

Assignment 4

Implementation of TCP Socket Programming

NAME: Shirish Manoj Bobde

Reg. No.: 812

Roll No.: ECE/21152

Problem Statement 1

Write a TCP socket program (in C/C++/Java/Python) to establish connection between client and server. The client program will send an input value n to the server and the server program will return the sum of the square of first n natural numbers. Client will display the value send by server. The communication between client and server will continue until client send 'Quit' message to the server.

Code

Server

```
import socket
port=50000
host="127.0.0.1"

server= socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
server.bind((host, port))
print("socket binded to %s" %(port))

server.listen(2)
print("Socket is listening...")

# Accepting/Establishing connection from client.
conn, addr = server.accept()
print('Got connection from', addr)

while True:
    recieved_data = conn.recv(2048)
    #print("Message from client: ",recieved_data.decode())
    if recieved_data.decode()=='quit':
        break
    else:
        n = recieved_data.decode()
        n = int(n)
        i=1
        sum = 0
        for i in range(n+1):
            sum += i*i
```

```
        print(i)
        print("Sum: ",sum)

        sum = str(sum)
        conn.send(sum.encode())

print("Connection closed from client")

#Close the connection with the client
conn.close()
```

Client

```
import socket
port=50000
portClient=8000
host="127.0.0.1"

client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.bind((host, portClient))
client.connect((host, port))

while True:
    data = input("Enter your message: ")
    client.send(data.encode())
    if data=='quit':
        break
    recieved_val = client.recv(2048)
    print("Echo Message from server: ",recieved_val.decode())

print("Connection closed from server")
client.close()
```

Results

```
1 import socket
2 port=50000
3 host="127.0.0.1"
4
5 server= socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6 server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
7 server.bind((host, port))
8 print("socket binded to %s" %(port))
9
10 server.listen(2)
11 print("Socket is listening...")
12
13 # Accepting/Establishing connection from client.
14 conn, addr = server.accept()
15 print('Got connection from', addr)
16
17 while True:
18     data = input("Enter your message: ")
19     client.send(data.encode())
20     if data=='quit':
21         break
22     recieved_val = client.recv(2048)
23     print("Echo Message from server: ",recieved_val.decode())
24
25 print("Connection closed from server")
```

```
PS C:\Users\ASUS> python -u "C:\Users\ASUS\Downloads\1server.py"
socket binded to 50000
Socket is listening...
Got connection from ('127.0.0.1', 8000)
0
1
2
3
4
5
Sum: 55
0
1
2
3
Sum: 14
Connection closed from client
```

```
1 import socket
2 port=50000
3 portClient=8000
4 host="127.0.0.1"
5
6 client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7 client.bind((host, portClient))
8 client.connect((host, port))
9
10 while True:
11     data = input("Enter your message: ")
12     client.send(data.encode())
13     if data=='quit':
14         break
15     recieved_val = client.recv(2048)
16     print("Echo Message from server: ",recieved_val.decode())
17
18 print("Connection closed from server")
```

```
PS C:\Users\ASUS> python -u "C:\Users\ASUS\Downloads\1client.py"
Enter your message: 5
Echo Message from server: 55
Enter your message: 3
Echo Message from server: 14
Enter your message: quit
Connection closed from server
PS C:\Users\ASUS>
```

Problem Statement 2

Write a TCP socket program (in C/C++/Java/Python) to establish connection between client and server. The client program will send a set of binary values to the server and the server program will return the number of 1s present in the data received. Client will display the value send by server. The communication between client and server will continue until client send 'Quit' message to the server.

Code

Server

```
import socket
port=50000
host="127.0.0.1"

server= socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
server.bind((host, port))
print("socket binded to %s" %(port))

server.listen(2)
print("Socket is listening...")

# Accepting/Establishing connection from client.
conn, addr = server.accept()
print('Got connection from', addr)

while True:
```

```

recieved_data = conn.recv(2048)
#print("Message from client: ",recieved_data.decode())
if recieved_data.decode()=='quit':
    break
else:
    n = recieved_data.decode()
    n = int(n)
    count = 0
    while (n):
        if n%2 == 1:
            count += 1
            n = n//10
        else:
            n = n//10
    print("No. of Ones: ",count)
    count = str(count)
    conn.send(count.encode())

print("Connection closed from client")

#Close the connection with the client
conn.close()

```

Client

```

import socket
port=50000
portClient=8000
host="127.0.0.1"

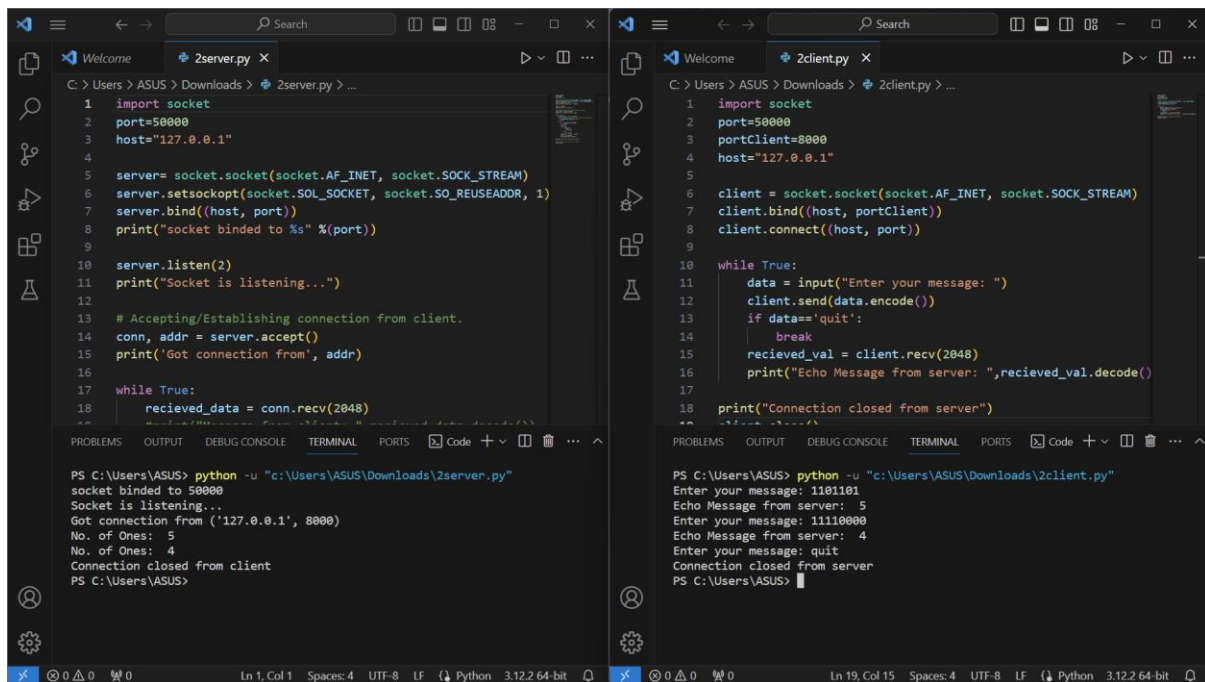
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.bind((host, portClient))
client.connect((host, port))

while True:
    data = input("Enter your message: ")
    client.send(data.encode())
    if data=='quit':
        break
    recieved_val = client.recv(2048)
    print("Echo Message from server: ",recieved_val.decode())

print("Connection closed from server")
client.close()

```

Results



```
1 import socket
2 port=50000
3 host="127.0.0.1"
4
5 server= socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6 server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
7 server.bind((host, port))
8 print("socket binded to %s" %(port))
9
10 server.listen(2)
11 print("Socket is listening...")
12
13 # Accepting/Establishing connection from client.
14 conn, addr = server.accept()
15 print('Got connection from', addr)
16
17 while True:
18     recieved_data = conn.recv(2048)
```

```
PS C:\Users\ASUS> python -u "C:\Users\ASUS\Downloads\2server.py"
socket binded to 50000
Socket is listening...
Got connection from ('127.0.0.1', 8000)
No. of Ones: 5
No. of Ones: 4
Connection closed from client
PS C:\Users\ASUS>
```

```
1 import socket
2 port=50000
3 portClient=8000
4 host="127.0.0.1"
5
6 client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7 client.bind((host, portClient))
8 client.connect((host, port))
9
10 while True:
11     data = input("Enter your message: ")
12     client.send(data.encode())
13     if data=='quit':
14         break
15     recieved_val = client.recv(2048)
16     print("Echo Message from server: ",recieved_val.decode())
17
18 print("Connection closed from server")
```

```
PS C:\Users\ASUS> python -u "C:\Users\ASUS\Downloads\2client.py"
Enter your message: 1101101
Echo Message from server: 5
Enter your message: 11110000
Echo Message from server: 4
Enter your message: quit
Connection closed from server
PS C:\Users\ASUS>
```

Problem Statement 3

Write a TCP socket program (in C/C++/Java/Python) to establish connection between client and server. The client program will send a postfix expression to the server and the server program will return the result of the input expression. Server program will use a stack to evaluate the postfix expression. Client will display the value send by server. The communication between client and server will continue until client send 'Quit' message to the server.

Code

Server

```
import socket

def evaluate_postfix(expression):
    stack = []
    operators = set(['+', '-', '*', '/'])

    for char in expression:
        if char.isdigit():
            stack.append(int(char))
        elif char in operators:
            operand2 = stack.pop()
            operand1 = stack.pop()

            if char == '+':
                result = operand1 + operand2
```

```

        elif char == '-':
            result = operand1 - operand2
        elif char == '*':
            result = operand1 * operand2
        elif char == '/':
            result = operand1 / operand2

        stack.append(result)

    return stack[0]

def start_server():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind(('127.0.0.1', 12345))
    server_socket.listen(1)

    print("Server listening on port 12345...")

    while True:
        client_socket, addr = server_socket.accept()
        print(f"Connection established with {addr}")

        while True:
            data = client_socket.recv(1024).decode('utf-8')

            if data.lower() == 'quit':
                print("Connection closed by client.")
                client_socket.close()
                break

            result = evaluate_postfix(data)
            client_socket.send(str(result).encode('utf-8'))

if __name__ == "__main__":
    start_server()

```

Client

```

import socket

def start_client():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect(('127.0.0.1', 12345))

    while True:
        postfix_expression = input("Enter a postfix expression (or type 'Quit' to exit): ")

```

```

        if postfix_expression.lower() == 'quit':
            client_socket.send('Quit'.encode('utf-8'))
            break

        client_socket.send(postfix_expression.encode('utf-8'))
        result = client_socket.recv(1024).decode('utf-8')
        print(f"Server response: Result of the expression: {result}")

    client_socket.close()

if __name__ == "__main__":
    start_client()

```

Results

The image shows two side-by-side Visual Studio Code windows. The left window displays the code for a postfix calculator server (3server.py) and its terminal output. The right window displays the code for a postfix calculator client (3client.py) and its terminal output.

Left Window (3server.py):

```

1 import socket
2
3 def evaluate_postfix(expression):
4     stack = []
5     operators = set(['+', '-', '*', '/'])
6
7     for char in expression:
8         if char.isdigit():
9             stack.append(int(char))
10        elif char in operators:
11            operand2 = stack.pop()
12            operand1 = stack.pop()
13
14            if char == '+':
15                result = operand1 + operand2
16            elif char == '-':
17                result = operand1 - operand2
18            elif char == '*':
19                result = operand1 * operand2
20            elif char == '/':
21                result = operand1 / operand2
22
23            stack.append(result)
24
25    return stack[-1]
26
27 if __name__ == '__main__':
28     server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
29     server_socket.bind(('127.0.0.1', 56413))
30     server_socket.listen(1)
31     print("Server listening on port 12345...")
32
33     while True:
34         client_socket, addr = server_socket.accept()
35         print(f"Connection established with {addr}")
36
37         postfix_expression = input("Enter a postfix expression (or type 'Quit' to exit): ")
38
39         if postfix_expression.lower() == 'quit':
40             client_socket.send('Quit'.encode('utf-8'))
41             break
42
43         client_socket.send(postfix_expression.encode('utf-8'))
44         result = client_socket.recv(1024).decode('utf-8')
45         print(f"Server response: Result of the expression: {result}")
46
47     client_socket.close()

```

Terminal Output (Left):

```

PS C:\Users\ASUS> python -u "c:\Users\ASUS\Downloads\3server.py"
Server listening on port 12345...
Connection established with ('127.0.0.1', 56413)

```

Right Window (3client.py):

```

1 import socket
2
3 def start_client():
4     client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5     client_socket.connect(('127.0.0.1', 12345))
6
7     while True:
8         postfix_expression = input("Enter a postfix expression (or type 'Quit' to exit): ")
9
10        if postfix_expression.lower() == 'quit':
11            client_socket.send('Quit'.encode('utf-8'))
12            break
13
14        client_socket.send(postfix_expression.encode('utf-8'))
15        result = client_socket.recv(1024).decode('utf-8')
16        print(f"Server response: Result of the expression: {result}")
17
18    client_socket.close()
19
20 if __name__ == '__main__':
21     start_client()

```

Terminal Output (Right):

```

PS C:\Users\ASUS> python -u "c:\Users\ASUS\Downloads\3client.py"
Enter a postfix expression (or type 'Quit' to exit): 341+9-
Server response: Result of the expression: 6
Enter a postfix expression (or type 'Quit' to exit): 8456+-/6*
Server response: Result of the expression: -6.857142857142857
Enter a postfix expression (or type 'Quit' to exit):

```