



Politechnika Wrocławska

---

## Sprawozdanie 4

Krzysztof Zalewa

26.5.2025

# Spis treści

1	Cel skryptu	2
2	Opis skryptu	2
3	Wnioski	4

## 1 Cel skryptu

Cele zajęć laboratoryjnych było przygotowanie prostego czatu przy użyciu pythona. Całe zadanie podzielone jest na dwie części serwer i klienta. Serwer obsługuje połączenia z klientami. Oraz przesyła między nimi wiadomości. Klienci to okna w terminalu obsługiwane przez użytkowników pozwalające na wyświetlanie i wysyłanie wiadomości. Zadanie wykonano przy użyciu bibliotek socket(połączenia) oraz threading(wielowątkowość by połączyć wielu klientów na raz oraz by odbieranie i wysyłanie mogło się dziać równocześnie)

## 2 Opis skryptu

```
1  import socket
2  import threading
3
4  def handle_client(client_socket, client_address):
5      print(f"[NEW CONNECTION] {client_address} connected.")
6
7      while True:
8          try:
9              message = client_socket.recv(1024).decode('utf-8')
10             if not message:
11                 break
12             print(f"[{client_address}] {message}")
13
14             # Broadcast to all clients
15             broadcast(message, client_socket)
16         except:
17             break
18
19     print(f"[DISCONNECTED] {client_address} disconnected.")
20     client_socket.close()
21
22 def broadcast(message, sender_socket):
23     for client in clients:
24         if client != sender_socket:
25             try:
26                 client.send(message.encode('utf-8'))
27             except:
28                 clients.remove(client)
29
30 def start_server():
31     server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
32     server.bind(('127.0.0.1', 8080))
33     server.listen()
```

```

34     print("[SERVER STARTED] Waiting for connections...")
35
36     while True:
37         client_socket, client_address = server.accept()
38         clients.append(client_socket)
39         thread = threading.Thread(target=handle_client, args=(client_socket,
40             ↪ client_address))
41         thread.start()
42         print(f"[ACTIVE CONNECTIONS] {threading.active_count() - 1}")
43
44 clients = []
45
46 if __name__ == "__main__":
47     start_server()

```

### Skrypt 1: Kod servera

```

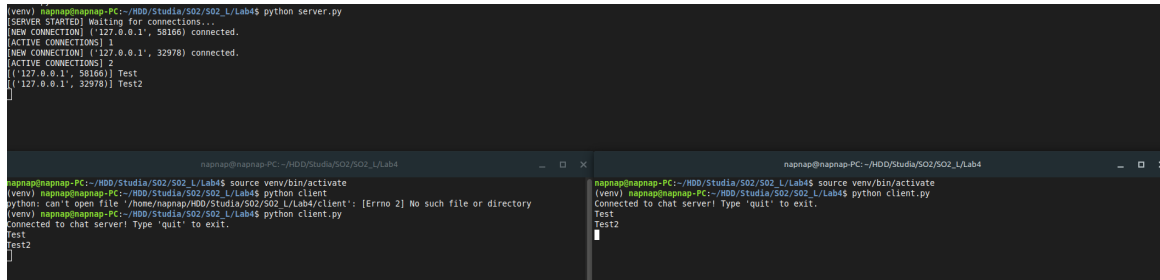
1  import socket
2  import threading
3
4  def receive_messages():
5      while True:
6          try:
7              message = client.recv(1024).decode('utf-8')
8              print(message)
9          except:
10             print("Disconnected from server!")
11             client.close()
12             break
13
14  def send_messages():
15      while True:
16          message = input()
17          if message.lower() == 'quit':
18              client.close()
19              break
20          client.send(message.encode('utf-8'))
21
22  def start_client():
23      global client
24      client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
25      client.connect(('127.0.0.1', 8080))
26
27      print("Connected to chat server! Type 'quit' to exit.")
28
29      receive_thread = threading.Thread(target=receive_messages)
30      receive_thread.start()
31
32      send_thread = threading.Thread(target=send_messages)
33      send_thread.start()
34
35  if __name__ == "__main__":
36      start_client()

```

## Skrypt 2: Kod klienta

### 3 Wnioski

Kod działa poprawnie. Po uruchomieniu serwera klienci mogą się do niego połączyć. A wiadomości wyświetlanie są poprawnie



```
(venv) napnap@napnap-PC:~/HDD/Studia/SO2/L/Lab4$ python server.py
[SERVER STARTED] Waiting for connections...
[NEW CONNECTION] ['127.0.0.1', 35166] connected.
[ACTIVE CONNECTIONS] 1
[NEW CONNECTION] ['127.0.0.1', 32978] connected.
[ACTIVE CONNECTIONS] 2
[{'127.0.0.1', 58166}] Test
[{'127.0.0.1', 32978}] Test2
]
```

```
napnap@napnap-PC:~/HDD/Studia/SO2/L/Lab4$
(venv) napnap@napnap-PC:~/HDD/Studia/SO2/L/Lab4$ source venv/bin/activate
(venv) napnap@napnap-PC:~/HDD/Studia/SO2/L/Lab4$ python client
python: can't open file '/home/napnap/HDD/Studia/SO2/SO2_L/Lab4/client': [Errno 2] No such file or directory
(venv) napnap@napnap-PC:~/HDD/Studia/SO2/L/Lab4$ python client.py
Connected to chat server! Type 'quit' to exit.
Test
Test2
]
```

```
napnap@napnap-PC:~/HDD/Studia/SO2/L/Lab4$
(venv) napnap@napnap-PC:~/HDD/Studia/SO2/L/Lab4$ source venv/bin/activate
(venv) napnap@napnap-PC:~/HDD/Studia/SO2/L/Lab4$ python client.py
Connected to chat server! Type 'quit' to exit.
Test
Test2
]
```

Rysunek 1: Przykład działania. Wiadomość "Test" wysłana z terminala po prawej, "Test1" po lewej