

Ex.No:10B

Date:01/10/2024

Roll No:231901004

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

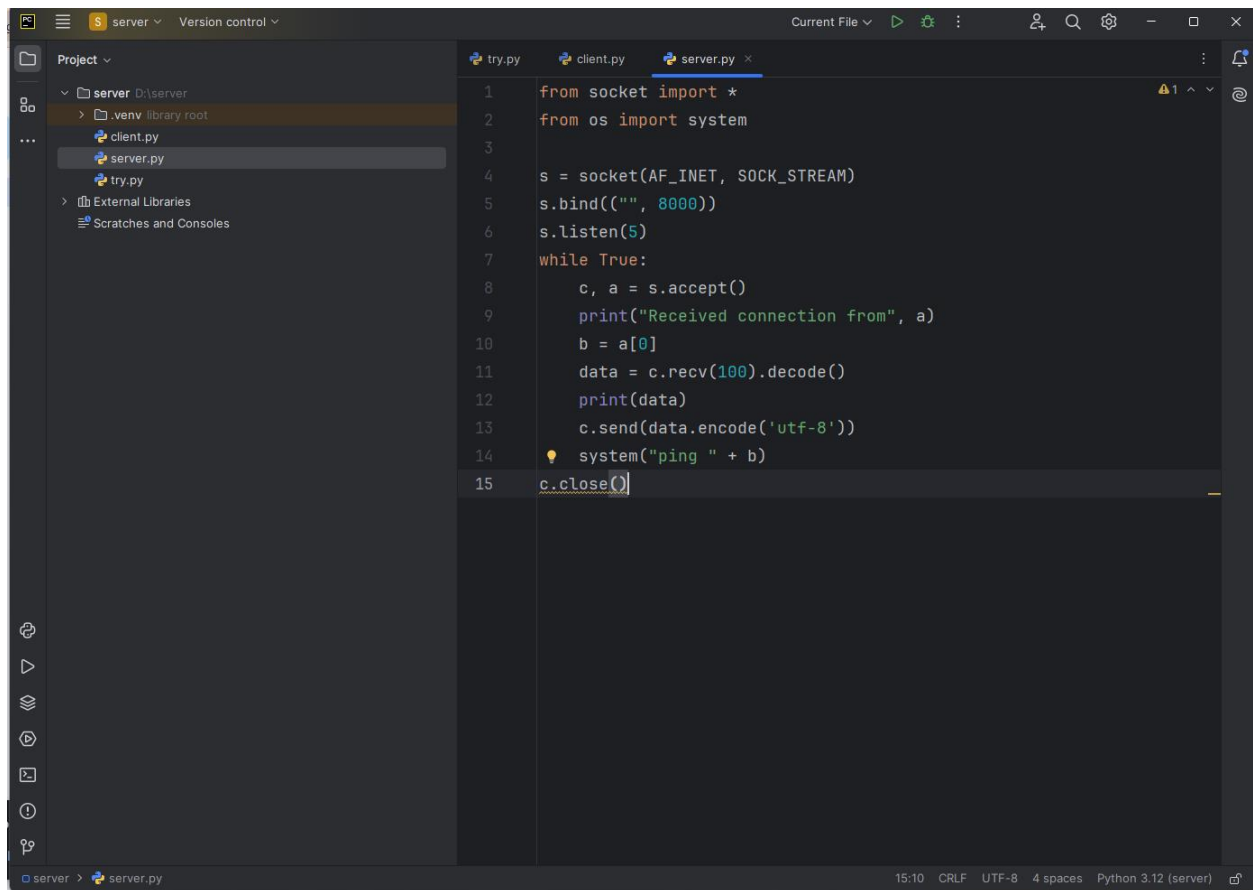
Client code:

```
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 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)
c.close()
```



The screenshot shows a code editor with a dark theme. On the left is a project explorer showing a folder named 'server' with sub-files 'client.py', 'server.py', and 'try.py'. The main editor area displays the code for 'server.py'. The code is a Python script that uses the 'socket' and 'os' modules to create a simple TCP server. It binds to port 8000, listens for connections, and when a connection is accepted, it prints the connection details, receives data, prints it, and sends a 'ping' response. The script ends with 'c.close()'.

```
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    b = a[0]
11    data = c.recv(100).decode()
12    print(data)
13    c.send(data.encode('utf-8'))
14    system("ping " + b)
15 c.close()
```

The status bar at the bottom indicates the file is 'server.py', the time is 15:10, and the encoding is UTF-8.

Result:

Thus, the server connectivity is tested using Sockets experiment was done.