

Description (Bilal Atim)

To run: g++ main.cpp -o simulation

./simulation input.txt

Output: input_log.txt

Purpose of project: implementing a multi-threaded application with synchronization and data consistency.

To do that: we use pthread, mutex lock and unlock operation.

main.cpp :

In the main function, I pulled the number of clients from the input file.

I created my client objects by breaking the lines in the input file and kept these objects in an array. (Not: my customer id according to input file)

I created client threads and gave these threads my client array objects as input.

I ensured the synchronization by joining in the continuation.

The client threads called the client function and here they transfer the information they have in the critical region to another object using mutex lock and give this object as input to the machine threads, and then the mutex lock is unlocked. In this way, more than one customer cannot use the same machine at the same time.

Machine threads run in the machine function. Here, before processing the information they receive from the client, the mutex is locked and they calculate the total price in the critical section, so that more than one machine thread cannot change the global data at the same time.

Machine threads also add the amounts paid by the customers to the file in the critical section and close the file. Then the mutex is unlocked.

At the end of the program main thread adds the total balances of the companies to the file.

Example

Input:

and

Output:

```
≡ input1.txt X
≡ input1.txt
1 14
2 92,1,Kevin,300
3 44,1,Bob,231
4 2,2,Stuart,50
5 322,9,Otto,30
6 123,8,Stuart,65
7 90,10,Bob,100
8 1,4,Dave,40
9 4,3,Kevin,300
10 45,9,Bob,231
11 12,2,Stuart,50
12 344,9,Otto,30
13 451,8,Stuart,65
14 99,9,Bob,100
15 926,4,Dave,40
16
```

```
≡ input1_log.txt X
≡ input1_log.txt
21 Customer7,40TL,Dave
22 Customer3,50TL,Stuart
23 Customer8,300TL,Kevin
24 Customer10,50TL,Stuart
25 Customer2,231TL,Bob
26 Customer9,231TL,Bob
27 Customer6,100TL,Bob
28 Customer1,300TL,Kevin
29 Customer13,100TL,Bob
30 Customer5,65TL,Stuart
31 Customer4,30TL,Otto
32 Customer11,30TL,Otto
33 Customer12,65TL,Stuart
34 Customer14,40TL,Dave
35 All payments are completed
36 Kevin: 600TL
37 Bob: 662TL
38 Stuart: 230TL
39 Otto: 60TL
40 Dave: 80TL
41
```