A client/server application that communicate based on port number

2021
Introduction To Networks

DR "Hesham Arafaat

The basic idea of the Project:

Creating a server/Client application.
Which client uses to send a lowercase message to the server then receive back the message in uppercase from the server in the same Computer using python programming language version 3, using (socket Library) to communicate with two ports in the same computer.

Information about Socket Library:

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server.

Hardware & software:

Hardware Components:

-Computer

Software Components:

- python Version 3
- IDE for python programming (Note pad++)

Tools Required:

- Socket Library

Socket Library functions used in the Project:

- socket.socket() Create a new socket using the given address family, socket type and protocol number.

- Socket.Bind(address) Bind the socket to address. The socket must not already be bound. (The format of address depends on the address family — see above.) Raises an auditing event socket.bind with arguments self, address.

- socket.recvfrom(bufsize[, flags]) Receive data from the socket. The return value is a pair (bytes(data), address) where bytes is a bytes object representing the data received and address is the address of the socket sending the data. it defaults to zero. (The format of address depends on the address family.

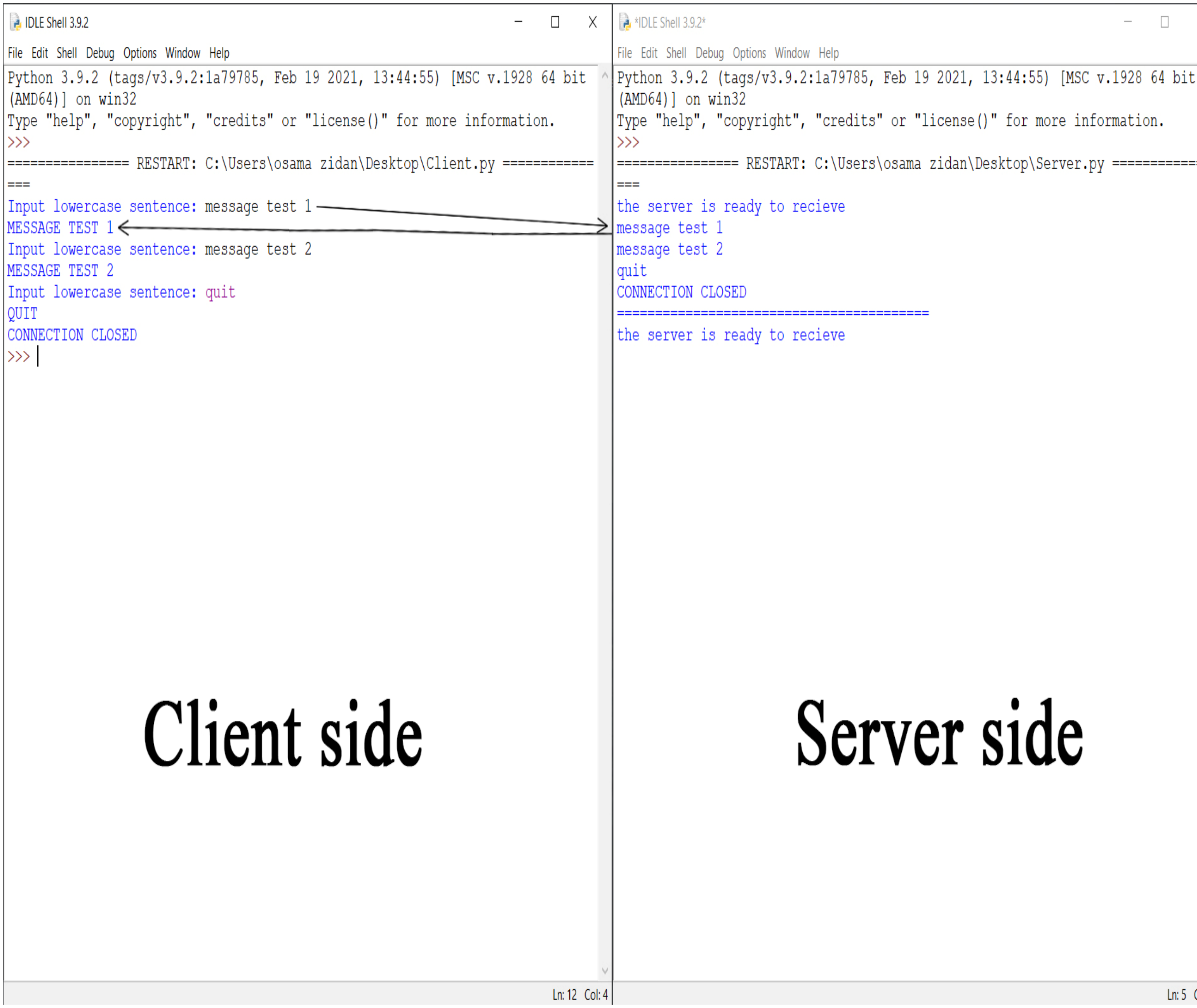
- socket.sendto(bytes(data) , address) Send data to the socket. The socket should not be connected to a remote socket, since the destination socket is specified by address, Return the number of bytes sent.

- socket.close() Mark the socket closed. The underlying system resources. Once that happens, all future operations on the socket object will fail. The remote end will receive no more data (after queued data is flushed).

The basic concept of the project:

- Using UDP protocol to send messages from one port to another without waiting any response from the Server.
- Using Some port numbers in the computer

THE APPLICATION :

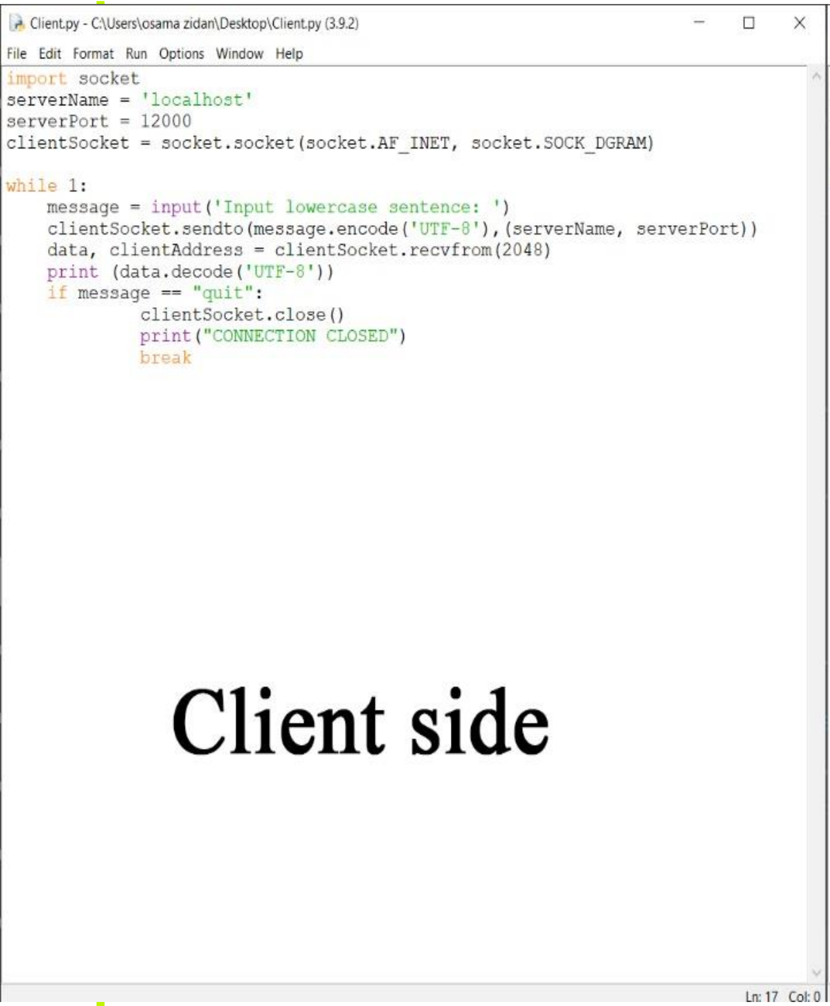

IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Ln: 12 Col: 4
Ln: 5
Client side
Server side

```
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: C:\Users\osama zidan\Desktop\Client.py ============
===
Input lowercase sentence: message test 1
MESSAGE TEST 1
Input lowercase sentence: message test 2
MESSAGE TEST 2
Input lowercase sentence: quit
QUIT
CONNECTION CLOSED
>>> |
```

```
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: C:\Users\osama zidan\Desktop\Server.py ===========
===
the server is ready to recieve
message test 1
message test 2
quit
CONNECTION CLOSED
=========================================
the server is ready to recieve
```

Steps:

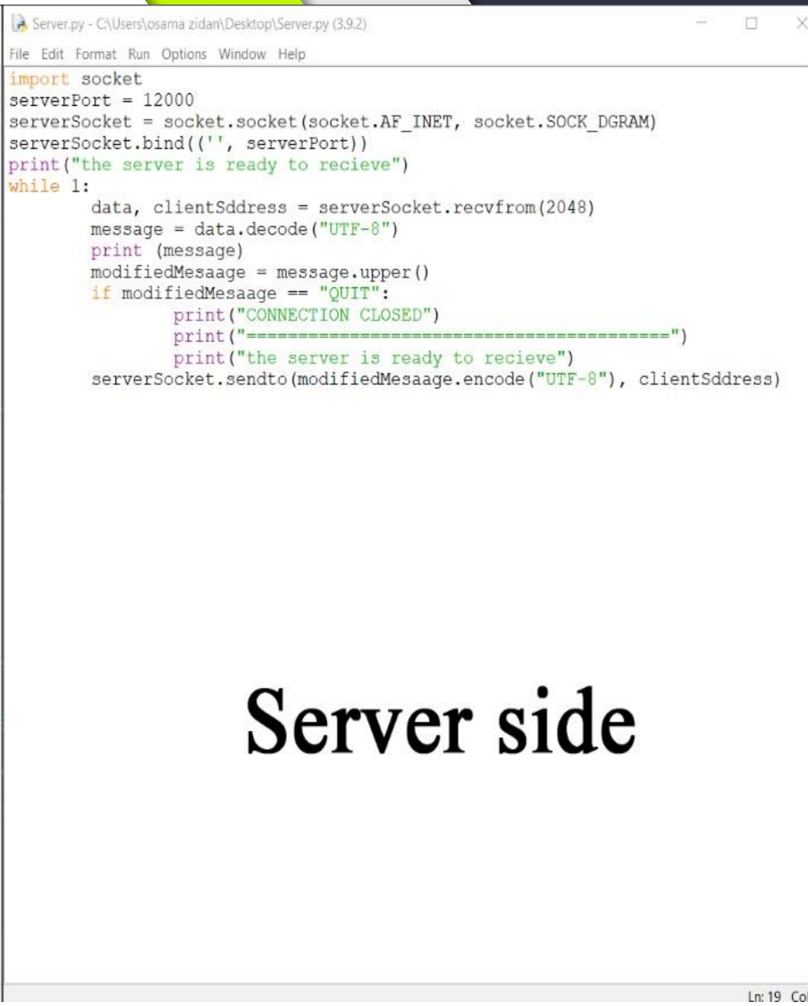
1- Client writes the lowercase message to client program to send it to the server.
2- The program encodes the message.
3- The program sends it to the server in port: 12000.
4- The server receives the message and the address (port number of client) from the client.
5- The server decodes the message.
6- The server makes the message characters uppercases.
7- The server sends the message back to the port number of the client but in Uppercase shape.
8- IF the client wrote “quit”, the client program automatically closes the connection with the server.
9- IF the server received “quit” from the client, the server closes the connection with the client too.


Client.py - C:\Users\osama zidan\Desktop\Client.py (3.9.2)
File Edit Format Run Options Window Help
Ln: 17 Col: 0

```python
import socket
serverName = 'localhost'
serverPort = 12000
clientSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

while 1:
    message = input('Input lowercase sentence: ')
    clientSocket.sendto(message.encode('UTF-8'),(serverName, serverPort))
    data, clientAddress = clientSocket.recvfrom(2048)
    print (data.decode('UTF-8'))
    if message == "quit":
            clientSocket.close()
            print("CONNECTION CLOSED")
            break
```

Client side


Server.py - C:\Users\osama zidan\Desktop\Server.py (3.9.2)
File Edit Format Run Options Window Help
Ln: 19 Col

```python
import socket
serverPort = 12000
serverSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
serverSocket.bind(('', serverPort))
print("the server is ready to recieve")
while 1:
        data, clientSddress = serverSocket.recvfrom(2048)
        message = data.decode("UTF-8")
        print (message)
        modifiedMesaage = message.upper()
        if modifiedMesaage == "QUIT":
                print("CONNECTION CLOSED")
                print("=========================================")
                print("the server is ready to recieve")
        serverSocket.sendto(modifiedMesaage.encode("UTF-8"), clientSddress)
```

Server side

Client CODE :

```
import socket
serverName = 'localhost'
serverPort = 12000
clientSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
while 1:
  message = input('Input lowercase sentence: ')
  clientSocket.sendto(message.encode('UTF-8'),(serverName, serverPort))
  data, clientAddress = clientSocket.recvfrom(2048)
   print (data.decode('UTF-8'))
   if message == "quit":
       clientSocket.close()
       print("CONNECTION CLOSED") break
```

SERVER CODE:

```
import socket
serverPort = 12000
serverSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
serverSocket.bind(('', serverPort))
print("the server is ready to recieve")
while 1:
    data, clientSddress = serverSocket.recvfrom(2048)
    message = data.decode("UTF-8")
    print (message)
    modifiedMesaage = message.upper()
    if modifiedMesaage == "QUIT":
        print("CONNECTION CLOSED")
        print("========================================")
        print("the server is ready to recieve")
    serverSocket.sendto(modifiedMesaage.encode("UTF-8"), clientSddress)
```

(THANKS)

BY :

1- Osama Mahmoud Sobhy Elzekred
2- Osama Mohamed Elsayed Zidan
3- Osama Salah Elalfy Mohamed
4- Ahmed Shaker Atia Elshura