Assignment-8(Socket_Prog)

Name:Aditya Rathor
Roll No: B22AI044

Client Application:

```
import socket
import time


HEADER=64
PORT=50521
SERVER=socket.gethostbyname(socket.gethostname())
# print(socket.gethostname())
ADDR=(SERVER,PORT)
disconnect='abort'
```

```
client=socket.socket(socket.AF_INET,socket.SOCK_STREAM) #socket.socket makes
a new socket and socket.AF_INET tells that what type of IP adddress are we looking
for
client.connect(ADDR)
```

```
def send(msg):
message=msg.encode('utf-8')
msg_length=len(message)
send_length=str(msg_length).encode('utf-8')
send_length += b' '*(HEADER-len(send_length))
client.send(send_length)
client.send(message)
print(client.recv(2048).decode('utf-8'))
```

```
inp_user=input("ENTER THE MESSAGE YOU WANT TO SEND TO THE SERVER:
")
send(inp_user)
time.sleep(10) #added so that before aborting I can connect other clients to the same
server
send(disconnect)
```

Server Application

```
import socket
import threading


HEADER=64
PORT=50521
SERVER=socket.gethostbyname(socket.gethostname())
# print(socket.gethostname())
ADDR=(SERVER,PORT)
disconnect='abort'
```

```
server=socket.socket(socket.AF_INET,socket.SOCK_STREAM) #socket.socket makes a new socket and socket.AF_INET tells that what type of IP adddress are we looking for
server.bind(ADDR)
```

```
def handle_client(conn,addr): #running concurrently for each client
print(f"[new connection] {addr} connected.")
connected=True
while connected:
msg_length=conn.recv(HEADER).decode('utf-8')
if msg_length:
msg_length=int(msg_length)
msg=conn.recv(msg_length).decode('utf-8')
if msg == disconnect:
conn.send("Abort Message received".encode('utf-8'))
connected=False
print(f"[{addr}] {msg}")
if connected:
conn.send("Msg received by the server".encode('utf-8')) #To give a confirmation that message has being recived to the user
```

```
conn.close()
```

```
def start():
server.listen()
print(f"[LISTENING] Server is listening on {SERVER}")
try:
while True:
```

```
conn,addr=server.accept()
thread=threading.Thread(target=handle_client,args=(conn,addr)) #passing the new connection as thread to handle_client
thread.start()
print(f"[ACTIVE CONNECTION]{threading.active_count()-1}") #as one thread of start is always running
except KeyboardInterrupt:
print("Server closed")
```

```
# max_client=int(input("Enter the number of max client it can handle"))
print("[starting] server is starting")
start()
```

Documentation

1.**Socket**: Provides access to the BSD socket interface. It's used for communication between processes over the network.

2. **Threading**: Provides high-level threading API for creating and managing threads.

3.**Socket.gethostbyname()**: Resolves a hostname to its IP address.

4.**socket.AF_INET**: Address format of IPv4

5.**socket.SOCK_STREAM**: Socket type for TCP connections which follow TCP protocol.

6.**socket.bind()**: Binds the socket to a specific address and port.

7.**socket.listen()**: Puts the socket into listening mode to accept connections.

8.**socket.accept()**: Accepts an incoming connection and returns a new socket object and the address of the client.

9.**socket.recv()**: Receives data from the socket.

10.**socket.send()**: Sends data through the socket.

11.**encode('utf-8')/decode('utf-8'):** Converts strings to bytes and vice versa using UTF-8 encoding as messages are transferred in bytes but we want to display using strings.

12.**threading.Thread**: Creates a new thread.

13. **threading.active_count()**: Returns the number of Thread objects currently alive.

14. **KeyboardInterrupt**: Raised when the user interrupts the program execution, typically by pressing Ctrl+C.

15. **time.sleep()**: Suspends execution for the given number of seconds.

For server side:

- Socket Setup: It initializes a server socket (server) and binds it to the host address and port number.
- Handle Client Function: This function is executed for each client connection. It receives messages from clients and sends a confirmation back.
- Start Function: It listens for incoming connections and handles each connection in a separate thread.
- Main Execution: It starts the server by calling the start() function.

For client side:

- Socket Setup: It initializes a client socket (client) and connects it to the server.
- Send Function: It sends messages to the server. Messages are prefixed with their length for proper message parsing.
- Main Execution: It prompts the user for a message to send to the server, sends it, and then sends a disconnect message after a delay.

Testing:

To run the server script use:

python3 server.py

and for the client script use:

python3 client.py

Single Client Test: The server successfully receives and processes messages from the single client.

Multiple Clients Test:

- Each client successfully connects to the server, sends messages, and receives confirmations independently.
- Also when it exits it sends the abort message and server successfully removes that connection without affection others.
- Since the concept of threading is applied so it is able to handle multiple connection concurrently without causing errors

Below are the screenshots of testing:

- *Handling 2 clients using 1 server*

```
adi_techbuddy  python3 server.py
[starting] server is starting
[LISTENING] Server is listening on 127.0.1.1
[new connection] ('127.0.0.1', 38366) connected.
[ACTIVE CONNECTION]1
[('127.0.0.1', 38366)] Hello this is a new client
[new connection] ('127.0.0.1', 38194) connected.
[ACTIVE CONNECTION]2
[('127.0.0.1', 38194)] This is the client 2
[('127.0.0.1', 38194)] abort
[('127.0.0.1', 38366)] abort
^CServer closed

[~/Desktop/labs_pcs2/assign8]
adi_techbuddy
```

```
adi_techbuddy  python3 client.py
ENTER THE MESSAGE YOU WANT TO SEND TO THE SERVER: Hello this is a new client
Msg received by the server
Abort Message received

[~/Desktop/labs_pcs2/assign8]
adi_techbuddy
```

```
3: adi_techbuddy@aditechbuddy-IdeaPad-Gaming-3-15ACH6:~/Desktop/labs_pcs2/assign8

[~/Desktop/labs_pcs2/assign8]
adi_techbuddy  python3 client2.py
Msg received by the server
Abort Message received

[~/Desktop/labs_pcs2/assign8]
adi_techbuddy
```

- *Handling multiple clients using 1 server:*

```
[~/Desktop/labs_pcs2/assign8]
adi_techbuddy  python3 server.py
[starting] server is starting
[LISTENING] Server is listening on 127.0.1.1
[new connection] ('127.0.0.1', 46470) connected.
[ACTIVE CONNECTION]1
[('127.0.0.1', 46470)] Hello this is a new client
[new connection] ('127.0.0.1', 46480) connected.
[ACTIVE CONNECTION]2
[('127.0.0.1', 46480)] Client: ('127.0.0.1', 46480) - This is a message from
[('127.0.0.1', 46480)] Client: ('127.0.0.1', 46480) - abort
[new connection] ('127.0.0.1', 48708) connected.
[('127.0.0.1', 48708)] Client: ('127.0.0.1', 48708) - This is a message from
[ACTIVE CONNECTION]3
[('127.0.0.1', 48708)] Client: ('127.0.0.1', 48708) - abort
[new connection] ('127.0.0.1', 48716) connected.
[ACTIVE CONNECTION]4
[('127.0.0.1', 48716)] Client: ('127.0.0.1', 48716) - This is a message from
[('127.0.0.1', 48716)] Client: ('127.0.0.1', 48716) - abort
[new connection] ('127.0.0.1', 48724) connected.
[ACTIVE CONNECTION]5
[('127.0.0.1', 48724)] Client: ('127.0.0.1', 48724) - This is a message from
[('127.0.0.1', 48724)] Client: ('127.0.0.1', 48724) - abort
[new connection] ('127.0.0.1', 48736) connected.
[ACTIVE CONNECTION]6
[('127.0.0.1', 48736)] Client: ('127.0.0.1', 48736) - This is a message from
[('127.0.0.1', 48736)] Client: ('127.0.0.1', 48736) - abort
[new connection] ('127.0.0.1', 48750) connected.
[ACTIVE CONNECTION]7
[('127.0.0.1', 48750)] Client: ('127.0.0.1', 48750) - This is a message from
[('127.0.0.1', 48750)] Client: ('127.0.0.1', 48750) - abort
[('127.0.0.1', 46470)] abort
^CServer closed
```

```
[~/Desktop/labs_pcs2/assign8]
adi_techbuddy  python3 client.py
ENTER THE MESSAGE YOU WANT TO SEND TO THE SERVER: Hello this is a new client
Msg received by the server
Abort Message received

[~/Desktop/labs_pcs2/assign8]
adi_techbuddy
```

```
4: adi_techbuddy@aditechbuddy-IdeaPad-Gaming-3-15ACH6:~/Desktop/labs_pcs2/assign8
Msg received by the server
Msg received by the server
```

```
5: adi_techbuddy@aditechbuddy-IdeaPad-Gaming-3-15ACH6:~/Desktop/labs_pcs2/assign8
Msg received by the server
Msg received by the server
```

```
6: adi_techbuddy@aditechbuddy-IdeaPad-Gaming-3-15ACH6:~/Desktop/labs_pcs2/assign8
Msg received by the server
Msg received by the server
```

```
3: adi_techbuddy@aditechbuddy-IdeaPad-Gaming-3-15ACH6:~/Desktop/labs_pcs2/assign8

[~/Desktop/labs_pcs2/assign8]
adi_techbuddy  python3 client2.py
Msg received by the server
Msg received by the server

[~/Desktop/labs_pcs2/assign8]
adi_techbuddy
```

```
7: adi_techbuddy@aditechbuddy-IdeaPad-Gaming-3-15ACH6:~/Desktop/labs_pcs2/assign8
Msg received by the server
Msg received by the server

[~/Desktop/labs_pcs2/assign8]
```

```
8: adi_techbuddy@aditechbuddy-IdeaPad-Gaming-3-15ACH6:~/Desktop/labs_pcs2/assign8
adi_techbuddy  python3 client2.py
Msg received by the server
Msg received by the server
```