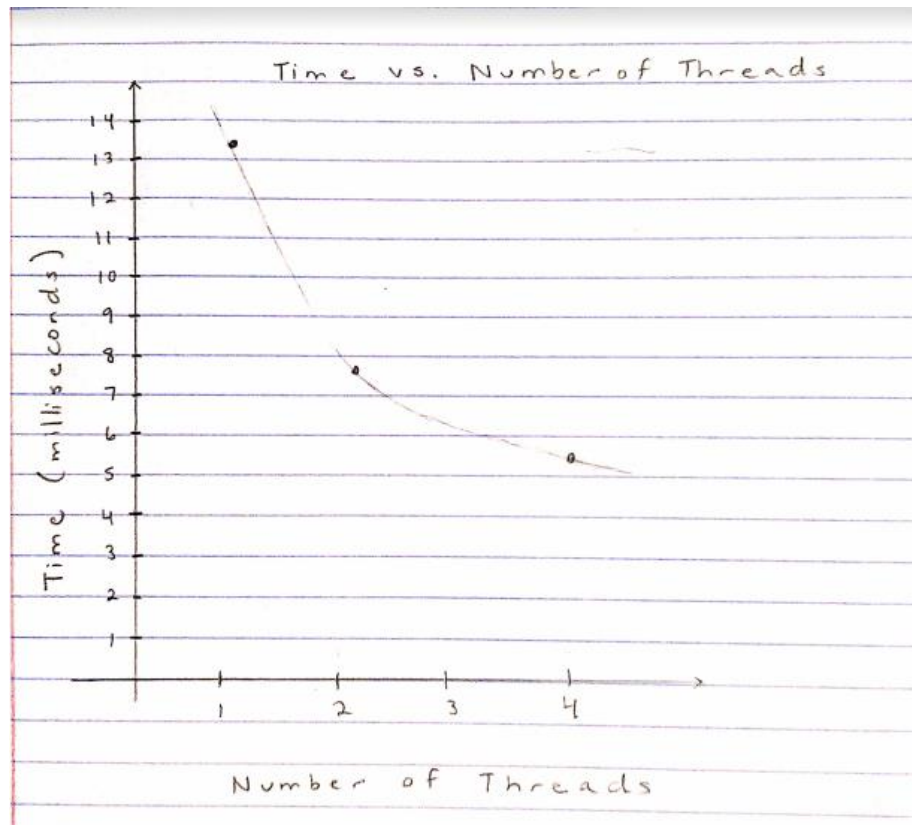
I ran my program using Shakespeare.txt from GitHub which searched for the substring “Romeo” in the string of Shakespeare’s works. My experiment consisted of running my program with 1 thread, 2 threads, and 4 threads. I ran each program 10 times and got the average of the times.

4. Detail your collected experimental results

The time for when 1 thread was used was the highest. The times for 2 threads were lower. The time for 4 threads were the lowest. This was the expected result since using more threads splits up the search of the substring, and therefore lowers the run time.

5. Analyze, graph, and interpret experimental results and draw conclusions

	1 Thread	2 Threads	4 Threads
Runtime 1	13.760	7.202	4.280
Runtime 2	13.321	7.869	5.897
Runtime 3	13.326	7.126	6.544
Runtime 4	13.394	8.234	4.866
Runtime 5	13.372	8.360	4.429
Runtime 6	13.262	7.434	6.448
Runtime 7	13.604	7.034	5.626
Runtime 8	13.293	8.260	5.764
Runtime 9	13.771	7.524	4.363
Runtime 10	13.575	7.263	5.079
AVERAGE	13.468	7.631	5.330



The more threads I used the lower the run times were. This was because the threads were executing in parallel and therefore the work of searching for the substring was split in between the threads. If I had tested with 50 threads, the time would have started to increase since we are only using 4 CPU's.

PART 2:

How to compile my program:

```
gcc Threads_Part2.c -lpthread
```

```
./a.out message.txt
```

1. Explain the problem and identify evaluation metrics for experiments

Implement the producer-consumer algorithm. Assume two threads: one producer and one consumer. The producer reads characters one by one from a string stored in a file named “message.txt”, then writes sequentially these characters into a circular queue. Meanwhile, the consumer reads sequentially from the queue and prints them in the same order. Assume a buffer (queue) size of 5 characters. An evaluation metric will be the functionality of the semaphores that are used.

2. Explain your choice of threading libraries

I picked the pthread library in C because I was familiar with it since I used it in a previous class.

3. Explain the design of the experiment and develop programs for evaluation

I used semaphores in my program to control when the producer added a character to the circular queue, and when the consumer printed out a character. I created temporary printf statements to check if the producer and consumer went back and forth. In other words, the producer had to insert a character, then the consumer would print that same character, then the producer would add another character, etc. I ran my program using three different txt files. One of them had a string that was 5 characters long. The other string had less than 5 characters. The third one had more than 5 characters.