YIXIN CAO
19536cy

PPROJECT

Description:
Running the program and demonstrate to me on Week 14th one by one. You have to use your own PC to show the result. We do NOT allow to share PC with others.

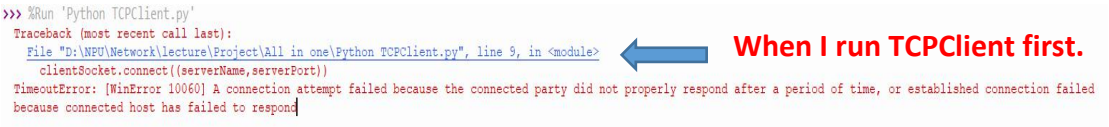
Application request:

- Client reads a line of characters (data) from its keyboard and sends the data to the server.
- The server receives the data and converts characters to uppercase.
- The server sends the modified data to the client.
- The client receives the modified data and displays the line on its screen.
- The 3.c result will be displayed on client side screen as well.

1. Install and compile the Python programs TCPClient and UDPClient on one host and TCPServer and UDPServer on another host.

a. Suppose you run TCPClient before you run TCPServer. What happens? Why?

When I run TCPClient first, the client will attempt to make a TCP connection with a non-existent server process. A TCP connection will not be made.



b. Suppose you run UDPClient before you run UDPServer. What happens? Why?

UDPClient doesn't establish a TCP connection with the server. Thus, everything should work fine if you first run UDPClient, then run UDPServer, and then type some input into the keyboard.

c. What happens if you use different port numbers for the client and server sides?

If you use different port numbers, then the client will attempt to establish a UDP connection with the wrong process or a non-existent process. Errors will occur.

Figure below is showed in server side:

```
>>> %Run 'Python UDPServer.py'
  Traceback (most recent call last):
    File "C:\Users\Arthur Ho\Downloads\Python UDPServer.py", line 8, in <module>
      serverSocket.bind(('', serverPort))
  OSError: [WinError 10048] Only one usage of each socket address (protocol/network
  address/port) is normally permitted
```

2. Suppose that in UDPClient.py, after we create the socket, we add the line: clientSocket.bind(('', 5432))

Will it become necessary to change UDPServer.py? What are the port numbers for the sockets in UDPClient and UDPServer? What were they before making this change?

The UDPServer.py is not changed.

Before modify the code:

```
> Frame 100: 51 bytes on wire (408 bits), 51 bytes captured (408 bits) on interface \Device\NPF_{45EDD1F7-1E
> Ethernet II, Src: IntelCor_a2:20:ab (80:86:f2:a2:20:ab), Dst: HonHaiPr_48:54:cf (d8:0f:99:48:54:cf)
> Internet Protocol Version 4, Src: 192.168.1.6, Dst: 192.168.1.5
v User Datagram Protocol, Src Port: 56542, Dst Port: 12000
    Source Port: 56542
    Destination Port: 12000
    Length: 17
    Checksum: 0x460c [unverified]
    [Checksum Status: Unverified]
    [Stream index: 5]
  > [Timestamps]
> Logical-Link Control
> Data (5 bytes)
```

After modify the code:

```
> Frame 74: 48 bytes on wire (384 bits), 48 bytes captured (384 bits) on interface \Device\NPF_{45EDD1F7-1E59-4B68-81B8
> Ethernet II, Src: IntelCor_a2:20:ab (80:86:f2:a2:20:ab), Dst: HonHaiPr_48:54:cf (d8:0f:99:48:54:cf)
> Internet Protocol Version 4, Src: 192.168.1.6, Dst: 192.168.1.5
v User Datagram Protocol, Src Port: 5432, Dst Port: 12000
    Source Port: 5432
    Destination Port: 12000
    Length: 14
    Checksum: 0xf721 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 7]
  > [Timestamps]
> Logical-Link Control
> Data (2 bytes)
```

3. Based on above codes. Add follow requirements:

a. Server has to received info as "Server received: "info from client xxxx"" on screen;

b. Client has to show "I have received modified info from server: "modified string from server. xxxxxx" "

UDPClient.py

```
#Python UDPClient
from socket import *
#ip address of my server host
serverName = '192.168.1.5'
serverPort = 12000
#create UDP socket for server
clientSocket = socket(AF_INET, SOCK_DGRAM)
#get user keyboary input
message = "Info from client: "+ input('Input lowercase sentence:')
#Attach server name, port to message; send into socket
clientSocket.sendto(message.encode(),(serverName, serverPort))
modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
print ("I have received modified info from server:\n"+ modifiedMessage.decode())
clientSocket.close()
```

UDPServer.py

```
#Python UDPServer
from socket import *
serverPort = 12000
#create UDP socket
serverSocket = socket(AF_INET, SOCK_DGRAM)
#bind socket to local port number 12000
serverSocket.bind(('', serverPort))
print ("The server is ready to receive")

while True:
    #Read from UDP socket into message, getting client's address (client IP and
port)
    message, clientAddress = serverSocket.recvfrom(2048)
    print("Server received: " + str(message) )
    modifiedMessage = "Modified string from server: "+ message.decode().upper()
    #send upper case string back to this client
    serverSocket.sendto(modifiedMessage.encode(),clientAddress)
```

TCPClient.py

```
from socket import *

serverName = '192.168.1.5'
serverPort = 12200
clientSocket = socket(AF_INET, SOCK_STREAM)
```

```
clientSocket.connect((serverName,serverPort))
sentence = "Info from client: " + input( "Input lowercase sentence:")
clientSocket.send(sentence.encode())
modifiedSentence = clientSocket.recv(1024)
print ("I have received modified info from server:\n" + modifiedSentence.decode())
clientSocket.close()
```


TCPServer.py

```
from socket import *
serverPort = 12200
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind(('',serverPort))
serverSocket.listen(1)
print ("The server is ready to receive")
while True:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    print("Server received: " + str(sentence) )
    capitalizedSentence = "Modified string from server: " + sentence.upper()
    connectionSocket.send(capitalizedSentence.encode())
connectionSocket.close()
```

---

c. Client side display information as “My BMI is xx” / “Today’s temperature is xx” / “My Lucky number is xx”.
i. The formula for BMI is weight in kilograms divided by height in meters squared.
ii. Fahrenheit to Celsius Conversion Formula. To convert temperatures in degrees Fahrenheit to Celsius, subtract 32 and multiply by .5556 (or 5/9).
iii. You can choose a lucky number as a 6x digital random number

---

TCPClient.py

```
from socket import *

serverName = '192.168.1.5'
serverPort = 12200

"""
Lower to upper
```

```
"""
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
#conver to upper
sentence = "Info from client: " + input( "Input lowercase sentence:")
clientSocket.send(sentence.encode())
#receive modified sentence
modifiedSentence = clientSocket.recv(1024)
print ("I have received modified info from server:\n" + modifiedSentence.decode())
clientSocket.close()

"""
BMI
"""
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
#weight and height of BMI
weight=input("Enter your weight(kg): ")
height=input("Enter your height(m): ")
CalBMI= str(round(float(weight) / (float(height)*float(height)),2))
clientSocket.send(CalBMI.encode())
#receive BMI
BMI = clientSocket.recv(1024)
print("Your BMI is " + str(BMI))
clientSocket.close()


"""
Temperature F conver to C
"""
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
#temperature in F
Fahrenheit = input("Enter temperature in Fahrenheit: ")
clientSocket.send(Fahrenheit.encode())
#receive temperature in Celsius
Celsius = clientSocket.recv(1024)
print(str(Fahrenheit)+" Fahrenheit in Celsius is: "+ str(Celsius))
clientSocket.close()

"""
Random Lucky Number
"""
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
#temperature in F
lucky = '0'
```

```
clientSocket.send(lucky.encode())
#receive temperature in Celsius
LuckyNumber = clientSocket.recv(1024)
print("Your lucky number is: "+ str(LuckyNumber))
clientSocket.close()
```

Sample Output:

```
Input lowercase sentence:have a nice day!
I have received modified info from server:
Modified string from server: INFO FROM CLIENT: HAVE A NICE DAY!
Enter your weight(kg): 90
Enter your height(m): 1.76
Your BMI is b'Unhealthy'
Enter temperature in Fahrenheit: 75
75 Fahrenheit in Celsius is: b'23.89(Converted in Server)'
Your lucky number is: b'578283(Get in Server)'
```

TCPServer.py

```
from socket import *
import random
serverPort = 12200
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind(('',serverPort))
serverSocket.listen(1)
print ("The server is ready to receive")

"""
Lower to Upper
"""
while True:
    connectionSocket, addr = serverSocket.accept()
    #get sentence in lower case
    sentence = connectionSocket.recv(1024).decode()
    print("Server received: " + str(sentence))
    #conver sentence to upper case and send back to client
    capitalizedSentence = "Modified string from server: " + sentence.upper()
    if(connectionSocket.send(capitalizedSentence.encode())):
        break
connectionSocket.close()


"""
BMI
"""
while True:
    connectionSocket, addr = serverSocket.accept()
```

```
        #get weight and height
        CALBMI = connectionSocket.recv(1024).decode()
        print("BMI received from client is: "+str(CALBMI))
        #calculate BMI and send back to client
        if float(CALBMI)>25:
                BMImessage='Unhealthy'
        else:
                BMImessage='Normal'
        if(connectionSocket.send(BMImessage.encode())):
                break
connectionSocket.close()

"""
Temperature convert
"""
while True:
        connectionSocket, addr = serverSocket.accept()
        #get weight and height
        tempF = connectionSocket.recv(1024).decode()
        print("Temperature in    Fahrenheit is: "+tempF)
        #convert Fahrenheit to celsius and send back to client
        tempC= str(round(float((float(tempF)-32)*(5/9)),2))+"(Converted in Server)"
        if(connectionSocket.send(tempC.encode())):
                break
connectionSocket.close()

"""
Random Lucky Number
"""
while True:
        connectionSocket, addr = serverSocket.accept()
        LuckyN = connectionSocket.recv(1024).decode()
        #get random 6 digits lucky number
        LuckyNum= str(random.randrange(100000, 1000000, 1))+"(Get in Server)"
        if(connectionSocket.send(LuckyNum.encode())):
                break
connectionSocket.close()
```

Sample Output:

```
Server received: Info from client: have a nice day!
BMI received from client is: 29.05
Temperature in Fahrenheit is: 75
```

UDPClient.py

```
#Python UDPClient
from socket import *
#ip address of my server host
serverName = '192.168.1.5'
serverPort = 12000
#create UDP socket for server
clientSocket = socket(AF_INET, SOCK_DGRAM)
#get user keyboary input
message = "Info from client: "+ input('Input lowercase sentence:')
#Attach server name, port to message; send into socket
clientSocket.sendto(message.encode(),(serverName, serverPort))
#weight and height of BMI
weight=input("Enter your weight(kg): ")
clientSocket.sendto(weight.encode(),(serverName, serverPort))
height=input("Enter your height(m): ")
clientSocket.sendto(height.encode(),(serverName, serverPort))
#temperature in F
Fahrenheit = input("Enter temperature in Fahrenheit: ")
clientSocket.sendto(Fahrenheit.encode(),(serverName, serverPort))
#get lucky number
lucky='0'
clientSocket.sendto(lucky.encode(),(serverName, serverPort))


modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
#receive BMI
BMI, serverAddress = clientSocket.recvfrom(2048)
#receive temperature in Celsius
Celsius, serverAddress = clientSocket.recvfrom(2048)
#Receive lucky number
luckyNumber, serverAddress = clientSocket.recvfrom(2048)
print ("I have received modified info from server:\n" + modifiedMessage.decode())
print("Your BMI is " + BMI.decode())
print(Fahrenheit +" Fahrenheit in Celsius is: "+ Celsius.decode())
print("Your lucky number is: "+luckyNumber.decode())


clientSocket.close()
```

Sample Output:

```
Input lowercase sentence:what a day!
Enter your weight(kg): 70
Enter your height(m): 1.76
Enter temperature in Fahrenheit: 100
```

I have received modified info from server:
Modified string from server: INFO FROM CLIENT: WHAT A DAY!
Your BMI is 22.6(Calculated in Server)
100 Fahrenheit in Celsius is: 37.78(Converted in Server)
Your lucky number is: 425668(Get in Server)

UDPServer.py

```
#Python UDPServer
from socket import *
import random
serverPort = 12000
#create UDP socket
serverSocket = socket(AF_INET, SOCK_DGRAM)
#bind socket to local port number 12000
serverSocket.bind(('', serverPort))
print ("The server is ready to receive")

while True:
	#Read from UDP socket into message, getting client's address (client IP and
port)
	message, clientAddress = serverSocket.recvfrom(2048)
	print("Server received: " + str(message) )
	#get weight and height
	weightinkg, clientAddress = serverSocket.recvfrom(2048)
	heightinm, clientAddress = serverSocket.recvfrom(2048)
	print("Weight and height is: " + str(weightinkg)+" and "+str(heightinm))
	#get weight and height
	tempF, clientAddress = serverSocket.recvfrom(2048)
	print("Temperature in Fahrenheit is: "+str(tempF))
	#get lucky number
	Lucky, clientAddress = serverSocket.recvfrom(2048)
	print("Lucky number is: " + str(Lucky))


	modifiedMessage = "Modified string from server: "+ message.decode().upper()
	#send upper case string back to this client
	serverSocket.sendto(modifiedMessage.encode(),clientAddress)
	#calculate BMI and send back to client
	CalBMI= str(round(float(weightinkg) /
(float(heightinm)*float(heightinm)),2))+"(Calculated in Server)"
	serverSocket.sendto(CalBMI.encode(),clientAddress)
	#convert Fahrenheit to celsius and send back to client
	tempC= str(round(((float(tempF)-32)*(5/9)),2))+"(Converted in Server)"
	serverSocket.sendto(tempC.encode(),clientAddress)
	#get random 6 digits lucky number
	LuckyN= str(random.randrange(100000, 1000000, 1))+"(Get in Server)"
```

```
    serverSocket.sendto(LuckyN.encode(),clientAddress)
```

Sample Output:

```
The server is ready to receive
Server received: b'Info from client: what a day!'
Weight and height is: b'70' and b'1.76'
Temperature in Fahrenheit is: b'100'
Lucky number is: b'0'
```

4. Write a short summary (around 150 words) to describe how you apply computer science principles and skills to work on the project. Present program and Update the essay into portal.

Before I started to write programs, I first analysis what is the output wanted from the project, what to be distinguished between TCP and UDP and how to run the whole project.

Firstly, the project is to practice the process between client and server. The outputs of this project is to receive input from client keyboard and sent it to server side. Within server side, the server program convert the information received from client to whatever wanted (in this case: convert lower case to upper case, calculate BMI, convert temperature from F to C and get random lucky number) and send it back to client side.

Secondly, different between TCP and UDP is that TCP request connection first but UDP doesn't. So in this case, when trying to do multiple tasks, need to be careful with re-connection in TCP.

And finally, to run the whole project, even though UDP does not require to run server first, but I think it will be better to run server first.

While writing and testing the program, it also gave me a clearer idea of how to write the code correctly and the logic between client and server.