LAB EXPERIMENT-8 (2023008358)

SOCKET PROGRAMMING

Aim:

To develop a client-server communication model using stream sockets in Python. The server will listen on port **5003**, accept a client connection, exchange a simple text message ("Hello"), and close the connection. This demonstrates fundamental socket programming concepts such as binding, listening, accepting connections, sending, and receiving messages.

Input code for server

```
import socket
# Create the client socket client_socket =
socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# Connect to the server at localhost and port 5003
client_socket.connect(('localhost', 5003)) # Send
the message "Hello" to the server
client_socket.sendall(b"Hello") # Receive the
response from the server response =
client_socket.recv(1024) print(f"Received from
server: {response.decode()}")
# Close the connection client_socket.close()
```

Input code for client

```
import socket
# Create the server socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# Bind the socket to localhost and port 5003 server_socket.bind(('localhost',
5003))
# Start listening for incoming connections (backlog of 1)
server_socket.listen(1)
```

print("Server is listening on port 5003...") #

Accept a connection from the client conn,

addr = server_socket.accept()

print(f"Connection established with {addr}")

# Receive the message from the client data

= conn.recv(1024)

print(f"Received from client: {data.decode()}")

# Send a response to the client

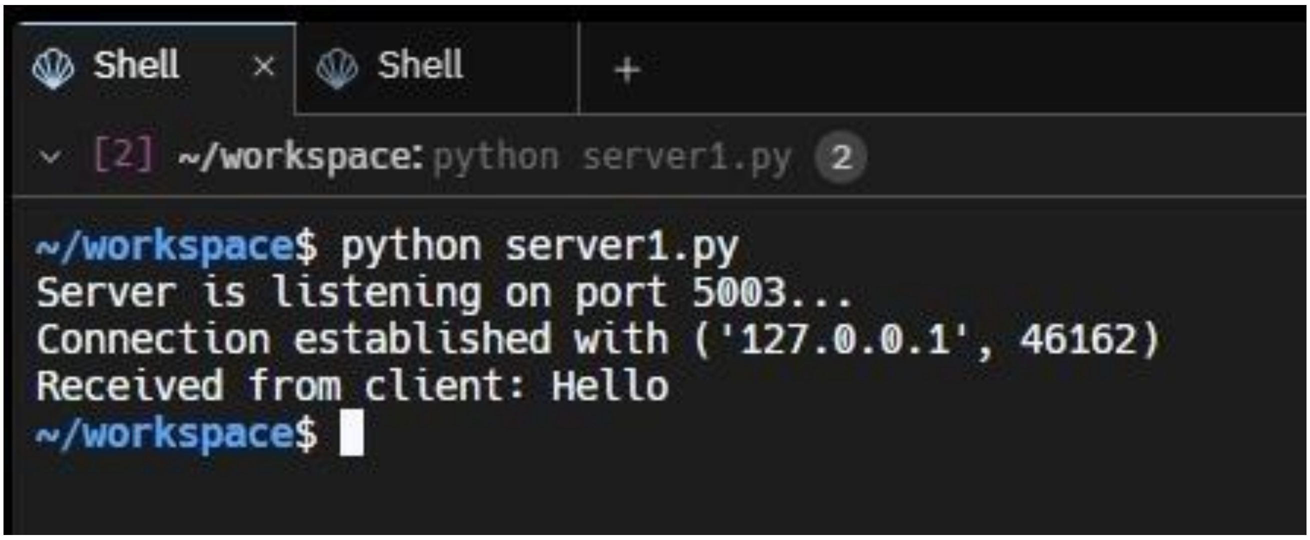
conn.sendall(b"Hello") # Close

the connection conn.close()

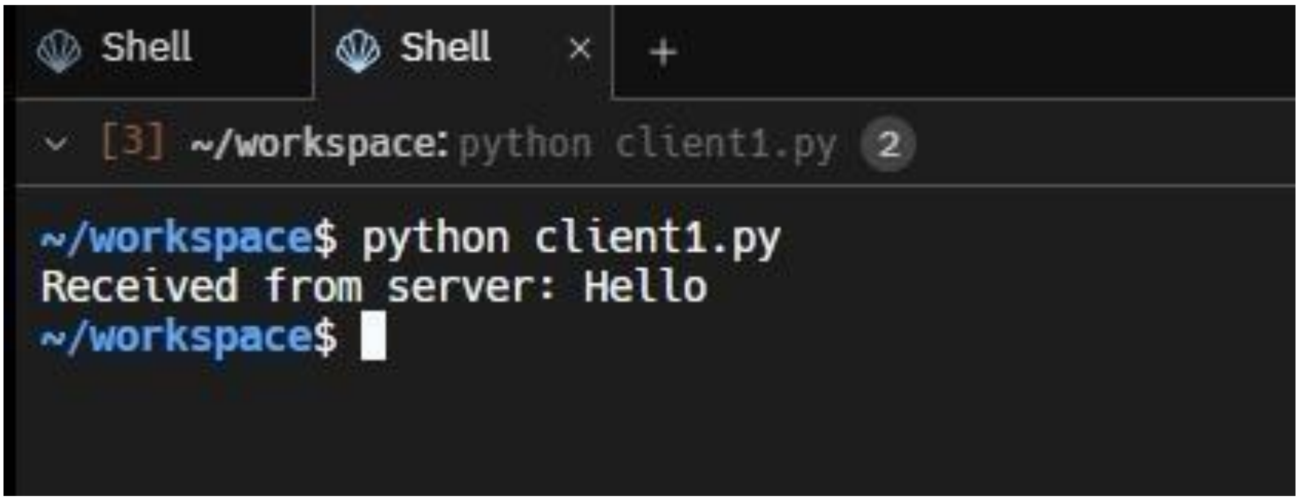
server_socket.close()

Outputs For Server And

Client



```
~/workspace$ python server1.py
Server is listening on port 5003...
Connection established with ('127.0.0.1', 46162)
Received from client: Hello
~/workspace$
```



```
~/workspace$ python client1.py
Received from server: Hello
~/workspace$
```

Conclusion:

This project successfully implemented a basic client-server architecture using stream sockets. The server listened on port **5003**, accepted a connection, and exchanged messages with the client. This experiment highlights the essential working of TCP-based communication, including socket creation, data transmission, and connection handling. Such concepts are foundational for building more advanced network applications.