

CENG322

PROGRAMMING ASSIGNMENT 3

REPORT

Umut YILDIZ - 260201028

1. Implementation details

Most of the functions that return 1 if something went wrong, return 0 if the process is successful.

An linked list is implemented for data list structure which also has insertion sort mechanism to help hold entries in ascending order.

An linked queue implemented for the task queue.

The data list and task queue is created globally to allow every function to reach and use these variables.

There are 3 critical sections in my code. Two of them are related to the task queue. Task enqueue and task dequeue needs to be controlled with lock in order not to do something wrong. Last critical section is the workers execution part where the worker does the task (insertion, search, delete). This part is controlled with the dequeue task part in the worker function.

There is a thread state which is used to control thread states. It can be IDLE which means thread is created but there is no task to do so wait idle. It can be WORKING which means try to get a task from the task queue and execute (Since the worker's job is a critical section, there can be only one thread working at the same time). It can be FINISHED which means there is no more task left to process. Thread can stop working.

2. Screenshots

Execute C program with 1 thread and 20 tasks. Time spent will be recorded to test.csv

```
mtylldz@Umuts-MacBook-Pro ~/D/S/C/H/260201028-HW3> ./a.out 1 20 test.csv
Tread 0: Task 0: 95 cannot be found
Tread 0: Task 1: 268 cannot be deleted
Tread 0: Task 2: 176 cannot be deleted
Tread 0: Task 3: 130 cannot be deleted
Tread 0: Task 4: 79 cannot be deleted
Tread 0: Task 5: 442 cannot be found
Tread 0: Task 6: 234 cannot be found
Tread 0: Task 7: 241 cannot be found
Tread 0: Task 8: 131 cannot be deleted
Tread 0: Task 9: 306 cannot be deleted
Tread 0: Task 10: 125 is inserted
Tread 0: Task 11: 11 is inserted
Tread 0: Task 12: 340 cannot be deleted
Tread 0: Task 13: 278 is inserted
Tread 0: Task 14: 93 cannot be deleted
Tread 0: Task 15: 18 cannot be deleted
Tread 0: Task 16: 394 cannot be deleted
Tread 0: Task 17: 372 is inserted
Tread 0: Task 18: 494 cannot be deleted
Tread 0: Task 19: 424 cannot be deleted
Main: Final Data List: 125 11 278 372
```

Execute C program with 4 threads and 20 tasks. Time spent will be recorded to test.csv

```
mtylldz@Umuts-MacBook-Pro ~/D/S/C/H/260201028-HW3> ./a.out 4 20 test.csv
Tread 3: Task 0: 273 cannot be deleted
Tread 2: Task 1: 280 cannot be deleted
Tread 2: Task 2: 362 cannot be found
Tread 2: Task 3: 88 is inserted
Tread 2: Task 4: 21 cannot be deleted
Tread 2: Task 5: 434 cannot be deleted
Tread 2: Task 6: 189 is inserted
Tread 2: Task 7: 315 is inserted
Tread 2: Task 8: 295 is inserted
Tread 2: Task 9: 398 is inserted
Tread 2: Task 10: 354 is inserted
Tread 2: Task 11: 139 is inserted
Tread 2: Task 12: 308 is inserted
Tread 2: Task 13: 235 is inserted
Tread 2: Task 14: 424 is inserted
Tread 2: Task 15: 407 is inserted
Tread 2: Task 16: 375 is inserted
Tread 2: Task 17: 341 cannot be found
Tread 2: Task 18: 231 cannot be found
Tread 2: Task 19: 205 is inserted
Main: Final Data List: 88 139 189 205 235 308 315 295 375 398 354 424 407
```

Execute C program with 8 threads and 20 tasks. Time spent will be recorded to test.csv

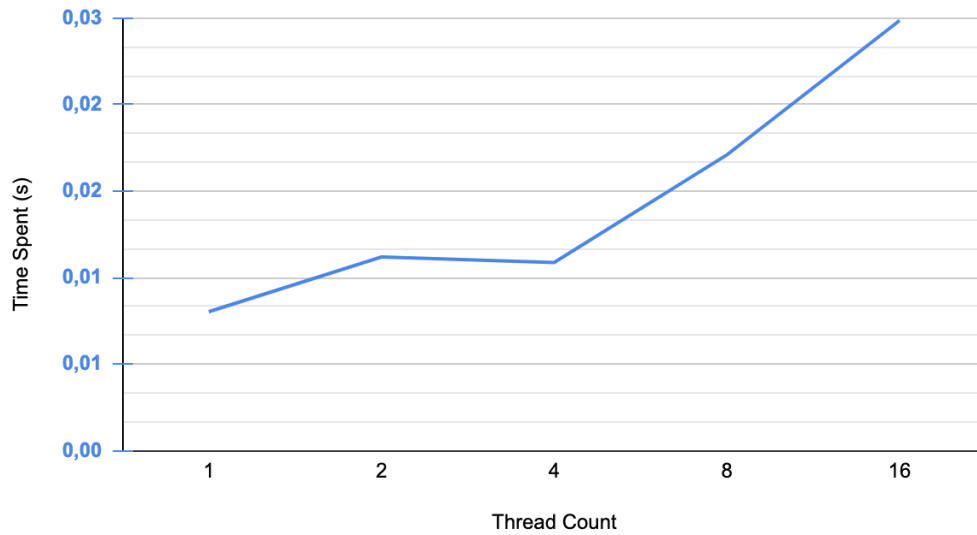
```
mtyldz@Umuts-MacBook-Pro ~/D/S/C/H/260201028-HW3> ./a.out 8 20 test.csv
Tread 7: Task 0: 295 cannot be found
Tread 1: Task 1: 43 cannot be found
Tread 0: Task 2: 385 cannot be found
Tread 4: Task 3: 295 cannot be deleted
Tread 1: Task 4: 30 cannot be found
Tread 4: Task 5: 356 is inserted
Tread 2: Task 6: 233 cannot be deleted
Tread 4: Task 7: 206 cannot be found
Tread 4: Task 8: 482 cannot be deleted
Tread 4: Task 9: 220 cannot be found
Tread 4: Task 10: 251 cannot be found
Tread 4: Task 11: 238 cannot be found
Tread 4: Task 12: 492 cannot be found
Tread 4: Task 13: 355 cannot be deleted
Tread 4: Task 14: 19 cannot be deleted
Tread 4: Task 15: 341 cannot be found
Tread 4: Task 16: 272 cannot be deleted
Tread 4: Task 17: 494 is inserted
Tread 4: Task 18: 133 cannot be found
Tread 4: Task 19: 271 is inserted
Main: Final Data List: 271 356 494
```

Execute C program with 16 threads and 20 tasks.

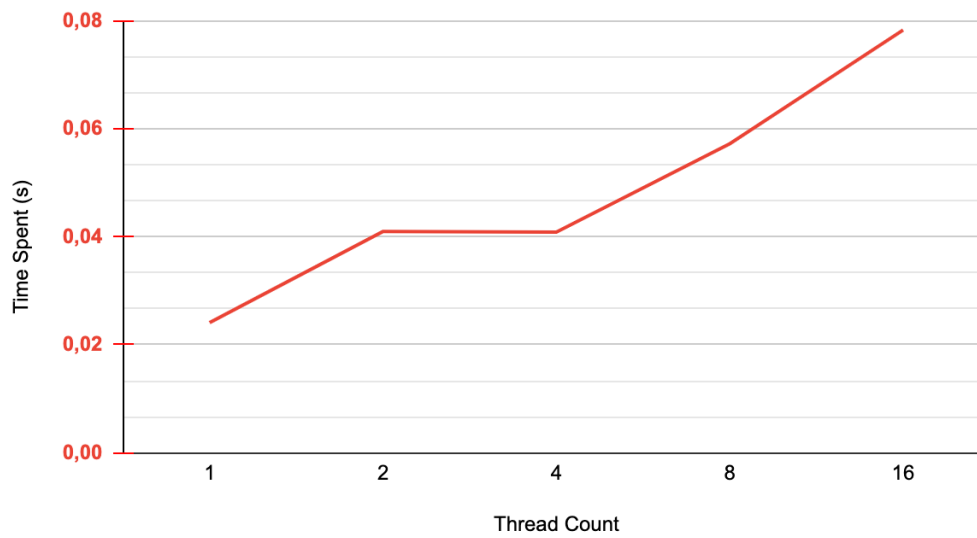
```
mtyldz@Umuts-MacBook-Pro ~/D/S/C/H/260201028-HW3> ./a.out 16 20 test.csv
Tread 0: Task 0: 208 cannot be found
Tread 0: Task 1: 107 cannot be found
Tread 0: Task 2: 489 cannot be deleted
Tread 0: Task 3: 285 cannot be deleted
Tread 3: Task 4: 156 cannot be found
Tread 9: Task 5: 168 cannot be found
Tread 3: Task 6: 59 cannot be deleted
Tread 9: Task 7: 33 cannot be found
Tread 12: Task 8: 44 is inserted
Tread 15: Task 9: 485 cannot be found
Tread 13: Task 10: 474 cannot be deleted
Tread 6: Task 11: 197 is inserted
Tread 6: Task 12: 177 is inserted
Tread 15: Task 13: 237 cannot be deleted
Tread 8: Task 14: 62 cannot be deleted
Tread 4: Task 15: 450 cannot be found
Tread 2: Task 16: 397 cannot be deleted
Tread 8: Task 17: 194 cannot be deleted
Tread 0: Task 18: 316 cannot be deleted
Tread 9: Task 19: 59 is inserted
Main: Final Data List: 44 59 197 177
```

3. Graphs

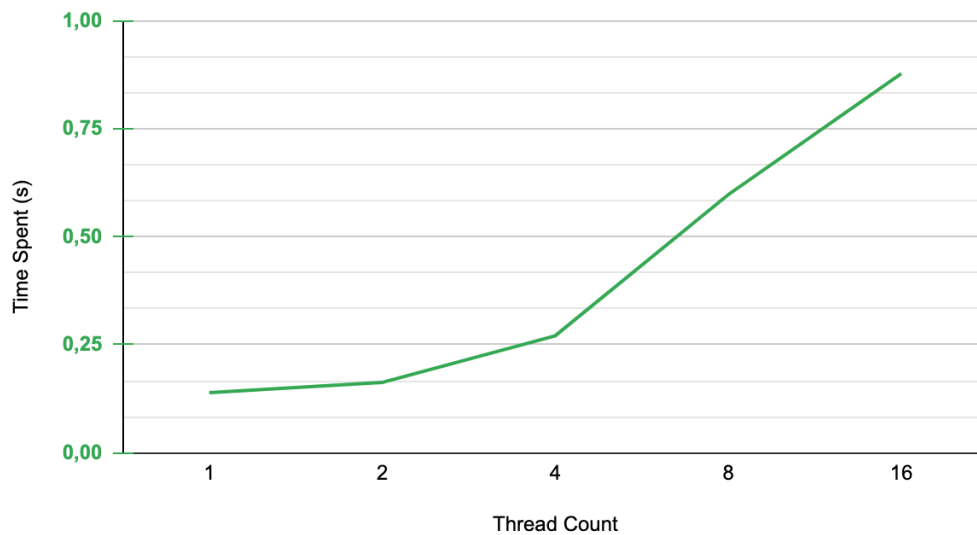
1K Tasks



10K Tasks



100K Tasks



4. Observation about the results

Time spent increases as the thread number increases. This is because while consumer(worker) threads are increasing, producer(main) thread is still one. When the thread count increases there are more lock processes which take time.