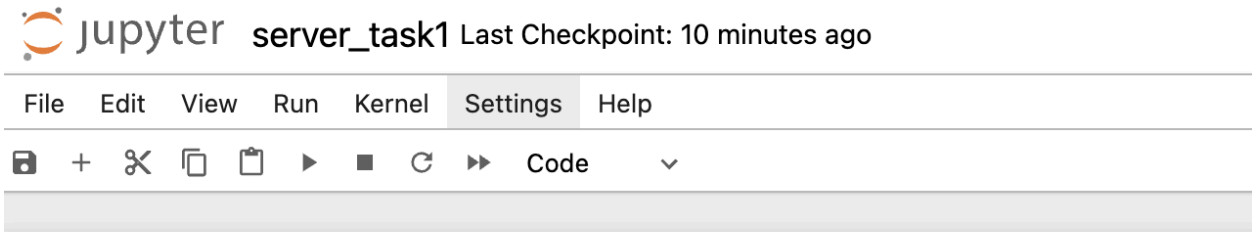
Lab Tasks

1. Design a simple Client Server chat application which sends files using the above concepts.



[*]:
```python
import socket

def start_server():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind(('localhost', 9999))
    server_socket.listen(5)
    print("Server is listening on port 9999...")

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

        file_name = client_socket.recv(1024).decode()
        print(f"Receiving file: {file_name}")

        with open(file_name, 'wb') as file:
            while True:
                data = client_socket.recv(1024)
                if not data:
                    break
                file.write(data)

        print(f"File {file_name} received successfully.")
        client_socket.close()

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

```
Server is listening on port 9999...
```

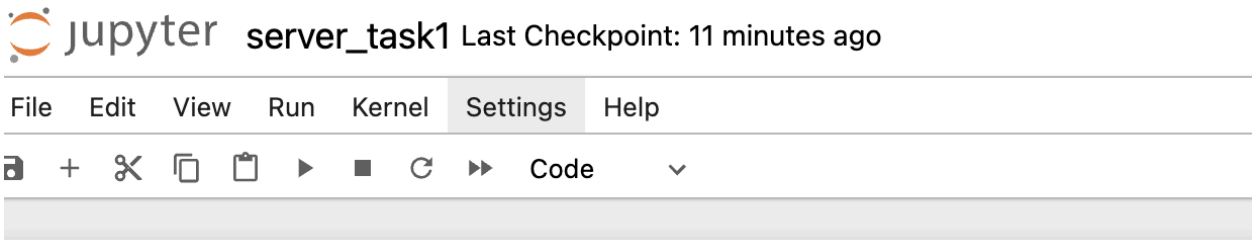
jupyter client_task1 Last Checkpoint: 9 minutes ago

File Edit View Run Kernel Settings Help

Code

[3]:
```
import socket

def send_file(file_name):
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect(('localhost', 9999))

    client_socket.send(file_name.encode())

    with open(file_name, 'rb') as file:
        while True:
            data = file.read(1024)
            if not data:
                break
            client_socket.send(data)

    print(f"File {file_name} sent successfully.")
    client_socket.close()

if __name__ == "__main__":
    file_name = input("Enter the file name to send: ")
    send_file(file_name)
```

```
Enter the file name to send:  CN Lab 03 - Tasks
File CN Lab 03 - Tasks sent successfully.
```



[*]:
```python
import socket

def start_server():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind(('localhost', 9999))
    server_socket.listen(5)
    print("Server is listening on port 9999...")

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

        file_name = client_socket.recv(1024).decode()
        print(f"Receiving file: {file_name}")

        with open(file_name, 'wb') as file:
            while True:
                data = client_socket.recv(1024)
                if not data:
                    break
                file.write(data)

        print(f"File {file_name} received successfully.")
        client_socket.close()

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

```
Server is listening on port 9999...
Connection established with ('127.0.0.1', 50758)
Receiving file: CN Lab 03 - Tasks
File CN Lab 03 - Tasks received successfully.
```

2. Develop a chat server application that can handle multiple clients at once, allowing them to communicate in a chat room-style application.

pyter chat_server_task_2 Last Checkpoint: 21 minutes ago

dit View Run Kernel Settings Help

Code

```
import socket
import threading

clients = {}
def broadcast(message, sender_socket):
    for client_socket in clients:
        if client_socket != sender_socket:
            try:
                client_socket.send(message.encode())
            except:
                del clients[client_socket]

def handle_client(client_socket, addr):
    print(f"New connection from {addr}")
    clients[client_socket] = addr

    while True:
        try:
            message = client_socket.recv(1024).decode()
            if not message:
                break

            if message.startswith("@"):
                target_client_id, private_message = message.split(" ", 1)
                target_client_id = target_client_id[1:]
                for sock, addr in clients.items():
                    if str(addr[1]) == target_client_id:
                        sock.send(f"(Private) {addr}: {private_message}".encode())
                        break
            else:
                broadcast(f"{addr}: {message}", client_socket)

        except:
            break

    del clients[client_socket]
    client_socket.close()
    print(f"Connection from {addr} closed.")

def start_chat_server():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind(('localhost', 10000))
    server_socket.listen(5)
    print("Chat server is listening on port 10000...")

    while True:
        client_socket, addr = server_socket.accept()
        threading.Thread(target=handle_client, args=(client_socket, addr)).start()

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

```
Chat server is listening on port 10000...
New connection from ('127.0.0.1', 52687)
New connection from ('127.0.0.1', 52688)
New connection from ('127.0.0.1', 52690)
Connection from ('127.0.0.1', 52690) closed.
New connection from ('127.0.0.1', 52744)
New connection from ('127.0.0.1', 52745)
New connection from ('127.0.0.1', 52746)
```

yter client1-task2 Last Checkpoint: 21 minutes ago

t View Run Kernel Settings Help

Code

```python
import socket
import threading

def receive_messages(client_socket):
    while True:
        try:
            message = client_socket.recv(1024).decode()
            print(message)
        except:
            print("Disconnected from the server.")
            break

def start_client():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect(('localhost', 10000))

    threading.Thread(target=receive_messages, args=(client_socket,)).start()

    while True:
        message = input()
        if message.lower() == "exit":
            break
        client_socket.send(message.encode())

    client_socket.close()

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

```
 CN Lab 03 - Tasks
('127.0.0.1', 52688): CN Lab 03 - Tasks
('127.0.0.1', 52744): CN Lab 03 - Tasks
('127.0.0.1', 52745): CN Lab 03 - Tasks
('127.0.0.1', 52746):  CN Lab 03 - Tasks
```

↑↓ for history. Search history with c-↑/c-↓

yter client2_task_2 Last Checkpoint: 21 minutes ago

t View Run Kernel Settings Help

Code

```python
import socket
import threading

def receive_messages(client_socket):
    while True:
        try:
            message = client_socket.recv(1024).decode()
            print(message)
        except:
            print("Disconnected from the server.")
            break

def start_client():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect(('localhost', 10000))

    threading.Thread(target=receive_messages, args=(client_socket,)).start()

    while True:
        message = input()
        if message.lower() == "exit":
            break
        client_socket.send(message.encode())

    client_socket.close()

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

```
 CN Lab 03 - Tasks
('127.0.0.1', 52744): CN Lab 03 - Tasks
('127.0.0.1', 52745): CN Lab 03 - Tasks
('127.0.0.1', 52746):  CN Lab 03 - Tasks
```

yter client3_task2 Last Checkpoint: 22 minutes ago

t View Run Kernel Settings Help

Code

```python
import socket
import threading

def receive_messages(client_socket):
    while True:
        try:
            message = client_socket.recv(1024).decode()
            print(message)
        except:
            print("Disconnected from the server.")
            break

def start_client():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect(('localhost', 10000))

    threading.Thread(target=receive_messages, args=(client_socket,)).start()

    while True:
        message = input()
        if message.lower() == "exit":
            break
        client_socket.send(message.encode())

    client_socket.close()

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

```
 CN Lab 03 - Tasks
('127.0.0.1', 52745): CN Lab 03 - Tasks
('127.0.0.1', 52746):  CN Lab 03 - Tasks
```

↑↓ for history. Search history with c-↑/c-↓

yter client4_task2 Last Checkpoint: 22 minutes ago

t View Run Kernel Settings Help

Code

```
import socket
import threading

def receive_messages(client_socket):
    while True:
        try:
            message = client_socket.recv(1024).decode()
            print(message)
        except:
            print("Disconnected from the server.")
            break

def start_client():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect(('localhost', 10000))

    threading.Thread(target=receive_messages, args=(client_socket,)).start()

    while True:
        message = input()
        if message.lower() == "exit":
            break
        client_socket.send(message.encode())

    client_socket.close()

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

```
 CN Lab 03 - Tasks
('127.0.0.1', 52746):  CN Lab 03 - Tasks
```

yter client5-task2 Last Checkpoint: 23 minutes ago

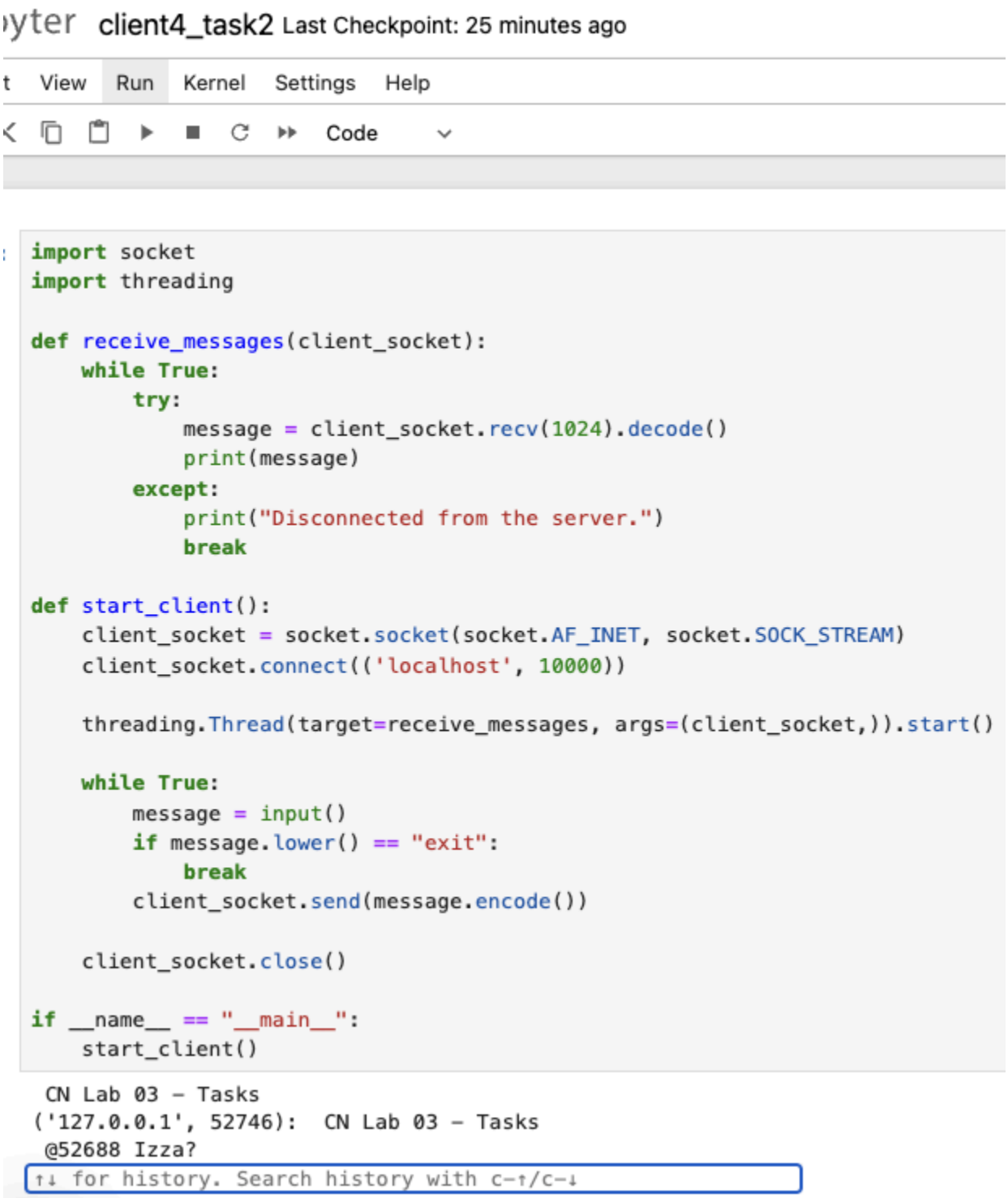
t View Run Kernel Settings Help

Code

```python
import socket
import threading

def receive_messages(client_socket):
    while True:
        try:
            message = client_socket.recv(1024).decode()
            print(message)
        except:
            print("Disconnected from the server.")
            break

def start_client():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect(('localhost', 10000))

    threading.Thread(target=receive_messages, args=(client_socket,)).start()

    while True:
        message = input()
        if message.lower() == "exit":
            break
        client_socket.send(message.encode())

    client_socket.close()

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

```
  CN Lab 03 - Tasks
↑↓ for history. Search history with c-↑/c-↓
```

BONUS TASK: CLIENT-CLIENT



yter client2_task_2 Last Checkpoint: 25 minutes ago

t View Run Kernel Settings Help

Code

```python
import socket
import threading

def receive_messages(client_socket):
    while True:
        try:
            message = client_socket.recv(1024).decode()
            print(message)
        except:
            print("Disconnected from the server.")
            break

def start_client():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect(('localhost', 10000))

    threading.Thread(target=receive_messages, args=(client_socket,)).start()

    while True:
        message = input()
        if message.lower() == "exit":
            break
        client_socket.send(message.encode())

    client_socket.close()

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

```
 CN Lab 03 - Tasks
('127.0.0.1', 52744): CN Lab 03 - Tasks
('127.0.0.1', 52745): CN Lab 03 - Tasks
('127.0.0.1', 52746):  CN Lab 03 - Tasks
(Private) ('127.0.0.1', 52688): Izza?
```

↑↓ for history. Search history with c-↑/c-↓