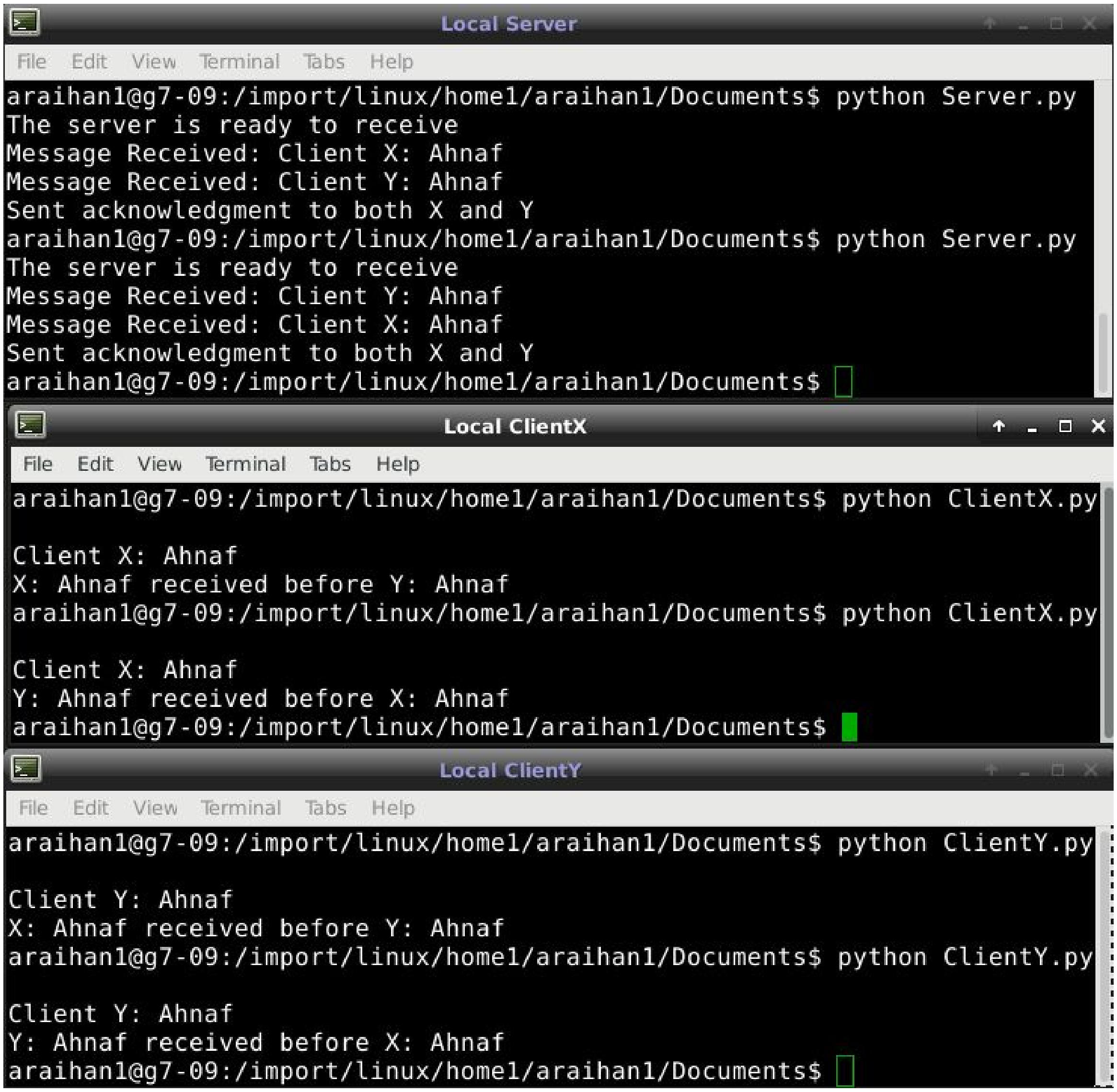
Ahnaf Raihan
CS428 - Assignment 2

Design:

For this program, I created a TCP socket in the server. In the server, I waited to receive connections from 2 clients (Client X and Client Y). Once these connections were created, the server received two messages, and took appropriate action according to the order the messages were received. If Client X’s message was received first, the server sent out a message to both Clients noting this, and then closed connections. If Client Y’s message was received first, it sent the opposite message. Client X and Client Y were virtually the same, except for the message they sent. Both clients created a TCP connection with the server, sent their messages, and waited to receive a response before closing connection.

Demo:

Local


Local Server
File Edit View Terminal Tabs Help
Local ClientX
File Edit View Terminal Tabs Help
Local ClientY
File Edit View Terminal Tabs Help

```
araihan1@g7-09:/import/linux/home1/araihan1/Documents$ python Server.py
The server is ready to receive
Message Received: Client X: Ahnaf
Message Received: Client Y: Ahnaf
Sent acknowledgment to both X and Y
araihan1@g7-09:/import/linux/home1/araihan1/Documents$ python Server.py
The server is ready to receive
Message Received: Client Y: Ahnaf
Message Received: Client X: Ahnaf
Sent acknowledgment to both X and Y
araihan1@g7-09:/import/linux/home1/araihan1/Documents$ 
```

```
araihan1@g7-09:/import/linux/home1/araihan1/Documents$ python ClientX.py

Client X: Ahnaf
X: Ahnaf received before Y: Ahnaf
araihan1@g7-09:/import/linux/home1/araihan1/Documents$ python ClientX.py

Client X: Ahnaf
Y: Ahnaf received before X: Ahnaf
araihan1@g7-09:/import/linux/home1/araihan1/Documents$ 
```

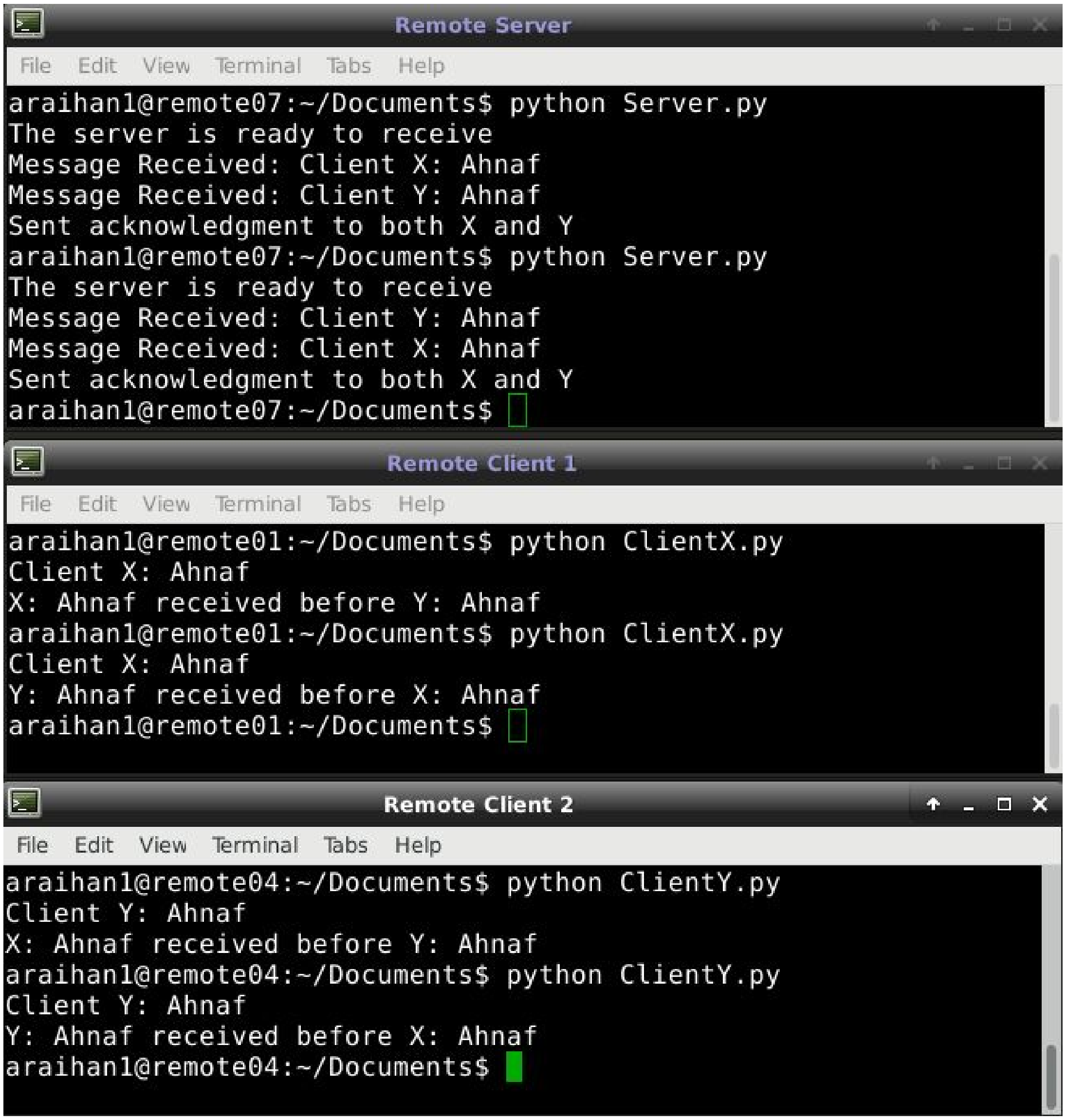
```
araihan1@g7-09:/import/linux/home1/araihan1/Documents$ python ClientY.py

Client Y: Ahnaf
X: Ahnaf received before Y: Ahnaf
araihan1@g7-09:/import/linux/home1/araihan1/Documents$ python ClientY.py

Client Y: Ahnaf
Y: Ahnaf received before X: Ahnaf
araihan1@g7-09:/import/linux/home1/araihan1/Documents$ 
```

Note: Terminals should read “Local Client 1” and “Local Client 2”, not “Local ClientX” and “Local ClientY”

Remote



```
araihan1@remote07:~/Documents$ python Server.py
The server is ready to receive
Message Received: Client X: Ahnaf
Message Received: Client Y: Ahnaf
Sent acknowledgment to both X and Y
araihan1@remote07:~/Documents$ python Server.py
The server is ready to receive
Message Received: Client Y: Ahnaf
Message Received: Client X: Ahnaf
Sent acknowledgment to both X and Y
araihan1@remote07:~/Documents$
```

```
araihan1@remote01:~/Documents$ python ClientX.py
Client X: Ahnaf
X: Ahnaf received before Y: Ahnaf
araihan1@remote01:~/Documents$ python ClientX.py
Client X: Ahnaf
Y: Ahnaf received before X: Ahnaf
araihan1@remote01:~/Documents$
```

```
araihan1@remote04:~/Documents$ python ClientY.py
Client Y: Ahnaf
X: Ahnaf received before Y: Ahnaf
araihan1@remote04:~/Documents$ python ClientY.py
Client Y: Ahnaf
Y: Ahnaf received before X: Ahnaf
araihan1@remote04:~/Documents$
```

To run the program on “Remote”, I had to change the IP Address in the clients from localhost to the IP of the server

Failures and Improvements:

A possible improvement with this program is the case in which the program stops unexpectedly. This will leave our address in use until a hard reboot takes place, which is a problem especially if the operation were to scale up. Adding safety checks with a try/catch could be beneficial. Additionally, after each run of the program, port number must be manually changed. This could be improved by setting port to 0 to find an open port automatically or allow the program to reuse ports.