МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский политехнический университет Петра Великого»

Институт компьютерных наук и кибербезопасности

Высшая школа искусственного интеллекта

Направление: 02.03.03 Математическое обеспечение и администрирование информационных систем

**Отчет о выполнении курсовой работы**

**«Client-Server messenger»**

по дисциплине «Основы программирования и алгоритмизации»

направление 02.03.03 — «Математическое обеспечение и администрирование информационных систем»

Студент,

группа 5130203/30002

В. Р. Ершов

Руководитель

Йовановски Ненад

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г.

Санкт-Петербург, 2024

**Оглавление**

[**Введение** 2](#_Toc169134702)

[**Постановка задач** 2](#_Toc169134703)

[**Реализация** 3](#_Toc169134704)

[**client.h** 3](#_Toc169134705)

[**server.h** 3](#_Toc169134706)

[**Тестирование** 6](#_Toc169134707)

[**Подготовка и запуск** 6](#_Toc169134708)

[**Отправка сообщений клиенту** 6](#_Toc169134709)

[**Отправка сообщения всем клиентам** 7](#_Toc169134710)

[**Получение содержимого файла с сервера** 8](#_Toc169134711)

[**Отправка файла на сервер** 8](#_Toc169134712)

[**Вывод** 9](#_Toc169134713)

# **Введение**

В данной курсовой работе мы реализовываем отправку сообщений от клиента к серверу и от сервера к клиенту. На основе содержимого сообщений мы будем выбирать, какая именно функция сейчас будет запускаться. Сервер реализован в файлах server.cpp и server.h, клиент соответственно реализован в файлах client.cpp и client.h. Вся работа пройдет в заголовочных файлах, так как в .cpp файлах реализована сама работа сервера и клиента, а не наши задачи.

# **Постановка задач**

1. Клиент может отправить конкретному клиенту сообщение (через сервер). Если второй клиент онлайн, то должен получить сообщение. Если второй клиент не онлайн, возвращаем ошибку первому клиенту.

2. Клиент может всем одновременно отправить сообщение.

3. Клиент может запросить содержимое файла с сервера. Если файла не существует, возвращаем ошибку клиенту.

4. Клиент может отправлять файлы на сервер.

# **Реализация**

## **client.h**

Начнем отчёт о курсовой с единственного изменения в client.h файле. В этом файле я реализовал отправку сообщения определенного типа от клиента серверу с помощью функции \_send\_to\_server. Я отправляю типизированное сообщение, а именно <текст>:<текст>, если вводится сообщение без двоеточия, код выдаст пользователю ошибку. Вся отправка реализована в функции void my\_client\_code():

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

## **server.h**

Работа в server.h полностью проходит в функции void recvFromClient(int clientNo, string message). Эта функция принимает на вход номер клиента, который отправил сообщение на сервер, а также само сообщение. Вызывается эта функция только когда сервер принимает какое-то сообщение.

Для начала на самом сервере выводится номер клиента и не отформатированное сообщение. Далее полученное сервером сообщение от клиента мы делим на до и после двоеточия (:). То, что до будет у нас string clNum, а то что после – string msg.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Дальше рассматриваем несколько вариантов развития событий, зависящих от того, какую функцию клиент прописал до двоеточия. Первый случай, когда до двоеточия никакой из «текстов» => у нас там номер клиента, которому хотим отправить сообщение. Тогда проверяем, есть ли этот клиент на сервере, и если есть отправляем ему отформатированное сообщение с помощью функции \_sendToClient(clNo, nummsg);

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, снимок экрана

Автоматически созданное описание

Далее рассмотрим случай, когда клиент ввёл file:<file\_name>. Эта функция подразумевает «отправку» файла от клиента на сервер. Для начала мы находим нужный файл с помощью прописанного заранее пути + имени файла и копируем все его содержимое в переменную contfile соблюдая все пробелы и переносы строки. Далее на сервере создаем файл с именем нашего файла + подписи \_copy. В этот файл мы записываем все содержимое изначального файла. Прописываем ошибки для случаев, когда файл на клиенте отсутствует, а так же случай, когда файл с данным именем не может создаться по какой-либо причине. При успешной отправке файла на сервер отправляем клиенту что файл успешно доставлен, а на сервер сообщение о создании копии такого-то файла с новым названием. На этом отправка файла завершена.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Следующим шагом рассмотрим случай, когда клиент вводит до двоеточия getfile. Эта функция подразумевает получение СОДЕРЖИМОГО файла, который лежит на сервере клиентом. Для этого используем вышеописанный метод для записи в переменную msgfile всего содержимого файла, который находится на сервере. Из-за того, что мы работает с файлом, который лежит на сервере в server.h нам не обязательно передавать в ifile.open(msg) полный путь файла. Прописываем ошибку на случай, если такого файла нет на сервере и если все хорошо то отправляем клиенту содержимое файла с помощью функции \_sendToClient().

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Самый последний случай будет отражать функцию all. Это функция отправки сообщения от одного клиента всем, кто щас онлайн. Для этого подключаем к нашему server.h файлу вектор, в котором хранятся данные, кто онлайн:

Изображение выглядит как текст, Шрифт, снимок экрана, Графика

Автоматически созданное описание

Далее, отправляем сообщение каждому клиенту с помощью \_sendToClient() из этого вектора с текстом после двоеточия, параллельно проверяя, онлайн ли до сих пор этот клиент и не отправляем ли мы сами себе сообщение: Изображение выглядит как текст, Мультимедийное программное обеспечение, программное обеспечение, снимок экрана

Автоматически созданное описание

Вот как выглядит server.h файл функция recvFromClient() в свернутом виде: Изображение выглядит как текст, Мультимедийное программное обеспечение, Шрифт, снимок экрана

Автоматически созданное описание

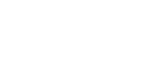
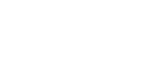
На этом отчет о курсовой заканчивается, в целом мне очень понравилось, дальше буду сосредотачиваться на практике!

# **Тестирование**

## **Подготовка и запуск**

Для начала запустим ПЕРВЫМ server.exe, после которого запустим трех пользователей (client.exe) и запомним в каким порядке они запускались. Проверим, все ли клиенты подключились и нет ли ошибки. Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание



## **Отправка сообщений клиенту**

Отправка сообщения от первого ко второму, от второго к третьему и от третьего к первому клиенту:

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

## **Отправка сообщения всем клиентам**

Отправка сообщения всем клиентам, которые сейчас онлайн сначала первым клиентом, потом вторым клиентом и в самом конце третьим клиентом:

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

## **Получение содержимого файла с сервера**

Получение СОДЕРЖИМОГО файла, который лежит в папке сервера с сохранением всех пробелов и переносов строки, а так же проверка ошибки при вводе названия несуществующего файла:

Изображение выглядит как текст, снимок экрана, черный, Шрифт

Автоматически созданное описание

## **Отправка файла на сервер**

Отправка файла «clientfile.txt», который находится в папке клиента на сервер и создание на сервере файла с именем «clientfile\_copy.txt»(проверим с помощью функции getfile), а так же проверка ситуации, когда вызывается ошибка:

Изображение выглядит как текст, снимок экрана, черный

Автоматически созданное описание

# **Вывод**

В заключение мы научились работать со строками, а именно обрабатывать их и выполнять определенные команды в зависимости от содержимого строки. Так же мы научились работать с фалами и правильно «доставать» их содержимое в одну переменную. Все это было под оболочкой некого локального мессенджера, который работал как клиент-сервер-клиент.