

## Application #2: Client Server Trading System using socket programming

A Client-Server Based Trading System is to be designed with the following specifications. There will be a set of traders who will trade with each other in the automated system. There will be a Server which will register requests from traders for buying and selling quantities of Items. The Server will also match the buy with the sell requests from different traders based on certain price rules (as listed below). Traders will log on to the trading system through the trading client (assume Trader ID and password is stored in a file). They will have the option to view the currently available items (for buy/sell), their quantities and their prices. They will also send requests for buying and selling items and specify the quantity and price. Traders will also have the option to view their matched trades at any time. There are ten known items which the traders can trade in with their codes from 1 to 10. There will be a maximum of 5 traders (with codes from 1 to 5) who can log on to the system and work. One trader should work from one client at a time only.

The functionalities of any client will be:

- **Login to the System:** The trader will execute the client, give the trader number and will be logged in. After that he/she will have the following options in a menu. Several clients will login (from different terminals) and assumed they don't trade simultaneously to reduce the complexity.
- **Send Buy Request:** The trader will send a buy request by stating the item code, the quantity and unit price.
- **Send Sell request:** The trader will send a sell request by stating the item code, the quantity and unit price.
- **View Order Status:** The Trader can view the position of buy and sell orders in the system. This will display the current best sell (least price) and the best buy (max price) for each item and their quantities.
- **View Trade Status:** The trader can view his/her matched trades. This will provide the trader with the details of what orders were matched, their quantities, prices and counterparty code.

There will be only one server which will be running and perform the functions of order processing and trade matching in addition to acknowledging logins by clients and servicing their requests. The order processing will be as follows. There will be a buy and a sell order queue for each item. On receiving buy/sell order request from a trader, the server will put it in the appropriate order queue. If there is a possibility of a trade match, then that trade match will take place, the traded items will be appropriately updated and the result of the trade along with the details of the counterparties, item, quantity and price will be stored in the traded set. The matching rule is as follows:

1. On a buy Request at price  $P$  and quantity  $Q$  of an item  $I$ , the server will check if there is any pending sell order for the same item at price  $P' \leq P$ .
2. Among all such pending sell orders, the match will be made with the one having the least selling price.
3. If both have same quantity, i.e.,  $Q' = Q$ , then both these orders will be removed from their respective queues and the result will be put into the traded set.
4. If  $Q' > Q$  then the buy request will be fully traded and the remaining part, i.e.,  $Q' - Q$  of the sell order will remain in the sell queue at the same price  $P'$ .
5. On the other hand, if  $Q' < Q$  then the sell order will be fully traded and the remaining buy order will be tested for more matches.
6. If the buy order cannot be matched, it will be put into the buy queue.
7. A similar rule will apply for a sell request and all these requests will be handled in a FCFS basis.

You should accept the IP Address and Port number from the command line (Don't use a hard-coded port number). \*Please make necessary and valid assumptions whenever required.

Prototype for command line is as follows:

### Prototypes for Client and Server

**Client:** <executable code><Server IP Address><Server Port number>

**Server:** <executable code><Server Port number>