Name: Pandey Shourya Prasad
Roll No:- IMT2021535

OVERVIEW

This is a project that operates on a command-line interface and simulates an online store. It employs a client-server architecture where client requests are transmitted via sockets to the server program for processing and return of necessary results. User privileges are granted based on the type of user; a customer, who is a regular user, has access to view their own shopping cart, whereas an Admin user has unique privileges to modify store items.

Socket programming is utilized for communication between the client and the server, with the client-side and server-side communicating through system calls, while file locking mechanisms are used by the server side to interact with data files.

FILES IN THE FOLDER

Following files are there in the folder:-

1. Client.c and main_client.c :- This is client side which allows client to signup/login and then use the functionality like - addition of product into cart, list all the products in cart, list all the product available in store and etc.
2. Server.c and main_server.c :- This is server side which processes all the request received from client side. File locking is also done by server side of the program
3. Customer.txt, orders.txt, receipt.txt, records.txt - These .txt files stores all the appropriate details in it like customer.txt stores all the customer details in it, receipt.txt has receipt of orders that are processed.
4. Headers.h- this contains the structs and macros required in the program.

HOW TO RUN THE CODE

1. For Server side:-
   gcc -o server sever.c main_server.c

2. For client side:-
   gcc -o server client.c main_client.c

3. Now in two/three terminals in which one of the terminals will have server side. (to run it we have to type “./server”).
4. In other terminals, we can have customer side.(to run that we type in terminal “./client”)

FUNCTIONALITIES

Customer Side:-

1. Customer has options of seeing the available product
2. See the cart

3. Add products to the cart
4. Edit the cart
5. Do the payment
6. Register the new customer.

Admin Side:-

Admin can:-

1. Add the products
2. Delete the products
3. Update the price of an existing product
4. Update the quantity of an existing product
5. See the inventory

Server side:-

Server basically entertains the requests that it receives from the client side (be it user or admin)

Locking Mechanism

The program utilizes both mandatory and record locking in specific instances. These are the four places where the locks are applied:-

1. To locate the customer ID in the customers.txt file, a mandatory read lock is imposed on the entire file.
2. Once the cart for the customer has been located in orders.txt, record locking is implemented at that offset to prevent unauthorized access for reading or writing.
3. To search for a product ID or display the inventory, a mandatory read lock is imposed on all products.
4. To modify a specific product, the read lock on the product is removed, and a write lock is obtained to make changes.

SOCKET PROGRAMMING

The program utilizes sockets to facilitate communication between the client and the server. The following are the ways in which the program uses sockets:-

1. The program employs server-side socket system calls such as socket, bind, listen, and accept to establish the socket at the server end, and client-side socket system calls such as socket and connect to establish the socket at the client end.
2. The fork() system call is utilized to establish a concurrent server.
3. Read and write system calls are used for all socket-related reads and writes.

FUNCTIONS AND THEIR USAGE:-

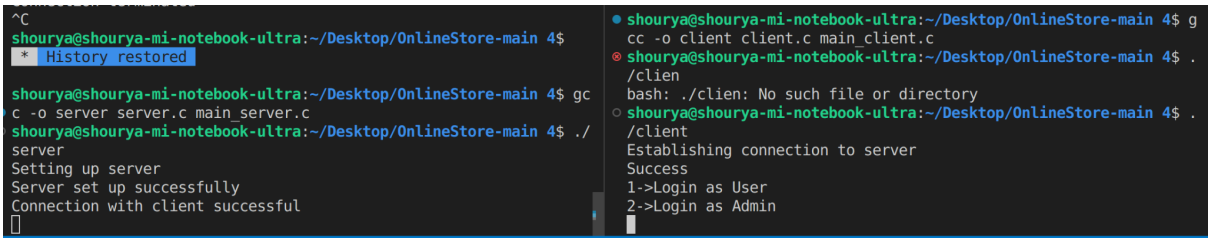
The functions perform the following tasks:

1. SetLockCustom() : Applies a mandatory read lock on the entire customers.txt file, allowing access to the current registered customers.
2. productWriteLock(): Sets a write lock on the current product to update or delete it.

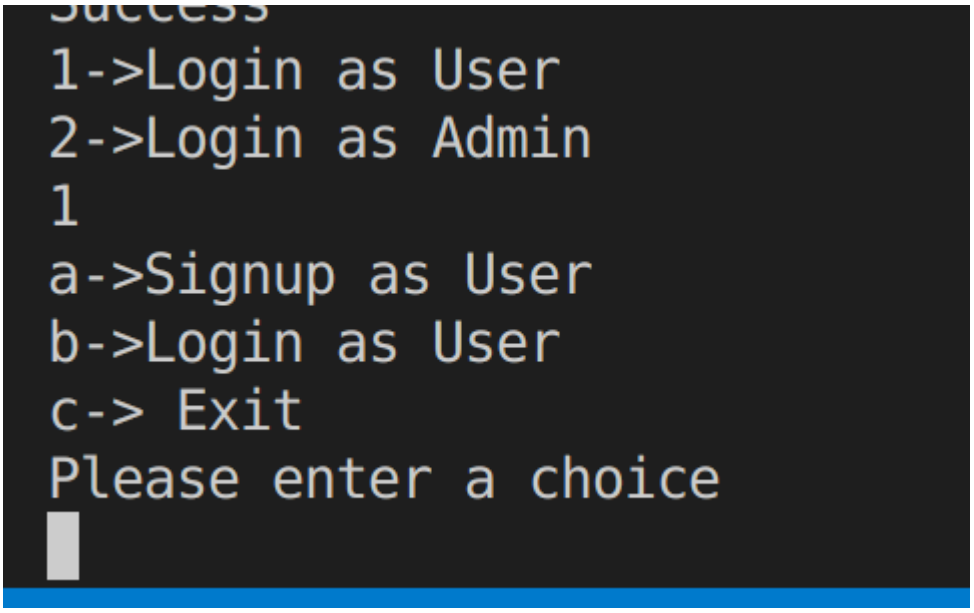
3. productReadLock() : Applies a mandatory read lock on the entire records.txt file, enabling us to check or display specific products.

4. unlock(): Unlocks any given lock as needed. From this point on, it is assumed that the appropriate lock is applied whenever a file access is performed and unlocking is also performed as required.
5. CartOffsetLock(): Locks the specific cart for reading or writing, taking the cartOffset as an argument which is obtained from the getOffset() function.

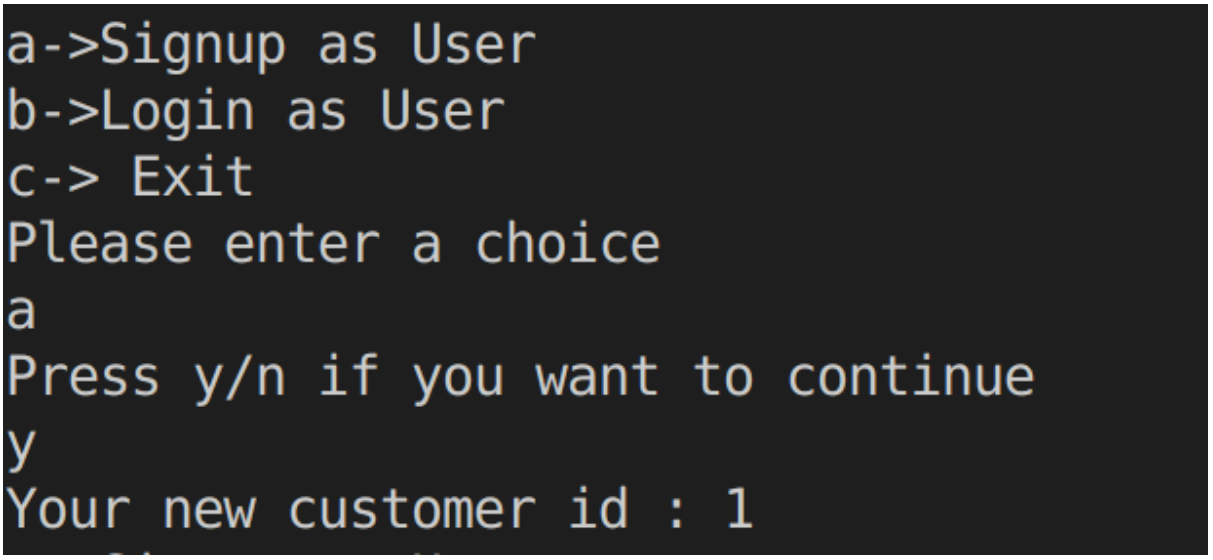
SCREENSHOTS



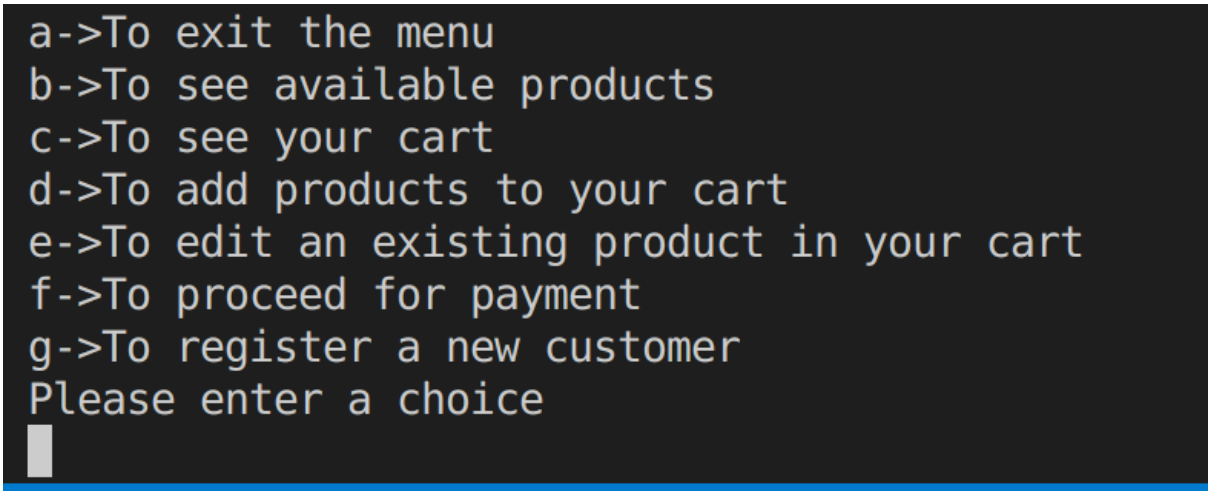
This is when server and client program is ran. Right side is customer side and left is server.



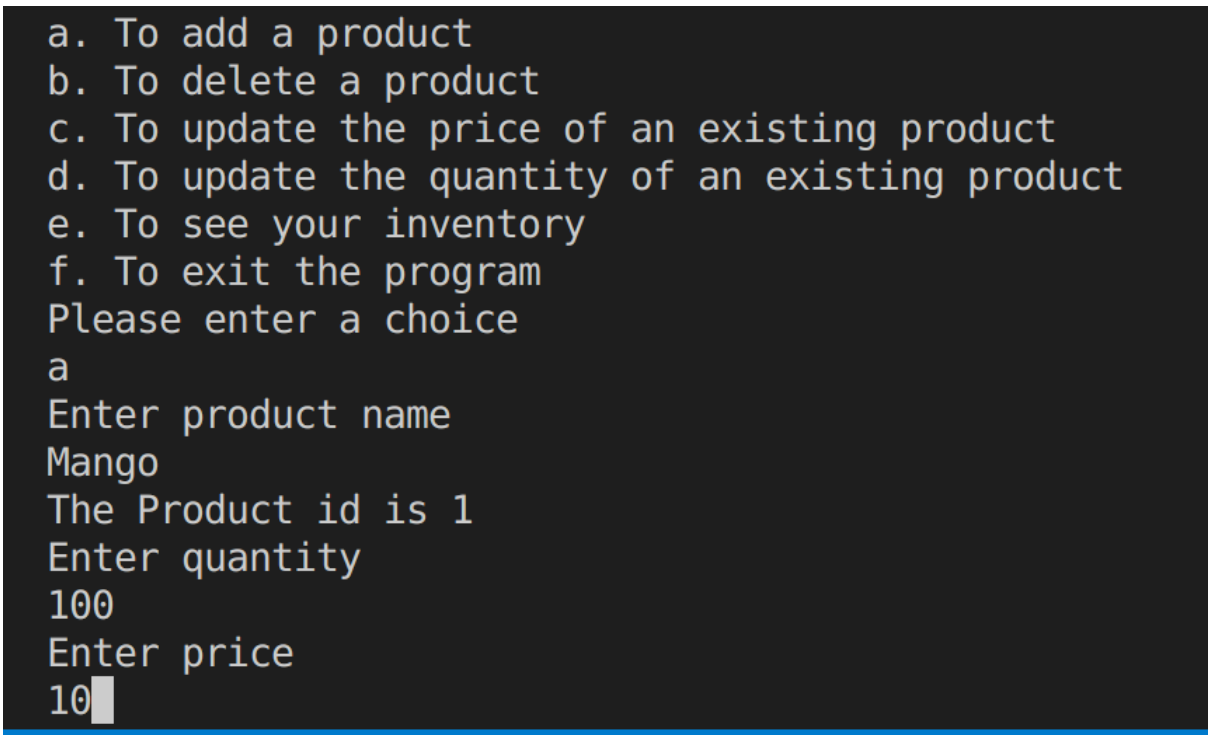
This choice is given to client side.



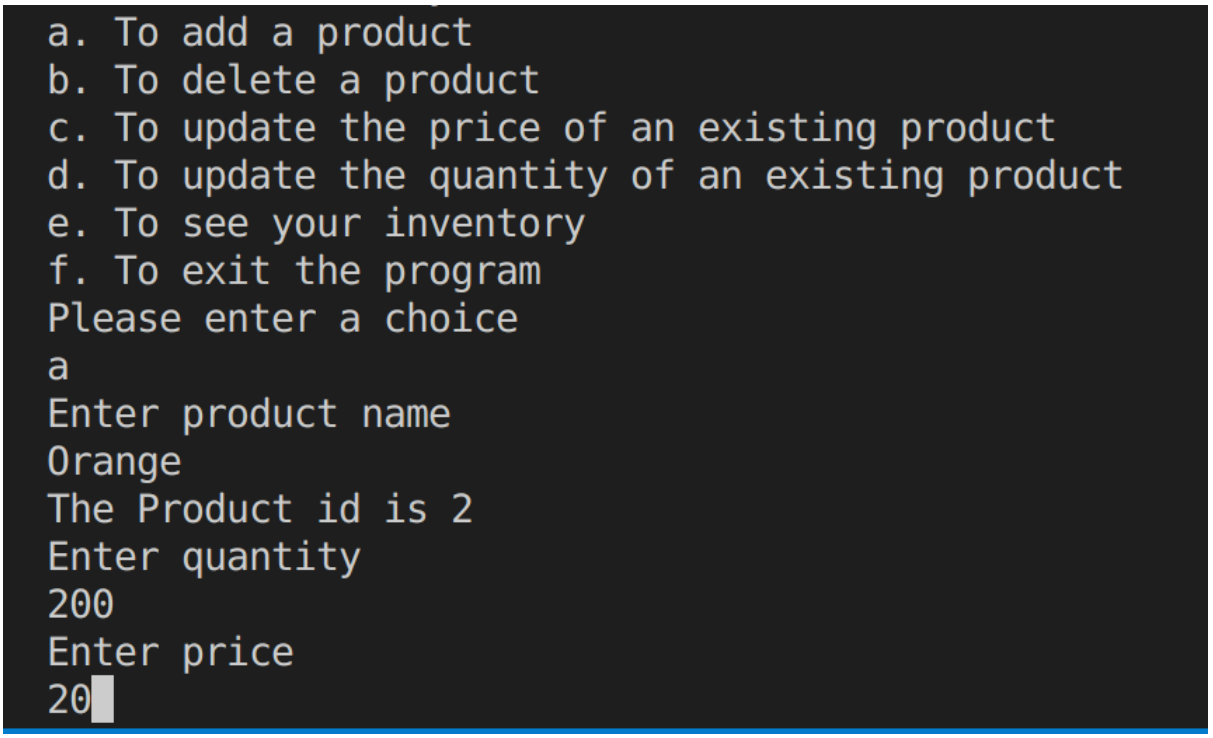
This shows the signup part of the client side



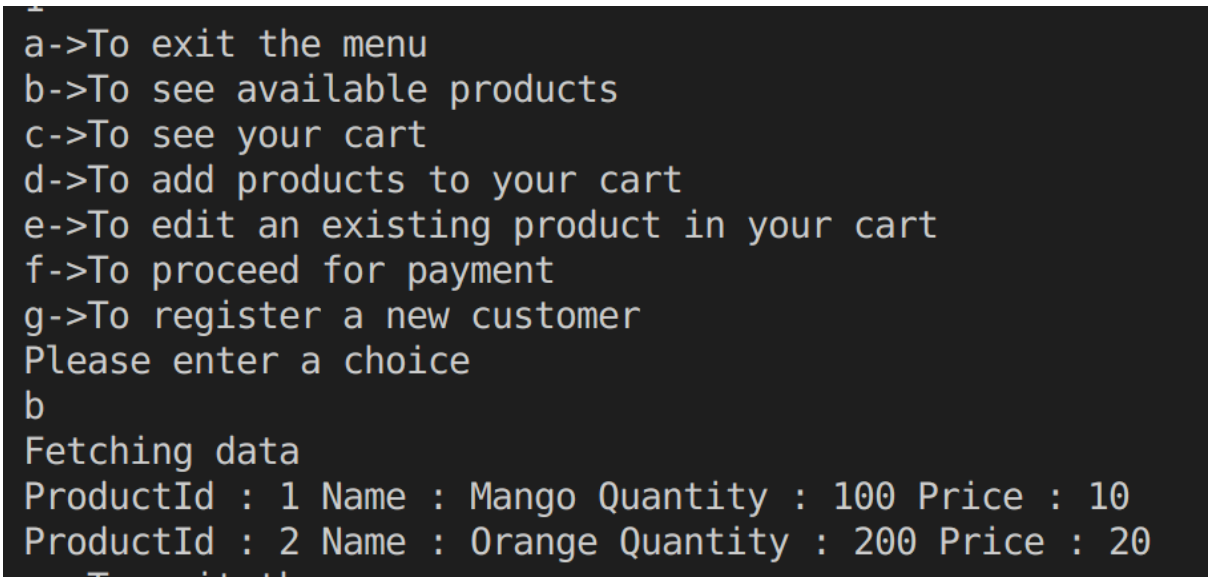
Available choices for client as a user.



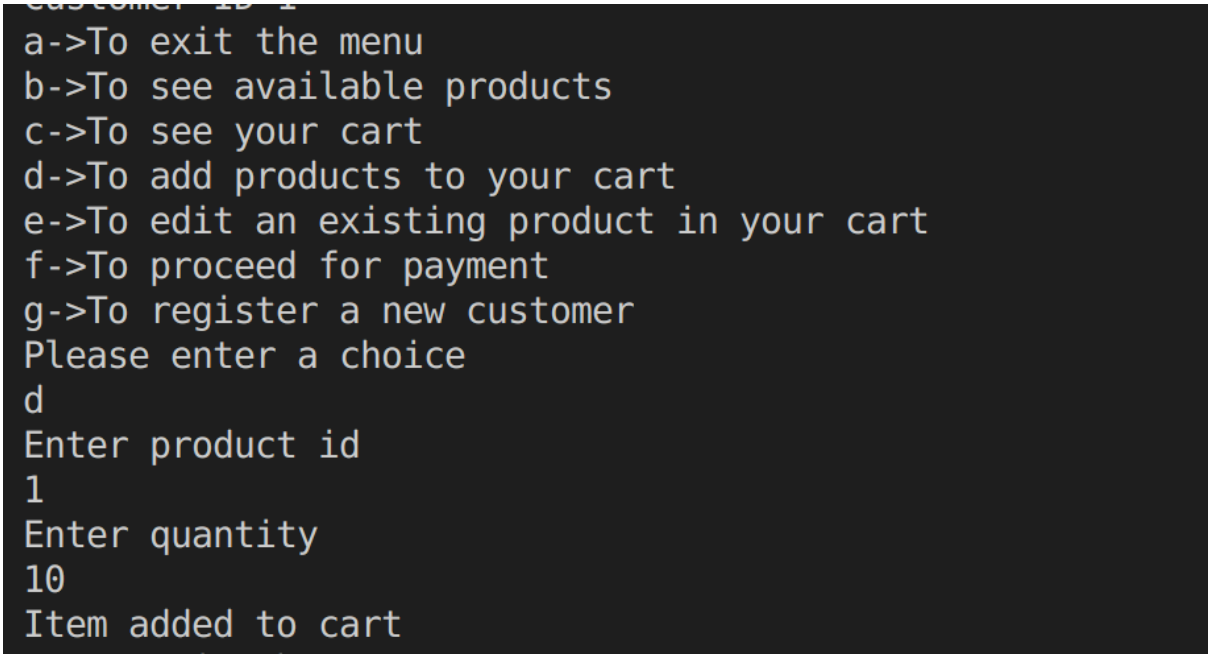
This is for the admin. Adding the product with name mango and quantity as 100 and price as 10.



This is addition of second product.



This is on the client side where they can fetch the available products and then add product to their cart.



User adding the product to their cart.

```
a->To exit the menu
b->To see available products
c->To see your cart
d->To add products to your cart
e->To edit an existing product in your cart
f->To proceed for payment
g->To register a new customer
Please enter a choice
c
Customer ID 1
ProductId: 1 Name: Mango Quantity: 10 Price: 10
```

Shows the cart of the customer.