Experiment 6: Socket Programming-I

Aim: To use TCP Sockets for Inter Process Communication

Objective: After carrying out this experiment, students will be able to:

- Apply TCP Socket programming technique to establish IPC between remote processes
- Analyse the difference between sockets and other enabling techniques for IPC such as Pipes and Message Queues

Problem statement: You are required to write programs to implement a TCP based echo server. The functionality of this server is that it should echo any data it receives from a client back to it.

Analysis: While analyzing your program, you are required to address the following points:

- How is socket programming different from other techniques for IPC such as Pipes and Message Queues?
- What happens if the number of incoming client requests exceeds the second argument of the `listen()` function in the server?

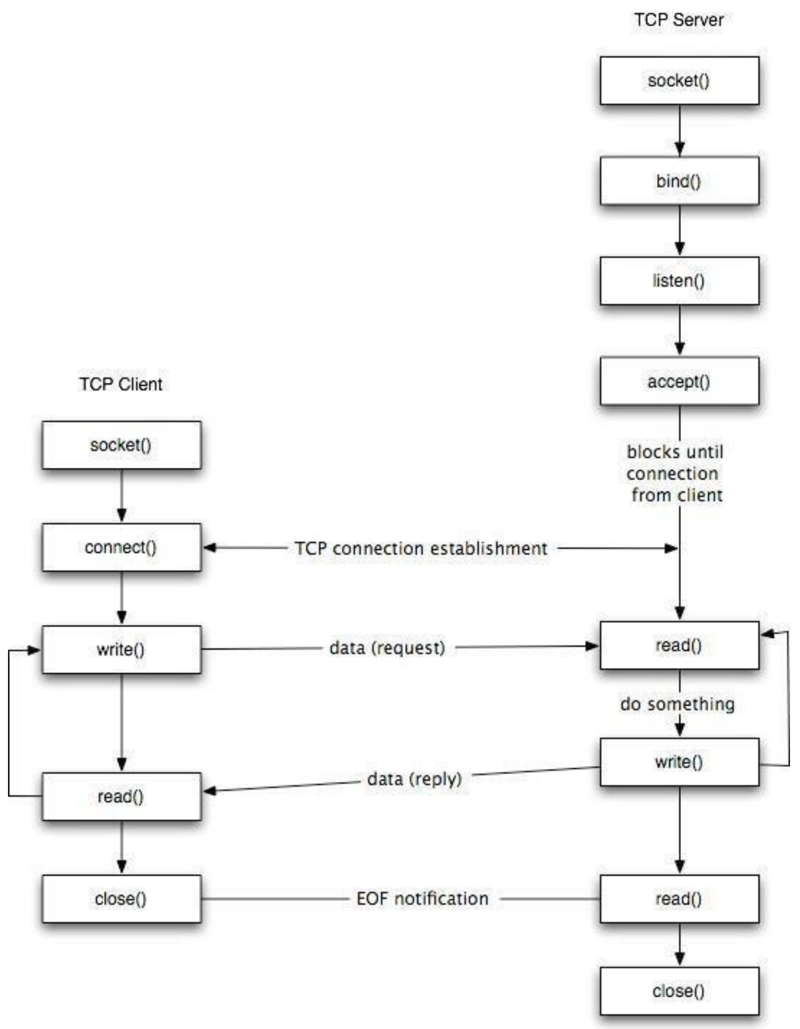
MARKS DISTRIBUTION

Component	Maximum Marks	Marks Obtained
Preparation of Document	7	
Results	7	
Viva	6	
Total	**20**	

Submitted by: Harshit Kumar

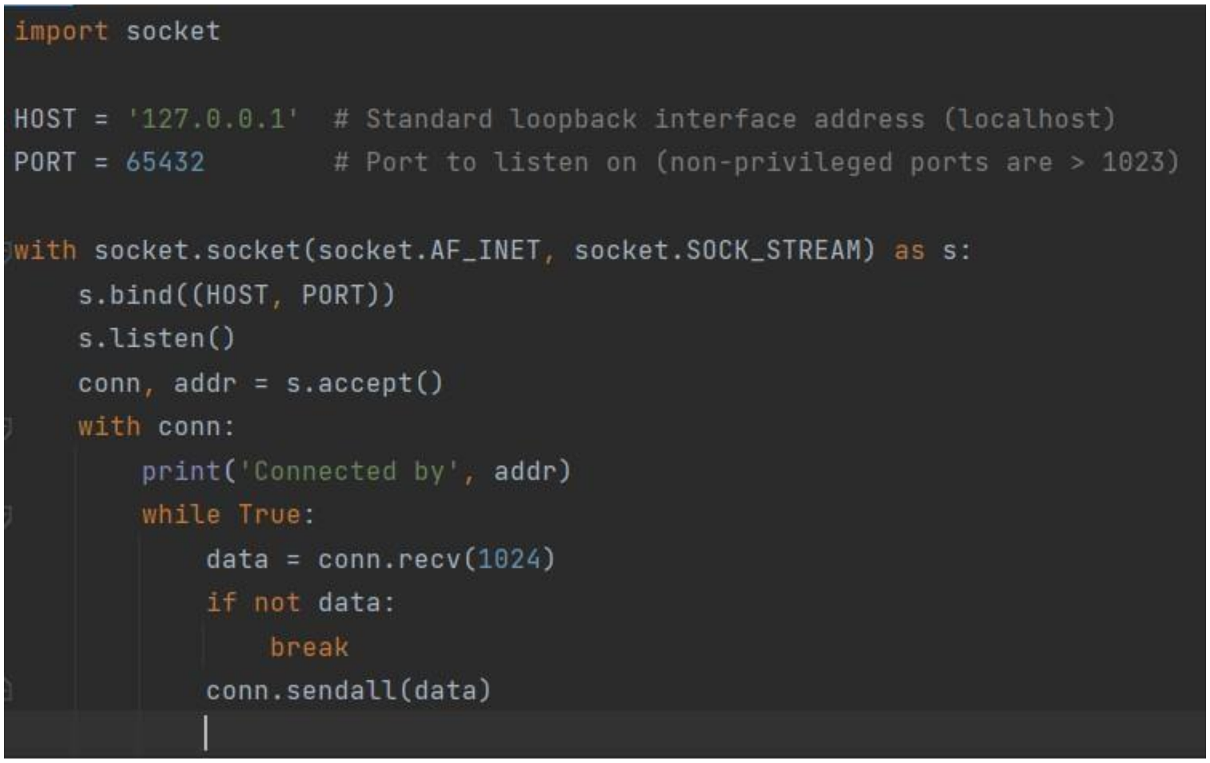
Register No: 21ETMC412011

1. Algorithm/Flowchart



2. Program

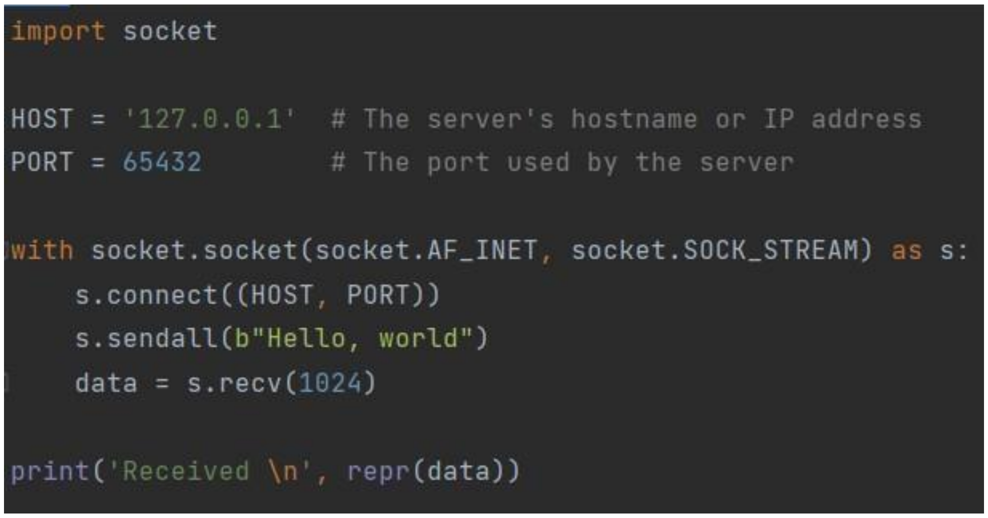
Server and client



```python
import socket

HOST = '127.0.0.1'  # Standard loopback interface address (localhost)
PORT = 65432        # Port to listen on (non-privileged ports are > 1023)

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    conn, addr = s.accept()
    with conn:
        print('Connected by', addr)
        while True:
            data = conn.recv(1024)
            if not data:
                break
            conn.sendall(data)
```



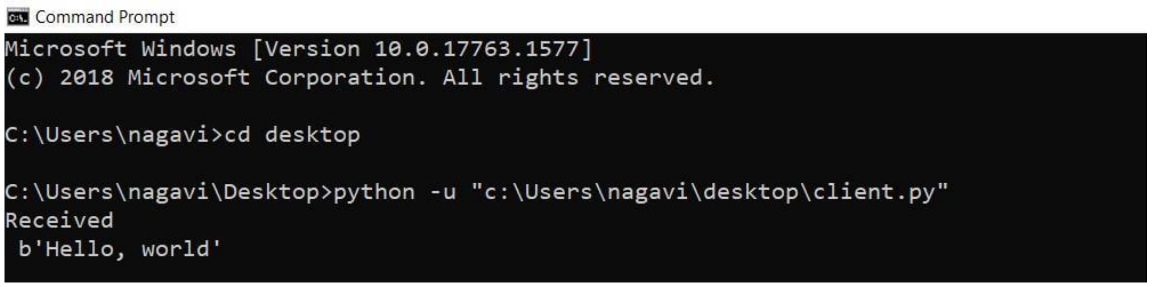
```python
import socket

HOST = '127.0.0.1'  # The server's hostname or IP address
PORT = 65432        # The port used by the server

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    s.sendall(b"Hello, world")
    data = s.recv(1024)

print('Received \n', repr(data))
```

3. Results

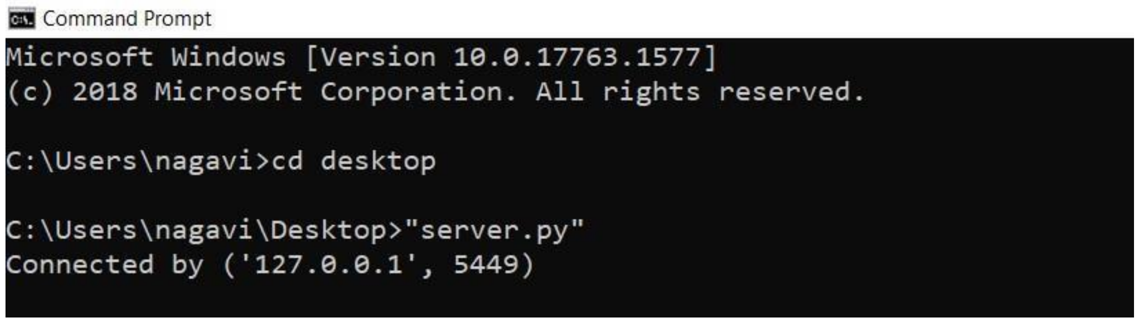


Command Prompt

```
Microsoft Windows [Version 10.0.17763.1577]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\nagavi>cd desktop

C:\Users\nagavi\Desktop>python -u "c:\Users\nagavi\desktop\client.py"
Received
 b'Hello, world'
```



Command Prompt

```
Microsoft Windows [Version 10.0.17763.1577]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\nagavi>cd desktop

C:\Users\nagavi\Desktop>"server.py"
Connected by ('127.0.0.1', 5449)
```

Analysis:

Answer[1.]: Socket programming facilitates network communication between processes on different machines, offering flexibility, platform independence, and support for various communication patterns compared to local IPC techniques like pipes and message queues.

Answer[2.]: If the number of incoming client requests exceeds the specified backlog limit in the `listen()`function, additional connection attempts are denied, potentially leading to connection refusal errors for clients and a risk of overflow in the pending connections queue, causing connection losses or delays.

Conclusion:

Understanding and implementing the technique of socket programming.

4. Comments

   a. Limitations of the experiment
   b. Limitations of the results obtained
   c. Learning
   d. Recommendations