REPORT

Name: Kushal Dasari

ROLLNo: IMt2021035

Instruction to run the code and brief description about which file contains what:

1)data.c: This file is just for loading the "data.txt" with some initial products.

2)server.c: This file is intended for the administrator's use, who needs to modify the products by adding new ones, updating the price and quantity of an existing product, and deleting a product. Therefore, to make any such changes, the administrator must execute this file.

3) servermain.c and clientmain.c -> Both these files have to be run simultaneously to allow communication between server and client so that client can ask server to display all the products or add items to the cart and after adding items to the cart it can ask to display the cart, update the cart and if client confirms the order, it will be redirected to the payment section.

Instructions:

1)First compile and execute the data.c file

2) Now, if server wants to access products, compile and execute server.c. After executing, it will display the line “want to access the products? type y or n" after typing “y” it gives you 3 options for add, update and delete. Depending on the operation you want to perform type the corresponding number.

Instructions to run: gcc –o admin server.c and then. /admin

3) servermain.c and clientmain.c ->

Note: Before compiling these two files you need to change the port number in both files.

Instructions to compile and execute:

```
gcc servermain.c -o servermain
```

```
./servermain
```

```
gcc clientmain.c -o clientmain
```

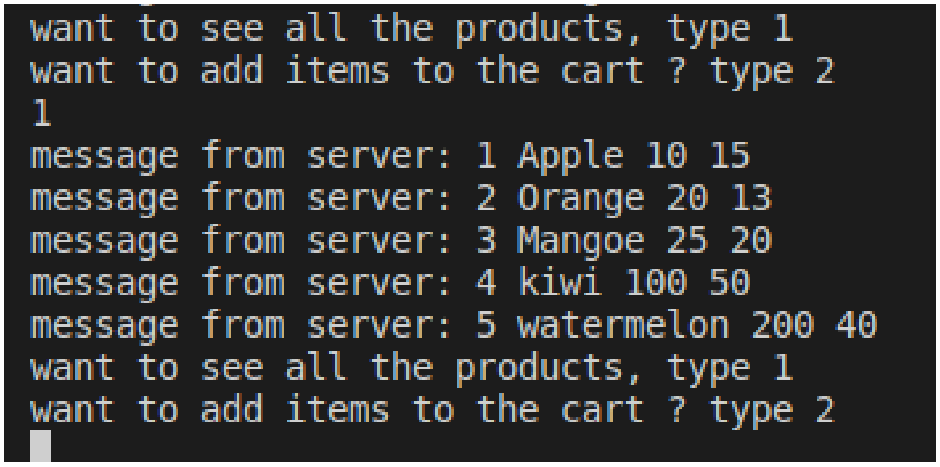
```
./clientmain
```

First compile and execute servermain.c, it will be waiting for the incoming connections, now compile and execute clientmain.c it shows up two messages:

want to see all the products? type 1

want to buy items? type 2

by entering 1 a message is sent from the client to server asking to display the products, after server receives this message, it will send details of all the products though messages:



```
want to see all the products, type 1
want to add items to the cart ? type 2
1
message from server: 1 Apple 10 15
message from server: 2 Orange 20 13
message from server: 3 Mangoe 25 20
message from server: 4 kiwi 100 50
message from server: 5 watermelon 200 40
want to see all the products, type 1
want to add items to the cart ? type 2
```

By entering 2, client sends a message saying “Add to cart” to the server and after server receives this message it also sends a message saying “enter the product id and quantity”. After entering product id and quantity, server can send any of the below 4 messages:

1)continue: valid product id and quantity

2)sorry that product is not in stock: if the available quantity is 0.

3)sorry only fewer items are left: if you enter quantity greater than the available quantity, quantity will be set to available quantity.

3)please enter valid product id: if you enter the product id which is not present or if that product has been deleted by the admin.

Adding items to the cart will be stopped if you enter -1 for product id and quantity can be any.

The photos for above statements are attached below.

And according to question we want to see our cart and update our cart :

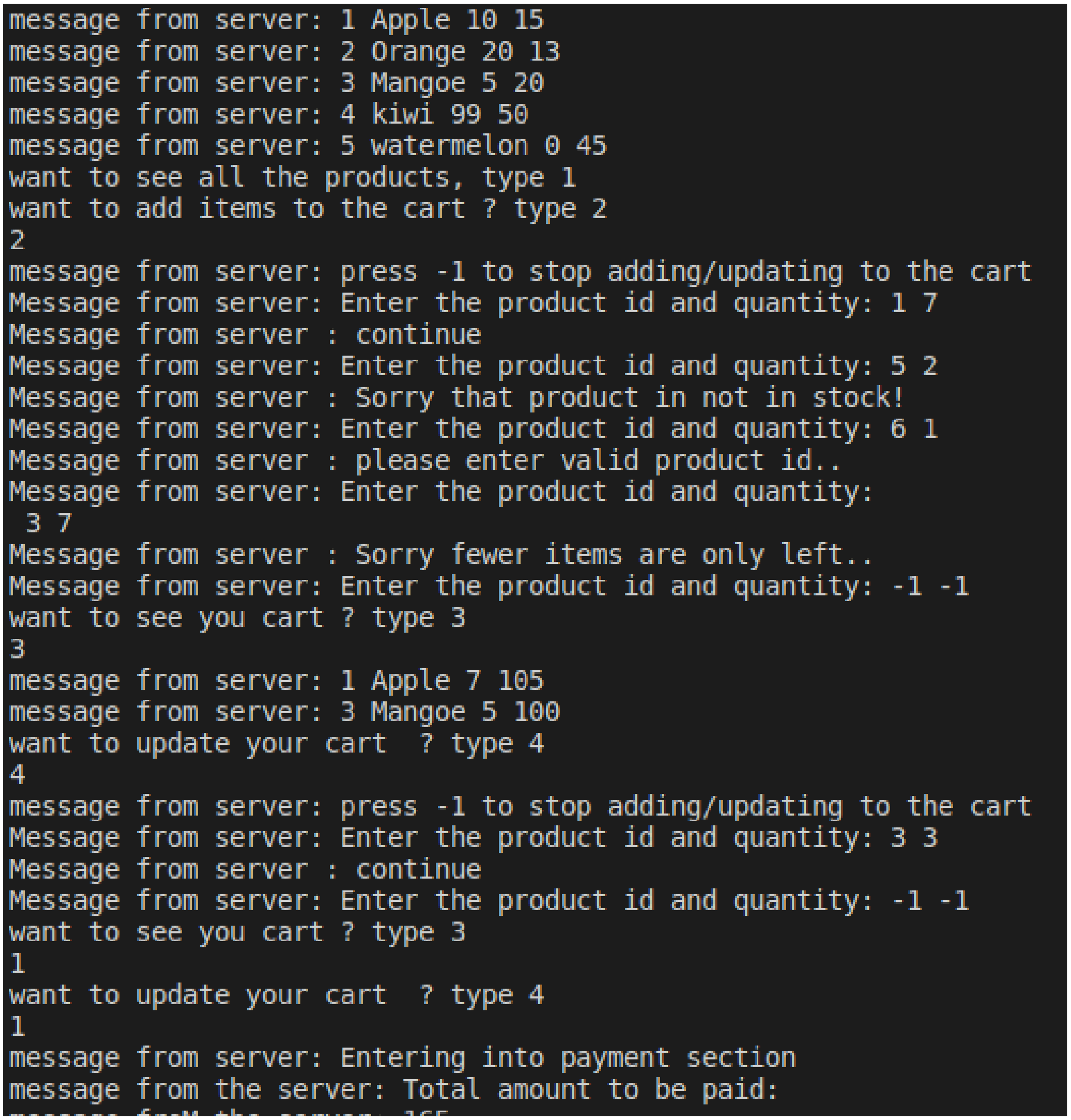
to see our cart type 3;

To update our cart type 4:You can see that thing in any of the below photos.

After updating it goes into payment session.

So this is the overview of my code.

Picture:



```
message from server: 1 Apple 10 15
message from server: 2 Orange 20 13
message from server: 3 Mangoe 5 20
message from server: 4 kiwi 99 50
message from server: 5 watermelon 0 45
want to see all the products, type 1
want to add items to the cart ? type 2
2
message from server: press -1 to stop adding/updating to the cart
Message from server: Enter the product id and quantity: 1 7
Message from server : continue
Message from server: Enter the product id and quantity: 5 2
Message from server : Sorry that product in not in stock!
Message from server: Enter the product id and quantity: 6 1
Message from server : please enter valid product id..
Message from server: Enter the product id and quantity:
 3 7
Message from server : Sorry fewer items are only left..
Message from server: Enter the product id and quantity: -1 -1
want to see you cart ? type 3
3
message from server: 1 Apple 7 105
message from server: 3 Mangoe 5 100
want to update your cart  ? type 4
4
message from server: press -1 to stop adding/updating to the cart
Message from server: Enter the product id and quantity: 3 3
Message from server : continue
Message from server: Enter the product id and quantity: -1 -1
want to see you cart ? type 3
1
want to update your cart  ? type 4
1
message from server: Entering into payment section
message from the server: Total amount to be paid:
message froM the server: 165
```

My approach:

First I created data.c file to store the products list in the data.txt file. Then I implement the code based on the use cases.

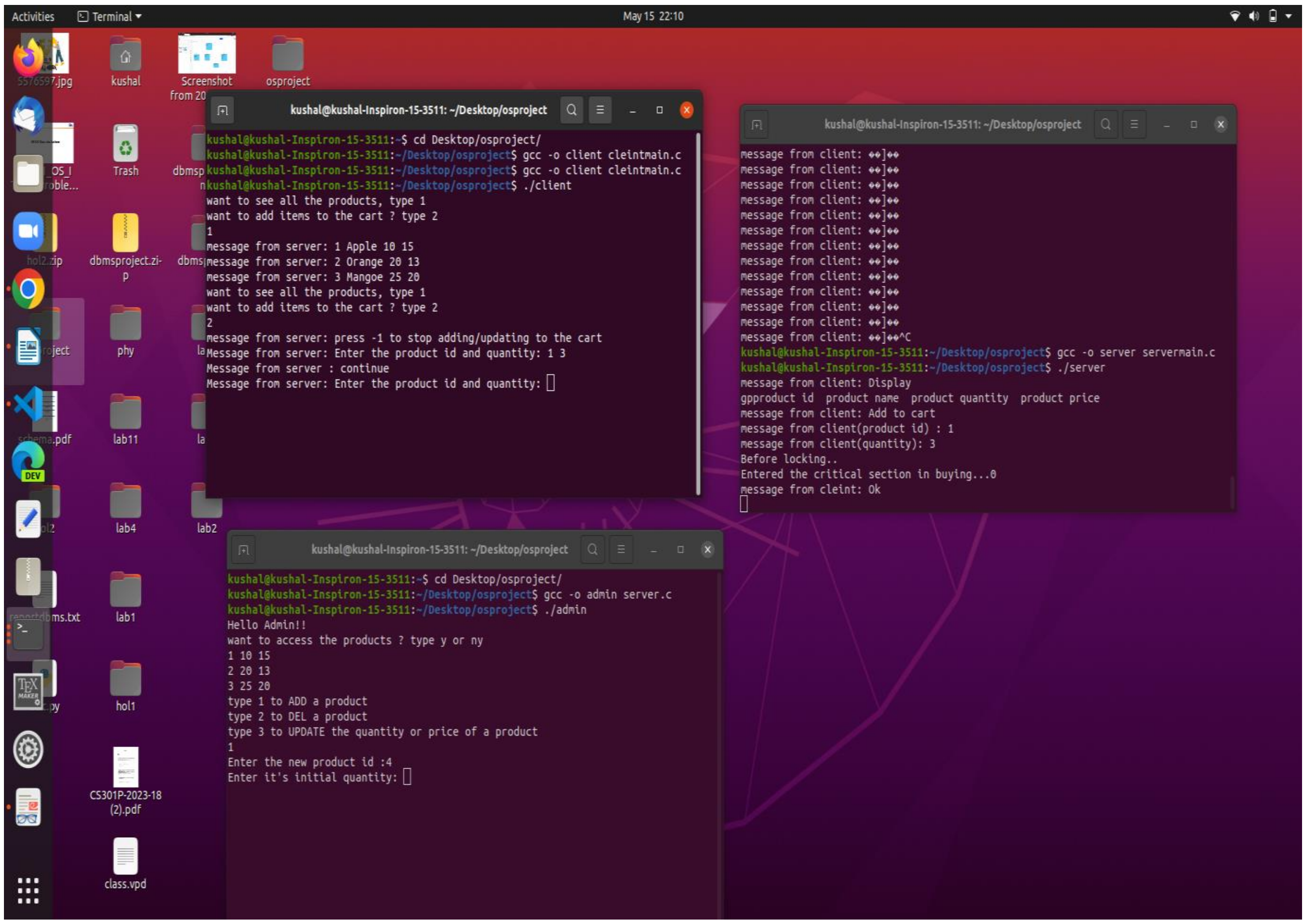
1st Use Case: (Admin adding a product deleting a product and updating the products).

After compiling data.c filen I created Admin.c file.

1a) Adding a product:

If we want to add a product, we can simply write the new id and quantity and name and price of the item to the data.txt file. It does not affect the server or client. Because the client cannot see the item before adding it and he cannot add that into the cart.

Here is the picture of the execution of 1a:



```
kushal@kushal-Inspiron-15-3511:~$ cd Desktop/osproject/
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o client cleintmain.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o client cleintmain.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ ./client
want to see all the products, type 1
want to add items to the cart ? type 2
1
message from server: 1 Apple 10 15
message from server: 2 Orange 20 13
message from server: 3 Mangoe 25 20
want to see all the products, type 1
want to add items to the cart ? type 2
2
message from server: press -1 to stop adding/updating to the cart
Message from server: Enter the product id and quantity: 1 3
Message from server : continue
Message from server: Enter the product id and quantity: 
```

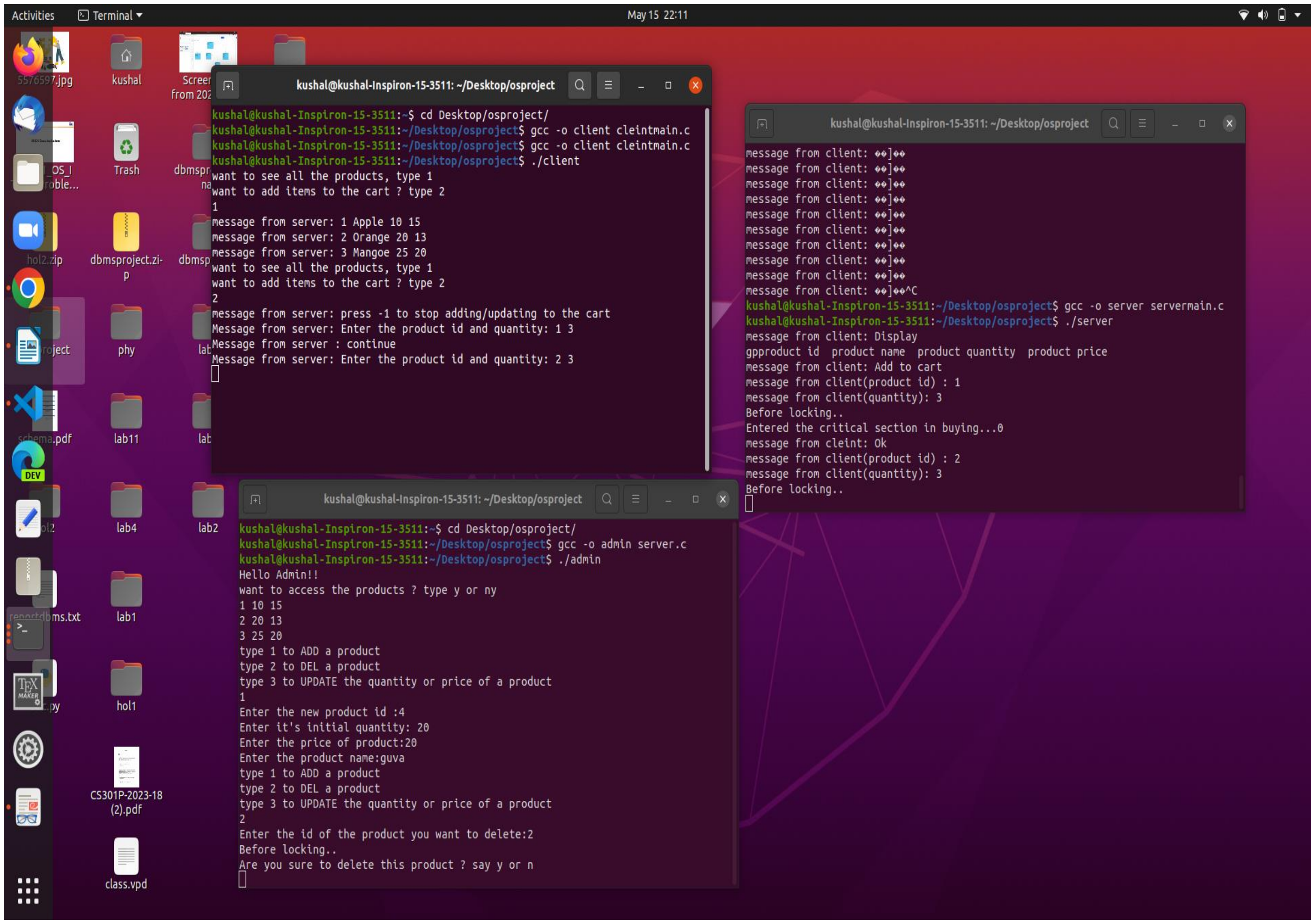
```
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��^C
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o server servermain.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ ./server
message from client: Display
gpproduct id  product name  product quantity  product price
message from client: Add to cart
message from client(product id) : 1
message from client(quantity): 3
Before locking..
Entered the critical section in buying...0
message from cleint: Ok
```

```
kushal@kushal-Inspiron-15-3511:~$ cd Desktop/osproject/
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o admin server.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ ./admin
Hello Admin!!
want to access the products ? type y or ny
1 10 15
2 20 13
3 25 20
type 1 to ADD a product
type 2 to DEL a product
type 3 to UPDATE the quantity or price of a product
1
Enter the new product id :4
Enter it's initial quantity: 
```

1b) Deleting:

If we want to delete a product, we need to first lock the file (file locking) and delete it. Because if a client wants to access the deleted product, it should not get accessed. Because the store does not contain that item.

Here is the picture of the execution of 1b:



```
kushal@kushal-Inspiron-15-3511:~$ cd Desktop/osproject/
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o client cleintmain.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o client cleintmain.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ ./client
want to see all the products, type 1
want to add items to the cart ? type 2
1
message from server: 1 Apple 10 15
message from server: 2 Orange 20 13
message from server: 3 Mangoe 25 20
want to see all the products, type 1
want to add items to the cart ? type 2
2
message from server: press -1 to stop adding/updating to the cart
Message from server: Enter the product id and quantity: 1 3
Message from server : continue
Message from server: Enter the product id and quantity: 2 3
```

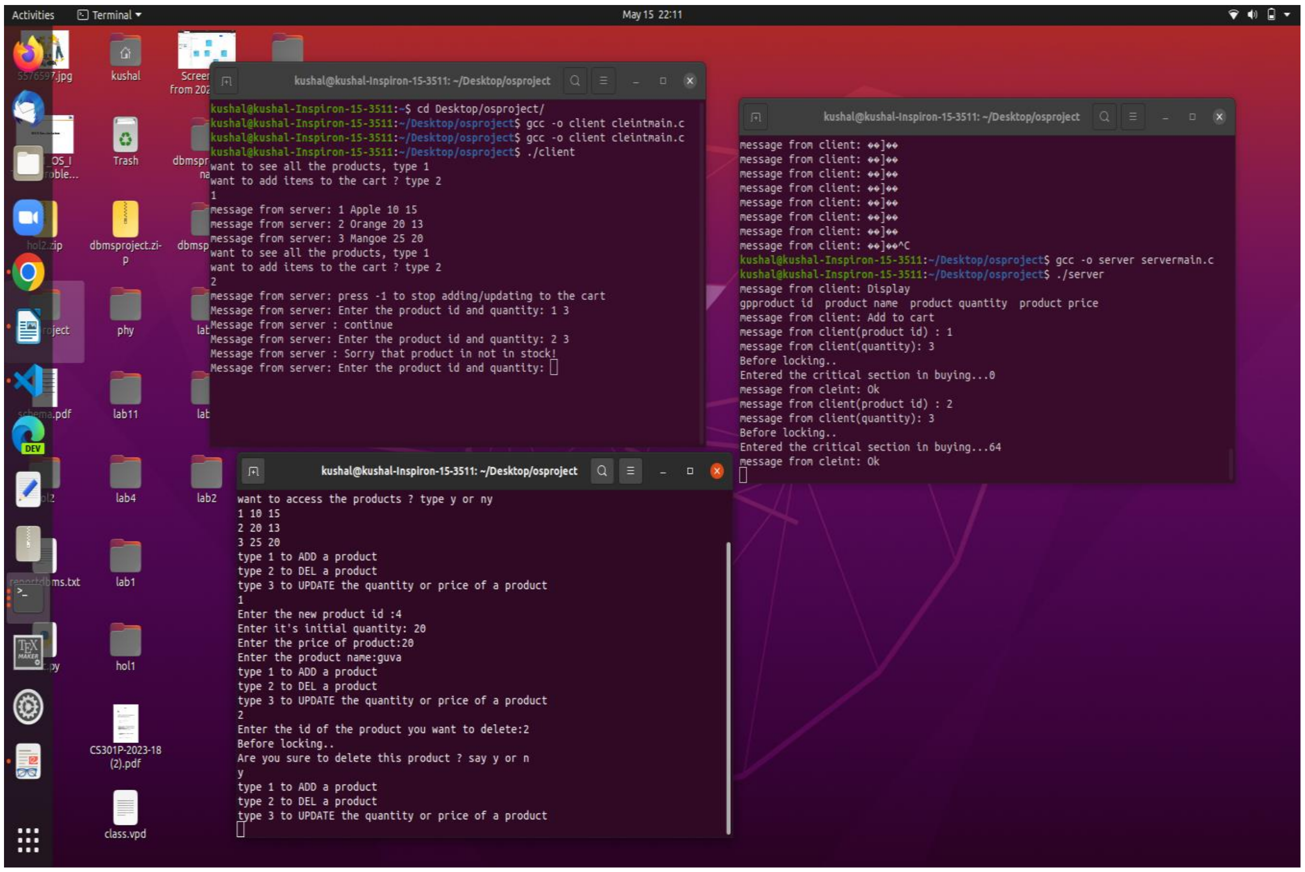
```
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��^C
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o server servermain.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ ./server
message from client: Display
gpproduct id  product name  product quantity  product price
message from client: Add to cart
message from client(product id) : 1
message from client(quantity): 3
Before locking..
Entered the critical section in buying...0
message from cleint: Ok
message from client(product id) : 2
message from client(quantity): 3
Before locking..
```

```
kushal@kushal-Inspiron-15-3511:~$ cd Desktop/osproject/
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o admin server.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ ./admin
Hello Admin!!
want to access the products ? type y or ny
1 10 15
2 20 13
3 25 20
type 1 to ADD a product
type 2 to DEL a product
type 3 to UPDATE the quantity or price of a product
1
Enter the new product id :4
Enter it's initial quantity: 20
Enter the price of product:20
Enter the product name:guva
type 1 to ADD a product
type 2 to DEL a product
type 3 to UPDATE the quantity or price of a product
2
Enter the id of the product you want to delete:2
Before locking..
Are you sure to delete this product ? say y or n
```

1c) Updating:

Similar to delete if we want to update, we need to lock it. Because if the admin is changing to decrease the item. In that time, if we want to add quantity of the item greater than store quantity (before decreasing) it will create a problem that's why we need to use file locking.

Here is the picture of the execution of 1c:



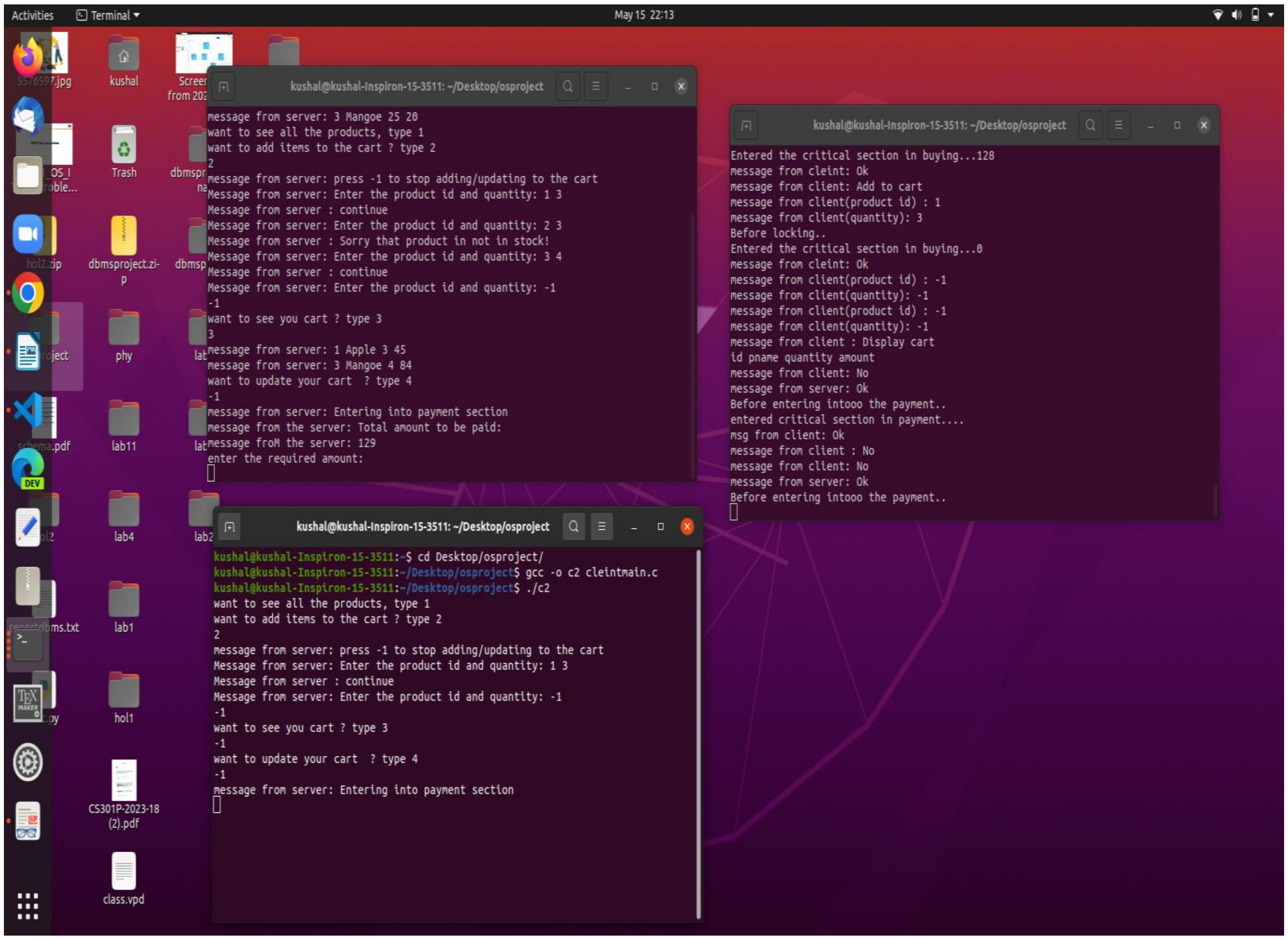
```
kushal@kushal-Inspiron-15-3511:~$ cd Desktop/osproject/
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o client cleintmain.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o client cleintmain.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ ./client
want to see all the products, type 1
want to add items to the cart ? type 2
1
message from server: 1 Apple 10 15
message from server: 2 Orange 20 13
message from server: 3 Mangoe 25 20
want to see all the products, type 1
want to add items to the cart ? type 2
2
message from server: press -1 to stop adding/updating to the cart
Message from server: Enter the product id and quantity: 1 3
Message from server : continue
Message from server: Enter the product id and quantity: 2 3
Message from server : Sorry that product in not in stock!
Message from server: Enter the product id and quantity: 
```

```
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��^C
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o server servermain.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ ./server
message from client: Display
gpproduct id  product name  product quantity  product price
message from client: Add to cart
message from client(product id) : 1
message from client(quantity): 3
Before locking..
Entered the critical section in buying...0
message from cleint: Ok
message from client(product id) : 2
message from client(quantity): 3
Before locking..
Entered the critical section in buying...64
message from cleint: Ok
```

```
want to access the products ? type y or ny
1 10 15
2 20 13
3 25 20
type 1 to ADD a product
type 2 to DEL a product
type 3 to UPDATE the quantity or price of a product
1
Enter the new product id :4
Enter it's initial quantity: 20
Enter the price of product:20
Enter the product name:guva
type 1 to ADD a product
type 2 to DEL a product
type 3 to UPDATE the quantity or price of a product
2
Enter the id of the product you want to delete:2
Before locking..
Are you sure to delete this product ? say y or n
y
type 1 to ADD a product
type 2 to DEL a product
type 3 to UPDATE the quantity or price of a product
```

Use case 2: If two Clients has the same item in the cart, we are using filelocking here because we need to update the quantity for second client to purchase that is after client 1 enter into payment session the file gets locked after purchasing done by client 1, client 2 get into payment session in this way we can reduce the problem(if there are 10 apples if client 1 puts 9 apples into the cart and client 2 also puts 9 apples into the cart if client 1 enters into payment session first it gets 9 apples and updates the store (use of locking) and intimates the client 2 to error in his cart).

here is the picture of 2nd use case:



```
message from server: 3 Mangoe 25 20
want to see all the products, type 1
want to add items to the cart ? type 2
2
message from server: press -1 to stop adding/updating to the cart
Message from server: Enter the product id and quantity: 1 3
Message from server : continue
Message from server: Enter the product id and quantity: 2 3
Message from server : Sorry that product in not in stock!
Message from server: Enter the product id and quantity: 3 4
Message from server : continue
Message from server: Enter the product id and quantity: -1
-1
want to see you cart ? type 3
3
message from server: 1 Apple 3 45
message from server: 3 Mangoe 4 84
want to update your cart  ? type 4
-1
message from server: Entering into payment section
message from the server: Total amount to be paid:
message froM the server: 129
enter the required amount:
```

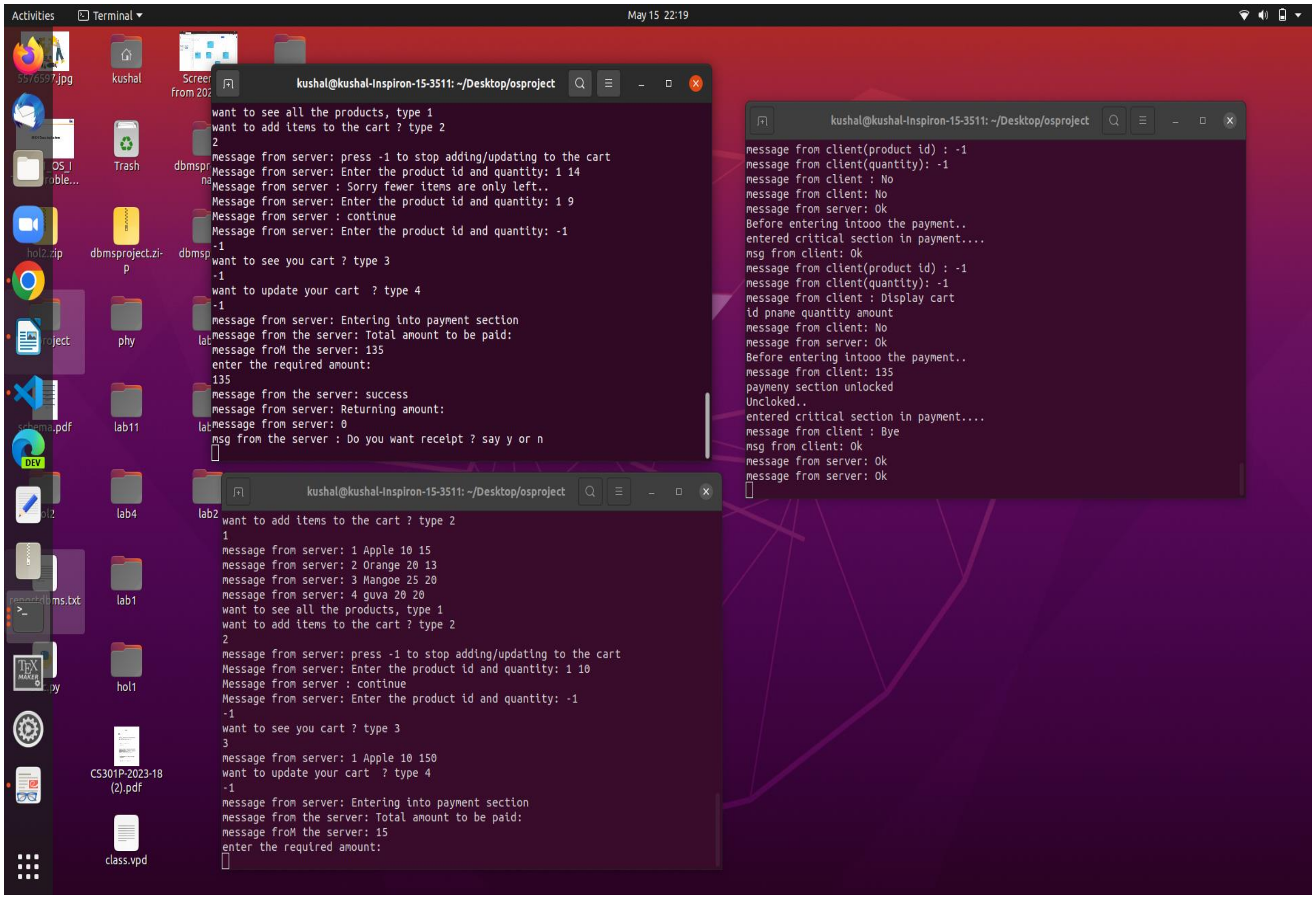
```
Entered the critical section in buying...128
message from cleint: Ok
message from client: Add to cart
message from client(product id) : 1
message from client(quantity): 3
Before locking..
Entered the critical section in buying...0
message from cleint: Ok
message from client(product id) : -1
message from client(quantity): -1
message from client(product id) : -1
message from client(quantity): -1
message from client : Display cart
id pname quantity amount
message from client: No
message from server: Ok
Before entering intooo the payment..
entered critical section in payment....
msg from client: Ok
message from client : No
message from client: No
message from server: Ok
Before entering intooo the payment..
```

```
kushal@kushal-Inspiron-15-3511:~$ cd Desktop/osproject/
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o c2 cleintmain.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ ./c2
want to see all the products, type 1
want to add items to the cart ? type 2
2
message from server: press -1 to stop adding/updating to the cart
Message from server: Enter the product id and quantity: 1 3
Message from server : continue
Message from server: Enter the product id and quantity: -1
-1
want to see you cart ? type 3
-1
want to update your cart  ? type 4
-1
message from server: Entering into payment section
```

3rd use case:

From the above example if the c1 has purchased 9 items and c2 has 9 items in his cart when it enters the payment session it displays only one apple because c1 already purchases 9 apples out of 1 apple.

Here is the picture of 3rd use case:



```
kushal@kushal-Inspiron-15-3511: ~/Desktop/osproject
want to see all the products, type 1
want to add items to the cart ? type 2
2
message from server: press -1 to stop adding/updating to the cart
Message from server: Enter the product id and quantity: 1 14
Message from server : Sorry fewer items are only left..
Message from server: Enter the product id and quantity: 1 9
Message from server : continue
Message from server: Enter the product id and quantity: -1
-1
want to see you cart ? type 3
-1
want to update your cart  ? type 4
-1
message from server: Entering into payment section
message from the server: Total amount to be paid:
message froM the server: 135
enter the required amount:
135
message from the server: success
message from server: Returning amount:
message from server: 0
msg from the server : Do you want receipt ? say y or n
```

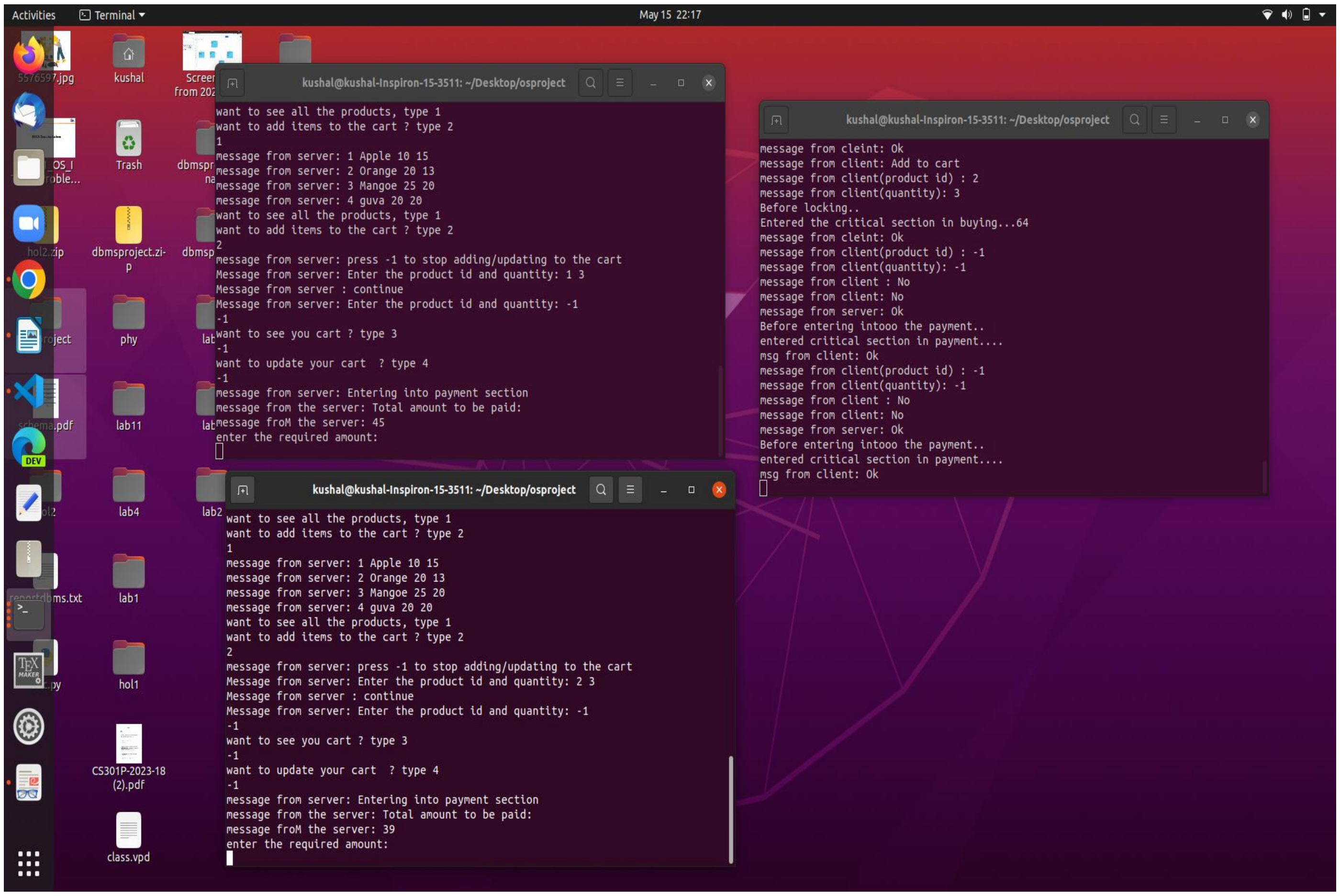
```
kushal@kushal-Inspiron-15-3511: ~/Desktop/osproject
message from client(product id) : -1
message from client(quantity): -1
message from client : No
message from client: No
message from server: Ok
Before entering intooo the payment..
entered critical section in payment....
msg from client: Ok
message from client(product id) : -1
message from client(quantity): -1
message from client : Display cart
id pname quantity amount
message from client: No
message from server: Ok
Before entering intooo the payment..
message from client: 135
paymeny section unlocked
Uncloked..
entered critical section in payment....
message from client : Bye
msg from client: Ok
message from server: Ok
message from server: Ok
```

```
kushal@kushal-Inspiron-15-3511: ~/Desktop/osproject
want to add items to the cart ? type 2
1
message from server: 1 Apple 10 15
message from server: 2 Orange 20 13
message from server: 3 Mangoe 25 20
message from server: 4 guva 20 20
want to see all the products, type 1
want to add items to the cart ? type 2
2
message from server: press -1 to stop adding/updating to the cart
Message from server: Enter the product id and quantity: 1 10
Message from server : continue
Message from server: Enter the product id and quantity: -1
-1
want to see you cart ? type 3
3
message from server: 1 Apple 10 150
want to update your cart  ? type 4
-1
message from server: Entering into payment section
message from the server: Total amount to be paid:
message froM the server: 15
enter the required amount:
```

4th Use Case:

If c1 has different products from c2 there is no use of file locking here. Both can enter the payment session at the same time.

Picture:



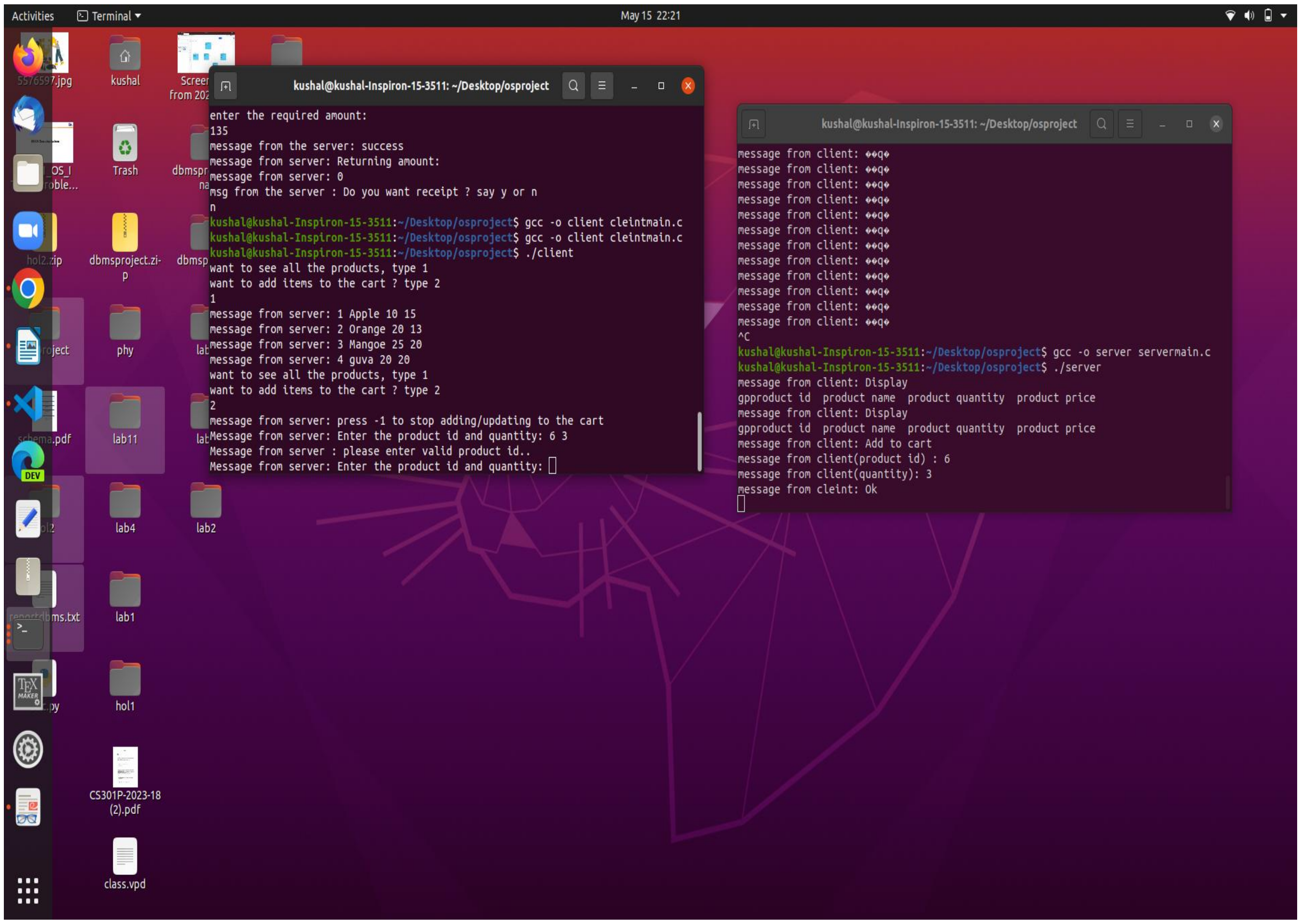
```
want to see all the products, type 1
want to add items to the cart ? type 2
1
message from server: 1 Apple 10 15
message from server: 2 Orange 20 13
message from server: 3 Mangoe 25 20
message from server: 4 guva 20 20
want to see all the products, type 1
want to add items to the cart ? type 2
2
message from server: press -1 to stop adding/updating to the cart
Message from server: Enter the product id and quantity: 1 3
Message from server : continue
Message from server: Enter the product id and quantity: -1
-1
want to see you cart ? type 3
-1
want to update your cart  ? type 4
-1
message from server: Entering into payment section
message from the server: Total amount to be paid:
message froM the server: 45
enter the required amount:
```

```
message from cleint: Ok
message from client: Add to cart
message from client(product id) : 2
message from client(quantity): 3
Before locking..
Entered the critical section in buying...64
message from cleint: Ok
message from client(product id) : -1
message from client(quantity): -1
message from client : No
message from client: No
message from server: Ok
Before entering intooo the payment..
entered critical section in payment....
msg from client: Ok
message from client(product id) : -1
message from client(quantity): -1
message from client : No
message from client: No
message from server: Ok
Before entering intooo the payment..
entered critical section in payment....
msg from client: Ok
```

```
want to see all the products, type 1
want to add items to the cart ? type 2
1
message from server: 1 Apple 10 15
message from server: 2 Orange 20 13
message from server: 3 Mangoe 25 20
message from server: 4 guva 20 20
want to see all the products, type 1
want to add items to the cart ? type 2
2
message from server: press -1 to stop adding/updating to the cart
Message from server: Enter the product id and quantity: 2 3
Message from server : continue
Message from server: Enter the product id and quantity: -1
-1
want to see you cart ? type 3
-1
want to update your cart  ? type 4
-1
message from server: Entering into payment section
message from the server: Total amount to be paid:
message froM the server: 39
enter the required amount:
```

And finally, it will print error messages also:

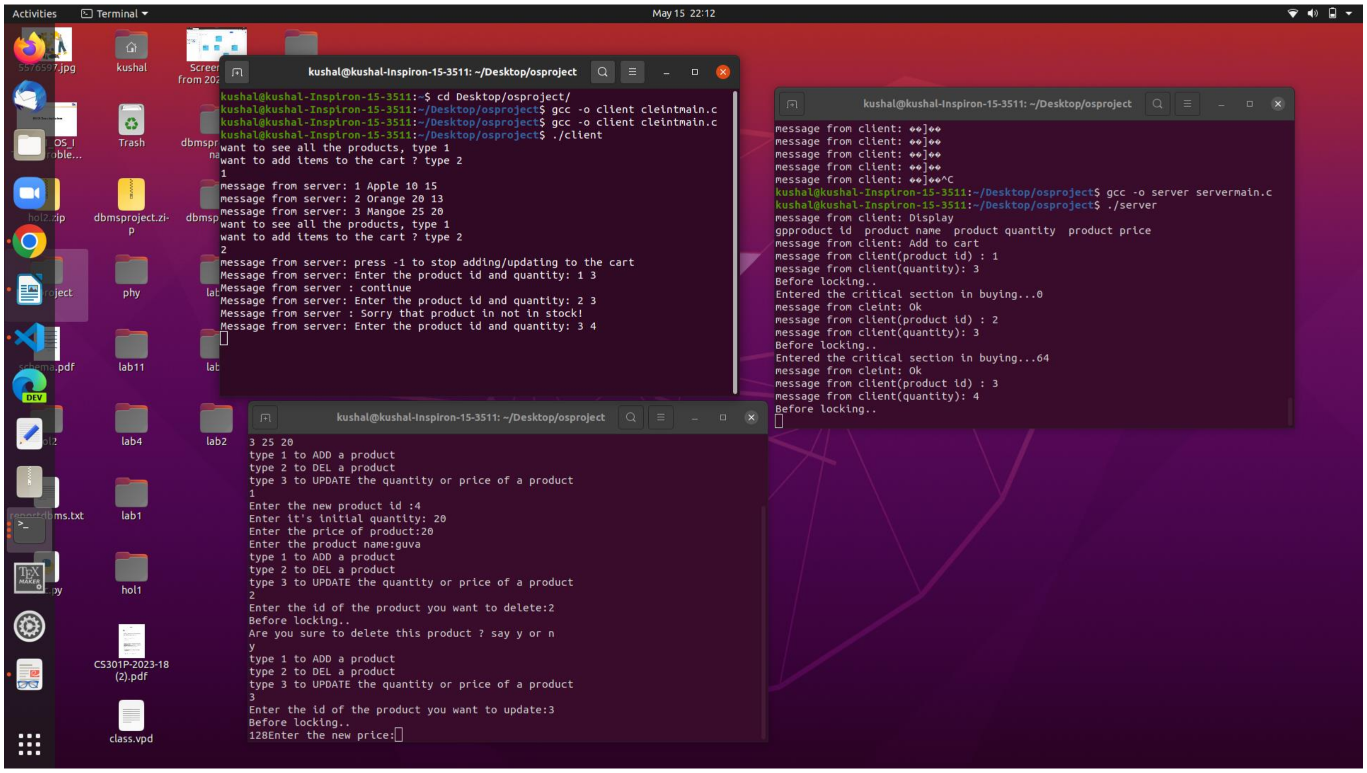
If we enter the product id not available, it prints "invalid product id"



```
enter the required amount:
135
message from the server: success
message from server: Returning amount:
message from server: 0
msg from the server : Do you want receipt ? say y or n
n
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o client cleintmain.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o client cleintmain.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ ./client
want to see all the products, type 1
want to add items to the cart ? type 2
1
message from server: 1 Apple 10 15
message from server: 2 Orange 20 13
message from server: 3 Mangoe 25 20
message from server: 4 guva 20 20
want to see all the products, type 1
want to add items to the cart ? type 2
2
message from server: press -1 to stop adding/updating to the cart
Message from server: Enter the product id and quantity: 6 3
Message from server : please enter valid product id..
Message from server: Enter the product id and quantity: 
```

```
message from client: ��q�
message from client: ��q�
message from client: ��q�
message from client: ��q�
message from client: ��q�
message from client: ��q�
message from client: ��q�
message from client: ��q�
message from client: ��q�
message from client: ��q�
message from client: ��q�
message from client: ��q�
^C
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o server servermain.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ ./server
message from client: Display
gpproduct id  product name  product quantity  product price
message from client: Display
gpproduct id  product name  product quantity  product price
message from client: Add to cart
message from client(product id) : 6
message from client(quantity): 3
message from cleint: Ok
```

If we enter the correct id and it contain 0 quantity, it prints: "out of stock"



```
kushal@kushal-Inspiron-15-3511:~$ cd Desktop/osproject/
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o client cleintmain.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o client cleintmain.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ ./client
want to see all the products, type 1
want to add items to the cart ? type 2
1
message from server: 1 Apple 10 15
message from server: 2 Orange 20 13
message from server: 3 Mangoe 25 20
want to see all the products, type 1
want to add items to the cart ? type 2
2
message from server: press -1 to stop adding/updating to the cart
Message from server: Enter the product id and quantity: 1 3
Message from server : continue
Message from server: Enter the product id and quantity: 2 3
Message from server : Sorry that product in not in stock!
Message from server: Enter the product id and quantity: 3 4
```

```
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��
message from client: ��]��^C
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ gcc -o server servermain.c
kushal@kushal-Inspiron-15-3511:~/Desktop/osproject$ ./server
message from client: Display
gpproduct id  product name  product quantity  product price
message from client: Add to cart
message from client(product id) : 1
message from client(quantity): 3
Before locking..
Entered the critical section in buying...0
message from cleint: Ok
message from client(product id) : 2
message from client(quantity): 3
Before locking..
Entered the critical section in buying...64
message from cleint: Ok
message from client(product id) : 3
message from client(quantity): 4
Before locking..
```

```
3 25 20
type 1 to ADD a product
type 2 to DEL a product
type 3 to UPDATE the quantity or price of a product
1
Enter the new product id :4
Enter it's initial quantity: 20
Enter the price of product:20
Enter the product name:guva
type 1 to ADD a product
type 2 to DEL a product
type 3 to UPDATE the quantity or price of a product
2
Enter the id of the product you want to delete:2
Before locking..
Are you sure to delete this product ? say y or n
y
type 1 to ADD a product
type 2 to DEL a product
type 3 to UPDATE the quantity or price of a product
3
Enter the id of the product you want to update:3
Before locking..
128Enter the new price:
```

If we enter the quantity greater than store quantity, it prints:" Sorry fewer items are only left".

Finally log file(receipt):

<u>Concepts used:</u>

I have utilized socket programming to establish bidirectional communication between client and server for message exchange. Additionally, to ensure data consistency, I have implemented file locking mechanisms in the server application. For instance, when an administrator wishes to update or delete a product, a write lock is applied to that specific record. Similarly, to display all available products to the client, the entire file is read-locked in the server application before sending product details through socket programming. Moreover, before any item is added to the cart, a write lock is applied to that specific item. If any of these locks are not available, the item cannot enter the payment section and waits until the lock is available. Finally, after the client has made the payment, all locks are released.

Github link:https://github.com/kushaldasari/newproject.git