Experiment : 5
Date : 07/05/2021
Author : Bonnie Simon

Client Server Using Socket Programming

AIM

Implement Client-Server communication using Socket Programming and TCP as transport layer protocol.

THEORY

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server.
They are the real backbones behind web browsing. In simpler terms there is a server and a client.

ALGORITHM

Server.py

1. Start
2. Create a socket instance
3. Bind host with port
4. Listen for connection
5. Stop

Client.py

1. Start
2. Create socket instance
3. Connect to the port
4. Listen for Response
5. Stop

PROGRAM

Server.py

```python
import socket
import time

# create a socket object
serversocket = socket.socket(
                socket.AF_INET, socket.SOCK_STREAM)

# get local machine name
host = socket.gethostname()

port = 3000

# bind to the port
serversocket.bind((host, port))

# queue up to 5 requests
serversocket.listen(5)

while True:
    # establish a connection
    clientsocket,addr = serversocket.accept()

    print("Got a connection from %s" % str(addr))
    currentTime = time.ctime(time.time()) + "\r\n"
    clientsocket.send(currentTime.encode('ascii'))
    clientsocket.close()
```

Client.py

```python
import socket

# create a socket object
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# get local machine name
host = socket.gethostname()

port = 3000

# connection to hostname on the port.
s.connect((host, port))

# Receive no more than 1024 bytes
tm = s.recv(1024)

s.close()

print("Server responds at time %s" % tm.decode('ascii'))
```

OUTPUT

RESULT

The program to implement Client-Server communication using Socket Programming and TCP as transport layer protocol has been executed, verified successfully.