

# CSC/ECE 573 Internet Protocols, Sections 001

## 2024 Fall Socket Programming Assignment

**Date:** 18th October 2024

### Team Members

- Pravallika Vasireddy (pvasire2)
- Karthik Masineni (kmasine)

### Project Overview

Auction\_Project1 is a sealed-bid auction system using TCP sockets. The project simulates an auction process where sellers submit items for auction, and buyers place bids. The server (Auctioneer) manages the entire process by handling multiple connections from both sellers and buyers.

This project demonstrates:

- TCP socket programming for client-server communication.
- Multi-threading to handle multiple clients.
- A sealed-bid auction system with support for first-price and second-price auctions.

### How to Run

#### Prerequisites

- Python 3.x installed.
- Socket programming support (available by default in Python).

#### 1. Starting the Server

To start the Auctioneer server:

```
python3 auc_server.py <port_number>
```

Example:

```
python3 auc_server.py 12345
```

This will start the server on the specified port. The server will wait for a seller to connect.

## 2. Running the Client

To start the client:

```
python3 auc_client.py <server_ip> <port_number>
```

Example:

```
python3 auc_client.py 127.0.0.1 12345
```

The client will connect to the Auctioneer server. Based on the state of the server, the client will either act as a Seller or a Buyer.

Seller mode

The seller provides auction details in the following format:

```
<auction_type> <min_price> <number_of_bidders> <item_name>
```

Example:

```
1 100 3 Shoes
```

- auction\_type: 1 for first-price auction, 2 for second-price auction.
- min\_price: Minimum price for the item (e.g., 100).
- number\_of\_bidders: The expected number of bidders (e.g., 3).
- item\_name: Name of the item being auctioned (e.g., Shoes).

Once the auction details are submitted, the auction will start after all buyers connect.

Buyer mode

Buyers connect and wait until all other buyers have connected. Once the auction starts, buyers can submit their bids by entering a positive integer.

Example bid:

```
Enter your bid: 150
```

## 3. Auction Process

The auction proceeds based on the number of buyers and the auction type.

- If the first-price auction is selected, the highest bidder wins and pays their bid.
- If the second-price auction is selected, the highest bidder wins but pays the second-highest bid.

After all bids are submitted, the server informs the seller and buyers of the result, then closes the connections.

#### 4. Example Outputs

The server output below shows a seller connecting and starting an auction for the item "shoes" with a minimum price of 20 and 2 bidders. Two buyers connected and placed bids of 100 and 200. The highest bid was 200, and the item was sold. The server is now ready for a new auction.

```
(base) pravallikavasireddy@PravallikasAir5 IP project % python3 server.py 2000
Auctioneer is ready for hosting auctions!
Seller connected from ('127.0.0.1', 56269)
>> New seller thread spawned
Auction request received. Now waiting for buyers...
Buyer 1 is connected from 127.0.0.1:56270
Buyer 2 is connected from 127.0.0.1:56271
Requested number of bidders arrived. Let's start bidding!
>> New bidding thread spawned
Buyer 1 bid: 100
Buyer 2 bid: 200
Item Sold! The highest bid is $200.
Auctioneer is ready for hosting auctions!
█
```

The seller output below shows that the seller got connected to the server and provided auction details for "shoes" with a minimum price of 20 and 2 bidders. The auction type is 1. The auction started, and the seller was notified that the item was sold for \$200. The server then disconnected the seller.

```
(base) pravallikavasireddy@PravallikasAir5 IP project % python3 client.py 127.0.0.1 2000
Connected to the Auctioneer server

Your role is: [Seller]
Please submit an auction request:
Enter auction details <auction_type> <min_price> <number_of_bidders> <item_name>: 1 20 2 shoes
Server: Auction Start
Auction Finished!
Success! Your item shoes has been sold for $200.
Disconnecting from the Auctioneer server. Auction is Over!
(base) pravallikavasireddy@PravallikasAir5 IP project % █
```

The buyer output below shows that the buyer has been successfully connected, placed a bid of 200, and won the auction for the item "shoes." The final payment due is \$200, and the connection is closed. Note that this is for auction type 1.

```
(base) pravallikavasireddy@PravallikasAir5 IP project % python3 client.py 127.0.0.1 2000
Connected to the Auctioneer server

Your role is: [Buyer]

The bidding has started!
Please submit your bid:
Enter your bid: 200
Server: Bid received. Please wait...
Auction Finished!
Success! You won the item shoes. Your payment due is $200.
Disconnecting from the Auctioneer server. Auction is Over!
(base) pravallikavasireddy@PravallikasAir5 IP project % █
```

The buyer below placed a bid of 100 but lost the auction. After submitting the bid, they were informed that they did not win, and the connection was closed.

```
(base) pravallikavasireddy@PravallikasAir5 IP project % python3 client.py 127.0.0.1 2000
Connected to the Auctioneer server

Your role is: [Buyer]
The auctioneer is still waiting for other buyers to connect...
The bidding has started!
Please submit your bid:
Enter your bid: 100
Server: Bid received. Please wait...
Auction Finished!
Unfortunately you did not win the last round.
Disconnecting from the Auctioneer server. Auction is Over!
(base) pravallikavasireddy@PravallikasAir5 IP project %
```

Note that if the auction type was chosen as 2, the second highest bid amount would be payable by the winner of the bid. For instance, the below images showcase the seller requesting for type2 bid for pants with minimum value as 20 and 2 bidders. One buyer bid the pants for 100 while the other bid 200. The winner was the buyer who bid 200 but the payable amount is 100. This is also notified to the seller.

```
(base) pravallikavasireddy@PravallikasAir5 IP project % python3 client.py 127.0.0.1 2000
Connected to the Auctioneer server

Your role is: [Seller]
Please submit an auction request:
Enter auction details <auction_type> <min_price> <number_of_bidders> <item_name>: 2 20 2 pants
Server: Auction Start
Auction Finished!
Success! Your item pants has been sold for $100.
Disconnecting from the Auctioneer server. Auction is Over!
(base) pravallikavasireddy@PravallikasAir5 IP project %
```

```
Disconnecting from the Auctioneer server. Auction is Over!
(base) pravallikavasireddy@PravallikasAir5 IP project % python3 client.py 127.0.0.1 2000
Connected to the Auctioneer server

Your role is: [Buyer]

The bidding has started!
Please submit your bid:
Enter your bid: 200
Server: Bid received. Please wait...
Auction Finished!
Success! You won the item pants. Your payment due is $100.
Disconnecting from the Auctioneer server. Auction is Over!
(base) pravallikavasireddy@PravallikasAir5 IP project %
```

```
(base) pravallikavasireddy@PravallikasAir5 IP project % python3 client.py 127.0.0.1 2000
Connected to the Auctioneer server

Your role is: [Buyer]
The auctioneer is still waiting for other buyers to connect...
The bidding has started!
Please submit your bid:
Enter your bid: 100
Server: Bid received. Please wait...
Auction Finished!
Unfortunately you did not win the last round.
Disconnecting from the Auctioneer server. Auction is Over!
```

The below example shows the case when the highest bidder did not reach the minimum price value set by the seller. In this case, the auction fails and the seller is notified. The servers starts to listen to server for another auction.

```
(base) pravallikavasireddy@PravallikasAir5 IP project %
python3 server.py 2000
Auctioneer is ready for hosting auctions!
Seller connected from ('127.0.0.1', 56308)
>> New seller thread spawned
Auction request received. Now waiting for buyers...
Buyer 1 is connected from 127.0.0.1:56311
Buyer 2 is connected from 127.0.0.1:56312
Requested number of bidders arrived. Let's start bidding!
>> New bidding thread spawned
Buyer 1 bid: 10
Buyer 2 bid: 20
Auction failed as the minimum price was not reached!
Auctioneer is ready for hosting auctions!
```

```
(base) pravallikavasireddy@PravallikasAir5 IP project % python3 client.py 127.0.0.1 2001
Connected to the Auctioneer server

Your role is: [Seller]
Please submit an auction request:
Enter auction details <auction_type> <min_price> <number_of_bidders> <item_name>: 1 100 2 shirts
Server: Auction Start
Unfortunately the item was not sold.
(base) pravallikavasireddy@PravallikasAir5 IP project %
```

```
(base) pravallikavasireddy@PravallikasAir5 IP project % python3 client.py 127.0.0.1 2001
Connected to the Auctioneer server

Your role is: [Buyer]
The auctioneer is still waiting for other buyers to connect...
The bidding has started!
Please submit your bid:
Enter your bid: 10
Server: Bid received. Please wait...
Unfortunately you have not won the auction.
(base) pravallikavasireddy@PravallikasAir5 IP project %
```

```

Unfortunately you have not won the auction!
(base) pravallikavasireddy@PravallikasAir5 IP project % python3 client.py 127.0.0.1 2001
Connected to the Auctioneer server

Your role is: [Buyer]

The bidding has started!
Please submit your bid:
Enter your bid: 20
Server: Bid received. Please wait...
Unfortunately you have not won the auction.
(base) pravallikavasireddy@PravallikasAir5 IP project %

```

## Extra Credit

1. To achieve this goal, we would choose second-price auction.
2. In a second price auction, the strategy for each buyer is to bid their true valuation (given by  $v_i$ ). The highest bidder wins but only pays the second highest bid. This will provide incentive to the buyers to bid their true valuation as there is incentive to avoid underbidding or overbidding. Overbidding could result in lower utility (pay more than they need to) and underbidding can risk losing the item

In case of a first price auction, buyers can bid less than their true valuation in order to maximize their utility. Since the winner pays exactly what they bid, the buyers can win the auction at a lower price, leading to underbidding. In such a case, the seller wouldn't know the buyers' true valuation.

By choosing the second-price auction, we are not only gathering honest valuations from buyers but also selling the item at a reasonable price.

### First-price Auction:

- If a buyer wins by bidding their true valuation ( $v_i$ ) and pays that price, their utility is:  $u = v_i - v_i = 0$
- If they bid below their true valuation ( $p < v_i$ ) and win, their utility is:  $u = v_i - p > 0$
- If they bid too low and lose, their utility is 0.

### Second-price Auction:

- If a buyer bids wins, they pay the second-highest bid ( $b_2$ ), so their utility is:  $u = v_i - b_2 > 0$
- If they bid below their true valuation ( $b < v_i$ ) and lose, their utility is 0.

3. Execution:

### First-price Auction:

In this case, if we consider the scenario where the seller asks for the price of 100 for nike shoes, buyer1 bid 95 and buyer2 bid 105. The winner - buyer2 has to pay 105.

```
(base) pravallikavasireddy@PravallikasAir5 IP project % python3 au_server.py 2001
Auctioneer is ready for hosting auctions!
Seller connected from ('127.0.0.1', 56423)
>> New seller thread spawned
Auction request received. Now waiting for buyers...
Buyer 1 is connected from 127.0.0.1:56425
Buyer 2 is connected from 127.0.0.1:56426
Requested number of bidders arrived. Let's start bidding!
>> New bidding thread spawned
Buyer 1 bid: 95
Buyer 2 bid: 105
Item Sold! The highest bid is $105.
Auctioneer is ready for hosting auctions!
```

```
(base) pravallikavasireddy@PravallikasAir5 IP project % python3 au_client.py 127.0.0.1 2001
Connected to the Auctioneer server

Your role is: [Seller]
Please submit an auction request:
Enter auction details <auction_type> <min_price> <number_of_bidders> <item_name>: 1 100 2 nike_shoes
Server: Auction Start
Auction Finished!
Success! Your item nike_shoes has been sold for $105.
Disconnecting from the Auctioneer server. Auction is Over!
(base) pravallikavasireddy@PravallikasAir5 IP project %
```

```
(base) pravallikavasireddy@PravallikasAir5 IP project % python3 au_client.py 127.0.0.1 2001
Connected to the Auctioneer server

Your role is: [Buyer]

The bidding has started!
Please submit your bid:
Enter your bid: 105
Server: Bid received. Please wait...
Auction Finished!
Success! You won the item nike_shoes. Your payment due is $105.
Disconnecting from the Auctioneer server. Auction is Over!
(base) pravallikavasireddy@PravallikasAir5 IP project %
```

## Second-price Auction:

In this case, buyer1 bids for 110 and buyer2 bids 120. Buyer2 wins and pays 110. Buyers are encouraged to bid the true value in order to win.

```
(base) pravallikavasireddy@PravallikasAir5 IP project % python3 au_client.py 127.0.0.1 2000
Connected to the Auctioneer server

Your role is: [Seller]
Please submit an auction request:
Enter auction details <auction_type> <min_price> <number_of_bidders> <item_name>: 2 100 2 nike_shoes
Server: Auction Start
Auction Finished!
Success! Your item nike_shoes has been sold for $110.
Disconnecting from the Auctioneer server. Auction is Over!
(base) pravallikavasireddy@PravallikasAir5 IP project %
```

```
(base) pravallikavasireddy@PravallikasAir5 IP project % python3 au_client.py 127.0.0.1 2000
Connected to the Auctioneer server
```

```
Your role is: [Buyer]
```

```
The bidding has started!
```

```
Please submit your bid:
```

```
Enter your bid: 120
```

```
Server: Bid received. Please wait...
```

```
Auction Finished!
```

```
Success! You won the item nike_shoes. Your payment due is $110.
```

```
Disconnecting from the Auctioneer server. Auction is Over!
```

```
(base) pravallikavasireddy@PravallikasAir5 IP project %
```

```
(base) pravallikavasireddy@PravallikasAir5 IP project % python3 au_client.py 127.0.0.1 2000
Connected to the Auctioneer server
```

```
Your role is: [Buyer]
```

```
The auctioneer is still waiting for other buyers to connect...
```

```
The bidding has started!
```

```
Please submit your bid:
```

```
Enter your bid: 110
```

```
Server: Bid received. Please wait...
```

```
Auction Finished!
```

```
Unfortunately you did not win the last round.
```

```
Disconnecting from the Auctioneer server. Auction is Over!
```

```
(base) pravallikavasireddy@PravallikasAir5 IP project %
```

```
(base) pravallikavasireddy@PravallikasAir5 IP project % python3 au_server.py 2000
Auctioneer is ready for hosting auctions!
```

```
Seller connected from ('127.0.0.1', 56452)
```

```
>> New seller thread spawned
```

```
Auction request received. Now waiting for buyers...
```

```
Buyer 1 is connected from 127.0.0.1:56453
```

```
Buyer 2 is connected from 127.0.0.1:56454
```

```
Requested number of bidders arrived. Let's start bidding!
```

```
>> New bidding thread spawned
```

```
Buyer 1 bid: 110
```

```
Buyer 2 bid: 120
```

```
Item Sold! The highest bid is $120.
```

```
Auctioneer is ready for hosting auctions!
```