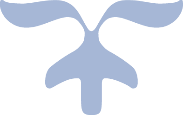


OS Assignment: Client Server programming

Submitted By : Nimish Gaurish Sinai Khandeparkar



May 15, 2023

Roll No: IMT 2021077

**The purpose of this project is to implement a online retail system using client – server programming on linux using C.**

Server has below functionalities implemented to serve client requests in Admin and User Modes.

**Prerequisites to run code:**

1. Have C language installed. Run following while all files are in pwd.
2. First run gcc -o init serv\_init.c and ./init.
3. Then run gcc -o admin admin.c servUtils.c clientUtils.c now ./admin will start the server.
4. Then running gcc -o user user.c clientUtils.c user.c to run multiple users.
5. Code supports one instance of a **particular user** at a time ie no User can login from 2 seperate terminals simultaneously.

NOTE : Names admin.c and user.c are misleading here admin.c is the server and user.c contains user and admin login and menu....

**The functionalities that Server supports when user connects in admin mode are as follows:**

* Creating a product database
* Adding product to the database with its Name, Quantity and Price. Each product can be identified using its Name and Key once created
* Updating Quantity and price of the existing product
* Deleting the product
* Displaying the products existing in the database
* Providing dump of all products existing in the database when server exits in a text file.

In addition, the server also has functionality of displaying all the products on admin console.

**The functionalities the server supports when user connects in user mode are as follows:**

* Checking availability of product: Responds to the user positively if the requested quantity of products is available when requested. Send a negative response if the product is not available or when requested quantity is not available to sell.
* Buy Functionality: Once client completes adding all products in cart and send a buy request, the server again validates if all the products are still available in database in requested quantity. If yes, it reduces the quantity of the selected products in the database and sends back details to client for generation of the bill.

**Note that all write operations are implemented with write locks to avoid multiple clients modifying the product data at the same time.**

**The client Side Functionalities:**

The client on the other has separate functionality for admin and user. When the client is started, it provides below options

* To login as “admin”
* To login as “user”

Login functionality not implemented as not mentioned as a requirement....

**When logged in as admin, it provides below functionalities:**

1. Add Product: Takes Product Name, Quantity and Price and sends to server. Product is added if it does not exist in database. Else it sends an error message that product already exists and prompts goes back to previous screen to allow admin user to update product instead. When a product is added, it also creates an unique key for the product which can be used to update the product.
2. Display Products: This fetches all the products from server and displays on user screen.
3. Update Product by Key: This command takes product key, quantity and price and sends the same to server to update the same in database. If the key does not exist in database, server returns error message that product does not exist. Else it updates the product details as requested
4. Update Product by Name: This is similar to update the product by key, except that the operation of update to product details is done by searching the name instead of key.
5. Delete Product: This command takes the product name and send sends to server. If product exists, it deletes the same and returns and acknowledgement. If Product requested does not exists in server database, client receives product does not exists message and displays the same accordingly
6. Quit : This command ensures graceful exit of client post release of resources like locks, socket IDs etc.

**When logged in as user, client it has user login functionality.**

It takes the user name and password and checks in server database, validates the user name and password and completes the login functionality.

**It provides below functionalities for logged in user:**

1. Add to Cart: This command takes Product name and quantity as input. It send the same to server. Server checks if available quantity is available or not and responds positively if the quantity is available. Client creates a local cart for the user and adds the product to it along with the quantity reuested and the price returned by server. If requested product or quantity is not available, client receives negative acknowledgement that requested product or quantity of product is not available and displays it back to client screen.
2. Display Cart : This command displays all products added to the cart along with their prices and requested quantities. It also summarizes the total quantity and total price.
3. Update Cart: This command also takes product name and quantity as input. If sends the same to server and revalidates if the requested functionality is available. If server responds with positive acknowledgement, client modifies the quantity in cart accordingly. If requested product or quantity is not available, client receives negative acknowledgement that requested product or quantity of product is not available and displays it back to client screen.
4. Delete from Cart: This command takes the product name as input and deletes the same from cart
5. Buy : This Option helps user to check out the products added to cart. It sends the details of all products in the cart to server to revalidate the availability. Once server returns and acknowledgement, it displays the total price to pay and prompts the user to pay the same. Once user keys in the exact amount, the same is sent to server. Server then reduces the product count and sends back acknowledgment. Client uses the cart information to generate a receipt for the user.
6. Quit : This command ensures graceful exit of client post release of resources like locks, socket IDs etc.