Proiect Retele

Kohan Alexandru

UDP Client

```python
import socket

#the ip "127.0.0.1" is our ip
ip="127.0.0.1"
#we match the port with the server
port=7777
#we create a socket because a communication realised between at least two sockets
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)

while True:
    msg=input("Message:")
    #send data to the server
    #we specify the ip and port of our server
    s.sendto(msg.encode(),(ip,port))
    #we recieve data from server
    ans,addr=s.recvfrom(60)
    print(str(addr)+" : "+ans.decode())
```

UDP Server

```python
#import library socket
import socket
#create a socket
#AF_INET - type of adress(IPv4 in this case) but it exists IPv6 too
#SOCK_DGRAM is for UDP protocol
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
#bind() assigns a socket to an address
#we use "0.0.0.0" to bind our socket to all addresses
#In this case it is not bound to a specific IP address
# but will be able to receive data send to any IP address of the machine
#7777 is the port
#When data arrives at a device, the network software
# looks at the port number and sends it to the right program
s.bind(("0.0.0.0",7777))

print("Server open")
while True:
    #in the data we have the message
    #in addr we have the address of the one who sends data
    #recvfrom is the function to recieve data
    #60 is the number of bytes to get
    data,addr=s.recvfrom(60)
    msg=data.decode()
    print(str(addr)+" : "+str(msg))
    ans=input("Answer: ")
    #sendto is used to send data
    s.sendto(ans.encode(),addr)
```

Run

```
Server open
('127.0.0.1', 57048) : salut
Answer: salut
```

```
Message:salut
('127.0.0.1', 7777) : salut
Message:
```

TCP Client

```python
import socket

while True:
    a=input("Press Enter to send message")
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(("127.0.0.1", 7777))
    s.send("Salut".encode())
    print(s.recv(120).decode())
    s.close()
```

TCP Server

```python
import socket
#SOCK_STREAM is for TCP protocol
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.bind(("0.0.0.0",7777))
#we need to use listen for the server to listen for a connection of a client
s.listen(5)
while True:
    #Accept a connection
    #cs is a socket
    #addr is an address
    cs, addr = s.accept()
    print("Accepted a connection request")
    #we recieve data from a client
    #120 bytes of data
    b=cs.recv(120)
    print(str(addr)+" : "+b.decode())
    # send data to a client
    cs.send("Hello".encode())
    #close the socket
    cs.close()
```

Problema 1

Se transmite o litera de la client la server, serveru trimite inapoi litera dublata

Client

```python
import socket

ip="127.0.0.1"
port=7777
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)

while True:
    msg=input("Message:")
    s.sendto(msg.encode(),(ip,port))
    ans,addr=s.recvfrom(60)
    print(str(addr)+" : "+ans.decode())
```

Server

```python
import socket
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
s.bind(("0.0.0.0",7777))

print("Server open")
while True:
    data,addr=s.recvfrom(60)
    msg=data.decode()
    print(str(addr)+" : "+str(msg))
    ans=msg+msg
    s.sendto(ans.encode(),addr)
```

Problema 2

Intoare cuvintele concatenate

Client

```
import socket

while True:
    msg1=input("Message:")
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(("127.0.0.1", 7777))
    s.send(msg1.encode())

    msg2 = input("Message:")
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(("127.0.0.1", 7777))
    s.send(msg2.encode())

    print(s.recv(120).decode())
    s.close()
```

Server

```python
import socket
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.bind(("0.0.0.0",7777))
s.listen(5)
while True:
    cs, addr = s.accept()
    print("Accepted a connection request")
    b=cs.recv(120)
    msg=b.decode();

    cs, addr = s.accept()
    b = cs.recv(120)
    msg=msg+b.decode();

    cs.send(msg.encode())
    cs.close()
```

Problema 3

Clientul trimite 2 numere, serverul intoarce suma

Client

```python
import socket

ip="127.0.0.1"
port=7777
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)

while True:
    msg1=input("Message:")
    s.sendto(msg1.encode(),(ip,port))

    msg2=input("Message:")
    s.sendto(msg2.encode(), (ip, port))

    ans,addr=s.recvfrom(60)
    print(str(addr)+" : "+ans.decode())
```

Server

```python
import socket
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
s.bind(("0.0.0.0",7777))

print("Server open")
while True:
    data,addr=s.recvfrom(60)
    msg1=int(data.decode())
    print(str(addr)+" connected")

    data, addr = s.recvfrom(60)
    msg2 = int(data.decode())
    print(str(addr) + " connected")

    ans=msg1+msg2
    s.sendto(str(ans).encode(),addr)
```

Problema 4

Clientul trimite serverului un sir de caractere (de exemplu numele utilizatorului citit de la tastatura). Serverul afiseaza pe ecran sirul primit si portul clientului si ii raspunde acestuia cu suma cifrelor din Portul clientului. Clientul va afisa pe ecran numarul primit.

Client

```
import socket

while True:
    msg=input("Write something: ")
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(("127.0.0.1", 7777))
    s.send(msg.encode())
    print(s.recv(120).decode())
    s.close()
```

Server

```
import socket
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.bind(("0.0.0.0",7777))
s.listen(5)
def sum(x):
    s=0
    while x!=0:
        s=s+x%10
        x=int(x/10)
    return s
while True:
    cs, addr = s.accept()
    ip,port = addr
    print("Accepted a connection request")
    b=cs.recv(120)
    print(str(addr)+" : "+b.decode())
    cs.send(str(sum(port)).encode())
    cs.close()
```

TCP Fork

Client

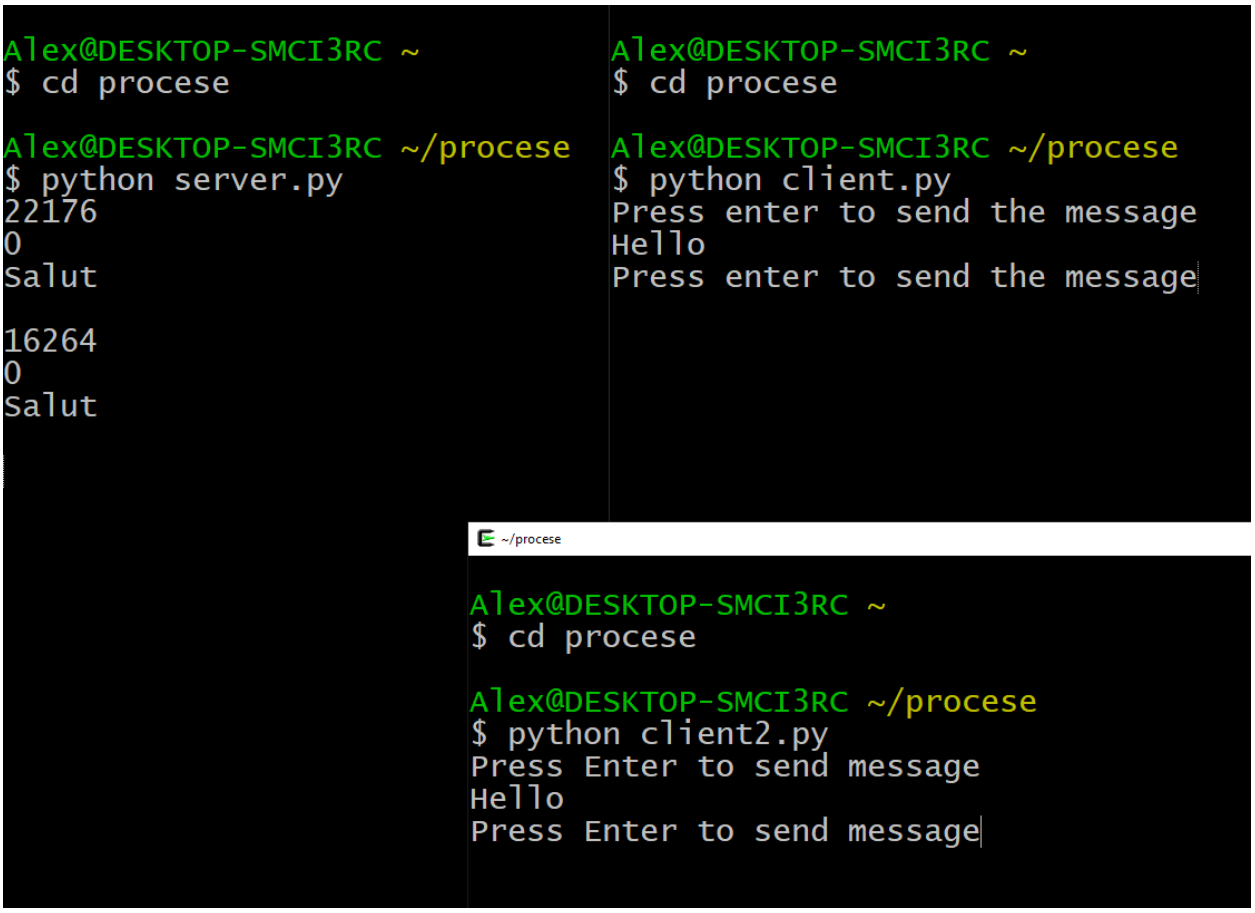
```
import socket

while True:
    a=raw_input("Press Enter to send message")
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(("127.0.0.1", 7777))
    s.send("Salut")
    print(s.recv(120))
    s.close()
```

Server

Fork is used to receive data from multiple clients. I run 2 clients in the example.

```
import socket
import os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.bind(("0.0.0.0",7777))
s.listen(5)
while True:
    cs, addr = s.accept()
    print(os.fork())
    if os.fork()==0:
        b=cs.recv(120)
        print b
        cs.send("Hello")
        cs.close()
        os._exit(0)
```

Run


Alex@DESKTOP-SMCI3RC ~
$ cd procese
Alex@DESKTOP-SMCI3RC ~/procese
$ python server.py
22176
0
Salut
16264
0
Salut
Alex@DESKTOP-SMCI3RC ~
$ cd procese
Alex@DESKTOP-SMCI3RC ~/procese
$ python client.py
Press enter to send the message
Hello
Press enter to send the message
~/procese
Alex@DESKTOP-SMCI3RC ~
$ cd procese
Alex@DESKTOP-SMCI3RC ~/procese
$ python client2.py
Press Enter to send message
Hello
Press Enter to send message