

# REȚELE DE CALCULATOARE

## PROIECT SERVER-CLIENT

***Student : ABDURAMAN CILLER MELIS***

**CSIE, an III, grupa 1066**

***Prof. Coordonator: TIMOFTE CARMEN***

## **1. CE FACE PROIECTUL?**

Proiectul constă într-o aplicație simplă de tipul client-server pe TCP. Serverul va deschide un socket și va aștepta conectarea clienților care, odată conectați, vor primi un mesaj prin care li se solicită adăugarea mai multor numere, despărțite prin “-”.

Serverul, la rândul lui, primește adresa IP a clientului, șirul de numere adăugat și returnează cel mai mic, dar și cel mai mare număr din șirul respectiv.

Operația se poate repeta până în momentul în care este apăsată tasta „x” de către client.

Codul este scris în Python, versiunea 3.5

## **2. PROGRAM SERVER**

### **2.1. COD COMENTAT**

```
#am importat librăria SOCKET(o interfață de nivel scăzut pentru comunicarea în rețea)
import socket
#am definit 2 funcții care primesc ca parametru o variabilă de tip string introdusă de către client
#aceste funcții returnează valoarea minimă și valoarea maximă din șir
#numerele vor fi separate prin “-”
def minim(dataTastatura):
    return min(int(i) for i in dataTastatura.split('-'))
def maxim(dataTastatura):
    return max(int(i) for i in dataTastatura.split('-'))
#am definit 2 variabile care rețin host-ul(gazda) și portul
host = "sys.ase.ro"
port = 6601
#mi-am creat o instanță a socket-ului, numită mySocket
mySocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
#folosesc funcția bind pentru a asocia socket-ul cu adresa serverului și cu portul
mySocket.bind((host,port))
#folosesc listen pentru a pune socket-ul în starea de ascultare
#parametrul 1 înseamnă că socket-ul ascultă până la închiderea conexiunii
mySocket.listen(1)
```

***#mi-am definit alte două variabile care vor reține conexiunea de la client, dar și adresa acestuia***

***#adresa va fi afișată***

```
conn, addr = mySocket.accept()
```

```
print("S-a conectat: "+str(addr))
```

***#atâta timp cât sunt introduse date de către client, acestea vor fi procesate***

***#in caz contrar, se va ieși din while prin break***

```
while True:
```

***#în variabila data este stocat șirul de numere introdus de către client***

***#conținutul acestei variabile trebuie decodificat***

```
    data = conn.recv(1024).decode()
```

```
    if not data:
```

```
        break
```

***#serverul afișează datele pe care le-a primit de la client***

***#în variabilele data1 și data2 se rețin rezultatul celor 2 funcții (minimul și maximul din șir)***

```
    print ("Serverul a primit "+str(data))
```

```
    data1 = minim(data)
```

```
    data2 = maxim(data)
```

***#prin send, trimit clientului datele procesate, prelucrate, dar codificate***

```
    conn.send((str(data1)+"-"+str(data2)).encode("ascii"))
```

***#prin close, închid conexiunea***

```
conn.close()
```

## 2.2. PRINT SCREEN CU RULAREA SERVER-ULUI

– prin comanda `python3.5 nume_fisier_server(extensia py)`

```
sys.ase.ro - PuTTY
login as: gr4_1
gr4_1@sys.ase.ro's password:
Last login: Tue Dec 19 21:58:16 2017 from 92.55.154.249
-bash-4.2$ ls
fis4  logs  proiect  socket  socket2  web  www
-bash-4.2$ cd proiect
-bash-4.2$ python3.5 server2.py
S-a conectat: ('37.120.249.45', 37754)
Serverul a primit 45-34-10
```

## 2.3. PORTUL CU IDENTIFICAREA PROTOCOLULUI

- prin comanda `netstat -a -n -p | grep 6601`

```
sys.ase.ro - PuTTY
login as: gr4_1
gr4_1@sys.ase.ro's password:
Last login: Tue Dec 19 22:02:32 2017 from 92.55.154.249
-bash-4.2$ netstat -a -n -p | grep 6601
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp        0      0 37.120.249.45:6601      0.0.0.0:*               LISTEN      2
7393/python3.5
-bash-4.2$
```

## 2.4. PROCESUL

- prin comanda `ps -fu gr4_1 | grep server2.py`

```
-bash-4.2$ ps -fu gr4_1 | grep server2.py
gr4_1      27393 27294  0 22:04 pts/72    00:00:00 python3.5 server2.py
gr4_1      27602 27401  0 22:07 pts/74    00:00:00 grep --color=auto server2.py
-bash-4.2$
```

### **3. PROGRAM CLIENT**

#### **3.1. COD COMENTAT**

*#am importat librăria SOCKET*

```
import socket
```

*#mi-am declarat variabilele host și port*

```
host = "sys.ase.ro"
```

```
port = 6601
```

*#mi-am creat o instanță de socket numită mySocket*

```
mySocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

*#spre deosebire de server care avea nevoie de bind și listen*

*#clientul folosește connect pentru a se conecta la host-ul și port-ul serverului*

```
mySocket.connect((host, port))
```

*#mesajul pe care îl primește clientul*

```
print("Introduceți datele separate prin '-'. Dacă vreți să renunțați, apăsați tasta x:  
")
```

```
message = input("****")
```

*#până în momentul în care clientul nu va apăsa tasta x, datele introduse de acesta vor fi transmise server-ului*

```
while message != 'x':
```

*#se trimite datele către socket, dar în variantă criptată*

```
    mySocket.sendall(message.encode("ascii"))
```

*#se primesc date de la socket(cu o lungime maximă de 1024 bytes), care vor fi decodate*

```
    data = mySocket.recv(1024).decode("ascii")
```

```
    if(message!='x'):
```

```
        print('Cel mai mic număr: ',data.split('-')[0],"\nCel mai mare număr:  
",data.split('-')[1])
```

*#clientul poate adăuga noi șiruri de numere până când va apăsa tasta x*

```
    message = input("****")
```

```
mySocket.close()
```

```
mc [gr4_1@sys.ase.ro]:~/proiect
/home/an3/gr4_1/proiect/client2.py 514/514 10
import socket
host = "sys.ase.ro"
port = 6601
mySocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
mySocket.connect((host, port))
print("Introduceti datele separate prin '-'. Daca vreti sa renuntati, apasati tasta x: ")
message = input("****")
while message != 'x':
    mySocket.sendall(message.encode("ascii"))
    data = mySocket.recv(1024).decode("ascii")
    if(message!='x'):
        print('Cel mai mic numar: ',data.split('-')[0],"\nCel mai mare numar: ",data.split('-')[1])
    message = input("****")
mySocket.close()
```

1Help 2UnWrap 3Quit 4Hex 5Goto 6 7Search 8Raw 9Format 10Quit

### 3.2. PRINT SCREEN CU RULAREA CLIENTULUI

- prin comanda **python3.5 nume\_fisier\_client** (cu extensia py)

```
sys.ase.ro - PuTTY
login as: gr4_1
gr4_1@sys.ase.ro's password:
Last login: Tue Dec 19 22:53:05 2017 from 92.55.154.249
-bash-4.2$ ls
fis4  logs  proiect  socket  socket2  web  www
-bash-4.2$ cd proiect
-bash-4.2$ python3.5 client2.py
Introduceti datele separate prin '-'. Daca vreti sa renuntati, apasati tasta x:
***45-100-780
Cel mai mic numar: 45
Cel mai mare numar: 780
***10-50-70350
Cel mai mic numar: 10
Cel mai mare numar: 70350
***0-4-9-10-67-30
Cel mai mic numar: 0
Cel mai mare numar: 67
***
```

```
mc [gr4_1@sys.ase.ro]:~/proiect
conn.send((str(data1)+"-"+str(data2)).encode("ascii"))
conn.close()

-bash-4.2$ python3.5 server22.py
Traceback (most recent call last):
  File "server22.py", line 12, in <module>
    mySocket.bind((host,port))
OSError: [Errno 98] Address already in use
-bash-4.2$ mc -ac

-bash-4.2$ python3.5 server22.py
Traceback (most recent call last):
  File "server22.py", line 13, in <module>
    mySocket.bind((host,port))
OSError: [Errno 98] Address already in use
-bash-4.2$ mc -ac

-bash-4.2$ python3.5 server22.py
S-a conectat: ('37.120.249.45', 48529)
Serverul a primit 45-100-780
Serverul a primit 10-50-70350
Serverul a primit 0-4-9-10-67-30

sys.ase.ro - PuTTY
login as: gr4_1
gr4_1@sys.ase.ro's password:
Last login: Tue Dec 19 22:53:05 2017 from 92.55.154.249
-bash-4.2$ ls
fis4  logs  proiect  socket  socket2  web  www
-bash-4.2$ cd proiect
-bash-4.2$ python3.5 client2.py
Introduceti datele separate prin '-'. Daca vreti sa renuntati, apasati tasta x:
***45-100-780
Cel mai mic numar: 45
Cel mai mare numar: 780
***10-50-70350
Cel mai mic numar: 10
Cel mai mare numar: 70350
***0-4-9-10-67-30
Cel mai mic numar: 0
Cel mai mare numar: 67
***
```