NAME: SIVARANGINI.Y

ROLL NO:231901051

DATE:18/9/2024

EXP 10: PING TO TEST SERVER CONNECTIVITY USING SOCKETS

Aim:

To develop ping program to test server connectivity using sockets.

Algorithm:

Server.py

1. Import the socket package

2. Initialize local IP address and local port.

3. Create a socket using socket() function

4. Bind the IP address and port number.

5. Accept client request for connection.

6. Print the received connection details

7. Send reply message to the client.
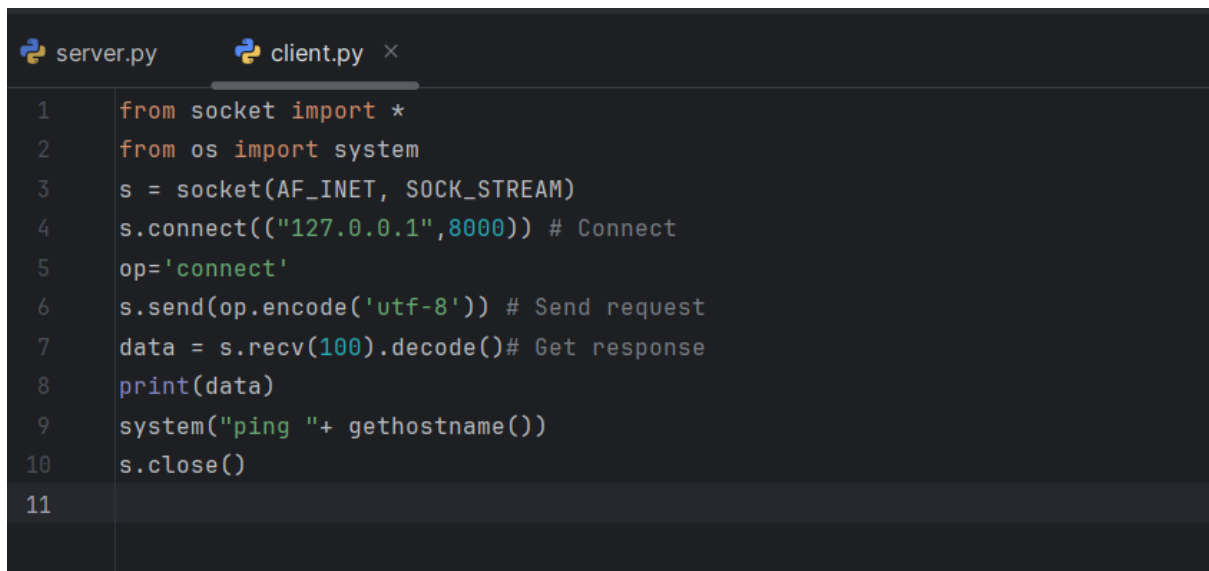
8. Close the connection.

Client.py

1. Import the socket package

2. Initialize server IP address and local port.

3. Create a socket using socket() function.

4. Start the timer.

5. Send message to the server.

6. The reply message of the server is received.

7. The timer is stopped.

8. Print the round trip time statistics.

Ping to test server connectivity using sockets

Client code:

from socket import *

from os import system

s = socket(AF_INET, SOCK_STREAM)

s.connect(("127.0.0.1",8000)) # Connect

op='connect'

s.send(op.encode('utf-8')) # Send request

data = s.recv(100).decode()# Get response

print(data)

system("ping "+ gethostname())

s.close()

```
server.py        client.py  ×
1      from socket import *
2      from os import system
3      s = socket(AF_INET, SOCK_STREAM)
4      s.connect(("127.0.0.1",8000)) # Connect
5      op='connect'
6      s.send(op.encode('utf-8')) # Send request
7      data = s.recv(100).decode()# Get response
8      print(data)
9      system("ping "+ gethostname())
10     s.close()
11
```

Server Code:

from socket import *

```
from os import system
s = socket(AF_INET,SOCK_STREAM)
s.bind(("",8000))
s.listen(5)
while True:
        c,a = s.accept()
        print("Received connection from", a)
        data=c.recv(100).decode()
        print(data)
        c.send(data.encode('utf-8'))
        system("ping "+ a[0])
 c.close()
```

```
server.py ×    client.py
1    from socket import *
2    from os import system
3
4    s = socket(AF_INET, SOCK_STREAM)
5    s.bind(("", 8000))
6    s.listen(5)
7    while True:
8        c, a = s.accept()
9        print("Received connection from", a)
10       data = c.recv(100).decode()
11       print(data)
12    💡  c.send(data.encode('utf-8'))
13       system("ping " + a[0])
14    c.close()
15
16
17
```

OUTPUT:

SERVER CODE:

```
C:\Users\Sivarangini\PycharmProjects\python\.venv\Scripts\python.exe C:\Users\Sivarangini\PycharmProjects\python\server.py
Received connection from ('127.0.0.1', 56854)
connect

Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

CLIENT CODE:

```
C:\Users\Sivarangini\PycharmProjects\python\.venv\Scripts\python.exe C:\Users\Sivarangini\PycharmProjects\python\client.py
connect

Pinging LAPTOP-KV7M51GT [fe80::6f9d:a40a:ca05:44cd%7] with 32 bytes of data:
Reply from fe80::6f9d:a40a:ca05:44cd%7: time<1ms
Reply from fe80::6f9d:a40a:ca05:44cd%7: time<1ms
Reply from fe80::6f9d:a40a:ca05:44cd%7: time<1ms
Reply from fe80::6f9d:a40a:ca05:44cd%7: time<1ms

Ping statistics for fe80::6f9d:a40a:ca05:44cd%7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

Process finished with exit code 0
```

RESULT:

ping program to test server connectivity using sockets is verified.