Lab 03: Single-Server Multi-Client Connection

Abhishek M J - CS21B2018

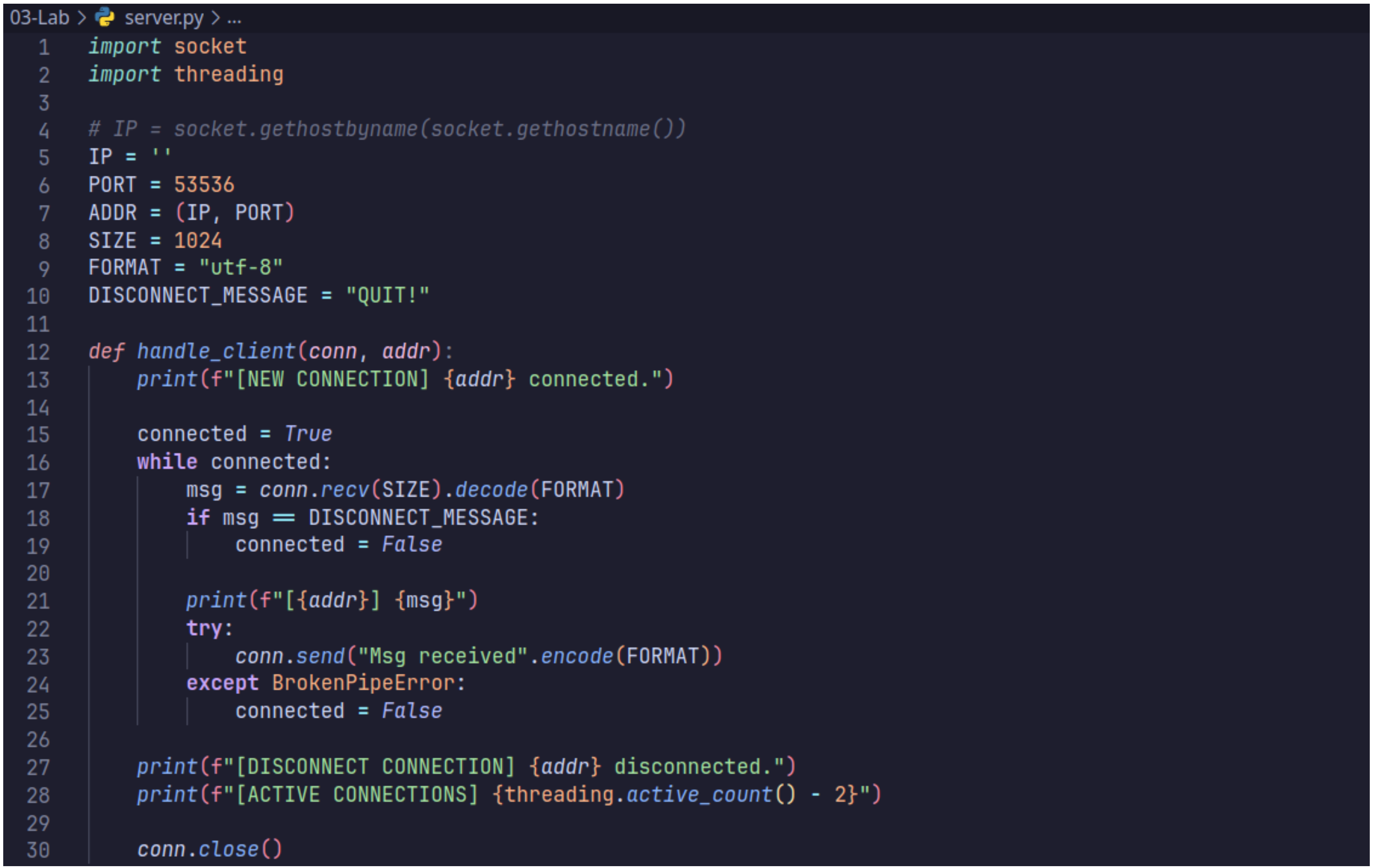
09-08-2023

Problem Statement

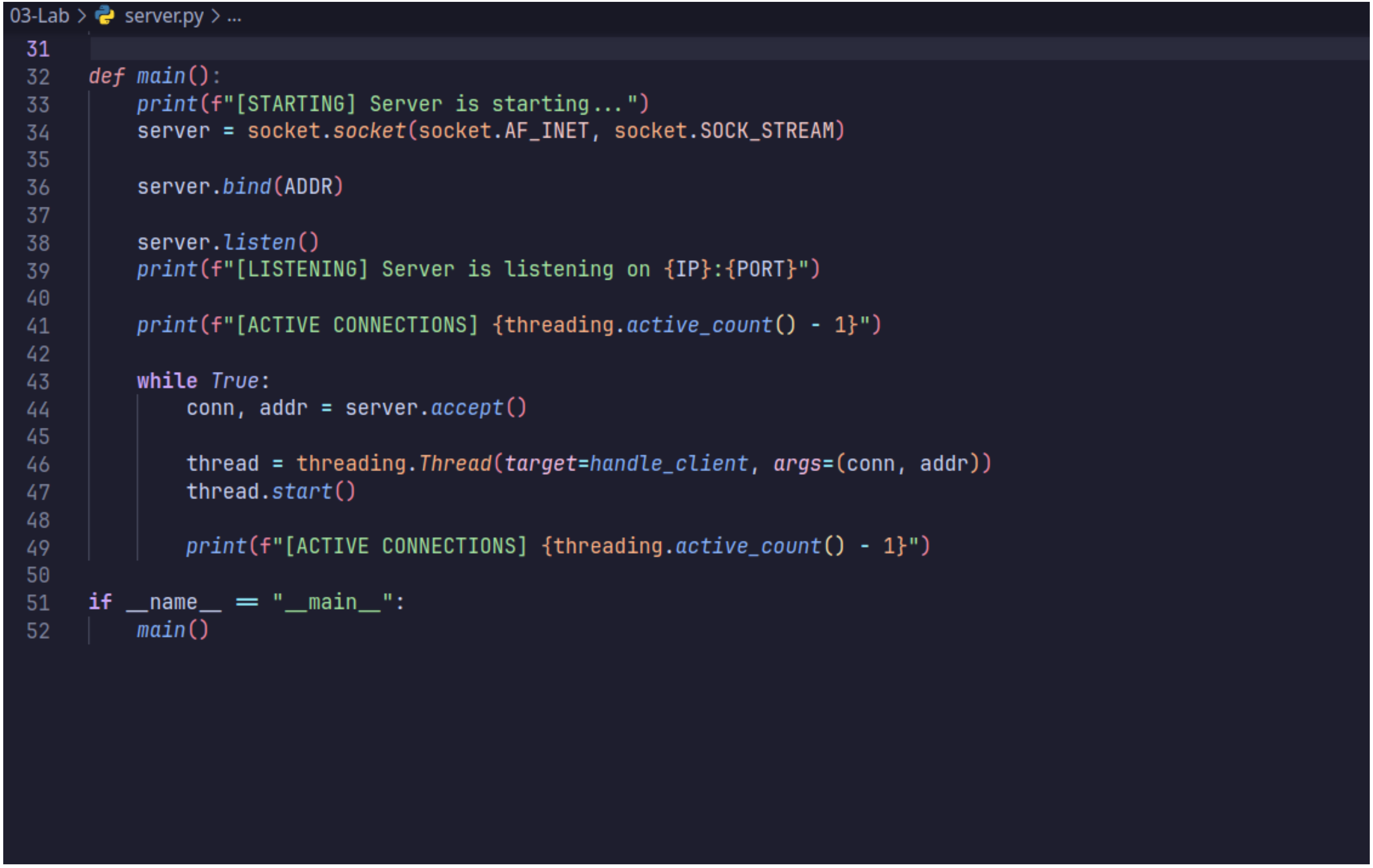
Write a Server program with threads, where each thread can handle a single client. - Ensure server get connected to multiple clients. - All the client messages to be displayed in server. - Provide a mechanism to disconnect the client.

Implementation 1: Demonstrate with local loop IP

`server.py`:

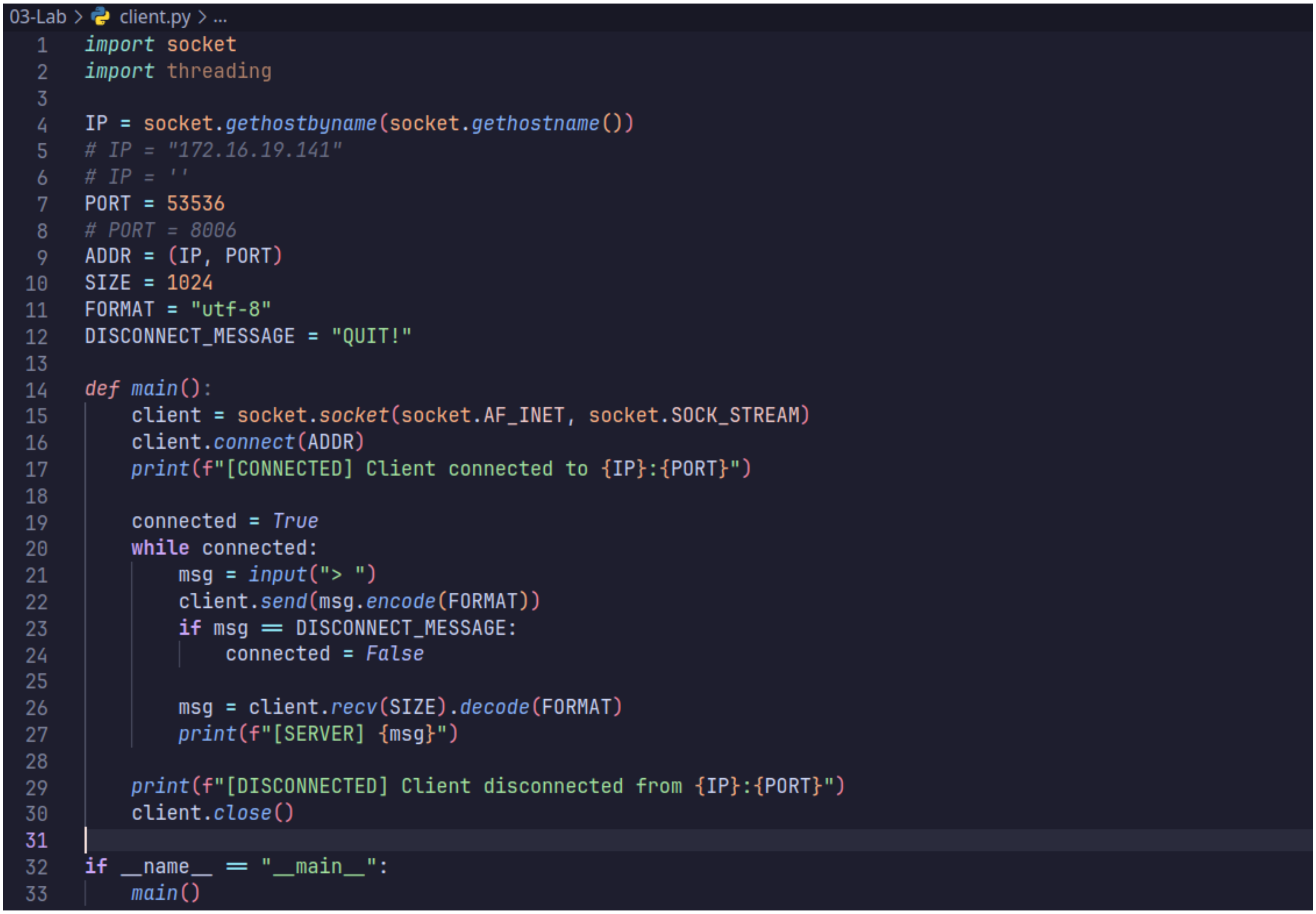


```python
import socket
import threading

# IP = socket.gethostbyname(socket.gethostname())
IP = ''
PORT = 53536
ADDR = (IP, PORT)
SIZE = 1024
FORMAT = "utf-8"
DISCONNECT_MESSAGE = "QUIT!"

def handle_client(conn, addr):
    print(f"[NEW CONNECTION] {addr} connected.")

    connected = True
    while connected:
        msg = conn.recv(SIZE).decode(FORMAT)
        if msg == DISCONNECT_MESSAGE:
            connected = False

        print(f"[{addr}] {msg}")
        try:
            conn.send("Msg received".encode(FORMAT))
        except BrokenPipeError:
            connected = False

    print(f"[DISCONNECT CONNECTION] {addr} disconnected.")
    print(f"[ACTIVE CONNECTIONS] {threading.active_count() - 2}")

    conn.close()
```



```python

def main():
    print(f"[STARTING] Server is starting...")
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    server.bind(ADDR)

    server.listen()
    print(f"[LISTENING] Server is listening on {IP}:{PORT}")

    print(f"[ACTIVE CONNECTIONS] {threading.active_count() - 1}")

    while True:
        conn, addr = server.accept()

        thread = threading.Thread(target=handle_client, args=(conn, addr))
        thread.start()

        print(f"[ACTIVE CONNECTIONS] {threading.active_count() - 1}")

if __name__ == "__main__":
    main()
```

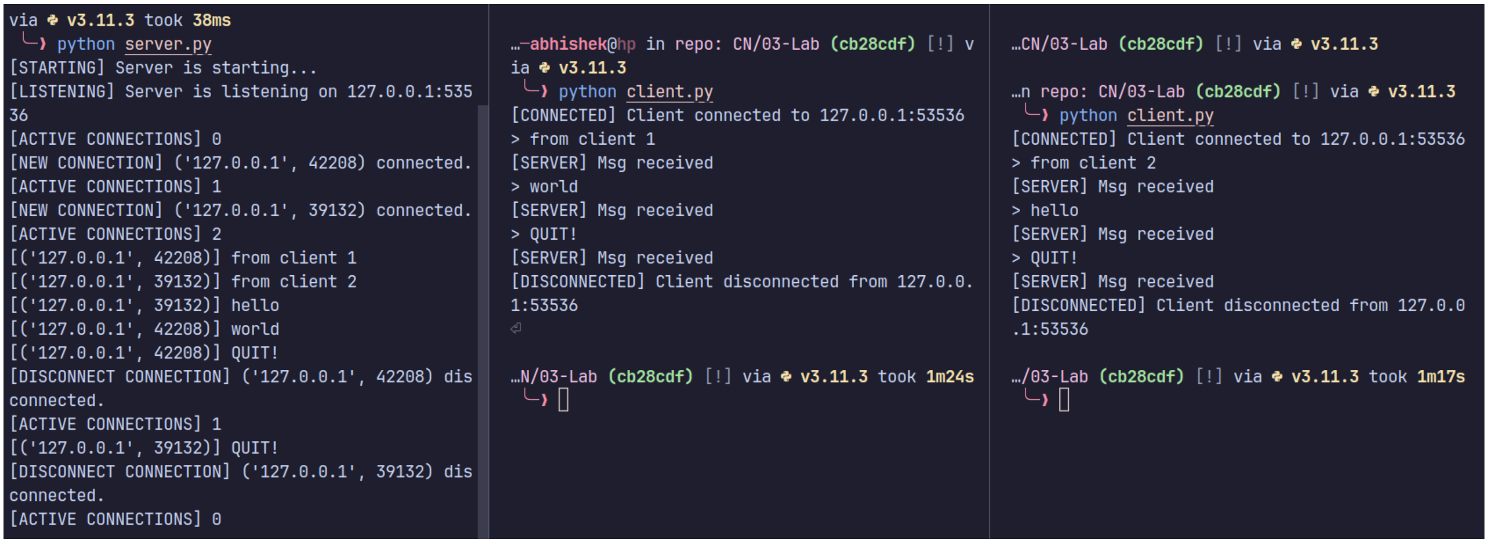
client.py:



```python
import socket
import threading

IP = socket.gethostbyname(socket.gethostname())
# IP = "172.16.19.141"
# IP = ''
PORT = 53536
# PORT = 8006
ADDR = (IP, PORT)
SIZE = 1024
FORMAT = "utf-8"
DISCONNECT_MESSAGE = "QUIT!"

def main():
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect(ADDR)
    print(f"[CONNECTED] Client connected to {IP}:{PORT}")

    connected = True
    while connected:
        msg = input("> ")
        client.send(msg.encode(FORMAT))
        if msg == DISCONNECT_MESSAGE:
            connected = False

        msg = client.recv(SIZE).decode(FORMAT)
        print(f"[SERVER] {msg}")

    print(f"[DISCONNECTED] Client disconnected from {IP}:{PORT}")
    client.close()

if __name__ == "__main__":
    main()
```

Figure 1: client-1-1

Output:



```
via v3.11.3 took 38ms
 ╰─❯ python server.py
[STARTING] Server is starting...
[LISTENING] Server is listening on 127.0.0.1:535
36
[ACTIVE CONNECTIONS] 0
[NEW CONNECTION] ('127.0.0.1', 42208) connected.
[ACTIVE CONNECTIONS] 1
[NEW CONNECTION] ('127.0.0.1', 39132) connected.
[ACTIVE CONNECTIONS] 2
[('127.0.0.1', 42208)] from client 1
[('127.0.0.1', 39132)] from client 2
[('127.0.0.1', 39132)] hello
[('127.0.0.1', 42208)] world
[('127.0.0.1', 42208)] QUIT!
[DISCONNECT CONNECTION] ('127.0.0.1', 42208) dis
connected.
[ACTIVE CONNECTIONS] 1
[('127.0.0.1', 39132)] QUIT!
[DISCONNECT CONNECTION] ('127.0.0.1', 39132) dis
connected.
[ACTIVE CONNECTIONS] 0
```

```
…─abhishek@hp in repo: CN/03-Lab (cb28cdf) [!] v
ia v3.11.3
 ╰─❯ python client.py
[CONNECTED] Client connected to 127.0.0.1:53536
> from client 1
[SERVER] Msg received
> world
[SERVER] Msg received
> QUIT!
[SERVER] Msg received
[DISCONNECTED] Client disconnected from 127.0.0.
1:53536
⏎

…N/03-Lab (cb28cdf) [!] via v3.11.3 took 1m24s
 ╰─❯ ▯
```

```
…CN/03-Lab (cb28cdf) [!] via v3.11.3

…n repo: CN/03-Lab (cb28cdf) [!] via v3.11.3
 ╰─❯ python client.py
[CONNECTED] Client connected to 127.0.0.1:53536
> from client 2
[SERVER] Msg received
> hello
[SERVER] Msg received
> QUIT!
[SERVER] Msg received
[DISCONNECTED] Client disconnected from 127.0.0
.1:53536

…/03-Lab (cb28cdf) [!] via v3.11.3 took 1m17s
 ╰─❯ ▯
```

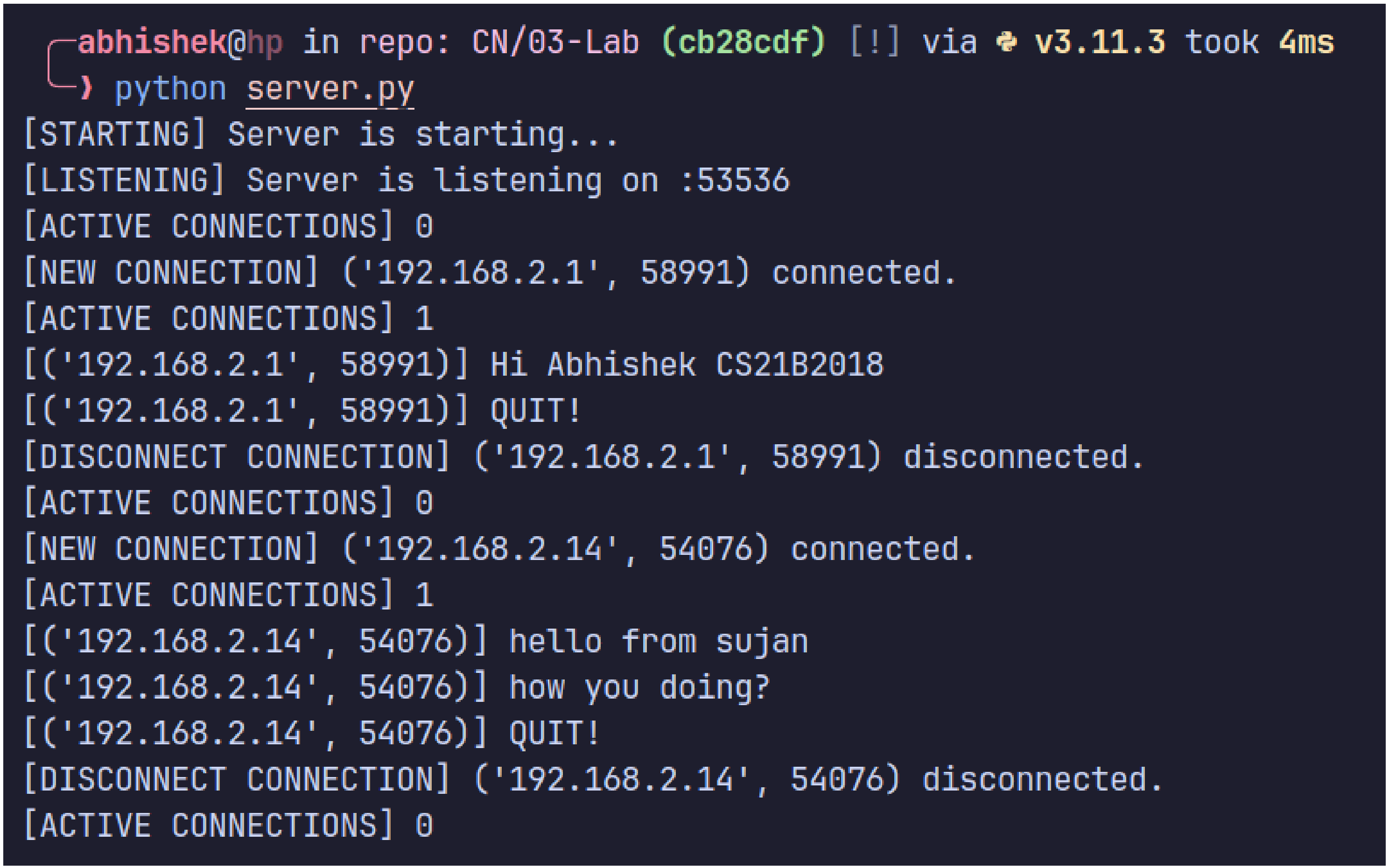
Figure 2: Output Implementation 1

Implementation 2: Connect with multiple client with different IPs

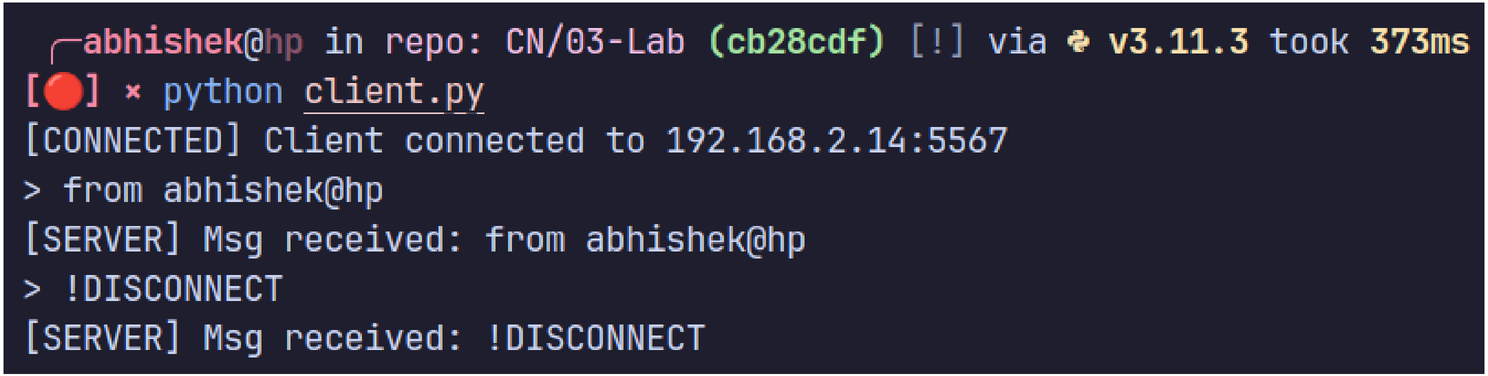
`server.py` and `client.py`:

Same as above

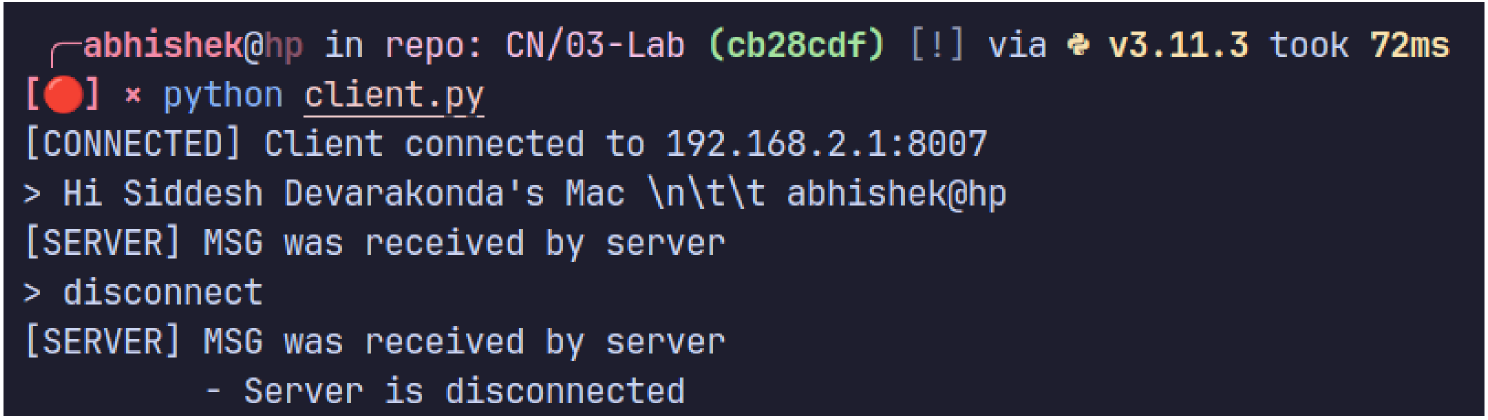
Output:



```
 ╭─abhishek@hp in repo: CN/03-Lab (cb28cdf) [!] via 🐍 v3.11.3 took 4ms
 ╰─❯ python server.py
[STARTING] Server is starting...
[LISTENING] Server is listening on :53536
[ACTIVE CONNECTIONS] 0
[NEW CONNECTION] ('192.168.2.1', 58991) connected.
[ACTIVE CONNECTIONS] 1
[('192.168.2.1', 58991)] Hi Abhishek CS21B2018
[('192.168.2.1', 58991)] QUIT!
[DISCONNECT CONNECTION] ('192.168.2.1', 58991) disconnected.
[ACTIVE CONNECTIONS] 0
[NEW CONNECTION] ('192.168.2.14', 54076) connected.
[ACTIVE CONNECTIONS] 1
[('192.168.2.14', 54076)] hello from sujan
[('192.168.2.14', 54076)] how you doing?
[('192.168.2.14', 54076)] QUIT!
[DISCONNECT CONNECTION] ('192.168.2.14', 54076) disconnected.
[ACTIVE CONNECTIONS] 0
```



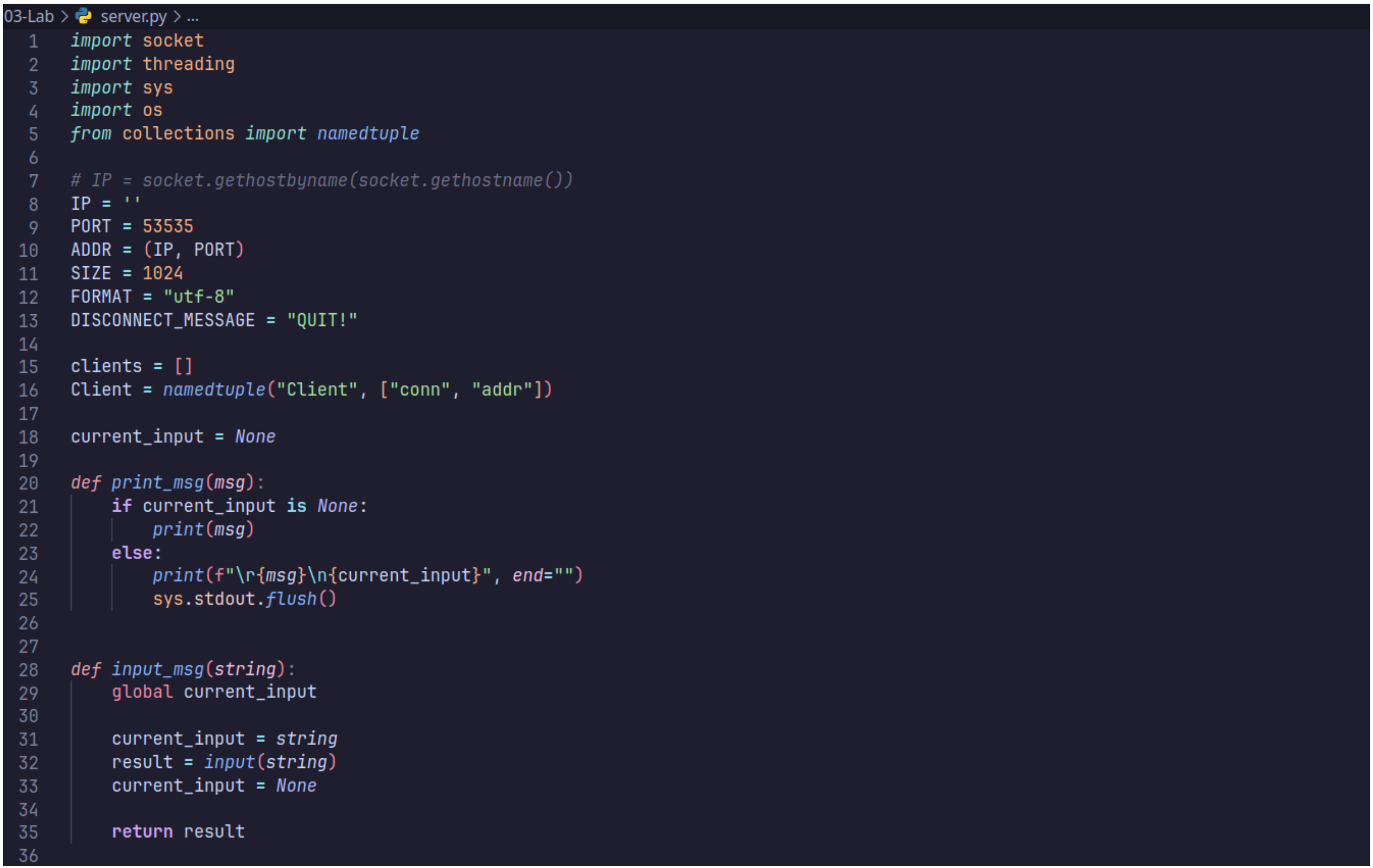
```
 ╭─abhishek@hp in repo: CN/03-Lab (cb28cdf) [!] via 🐍 v3.11.3 took 373ms
[🔴] × python client.py
[CONNECTED] Client connected to 192.168.2.14:5567
> from abhishek@hp
[SERVER] Msg received: from abhishek@hp
> !DISCONNECT
[SERVER] Msg received: !DISCONNECT
```



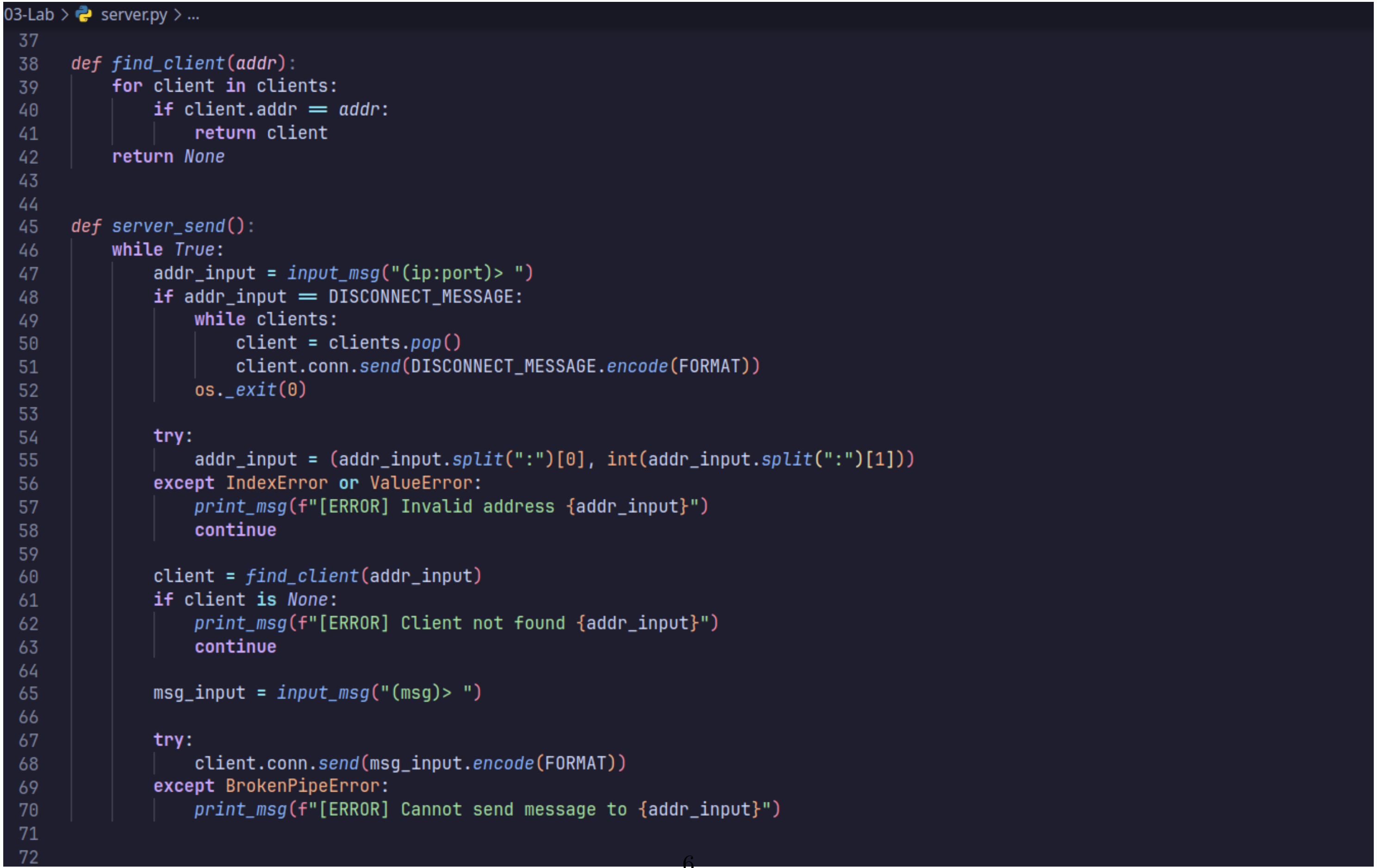
```
 ╭─abhishek@hp in repo: CN/03-Lab (cb28cdf) [!] via 🐍 v3.11.3 took 72ms
[🔴] × python client.py
[CONNECTED] Client connected to 192.168.2.1:8007
> Hi Siddesh Devarakonda's Mac \n\t\t abhishek@hp
[SERVER] MSG was received by server
> disconnect
[SERVER] MSG was received by server
         - Server is disconnected
```

Implementation 3: Modify the program where server can send messages to specific client

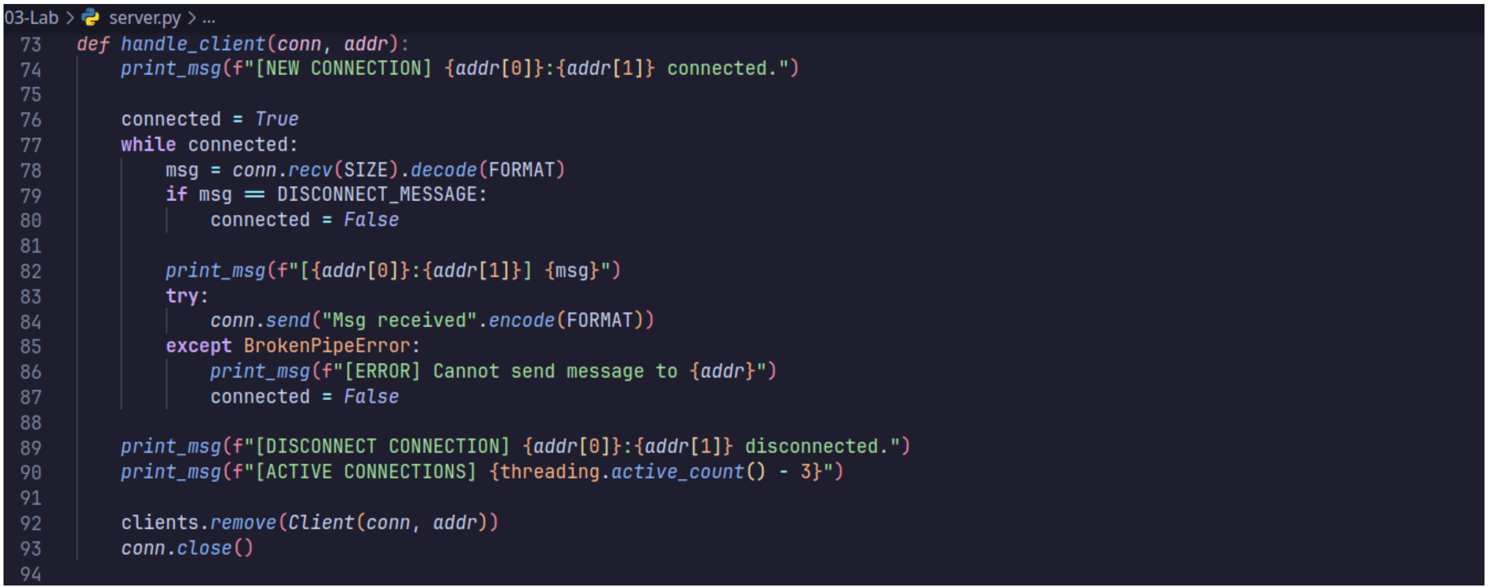
`server.py`:



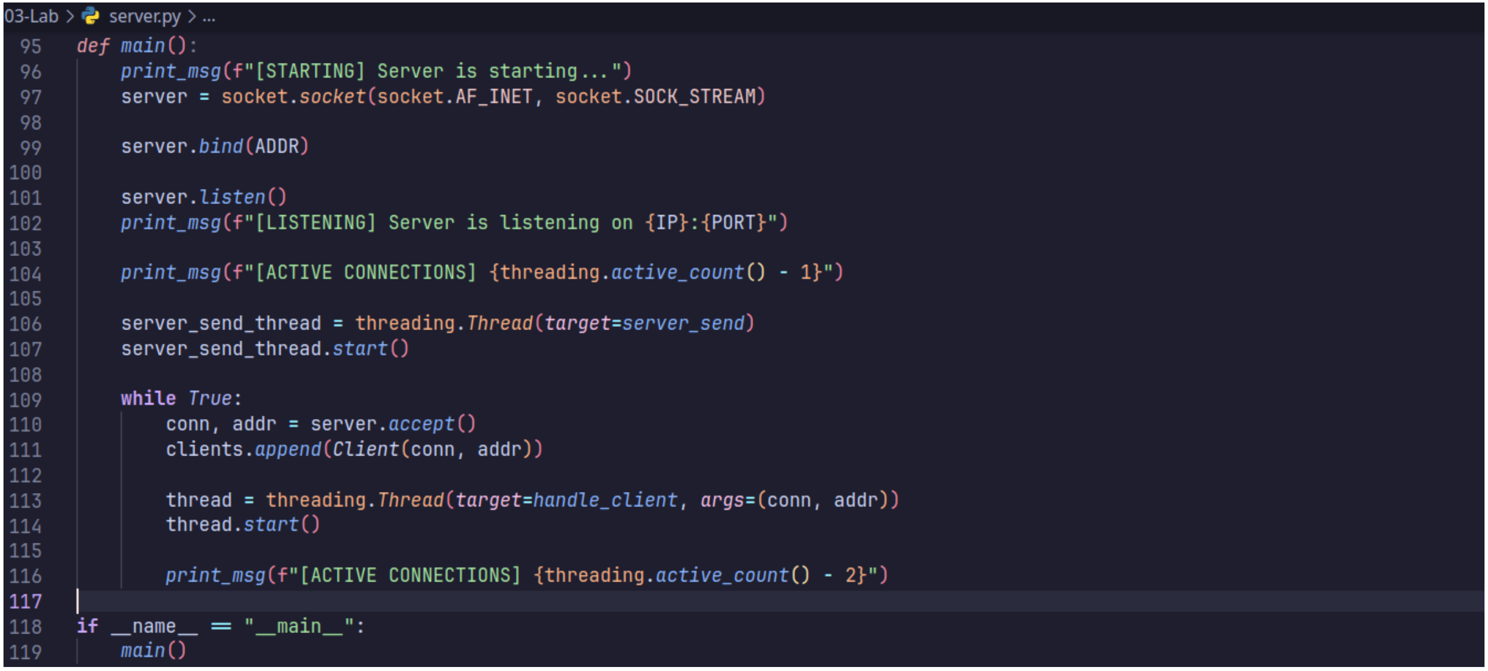
```python
import socket
import threading
import sys
import os
from collections import namedtuple

# IP = socket.gethostbyname(socket.gethostname())
IP = ''
PORT = 53535
ADDR = (IP, PORT)
SIZE = 1024
FORMAT = "utf-8"
DISCONNECT_MESSAGE = "QUIT!"

clients = []
Client = namedtuple("Client", ["conn", "addr"])

current_input = None

def print_msg(msg):
    if current_input is None:
        print(msg)
    else:
        print(f"\r{msg}\n{current_input}", end="")
        sys.stdout.flush()


def input_msg(string):
    global current_input

    current_input = string
    result = input(string)
    current_input = None

    return result

```



```python

def find_client(addr):
    for client in clients:
        if client.addr == addr:
            return client
    return None


def server_send():
    while True:
        addr_input = input_msg("(ip:port)> ")
        if addr_input == DISCONNECT_MESSAGE:
            while clients:
                client = clients.pop()
                client.conn.send(DISCONNECT_MESSAGE.encode(FORMAT))
            os._exit(0)

        try:
            addr_input = (addr_input.split(":")[0], int(addr_input.split(":")[1]))
        except IndexError or ValueError:
            print_msg(f"[ERROR] Invalid address {addr_input}")
            continue

        client = find_client(addr_input)
        if client is None:
            print_msg(f"[ERROR] Client not found {addr_input}")
            continue

        msg_input = input_msg("(msg)> ")

        try:
            client.conn.send(msg_input.encode(FORMAT))
        except BrokenPipeError:
            print_msg(f"[ERROR] Cannot send message to {addr_input}")


```

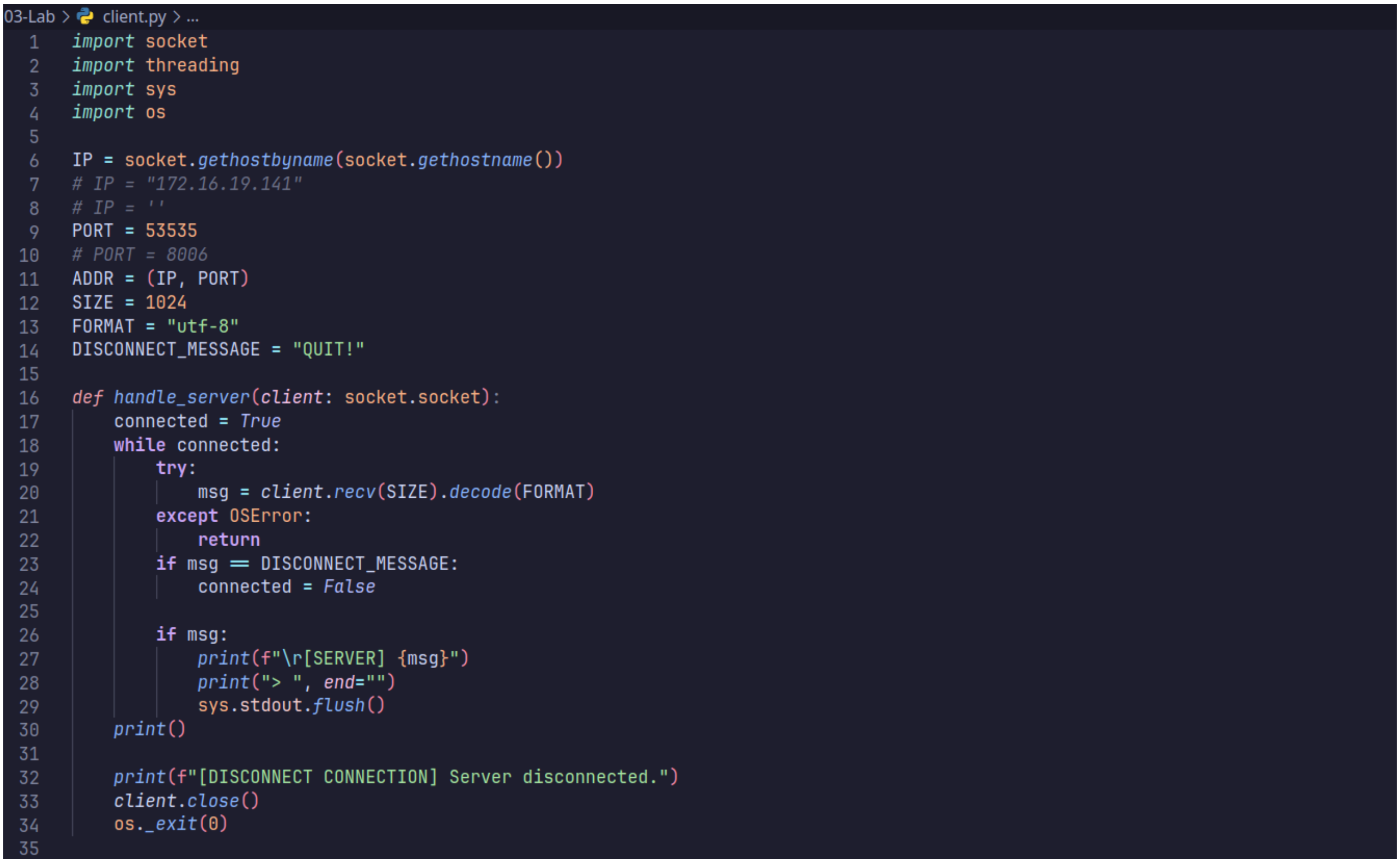


```python
def handle_client(conn, addr):
    print_msg(f"[NEW CONNECTION] {addr[0]}:{addr[1]} connected.")

    connected = True
    while connected:
        msg = conn.recv(SIZE).decode(FORMAT)
        if msg == DISCONNECT_MESSAGE:
            connected = False

        print_msg(f"[{addr[0]}:{addr[1]}] {msg}")
        try:
            conn.send("Msg received".encode(FORMAT))
        except BrokenPipeError:
            print_msg(f"[ERROR] Cannot send message to {addr}")
            connected = False

    print_msg(f"[DISCONNECT CONNECTION] {addr[0]}:{addr[1]} disconnected.")
    print_msg(f"[ACTIVE CONNECTIONS] {threading.active_count() - 3}")

    clients.remove(Client(conn, addr))
    conn.close()

```

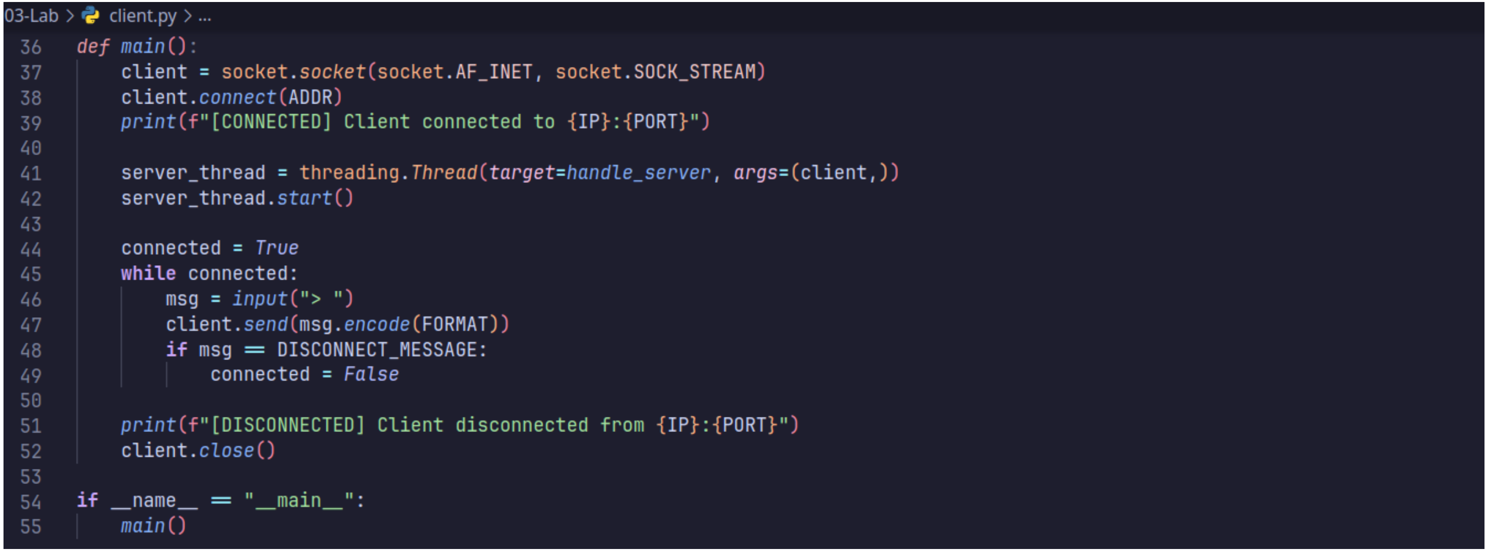


```python
def main():
    print_msg(f"[STARTING] Server is starting...")
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    server.bind(ADDR)

    server.listen()
    print_msg(f"[LISTENING] Server is listening on {IP}:{PORT}")

    print_msg(f"[ACTIVE CONNECTIONS] {threading.active_count() - 1}")

    server_send_thread = threading.Thread(target=server_send)
    server_send_thread.start()

    while True:
        conn, addr = server.accept()
        clients.append(Client(conn, addr))

        thread = threading.Thread(target=handle_client, args=(conn, addr))
        thread.start()

        print_msg(f"[ACTIVE CONNECTIONS] {threading.active_count() - 2}")

if __name__ == "__main__":
    main()
```

client.py:

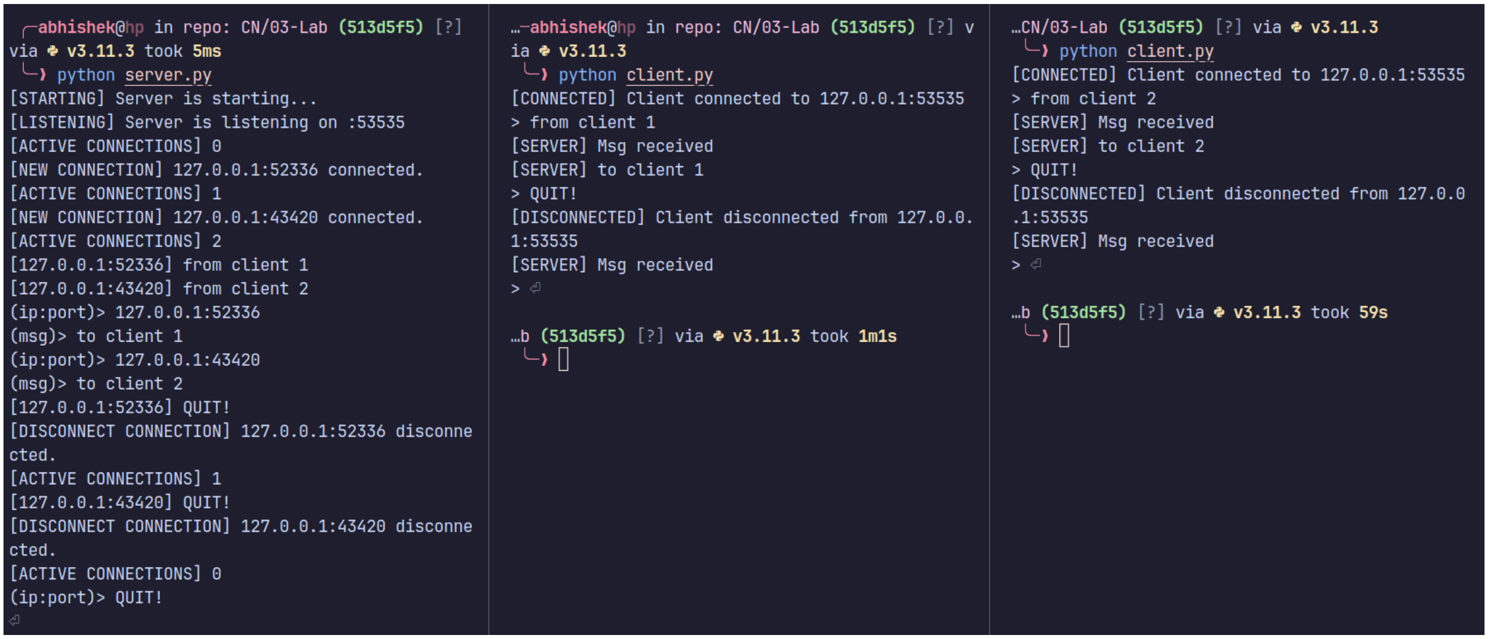


```python
import socket
import threading
import sys
import os

IP = socket.gethostbyname(socket.gethostname())
# IP = "172.16.19.141"
# IP = ''
PORT = 53535
# PORT = 8006
ADDR = (IP, PORT)
SIZE = 1024
FORMAT = "utf-8"
DISCONNECT_MESSAGE = "QUIT!"

def handle_server(client: socket.socket):
    connected = True
    while connected:
        try:
            msg = client.recv(SIZE).decode(FORMAT)
        except OSError:
            return
        if msg == DISCONNECT_MESSAGE:
            connected = False

        if msg:
            print(f"\r[SERVER] {msg}")
            print("> ", end="")
            sys.stdout.flush()
    print()

    print(f"[DISCONNECT CONNECTION] Server disconnected.")
    client.close()
    os._exit(0)

```



```python
def main():
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect(ADDR)
    print(f"[CONNECTED] Client connected to {IP}:{PORT}")

    server_thread = threading.Thread(target=handle_server, args=(client,))
    server_thread.start()

    connected = True
    while connected:
        msg = input("> ")
        client.send(msg.encode(FORMAT))
        if msg == DISCONNECT_MESSAGE:
            connected = False

    print(f"[DISCONNECTED] Client disconnected from {IP}:{PORT}")
    client.close()

if __name__ == "__main__":
    main()
```

Output:



```
╭─abhishek@hp in repo: CN/03-Lab (513d5f5) [?]
via 🐍 v3.11.3 took 5ms
╰─❯ python server.py
[STARTING] Server is starting...
[LISTENING] Server is listening on :53535
[ACTIVE CONNECTIONS] 0
[NEW CONNECTION] 127.0.0.1:52336 connected.
[ACTIVE CONNECTIONS] 1
[NEW CONNECTION] 127.0.0.1:43420 connected.
[ACTIVE CONNECTIONS] 2
[127.0.0.1:52336] from client 1
[127.0.0.1:43420] from client 2
(ip:port)> 127.0.0.1:52336
(msg)> to client 1
(ip:port)> 127.0.0.1:43420
(msg)> to client 2
[127.0.0.1:52336] QUIT!
[DISCONNECT CONNECTION] 127.0.0.1:52336 disconne
cted.
[ACTIVE CONNECTIONS] 1
[127.0.0.1:43420] QUIT!
[DISCONNECT CONNECTION] 127.0.0.1:43420 disconne
cted.
[ACTIVE CONNECTIONS] 0
(ip:port)> QUIT!
⏎
```

```
…─abhishek@hp in repo: CN/03-Lab (513d5f5) [?] v
ia 🐍 v3.11.3
 ╰─❯ python client.py
[CONNECTED] Client connected to 127.0.0.1:53535
> from client 1
[SERVER] Msg received
[SERVER] to client 1
> QUIT!
[DISCONNECTED] Client disconnected from 127.0.0.
1:53535
[SERVER] Msg received
> ⏎

…b (513d5f5) [?] via 🐍 v3.11.3 took 1m1s
 ╰─❯ ▯
```

```
…CN/03-Lab (513d5f5) [?] via 🐍 v3.11.3
 ╰─❯ python client.py
[CONNECTED] Client connected to 127.0.0.1:53535
> from client 2
[SERVER] Msg received
[SERVER] to client 2
> QUIT!
[DISCONNECTED] Client disconnected from 127.0.0
.1:53535
[SERVER] Msg received
> ⏎

…b (513d5f5) [?] via 🐍 v3.11.3 took 59s
 ╰─❯ ▯
```

Figure 3: Output Implementation 3