Vigomar Kim Algador

CPE 138 – 03

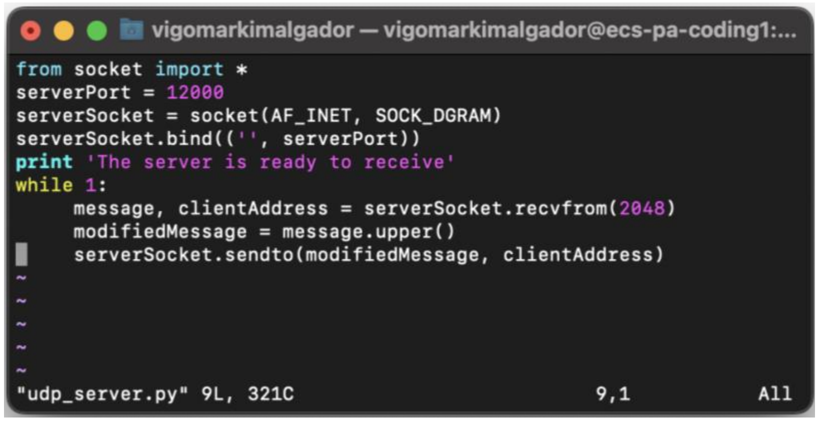
Professor Jun Dai

26 February 2023

SOCKET PROGRAMMING ASSIGNMENT 1

For this assignment, we were assigned to apply the conceptual knowledge with socket to build client/server applications to communicate. This assignment is divided into two: UDP and TCP. Within the assignment, we must input lowercase sentence and receive the same input in all uppercase.

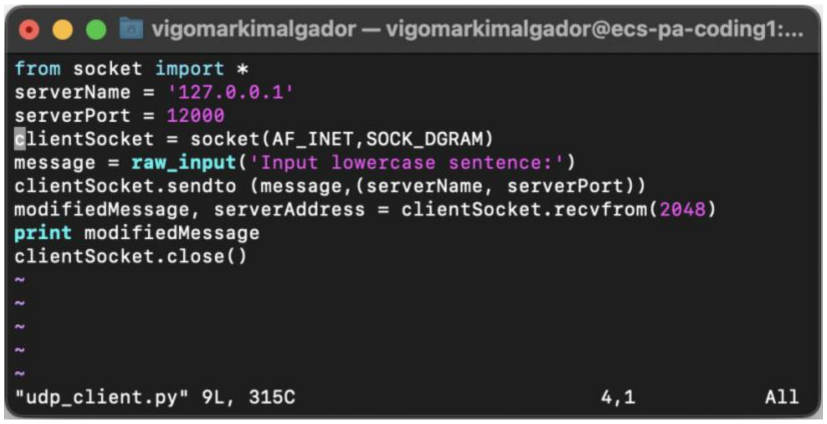
For the first part, we are required to use UDP. Below is the source code for UDP client and server.



vigomarkimalgador — vigomarkimalgador@ecs-pa-coding1:...

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('', serverPort))
print 'The server is ready to receive'
while 1:
     message, clientAddress = serverSocket.recvfrom(2048)
     modifiedMessage = message.upper()
     serverSocket.sendto(modifiedMessage, clientAddress)
~
~
~
~
~
"udp_server.py" 9L, 321C                          9,1           All
```

Figure 1. UDP server.py source code

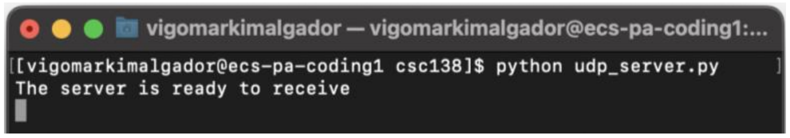


vigomarkimalgador — vigomarkimalgador@ecs-pa-coding1:...

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET,SOCK_DGRAM)
message = raw_input('Input lowercase sentence:')
clientSocket.sendto (message,(serverName, serverPort))
modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
print modifiedMessage
clientSocket.close()
~
~
~
~
~
"udp_client.py" 9L, 315C                           4,1           All
```

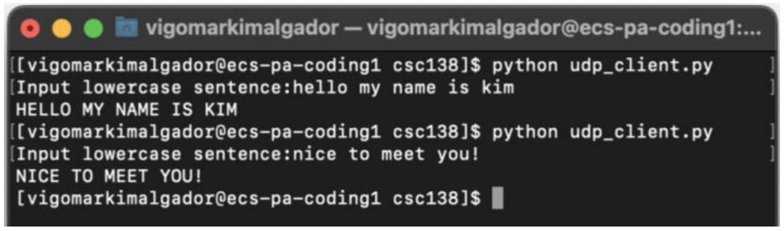
Figure 2. UDP client.py source code

After that, we are required to run first the server and then run the client showed below.



```
vigomarkimalgador — vigomarkimalgador@ecs-pa-coding1:...
[vigomarkimalgador@ecs-pa-coding1 csc138]$ python udp_server.py
The server is ready to receive
```

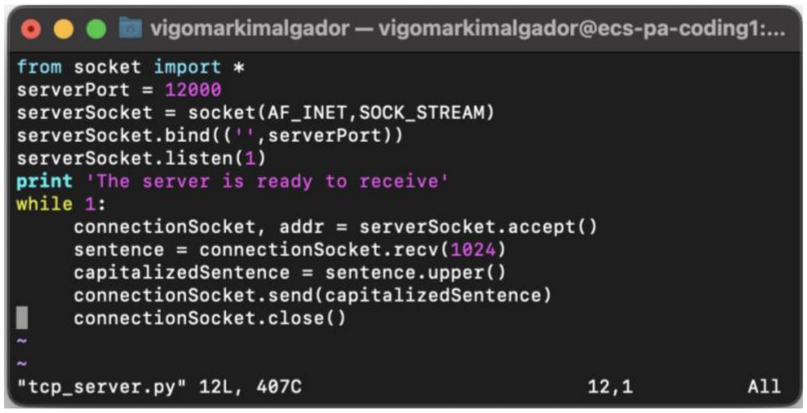
Figure 3. UDP Server-side snapshot



```
vigomarkimalgador — vigomarkimalgador@ecs-pa-coding1:...
[vigomarkimalgador@ecs-pa-coding1 csc138]$ python udp_client.py
Input lowercase sentence:hello my name is kim
HELLO MY NAME IS KIM
[vigomarkimalgador@ecs-pa-coding1 csc138]$ python udp_client.py
Input lowercase sentence:nice to meet you!
NICE TO MEET YOU!
[vigomarkimalgador@ecs-pa-coding1 csc138]$
```

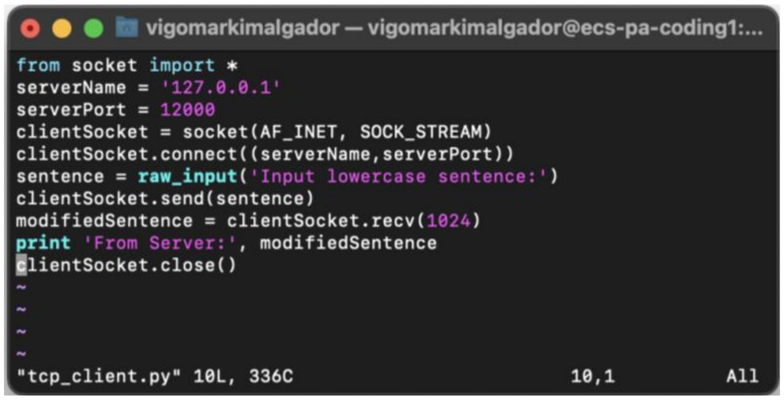
Figure 4. UDP client-side snapshot

The second part, we are required to program with TCP. Below is the source code for TCP client and server.



```
vigomarkimalgador — vigomarkimalgador@ecs-pa-coding1:...
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind(('',serverPort))
serverSocket.listen(1)
print 'The server is ready to receive'
while 1:
     connectionSocket, addr = serverSocket.accept()
     sentence = connectionSocket.recv(1024)
     capitalizedSentence = sentence.upper()
     connectionSocket.send(capitalizedSentence)
     connectionSocket.close()
~
~
"tcp_server.py" 12L, 407C                          12,1          All
```

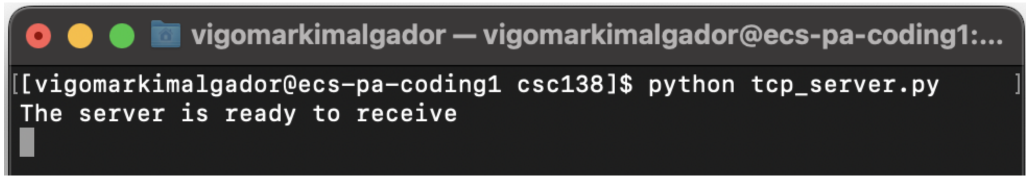
Figure 5. TCP server.py source code



```
vigomarkimalgador — vigomarkimalgador@ecs-pa-coding1:...
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = raw_input('Input lowercase sentence:')
clientSocket.send(sentence)
modifiedSentence = clientSocket.recv(1024)
print 'From Server:', modifiedSentence
clientSocket.close()
~
~
~
~
"tcp_client.py" 10L, 336C                          10,1          All
```

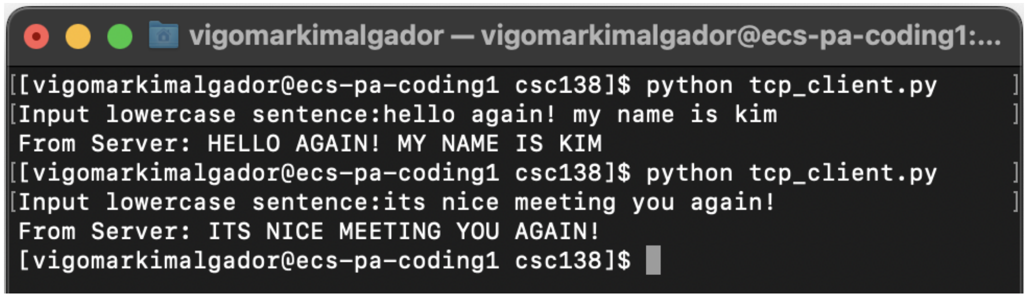
Figure 6. TCP client.py source code

After that, we need to do the same thing to run both client and server.



```
[vigomarkimalgador@ecs-pa-coding1 csc138]$ python tcp_server.py
The server is ready to receive
```

Figure 7. TCP Server-side snapshot



```
[vigomarkimalgador@ecs-pa-coding1 csc138]$ python tcp_client.py
Input lowercase sentence:hello again! my name is kim
From Server: HELLO AGAIN! MY NAME IS KIM
[vigomarkimalgador@ecs-pa-coding1 csc138]$ python tcp_client.py
Input lowercase sentence:its nice meeting you again!
From Server: ITS NICE MEETING YOU AGAIN!
[vigomarkimalgador@ecs-pa-coding1 csc138]$
```

Figure 8. TCP Client-side snapshot