Operating Systems Lab
Mini Project Report

Name: Barath S Narayan Roll No: IMT2021524

Overview

The goal of the project was to create a server client based shopping portal in which a client can connect to a server and then send requests to purchase products. The project was implemented in the C programming language and uses Operating system concepts such as locking mechanisms like semaphore and socket programming to achieve the purpose. There are two main programs, client.c and server.c. The client.c program contains the user portal to connect to the server and the server.c program takes in the requests of the clients does the appropriate action and then sends the appropriate response to the client. Multiple clients can connect to the server and the server will handle all the requests simultaneously. This done using a concurrent server which calls the fork function to handle each new connection.

Major files

Server.c

The server.c file has many functions which correspond to the various actions necessary for a online ecommerce portal. The file uses the concept of a concurrent server to take care of multiple clients.

Server():

> The function sets up a socket connection as a server and waits for a client to connect. Once a client is connected it first checks if the client is an Admin or a User based on the client option sent by the client. If the client option is User it then goes into a loop where it takes in the option sent by the client and does the appropriate User action. If the client option is Admin it then goes into a different loop where it takes in the options sent by the client(Admin) and does the appropriate Admin action

AddProduct():

> This function adds the product to the list of products. It takes in the Product Id, Name, Quantity and Price. A product structure is created and is written into the “Products.dat” file.

delProduct():

This function takes the product Id as input and deletes the product structure from the “Products.dat” file. This is done by writing all the other products into a duplicate file and then the original file is deleted. The duplicate is renamed to the original file

Update():

The update function takes in the Product Id, Quantity and Price as input and then updates the product if it is present in the “Products.dat” files.

Products():

The products function prints all the products in the “Products.dat” file and sends the output to the client. The client can be User or Admin.

Cart():

The cart function takes the customer Id as input and then sends the info regarding the products present in the customer cart to the client which will then print it out.

AddtoCart():

The addtoCart() takes in the customer Id, Product Id and the quantity. The function checks if it is a valid customer Id, Product Id and quantity. If all the conditions are satisfied the product is added to the cart in the “Customer.dat” file.

EditcartItem():

This function takes in Customer Id, Product Id and quantity as input. If the customer Id and product Id is valid then the quantity is updated in the “Customer.dat” file.

DeletecartItem():

This function takes in Customer Id and Product Id as input and then deletes the cart product if it is a valid Customer Id and Product Id from the “Customer.dat” file.

Createcart():

This function creates the cart for a new user. This adds a structure called Customer to the “Customer.dat” file which is a structure which contains the customer Id and the cart items as an array of product structures.

Buycart():

The buycart function buys the cart of a particular customer if the products and the corresponding quantity is there in the “Products.dat” file. If the products are there then the corresponding total is calculated. The user then must then type the appropriate amount again as confirmation to buy the products in the cart. Once

bought the cart is then cleared. The quantity of products is also updated in the "Products.dat" file.

Acquirep() and Acquirec():

Both these functions try to acquire the semaphore lock where acquirep() is a binary semaphore which is for the "Products.dat" file while the acquirec() is a binary semaphore which is for the "Client.dat" file. A function which uses the "Products.dat" file must call the Acquirep() while a function which uses the "Clients.dat" file must call the Acquirec()

Releasep() Releasec():

These functions release the lock acquired from the acquirep() and acquirec()

Client.c

LoginMenu():

The login menu function prints the login menu of the client program

AdminMenu():

The admin menu function prints the admin menu of the client program

UserMenu():

The user menu function prints the user menu of the client program

User():

The user function takes care of the user part of the client. It allows you to choose between several user related options and does the appropriate action based on the options chosen. The possible options are:

- Add Product to Cart
- Delete Product from Cart
- Edit Cart
- List all Products
- List all Cart Items
- Proceed to Pay
- Logout

Based on what option gets chosen, the user must have to give the necessary details needed for the operation to be done without any issues

Admin():

The admin function takes care of the admin part of the client. It allows you to choose between several admin related options and does the appropriate action based on the options chosen. The possible options are:

- Add a Product
- Delete a Product
- Update a Product
- List of products
- Logout

Based on what options get chosen, the admin must have to give the necessary details needed for the operation to be done without any issues.

Main():

The main function allows you to choose between two options. The 2 options are user and admin. If the client chooses option 1 they get admin privileges and if option 2 is chosen they get user privilege. From there they will be taken to the appropriate menu

Locking Mechanisms

The project uses Semaphores to ensure that there is no race condition between multiple clients. So if two clients are ready to buy the semaphore lock ensures that the second person waits till the first person completes the payment. This way the program ensures that the quantities and items are checked everytime a new customer is ready to buy and checks if the necessary quantity is still present. The server uses two semaphores. One for the “Products.dat” file and the other for the “Clients.dat” file.

Structures Used

The structures used in the program are

- Product:

  The product structure contains information such as product Id, name, quantity and price.

- Client:

  The client structure contains client id and an array of products. The client can add a product structure to the array and this array represents the cart of a user.

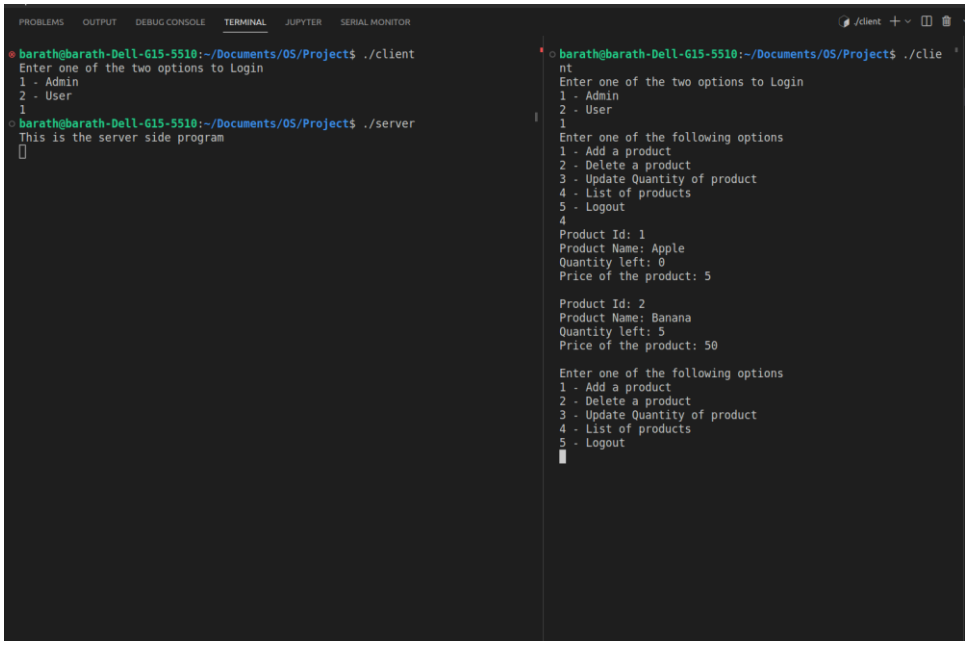
Socket Programming

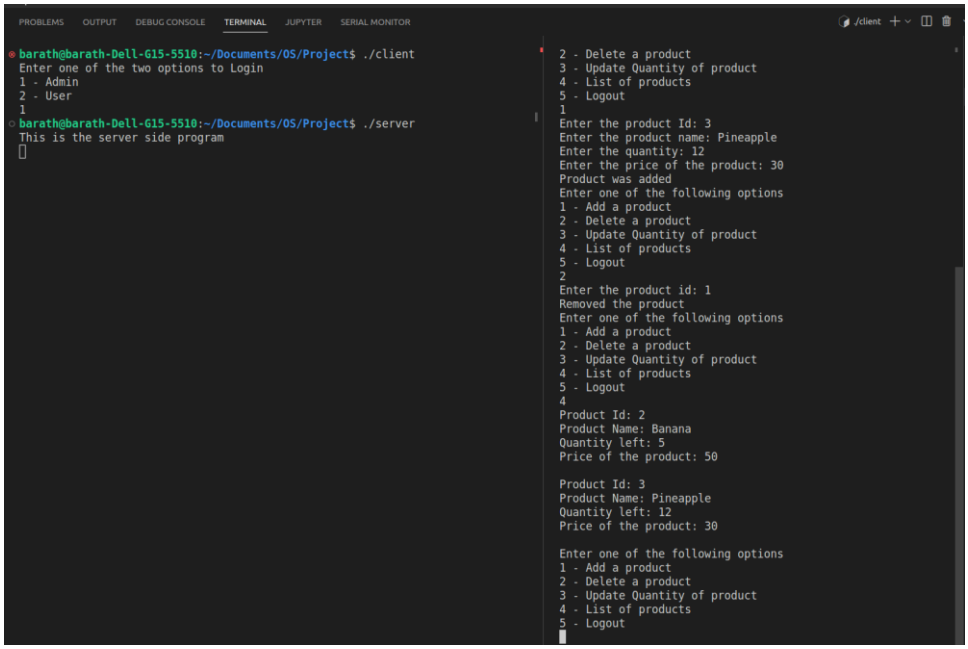
The program uses socket programming to help the client and server programs to communicate. The server acts as a concurrent server, it waits for a client to connect and once it receives a connection request it calls the fork function. The child process takes care of the requests of the client while the parent process waits for anymore clients to

connect. This way the server can take care of the requests of multiple clients. We use the semaphore locking mechanism to ensure that there is no racing condition between multiple clients.

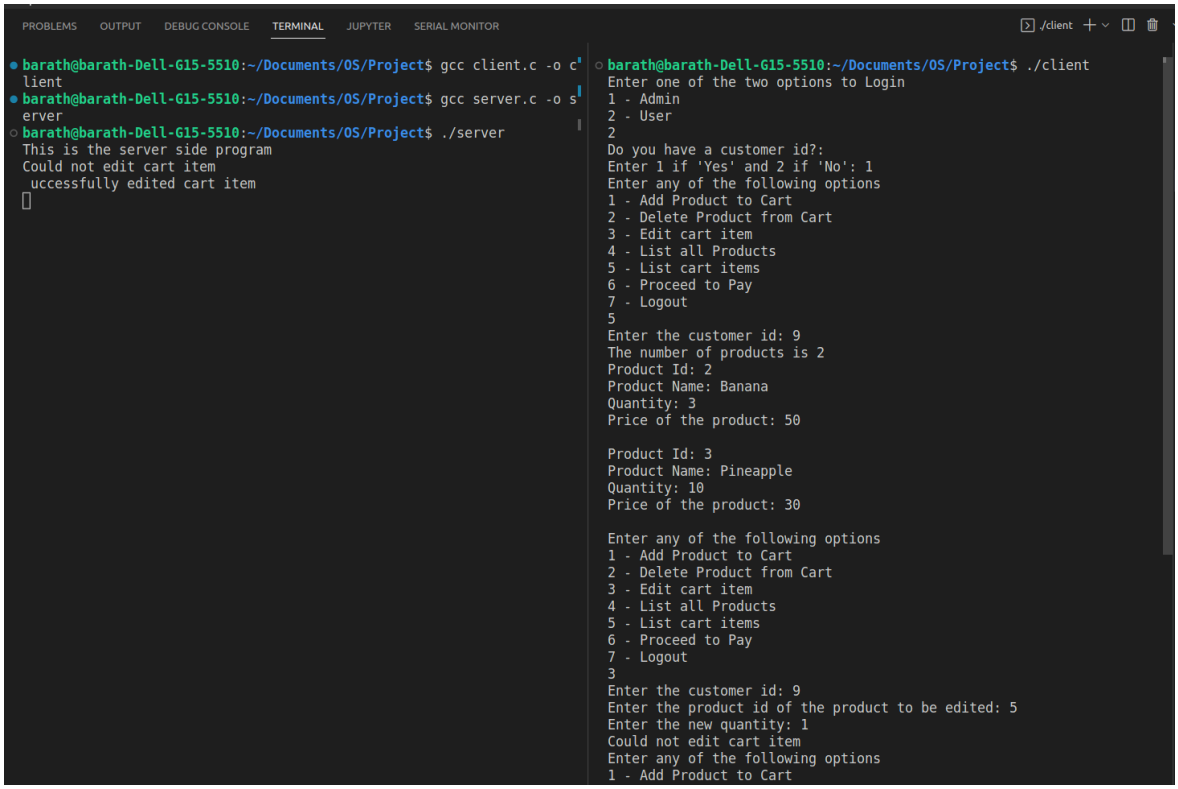
Program Execution Screenshots

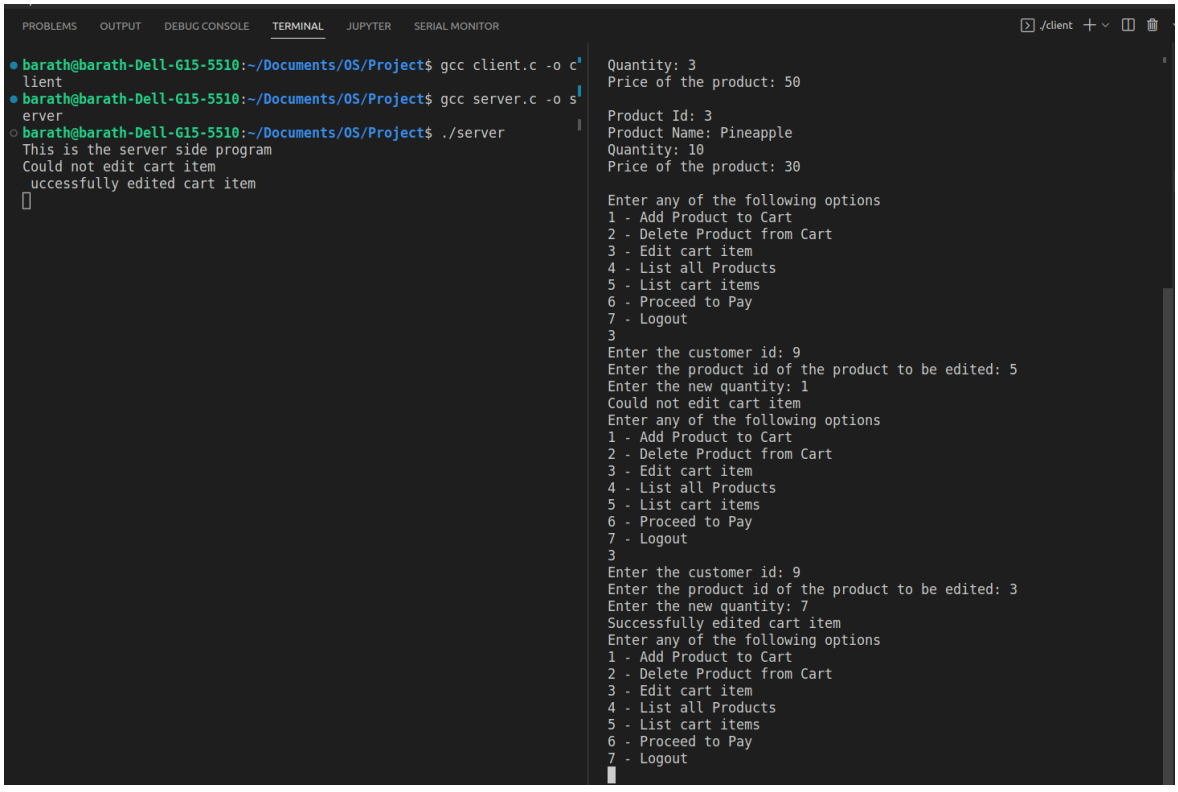
Admin execution screenshots:

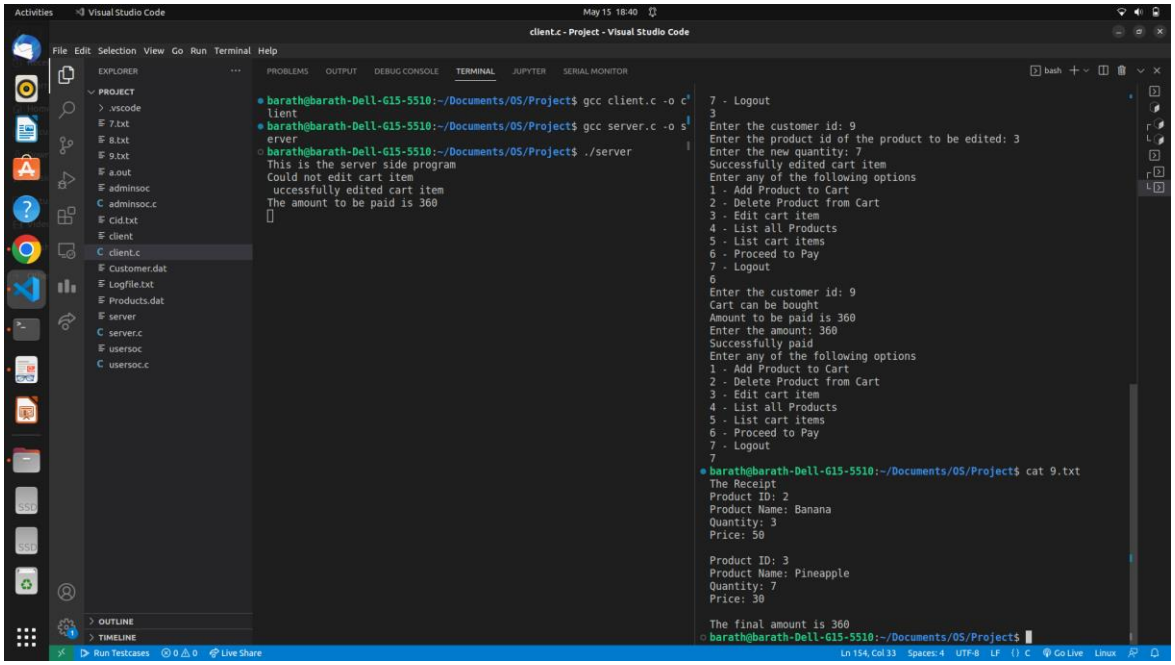




User Execution Screenshots:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
JUPYTER
SERIAL MONITOR
./client
barath@barath-Dell-G15-5510:~/Documents/OS/Project$ gcc client.c -o client
barath@barath-Dell-G15-5510:~/Documents/OS/Project$ gcc server.c -o server
barath@barath-Dell-G15-5510:~/Documents/OS/Project$ ./server
This is the server side program
Could not edit cart item
uccessfully edited cart item
barath@barath-Dell-G15-5510:~/Documents/OS/Project$ ./client
Enter one of the two options to Login
1 - Admin
2 - User
2
Do you have a customer id?:
Enter 1 if 'Yes' and 2 if 'No': 1
Enter any of the following options
1 - Add Product to Cart
2 - Delete Product from Cart
3 - Edit cart item
4 - List all Products
5 - List cart items
6 - Proceed to Pay
7 - Logout
5
Enter the customer id: 9
The number of products is 2
Product Id: 2
Product Name: Banana
Quantity: 3
Price of the product: 50
Product Id: 3
Product Name: Pineapple
Quantity: 10
Price of the product: 30
Enter any of the following options
1 - Add Product to Cart
2 - Delete Product from Cart
3 - Edit cart item
4 - List all Products
5 - List cart items
6 - Proceed to Pay
7 - Logout
3
Enter the customer id: 9
Enter the product id of the product to be edited: 5
Enter the new quantity: 1
Could not edit cart item
Enter any of the following options
1 - Add Product to Cart


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
JUPYTER
SERIAL MONITOR
./client
barath@barath-Dell-G15-5510:~/Documents/OS/Project$ gcc client.c -o client
barath@barath-Dell-G15-5510:~/Documents/OS/Project$ gcc server.c -o server
barath@barath-Dell-G15-5510:~/Documents/OS/Project$ ./server
This is the server side program
Could not edit cart item
uccessfully edited cart item
Quantity: 3
Price of the product: 50
Product Id: 3
Product Name: Pineapple
Quantity: 10
Price of the product: 30
Enter any of the following options
1 - Add Product to Cart
2 - Delete Product from Cart
3 - Edit cart item
4 - List all Products
5 - List cart items
6 - Proceed to Pay
7 - Logout
3
Enter the customer id: 9
Enter the product id of the product to be edited: 5
Enter the new quantity: 1
Could not edit cart item
Enter any of the following options
1 - Add Product to Cart
2 - Delete Product from Cart
3 - Edit cart item
4 - List all Products
5 - List cart items
6 - Proceed to Pay
7 - Logout
3
Enter the customer id: 9
Enter the product id of the product to be edited: 3
Enter the new quantity: 7
Successfully edited cart item
Enter any of the following options
1 - Add Product to Cart
2 - Delete Product from Cart
3 - Edit cart item
4 - List all Products
5 - List cart items
6 - Proceed to Pay
7 - Logout



How to run the program:

- First the server program (Server.c) must be executed in one terminal.
- Then the client program (Client.c) must be run.
- The client can choose between two options. The first option is admin while the second option is user.
- If the admin option is choosen the admin menu is printed and one can choose between the multiple options present.
- If the user option is choosen, then the program asks if the user already has a customer id. If the user has a customer id he can choose the appropriate option and get to the user menu.If the user does not have a customer id they can choose the appropriate option and they will get a new customer id.
- The user menu is printed and one can choose between the multiple options present and proceed.
- In the admin case a log file is generated everytime the admin logs out.
- In the user case a log file with the appropriate customer id is generated everytime a purchase occurs.