Санкт-Петербургский государственный университет

Факультет прикладной математики – процессов управления

**Отчет №3**

по дисциплине «Функциональное программирование»

**Разработка асинхронного чат-сервера**

Автор работы: Цыбина А.И.

Группа: 22.Б15-пу

Преподаватель: Киямов Ж. У.

Санкт-Петербург, 2023

## Оглавление

[1. Цель работы 3](#_Toc150962359)

[2. Теоретическая часть 3](#_Toc150962360)

[3. Алгоритм метода 3](#_Toc150962361)

[4. Описание программы 4](#_Toc150962362)

[5. Рекомендации программиста 5](#_Toc150962363)

[6. Контрольный пример 5](#_Toc150962364)

[7. Вывод 6](#_Toc150962365)

[8. Список использованной литературы 6](#_Toc150962366)

# Цель работы

Разработать чат-сервер с поддержкой асинхронной работы, способный обслуживать одновременно множество пользователей и обеспечивающий мгновенный обмен сообщениями в реальном времени.

# Теоретическая часть

Задача создания асинхронного чат-сервера, способного обслуживать множество клиентов одновременно и обеспечивающего обмен сообщениями в режиме реального времени, представляет собой актуальное исследование в области разработки сетевых приложений. Асинхронность здесь играет ключевую роль, позволяя эффективно управлять одновременными подключениями без блокировки основного потока выполнения. Этот подход особенно важен в контексте чат-сервера, где быстрая и отзывчивая передача сообщений между клиентами является неотъемлемой частью функционала.

Основной целью создания такого сервера является обеспечение пользователей возможностью общаться в реальном времени, создавая динамичное и многопользовательское окружение. Для этого необходимо разработать механизм обработки асинхронных событий, таких как поступление новых сообщений от клиентов, и мгновенной рассылки этих сообщений всем участникам чата. Эффективное управление подключениями и обмен сообщениями в асинхронном режиме становится ключевым компонентом достижения цели.

Важным аспектом в разработке асинхронного чат-сервера также является обеспечение масштабируемости и стабильности работы даже при значительной активности пользователей. Это включает в себя реализацию многопоточности для обработки множества подключений одновременно и обработку возможных исключений, гарантируя непрерывную работу сервера. Такой подход открывает перспективы для создания высокопроизводительных и надежных сетевых чат-приложений, способных удовлетворить современные требования многопользовательского общения в режиме реального времени.

# Описание программы

Программная реализация написана на языке python 3.11 с использованием

следующих пакетов: asyncio, websockets, json, tkinter, datetime.

Таблица 3.1. Описание функций

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя функции | Аргументы | Асинхронная | Класс | Описание функции |
| handle\_client | Сокет | Да | ChatServer | Обрабатывает подключение клиента |
| handle\_message | Отправитель, сообщение | Да | ChatServer | Обрабатывает тип поступившего сообщения |
| handle\_join | Отправитель, дата | Да | ChatServer | Обрабатывает добавление клиента в комнату |
| handle\_chat\_message | Дата | Да | ChatServer | Обрабатывает отправку сообщения в комнату |
| handle\_private\_message | Дата | Да | ChatServer | Обрабатывает отправку личного сообщения |
| broadcast | Комната, сообщение | Да | ChatServer | Вспомогательная функция для рассылки сообщения всем участникам комнаты |
| server | Хост, порт | Да | ChatServer | Осуществляет создание локального сервера |
| connect |  | Да | ChatClientGUI | Инициализирует клиента и запускает процесс подключения его к серверу |
| send\_message\_room |  | Нет | ChatClientGUI | Посылает сигнал классу клиента для отправки сообщения в комнату |
| send\_message\_user |  | Нет | ChatClientGUI | Посылает сигнал классу клиента для отправки сообщения пользователю |
| join\_room |  | Нет | ChatClientGUI | Посылает сигнал классу клиента для добавления в комнату |
| run |  | Да | ChatClientGUI | Асинхронно запускает графический интерфейс |
| connect |  | Да | ChatClient | Подключает клиента к серверу |
| receive\_message |  | Да | ChatClient | Обрабатывает входящее сообщение |
| send\_message\_room | Название комнаты, соощение | Да | ChatClient | Отправляет сообщение в комнату на сервер |
| send\_message\_user | Имя получателя, сообщение | Да | ChatClient | Отправляет сообщение пользователю на сервер |
| join\_room | Название комнаты | Да | ChatClient | Отправляет запрос на вступление в комнату на сервер |

# Рекомендации программиста

Для запуска программы необходима 64-битная операционная система Windows, Linux или macOS. Для работы с кодом необходима среда разработки, совместимая с python 3.11 и библиотеки asyncio, websockets, json, tkinter, datetime.

Ссылка на репозиторий с кодом:

https://github.com/sastsy/func-prog/tree/main/task3

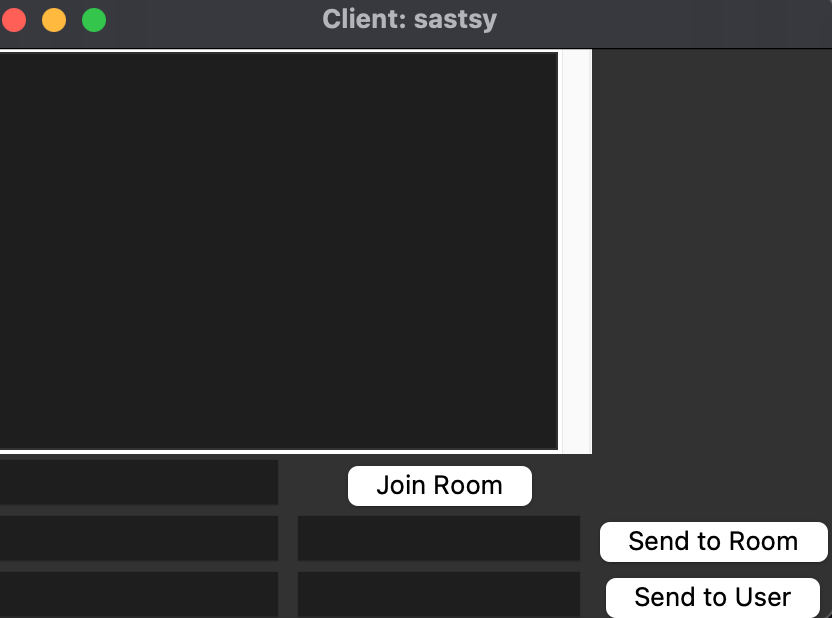
# Рекомендации пользователя

При запуске программы пользователю следует ввести свое имя. Далее открывается графический интерфейс. В названии окна отображается имя пользователя. В большом поле отображаются сообщения пользователя с именем, датой отправки и местом получения.

Далее в интерфейсе есть поле для ввода названия комнаты и кнопка “Join Room” чтобы присоединиться к этой комнате.

Далее идут поля для воода названия комнаты и сообщения, кнопка “Send to Room" запускает процесс отправления введенного сообщения в эту комнату.

Внизу находятся поля для ввода имени пользователя и сообщения к нему, кнопка “Send to User” запускает процесс отправки введенного сообщения этому пользователю.



# Контрольный пример

В данном разделе представлен контрольный пример, демонстрирующий

работу программы.

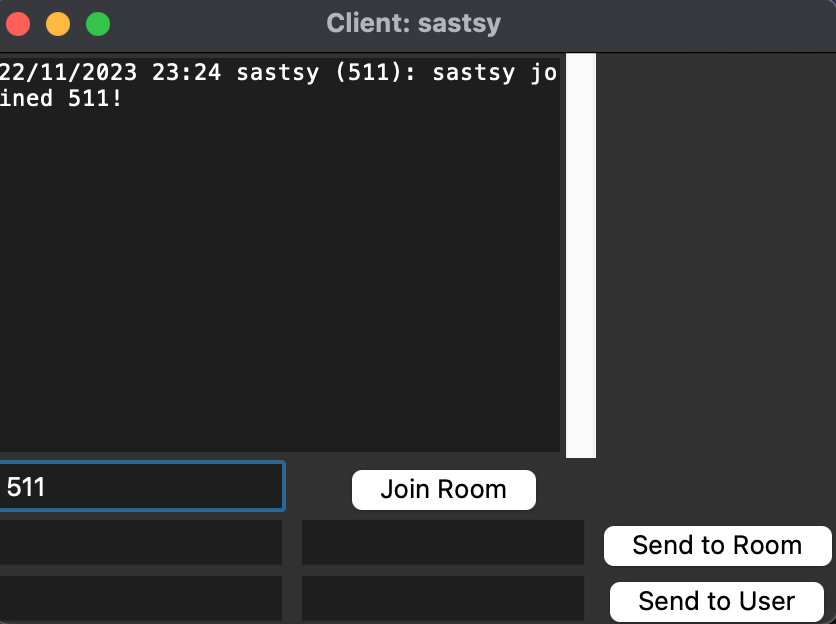


Рисунок 6.1 Присоединение к комнате

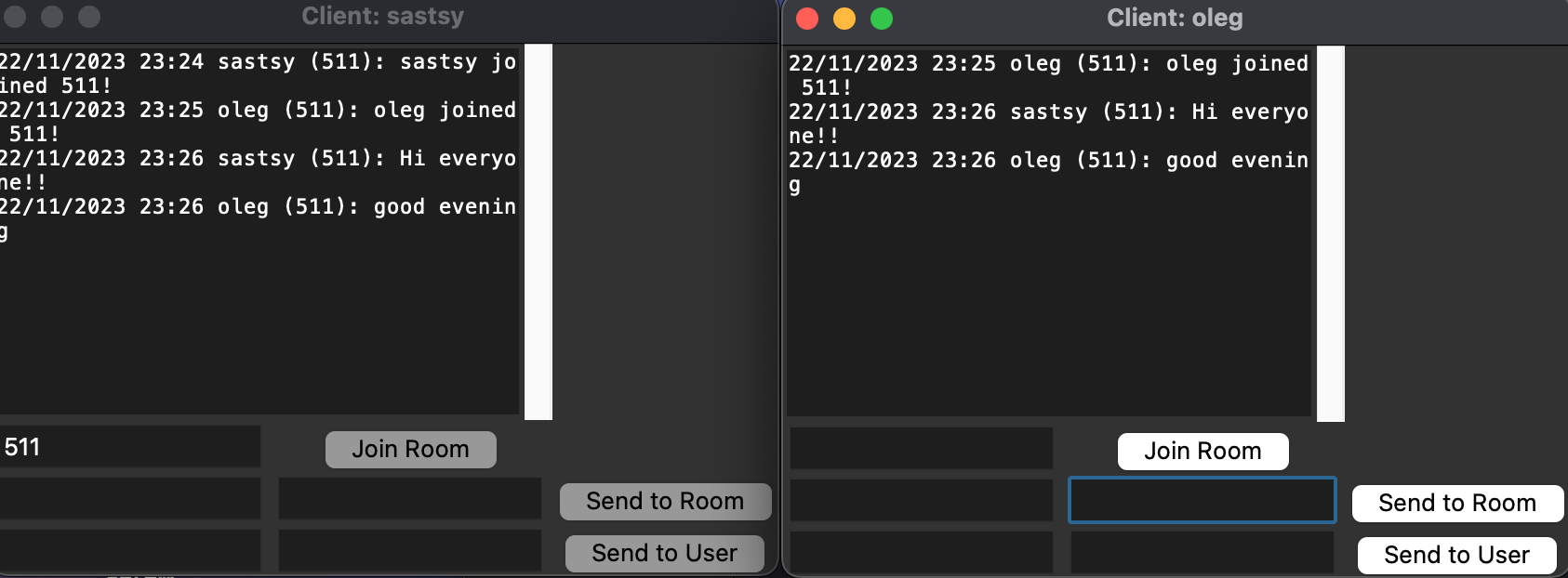


Рисунок 6.2 Отправка сообщения в комнату

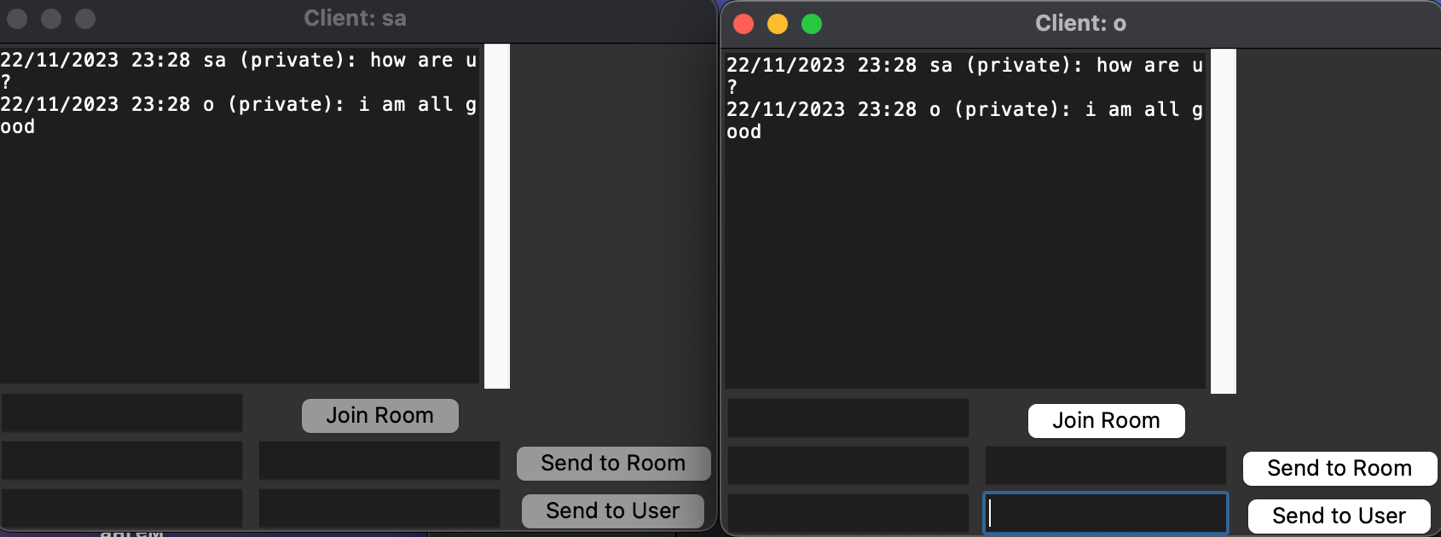


Рисунок 6.3 Отправка личного сообщения

# Вывод

В ходе выполнения данной работы были изучены основные методы разработки асинхронного чат-сервера.

# Список использованной литературы

1. Документация библиотеки asyncio: https://docs.python.org/3/library/asyncio.html
2. Документация библиотеки websockets: https://docs.python.org/3/library/websockets.html
3. Документация библиотеки tkinter: https://docs.python.org/3/library/tkinter.html