

Assignment 7

Implementation of TCP/UDP Socket Programming

NAME: Shirish Manoj Bobde

Reg. No.: 812

Roll No.: ECE/21152

Problem Statement

Write a TCP socket program (in C/C++/Java/Python) to implement a client-server program using TCP sockets. The client will send a message to the server, and the server will perform a cyclic redundancy check (CRC) on the message to detect errors. The server will then send the result back to the client. Display appropriate messages to the user indicating the status of the connection and the result of the CRC check.

Code:

Server

```
import socket

def xor(a, b):
    result = []
    for i in range(1, len(b)):
        if a[i] == b[i]:
            result.append('0')
        else:
            result.append('1')
    return ''.join(result)

def mod2div(dividend, divisor):
    pick = len(divisor)
    tmp = dividend[0: pick]

    print("\nDivisor:", divisor)
    print("Dividend:", dividend)

    while pick < len(dividend):
        if tmp[0] == '1':
            tmp = xor(divisor, tmp) + dividend[pick]
        else:
            tmp = xor('0'*pick, tmp) + dividend[pick]
        pick += 1

    if tmp[0] == '1':
        tmp = xor(divisor, tmp)
```

```

else:
    tmp = xor('0'*pick, tmp)

    checkword = tmp
    return checkword

def receiveData(client_socket):
    # Receive data and CRC key from the client
    received_data = client_socket.recv(1024).decode()
    key, data = received_data.split(';')

    l_key = len(key)
    appended_data = data + '0'*(l_key-1)
    checksum = mod2div(appended_data, key)

    print("\nKey:", key)
    print("Original Data:", data)
    print("Checksum:", checksum)
    print("Data Sent by Client:", data + checksum )
    print("Verification Result:", "Error Detected!" if checksum != '0'*(l_key-1) else "No Errors Detected")

# Set up the server socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind(('localhost', 12345))
server_socket.listen(1)

print("Server is listening for incoming connections...")

# Accept a client connection
client_socket, client_address = server_socket.accept()
print("Connection established with", client_address)

# Receive and process data from the client
receiveData(client_socket)

# Close the sockets
client_socket.close()
server_socket.close()

```

Client

```
import socket

def xor(a, b):
    result = []
    for i in range(1, len(b)):
        if a[i] == b[i]:
            result.append('0')
        else:
            result.append('1')
    return ''.join(result)

def sendData(server_socket, data, key):
    l_key = len(key)
    appended_data = data + '0'*(l_key-1)

    print("\nKey:", key)
    print("Original Data:", data)

    # Send CRC key and data to the server
    server_socket.send((key + ';' + data).encode())

# Set up the client socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect(('localhost', 12345))

# Input the CRC key from the client
key = input("Enter the CRC Key: ")

# Input data to be sent
data = input("Enter the Data to be Sent: ")

# Send CRC key and data to the server
sendData(client_socket, data, key)

# Close the socket
client_socket.close()
```

Output

The image displays two side-by-side Visual Studio Code windows. The left window shows the code for `client.py`, and the right window shows the code for `server.py`. Both windows have a terminal pane at the bottom showing the execution of the programs.

client.py

```
1 import socket
2
3 def xor(a, b):
4     result = []
5     for i in range(1, len(b)):
6         if a[i] == b[i]:
7             result.append('0')
8         else:
9             result.append('1')
10    return ''.join(result)
11
12 def sendData(server_socket, data, key):
13     l_key = len(key)
14     appended_data = data + '0'*(l_key-1)
15
16     print("\nKey:", key)
17     print("Original Data:", data)
18
19     # Send CRC key and data to the server
```

server.py

```
1 import socket
2
3 def xor(a, b):
4     result = []
5     for i in range(1, len(b)):
6         if a[i] == b[i]:
7             result.append('0')
8         else:
9             result.append('1')
10    return ''.join(result)
11
12 def mod2div(dividend, divisor):
13     pick = len(divisor)
14     tmp = dividend[0: pick]
15
16     print("\nDivisor:", divisor)
17     print("Dividend:", dividend)
18
19     while pick < len(dividend):
20         if tmp[0] == '1':
```

Terminal Output for client.py:

```
PS C:\Users\ASUS> python -u "c:\Users\ASUS\Desktop\Lab Amit Sir\assignment 7\client.py"
Enter the CRC Key: 1001
Enter the Data to be Sent: 1100001

Key: 1001
Original Data: 1100001
PS C:\Users\ASUS>
```

Terminal Output for server.py:

```
Connection established with ('127.0.0.1', 56667)

Divisor: 1001
Dividend: 1100001000

Key: 1001
Original Data: 1100001
Checksum: 100
Data Sent by Client: 1100001100
Verification Result: Error Detected!
PS C:\Users\ASUS>
```