CMPE322 PROJECT 3

Furkan Bülbül-2020400318

Execution

I wrote the program on an Ubuntu virtual machine. It can be be compiled with g++ compiler, also with Makefile. After make command is executed please run the command "./main.o input.txt". Input file should be in the same directory. Output file will be placed into the same directory again.

Implementation

I implemented the code by utilizing from pthread library with C++. Specifically used mutex locking-unlocking- and cond variable for waiting and signaling for threads. Firstly, created machine threads that handle the payment of a customer. Secondly, I created the customer threads as many as specified in the input file. I used mutexes for the following conditions:

- Machine - customer: used when machine might be reading and processing the data that belongs to another customer so that it should not be overwritten.
- Customer - Customer: used when different customer threads trying to send data to machine. This lock should be acquired to send data to the machine.
- Common case: Some variables is accessed from any part of the program. They are updated and read many time such as finished variable and printing the logs of the program - threads print in a way that interfere with each other-.
- Company balance: Different machine threads should acquire the mutex that belongs to the specific company so that machine can update the balance. Each company has its own balance mutex.

Conditions are used for the machine to wait for a customer and customer to wait for the machine until it becomes available . When the conditions are met, signal is send to the machine or the customer.
I wanted to sleep the all customer threads. To achieve that I used the pthread barrier so that order of creating the customers would not matter in the execution.

Evaluation

I indexed customers as given in the description file of the project. I compared my outputs with the given outputs. I evaluated the output considering the relative order of customers in a machine. Observed that they are compatible with the FIFO functionality - except rare cases when only 1 ms differs between sleeping times. Also, the balances of the companies were the same.